The differential $\lambda\mu$-calculus

Lionel Vaux
Institut de Mathématiques de Luminy
vaux@iml.univ-mrs.fr

6th December 2006

Abstract

We define a differential $\lambda\mu$-calculus which is an extension of both Parigot's $\lambda\mu$-calculus and Ehrhard-Régnier's differential λ-calculus. We prove some basic properties of the system: reduction enjoys Church-Rosser and simply typed terms are strongly normalizing.

1 Introduction

Thomas Ehrhard and Laurent Régnier showed in [ER03] how to extend λ-calculus by means of formal derivatives of λ-terms, following the well-known rules of usual differential calculus. This differential λ-calculus involves a strong relationship between linearity in the logical or computational sense (the argument of a function is used exactly once by this function) and linearity in the usual algebraic sense. As linearity is the founding notion derivation is built upon, this relationship is made explicit by the interaction between the structural rules of linear logic (contraction, dereliction and weakening), and some new semantic *co-structural* constructions that were introduced by Ehrhard in [Ehr01] and [Ehr04]. Structural constructs are used to manage arguments in the denotational semantics of λ-calculus. Co-structural constructs allow to differentiate morphisms: they form the semantic basis of the differential part of the calculus.

Michel Parigot introduced $\lambda\mu$-calculus in [Par92]: this extension of λ-calculus lifts the Curry-Howard correspondence from intuitionistic logic to classical logic. In this setting, proofs in classical natural deduction (also introduced in [Par92]) are mapped to terms with several outputs: this accounts for the fact that classical proofs come with multiple conclusion formulas, which may be contracted or weakened. In [Lau98, Lau03], Olivier Laurent showed how to encode $\lambda\mu$-calculus into polarized linear logic. Polarized linear logic is roughly linear logic where all formulas are polarized, and weakening and contraction are allowed on every negative formula. This enlightens the fact that $\lambda\mu$-calculus involves more general structural rules than ordinary λ-calculus.

In short, differential λ-calculus introduces some new means of studying structural rules through differential constructions, whereas $\lambda\mu$-calculus gives a syntax to an extension of these rules. This paper is a first attempt to uncover possible interactions between differential constructions and extended structural rules: we define a differential $\lambda\mu$-calculus, which extends both $\lambda\mu$-calculus and differential λ-calculus in a common framework. In particular, we give a computational meaning to the derivative of a μ-abstraction.

In the following, we quickly survey $\lambda\mu$-calculus and differential λ-calculus, giving intuitions which, we hope, will make the definition of differential $\lambda\mu$-calculus seem natural. For that purpose, we allow ourself some inaccuracies, keeping things fictitiously simple.

1.1 $\lambda\mu$-calculus

Classical logic and multiple conclusions It has long been known that one could switch from an intuitionistic logical system to a classical one by allowing structural operations on conclusions: in sequent calculus, for instance, LK is exactly LJ with any number of formulas in the right hand side of each sequent (rather than at most one), and contraction and weakening allowed on that side. The algorithmic interpretation of this extension has been clear since Timothy Griffin's proposal to type the

C operator of Felleisen with type $\neg\neg A \to A$: classical constructs correspond to control operators such as `call/cc`.

Until classical natural deduction and $\lambda\mu$-calculus, however, functional languages augmented with control operators were not satisfatory systems in the proofs-as-programs point of view: control operators were introduced together with new typing rules, but without changing the deduction system in which they were expressed, namely usual natural deduction. Since the restriction to exactly one output type (or conclusion) prevents the most natural transformations on proofs, those early attempts failed in providing a satisfactory notion of reduction: they rather forced a reduction strategy.

Parigot introduced $\lambda\mu$-calculus in [Par92]. It is an extension of λ-calculus, lifting the Curry-Howard correspondence from intuitionistic logic to classical logic. The associated logical system is a restriction of free deduction [Par91], called classical natural deduction, also introduced in [Par92]. As such, classical natural deduction enjoys an internal notion of cut, similar to that of intuitionistic natural deduction, although it operates on classical sequents: proofs have multiple conclusion formulas, and structural rules (weakening and contraction) are allowed on these. The $\lambda\mu$-calculus is the algorithmic pure calculus extracted from classical natural deduction in the Curry-Howard correspondence. As a consequence, it is a calculus of terms with multiple outputs.

Terms with multiple outputs Each $\lambda\mu$-term has at most one active output, which is currently evaluated, together with several auxiliary outputs, identified by names. Two new constructs are introduced, in order to manage these.

- *Naming*: if s is a term with an active output, and α is a name, then $[\alpha]\, s$ is a term; the former active output becomes an auxiliary output with name α. If one already existed with the same name, they are merged together: this accounts for logical contraction on conclusions.
- *Abstraction on names (called μ-abstraction)*: if s is a term without an active output, and α is a name, then $\mu\alpha\, s$ is a term; the former auxiliary output with name α becomes the active output. Hence there is no longer an output with name α: μ is a binder. If there was no auxiliary output with name α, the active output now points to nothing: this accounts for logical weakening on conclusions.

Hence terms are given by the following grammar:

$$\begin{aligned} s, t &::= x \mid \lambda x\, s \mid (s)\, t \mid \mu\alpha\, \nu \\ \nu &::= [\alpha]\, s \end{aligned}$$

where we distiguish terms with an active output, or simply terms, from terms without an active output, or simply named terms.

Remark 1.1. Quite often in the literature (*e.g.*, in [Par92]), terms and named terms are denoted by the same symbols. Moreover, there is a variant of $\lambda\mu$-calculus with only one syntactic group: one can form $\mu\alpha\, s$ and $[\alpha]\, s$ whatever the shape of s. In [Sau05], however, Alexis Saurin shows that Parigot's $\lambda\mu$-calculus and this alternative syntax, which he calls $\Lambda\mu$-calculus, are distinct calculi. In particular, he proves that the separation property holds in $\Lambda\mu$-calculus, whereas it fails in $\lambda\mu$-calculus (see also [DP01]). For this reason, we think it better to avoid confusion by denoting differently terms and named terms.

Reducing terms with several outputs The reduction of $\lambda\mu$-calculus is given by the usual β-reduction, together with a new reduction rule, with redexes of shape $(\mu\alpha\, \nu)\, t$. Recall that the term $\mu\alpha\, \nu$ is obtained by choosing the auxiliary output α of ν as the active output. Under α are merged any number (possibly zero) of active outputs of subterms of ν. The new reduction allows for arguments of $\mu\alpha\, \nu$ to be passed to those subterms.

It is defined by the reduction rule:

$$(\mu\alpha\, \nu)\, t \rightsquigarrow \mu\alpha\, (\nu)_\alpha\, t,$$

where the named term $(\nu)_\alpha\, t$ is intuitively ν, in which all those active outputs merged into α are provided with a copy of t as an argument: shortly, it is ν applied to t through α. This amounts to replace inductively each named subterm of ν of shape $[\alpha]\, u$ with the term $[\alpha]\, (u)\, t$; hence the usual notation $(\nu)_\alpha\, t = \nu\, [[\alpha]\, (u)\, t/[\alpha]\, u]$.

Remark 1.2. Here, we break conventional notations for several reasons. First, our notation is more concise, which is not to be neglected. Moreover, this notation fits well with intuition: as terms may have several auxiliary outputs together with the active one, $(s)_\alpha\, t$ denotes s applied to t through α, the same as $(s)\, t$ denotes s applied to t through its active output. Last, the use of this notation helps in understanding our definition of the derivative of a μ-abstraction, to be introduced in differential $\lambda\mu$-calculus: see definitions 2.8 and 2.9 in subsection 2.4.

A classical type system The intended meaning of the new syntactic constructs of $\lambda\mu$-calculus might be better understood in a typed setting. In typed $\lambda\mu$-calculus, one derives judgements of shape

$$x_1 : A_1, \ldots, x_n : A_n \vdash s : A \mid \alpha_1 : B_1, \ldots, \alpha_p : B_p$$

and

$$x_1 : A_1, \ldots, x_n : A_n \vdash \nu \mid \alpha_1 : B_1, \ldots, \alpha_p : B_p$$

where inputs are identified by variables x_i with types A_i and auxiliary outputs are identified by names α_j with types B_j. If it exists, the active output has type A. The typing rules of simply typed λ-calculus are easily reproduced in this setting, leaving names and auxiliary outputs untouched. The rules for naming and μ-abstraction closely match our previous statements:

$$\dfrac{\Gamma \vdash s : A \mid \Delta}{\Gamma \vdash [\alpha]\, s \mid \alpha : A, \Delta}$$

and

$$\dfrac{\Gamma \vdash \nu \mid \Delta}{\Gamma \vdash \mu\alpha\, \nu : A \mid \Delta \setminus \{\alpha : A\}}$$

where α has type A in Δ or α is not declared in Δ. If α is declared in Δ, one recognizes contraction in the first rule. Otherwise, one recognizes weakening in the second rule.

One can derive:

$$\dfrac{\dfrac{\begin{matrix} d \\ \vdots \\ \Gamma \vdash \nu \mid \alpha : A \to B, \Delta \end{matrix}}{\Gamma \vdash \mu\alpha\, \nu : A \to B \mid \Delta} \quad \begin{matrix} d' \\ \vdots \\ \Gamma' \vdash t : A \mid \Delta' \end{matrix}}{\Gamma, \Gamma' \vdash (\mu\alpha\, \nu)\, t : B \mid \Delta, \Delta'}$$

where the auxiliary output α in ν has type $A \to B$. The reduction

$$(\mu\alpha\, \nu)\, t \rightsquigarrow \mu\alpha\, (\nu)_\alpha\, t$$

then gives t as an argument of type A to those subterms of type $A \to B$ that were merged into α. In $(\nu)_\alpha\, t$, the auxiliary output with name α has thus type B, in accordance with subject reduction: one can also derive

$$\dfrac{\begin{matrix} d'' \\ \vdots \\ \Gamma, \Gamma' \vdash (\nu)_\alpha\, t \mid \alpha : B, \Delta, \Delta' \end{matrix}}{\Gamma, \Gamma' \vdash \mu\alpha\, (\nu)_\alpha\, t : B \mid \Delta, \Delta'}$$

where d'' is obtained from d by replacing inductively each rule of form

$$\dfrac{\Gamma'' \vdash u : A \to B \mid \Delta''}{\Gamma'' \vdash [\alpha]\, u \mid \alpha : A \to B, \Delta''}$$

with

$$\dfrac{\dfrac{\Gamma'' \vdash u : A \to B \mid \Delta'' \quad \begin{matrix} d' \\ \vdots \\ \Gamma' \vdash t : A \mid \Delta' \end{matrix}}{\Gamma'', \Gamma' \vdash (u)\, t : B \mid \Delta'', \Delta'}}{\Gamma'', \Gamma' \vdash [\alpha]\, (u)\, t \mid \alpha : B, \Delta'', \Delta'}\ .$$

In [Par92], Parigot proves confluence of $\lambda\mu$-calculus, and a proof of strong normalization in the typed case can be found in [Par97].

Like λ-calculus, $\lambda\mu$-calculus can be given a denotational semantics in a variant of linear logic. This variant is Laurent's polarized linear logic [Lau98], *i.e.* linear logic in which all formulas are polarized, and contraction and weakening are allowed on every negative formula.

1.2 Differential λ-calculus

Linearity in λ-calculus In the mainstream mathematics, linearity is a fundamental notion of algebra. In λ-calculus, however, as in proof theory in general, linearity is a completely different concept at first sight. Jean-Yves Girard's linear logic [Gir87], by decomposing intuitionistic implication, made this syntactic concept of linearity prominent. As we stated informally before, a term is said to be linear if it uses its argument exactly once. This vague definition can be made more precise:

- in a term which is only a variable x, that occurrence of variable is in linear position;
- in an abstraction $\lambda x\, s$, the linear positions are those of the abstracted subterm s, and the abstraction itself;
- in an application $(s)\, t$, the linear positions are those of the function subterm s, and the application itself.

In particular, application is linear in the function but not in the argument. This is to be related with head reduction and memory management: those subterms that are in linear position are evaluated exactly once in the head reduction, they are not copied nor discarded.

These remarks hint that both notions of linearity actually coincide in the following sense. Algebraic linearity is generally thought of as commutation to sums. Moreover, it is well known that the space of all functions from some set to, *e.g.*, some fixed vector space is itself a vector space, with operations on functions defined pointwise: for instance, the sum of two functions is defined by $(f+g)(x) = f(x) + g(x)$. In [Ehr01] and [Ehr04], Ehrhard introduced denotational models of linear logic where formulas are interpreted as particular vector spaces and proofs corresponding to λ-terms are interpreted as functions defined by power series on these spaces. This offers serious grounding to the introduction of sums of terms in the λ-calculus, subject to the following two identities:

$$\begin{aligned} (s+t)\, u &= (s)\, u + (t)\, u \\ \lambda x\, (s+t) &= \lambda x\, s + \lambda x\, t \end{aligned}$$

We recover the fact that application is linear in the function and not in the argument, in accordance with the computational notion of linearity.

Although it is argued, in the introduction of [ER03], that non-deterministic choice (in the sense of [dLP95]) provides a possible computational interpretation of the sum, this is not explicitly introduced in the calculus, if only because it would break confluence. We will rather allow the formation of formal sums of terms, or even linear combinations with coefficients in some fixed rig (typically $\mathbf{N}$): the set of terms will be a module over that rig. In particular, well known identities between linear combinations hold: for instance, we consider $s+t$ and $t+s$ denote the same term; similarly, if a and b are coefficients, then $as + bs$ and $(a+b)s$ denote identical terms. Naturally, the above two identies are part of that metatheory: *e.g.*, $(s+t)\, u$ and $(s)\, u + (t)\, u$ are two writings for the same term.

Remark 1.3. One could introduce the previous identifications as rewriting rules *inside* the calculus rather than at a metatheoretical level. However, the syntax of the calculus we present in sections 2 and 3 is already quite complex, and we think it better not to add to that complexity (which does not mean we needn't be cautious when dealing with sums). The reader may refer to [Vau06] for a presentation of λ-terms with sums handled inside the calculus.

The derivative of a term In the abovementioned models of linear logic by Ehrhard, all functions are differentiable. The differential λ-calculus of [ER03] brings this property back to the syntax: one can give an account of differentiation in λ-calculus in accordance with the properties of these *analytic* denotational models.

Let's fix some general intuitions on differentiation first. Let $f : E \longrightarrow F$ be a differentiable function, its derivative f' is generally thought of as a function from E to the space of linear functions $E \multimap F$:

$f' : E \longrightarrow (E \multimap F)$. Then, if $a \in E$, $f'(x) \cdot a$ is read as the linear application of $f'(x)$ to a, *i.e.* the derivative of f at point x along a. The function $x \mapsto f'(x) \cdot a$ is a map from E to F, which depends linearly on a: call it the derivative of f along a, which we can denote by $\mathrm{D}\, f \cdot a : E \longrightarrow F$.

Keeping in mind these considerations about differentiation, and in the light of the aforementioned analytic models of linear logic, one can extend the ordinary constructions of λ-calculus as follows.

First, a new syntactic construct is introduced: if s and t are terms, then $\mathrm{D}\, s \cdot t$ is a term, called the derivative of s along t. It is linear with respect to s and t, *i.e.* s and t are in linear position in $\mathrm{D}\, s \cdot t$. Recall that the set of terms is endowed with a structure of module over a fixed commutative rig R: along with usual λ-terms and derivatives, one can form linear combinations of terms, with coefficients in R. Terms are then given by the following grammar:

$$s, t ::= x \mid \lambda x\, s \mid (s)\, t \mid \mathrm{D}\, s \cdot t \mid 0 \mid as + bt$$

where a and b range over R. Linear positions commute to linear combinations:

$$\begin{aligned}
\lambda x \left(\sum_{i=1}^{n} a_i s_i \right) &= \sum_{i=1}^{n} a_i \lambda x\, s_i \\
\left(\sum_{i=1}^{n} a_i s_i \right) t &= \sum_{i=1}^{n} a_i\, (s_i)\, t \\
\mathrm{D} \left(\sum_{i=1}^{n} a_i s_i \right) \cdot \left(\sum_{j=1}^{p} b_j t_j \right) &= \sum_{i,j} a_i b_j \mathrm{D}\, s_i \cdot t_j
\end{aligned}$$

and usual module equations on linear combinations also hold.

The term $\mathrm{D}\, s \cdot t$ is intuitively the linear application of s to t: s is provided with exactly one linear copy of t. If s has a function type $A \to B$ and t has type A, then $\mathrm{D}\, s \cdot t$ has the same function type as s:

$$\frac{\Gamma \vdash s : A \to B \quad \Gamma' \vdash u : A}{\Gamma, \Gamma' \vdash \mathrm{D}\, s \cdot t : A \to B} \, .$$

Hence derivative does not decrease the type.

Differentiation If s is a function, *i.e.* a λ-abstraction $s = \lambda x\, t$, then $\mathrm{D}\, s \cdot u$ reduces to a function which is like s, with one linear call to its argument replaced with u. Here, the use of sums interpreted as non-deterministic choice arise naturally: we write the reduction rule

$$\mathrm{D}\, \lambda x\, t \cdot u \rightsquigarrow \lambda x \left(\frac{\partial t}{\partial x} \cdot u \right)$$

where $\frac{\partial t}{\partial x} \cdot u$ stands for the sum of all possible terms obtained by replacing one linear occurrence of x in s with u. More formally, we define $\frac{\partial s}{\partial x} \cdot u$ by induction on s:

$$\begin{aligned}
\frac{\partial y}{\partial x} \cdot u &= \begin{cases} u \text{ if } x = y \\ 0 \text{ otherwise} \end{cases} \\
\frac{\partial \lambda y\, s}{\partial x} \cdot u &= \lambda y \left(\frac{\partial s}{\partial x} \cdot u \right) \\
\frac{\partial (s)\, t}{\partial x} \cdot u &= \left(\frac{\partial s}{\partial x} \cdot u \right) t + \left(\mathrm{D}\, s \cdot \left(\frac{\partial t}{\partial x} \cdot u \right) \right) t \\
\frac{\partial \mathrm{D}\, s \cdot t}{\partial x} \cdot u &= \mathrm{D} \left(\frac{\partial s}{\partial x} \cdot u \right) \cdot t + \mathrm{D}\, s \cdot \left(\frac{\partial t}{\partial x} \cdot u \right) \\
\frac{\partial as + bt}{\partial x} \cdot u &= a \frac{\partial s}{\partial x} \cdot u + b \frac{\partial t}{\partial x} \cdot u
\end{aligned}$$

For instance, since derivative is bilinear, a linear occurrence of x in $\mathrm{D}\, s \cdot t$ is a linear occurrence of x in either s or t: we just sum over those two possibilities. Since all occurences of x in s need not be in linear position, one has to introduce some linearization of applications: in term $(s)\, t$, x can occur in the function s, which is in linear position, or in the argument t, which is not. In that last case, we linearize application on the fly: the substitution is performed in one linear copy of the argument.

This is very similar to the well known formula for the derivative of the composition of two functions:

$$\begin{aligned}(f \circ g)'(x) &= f'(g(x)) \times g'(x) \\ &= (\mathrm{D}\, f \cdot g'(x))(g(x))\end{aligned}$$

with the notations introduced before.

The reduction of differential λ-calculus is given by the usual β-reduction together with the abovedefined differential reduction rule. In [ER03], Ehrhard and Régnier prove it is confluent, and simply typed terms are strongly normalizing. Moreover, differential λ-calculus has naturally a denotational semantics in those models of linear logic defined in [Ehr01, Ehr04], which gives rise to standard constructions when restricted to ordinary λ-calculus.

Remark 1.4. Ehrhard and Régnier actually defined the derivative $\mathrm{D}_i\, s \cdot u$ of the i-th abstraction of s along u, where s and u are terms. This is in conflict with the intrinsic currying of λ-calculus; indeed, one can encode $\mathrm{D}_{i+1}\, s \cdot u$ by $\lambda x_1 \dots \lambda x_i\, (\mathrm{D}_1\, s \cdot u)\, x_1 \dots x_i$. Up-to the extension of syntactic constructs introduced in the beginning of [ER03, Section 1], this encoding amounts to η-expansion only. It is then legitimate to get rid of D_{i+1} and define only $\mathrm{D}_1\, s \cdot u$, which we write $\mathrm{D}\, s \cdot u$. All our work is successfully carried in this setting.

1.3 Outline of the differential $\lambda\mu$-calculus

Motivations Differential λ-calculus provides a powerful framework to study the nature of structural rules. From a computer scientist's point of view, derivatives allow for a precise investigation of the use of arguments by λ-terms, *i.e.* purely functional programs. In particular, in [ER04] and [ER05], Ehrhard and Régnier obtain very deep results relating the summands of the Taylor expansion of a pure λ-term with executions of this term in a Krivine machine.

One may want to extend these results to non-purely functional programming languages. $\lambda\mu$-calculus is a good candidate: it can be run into a Krivine abstract machine [SR98, dG98], and it has a quite simple denotational semantics in an extension of linear logic with polarities [Lau98, Lau03], both giving standard definitions when restricted to ordinary λ-calculus.

The target language of the Taylor expansion defined in [ER04] is called *resource λ-calculus*: this is differential λ-calculus with the restriction that all applications are to zero (one restricts derivatives and applications to terms of shape $(\mathrm{D}^n\, s \cdot (u_1, \dots, u_n))\, 0$, which is denoted by $\langle s \rangle\; u_1 \dots u_n$). The first step in the direction of extending Ehrhard and Régnier's results to $\lambda\mu$-calculus is then to define a differential $\lambda\mu$-calculus, which extends both $\lambda\mu$-calculus and differential λ-calculus in a common framework. This is the goal of this paper.

Syntax The grammar for terms is quite straightfroward: just add to $\lambda\mu$-terms those syntactic constructs introduced by differential λ-calculus. Terms are given by:

$$\begin{aligned} s, t &::= x \mid \lambda x\, s \mid (s)\, t \mid \mu\alpha\, \nu \mid \mathrm{D}\, s \cdot t \mid 0 \mid as + bt \\ \nu &::= [\alpha]\, s \end{aligned}$$

with the same identities as in differential λ-calculus. Moreover, μ-abstraction and naming are linear, *i.e.*

$$\mu\alpha\, [\beta]\, (as + bt) \quad = \quad a\mu\alpha\, [\beta]\, s + b\mu\alpha\, [\beta]\, t.$$

Reduction Recall intuitions about the reduction rules of $\lambda\mu$-calculus and differential λ-calculus. The β-reduction

$$(\lambda x\, s)\, t \rightsquigarrow s\, [t/x]$$

of ordinary λ-calculus gives a computational meaning to the application of a function (*i.e.* a λ-abstraction) to an argument: it is substituting t for x in s. The reduction rule specific to $\lambda\mu$-calculus

$$(\mu\alpha\, \nu)\, t \rightsquigarrow \mu\alpha\, \left((\nu)_\alpha\, t\right)$$

describes how applications distribute over multiple outputs merged into an active output. Ordinary aplication becomes application through the abstracted name α: each subterm of ν merged into α is given a copy of t as an argument. The differential reduction

$$\mathrm{D}\,\lambda x\, s \cdot t \rightsquigarrow \lambda x \left(\frac{\partial s}{\partial x} \cdot t\right)$$

is a linearized version of β-reduction: the argument t is substituted for one linear occurrence of x in s.

The only new reduction rule of differential $\lambda\mu$-calculus has naturally redexes of shape $\mathrm{D}\,\mu\alpha\,\nu \cdot t$. The active output of $\mu\alpha\,\nu$ is provided with one linear copy of t, to be distributed to those subterms of ν merged into α. Reduction then amounts to feeding one (linear copy of a) subterm of ν named by α with that linear argument, necessarily by means of a derivative. Again, non-determinism arises, and we write the reduction rule:

$$\mathrm{D}\,\mu\alpha\,\nu \cdot t \rightsquigarrow \mu\alpha\,(\mathrm{D}_\alpha\,\nu \cdot t)$$

where $\mathrm{D}_\alpha\,\nu \cdot t$ is defined inductively by:

$$\begin{array}{rcl}
\mathrm{D}_\alpha\, x \cdot u & = & 0 \\
\mathrm{D}_\alpha\, \lambda x\, s \cdot u & = & \lambda x\,(\mathrm{D}_\alpha\, s \cdot u) \\
\mathrm{D}_\alpha\,(s)\, t \cdot u & = & (\mathrm{D}_\alpha\, s \cdot u)\, t + (\mathrm{D}\, s \cdot (\mathrm{D}_\alpha\, t \cdot u))\, t \\
\mathrm{D}_\alpha\, \mu\beta\, \nu \cdot u & = & \mu\beta\,(\mathrm{D}_\alpha\, \nu \cdot u) \\
\mathrm{D}_\alpha\,[\beta]\, s \cdot u & = & \delta_{\alpha,\beta}[\alpha]\,(\mathrm{D}\, s \cdot u) + [\beta]\,(\mathrm{D}_\alpha\, s \cdot u) \\
\mathrm{D}_\alpha\,(\mathrm{D}\, s \cdot t) \cdot u & = & \mathrm{D}\,(\mathrm{D}_\alpha\, s \cdot u) \cdot t + \mathrm{D}\, s \cdot (\mathrm{D}_\alpha\, t \cdot u) \\
\mathrm{D}_\alpha\,(as + bt) \cdot u & = & a\mathrm{D}_\alpha\, s \cdot u + b\mathrm{D}_\alpha\, t \cdot u
\end{array}$$

where $\delta_{\alpha,\beta} = 1$ if $\alpha = \beta$ and $\delta_{\alpha,\beta} = 0$ otherwise. Again, we introduce a linearization in the application case. Consider the case in which the linear copy of u is actually fed to a subterm: in $[\alpha]\, s$, we can derivate s along u, but it is also possible that we perform the derivation in a subterm of s, hence the sum in

$$\mathrm{D}_\alpha\,[\alpha]\, s \cdot u = [\alpha]\,(\mathrm{D}\, s \cdot u) + [\alpha]\,(\mathrm{D}_\alpha\, s \cdot u).$$

1.4 Outcome

We have just given a quite informal account of differential $\lambda\mu$-calculus. We provide precise definitions and results in sections 2 to 5. In section 2, we define terms, substitution operations and some basic properties about these. Then we define reduction of differential $\lambda\mu$-terms and establish the Church-Rosser property in section 3. In section 5 we prove strong normalization of terms typed in the system we give in section 4.

Many of our results and proof techniques are adapted from those in [ER03] for differential λ-calculus. Our proof of confluence is a minor variation on the usual Tait-Martin-Löf technique in [Bar84], in which the common reduct is explicitely given (see subsection 3.5). Like that of [ER03], our proof of strong normalization involves preliminary work on the structure of the set of strongly normalizing terms (see subsection 5.1). Then we extend the Tait reducibility method used in [ER03] with ideas borrowed from [Par97]: the type associated with a name is interpreted by a set of stacks compatible with that type (see subsection 5.3).

We also contribute some fixes to the original paper about differential λ-calculus [ER03].

- ▷ As we stated before, we show that differential $\lambda(\mu)$-calculus can be defined in accordance with the intrinsic currying of λ-calculus: one can restrict derivatives to the first abstraction only, without losing important properties such as confluence or strong normalization.
- ▷ Also, we establish that differential $\lambda(\mu)$-calculus is a conservative extension of ordinary $\lambda(\mu)$-calculus, as an equational theory. Contrary to what is said in [ER03, Proposition 19], this is not trivial, and we give a proof only under the assumption that the underlying set of scalars is positive, in a sense to be precised below. Otherwise, under mild additional hypotheses, we show that the equational theory collapses: 0 reduces to any term. See subsection 3.6.

- ▷ It is proved in [ER03, Section 4] that simply typed differential λ-terms are strongly normalizing, assuming that scalars are natural numbers; it is then claimed that this proof can be extended to any set of scalars satisfying some conditions. We show that these conditions have to be lifted to stronger ones for the proof to be valid. See remark 5.9.
- ▷ Last, we formalize the weak normalization scheme developped at the very end of [ER03, Section 4]. See subsection 5.5.

We actually prove that, provided the rig of scalars is positive, this differential $\lambda\mu$-calculus is a conservative extension of both $\lambda\mu$-calculus and differential λ-calculus (in the sense that it preserves equational theories associated with their respective reductions). Hence we achieve a first step in the study of possible interaction between those two extensions of ordinary λ-calculus.

2 Syntax

2.1 Preliminary definitions and notations

Let R be a commutative rig. We denote by letters a, b, c the elements of R. We say that R is positive if for all $a, b \in \mathsf{R}$, $a + b = 0 \Rightarrow (a = 0 \wedge b = 0)$. An example of positive commutative rig is $\mathbf{N}$, the set of non-negative integers. We write $\mathsf{R}^\bullet$ for $\mathsf{R} \setminus \{0\}$.

Given a set $\mathcal{X}$, we write $\mathsf{R}\langle\mathcal{X}\rangle$ for the free R-module generated by $\mathcal{X}$, *i.e.* the set of all R-valued functions defined on $\mathcal{X}$, which vanish for almost all values of their argument. Addition and scalar multiplication have the obvious pointwise definitions on $\mathsf{R}\langle\mathcal{X}\rangle$. If $S \in \mathsf{R}\langle\mathcal{X}\rangle$, we denote by $S_{(s)}$ its value at point $s \in \mathcal{X}$ and $\mathrm{Supp}(S) = \{s \in \mathcal{X};\ S_{(s)} \neq 0\}$ its support. $\mathrm{Supp}(S)$ is always finite. The general shape of S is then

$$S = \sum_{s \in \mathcal{X}} S_{(s)} s = \sum_{s \in \mathrm{Supp}(S)} S_{(s)} s$$

which is a finite sum. Since R has unit 1, $\mathcal{X}$ can naturally be seen as a subset of $\mathsf{R}\langle\mathcal{X}\rangle$.

We write $\mathcal{M}_{\mathrm{fin}}(\mathcal{X})$ for the set of finite multisets over $\mathcal{X}$. If $s_1, \ldots, s_n \in \mathcal{X}$, possibly with repetitions, then we write $\langle s_1, \ldots, s_n\rangle$ for the multiset containing exactly $s_1, \ldots, s_n$, taking repetitions into account. If $i, j \in \mathbf{N}$, we write $[i, j]$ for the set $\{k;\ i \leq k \leq j\}$. If $\langle s_1, \ldots, s_n\rangle$ is a multiset, and $I \subseteq [1, n]$, we may write s_I for $\langle s_i\rangle_{i \in I}$. As for sets, we write $\subseteq$ for multiset inclusion, *i.e.* $\langle s_1, \ldots, s_n\rangle \subseteq \langle t_1, \ldots, t_p\rangle$ iff there is an injection $f : [1, n] \longrightarrow [1, p]$ such that for all $i \in [1, n]$, $s_i = t_{f(i)}$.

We use the usual notation for Kronecker's delta:

$$\delta_{s,t} = \begin{cases} 1 \text{ if } s = t \\ 0 \text{ if } s \neq t \end{cases}$$

where 0 and 1 are those of R.

2.2 Terms

Let be given two denumerable sets: $\mathfrak{V}$ the set of λ-variables or, shortly, variables, and $\mathfrak{N}$ the set of μ-variables or, shortly, names. We use letters among x, y, z to denote variables, and α, β, γ to denote names.

We simultaneously define by induction on $k \in \mathbf{N}$ three increasing families of sets: Θ_k is the set of simple pre-terms of height at most k, Δ_k is the set of simple terms of height at most k and $\Delta_k^\square$ is the set of named simple terms of height at most k.

Definition 2.1. We set $\Theta_0 = \Delta_0 = \Delta_0^\square = \emptyset$. Assume Θ_k, Δ_k and $\Delta_k^\square$ are defined.

- Θ_{k+1} is defined as follows:
  - *Variable:* $\mathfrak{V} \subset \Theta_{k+1}$
  - *λ-abstraction:* if $x \in \mathfrak{V}$ and $s \in \Delta_k$ then $\lambda x\, s \in \Theta_{k+1}$;
  - *Application:* if $s \in \Delta_k$ and $T \in \mathsf{R}\langle\Delta_k\rangle$ then $(s)\, T \in \Theta_{k+1}$;
  - *μ-abstraction:* if $\alpha \in \mathfrak{N}$ and $\nu \in \Delta_k^\square$ then $\mu\alpha\, \nu \in \Theta_{k+1}$.

- *Derivative:* If $n \in \mathbf{N}$, $s \in \Theta_{k+1}$, and $u_1, \ldots, u_n \in \Delta_k$ then
$$\mathrm{D}^n\, s \cdot (u_1, \ldots, u_n) \in \Delta_{k+1}.$$
In the case $n = 0$, we simply write s for $\mathrm{D}^0\, s \cdot ()$, *i.e.* we consider $\Theta_{k+1} \subset \Delta_{k+1}$.
- *Naming:* If $\alpha \in \mathfrak{N}$ and $s \in \Delta_{k+1}$ then $[\alpha]\, s \in \Delta^{\square}_{k+1}$.

Notice that we distinguish between simple pre-terms and simple terms for technical reasons only: it allows to keep syntactic equality of terms as simple as possible. The derivative of any term will be introduced later (subsection 2.3).

In the definition of Δ_{k+1}, the sequence $(u_1, \ldots, u_n)$ should be read as the multiset $\langle u_1, \ldots, u_n \rangle$, *i.e.* we identify $\mathrm{D}^n\, s \cdot (u_1, \ldots, u_n)$ with every simple term $\mathrm{D}^n\, s \cdot \left(u_{\sigma(1)}, \ldots, u_{\sigma(n)}\right)$ such that $\sigma \in \mathfrak{S}_n$. This identification is called permutative equality. Moreover, λ and μ are binders: λ binds variables and μ binds names.

We define as usual free and bound variables and names, and α-equality, with some important precaution in the case of a sum: let $S \in \mathsf{R}\langle\Delta_k\rangle$ and let x be a variable (resp. α a name), then x (resp. α) is free in S iff there exists $s \in \mathrm{Supp}(S)$ such that x (resp. α) is free in s. Indeed, simple terms are meant to form a basis of the module of terms; this will be underlined by the way we define one-step reduction in subsection 3.2.

We always consider terms up to α-equality and permutative equality. In the following, it is implicit that our constructs are compatible with both equivalence relations, *i.e.* they do not depend on the choice of a representative.

Lemma 2.2. *(Θ_k), (Δ_k) and $(\Delta^{\square}_k)$ are increasing families.*

Proof. This is easily proved by induction on k. $\square$

Let $\Theta = \bigcup_{k\in\mathbf{N}} \Theta_k$, $\Delta = \bigcup_{k\in\mathbf{N}} \Delta_k$ and $\Delta^{\square} = \bigcup_{k\in\mathbf{N}} \Delta^{\square}_k$. We call simple pre-terms the elements of Θ, simple terms the elements of Δ and named simple terms the elements of $\Delta^{\square}$.

Observe that $\mathsf{R}\langle\Theta\rangle = \bigcup_{k\in\mathbf{N}} \mathsf{R}\langle\Theta_k\rangle$, $\mathsf{R}\langle\Delta\rangle = \bigcup_{k\in\mathbf{N}} \mathsf{R}\langle\Delta_k\rangle$ and $\mathsf{R}\langle\Delta^{\square}\rangle = \bigcup_{k\in\mathbf{N}} \mathsf{R}\langle\Delta^{\square}_k\rangle$. We call pre-terms the elements of $\mathsf{R}\langle\Theta\rangle$, terms the elements of $\mathsf{R}\langle\Delta\rangle$ and named terms the elements of $\mathsf{R}\langle\Delta^{\square}\rangle$.

In the following, simple terms are denoted by s, t, u, v, w, terms by S, T, U, V, W, named simple terms by ν, and named terms by N. If $\mathcal{X}$ is a set of simple terms, we denote by $\mathcal{X}^{\square}$ the set of named simple terms the underlying terms of which are in $\mathcal{X}$:
$$\mathcal{X}^{\square} = \{[\alpha]\, s;\ \alpha \in \mathfrak{N} \text{ and } s \in \mathcal{X}\}\,.$$

We write $x \in s$ for "variable x occurs free in (named, simple) term s" and $\alpha \in s$ for "name α occurs free in (named, simple) term s": in this notation, we identify s with the set of all its free variables and names, which allows us to write $x \in s \cup t$ for "x is free in s or in t".

Remark 2.3 (Induction on terms)**.** The definition of terms is by induction on height. More precisely, we call height of term S the least $k \in \mathbf{N}$ such that $S \in \mathsf{R}\langle\Delta_k\rangle$. Proving a property Φ by induction on terms is then proving $\Phi(S)$ for every term S by induction on its height. More generally, it is proving Φ on the elements of $\mathsf{R}\langle\Delta_k\rangle$ *and* an auxiliary property $\Phi^{\square}$ on the elements of $\mathsf{R}\langle\Delta^{\square}_k\rangle$ simultaneously by induction on k. In the following, $\Phi^{\square}$ is often so close to Φ that we don't even phrase it.

Similarly, defining a function inductively on terms is actually defining that function on $\mathsf{R}\langle\Delta_k\rangle$, together with an auxiliary function on $\mathsf{R}\langle\Delta^{\square}_k\rangle$, recursively on k.

We say that a property Φ is linear if, for all term S, $\Phi(S)$ holds as soon as $\Phi(s)$ holds for all $s \in \mathrm{Supp}(S)$. In this case, it is clear that if Φ holds on Δ_k then it holds on $\mathsf{R}\langle\Delta_k\rangle$. Thus we only have to consider the case of simple terms while proving Φ by induction. Similarly, if we specify that function f (to be defined inductively on terms) is linear, then it is sufficient to define f on simple terms.

2.3 Extended syntax

We now extend the syntactic constructs of the calculus so that we can write 0, aS, $S + T$, $\lambda x\, S$, $(S)\, T$, $\mu\alpha\, N$, $\mathrm{D}\, S \cdot T$ and $[\alpha]\, S$ for all $a, b \in \mathsf{R}$, $x \in \mathfrak{V}$, $\alpha \in \mathfrak{N}$, $S, T \in \mathsf{R}\langle\Delta\rangle$ and $N \in \mathsf{R}\langle\Delta^{\square}\rangle$.

Remark 2.4 (Status of extended syntax)**.** Of course, term $0 \in \mathsf{R}\langle\Delta\rangle$ is the empty sum, and aS and $S+T$ are given by the structure of R-module on terms. For the other constructs, we actually define *functions*

$$\begin{array}{rrcl} \lambda : & \mathfrak{V} \times \mathsf{R}\langle\Delta\rangle & \longrightarrow & \mathsf{R}\langle\Delta\rangle \\ () : & \mathsf{R}\langle\Delta\rangle \times \mathsf{R}\langle\Delta\rangle & \longrightarrow & \mathsf{R}\langle\Delta\rangle \\ \mu : & \mathfrak{N} \times \mathsf{R}\langle\Delta\rangle & \longrightarrow & \mathsf{R}\langle\Delta\rangle \\ [] : & \mathfrak{N} \times \mathsf{R}\langle\Delta\rangle & \longrightarrow & \mathsf{R}\langle\Delta\rangle \\ \mathrm{D} : & \mathsf{R}\langle\Delta\rangle \times \mathsf{R}\langle\Delta\rangle & \longrightarrow & \mathsf{R}\langle\Delta\rangle \end{array}$$

We write them the same as the corresponding syntactic constructs for obvious readability reasons, but one has to keep in mind that they are not part of the basic term syntax. For instance, the expression $\lambda x\, x$ can be considered both as the litteral writing of a term, or as the result of applying function λ to variable x and term x. This ambiguity, however, is harmless, since both possible readings always produce the same term.

We first define the derivative of any simple term: if $s \in \Theta$ and $u, u_1, \ldots, u_n \in \Delta$ then we define

$$\mathrm{D}\left(\mathrm{D}^n\, s \cdot (u_1, \ldots, u_n)\right) \cdot u = \mathrm{D}^{n+1}\, s \cdot (u, u_1, \ldots, u_n) \in \Delta.$$

In particular, if $s \in \Theta$, $\mathrm{D}\, s \cdot u = \mathrm{D}^1\, s \cdot (u)$.

Then we extend all constructs by linearity: let $S, T, U \in \mathsf{R}\langle\Delta\rangle$ and $N \in \mathsf{R}\langle\Delta^\square\rangle$, we set

$$\begin{array}{rcl} \lambda x\, S & = & \sum_{s \in \Delta} S_{(s)} \lambda x\, s \\ (S)\, T & = & \sum_{s \in \Delta} S_{(s)}\, (s)\, T \\ \mu\alpha\, N & = & \sum_{\nu \in \Delta^\square} N_{(\nu)} \mu\alpha\, \nu \\ \mathrm{D}\, S \cdot U & = & \sum_{s, u \in \Delta} S_{(s)} U_{(u)} \mathrm{D}\, s \cdot u \\ [\alpha]\, S & = & \sum_{s \in \Delta} S_{(s)} [\alpha]\, s. \end{array}$$

This is to say that λ-abstraction, μ-abstraction and naming are linear, application is linear in the function, and derivative is bilinear.

It is straightforward from the previous definitions that $\mathrm{D}\,(\mathrm{D}\, S \cdot U) \cdot V = \mathrm{D}\,(\mathrm{D}\, S \cdot V) \cdot U$ for all terms S, U and V. Hence we denote by $\mathrm{D}^n\, S \cdot (U_1, \ldots, U_n)$ the term

$$\mathrm{D}\,(\ldots(\mathrm{D}\, S \cdot U_1)\ldots) \cdot U_n,$$

and for any permutation $\sigma \in \mathfrak{S}_n$ we have:

$$\mathrm{D}^n\, S \cdot (U_1, \ldots, U_n) = \mathrm{D}^n\, S \cdot \left(U_{\sigma(1)}, \ldots, U_{\sigma(n)}\right).$$

If furthermore $\forall i$, $U_i = U$ we simply write $\mathrm{D}^n\, S \cdot U^n$.

Remark 2.5 (About the presentation of syntax)**.** In this extended syntax, one can write any term of the grammar we gave in section 1.3:

$$\begin{array}{rcl} \sigma, \tau & ::= & x \mid \lambda x\, \sigma \mid (\sigma)\, \tau \mid \mu\alpha\, \upsilon \mid \mathrm{D}\, \sigma \cdot \tau \mid 0 \mid a\sigma + b\tau \\ \upsilon & ::= & [\alpha]\, \sigma \end{array}$$

where a and b range over R. For the sake of clarity, call these terms abstract terms.

As we stated in remark 1.3, one may use those abstract terms as the basis of the calculus, with syntactic equality induced by α-equality and permutative equality only. Then one would have to introduce rewriting rules to handle linearity and equality between linear combinations, and obtain some congruence on terms. In this setting, terms given by the restricted syntax of subsection 2.2 would only be canonical forms for this congruence. This is akin to the viewpoint adopted in [Vau06].

The goal of this paper, however, is to study basic mathematical properties of differential $\lambda\mu$-calculus. In this purpose, we think such a presentation would introduce too much bureaucracie, and we consider our solution (that of Ehrhard-Régnier's [ER03], actually) better. Terms are those defined in subsection 2.2, and the identity relation on terms is given by α-equality, permutative equality, and those equality relations induced by the module structure. The definitions and proofs we give in the following are then by induction on terms as explained in remark 2.3. When we write terms in the extended syntax, we do not mean them as abstract terms, but as the result of the functions mentioned in remark 2.4.

2.4 Operations on terms

There are four operations on terms: substitution $S\,[T/x]$, partial derivative $\frac{\partial S}{\partial x}\cdot T$, named application $(S)_\alpha\,T$ and named derivative $\mathrm{D}_\alpha\, S\cdot T$. Each one introduces a way to use an argument as described in section 1.3, and is typically the reduced form of a redex (see the definition of one-step reduction in section 3.2). All of them are linear in S and defined inductively on S.

Definition 2.6. Define substitution $S\,[T/x]$ by:

$$
\begin{array}{rcl}
y\,[T/x] & = & \begin{cases} T \text{ if } x = y \\ y \ \text{ otherwise} \end{cases} \\
(\lambda y\, s)\,[T/x] & = & \lambda y\,(s\,[T/x]) \\
((s)\,U)\,[T/x] & = & (s\,[T/x])\,(U\,[T/x]) \\
(\mu\alpha\,\nu)\,[T/x] & = & \mu\alpha\,(\nu\,[T/x]) \\
([\alpha]\,s)\,[T/x] & = & [\alpha]\,(s\,[T/x]) \\
(\mathrm{D}^n\, s\cdot(u_1,\ldots,u_n))\,[T/x] & = & \mathrm{D}^n\, s\,[T/x]\cdot(u_1\,[T/x],\ldots,u_n\,[T/x]) \\
S\,[T/x] & = & \displaystyle\sum_{s\in\mathrm{Supp}(S)} S_{(s)}\,(s\,[T/x])
\end{array}
$$

where $x\neq y$ and $y\notin T$ in the case of λ-abstraction, and $\alpha\notin T$ in the case of μ-abstraction.

Definition 2.7. Define partial derivative $\frac{\partial S}{\partial x}\cdot T$ by:

$$
\begin{array}{rcl}
\dfrac{\partial y}{\partial x}\cdot T & = & \delta_{x,y}T \\
\dfrac{\partial \lambda y\, s}{\partial x}\cdot T & = & \lambda y\left(\dfrac{\partial s}{\partial x}\cdot T\right) \\
\dfrac{\partial (s)\,U}{\partial x}\cdot T & = & \left(\dfrac{\partial s}{\partial x}\cdot T\right)U + \left(\mathrm{D}\, s\cdot\left(\dfrac{\partial U}{\partial x}\cdot T\right)\right)U \\
\dfrac{\partial \mu\alpha\,\nu}{\partial x}\cdot T & = & \mu\alpha\left(\dfrac{\partial \nu}{\partial x}\cdot T\right) \\
\dfrac{\partial [\alpha]\,s}{\partial x}\cdot T & = & [\alpha]\left(\dfrac{\partial s}{\partial x}\cdot T\right) \\
\dfrac{\partial}{\partial x}\left(\mathrm{D}^n\, s\cdot(u_1,\ldots,u_n)\right)\cdot T & = & \mathrm{D}^n\left(\dfrac{\partial s}{\partial x}\cdot T\right)\cdot(u_1,\ldots,u_n) \\
 & + & \displaystyle\sum_{j=1}^{n}\mathrm{D}^n\, s\cdot\left(u_1,\ldots,\dfrac{\partial u_j}{\partial x}\cdot T,\ldots,u_n\right) \\
\dfrac{\partial S}{\partial x}\cdot T & = & \displaystyle\sum_{s\in\mathrm{Supp}(S)} S_{(s)}\left(\dfrac{\partial s}{\partial x}\cdot T\right)
\end{array}
$$

where $x\neq y$ and $y\notin T$ in the case of λ-abstraction, and $\alpha\notin T$ in the case of μ-abstraction.

This is partial derivative as given in [ER03]. It extends naturally to μ-abstraction and named terms. It is intuitively the sum of all terms that can be obtained by replacing one linear occurence of x in S by T. Recall that, since all free occurences of x in S are not necessarily linear, one has to force some kind of linearization: see the application case, in the rightmost summand.

Definition 2.8. Define named application $(S)_\alpha T$ by:

$$\begin{array}{rcl}
(x)_\alpha T & = & x \\
(\lambda x\, s)_\alpha T & = & \lambda x \left((s)_\alpha T\right) \\
((s)\, U)_\alpha T & = & \left((s)_\alpha T\right) \left((U)_\alpha T\right) \\
(\mu\beta\, \nu)_\alpha T & = & \mu\beta \left((\nu)_\alpha T\right) \\
([\beta]\, s)_\alpha T & = & \left\{ \begin{array}{ll} [\alpha] \left((s)_\alpha T\right) T & \text{if } \alpha = \beta \\ [\beta]\, (s)_\alpha T & \text{otherwise} \end{array} \right. \\
(\mathrm{D}^n\, s \cdot (u_1, \ldots, u_n))_\alpha T & = & \mathrm{D}^n\, (s)_\alpha T \cdot \left((u_1)_\alpha T, \ldots, (u_n)_\alpha T\right) \\
(S)_\alpha T & = & \displaystyle\sum_{s \in \mathrm{Supp}(S)} S_{(s)} \left((s)_\alpha T\right)
\end{array}$$

where $\alpha \neq \beta$ and $\beta \notin T$ in the case of μ-abstraction, and $x \notin T$ in the case of λ-abstraction.

This is actually the structural substitution of ordinary $\lambda\mu$-calculus (see [Par92]), which is often denoted by $S\,[[\alpha]\,(U)\,T/[\alpha]\,U]$. As we explained in remark 1.2, we prefer this new notation for several reasons, among which concision is not the lesser. Moreover, it fits well with the intuition of $\lambda\mu$-terms being a generalization of λ-terms with several named outputs: $(S)_\alpha T$ is the application of the output of S named by α to the argument T. Many properties about named application are then reminiscent of corresponding properties of application: see, *e.g.*, commutation lemmas 2.15 to 2.18, or lemma 4.2 about typing. This intuition was crucial in designing the linear counterpart of named application as named derivative.

Definition 2.9. Define named derivative $\mathrm{D}_\alpha\, S \cdot T$ by:

$$\begin{array}{rcl}
\mathrm{D}_\alpha\, x \cdot T & = & 0 \\
\mathrm{D}_\alpha\, \lambda x\, s \cdot T & = & \lambda x\, (\mathrm{D}_\alpha\, s \cdot T) \\
\mathrm{D}_\alpha\, (s)\, U \cdot T & = & (\mathrm{D}_\alpha\, s \cdot T)\, U + (\mathrm{D}\, s \cdot (\mathrm{D}_\alpha\, U \cdot T))\, U \\
\mathrm{D}_\alpha\, \mu\beta\, \nu \cdot T & = & \mu\beta\, (\mathrm{D}_\alpha\, \nu \cdot T) \\
\mathrm{D}_\alpha\, [\beta]\, s \cdot T & = & \delta_{\alpha,\beta} [\alpha]\, (\mathrm{D}\, s \cdot T) + [\beta]\, (\mathrm{D}_\alpha\, s \cdot T) \\
\mathrm{D}_\alpha\, (\mathrm{D}^n\, s \cdot (u_1, \ldots, u_n)) \cdot T & = & \mathrm{D}^n\, (\mathrm{D}_\alpha\, s \cdot T) \cdot (u_1, \ldots, u_n) \\
 & + & \displaystyle\sum_{j=1}^{n} \mathrm{D}\, s \cdot (u_1, \ldots, \mathrm{D}_\alpha\, u_j \cdot T, \ldots, u_n) \\
\mathrm{D}_\alpha\, S \cdot T & = & \displaystyle\sum_{s \in \mathrm{Supp}(S)} S_{(s)}\, (\mathrm{D}_\alpha\, s \cdot T)
\end{array}$$

where $\alpha \neq \beta$ and $\beta \notin T$ in the case of μ-abstraction, and $x \notin T$ in the case λ-abstraction.

Here we give the only new operation. The best way to understand it may be by analogy: the linear counterpart of application is derivative; substitution replaces all occurences of a variable with copies of the argument; partial derivative, as the linear counterpart of substitution replaces exactly one linear occurence of a variable with the argument; named application distributes copies of its argument to all subterms merged into a named output; and named derivative, as the linear conterpart of named application, differentiates one linear subterm merged into a named output along its argument. Again, many properties about named derivative are reminiscent of corresponding properties for derivative: notably lemmas 2.15 to 2.18, and 4.2.

2.5 Basic properties of operations

First, operations act on extended syntactic constructions the same way they act on basic ones. For instance, if S, T, U are terms, then $(\mu\alpha\, S)\,[U/x] = \mu\alpha\, S\,[U/x]$ (provided $\alpha \notin U$) and $\mathrm{D}_\alpha\, (S)\, T \cdot U = (\mathrm{D}_\alpha\, S \cdot U)\, T + (\mathrm{D}\, S \cdot (\mathrm{D}_\alpha\, T \cdot U))\, T$. This is a straightforward consequence of linearity for abstractions, application and naming. The case of derivative is stated in the following lemma.

Lemma 2.10. *If $x \in \mathfrak{V}$, $\alpha \in \mathfrak{N}$, and $S, U, V \in \mathsf{R}\langle\Delta\rangle$, then we have:*

$$\begin{array}{rcl} (\mathrm{D}\, S \cdot U)\,[V/x] & = & \mathrm{D}\, S\,[V/x] \cdot U\,[V/x] \\ \dfrac{\partial \mathrm{D}\, S \cdot U}{\partial x} \cdot V & = & \mathrm{D}\left(\dfrac{\partial S}{\partial x} \cdot V\right) \cdot U + \mathrm{D}\, S \cdot \left(\dfrac{\partial U}{\partial x} \cdot V\right) \\ (\mathrm{D}\, S \cdot U)_\alpha\, V & = & \mathrm{D}\,(S)_\alpha\, V \cdot (U)_\alpha\, V \\ \mathrm{D}_\alpha\,(\mathrm{D}\, S \cdot U) \cdot V & = & \mathrm{D}\,(\mathrm{D}_\alpha\, S \cdot V) \cdot U + \mathrm{D}\, S \cdot (\mathrm{D}_\alpha\, U \cdot V) \end{array}$$

Proof. If S and U are simple terms (s and u), this is just the definition of each operation: $\mathrm{D}\, s \cdot u$ is of shape $\mathrm{D}^{n+1}\, t \cdot (u, u_1, \ldots, u_n)$, where t is a simple pre-term (see the case of derivative in definitions 2.6 to 2.9). The result extends to terms by linearity, along the lines of remark 2.3. □

Lemma 2.10 is used in nearly all of the following proofs so that we generally omit to mention it. The following lemma states the usual irrelevance property for substitution and named application.

Lemma 2.11. *If x is a variable and S a term such that $x \notin S$, then for any term T, $S\,[T/x] = S$. If α is a name and S a term such that $\alpha \notin S$, then for any term T, $(S)_\alpha\, T = S$.*

Proof. This is proved by a simple induction on term S. □

Substitution preserves simple terms as soon as the substituted term is simple; the same holds for named application without condition on the term given as argument. More formally:

Lemma 2.12. *If $s, t \in \Delta$ and $x \in \mathfrak{V}$ then $s\,[t/x] \in \Delta$. If $s \in \Delta$, $T \in \mathsf{R}\langle\Delta\rangle$ and $\alpha \in \mathfrak{N}$ then $(s)_\alpha\, T \in \Delta$.*

Proof. The proof of both results is an easy induction on s. □

The following two lemmas enlighten both notions of linearity (logical and algebraic). The first one is about linearity of partial and named derivative in the logical sense: since there is no free occurence of variable x or name α in S, one can't substitute U for exactly one linear occurence of x or α. The second one is about the other kind of linearity: partial derivative and named derivative are not only linear in the function but also in the argument (as is derivative). Both extend lemma 3 of [ER03] to the differential $\lambda\mu$-calculus.

Lemma 2.13. *If x is a variable and S a term such that $x \notin S$, then for any term U, $\frac{\partial S}{\partial x} \cdot U = 0$. If α is a name and S a term such that $\alpha \notin S$, then for any term U, $\mathrm{D}_\alpha\, S \cdot U = 0$.*

Lemma 2.14. *If x is a variable, α is a name, and S and U are terms, then*

$$\frac{\partial S}{\partial x} \cdot U = \sum_{u \in \mathrm{Supp}(U)} U_{(u)} \frac{\partial S}{\partial x} \cdot u$$

and

$$\mathrm{D}_\alpha\, S \cdot U = \sum_{u \in \mathrm{Supp}(U)} U_{(u)} \mathrm{D}_\alpha\, S \cdot u.$$

Proof. The last two lemmas are easily proved by induction on term S. In particular, the application case holds thanks to the linearization on the fly involved in the definitions of partial derivative and named derivative. □

2.6 Commutations

In this section, we state commutation lemmas which exhibit how operations interact with each other. All of them are proved separately by induction on S. These proofs are boring and repetitive but bear no particular difficulty: they are analogues of the proofs of lemmas 2, 4, 5 and 6 of [ER03]. Similar results where term S is replaced by any named term N obviously hold too.

For all variables x, y, for all names α, β, for all terms S, U, V, the following four lemmas hold.

Lemma 2.15. *If $x \neq y$ and $x \notin V$ then*

$$S\,[U/x]\,[V/y] = S\,[V/y]\,[U\,[V/y]/x]\,.$$

If $x \neq y$ and $x \notin V$ then

$$\left(\frac{\partial S}{\partial x} \cdot U\right)[V/y] = \frac{\partial S\,[V/y]}{\partial x} \cdot (U\,[V/y])\,.$$

If $\alpha \notin V$ then

$$\big((S)_\alpha\, U\big)\,[V/x] = (S\,[V/x])_\alpha\, (U\,[V/x])$$

and

$$(\mathrm{D}_\alpha\, S \cdot U)\,[V/x] = \mathrm{D}_\alpha\, (S\,[V/x]) \cdot (U\,[V/x]).$$

Lemma 2.16. *If $x \neq y$ and $x \notin U \cup V$ then*

$$\frac{\partial S\,[U/x]}{\partial y} \cdot V = \left(\frac{\partial S}{\partial y} \cdot V\right)[U/x] + \left(\frac{\partial S}{\partial x} \cdot \left(\frac{\partial U}{\partial y} \cdot V\right)\right)[U/x]\,.$$

If $x \notin V$ then

$$\frac{\partial}{\partial y}\left(\frac{\partial S}{\partial x} \cdot U\right) \cdot V = \frac{\partial}{\partial x}\left(\frac{\partial S}{\partial y} \cdot V\right) \cdot U + \frac{\partial S}{\partial x} \cdot \left(\frac{\partial U}{\partial y} \cdot V\right).$$

If $\alpha \notin U \cup V$ then

$$\frac{\partial (S)_\alpha\, U}{\partial x} \cdot V = \left(\frac{\partial S}{\partial x} \cdot V\right)_\alpha U + \left(\mathrm{D}_\alpha\, S \cdot \left(\frac{\partial U}{\partial x} \cdot V\right)\right)_\alpha U.$$

If $\alpha \notin V$ then

$$\frac{\partial \mathrm{D}_\alpha\, S \cdot U}{\partial x} \cdot V = \mathrm{D}_\alpha \left(\frac{\partial S}{\partial x} \cdot V\right) \cdot U + \mathrm{D}_\alpha\, S \cdot \left(\frac{\partial U}{\partial x} \cdot V\right).$$

Lemma 2.17. *If $x \notin V$ then*

$$(S\,[U/x])_\alpha\, V = \big((S)_\alpha\, V\big)\,\big[(U)_\alpha\, V/x\big]$$

and

$$\left(\frac{\partial S}{\partial x} \cdot U\right)_\alpha V = \frac{\partial (S)_\alpha\, V}{\partial x} \cdot \big((U)_\alpha\, V\big)\,.$$

If $\alpha \neq \beta$ and $\alpha \notin V$ then

$$\big((S)_\alpha\, U\big)_\beta\, V = \big((S)_\beta\, V\big)_\alpha\, \big((U)_\beta\, V\big)$$

and

$$(\mathrm{D}_\alpha\, S \cdot U)_\beta\, V = \mathrm{D}_\alpha\, (S)_\beta\, V \cdot \big((U)_\beta\, V\big).$$

Lemma 2.18. *If $x \notin U \cup V$ then*

$$\mathrm{D}_\alpha\, S\,[U/x] \cdot V = (\mathrm{D}_\alpha\, S \cdot V)\,[U/x] + \left(\frac{\partial S}{\partial x} \cdot (\mathrm{D}_\alpha\, U \cdot V)\right)[U/x]\,.$$

If $x \notin V$ then

$$\mathrm{D}_\alpha \left(\frac{\partial S}{\partial x} \cdot U\right) \cdot V = \frac{\partial \mathrm{D}_\alpha\, S \cdot V}{\partial x} \cdot U + \frac{\partial S}{\partial x} \cdot (\mathrm{D}_\alpha\, U \cdot V)\,.$$

If $\alpha \neq \beta$ and $\alpha \notin U \cup V$ then

$$\mathrm{D}_\beta\, (S)_\alpha\, U \cdot V = (\mathrm{D}_\beta\, S \cdot V)_\alpha\, U + (\mathrm{D}_\alpha\, S \cdot (\mathrm{D}_\beta\, U \cdot V))_\alpha\, U.$$

If $\alpha \notin V$ then

$$\mathrm{D}_\beta\, (\mathrm{D}_\alpha\, S \cdot U) \cdot V = \mathrm{D}_\alpha\, (\mathrm{D}_\beta\, S \cdot V) \cdot U + \mathrm{D}_\alpha\, S \cdot (\mathrm{D}_\beta\, U \cdot V).$$

Notice how the named application case in lemmas 2.15 to 2.18 closely match the application case in definitions 2.6 to 2.9. In particular, lemmas 2.16 and 2.18 reproduce the linearization on the fly we used in the definition of partial and named derivatives of an application. This is one point where the notations we introduced fit nicely with intuition. Similarly, notice how the named derivative case in lemmas 2.15 to 2.18 closely match items of lemma 2.10.

2.7 Iterated operations

Lemmas 2.15 to 2.18 allow us to define parallel generalizations of substitution, named application, partial derivative and named derivative without ambiguity:

Corollary 2.19. *If $x_1, \dots, x_n$ are distinct variables not free in any of the terms $U_1, \dots, U_n$, then for all term S,*

$$S\left[U_{\sigma(1)}/x_{\sigma(1)}\right] \dots \left[U_{\sigma(n)}/x_{\sigma(n)}\right]$$

does not depend on permutation $\sigma \in \mathfrak{S}_n$. We write it $S\left[U_1, \dots, U_n/x_1, \dots, x_n\right]$.

Corollary 2.20. *If variables $x_1, \dots, x_n$ are not free in any of the terms $U_1, \dots, U_n$, then for all term S,*

$$\frac{\partial}{\partial x_{\sigma(n)}}\left(\dots \frac{\partial S}{\partial x_{\sigma(1)}} \cdot U_{\sigma(1)} \dots\right) \cdot U_{\sigma(n)}$$

does not depend on permutation $\sigma \in \mathfrak{S}_n$. We write it $\frac{\partial^n S}{\partial x_1 \dots \partial x_n} \cdot (U_1, \dots, U_n)$. If $x_1 = \dots = x_n = x$, we write it $\frac{\partial^n S}{\partial x^n} \cdot (U_1, \dots, U_n)$. If furthermore $U_1 = \dots = U_n = U$, we write it $\frac{\partial^n S}{\partial x^n} \cdot U^n$.

Corollary 2.21. *If $\alpha_1, \dots, \alpha_n$ are distinct names not free in any of the terms $U_1, \dots, U_n$, then for all term S,*

$$\left(\dots (S)_{\alpha_{\sigma(1)}} U_{\sigma(1)} \dots\right)_{\alpha_{\sigma(n)}} U_{\sigma(n)}$$

does not depend on permutation $\sigma \in \mathfrak{S}_n$. We write it $(S)_{\alpha_1, \dots, \alpha_n} (U_1, \dots, U_n)$.

Corollary 2.22. *If names $\alpha_1, \dots, \alpha_n$ are not free in any of the terms $U_1, \dots, U_n$, then for all term S,*

$$\mathrm{D}_{\alpha_{\sigma(n)}}\left(\dots \mathrm{D}_{\alpha_{\sigma(1)}} S \cdot U_{\sigma(1)} \dots\right) \cdot U_{\sigma(n)}$$

does not depend on permutation $\sigma \in \mathfrak{S}_n$. We write it $\mathrm{D}_{\alpha_1, \dots, \alpha_n} S \cdot (U_1, \dots, U_n)$. If $\alpha_1 = \dots = \alpha_n = \alpha$, we write it $\mathrm{D}_\alpha^n S \cdot (U_1, \dots, U_n)$. If furthermore $U_1 = \dots = U_n = U$, we write it $\mathrm{D}_\alpha^n S \cdot U^n$.

The last five lemmas of this section explicit the general shape of an iterated partial derivative or named derivative of a derivative, application or named term. These will be useful in the strong normalization proof at the very end of this paper.

Lemma 2.23. *If the variables $x_1, \dots, x_n$ are not free in any of the terms $U_1, \dots, U_n$, then*

$$\frac{\partial^n \mathrm{D}\, S \cdot T}{\partial x_1 \dots \partial x_n} \cdot (U_1, \dots, U_n)$$

is a sum of terms of shape $\mathrm{D}\, S' \cdot T'$ where

$$S' = \frac{\partial^p S}{\partial y_1 \dots \partial y_p} \cdot (V_1, \dots, V_p)$$

and

$$T' = \frac{\partial^q T}{\partial z_1 \dots \partial z_q} \cdot (W_1, \dots, W_q)$$

with $p + q = n$ and $\langle (y_1, V_1), \dots, (y_p, V_p), (z_1, W_1), \dots, (z_q, W_q) \rangle = \langle (x_1, U_1), \dots, (x_n, U_n) \rangle$.

Lemma 2.24. *If the names $\alpha_1, \dots, \alpha_n$ are not free in any of the terms $U_1, \dots, U_n$, then*

$$\mathrm{D}_{\alpha_1, \dots, \alpha_n} (\mathrm{D}\, S \cdot T) \cdot (U_1, \dots, U_n)$$

is a sum of terms of shape $\mathrm{D}\, S' \cdot T'$ where

$$S' = \mathrm{D}_{\beta_1, \dots, \beta_p} S \cdot (V_1, \dots, V_p)$$

and

$$T' = \mathrm{D}_{\gamma_1, \dots, \gamma_q} T \cdot (W_1, \dots, W_q)$$

with $p + q = n$ and $\langle (\beta_1, V_1), \dots, (\beta_p, V_p), (\gamma_1, W_1), \dots, (\gamma_q, W_q) \rangle = \langle (\alpha_1, U_1), \dots, (\alpha_n, U_n) \rangle$.

Lemma 2.25. *If the variables $x_1, \dots, x_n$ are not free in any of the terms $U_1, \dots, U_n$, then*

$$\frac{\partial^n (S)\, T}{\partial x_1 \dots \partial x_n} \cdot (U_1, \dots, U_n)$$

is a sum of terms of shape $(\mathrm{D}^r\, S' \cdot (T'_1, \dots, T'_r))\, T$ where

$$S' = \frac{\partial^{q_0} S}{\partial z^{(0)}_1 \dots \partial z^{(0)}_{q_0}} \cdot \left(W^{(0)}_1, \dots, W^{(0)}_{q_0}\right)$$

and

$$T'_j = \frac{\partial^{q_j} T}{\partial z^{(j)}_1 \dots \partial z^{(j)}_{q_j}} \cdot \left(W^{(j)}_1, \dots, W^{(j)}_{q_j}\right)$$

with $\sum_{j=0}^{r} q_j = n$ and $\left\langle (z^{(0)}_1, W^{(0)}_1), \dots, (z^{(r)}_{q_r}, W^{(r)}_{q_r}) \right\rangle = \langle (x_1, U_1), \dots, (x_n, U_n) \rangle$.

Lemma 2.26. *If the names $\alpha_1, \dots, \alpha_n$ are not free in any of the terms $U_1, \dots, U_n$, then*

$$\mathrm{D}_{\alpha_1, \dots, \alpha_n}\, ((S)\, T) \cdot (U_1, \dots, U_n)$$

is a sum of terms of shape $(\mathrm{D}^r\, S' \cdot (T'_1, \dots, T'_r))\, T$ where

$$S' = \mathrm{D}_{\gamma^{(0)}_1, \dots, \gamma^{(0)}_{q_0}}\, S \cdot \left(W^{(0)}_1, \dots, W^{(0)}_{q_0}\right)$$

and

$$T'_j = \mathrm{D}_{\gamma^{(j)}_1, \dots, \gamma^{(j)}_{q_j}}\, S \cdot \left(W^{(j)}_1, \dots, W^{(j)}_{q_j}\right)$$

with $\sum_{j=0}^{r} q_j = n$ and $\left\langle (z^{(0)}_1, W^{(0)}_1), \dots, (z^{(r)}_{q_r}, W^{(r)}_{q_r}) \right\rangle = \langle (x_1, U_1), \dots, (x_n, U_n) \rangle$.

Lemma 2.27. *If the name α is not free in any of the terms $U_1, \dots, U_n$, then*

$$\mathrm{D}^n_\alpha\, [\alpha]\, S \cdot (U_1, \dots, U_n)$$

is a sum of terms of shape

$$[\alpha]\, \mathrm{D}^r \left(\mathrm{D}^{n-r}_\alpha\, S \cdot U_J\right) \cdot U_I$$

where $I \subseteq [1, n]$, r is the cardinality of I and $J = [1, n] \setminus I$.

Proof. Each of the previous five lemmas is easily proved by induction on n, using linearity of operations and lemma 2.10. We give a full proof of lemma 2.23. If $n = 0$, the result is trivial. Now assume that

$$\frac{\partial^n \mathrm{D}\, S \cdot T}{\partial x_1 \dots \partial x_n} \cdot (U_1, \dots, U_n) = \sum_{i=0}^{r} \mathrm{D}\, S_i \cdot T_i$$

where

$$S_i = \frac{\partial^p S}{\partial y^{(i)}_1 \dots \partial y^{(i)}_{p_i}} \cdot \left(V^{(i)}_1, \dots, V^{(i)}_p\right)$$

and

$$T_i = \frac{\partial^q T}{\partial z^{(i)}_1 \dots \partial z^{(i)}_{q_i}} \cdot \left(W^{(i)}_1, \dots, W^{(i)}_q\right)$$

with

$$\left\langle (y^{(i)}_1, V^{(i)}_1), \dots, (y^{(i)}_{p_i}, V^{(i)}_{p_i}), (z^{(i)}_1, W^{(i)}_1), \dots, (z^{(i)}_{q_i}, W^{(i)}_{q_i}) \right\rangle = \langle (x_1, U_1), \dots, (x_n, U_n) \rangle .$$

Then

$$\begin{aligned}
\frac{\partial^{n+1} \mathrm{D}\, S \cdot T}{\partial x_1 \dots \partial x_{n+1}} \cdot (U_1, \dots, U_{n+1}) &= \frac{\partial}{\partial x_{n+1}} \left(\frac{\partial^n \mathrm{D}\, S \cdot T}{\partial x_1 \dots \partial x_n} \cdot (U_1, \dots, U_n)\right) \cdot U_{n+1} \\
&= \sum_{i=0}^{r} \frac{\partial \mathrm{D}\, S_i \cdot T_i}{\partial x_{n+1}} \cdot U_{n+1} \\
&= \sum_{i=0}^{r} \mathrm{D} \left(\frac{\partial S_i}{\partial x_{n+1}} \cdot U_{n+1}\right) \cdot T_i + \sum_{i=0}^{r} \mathrm{D}\, S_i \cdot \left(\frac{\partial T_i}{\partial x_{n+1}} \cdot U_{n+1}\right)
\end{aligned}$$

where the second equation holds by linearity and the third one by lemma 2.10. This ends the proof by definition of iterated partial derivative. □

3 Reductions

In this section, we define the reductions of differential $\lambda\mu$-calculus. One-step reduction ρ is the analogue of usual β-reduction. Recall basic reduction rules from section 1.3:

$$\begin{array}{rcl} (\lambda x\, S)\, T & \rho & S\,[T/x] \\ (\mu\alpha\, N)\, T & \rho & \mu\alpha\, (N)_\alpha\, T \\ \mathrm{D}\,\lambda x\, S \cdot T & \rho & \lambda x \left(\dfrac{\partial S}{\partial x} \cdot T\right) \\ \mathrm{D}\,\mu\alpha\, N \cdot T & \rho & \mu\alpha\, (\mathrm{D}_\alpha\, N \cdot T). \end{array}$$

Moreover, we define reduction of a sum as follows: we set $S\;\rho\;S'$ iff $S = at + U$ and $S' = aT' + U$ where t is a simple term, $a \in \mathsf{R}^\bullet$ and $t\;\rho\;T'$. This means that ρ reduces only part of exactly one underlying simple term of S. Intuitively, this definition is motivated by the following two facts.

- ▷ In order to prove confluence of ρ we adapt the Tait-Martin-Löf technique as presented in [Bar84]. We define some parallel extension π of ρ: in π, any number of distinct redexes may be reduced simultaneaously. We then prove that π enjoys the diamond property: if $S\;\pi\;T$ and $S\;\pi\;T'$ then there exists $U \in \mathsf{R}\langle\Delta\rangle$ such that $T\;\pi\;U$ and $T'\;\pi\;U$. In this setting, it is necessary that, *e.g.*, $S + T\;\rho\;S' + T$ as soon as $S\;\rho\;S'$.
- ▷ In order to get a strong normalization property, we require that a is non-zero in the reduction $at + U\;\rho\;aT' + U$: otherwise ρ is reflexive.

Hence we cannot define reduction by induction on terms: if for instance $-1 \in \mathsf{R}$, term 0 may reduce, as in $0 = u - u\;\rho\;U' - u$.

In the following, we define precisely one-step and parallel reductions, and prove confluence. Then we study conservativity of the equational theory associated with reduction, with respect to that of pure $\lambda\mu$-calculus and differential λ-calculus.

3.1 Preliminaries

We call relation from simple terms to terms any subset of $\Delta \times \mathsf{R}\langle\Delta\rangle$ and relation from terms to terms any subset of $\mathsf{R}\langle\Delta\rangle \times \mathsf{R}\langle\Delta\rangle$. We of course extend any such relation τ to named terms by: $[\alpha]\, S\;\tau\;N$ iff $N = [\alpha]\, S'$ with $S\;\tau\;S'$.

Definition 3.1. A relation τ from terms to terms is said linear if $0\;\tau\;0$ and $aS + bT\;\tau\;aS' + bT'$ as soon as $S\;\tau\;S'$ and $T\;\tau\;T'$.

Remark 3.2 (Linearity)**.** This notion of linearity is quite different from that of remark 2.3: even if τ is linear, it is not sufficient to define the restriction of τ on simple terms. Indeed, two distinct linear relations $\tau, \tau' \subset \mathsf{R}\langle\Delta\rangle \times \mathsf{R}\langle\Delta\rangle$ may coincide on $\Delta \times \mathsf{R}\langle\Delta\rangle$ or $\mathsf{R}\langle\Delta\rangle \times \Delta$ or even both: take $\mathsf{R} = \mathbf{N}$, $\tau = \{(0,0)\}$ and $\tau' = \{(2n\lambda x\, x, 2n\lambda x\, x);\; n \in \mathbf{N}\}$. On the other hand, if f is a function from terms to terms, one can define the graph of f as the relation $\tau_f = \{(S, f(S));\; S \in \mathsf{R}\langle\Delta\rangle\}$, which is linear if and only if f is linear.

Definition 3.3. A relation τ from terms to terms is said contextual if it is reflexive, linear, and satisfies the following conditions as soon as $S\;\tau\;S'$, $T\;\tau\;T'$ and $N\;\tau\;N'$:

$$\begin{array}{rcl} \lambda x\, S & \tau & \lambda x\, S' \\ (S)\, T & \tau & \left(S'\right) T' \\ \mu\alpha\, N & \tau & \mu\alpha\, N' \\ [\alpha]\, S & \tau & [\alpha]\, S' \\ \mathrm{D}\, S \cdot T & \tau & \mathrm{D}\, S' \cdot T'. \end{array}$$

Lemma 3.4. *If τ is a contextual relation and S, U, U' are terms such that $U\;\tau\;U'$ then:*

$$\begin{array}{rcl} S\,[U/x] & \tau & S\left[U'/x\right] \\ (S)_\alpha\, U & \tau & (S)_\alpha\, U' \end{array}$$

$$\begin{array}{rcl} \frac{\partial S}{\partial x} \cdot U & \tau & \frac{\partial S}{\partial x} \cdot U' \\ \mathrm{D}_\alpha\, S \cdot U & \tau & \mathrm{D}_\alpha\, S \cdot U'. \end{array}$$

Proof. This is proved by a straightforward induction on S. □

Given a relation τ from simple terms to terms we define two new relations $\overline{\tau}$ and $\widetilde{\tau}$ from terms to terms by:

- $S \,\overline{\tau}\, S'$ if $S = \sum_{i=1}^n a_i s_i$ and $S' = \sum_{i=1}^n a_i S'_i$, where for all $i \in [1, n]$, $s_i \;\tau\; S'_i$;
- $S \,\widetilde{\tau}\, S'$ if $S = at + U$ and $S' = aT' + U$, where $a \neq 0$ and the term T' is such that $t \;\tau\; T'$.

In general, neither $\overline{\tau} \subseteq \widetilde{\tau}$ nor $\widetilde{\tau} \subseteq \overline{\tau}$ hold.

Proposition 3.5. *The following properties hold:*

(i) $\overline{\tau}$ *is the least linear relation from terms to terms that contains* τ. $\widetilde{\tau}$ *is not linear in general; however, the reflexive and transitive closure* τ^* *of* $\widetilde{\tau}$ *is.*

(ii) $\overline{\cdot}$ *and* $\widetilde{\cdot}$ *are* ω*-continuous constructions in the sense that: if* $(\tau_k)_{k\in\mathbf{N}}$ *is an increasing sequence of relations from simple terms to terms, then, denoting* $\tau = \bigcup_{k\in\mathbf{N}} \tau_k$, $\overline{\tau} = \bigcup_{k\in\mathbf{N}} \overline{\tau_k}$ *and* $\widetilde{\tau} = \bigcup_{k\in\mathbf{N}} \widetilde{\tau_k}$.

3.2 One-step reduction

We define an increasing sequence of relations from simple terms to terms by the following statements. Write ρ_0 for the empty relation. Assume ρ_k is defined, then we define ρ_{k+1} by induction on its first argument:

- if $s \;\rho_k\; S'$ then $\lambda x\, s \;\rho_{k+1}\; \lambda x\, S'$, $(s)\, T \;\rho_{k+1}\; (S')\, T$ and $\mathrm{D}^n\, s \cdot (u_1, \dots, u_n) \;\rho_{k+1}\; \mathrm{D}^n\, S' \cdot (u_1, \dots, u_n)$;
- if $\nu \;\rho_k\; N'$ then $\mu\alpha\, \nu \;\rho_{k+1}\; \mu\alpha\, N'$;
- if $T \;\widetilde{\rho_k}\; T'$ then $(s)\, T \;\rho_{k+1}\; (s)\, T'$;
- if $u_0 \;\rho_k\; U'_0$ then $\mathrm{D}^{n+1}\, s \cdot (u_0, u_1, \dots, u_n) \;\rho_{k+1}\; \mathrm{D}^{n+1}\, s \cdot (U'_0, u_1, \dots, u_n)$;
- $(\lambda x\, s)\, T \;\rho_{k+1}\; s\,[T/x]$;
- $(\mu\alpha\, \nu)\, T \;\rho_{k+1}\; \mu\alpha\, \left((\nu)_\alpha\, T\right)$, assuming $\alpha \notin T$;
- $\mathrm{D}^{n+1}\, \lambda x\, s \cdot (u_0, u_1, \dots, u_n) \;\rho_{k+1}\; \mathrm{D}^n\, \lambda x\, \left(\frac{\partial s}{\partial x} \cdot u_0\right) \cdot (u_1, \dots, u_n)$, assuming $x \notin u_0$;
- $\mathrm{D}^{n+1}\, \mu\alpha\, \nu \cdot (u_0, u_1, \dots, u_n) \;\rho_{k+1}\; \mathrm{D}^n\, \mu\alpha\, (\mathrm{D}_\alpha\, \nu \cdot u_0) \cdot (u_1, \dots, u_n)$, assuming $\alpha \notin u_0$.

Recall that, by permutative equality, the order on linear arguments of derivatives do not matter. Thus we have, *e.g.*: for all $i \in [1, n]$, if $u_i \;\rho_k\; U'_i$ then $\mathrm{D}^n\, s \cdot (u_1, \dots, u_n) \;\rho_{k+1}\; \mathrm{D}^n\, s \cdot (u_1, \dots, U'_i, \dots, u_n)$, by the fourth induction clause. Let $\rho = \bigcup_{k\in\mathbf{N}} \rho_k$.

Proposition 3.6. *By* ω*-continuity of* $\widetilde{\cdot}$, $\widetilde{\rho} = \bigcup_{k\in\mathbf{N}} \widetilde{\rho_k}$.

Lemma 3.7. *If* $s, u \in \Delta$, $S', T, T' \in \mathsf{R}\langle\Delta\rangle$, $\nu \in \Delta^\square$ *and* $N' \in \mathsf{R}\langle\Delta^\square\rangle$ *are such that* $s \;\rho\; S'$, $T \;\widetilde{\rho}\; T'$ *and* $\nu \;\rho\; N'$, *then the following relations hold:*

$$\begin{array}{rcll} \lambda x\, s & \rho & \lambda x\, S' & \\ (s)\, T & \rho & (S')\, T & \\ (s)\, T & \rho & (s)\, T' & (1) \\ \mu\alpha\, \nu & \rho & \mu\alpha\, N' & \\ [\alpha]\, s & \rho & [\alpha]\, S' & \\ \mathrm{D}\, s \cdot u & \rho & \mathrm{D}\, S' \cdot u & (2l) \\ \mathrm{D}\, u \cdot s & \rho & \mathrm{D}\, u \cdot S'. & (2r) \end{array}$$

Proof. All relations but (1) are just rephrasing the definition of ρ (together with the definition of the derivative of a simple term, for relations (2l) and (2r)). The same holds for relation (1), through proposition 3.6. □

Let ρ^* be the reflexive and transitive closure of $\widetilde{\rho}$.

Lemma 3.8. *The relation* ρ^* *is contextual.*

Proof. ρ^* is reflexive and linear (proposition 3.5). The other conditions result from reflexivity, transitivity and linearity together with lemma 3.7. $\square$

In ordinary λ-calculus, it is well known that $s\,[t/x]\ \rho\ s'\,[t/x]$ as soon as $s\ \rho\ s'$. A similar property holds in differential $\lambda\mu$-calculus, as soon as the substituted term is simple. The following remarks underline the need of such a condition. Assume s is simple, $s\ \rho\ S'$ and x is in head linear position in s:

- $s\,[0/x] = 0$ does not necessarily reduce;
- more generally, if T is not a simple term, then $s\,[T/x]$ is not simple and it may need several reduction steps before reaching $S'\,[T/x]$.

All this because $\widetilde{\rho}$ reduces only one underlying simple term at each step. The following lemma states our claim more formally.

Lemma 3.9. *If $S, S' \in \mathsf{R}\langle\Delta\rangle$ are such that $S\ \widetilde{\rho}\ S'$, then, for all variable x and all $t \in \Delta$, $S\,[t/x]\ \widetilde{\rho}\ S'\,[t/x]$.*

Proof. We prove by induction on k that if $s\ \rho_k\ S'$, resp. $S\ \widetilde{\rho_k}\ S'$, then $s\,[t/x]\ \rho\ S'\,[t/x]$, resp. $S\,[t/x]\ \widetilde{\rho}\ S'\,[t/x]$ — recall that, by lemma 2.12, if s and t are simple then $s\,[t/x]$ is simple too. If $k = 0$, both relations are empty, hence the conclusion. Assume the result holds until height k, and we have $s\ \rho_{k+1}\ S'$; we study all possible cases for this reduction. In the following cases:

- $s = \lambda y\, u$ and $S' = \lambda y\, U'$ with $u\ \rho_k\ U'$, $x \neq y$ and $y \notin t$;
- $s = \mu\beta\, \nu$ and $S' = \mu\beta\, N'$ with $\nu\ \rho_k\ N'$;
- $s = (u)\, V$ and $S' = (U')\, V$ with $u\ \rho_k\ U'$;
- $s = (u)\, V$ and $S' = (u)\, V'$ with $V\ \widetilde{\rho_k}\ V'$;
- $s = \mathrm{D}^n\, u \cdot (v_1, \dots, v_n)$ and $S' = \mathrm{D}^n\, U' \cdot (v_1, \dots, v_n)$ with $u\ \rho_k\ U'$;
- $s = \mathrm{D}^n\, u \cdot (v_1, \dots, v_n)$ and $S' = \mathrm{D}^n\, u \cdot (V'_1, \dots, V'_n)$ with $v_i\ \rho_k\ V'_i$ for some i and $V'_j = v_j$ for all $j \neq i$;

we can apply induction hypothesis to those reductions at height k we mention, and we get $s\,[t/x]\ \rho\ S'\,[t/x]$ by lemma 3.7 and the definition of substitution. In the following cases:

- $s = (\lambda y\, u)\, V$ and $S' = u\,[V/y]$ with $y \neq x$ and $y \notin t$;
- $s = (\mu\beta\, \nu)\, V$ and $S' = \mu\beta\, (\nu)_\beta\, V$ with $\beta \notin t \cup V$;
- $s = \mathrm{D}^{n+1}\, \lambda y\, u \cdot (v, v_1, \dots, v_n)$ and $S' = \mathrm{D}^n \left(\lambda y\, \frac{\partial u}{\partial y} \cdot v\right) \cdot (v_1, \dots, v_n)$ with $y \neq x$ and $y \notin t \cup v$;
- $s = \mathrm{D}^{n+1}\, \mu\beta\, \nu \cdot (v, v_1, \dots, v_n)$ and $S' = \mathrm{D}^n\, (\mu\beta\, \mathrm{D}_\alpha\, \nu \cdot v) \cdot (v_1, \dots, v_n)$ with $\beta \notin t \cup v$;

the result is direct by definition of ρ and lemma 2.15. Now assume $S\ \widetilde{\rho_{k+1}}\ S'$: we have $S = au + V$ for some scalar $a \neq 0$, $u \in \Delta$ and $V \in \mathsf{R}\langle\Delta\rangle$, and $S' = aU' + V$ with $u\ \rho_{k+1}\ U'$. We have just proved that $u\,[t/x]\ \rho\ U'\,[t/x]$, and we have $S\,[t/x] = a\,(u\,[t/x]) + V\,[t/x]$ and $S'\,[t/x] = a\,(U'\,[t/x]) + V\,[t/x]$. Since $u\,[t/x]$ is simple, this matches the definition of $\widetilde{\rho}$, and we have $S\,[t/x]\ \widetilde{\rho}\ S'\,[t/x]$. $\square$

In $\lambda\mu$-calculus, the same property holds for named application $(s)_\alpha\, t\ \rho\ (s')_\alpha\, t$ as soon as $s\ \rho\ s'$. Again, we recover this result in differential $\lambda\mu$-calculus. There is no need of any extra condition on the substituted term: if s is a simple term, then $(s)_\alpha\, T$ is simple too (lemma 2.12).

Lemma 3.10. *If $S, S' \in \mathsf{R}\langle\Delta\rangle$ are such that $S\ \widetilde{\rho}\ S'$ then, for all name α and all $T \in \mathsf{R}\langle\Delta\rangle$, $(S)_\alpha\, T\ \widetilde{\rho}\ (S')_\alpha\, T$.*

Proof. The proof is the same as the previous one, using lemma 2.17 in the cases involving a redex. Notice also that this result is quite close to the second formula of lemma 3.7 (or item 1 of [ER03, Lemma 10]). $\square$

As a corollary of the previous two lemmas, we obtain a sufficient condition for a term to be strongly normalizing (by definition, a term S is strongly normalizing if there is no infinite sequence of reductions from S). It will be used in the proof of adequation theorem 5.31.

Corollary 3.11. *Let $S \in \mathsf{R}\langle\Delta\rangle$. Assume there are*

- *either $x \in \mathfrak{V}$ and $t \in \Delta$, such that $S\,[t/x]$ is strongly normalizing;*

- *or* $\alpha \in \mathfrak{N}$ *and* $T \in \mathsf{R}\langle\Delta\rangle$ *such that* $(S)_\alpha T$ *is strongly normalizing;*

then S *is strongly normalizing.*

Proof. We prove the contrapositive. Assume there is an infinite sequence of reductions from S:

$$S = S_0 \, \widetilde{\rho} \, S_1 \, \widetilde{\rho} \cdots \widetilde{\rho} \, S_n \, \widetilde{\rho} \cdots$$

then by lemmas 3.9 and 3.10, there are infinite sequences of reduction both from $S\,[t/x]$ and $(S)_\alpha T$:

$$S\,[t/x] = S_0\,[t/x] \, \widetilde{\rho} \, S_1\,[t/x] \, \widetilde{\rho} \cdots \widetilde{\rho} \, S_n\,[t/x] \, \widetilde{\rho} \cdots$$

and

$$(S)_\alpha T = (S_0)_\alpha T \, \widetilde{\rho} \, (S_1)_\alpha T \, \widetilde{\rho} \cdots \widetilde{\rho} \, (S_n)_\alpha T \, \widetilde{\rho} \cdots$$

□

3.3 Parallel reduction

We define an increasing sequence of relations from simple terms to terms by the following statements. Write π_0 for the identity relation. Assume π_k is defined, then we define π_{k+1} by induction on its first argument. If $s \; \pi_k \; S'$, $T \; \overline{\pi_k} \; T'$, $\nu \; \pi_k \; N'$ and, for all $i \in [1,n]$, $u_i \; \pi_k \; U'_i$, then we set:

- $\lambda x\, s \; \pi_{k+1} \; \lambda x\, S'$;
- $\mu\alpha\, \nu \; \pi_{k+1} \; \mu\alpha\, N'$;
- $(s)\, T \; \pi_{k+1} \; (S')\, T'$;
- $\mathrm{D}^n\, s \cdot (u_1, \ldots, u_n) \; \pi_{k+1} \; \mathrm{D}^n\, S' \cdot (U'_1, \ldots, U'_n)$;
- $(\mathrm{D}^n\, \lambda x\, s \cdot (u_1, \ldots, u_n))\, T \; \pi_{k+1} \; \left(\frac{\partial^n S'}{\partial x^n} \cdot (U'_1, \ldots, U'_n)\right) [T'/x]$, assuming $x \notin U'_1 \cup \cdots \cup U'_n$;
- $(\mathrm{D}^n\, \mu\alpha\, \nu \cdot (u_1, \ldots, u_n))\, T \; \pi_{k+1} \; \mu\alpha\, (\mathrm{D}^n_\alpha\, N' \cdot (U'_1, \ldots, U'_n))_\alpha\, T'$, assuming $\alpha \notin U'_1 \cup \cdots \cup U'_n \cup T'$;
- for all $p \in [1,n]$, $\mathrm{D}^n\, \lambda x\, s \cdot (u_1, \ldots, u_n) \; \pi_{k+1} \; \mathrm{D}^{n-p}\, \lambda x \left(\frac{\partial^p S'}{\partial x^p} \cdot \left(U'_1, \ldots, U'_p\right)\right) \cdot \left(U'_{p+1}, \ldots, U'_n\right)$, assuming $x \notin U'_1 \cup \cdots \cup U'_p$;
- for all $p \in [1,n]$, $\mathrm{D}^n\, \mu\alpha\, \nu \cdot (u_1, \ldots, u_n) \; \pi_{k+1} \; \mathrm{D}^{n-p}\, \mu\alpha \left(\mathrm{D}^p_\alpha\, N' \cdot \left(U'_1, \ldots, U'_p\right)\right) \cdot \left(U'_{p+1}, \ldots, U'_n\right)$, assuming $\alpha \notin U'_1 \cup \cdots \cup U'_p$.

By permutative equality, we also have have, *e.g.*: for all $I \subseteq [1,n]$, if we write $J = [1,n] \setminus I$ and $p = |I|$, then

$$\mathrm{D}^n\, \mu\alpha\, \nu \cdot (u_1, \ldots, u_n) \; \pi_{k+1} \; \mathrm{D}^{n-p}\, \mu\alpha \left(\mathrm{D}^p_\alpha\, N' \cdot U'_I\right) \cdot U'_J$$

by the last induction clause.

Remark 3.12 (Redexes)**.** Notice that the archetypical shape of an application redex is

$$(\mathrm{D}^n\, \lambda x\, v \cdot (w_1, \ldots, w_n))\, T$$

or

$$(\mathrm{D}^n\, \mu\alpha\, \nu \cdot (w_1, \ldots, w_n))\, T$$

and not only $(\lambda x\, v)\, T$ or $(\mu\alpha\, \nu)\, T$. This is necessary for lemmas 3.18 and 3.20 to hold (see, *e.g.*, the first two items of the proof of lemma 3.18 in appendix A), and more generally the diamond property for parallel reduction (lemma 3.24). It is also in accordance with the intuitions of section 5, in which we define an elimination as a couple $(\langle u_1, \ldots, u_n\rangle, T)$: this is a generalized notion of argument, in which $\langle u_1, \ldots, u_n\rangle$ is a multiset of linear arguments, and T is an intuitionistic argument.

Let $\pi = \bigcup_{k \in \mathbf{N}} \pi_k$.

Proposition 3.13. *By* ω*-continuity of* $\bar{\cdot}$, $\overline{\pi} = \bigcup_{k \in \mathbf{N}} \overline{\pi_k}$.

Lemma 3.14. *The relation* $\overline{\pi}$ *is contextual.*

Proof. $\overline{\pi}$ is linear as stated in proposition 3.5 and $\overline{\pi_0} \subset \overline{\pi}$ is clearly reflexive. Like in lemma 3.7, the other conditions are just rephrasing the definitions of π and $\overline{\pi}$, with the notable exception of the application case which involves proposition 3.13. □

Corollary 3.15. *Let $S, S', T, T', U_1, \ldots, U_n, U'_1, \ldots, U'_n \in \mathsf{R}\langle\Delta\rangle$, and $N, N' \in \mathsf{R}\langle\Delta^{\square}\rangle$. With assumptions similar to those of the definition of parallel reduction, we get:*

$$\begin{array}{rcl}
\mathrm{D}^n\,\lambda x\, S \cdot (U_1, \ldots, U_n) & \overline{\pi} & \mathrm{D}^{n-p}\,\lambda x \left(\dfrac{\partial^p S'}{\partial x^p} \cdot U'_I\right) \cdot U'_J \\
\mathrm{D}^n\,\mu\alpha\, N \cdot (U_1, \ldots, U_n) & \overline{\pi} & \mathrm{D}^{n-p}\,\mu\alpha \left(\mathrm{D}^p_\alpha\, N' \cdot U'_I\right) \cdot U'_J \\
(\mathrm{D}^n\,\lambda x\, S \cdot (U_1, \ldots, U_n))\, T & \overline{\pi} & \left(\dfrac{\partial^n S'}{\partial x^n} \cdot \left(U'_1, \ldots, U'_n\right)\right) \left[T'/x\right] \\
(\mathrm{D}^n\,\mu\alpha\, N \cdot (U_1, \ldots, U_n))\, T & \overline{\pi} & \left(\mathrm{D}^n_\alpha\, S' \cdot \left(U'_1, \ldots, U'_n\right)\right)_\alpha T'.
\end{array}$$

Lemma 3.16. $\rho \subset \pi \subset \rho^*$.

Proof. $\rho \subset \pi$ should be clear. $\pi \subset \rho^*$ follows from contextuality of ρ^*. □

3.4 Reductions and operations

The following four lemmas state that parallel reduction is compatible with operations on terms. This is a key preliminary result to prove that it has the diamond property.

Lemma 3.17. *Let x be a variable and S, U, S', U' be terms. If $S\,\overline{\pi}\,S'$ and $U\,\overline{\pi}\,U'$ then*

$$S\,[U/x]\,\overline{\pi}\,S'\left[U'/x\right].$$

Proof. One proves by induction on k that if $S\,\overline{\pi_k}\,S'$ and $U\,\overline{\pi}\,U'$ then $S\,[U/x]\,\overline{\pi}\,S'\,[U'/x]$, using lemma 2.15 in redex cases. The reader may refer to the full proof given in appendix A. □

Lemma 3.18. *Let x be a variable and S, U, S', U' be terms. If $S\,\overline{\pi}\,S'$ and $U\,\overline{\pi}\,U'$ then*

$$\frac{\partial S}{\partial x} \cdot U\,\overline{\pi}\,\frac{\partial S'}{\partial x} \cdot U'.$$

Proof. The proof is very similar to the previous one, using lemma 2.16 in redex cases. Key cases are given in appendix A. □

Lemma 3.19. *Let α be a name and S, U, S', U' be terms. If $S\,\overline{\pi}\,S'$ and $U\,\overline{\pi}\,U'$ then*

$$(S)_\alpha\, U\,\overline{\pi}\,\left(S'\right)_\alpha U'.$$

Proof. Again, the proof is practically the same. It involves lemma 2.17 in redex cases. □

Lemma 3.20. *Let α be a name and S, U, S', U' be terms. If $S\,\overline{\pi}\,S'$ and $U\,\overline{\pi}\,U'$ then*

$$\mathrm{D}_\alpha\, S \cdot U\,\overline{\pi}\,\mathrm{D}_\alpha\, S' \cdot U'.$$

Proof. The same as before, using lemma 2.18 in redex cases. □

From the previous four lemmas and the inclusions $\rho \subseteq \pi \subseteq \rho^*$, we can derive very similar results for ρ^*:

Corollary 3.21. *Let x be a variable and S, U, S', U' be terms. If $S\,\rho^*\,S'$ and $U\,\rho^*\,U'$ then*

$$\begin{array}{rcl}
S\,[U/x] & \rho^* & S'\left[U'/x\right] \\
\dfrac{\partial S}{\partial x} \cdot U & \rho^* & \dfrac{\partial S'}{\partial x} \cdot U' \\
(S)_\alpha\, U & \rho^* & \left(S'\right)_\alpha U' \\
\mathrm{D}_\alpha\, S \cdot U & \rho^* & \mathrm{D}_\alpha\, S' \cdot U'.
\end{array}$$

3.5 Confluence

Definition 3.22. We define inductively on term S its full parallel reduction $S{\downarrow}$ by:

$$\begin{array}{rcl}
x{\downarrow} & = & x \\
((s)\,T){\downarrow} & = & \left\{\begin{array}{ll} \left(\frac{\partial^n v{\downarrow}}{\partial x^n}\cdot(u_1{\downarrow},\ldots,u_n{\downarrow})\right)[T{\downarrow}/x] & \text{if } s = \mathrm{D}^n\,\lambda x\,v\cdot(u_1,\ldots,u_n) \\ \mu\alpha\left((\mathrm{D}^n_\alpha\,\nu{\downarrow}\cdot(u_1{\downarrow},\ldots,u_n{\downarrow}))_\alpha\,T{\downarrow}\right) & \text{if } s = \mathrm{D}^n\,\mu\alpha\,\nu\cdot(u_1,\ldots,u_n) \\ (s{\downarrow})\,T{\downarrow} & \text{otherwise} \end{array}\right. \\
(\mathrm{D}^n\,s\cdot(u_1,\ldots,u_n)){\downarrow} & = & \left\{\begin{array}{ll} \lambda x\left(\frac{\partial^n v{\downarrow}}{\partial x^n}\cdot(u_1{\downarrow},\ldots,u_n{\downarrow})\right) & \text{if } s = \lambda x\,v \\ \mu\alpha\,(\mathrm{D}^n_\alpha\,\nu{\downarrow}\cdot(u_1{\downarrow},\ldots,u_n{\downarrow})) & \text{if } s = \mu\alpha\,\nu \\ \mathrm{D}^n\,s{\downarrow}\cdot(u_1{\downarrow},\ldots,u_n{\downarrow}) & \text{otherwise} \end{array}\right. \\
([\alpha]\,s){\downarrow} & = & [\alpha]\,s{\downarrow} \\
S{\downarrow} & = & \sum_{s\in\Delta} S_{(s)}s{\downarrow}
\end{array}$$

Lemma 3.23. *For all $S \in \mathsf{R}\langle\Delta\rangle$, $S\ \overline{\pi}\ S{\downarrow}$.*

Proof. This is straightforward by induction on terms. □

Lemma 3.24. *If S and S' are terms such that $S\ \overline{\pi}\ S'$, then $S'\ \overline{\pi}\ S{\downarrow}$.*

Proof. One proves by induction on k that if $s\ \pi_k\ S'$ then $S'\ \overline{\pi}\ s{\downarrow}$, and if $S\ \overline{\pi_k}\ S'$ then $S'\ \overline{\pi}\ S{\downarrow}$, using lemmas 3.17 to 3.20 in redex cases. A full proof is given in appendix A. □

Theorem 3.25. *Relation $\overline{\pi}$ enjoys the diamond property. Hence relation $\widetilde{\rho}$ is confluent, in the sense of Church-Rosser: if $S\ \widetilde{\rho}\ T$ and $S\ \widetilde{\rho}\ T'$ then there exists $U \in \mathsf{R}\langle\Delta\rangle$ such that $T\ \rho^*\ U$ and $T'\ \rho^*\ U$.*

Proof. The diamond property for $\overline{\pi}$ is a straightforward corollary of lemma 3.24. It implies confluence of $\widetilde{\rho}$ by lemma 3.16. □

Remark 3.26 (Trivia)**.** There is a case in which confluence is much easier to establish: if 1 admits an opposite $-1 \in \mathsf{R}$. In this case, assume $S\ \rho^*\ S'$. Since ρ^* is linear and reflexive, $S' = S' - S + S\ \rho^*\ S$. Hence ρ^* is symmetric, which obviously implies Church-Rosser. In this case reduction is even more degenerate: as we shall see below, $S\ \rho^*\ T$ for all terms S and T.

3.6 Subcalculi

Notice that any ordinary $\lambda\mu$-term is also a simple term of differential $\lambda\mu$-calculus. Let Λ denote the set of all $\lambda\mu$-terms and $\beta \subset \Lambda\times\Lambda$ the usual β-reduction of $\lambda\mu$-calculus. It is clear that $\beta \subset \rho$.

Denote by $\leftrightarrow_\rho$ the reflexive, symmetric and transitive closure of $\widetilde{\rho}$ and $\leftrightarrow_\beta$ the usual β-equivalence of $\lambda\mu$-calculus.

Lemma 3.27. *Differential $\lambda\mu$-calculus preserves the equalities of $\lambda\mu$-calculus,* i.e. *for all $\lambda\mu$-terms s, t, $s \leftrightarrow_\beta t \Rightarrow s \leftrightarrow_\rho t$.*

Proof. This is a straightforward consequence of the confluence of β and the fact that $\beta \subset \widetilde{\rho}$. □

Conservativity. One may wonder if the reverse also holds, *i.e.* if equivalence classes of $\lambda\mu$-terms in differential $\lambda\mu$-calculus are the same as in ordinary $\lambda\mu$-calculus. In [ER03, Proposition 19], Erhrard and Régnier assert that confluence easily enforces such a result for λ-calculus, without proof. If R is $\mathbf{N}$ or $\mathbf{N}[\chi_1,\ldots,\chi_n]$, then $\widetilde{\rho}$-reductions from $\lambda\mu$-terms are exactly β-reductions, and the result holds by the same argument as in lemma 3.27. In the general case, however, a $\lambda\mu$-term does not necessarily reduce to another $\lambda\mu$-term, hence the proof is not as easy as Ehrhard and Régnier first seemed to believe.

Recall that a rig is said to be positive if, for all two elements a and b, $a + b = 0$ implies $a = b = 0$. In the following, we prove that $\leftrightarrow_\rho \cap (\Lambda\times\Lambda) = \leftrightarrow_\beta$ as soon as R is positive. Write 2^Λ for the set of all subsets of Λ.

Definition 3.28. Let Δ' denote the set of all simple terms that do not contain any derivative, *i.e.* all the subterms of which are pre-terms only. We define $\Lambda(\cdot) : \mathsf{R}\langle\Delta'\rangle \longrightarrow 2^{\Lambda}$ by the following statements:

$$\begin{array}{rcl} \Lambda(x) & = & \{x\} \\ \Lambda(\lambda x\, s) & = & \{\lambda x\, t;\ t \in \Lambda(s)\} \\ \Lambda(\mu\alpha\, \nu) & = & \{\mu\alpha\, \nu';\ \nu' \in \Lambda(\nu)\} \\ \Lambda([\alpha]\, s) & = & \{[\alpha]\, t;\ t \in \Lambda(s)\} \\ \Lambda((s)\, T) & = & \{(u)\, v;\ u \in \Lambda(s) \wedge v \in \Lambda(T)\} \\ \Lambda(S) & = & \bigcup_{s \in \mathrm{Supp}(S)} \Lambda(s) \end{array}$$

Proposition 3.29. *If $s \in \Lambda$, then $\Lambda(s) = \{s\}$.*

Lemma 3.30. *If R is positive, and terms $S \in \mathsf{R}\langle\Delta'\rangle$ and $S' \in \mathsf{R}\langle\Delta\rangle$ are such that $S\, \widetilde{\rho}\, S'$, then $S' \in \mathsf{R}\langle\Delta'\rangle$ and, for all $t' \in \Lambda(S')$, there exists $t \in \Lambda(S)$ such that $t\ \beta^*\ t'$.*

Proof. The proof is by induction on the height of the reduction $S\, \widetilde{\rho}\, S'$. All induction steps are straightforward, except for the extension from ρ_k to $\widetilde{\rho_k}$: assume $S = au + V$ and $S' = aU' + V$ with $a \neq 0$, $u\ \rho_k\ U'$ and $V \in \mathsf{R}\langle\Delta\rangle$; since R is positive, $u \in \Delta'$ and $V \in \mathsf{R}\langle\Delta'\rangle$. By induction hypothesis, $U' \in \mathsf{R}\langle\Delta'\rangle$ and, for all $w' \in \Lambda(U')$, there exists $w \in \Lambda(u)$ such that $w\ \beta^*\ w'$. Now assume $w' \in \Lambda(S')$:

- either $w' \in \Lambda(aU') \subseteq \Lambda(U')$, and we have just shown that this implies there exists $w \in \Lambda(u)$ such that $w\ \beta^*\ w'$;
- or $w' \in \Lambda(V)$ and we can chose $w = w'$.

□

Corollary 3.31. *If R is positive, and terms $s \in \Lambda$ and $T \in \mathsf{R}\langle\Delta\rangle$ are such that $s\ \rho^*\ T$, then $T \in \mathsf{R}\langle\Delta'\rangle$ and, for all $t \in \Lambda(T)$, $s\ \beta^*\ t$.*

Lemma 3.32. *If $S, T \in \mathsf{R}\langle\Delta\rangle$ are such that $S\ \overline{\pi}\ T$ then $S{\downarrow}\ \overline{\pi}\ T{\downarrow}$.*

Proof. The proof is easy and very close to that of lemma 3.24. □

We define iterated full reduction by $S{\downarrow}^0 = S$ and $S{\downarrow}^{n+1} = (S{\downarrow}^n){\downarrow}$.

Lemma 3.33. *If S, T are terms and $S\ \overline{\pi}^n\ T$ then $T\ \rho^*\ S{\downarrow}^n$*

Proof. The proof is by induction on n. If $n = 0$, $S = T = S{\downarrow}^0$ and this is reflexivity of ρ^*. Assume the result holds at rank n. If $S\ \overline{\pi}^n\ T\ \overline{\pi}\ T'$, then, by induction hypothesis, $T\ \rho^*\ S{\downarrow}^n$. Since ρ^* is also the transitive closure of $\overline{\pi}$, lemma 3.32 entails $T{\downarrow}\ \rho^*\ S{\downarrow}^{n+1}$. By lemma 3.24, we have $T'\ \overline{\pi}\ T{\downarrow}$, hence $T'\ \rho^*\ S{\downarrow}^{n+1}$. □

Theorem 3.34. *If R is positive, and terms $s, t \in \Lambda$ are such that $s \leftrightarrow_\rho t$ then $s \leftrightarrow_\beta t$.*

Proof. Assume $s, t \in \Lambda$ and $s \leftrightarrow_\rho t$. By the Church-Rosser property of $\widetilde{\rho}$ (theorem 3.25), there exists $U \in \mathsf{R}\langle\Delta\rangle$ such that $s\ \rho^*\ U$ and $t\ \rho^*\ U$. By lemma 3.33, there exists some $n \in \mathbf{N}$ such that $U\ \rho^*\ v = s{\downarrow}^n$. Notice that if $w \in \Lambda$, then $w{\downarrow} \in \Lambda$, hence $v \in \Lambda$. We have $s\ \rho^*\ v$ and $t\ \rho^*\ v$, hence by positivity of R and lemma 3.31, for all $v' \in \Lambda(v)$ there are $s' \in \Lambda(s)$ and $t' \in \Lambda(t)$ such that $s'\ \beta^*\ v'$ and $t'\ \beta^*\ v'$. By proposition 3.29, $\Lambda(s) = \{s\}$, $\Lambda(t) = \{t\}$ and $\Lambda(v) = \{v\}$, hence the conclusion. □

Collapse. If R is a commutative rig which is not positive, then there are $a, b \in \mathsf{R}$ such that $a+b=0$ and $a \neq 0$ (hence $b \neq 0$). Write Ψ for some fixpoint operator of pure λ-calculus such that $(\Psi)\, f\ \beta^*\ (f)\,(\Psi)\, f$ for all λ-term f (for instance, Turing's fixpoint). Then, for all term $S \in \mathsf{R}\langle\Delta\rangle$, we define $\Sigma_S = (\Psi)\,\lambda x\,(S+x)$. Let $y \in \mathfrak{V}$ be a fresh variable: we have $(\Psi)\, y\ \beta^*\ (y)\,(\Psi)\, y$, hence $(\Psi)\, y\ \rho^*\ (y)\,(\Psi)\, y$ and lemma 3.21 implies

$$\Sigma_S = ((\Psi)\, y)\,[\lambda x\,(S+x)/y]\ \rho^*\ ((y)\,(\Psi)\, y)\,[\lambda x\,(S+x)/y] = (\lambda x\,(S+x))\,\Sigma_S.$$

We get

$$\Sigma_S\ \rho^*\ S + \Sigma_S.$$

Then, by contextuality,

$$0 = a\Sigma_S + b\Sigma_S\ \rho^*\ a(S+\Sigma_S) + b\Sigma_S = aS$$

and

$$aS = aS + a\Sigma_S + b\Sigma_S\ \rho^*\ aS + a\Sigma_S + b(S+\Sigma_S) = 0.$$

Hence, for all terms S and T, $aS\ \rho^*\ aT$. In particular, if $1 \in \mathsf{R}$ has an opposite $-1 \in \mathsf{R}$, then $S\ \rho^*\ T$ for all terms S and T.

Differential λ-calculus. Now we compare the equational theory of differential λ-calculus over some R with that of differential $\lambda\mu$-calculus over the same R. Write $\mathsf{R}\langle\Delta_\lambda\rangle$ for the set of all differential λ-terms with coefficients in R: these are terms as defined in subsection 2.2, removing the μ-abstraction and naming cases. Write $\widetilde{\rho}_\lambda$ for one-step reduction in the corresponding differential λ-calculus and $\leftrightarrow_{\rho,\lambda}$ for the reflexive, symmetric and transitive closure of $\widetilde{\rho}_\lambda$.

It is clear that $\mathsf{R}\langle\Delta_\lambda\rangle \subseteq \mathsf{R}\langle\Delta\rangle$ and, by confluence of $\widetilde{\rho}_\lambda$ and the fact that $\widetilde{\rho}_\lambda \subset \widetilde{\rho}$, we easily get $\leftrightarrow_{\rho,\lambda} \subset \leftrightarrow_\rho$. Moreover, if R is positive, it is easily seen that $\widetilde{\rho} \cap \mathsf{R}\langle\Delta_\lambda\rangle \times \mathsf{R}\langle\Delta_\lambda\rangle = \widetilde{\rho}_\lambda$; by confluence of $\widetilde{\rho}$, this implies that $\leftrightarrow_{\rho,\lambda}$ is exactly the restriction of $\leftrightarrow_\rho$ to differential λ-terms.

If R is not positive, reduction is degenerate in both differential λ-calculus and differential $\lambda\mu$-calculus so that conservativity has little meaning. This, however, does not hamper the fact that most of the results of this paper apply directly to differential λ-calculus: just strip everything concerning the μ side of the calculus.

4 Type system

Like differential λ-calculus, differential $\lambda\mu$-calculus can be typed with implicative propositional types. Assume we have a denumerable set of basic types $\phi, \psi, \ldots$, we build types from basic types using intuitionistic arrow: if A and B are types, then so is $A \to B$.

4.1 Rules

Typing judgements are those of $\lambda\mu$-calculus and typing rules are given in figure 1. These are roughly the union of the rules for typed $\lambda\mu$-calculus and typed differential λ-calculus. The notation

$$(\Gamma \vdash s : A \mid \Delta)_{s \in \mathrm{Supp}(S)}$$

stands for the sequence of all judgements of form $\Gamma \vdash s : A \mid \Delta$ where $s \in \mathrm{Supp}(S)$ — and similarly for $(\Gamma \vdash u_i : A \mid \Delta)_{i=1,\ldots,n}$. Typing derivations are finitely branching because $\mathrm{Supp}(S)$ is always finite.

Proposition 4.1. *The following two properties hold:*

(i) if $\Gamma \vdash S : A \mid \Delta$ then free variables of S are declared in Γ and free names of S are declared in Δ;

(ii) if $\Gamma \vdash S : A \mid \Delta$ then for all Γ' and Δ', $\Gamma, \Gamma' \vdash S : A \mid \Delta, \Delta'$.

4.2 Typing and reduction

Lemma 4.2. *The following two properties hold:*

(i) If $\Gamma, x : A \vdash S : B \mid \Delta$ and $\Gamma \vdash U : A \mid \Delta$, then $\Gamma \vdash S\,[U/x] : B \mid \Delta$ and $\Gamma, x : A \vdash \frac{\partial S}{\partial x} \cdot U : B \mid \Delta$.

(ii) If $\Gamma \vdash S : C \mid \alpha : A \to B, \Delta$ and $\Gamma \vdash U : A \mid \Delta$, then $\Gamma \vdash (S)_\alpha U : C \mid \alpha : B, \Delta$ and $\Gamma \vdash \mathrm{D}_\alpha\, S \cdot U : C \mid \alpha : A \to B, \Delta$.

$$\frac{}{\Gamma, x : A \vdash x : A \mid \Delta}$$

$$\frac{\Gamma, x : A \vdash s : B \mid \Delta}{\Gamma \vdash \lambda x\, s : A \to B \mid \Delta} \qquad \frac{\Gamma \vdash s : A \to B \mid \Delta \quad \Gamma \vdash T : A \mid \Delta}{\Gamma \vdash (s)\, T : B \mid \Delta}$$

$$\frac{\Gamma \vdash \nu \mid \Delta, \alpha : A}{\Gamma \vdash \mu\alpha\, \nu : A \mid \Delta} \qquad \frac{\Gamma \vdash s : A \mid \Delta, \alpha : A}{\Gamma \vdash [\alpha]\, s \mid \Delta, \alpha : A}$$

$$\frac{\Gamma \vdash s : A \to B \mid \Delta \quad (\Gamma \vdash u_i : A \mid \Delta)_{i=1,\ldots,n}}{\Gamma \vdash \mathrm{D}^n\, s \cdot (u_1, \ldots, u_n) : A \to B \mid \Delta}$$

$$\frac{(\Gamma \vdash s : A \mid \Delta)_{s \in \mathrm{Supp}(S)}}{\Gamma \vdash S : A \mid \Delta}$$

Figure 1: Typing rules for differential $\lambda\mu$-calculus.

Proof. The proof is straightforward by induction on the typing derivation of $\Gamma, x : A \vdash S : B \mid \Delta$ and $\Gamma \vdash S : C \mid \alpha : A \to B, \Delta$ respectively. $\square$

Notice how the statements of lemma 4.2 closely resemble the typing rules of application and derivative. This enforces once again our choice of notations for named application and named derivative.

Theorem 4.3. *Assume* R *is positive. Subject reduction holds: if* $S\ \widetilde{\rho}\ S'$ *and* $\Gamma \vdash S : A \mid \Delta$ *then* $\Gamma \vdash S' : A \mid \Delta$.

Proof. One proves that property by induction on the typing derivation $\Gamma \vdash S : A \mid \Delta$. Each induction step is proved by inspecting all possible cases for the reduction $S\ \widetilde{\rho}\ S'$ using the previous two lemmas in the case of a redex. The positivity condition is used to handle the case in which $S = at + U$ and $S' = aT' + U$ with $U \in \mathsf{R}\langle\Delta\rangle$ and $t\ \rho\ T'$: since R is positive, $t \in \mathrm{Supp}(S)$ and $\mathrm{Supp}(U) \subseteq \mathrm{Supp}(S)$, so that we necessarily have $\Gamma \vdash t : A \mid \Delta$ and $\Gamma \vdash U : A$. $\square$

Of course, no such result holds if R is not positive: 0 (typable with any type) may reduce to some non typable term.

5 Strong normalization

Unsurprisingly, if R is not positive, there is no normal term: if $a, b \in \mathsf{R}$ are such that $a + b = 0$ and $a \neq 0$, then for all $s \in \Delta$ and all $S', T \in \mathsf{R}\langle\Delta\rangle$ such that $s\ \rho\ S'$, we have $T = as + bs + T$ and then $T\ \widetilde{\rho}\ aS' + bs + T$; hence every term T reduces. Moreover, positivity is not a sufficient condition for strong normalization to hold: if R is the set $\mathbf{Q}^+$ of non-negative rational numbers, and s and S' are typed terms such that $s\ \rho\ S'$, then there is an infinite sequence of reductions from s:

$$s \;=\; \frac{1}{2}s + \frac{1}{2}s \;\widetilde{\rho}\; \frac{1}{2}s + \frac{1}{2}S' \;\widetilde{\rho}\; \frac{1}{4}s + \frac{3}{4}S' \;\widetilde{\rho}\; \ldots \;\widetilde{\rho}\; \frac{1}{2^n}s + \frac{2^n - 1}{2^n}S' \;\widetilde{\rho}\; \ldots$$

In this section, we assume that R is finitely splitting in the following sense: for all $a \in \mathsf{R}$ the set

$$\left\{(a_1, \ldots, a_n) \in (\mathsf{R}^\bullet)^n\, ;\ n \in \mathbf{N} \text{ and } a = a_1 + \cdots + a_n\right\}$$

is finite. We also assume that the width function $\mathrm{w} : \mathsf{R} \longrightarrow \mathbf{N}$ defined by

$$\mathrm{w}\,(a) = \max\,\{n \in \mathbf{N};\ \exists a_1, \ldots, a_n \in \mathsf{R}^\bullet \text{ s.t. } a = a_1 + \cdots + a_n\}$$

is a morphism of rigs:

$$\mathrm{w}\,(a + b) = \mathrm{w}\,(a) + \mathrm{w}\,(b)$$

and

$$\mathrm{w}\,(ab) = \mathrm{w}\,(a)\,\mathrm{w}\,(b)$$

(this clearly entails $\mathrm{w}\,(0) = 0$ and $\mathrm{w}\,(1) = 1$). Hence R is positive and has no zero divisor, since it is clear that $\mathrm{w}\,(a) = 0$ iff $a = 0$. If $S \in \mathsf{R}\langle\Delta\rangle$ and $s \in \Delta$, we write $\mathrm{w}_s(S)$ for $\mathrm{w}\,\big(S_{(s)}\big)$.

Remark 5.1 (Examples)**.** Obviously, $\mathsf{R} = \mathbf{N}$ satisfies these conditions, with $\mathrm{w}\,(n) = n$ for all $n \in \mathbf{N}$. One more interesting instance is the rig of all polynomials over indeterminates $\xi_1, \ldots, \xi_n$ with non-negative integer coefficients $\mathsf{P}_n = \mathbf{N}[\xi_1, \ldots, \xi_n]$: for all $P \in \mathsf{P}_n$, $\mathrm{w}\,(P) = P(1, \ldots, 1)$. It is the archetypical rig the structure of which inspired the proof. Every rig admitting a width morphism is similar to a rig of polynomials with non-negative integer coefficients: call unitary monomials those elements $a \in \mathsf{R}$ such that $w(a) = 1$.

In the following, we prove that typed terms are strongly normalizing, under the aforementioned conditions on R. The structure of the proof is borrowed from [ER03, Section 4], using the Tait reducibility method as presented in [Kri90]. In subsection 5.1, we prove that the set of strongly normalizing terms is exactly the module generated by strongly normalizing simple terms: this uses the conditions we introduced on R and is essential in the following. In section 5.2, we define saturated sets of terms: intuitively, these are sets closed under backwards reduction. In section 5.3, we interpret types into some particular saturated subsets of the set of strongly normalizing terms. We adapt ideas by Parigot in [Par97] in order to extend that notion of reducibility to the types associated with names. In section 5.4, we prove that typed terms lie in the interpretation of their types: this entails strong normalization. Last, in section 5.5, we formalize the weak normalization scheme outlined in [ER03].

5.1 Strongly normalizing terms

Lemma 5.2. *If $S = aT + U \in \mathsf{R}\langle\Delta\rangle$ with $a \neq 0$ then $\mathrm{Supp}(T) \subseteq \mathrm{Supp}(S)$.*

Proof. This is just positivity of R together with the fact that R has no zero divisor. □

Lemma 5.3. *Let $S \in \mathsf{R}\langle\Delta\rangle$. There are only finitely many terms S' such that $S\ \widetilde{\rho}\ S'$.*

Proof. The proof is by induction on S. If $S = 0$ the property holds trivially by lemma 5.2. Assume that the property holds for all terms in $\mathsf{R}\langle\Delta_k\rangle$. Let $S \in \mathsf{R}\langle\Delta_{k+1}\rangle$. For each term S' such that $S\ \widetilde{\rho}\ S'$, there are $t \in \Delta$, $T', U \in \mathsf{R}\langle\Delta\rangle$ and $a \in \mathsf{R}$ such that $S = at + U$, $S' = aT' + U$, $t\ \rho\ T'$ and $a \neq 0$. By lemma 5.2, $t \in \mathrm{Supp}(S) \subset \Delta_{k+1}$: there are finitely many such simple terms. Moreover, due to the finite splitting condition on R, for each such t there exist finitely many a and U such that S can be written $at + U$. A simple inspection of the definition of ρ shows that, by inductive hypothesis applied to subterms of t (all of them belong in $\mathsf{R}\langle\Delta_k\rangle$), t ρ-reduces to finitely many terms, which are all the possible choices for T'. □

König's lemma hence allows the following definition, denoting by $\mathcal{N}$ the set of strongly normalizing simple terms.

Definition 5.4. If S is a strongly normalizing term, we denote by $|S|$ the length of the longest sequence of $\widetilde{\rho}$-reductions of S to its normal form. If $S \in \mathsf{R}\langle\mathcal{N}\rangle$, we define $\|S\| = \sum_{s \in \mathcal{N}} \mathrm{w}_s(S)\,|s|$.

Lemma 5.5. *For all $a \in \mathsf{R}$ and all $S, T \in \mathsf{R}\langle\Delta\rangle$:*

$$\|aS\| = \mathrm{w}\,(a)\,\|S\|$$

and

$$\|S + T\| = \|S\| + \|T\|\,.$$

Proof. This holds just by definition of $\|\cdot\|$, together with the width function being a morphism. □

Proposition 5.6. *For all S, S' such that $S\ \widetilde{\rho}\ S'$, $S + U\ \widetilde{\rho}\ S' + U$ also holds. Hence the support of every strongly normalizing term S is a finite subset of $\mathcal{N}$,* i.e. *$S \in \mathsf{R}\langle\mathcal{N}\rangle$.*

Lemma 5.7. *Let $s \in \mathcal{N}$ and let S' be such that $s\ \rho\ S'$. Then $\|S'\| < |s|$.*

Proof. The term $S' = \sum_{t\in\mathcal{N}} S'_{(t)} t$ can be written $S' = \sum_{t\in\mathcal{N}} \sum_{i=1}^{\mathrm{w}_t(S')} a_i^t t$ where, for all $t \in \mathcal{N}$, $\sum_{i=1}^{\mathrm{w}_t(S')} a_i^t = S'_{(t)}$ and, for all i, $a_i^t \neq 0$. For each $t \in \mathcal{N}$ such that $S'_{(t)} \neq 0$, one can find a reduction of length $|t|$ from t to its normal form and concatenating these reductions, we get a reduction from S' of length $\|S'\|$. Hence $\|S'\| + 1 \leq |s|$. $\square$

Lemma 5.8. *The set of all strongly normalizing terms is* $\mathsf{R}\langle\mathcal{N}\rangle$.

Proof. It remains to prove that if $S \in \mathsf{R}\langle\mathcal{N}\rangle$ then S is strongly normalizing. This is proved by induction on $\|S\|$.

- If $\|S\| = 0$, then positivity of R implies that for all $s \in \mathrm{Supp}(S)$, $\mathrm{w}_s(S)\,|s| = 0$: since R has no zero divisor, we have $|s| = 0$. Hence, as soon as S can be written $as + T$ with $a \neq 0$, since lemma 5.2 implies $s \in \mathrm{Supp}(S)$, s is normal and doesn't give rise to a reduction from S.
- Suppose the result holds for all $T \in \mathsf{R}\langle\mathcal{N}\rangle$ such that $\|T\| < \|S\|$. It is sufficient to prove that, for all S' such that $S \mathrel{\widetilde{\rho}} S'$, S' is strongly normalizing. Such an S' is given by $a \in \mathsf{R}^\bullet$, $u \in \Delta$ and $T, U' \in \mathsf{R}\langle\Delta\rangle$ such that $S = au+T$, $u \mathrel{\rho} U'$ and $S' = aU'+T$. By lemma 5.2 and since $\mathrm{Supp}(S) \subset \mathcal{N}$, $u \in \mathcal{N}$ (so U' has to be strongly normalizing, which by proposition 5.6 implies $U' \in \mathsf{R}\langle\mathcal{N}\rangle$) and $\mathrm{Supp}(T) \subset \mathcal{N}$; hence $S' \in \mathsf{R}\langle\mathcal{N}\rangle$. By lemma 5.5, we have $\|S'\| = \|aU' + T\| = \mathrm{w}(a)\,\|U'\| + \|T\|$ and $\|S\| = \|au + T\| = \mathrm{w}(a)\,|u| + \|T\|$ with, by lemma 5.7, $\|U'\| < |u|$. Since $\mathrm{w}(a) \neq 0$, we get $\|S'\| < \|S\|$ and induction hypothesis applies.

$\square$

Remark 5.9 (About sufficient conditions)**.** Ehrhard and Régnier proved lemma 5.8 in [ER03, Lemma 26] under the assumption that $\mathsf{R} = \mathbf{N}$. They claimed their proof (which we closely reproduced) could be easily extended to any commutative rig R such that:

- R is positive;
- $\forall a, b \in \mathsf{R}$, if $ab = 0$ then either $a = 0$ or $b = 0$;
- $\forall a \in \mathsf{R}$, there are finitely many $b, c \in \mathsf{R}$ such that $a = b + c$.

First these conditions do not directly imply the width function is well defined (unless R is also simplifiable, *i.e.* $\forall a, b, c,\ a + c = b + c \Rightarrow a = b$), whereas its existence is crucial in the proof of lemma 5.7 [ER03, Lemma 25]. Moreover, in the proof of lemma 5.8, we need $\mathrm{w}(\cdot)$ to be a morphism in order to get $\|aS + bT\| = \mathrm{w}(a)\,\|S\| + \mathrm{w}(b)\,\|T\|$. Recall that if $\mathsf{R} = \mathbf{N}$ then $\mathrm{w}(n) = n$, which may explain why the role of the width function was not clear in their proof.

5.2 Saturated sets

Definition 5.10. Let $\mathcal{X}$ be a set of simple terms. An $\mathcal{X}$-elimination e is a couple:

$$e = (\langle u_1, \ldots, u_n\rangle, T) \in \mathcal{M}_{\mathrm{fin}}(\mathcal{X}) \times \mathsf{R}\langle\mathcal{X}\rangle.$$

An $\mathcal{X}$-stack π is a sequence of $\mathcal{X}$-eliminations

$$\pi = e_1 \ldots e_n$$

(if $n = 0$, π is the empty stack ε).

In the following, we write more concisely $e = (u_1, \ldots, u_n; T)$. We simply call eliminations all Δ-eliminations and stacks all Δ-stacks. Eliminations are denoted by letters e, f, and stacks by θ, π. If e is an elimination and $\pi = e_1 \ldots e_n$ is a stack we write $e :: \pi$ for the stack $ee_1 \ldots e_n$. If $\theta = f_1 \ldots f_m$ is another stack, we write $\pi\theta$ for the concatenation $e_1 \ldots e_n f_1 \ldots f_m$. Notice that any $\mathcal{X}$-elimination e can be considered as an $\mathcal{X}$-stack of length 1 so that definitions and results about stacks generally apply to eliminations.

One generalizes application, substitution and named application with stacks in argument position as follows:

Definition 5.11. If $e = (u_1, \ldots, u_n; T)$ is an $\mathcal{X}$-elimination and s is a simple term, one defines:

$$\begin{aligned} (s)\,e &= (\mathrm{D}^n\, s \cdot (u_1, \ldots, u_n))\, T \\ s\,[e/x] &= \frac{\partial^n s}{\partial x^n} \cdot (u_1, \ldots, u_n)\,[T/x] \\ (s)_\alpha\, e &= (\mathrm{D}^n_\alpha\, s \cdot (u_1, \ldots, u_n))_\alpha\, T \end{aligned}$$

assuming, of course, that $x \notin u_1 \cup \cdots \cup u_n$ and $\alpha \notin u_1 \cup \cdots \cup u_n$. In the case of application and named application, if $\pi = e_1 \ldots e_m$ is a stack, we also set:

$$\begin{aligned} (s)\,\pi &= (s)\,e_1 \ldots e_m \\ (s)_\alpha\, \pi &= (s)_\alpha\, e_1 \ldots e_m \end{aligned}$$

In particular, if $n = 0$, the first three equations become:

$$\begin{aligned} (s)\,(;T) &= (s)\,T \\ s\,[(;T)/x] &= s\,[T/x] \\ (s)_\alpha\,(;T) &= (s)_\alpha\, T. \end{aligned}$$

Remark 5.12 (Terminology)**.** As mentioned in remark 3.12, eliminations are the general form of arguments of a function. Hence their name: if s is a simple term of type $A \to B$, $u_1, \ldots, u_n$ are simple terms of type A, and T is a term of type A, then $(\mathrm{D}^n\, s \cdot (u_1, \ldots, u_n))\, T$ is a simple term of type B.

Proposition 5.13. *If s is a simple term and π is a stack then $(s)\,\pi$ is also a simple term.*

Definition 5.14. An $\mathcal{X}$-redex is a simple term of one of the following shapes:

$$\begin{aligned} t &= (\mathrm{D}^n\, \lambda x\, s \cdot (u_1, \ldots, u_n))\, T \\ t' &= (\mathrm{D}^n\, \mu\alpha\, \nu \cdot (u_1, \ldots, u_n))\, T \end{aligned}$$

with $s, u_1, \ldots, u_n \in \mathcal{X}$, $\nu \in \mathcal{X}^\square$ and $T \in \mathsf{R}\langle \mathcal{X} \rangle$, *i.e.* it is of shape $(\lambda x\, s)\, e$ or $(\mu\alpha\, \nu)\, e$ with e an $\mathcal{X}$-elimination. We write $\mathsf{Red}\,(t)$ and $\mathsf{Red}\,(t')$ for the sets of terms obtained by firing these redexes in one ρ-step:

- if $n = 0$, then $t = (\lambda x\, s)\, T$ and $t' = (\mu\alpha\, \nu)\, T$; we set

  $$\mathsf{Red}\,(t) = \{s\,[T/x]\}$$

  and

  $$\mathsf{Red}\,(t') = \left\{\mu\alpha\,(\nu)_\alpha\, T\right\};$$

- otherwise, $\mathsf{Red}\,(t)$ is the set of all terms

  $$\left(\mathrm{D}^{n-1}\, \lambda x \left(\frac{\partial s}{\partial x} \cdot u_i\right) \cdot u_{[1,n]\setminus\{i\}}\right) T$$

  and $\mathsf{Red}\,(t')$ is the set of all terms

  $$\left(\mathrm{D}^{n-1}\, \mu\alpha\, (\mathrm{D}_\alpha\, \nu \cdot u_j) \cdot u_{[1,n]\setminus\{j\}}\right) T.$$

Definition 5.15. A set $\mathcal{S}$ of simple terms is saturated if it satisfies the two following conditions:

- for all $\mathcal{N}$-redex t and all $\mathcal{N}$-stack π, if for all $T' \in \mathsf{Red}\,(t)$, $(T')\,\pi \in \mathsf{R}\langle \mathcal{S} \rangle$, then $(t)\,\pi \in \mathcal{S}$;
- $\mathcal{S}$ is closed under renaming of free variables and names.

These two properties of saturated sets will be used in the proof of adequation theorem 5.31. The condition about renaming is present for technical reasons.[1] The first saturation condition is the most important one, as it is essential in the proof of theorem 5.31, in the case of (λ- or μ-) abstractions. We will actually need a generalized version of this property, that is stated in next lemma and its corollary.

[1] We defined simultaneous substitution (as in $s\,[T_1, \ldots, T_n/x_1, \ldots, x_n]$) using one-variable substitution, and not as a primitive operation, as it is done in, *e.g.*, Krivine [Kri90]. Thus, $s\,[T_1, \ldots, T_n/x_1, \ldots, x_n]$ is well defined only if no x_i is free in any T_j. This enforces a constrained formulation of the latter adequation lemma (theorem 5.31), in the proof of which closedness under renaming is crucial.

Lemma 5.16. *Let $\mathcal{S}$ be a saturated subset of $\mathcal{N}$, $s \in \mathcal{N}$, $\nu \in \mathcal{N}^{\square}$ and $e = (u_1, \ldots, u_n; V)$ be an $\mathcal{N}$-elimination. Then $(\lambda x\, s)\, e \in \mathcal{S}$ as soon as*

- *for all $I \subseteq [1, n]$, $\frac{\partial^k s}{\partial x^k} \cdot u_I \in \mathsf{R}\langle\mathcal{S}\rangle$ (with $k = |I|$);*
- *$s\,[e/x] \in \mathsf{R}\langle\mathcal{S}\rangle$.*

Similarly, $(\mu\alpha\, \nu)\, e \in \mathcal{S}$ as soon as

- *for all $I \subseteq [1, n]$, $\mu\alpha\, \mathrm{D}^k_\alpha\, \nu \cdot u_I \in \mathsf{R}\langle\mathcal{S}\rangle$ (with $k = |I|$);*
- *$\mu\alpha\, (\nu)_\alpha\, e \in \mathsf{R}\langle\mathcal{S}\rangle$.*

Proof. We only prove the second part: the proof of the first part is very similar, and is also very close to the proof of [ER03, Lemma 28]. Since $t = (\mu\alpha\, \nu)\, e$ is an $\mathcal{N}$-redex and $\mathcal{S}$ is saturated, it is sufficient to show that $T' \in \mathsf{R}\langle\mathcal{S}\rangle$, for all $T' \in \mathsf{Red}\,(t)$ (apply saturation with the empty stack). The proof is by induction on n. If $n = 0$, $T' = \mu\alpha\, (\nu)_\alpha\, V$, and our hypothesis gives directly $T' \in \mathsf{R}\langle\mathcal{S}\rangle$. Assume the property holds for n; we prove it for $n + 1$. Without loss of generality, we can suppose $T' = (\mathrm{D}^n\, \mu\alpha\, (\mathrm{D}_\alpha\, \nu \cdot u_{n+1}) \cdot \langle u_1, \ldots, u_n\rangle)\, V$. We write $\mathrm{D}_\alpha\, \nu \cdot u_{n+1} = \sum_{q=1}^{m} a_q \nu_q$ where, for all q, $a_q \neq 0$ and $\nu_q \in \Delta^{\square}$. It is then sufficient to show that each $(\mathrm{D}^n\, \mu\alpha\, \nu_q \cdot \langle u_1, \ldots, u_n\rangle)\, V \in \mathcal{S}$. By induction hypothesis, we are led to prove that, for all $I \subseteq [1, n]$, every $\mu\alpha\, \left(\mathrm{D}^k_\alpha\, \nu_q \cdot u_I\right) \in \mathsf{R}\langle\mathcal{S}\rangle$, and every $\mu\alpha\, (\mathrm{D}^n_\alpha\, \nu_q \cdot (u_1, \ldots, u_n))_\alpha\, V \in \mathsf{R}\langle\mathcal{S}\rangle$. Since $\mathrm{D}_\alpha\, \nu \cdot u_{n+1} = \sum_{q=1}^{m} a_q \nu_q$, we have

$$\begin{aligned} \sum_{q=1}^{m} a_q \mu\alpha \left(\mathrm{D}^k_\alpha\, \nu_q \cdot u_I\right) &= \mu\alpha \left(\mathrm{D}^k_\alpha\, (\mathrm{D}_\alpha\, \nu \cdot u_{n+1}) \cdot u_I\right) \\ &= \mu\alpha \left(\mathrm{D}^{k+1}_\alpha\, \nu \cdot u_{I \cup \{n+1\}}\right) \in \mathsf{R}\langle\mathcal{S}\rangle \end{aligned}$$

by hypothesis; since all $a_q \neq 0$, lemma 5.2 implies $\mu\alpha \left(\mathrm{D}^k_\alpha\, \nu_q \cdot u_I\right) \in \mathsf{R}\langle\mathcal{S}\rangle$ for all q. Similarly,

$$\begin{aligned} \sum_{q=1}^{m} a_q \mu\alpha\, (\mathrm{D}^n_\alpha\, \nu_q \cdot (u_1, \ldots, u_n))_\alpha\, V &= \mu\alpha\, (\mathrm{D}^n_\alpha\, (\mathrm{D}_\alpha\, \nu \cdot u_{n+1}) \cdot (u_1, \ldots, u_n))_\alpha\, V \\ &= \mu\alpha \left(\mathrm{D}^{n+1}_\alpha\, \nu \cdot (u_1, \ldots, u_{n+1})\right)_\alpha V \in \mathsf{R}\langle\mathcal{S}\rangle \end{aligned}$$

by hypothesis; again, since all $a_q \neq 0$, lemma 5.2 implies $\mu\alpha\, (\mathrm{D}^n_\alpha\, \nu_q \cdot (u_1, \ldots, u_n))_\alpha\, V \in \mathsf{R}\langle\mathcal{S}\rangle$ for all q. $\square$

Corollary 5.17. *Let $\mathcal{S}$ be a saturated subset of $\mathcal{N}$, $\nu \in \mathcal{N}^{\square}$ and $\pi = e_1 \ldots e_m$ be an $\mathcal{N}$-stack. Write $e_j = (u^{(j)}_1, \ldots, u^{(j)}_{n_j}; V_j)$. Then $(\mu\alpha\, \nu)\, \pi \in \mathcal{S}$ as soon as*

- *for all $j \in [1, m]$, for all $I \subseteq [1, n_j]$, $\mu\alpha \left(\mathrm{D}^k_\alpha\, (\nu)_\alpha\, e_1 \ldots e_{j-1} \cdot u^{(j)}_I\right) \in \mathsf{R}\langle\mathcal{S}\rangle$ (with $k = |I|$);*
- *$\mu\alpha\, (\nu)_\alpha\, \pi \in \mathsf{R}\langle\mathcal{S}\rangle$.*

Lemma 5.18. *The set $\mathcal{N}$ is saturated.*

Proof. $\mathcal{N}$ is trivially closed under renaming (see also lemma 3.11). The main condition for saturation is:

For all $\mathcal{N}$-redex t and all $\mathcal{N}$-stack π, if for all $T' \in \mathsf{Red}\,(t)$, $(T')\, \pi \in \mathsf{R}\langle\mathcal{N}\rangle$, then $(t)\, \pi \in \mathcal{N}$.

We prove it only for $\mathcal{N}$-redexes of shape $(\mu\alpha\, \nu)\, e_0$ where e_0 is a $\mathcal{N}$-elimination and $\nu \in \mathcal{N}^{\square}$; the proof for the other kind of redexes is exactly the same. We write $\pi = e_1 \ldots e_m$ where the e_i are $\mathcal{N}$-eliminations. For $i \in [0, m]$ we write

$$e_i = (u^{(i)}_1, \ldots, u^{(i)}_{p_i}; V_i).$$

We define $|e_i| = \sum_{j=1}^{p_i} \left|u^{(i)}_j\right| + |V_i|$.

With these notations, we prove by induction on $\sum_{i=0}^{m} |e_i| + |\nu|$, that:

For all $\nu \in \mathcal{N}^{\square}$ and all $\mathcal{N}$-stack $e_0 \ldots e_m$, if for all $T' \in \mathsf{Red}\,((\mu\alpha\, \nu)\, e_0)$, $(T')\, e_1 \ldots e_m \in \mathsf{R}\langle\mathcal{N}\rangle$, then $s = (t)\, e_0 \ldots e_m \in \mathcal{N}$, i.e. *for all S' such that $s\ \rho\ S'$, $S' \in \mathsf{R}\langle\mathcal{N}\rangle$.*

The reduction $s\ \rho\ S'$ can occur at several places:

- at the root of the $\mathcal{N}$-redex;

- inside ν;
- inside one of the $u_j^{(i)}$;
- inside one of the V_i.

Head reduction. In the first case, which is the only possible one if $\sum_{i=0}^m |e_i| + |\nu| = 0$, $S' = (T')\,\pi$ with $T' \in \mathsf{Red}\,((\mu\alpha\,\nu)\,e_0)$, so the hypothesis applies directly.

Reduction in linear position. Consider the case in which the reduction occurs inside ν — it would be the same in the case of some $u_j^{(i)}$. So $S' = (\mu\alpha\,N')\,e_0 :: \pi$ with $\nu\ \rho\ N'$. Write $N' = \sum_{l=1}^q a_l\nu_l$ and, for all $l \in [1,q]$, define $s'_l = (\mu\alpha\,\nu_l)\,e_0 :: \pi$ so that $S' = \sum_{l=1}^q a_l s'_l$. By lemma 5.8, it is then sufficient to prove that $s'_l \in \mathcal{N}$, for each $l \in [1,q]$. For all l, $|\nu_l| < |\nu|$ and induction hypothesis applies to the data $\nu_l, e_0, \ldots, e_m$. Hence it is sufficient to show that for all $U' \in \mathsf{Red}\,((\mu\alpha\,\nu_l)\,e_0)$, $(U')\,\pi \in \mathsf{R}\langle\mathcal{N}\rangle$. Recall that $e_0 = (u_1^{(0)}, \ldots, u_{p_0}^{(0)}; V_0)$.

- If $p_0 = 0$ then $U' = \mu\alpha\,(\nu_l)_\alpha\,V_0$. Let $T' = \mu\alpha\,(\nu)_\alpha\,V_0$: $T' \in \mathsf{Red}\,((\mu\alpha\,\nu)\,e_0)$ thus, by hypothesis $(T')\,\pi \in \mathsf{R}\langle\mathcal{N}\rangle$. By lemma 3.21 and contextuality of ρ^*, $(T')\,\pi\ \rho^*\ \sum_{l=1}^q a_l\,\big(\mu\alpha\,(\nu_l)_\alpha\,V_0\big)\,\pi$. Hence each $\big(\mu\alpha\,(\nu_l)_\alpha\,V_0\big)\,\pi \in \mathsf{R}\langle\mathcal{N}\rangle$ (lemma 5.2).
- If $p_0 > 0$ the proof is exactly the same, with another reduction.

Reduction in non-linear position. Consider the case in which the reduction occurs inside a V_i. So $S' = (\mu\alpha\,\nu)\,e_0 \ldots e'_i \ldots e_m$ with $e'_i = (u_1^{(i)}, \ldots, u_{p_i}^{(i)}; V'_i)$ and $V_i\ \widetilde{\rho}\ V'_i$. $|V'_i| < |V_i|$ and the induction hypothesis applies to the data $\nu, e_0, \ldots, e_{i-1}, e'_i, e_{i+1}, \ldots, e_m$. It is sufficient to show that for all $U' \in \mathsf{Red}\,((\mu\alpha\,\nu)\,e_0)$, $(U')\,e_1 \ldots e'_i \ldots e_m \in \mathsf{R}\langle\mathcal{N}\rangle$ — or something similar if $i = 0$, also using lemma 3.21 in this case. The end of the proof is the same as before. $\square$

5.3 Reducibility

Definition 5.19. If $\mathcal{X}$ and $\mathcal{Y}$ are sets of simple terms, one defines $\mathcal{X} \to \mathcal{Y} \subseteq \Delta$ by:

$$\mathcal{X} \to \mathcal{Y} = \{s \in \Delta;\ \text{for all } \mathcal{X}\text{-elimination } e, (s)\,e \in \mathcal{Y}\}\,.$$

More generally, if $\mathcal{P}$ is a set of stacks, one defines $\mathcal{P} \to \mathcal{Y} \subseteq \Delta$ by:

$$\mathcal{P} \to \mathcal{Y} = \{s \in \Delta;\ \forall \pi \in \mathcal{P},\ (s)\,\pi \in \mathcal{Y}\}\,.$$

Proposition 5.20. *The following two properties hold:*

(i) If $t \in \mathcal{X} \to \mathcal{Y}$ and $u \in \mathcal{X}$, then $\mathrm{D}\,t \cdot u \in \mathcal{X} \to \mathcal{Y}$.

(ii) If $\mathcal{P} \subseteq \mathcal{P}'$ are sets of stacks and $\mathcal{Y}' \subseteq \mathcal{Y} \subseteq \Delta$ then $\mathcal{P}' \to \mathcal{Y}' \subseteq \mathcal{P} \to \mathcal{Y}$.

Lemma 5.21. *If $\mathcal{S}$ is a saturated set and $\mathcal{P}$ is a set of $\mathcal{N}$-stacks and is stable under renaming, then $\mathcal{P} \to \mathcal{S}$ is saturated.*

Proof. $\mathcal{P} \to \mathcal{S}$ is clearly closed under renaming. We have to show the following: for any $\mathcal{N}$-redex t and any $\mathcal{N}$-stack π, if for all $T' \in \mathsf{Red}\,(t)$, $(T')\,\pi \in \mathsf{R}\langle\mathcal{P} \to \mathcal{S}\rangle$, then $(t)\,\pi \in \mathcal{P} \to \mathcal{S}$. By definition of $\mathcal{P} \to \mathcal{S}$, it amounts to prove that for all $\theta \in \mathcal{P}$, $(t)\,\pi\theta \in \mathcal{S}$. Since π and θ are $\mathcal{N}$-stacks, $\pi\theta$ is an $\mathcal{N}$-stack too; thus by saturation of $\mathcal{S}$, it is sufficient to prove that $\forall T' \in \mathsf{Red}\,(t)$, $(T')\,\pi\theta \in \mathsf{R}\langle\mathcal{S}\rangle$. By hypothesis, $(T')\,\pi \in \mathsf{R}\langle\mathcal{P} \to \mathcal{S}\rangle$, which ends the proof, using the definition of $\mathcal{P} \to \mathcal{S}$ and the fact that $\theta \in \mathcal{P}$. $\square$

Definition 5.22. We define the interpretation A^* of type A by induction on A:

- $\phi^* = \mathcal{N}$ if ϕ is a basic type;
- $(A \to B)^* = A^* \to B^*$.

Let $\mathcal{N}_0$ be the set of all simple terms of shape $(x)\,\pi$, where π is an $\mathcal{N}$-stack.

Lemma 5.23. *The following inclusions hold:*

$$\mathcal{N}_0 \subseteq (\mathcal{N} \to \mathcal{N}_0) \subseteq (\mathcal{N}_0 \to \mathcal{N}) \subseteq \mathcal{N}.$$

Proof. Of course, $\mathcal{N}_0 \subseteq \mathcal{N}$, hence the central inclusion, by proposition 5.20. The first inclusion holds by definition of $\mathcal{N}_0$. If $t \in \mathcal{N}_0 \to \mathcal{N}$, let x be any variable, $x \in \mathcal{N}_0$ and we have $(t)\, x \in \mathcal{N}$, which clearly implies $t \in \mathcal{N}$ by contextuality of $\widetilde{\rho}$; hence the last inclusion. $\square$

Corollary 5.24. *For all type A, $\mathcal{N}_0 \subseteq A^* \subseteq \mathcal{N}$.*

Definition 5.25 (Predual)**.** For all type A we define a set of $\mathcal{N}$-stacks $A^\perp$, by induction on A.

$$\begin{array}{rcl} \phi^\perp & = & \{\varepsilon\} \text{ if } \phi \text{ is a basic type;} \\ (A \to B)^\perp & = & \{\varepsilon\} \cup \{e :: \pi;\ e \text{ is an } A^*\text{-elimination and } \pi \in B^\perp\}. \end{array}$$

Since all A^* are saturated, the following property holds trivially.

Proposition 5.26. *For all type A, $A^\perp$ is stable under renaming.*

Lemma 5.27. *$A^\perp$ is closed under prefixes: if $\pi\theta \in A^\perp$ then $\pi \in A^\perp$.*

Proof. The proof is by induction on A. If $A = \phi$ is a basic type, $A^\perp = \{\varepsilon\}$ and the result is clear. Suppose $B^\perp$ is closed under prefix and let $\pi\theta \in (A \to B)^\perp$; then either $\pi = \varepsilon \in (A \to B)^\perp$, or $\pi = e :: \pi'$ with e an A^*-elimination and $\pi'\theta \in B^\perp$. Since $B^\perp$ is closed under prefix, $\pi' \in B^\perp$ and $\pi = e :: \pi' \in (A \to B)^\perp$. $\square$

Lemma 5.28. *$A^* = A^\perp \to \mathcal{N}$.*

Proof. The proof is by induction on A. If ϕ is a basic type, $\phi^* = \mathcal{N} = \{\varepsilon\} \to \mathcal{N}$. Suppose $B^* = B^\perp \to \mathcal{N}$; then

$$\begin{array}{rcl} (A \to B)^* & = & A^* \to B^* \\ & = & \{s \in \Delta;\ \text{for all } A^*\text{-elimination } e,\ (s)\, e \in B^*\} \\ & = & \left\{s \in \Delta;\ \text{for all } A^*\text{-elimination } e, \text{ for all } \pi \in B^\perp, ((s)\, e)\, \pi \in \mathcal{N}\right\} \\ & = & \mathcal{E} \to \mathcal{N} \end{array}$$

where $\mathcal{E} = \left\{e :: \pi;\ e \text{ is an } A^*\text{-elimination and } \pi \in B^\perp\right\}$. Since $(A \to B)^* \subseteq \mathcal{N}$,

$$\mathcal{E} \to \mathcal{N} = (\mathcal{E} \to \mathcal{N}) \cap \mathcal{N} = (\{\varepsilon\} \cup \mathcal{E}) \to \mathcal{N} = (A \to B)^\perp \to \mathcal{N}$$

$\square$

Definition 5.29. We define a reflexive and transitive binary relation $\preceq$ on eliminations by

$$(u_1, \ldots, u_m; S) \preceq (v_1, \ldots, v_n; T) \text{ if } S = T \text{ and } \langle u_1, \ldots, u_m\rangle \subseteq \langle v_1, \ldots, v_n\rangle.^2$$

We extend it to stacks by:

$$e_1 \ldots e_p \preceq f_1 \ldots f_q \text{ if } p = q \text{ and } e_j \preceq f_j \text{ for } j = 1, \ldots, p.$$

Proposition 5.30. *The following two properties are trivially derived from the previous definition:*

(i) If $\mathcal{X}$ is a set of simple terms and π is an $\mathcal{X}$-stack, then every stack θ such that $\theta \preceq \pi$ is an $\mathcal{X}$-stack of the same length.

(ii) For all type A, $A^\perp$ is downwards closed under $\preceq$, i.e. *if $\pi \in A^\perp$ and θ is a stack such that $\theta \preceq \pi$ then $\theta \in A^\perp$.*

[2] Here $\subseteq$ denotes multiset inclusion as mentioned in section 2.1.

5.4 Adequation

Theorem 5.31. *Let S be a term and assume*

$$x_1 : A_1, \ldots, x_m : A_m \vdash S : A \mid \alpha_1 : B_1, \ldots, \alpha_n : B_n$$

is derivable. Let

- *e_1 be an A_1^*-elimination, ..., e_m be an A_m^*-elimination;*
- *$\pi_1 \in B_1^\perp$, ..., $\pi_n \in B_n^\perp$.*

Assume $x_1, \ldots, x_m$ and $\alpha_1, \ldots, \alpha_n$ are not free in any of these eliminations and stacks. Then

$$(S\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha_1,\ldots,\alpha_n}(\pi_1, \ldots, \pi_n) \in \mathsf{R}\langle A^* \rangle.$$

Proof. The proof is by induction on the type derivation. The application and derivation cases hold by definition of the interpretation of arrow types. Lemma 5.16 is crucial in the λ-abstraction case. The case of μ-abstraction involves lemma 5.28, which enables the use of corollary 5.17. The naming case holds by definition of $A^\perp$. A full proof of this result is given in appendix A. □

We get the following theorem as a corollary of theorem 5.31.

Theorem 5.32. *All typable terms of the differential $\lambda\mu$-calculus are strongly normalizing.*

Proof. Let $S \in \mathsf{R}\langle\Delta\rangle$ such that $x_1 : A_1, \ldots, x_m : A_m \vdash S : A \mid \alpha_1 : B_1, \ldots, \alpha_n : B_n$ is derivable. Let $y_1, \ldots, y_m$ be m distinct, fresh variables. For all $i \in [1, m]$, since $\mathcal{N}_0 \subset A_i^*$, $(; y_i)$ is an A_i^*-elimination. Moreover, for all $j \in [1, n]$, $\varepsilon \in B_j^\perp$. Since variables $x_1, \ldots, x_m$ and names $\alpha_1, \ldots, \alpha_n$ are not free in these eliminations nor in ε, by theorem 5.31, $S' = S\,[y_1, \ldots, y_m/x_1, \ldots, x_m] \in \mathsf{R}\langle A^*\rangle$. Since $A^* \subseteq \mathsf{R}\langle\mathcal{N}\rangle$, we get $S' \in \mathsf{R}\langle\mathcal{N}\rangle$: this implies $S \in \mathsf{R}\langle\mathcal{N}\rangle$ by corollary 3.11. □

5.5 Weak normalization

Remember that we forced strong conditions on R in the beginning of this section: we assumed that the width function $\mathrm{w} : \mathsf{R} \longrightarrow \mathbf{N}$ was well defined and was a homomorphism of rigs (which in particular entails positivity of R). One can however get rid of this problem by slightly changing the notion of normal form and still obtain a weak normalization result.

Definition 5.33. We simultaneously define simple passive forms, passive forms, neutral forms and pre-neutral forms by the following statements:

- $s \in \Delta$ is a pre-neutral form if $s = x \in \mathfrak{V}$, or $s = (t)\,U$, where t is a neutral form and U is a passive form;
- $s \in \Delta$ is a neutral form if $s = \mathrm{D}^n\, t \cdot (u_1, \ldots, u_n)$, where t is a pre-neutral form and each u_i is a simple passive form;
- $s \in \Delta$ is a simple passive form if s is a neutral form, or $s = \lambda x\, t$ where t is a simple passive form, or $s = \mu\alpha\,[\beta]\,t$ where t is a simple passive form;
- S is a passive form if, for all $s \in \mathrm{Supp}(S)$, s is a simple passive form.

Intuitively, passive forms are those terms which do not contain a redex with a non-zero coefficient.

Proposition 5.34. *Any normal form is also a passive form. Moreover, if R is positive then the passive forms are exactly the normal forms.*

Lemma 5.35. *Let $\mathsf{P}_m = \mathbf{N}[\chi_1, \ldots, \chi_m]$ be the rig of polynomials over indeterminates $\chi_1, \ldots, \chi_m$ with non-negative integer coefficients. The width function is well defined and is a morphism of rigs from P_m to $\mathbf{N}$.*

Proof. The width of a polynomial is exactly the sum of all its coefficients:

$$\mathrm{w}\left(\sum_{p_1,\ldots,p_m \in \mathbf{N}} a_{p_1,\ldots,p_m} \chi_1^{p_1} \cdots \chi_n^{p_m}\right) = \sum_{p_1,\ldots,p_m \in \mathbf{N}} a_{p_1,\ldots,p_m}$$

which is finite. □

Recall that if R is a commutative rig, $\mathsf{R}\langle\Delta\rangle$ denotes the set of terms with coefficients in R.

Corollary 5.36. *Any typable term in $\mathsf{P}_m\langle\Delta\rangle$ is strongly normalizing.*

Let R be any commutative rig, $P \in \mathsf{P}_m$ and $a_1, \dots, a_m \in \mathsf{R}$. We denote as usual by $P(a_1, \dots, a_m)$ the evaluation of P at point $(a_1, \dots, a_m)$: $P(a_1, \dots, a_m) \in \mathsf{R}$. We extend this notation to terms as follows.

Definition 5.37. If $S \in \mathsf{P}_m\langle\Delta\rangle$, $S(a_1, \dots, a_m) \in \mathsf{R}\langle\Delta\rangle$ is the term obtained by replacing every coefficient P in S by its evaluation $P(a_1, \dots, a_m)$. [3]

Proposition 5.38. *For all $S \in \mathsf{P}_m\langle\Delta\rangle$, if S is a passive form, then $S(a_1, \dots, a_m)$ is a passive form of $\mathsf{R}\langle\Delta\rangle$.*

Lemma 5.39. *Let $a_1, \dots, a_m \in \mathsf{R}$. For all $S, S' \in \mathsf{P}_m\langle\Delta\rangle$, if $S \mathrel{\widetilde{\rho}} S'$, then*

$$S(a_1, \dots, a_m) \;\rho^*\; S'(a_1, \dots, a_m).$$

Proof. The proof is easy by induction on the height of reduction $S \mathrel{\widetilde{\rho}} S'$. □

Definition 5.40. Let $a_1, \dots, a_m$ be m distinct elements of R. Assume all the non-zero coefficient appearing in the writing of term S are elements of $\{a_1, \dots, a_m\}$. We define $\check{S} \in \mathsf{P}_m\langle\Delta\rangle$ as the term obtained from S by replacing every non-zero coefficent a_i by the monomial χ_i.

Proposition 5.41. *$S = \check{S}(a_1, \dots, a_m)$ and $\mathrm{Supp}(S) = \{s(a_1, \dots, a_m);\ s \in \mathrm{Supp}(\check{S})\}$.*

Lemma 5.42. *$\Gamma \vdash \check{S} : A \mid \Delta$ iff $\Gamma \vdash S : A \mid \Delta$.*

Proof. The proof is by straightforward induction on the typing derivation $\Gamma \vdash S : A \mid \Delta$, using proposition 5.41 in the case of a sum of simple terms. □

Theorem 5.43. *Let R be any commutative rig and $S \in \mathsf{R}\langle\Delta\rangle$ be a typable term. Then S is weakly normalizing in the sense that it reduces to a passive form.*

Proof. If S is typable then, by lemma 5.42, $\check{S}$ is typable too. By theorem 5.32, $\check{S}$ is strongly normalizing, hence $\check{S} \;\rho^*\; T$ with T a normal form. By proposition 5.34, T is a passive form, and so is $T(a_1, \dots, a_m)$ by proposition 5.38. By lemma 5.39, $S \;\rho^*\; T(a_1, \dots, a_m)$, hence the conclusion. □

Recall that if R is positive, then every passive form is a normal form; in this case theorem 5.43 states a genuine weak normalization property.

6 Concluding remarks: reduction and coefficients

It is clear that the equational collapse we described in section 3.6 is actually not related with differential reduction: it is only a consequence of the way we defined reduction on linear combinations. Let $\mathcal{X}$ be any R-module, and $\tau \subseteq \mathcal{X} \times \mathcal{X}$ be any reflexive, transitive and linear binary relation on $\mathcal{X}$. If for all $x \in \mathcal{X}$, there exists $\Sigma_x \in \mathcal{X}$ such that $\Sigma_x \;\tau\; x + \Sigma_x$, then τ is trivial as soon as $-1 \in \mathsf{R}$.

The same holds for the conditions we imposed on R for strong normalization, as well as for the weak normalization scheme we have just presented. Let $\mathsf{R} = \mathbf{Q}^+$ be the set of all non-negative rational numbers, and assume τ is reflexive, transitive and linear. Then as soon as $x, y \in \mathcal{X}$ are such that $x \;\tau\; y$, there is an infinite sequence of reductions from x:

$$x \;\tau\; \frac{x}{2} + \frac{y}{2} \;\tau\; \frac{x}{4} + \frac{3y}{4} \;\tau\; \dots$$

One may think of turning this seeming bug into a feature, following an idea by Thomas Ehrhard: assume τ is any binary relation on $\mathcal{X}$, and define τ_a by:

$$x \;\tau_a\; x' \text{ iff } x = ay + z \text{ and } x' = ay' + z \text{ with } y \;\tau\; y'.$$

Intuitively, this reduces y in quantity a inside x. If $a = 0$, nothing is reduced: τ_0 is identity. If $a + b = 0$, then τ_b reduces backwards w.r.t. τ_a: if $x \;\tau_a\; x'$, then $x' \;\tau_b\; x$. Indeed, if $x = ay + z$ and $x' = ay' + z$ with $y \;\tau\; y'$,

$$x' = ay' + z = ay' + z + ay + by \;\tau_b\; ay' + z + ay + by' = ay + z = x.$$

[3]Of course, the set of simple terms Δ is not the same when we write $\mathsf{R}\langle\Delta\rangle$ and $\mathsf{P}_m\langle\Delta\rangle$.

In this setting, we can write the above-mentioned infinite sequence of reductions as follows:

$$x \, \tau_{\frac{1}{2}} \, \frac{x}{2} + \frac{y}{2} \, \tau_{\frac{1}{4}} \, \frac{x}{4} + \frac{3y}{4} \, \tau_{\frac{1}{8}} \, \dots$$

Notice that the sum of all subscripts of reductions in this sequence never reaches 1, so that reduction is bounded in some sense.

Although we didn't manage to turn those simple remarks into any useful properties in the setting of differential $\lambda\mu$-calculus yet, such a study may be more fruitfully carried in the simpler case of pure λ-calculus extended with linear combinations, as presented in [Vau06].

Aknowedgements

I would first like to thank Thomas Ehrhard and Laurent Régnier for their support and for providing first hand material on differential λ-calculus. I also had some enlightening discussions with Simona Rochi della Rocca, about how reduction interferes with sums. Last, I am most grateful to Alexandre Miquel, who suggested to use a fixpoint operator in order to prove the collapse of reduction in section 3.6.

A Some detailed proofs

Proof of lemma 3.17

Proof. We prove by induction on k that if $S \; \overline{\pi_k} \; S'$ and $U \; \overline{\pi} \; U'$ then $S\,[U/x] \; \overline{\pi} \; S'\,[U'/x]$. If $k = 0$ then $S' = S$ and this is just the consequence of lemmas 3.4 and 3.14. Suppose the result holds for some k, then we extend it to $k+1$ by a case study on S. By linearity, it is sufficient to address the case in which S is simple and $S \; \pi_{k+1} \; S'$. Then one of the following statements applies:

- $S = y$: then $S' = y$ and $S \; \overline{\pi_0} \; S'$ and this reduces to the case $k = 0$;
- $S = \lambda y\, t$ with $y \neq x$ and $y \notin U$: then $S' = \lambda y\, T'$ with $t \; \pi_k \; T'$; hence, by induction hypothesis, $t\,[U/x] \; \overline{\pi} \; T'\,[U'/x]$ and we get

  $$S\,[U/x] = \lambda y\,(t\,[U/x]) \; \overline{\pi} \; \lambda y \left(T' \left[U'/x\right]\right) = S' \left[U'/x\right]$$

  by lemma 3.14;
- $S = \mu\alpha\, \nu$ with $\alpha \notin U$: then $S' = \mu\alpha\, N'$ with $\nu \; \pi_k \; N'$; hence, by induction hypothesis, $\nu\,[U/x] \; \overline{\pi} \; N'\,[U'/x]$ and we get

  $$S\,[U/x] = \mu\alpha\,(\nu\,[U/x]) \; \overline{\pi} \; \mu\alpha \left(N' \left[U'/x\right]\right) = S' \left[U'/x\right]$$

  by lemma 3.14;
- $N = [\alpha]\, s$: then $N' = [\alpha]\, S'$ with $s \; \pi_k \; S'$; hence, by induction hypothesis, $s\,[U/x] \; \overline{\pi} \; S'\,[U'/x]$ and we get

  $$N\,[U/x] = [\alpha]\,(s\,[U/x]) \; \overline{\pi} \; [\alpha] \left(S' \left[U'/x\right]\right) = N' \left[U'/x\right]$$

  by lemma 3.14;
- $S = (t)\, V$ and $S' = (T')\, V'$ with $t \; \pi_k \; T'$ and $V \; \overline{\pi_k} \; V'$: by induction hypothesis, $t\,[U/x] \; \overline{\pi} \; T'\,[U'/x]$ and $V\,[U/x] \; \overline{\pi} \; V'\,[U'/x]$ and we get

  $$S\,[U/x] = (t\,[U/x])\,(V\,[U/x]) \; \overline{\pi} \left(T' \left[U'/x\right]\right) \left(V' \left[U'/x\right]\right) = S' \left[U'/x\right]$$

  by lemma 3.14;
- $S = (\mathrm{D}^n\, \lambda y\, t \cdot (w_1, \dots, w_n))\, V$ and $S' = \left(\frac{\partial^n T'}{\partial y^n} \cdot (W'_1, \dots, W'_n)\right) [V'/y]$ with $t \; \pi_k \; T'$, $V \; \overline{\pi_k} \; V'$, $w_i \; \pi_k \; W'_i$ for all i, $x \neq y$ and $y \notin U$: by induction hypothesis, $t\,[U/x] \; \overline{\pi} \; T'\,[U'/x]$, $V\,[U/x] \; \overline{\pi} \; V'\,[U'/x]$

and $w_i\,[U/x] \mathrel{\overline{\pi}} W'_i\,[U'/x]$ for all i, and we get

$$\begin{array}{rcll}
S\,[U/x] & = & (\mathrm{D}^n\,\lambda y\,(t\,[U/x])\cdot(w_1\,[U/x],\dots,w_n\,[U/x]))\,(V\,[U/x]) & \\
& \overline{\pi} & \left(\frac{\partial^n T'[U'/x]}{\partial y^n}\cdot(W'_1\,[U'/x],\dots,W'_n\,[U'/x])\right)[V'\,[U'/x]/y] & \\
& & & \text{(corollary 3.15)} \\
& = & \left(\frac{\partial^n T'}{\partial y^n}\cdot(W'_1,\dots,W'_n)\right)[U'/x]\,[V'\,[U'/x]/y] & \text{(lemma 2.15)} \\
& = & \left(\frac{\partial^n T'}{\partial y^n}\cdot(W'_1,\dots,W'_n)\right)[V'/y]\,[U'/x] & \text{(lemma 2.15)} \\
& = & S'\,[U'/x]\,; &
\end{array}$$

- $S = (\mathrm{D}^n\,\mu\alpha\,\nu\cdot(w_1,\dots,w_n))\,V$ and $S' = \mu\alpha\,(\mathrm{D}^n_\alpha\,N'\cdot(W'_1,\dots,W'_n))_\alpha\,V'$ with $\nu\ \pi_k\ N'$, $V\ \overline{\pi_k}\ V'$, $w_i\ \pi_k\ W'_i$ for all i, and $\alpha\notin U$: by induction hypothesis, $\nu\,[U/x] \mathrel{\overline{\pi}} N'\,[U'/x]$ and $V\,[U/x] \mathrel{\overline{\pi}} V'\,[U'/x]$ and $w_i\,[U/x] \mathrel{\overline{\pi}} W'_i\,[U'/x]$ for all i, and we get

$$\begin{array}{rcll}
S\,[U/x] & = & (\mathrm{D}^n\,\mu\alpha\,(\nu\,[U/x])\cdot(w_1\,[U/x],\dots,w_n\,[U/x]))\,(V\,[U/x]) & \\
& \overline{\pi} & \mu\alpha\,(\mathrm{D}^n_\alpha\,N'\,[U'/x]\cdot(W'_1\,[U'/x],\dots,W'_n\,[U'/x]))_\alpha\,(V'\,[U'/x]) & \\
& & & \text{(corollary 3.15)} \\
& = & \mu\alpha\,((\mathrm{D}^n_\alpha\,N'\cdot(W'_1,\dots,W'_n))\,[U'/x])_\alpha\,(V'\,[U'/x]) & \text{(lemma 2.15)} \\
& = & \mu\alpha\,\left(((\mathrm{D}^n_\alpha\,N'\cdot(W'_1,\dots,W'_n))_\alpha\,V')\,[U'/x]\right) & \text{(lemma 2.15)} \\
& = & S'\,[U'/x]\,; &
\end{array}$$

- $S = \mathrm{D}^n\,t\cdot(v_1,\dots,v_n)$ and $S' = \mathrm{D}^n\,T'\cdot(V'_1,\dots,V'_n)$ with $t\ \pi_k\ T'$ and $v_i\ \overline{\pi_k}\ V'_i$ for all i: by induction hypothesis, $t\,[U/x] \mathrel{\overline{\pi}} T'\,[U'/x]$ and $v_i\,[U/x] \mathrel{\overline{\pi}} V'_i\,[U'/x]$ for all i, and we get

$$\begin{array}{rcll}
S\,[U/x] & = & \mathrm{D}^n\,t\,[U/x]\cdot(V_1\,[U/x],\dots,V_n\,[U/x]) & \\
& \overline{\pi} & \mathrm{D}^n\,T'\,[U'/x]\cdot(V'_1\,[U'/x],\dots,V'_n\,[U'/x]) & \text{(lemma 3.14)} \\
& = & S'\,[U'/x]\,; &
\end{array}$$

- $S = \mathrm{D}^n\,\lambda y\,t\cdot(v_1,\dots,v_n)$ and $S' = \mathrm{D}^{n-p}\,\lambda y\left(\frac{\partial^p T'}{\partial y^p}\cdot V'_I\right)\cdot V'_J$ with $t\ \pi_k\ T'$, $v_i\ \overline{\pi_k}\ V'_i$ for all i, $|I| = p$, $J = \{1,\dots,n\}\setminus I$, $x\neq y$ and $y\notin U$: by induction hypothesis, $t\,[U/x] \mathrel{\overline{\pi}} T'\,[U'/x]$ and $v_i\,[U/x] \mathrel{\overline{\pi}} V'_i\,[U'/x]$ for all i, and we get

$$\begin{array}{rcll}
S\,[U/x] & = & \mathrm{D}^n\,\lambda y\,(t\,[U/x])\cdot(V_1\,[U/x],\dots,V_n\,[U/x]) & \\
& \overline{\pi} & \mathrm{D}^{n-p}\,\lambda y\left(\frac{\partial^p T'[U'/x]}{\partial y^p}\cdot V'_I\,[U'/x]\right)\cdot(V'_J\,[U'/x]) & \text{(corollary 3.15)} \\
& = & \mathrm{D}^{n-p}\,\lambda y\left(\left(\frac{\partial^p T'}{\partial y^p}\cdot V'_I\right)[U'/x]\right)\cdot(V'_J\,[U'/x]) & \text{(lemma 2.15)} \\
& = & S'\,[U'/x]\,; &
\end{array}$$

- $S = \mathrm{D}^n\,\mu\alpha\,\nu\cdot(v_1,\dots,v_n)$ and $S' = \mathrm{D}^{n-p}\,\mu\alpha\,(\mathrm{D}^p_\alpha\,N'\cdot V'_I)\cdot V'_J$ with $\nu\ \pi_k\ N'$, $v_i\ \overline{\pi_k}\ V'_i$ for all i, $|I| = p$, $J = \{1,\dots,n\}\setminus I$ and $\alpha\notin U$: by induction hypothesis, $\nu\,[U/x] \mathrel{\overline{\pi}} N'\,[U'/x]$ and $v_i\,[U/x] \mathrel{\overline{\pi}} V'_i\,[U'/x]$ for all i, and we get

$$\begin{array}{rcll}
S\,[U/x] & = & \mathrm{D}^n\,\mu\alpha\,(\nu\,[U/x])\cdot(V_1\,[U/x],\dots,V_n\,[U/x]) & \\
& \overline{\pi} & \mathrm{D}^{n-p}\,\mu\alpha\,(\mathrm{D}^p_\alpha\,N'\,[U'/x]\cdot V'_I\,[U'/x])\cdot(V'_J\,[U'/x]) & \text{(corollary 3.15)} \\
& = & \mathrm{D}^{n-p}\,\mu\alpha\,((\mathrm{D}^p_\alpha\,N'\cdot V'_I)\,[U'/x])\cdot(V'_J\,[U'/x]) & \text{(lemma 2.15)} \\
& = & S'\,[U'/x]\,. &
\end{array}$$

□

Proof of lemma 3.18

Proof. The proof is very similar to the previous one and we only study the most interesting cases, i.e., those involving a redex.

- if $S = (\mathrm{D}^n\,\lambda y\,t \cdot (w_1,\ldots,w_n))\,V$ and $S' = \left(\frac{\partial^n T'}{\partial y^n} \cdot (W'_1,\ldots,W'_n)\right)[V'/y]$ with $t\ \pi_k\ T'$, $V\ \overline{\pi_k}\ V'$, $w_i\ \pi_k\ W'_i$ for all i, $x \neq y$ and $y \notin U$: by induction hypothesis, $\frac{\partial t}{\partial x} \cdot U\ \overline{\pi}\ \frac{\partial T'}{\partial x} \cdot U'$, $\frac{\partial V}{\partial x} \cdot U\ \overline{\pi}\ \frac{\partial V'}{\partial x} \cdot U'$ and $\frac{\partial w_i}{\partial x} \cdot U\ \overline{\pi}\ \frac{\partial W'_i}{\partial x} \cdot U'$ for all i; denoting

$$W_j^{(i)} = \begin{cases} \frac{\partial w_i}{\partial x} \cdot U \text{ if } i = j \\ w_j \text{ otherwise} \end{cases}$$

and

$$W'^{(i)}_j = \begin{cases} \frac{\partial W'_i}{\partial x} \cdot U \text{ if } i = j \\ W'_j \text{ otherwise} \end{cases} ,$$

we get that $W_j^{(i)}\ \overline{\pi}\ W'^{(i)}_j$ for all i,j, and then

$$\begin{array}{rcll}
\frac{\partial S}{\partial x} \cdot U & = & \left(\mathrm{D}^n\,\lambda y \left(\frac{\partial t}{\partial x} \cdot U\right) \cdot (w_1,\ldots,w_n)\right) V & \\
& & + \sum_{i=1}^n \left(\mathrm{D}^n\,\lambda y\,t \cdot \left(W_1^{(i)},\ldots,W_n^{(i)}\right)\right) V & \\
& & + \left(\mathrm{D}^{n+1}\,\lambda y\,t \cdot \left(w_1,\ldots,w_n, \frac{\partial V}{\partial x} \cdot U\right)\right) V & \\
& \overline{\pi} & \left(\frac{\partial^n}{\partial y^n}\left(\frac{\partial T'}{\partial x} \cdot U'\right) \cdot (W'_1,\ldots,W'_n)\right)[V'/y] & \text{(corollary 3.15)} \\
& & + \sum_{i=1}^n \left(\frac{\partial^n T'}{\partial y^n} \cdot \left(W'^{(i)}_1,\ldots,W'^{(i)}_n\right)\right)[V'/y] & \\
& & + \left(\frac{\partial^{n+1} T'}{\partial y^{n+1}} \cdot \left(W'_1,\ldots,W'_n, \frac{\partial V'}{\partial x} \cdot U'\right)\right)[V'/y] & \\
& = & \left(\frac{\partial}{\partial x}\left(\frac{\partial^n T'}{\partial y^n} \cdot (W'_1,\ldots,W'_n)\right) \cdot U'\right)[V'/y] & \text{(lemma 2.16)} \\
& & + \left(\frac{\partial}{\partial y}\left(\frac{\partial^n T'}{\partial y^n} \cdot (W'_1,\ldots,W'_n)\right) \cdot \left(\frac{\partial V'}{\partial x} \cdot U'\right)\right)[V'/y] & \\
& = & \frac{\partial}{\partial x}\left(\left(\frac{\partial^n T'}{\partial y^n} \cdot (W'_1,\ldots,W'_n)\right)[V'/y]\right) \cdot U' & \text{(lemma 2.16)} \\
& = & \frac{\partial S'}{\partial x} \cdot U'; &
\end{array}$$

- if $S = (\mathrm{D}^n\,\mu\alpha\,\nu \cdot (w_1,\ldots,w_n))\,V$ and $S' = \mu\alpha\,(\mathrm{D}^n_\alpha\,N' \cdot (W'_1,\ldots,W'_n))_\alpha\,V'$ with $\nu\ \pi_k\ N'$, $V\ \overline{\pi_k}\ V'$, $w_i\ \pi_k\ W'_i$ for all i, and $\alpha \notin U$: by induction hypothesis, $\frac{\partial \nu}{\partial x} \cdot U\ \overline{\pi}\ \frac{\partial N'}{\partial x} \cdot U'$, $\frac{\partial V}{\partial x} \cdot U\ \overline{\pi}\ \frac{\partial V'}{\partial x} \cdot U'$ and $\frac{\partial w_i}{\partial x} \cdot U\ \overline{\pi}\ \frac{\partial W'_i}{\partial x} \cdot U'$ for all i; with the same notations as above, we get

$$\begin{array}{rcll}
\frac{\partial S}{\partial x} \cdot U & = & \left(\mathrm{D}^n\,\mu\alpha \left(\frac{\partial \nu}{\partial x} \cdot U\right) \cdot (w_1,\ldots,w_n)\right) V & \\
& & + \sum_{i=1}^n \left(\mathrm{D}^n\,\mu\alpha\,\nu \cdot \left(W_1^{(i)},\ldots,W_n^{(i)}\right)\right) V & \\
& & + \left(\mathrm{D}^{n+1}\,\mu\alpha\,\nu \cdot \left(w_1,\ldots,w_n, \frac{\partial V}{\partial x} \cdot U\right)\right) V & \\
& \overline{\pi} & \mu\alpha\left(\left(\mathrm{D}^n_\alpha\left(\frac{\partial N'}{\partial x} \cdot U'\right) \cdot (W'_1,\ldots,W'_n)\right)_\alpha V'\right) & \text{(corollary 3.15)} \\
& & + \sum_{i=1}^n \mu\alpha\left(\left(\mathrm{D}^n_\alpha\,N' \cdot \left(W'^{(i)}_1,\ldots,W'^{(i)}_n\right)\right)_\alpha V'\right) & \\
& & + \mu\alpha\left(\left(\mathrm{D}^{n+1}_\alpha\,N' \cdot \left(W'_1,\ldots,W'_n, \frac{\partial V'}{\partial x} \cdot U'\right)\right)_\alpha V'\right) & \\
& = & \mu\alpha\left(\left(\frac{\partial \mathrm{D}^n_\alpha\,N' \cdot \left(W'_1,\ldots,W'_n\right)}{\partial x} \cdot U'\right)_\alpha V'\right) & \text{(lemma 2.16)} \\
& & + \mu\alpha\left(\left(\mathrm{D}_\alpha\left(\mathrm{D}^n_\alpha\,N' \cdot (W'_1,\ldots,W'_n)\right) \cdot \left(\frac{\partial V'}{\partial x} \cdot U'\right)\right)_\alpha V'\right) & \\
& = & \mu\alpha\left(\frac{\partial\left(\mathrm{D}^n_\alpha\,N' \cdot \left(W'_1,\ldots,W'_n\right)\right)_\alpha V'}{\partial x} \cdot U'\right) & \text{(lemma 2.16)} \\
& = & \frac{\partial S'}{\partial x} \cdot U'; &
\end{array}$$

- $S = \mathrm{D}^n\,\lambda y\,t \cdot (v_1,\ldots,v_n)$ and $S' = \mathrm{D}^{n-p}\,\lambda y\left(\frac{\partial^p T'}{\partial y^p} \cdot V'_I\right) \cdot V'_J$ with $t\ \pi_k\ T'$, $v_i\ \overline{\pi_k}\ V'_i$ for all i, $|I| = p$, $J = \{1,\ldots,n\} \setminus I$, $x \neq y$ and $y \notin U$: by induction hypothesis, $\frac{\partial t}{\partial x} \cdot U\ \overline{\pi}\ \frac{\partial T'}{\partial x} \cdot U'$ and $\frac{\partial v_i}{\partial x} \cdot U\ \overline{\pi}\ \frac{\partial V'_i}{\partial x} \cdot U'$

for all i; with notations similar to the previous ones, we get

$$\begin{array}{rcll} \frac{\partial S}{\partial x}\cdot U & = & \mathrm{D}^n\,\lambda y\left(\frac{\partial t}{\partial x}\cdot U\right)\cdot(v_1,\dots,v_n) & \\ & & +\sum_{i=1}^n \mathrm{D}^n\,\lambda y\,t\cdot\left(V_1^{(i)},\dots,V_n^{(i)}\right) & \\ & \overline{\pi} & \mathrm{D}^{n-p}\,\lambda y\left(\frac{\partial^p}{\partial y^p}\left(\frac{\partial T'}{\partial x}\cdot U'\right)\cdot V'_I\right)\cdot V'_J & \text{(corollary 3.15)} \\ & & +\sum_{i=1}^n \mathrm{D}^{n-p}\,\lambda y\left(\frac{\partial^p T'}{\partial y^p}\cdot {V'}_I^{(i)}\right)\cdot {V'}_J^{(i)} & \\ & = & \mathrm{D}^{n-p}\,\lambda y\left(\frac{\partial}{\partial x}\left(\frac{\partial^p T'}{\partial y^p}\cdot V'_I\right)\cdot U'\right)\cdot V'_J & \text{(lemma 2.16)} \\ & & +\sum_{i\in J} \mathrm{D}^{n-p}\,\lambda y\left(\frac{\partial^p T'}{\partial y^p}\cdot {V'}_I\right)\cdot {V'}_J^{(i)} & \\ & = & \frac{\partial S'}{\partial x}\cdot U'; & \end{array}$$

- $S=\mathrm{D}^n\,\mu\alpha\,\nu\cdot(v_1,\dots,v_n)$ and $S'=\mathrm{D}^{n-p}\,\mu\alpha\,(\mathrm{D}^p_\alpha\,N'\cdot V'_I)\cdot V'_J$ with $\nu\ \pi_k\ N'$, $v_i\ \overline{\pi_k}\ V'_i$ for all i, $|I|=p$, $J=\{1,\dots,n\}\setminus I$ and $\alpha\notin U$: by induction hypothesis, $\frac{\partial\nu}{\partial x}\cdot U\ \overline{\pi}\ \frac{\partial N'}{\partial x}\cdot U'$ and $\frac{\partial v_i}{\partial x}\cdot U\ \overline{\pi}\ \frac{\partial V'_i}{\partial x}\cdot U'$ for all i; with similar notations again, we get

$$\begin{array}{rcll} \frac{\partial S}{\partial x}\cdot U & = & \mathrm{D}^n\,\mu\alpha\left(\frac{\partial\nu}{\partial x}\cdot U\right)\cdot(v_1,\dots,v_n) & \\ & & +\sum_{i=1}^n \mathrm{D}^n\,\mu\alpha\,\nu\cdot\left(V_1^{(i)},\dots,V_n^{(i)}\right) & \\ & \overline{\pi} & \mathrm{D}^{n-p}\,\mu\alpha\left(\mathrm{D}^p_\alpha\left(\frac{\partial N'}{\partial x}\cdot U'\right)\cdot V'_I\right)\cdot V'_J & \text{(corollary 3.15)} \\ & & +\sum_{i=1}^n \mathrm{D}^{n-p}\,\mu\alpha\left(\mathrm{D}^p_\alpha\,N'\cdot {V'}_I^{(i)}\right)\cdot {V'}_J^{(i)} & \\ & = & \mathrm{D}^{n-p}\,\mu\alpha\left(\frac{\partial}{\partial x}\left(\mathrm{D}^p_\alpha\,N'\cdot V'_I\right)\cdot U'\right)\cdot V'_J & \text{(lemma 2.16)} \\ & & +\sum_{i\in J} \mathrm{D}^{n-p}\,\mu\alpha\left(\mathrm{D}^p_\alpha\,N'\cdot {V'}_I\right)\cdot {V'}_J^{(i)} & \\ & = & \frac{\partial S'}{\partial x}\cdot U'. & \end{array}$$

□

Proof of lemma 3.24

Proof. We prove by induction on k that if $s\ \pi_k\ S'$ then $S'\ \overline{\pi}\ s{\downarrow}$, and if $S\ \overline{\pi_k}\ S'$ then $S'\ \overline{\pi}\ S{\downarrow}$. If $k=0$, this is lemma 3.23. Assume the result holds until k, and $s\ \pi_{k+1}\ S'$, we prove $S'\ \overline{\pi}\ s{\downarrow}$. We study all possible cases for the reduction $s\ \pi_{k+1}\ S'$.

- $s=\lambda x\,t$ and $S'=\lambda x\,T'$ with $t\ \pi_k\ T'$. By induction hypothesis, $T'\ \overline{\pi}\ t{\downarrow}$, and, by contextuality of $\overline{\pi}$, $S'\ \overline{\pi}\ S{\downarrow}$.
- $s=\mu\alpha\,\nu$ and $S'=\mu\alpha\,N'$ with $\nu\ \pi_k\ N'$. By induction hypothesis, $N'\ \overline{\pi}\ \nu{\downarrow}$, and, by contextuality of $\overline{\pi}$, $S'\ \overline{\pi}\ s{\downarrow}$.
- $s=[\alpha]\,t$ and $S'=[\alpha]\,T'$ with $t\ \pi_k\ T'$. By induction hypothesis, $T'\ \overline{\pi}\ t{\downarrow}$, and, by contextuality of $\overline{\pi}$, $S'\ \overline{\pi}\ s{\downarrow}$.
- $s=(\mathrm{D}^n\,\lambda x\,t\cdot(v_1,\dots,v_n))\,U$ and $S'=\left(\frac{\partial^n T'}{\partial x^n}\cdot(V'_1,\dots,V'_n)\right)[U'/x]$ with $t\ \pi_k\ T'$, $U\ \overline{\pi_k}\ U'$ and $v_i\ \pi_k\ V'_i$ for all i. By induction hypothesis, $T'\ \overline{\pi}\ t{\downarrow}$, $U'\ \overline{\pi}\ U{\downarrow}$ and $V'_i\ \overline{\pi}\ v_i{\downarrow}$ for all i, and, by lemmas 3.17 and 3.18,

  $$S'=\frac{\partial^n T'}{\partial x^n}\cdot\left(V'_1,\dots,V'_n\right)\left[U'/x\right]\ \overline{\pi}\ \frac{\partial^n t{\downarrow}}{\partial x^n}\cdot(v_1{\downarrow},\dots,v_n{\downarrow})\,[U{\downarrow}/x]=s{\downarrow}$$

- $s=(\mathrm{D}^n\,\mu\alpha\,\nu\cdot(v_1,\dots,v_n))\,U$ and $S'=\mu\alpha\,(\mathrm{D}^n_\alpha\,N'\cdot(V'_1,\dots,V'_n))_\alpha\,U'$ with $\nu\ \pi_k\ N'$, $U\ \overline{\pi_k}\ U'$ and $v_i\ \pi_k\ V'_i$ for all i. By induction hypothesis, $N'\ \overline{\pi}\ \nu{\downarrow}$ and $U'\ \overline{\pi}\ U{\downarrow}$ and $V'_i\ \overline{\pi}\ v_i{\downarrow}$ for all i, and, by lemmas 3.19 and 3.20,

  $$\left(\mathrm{D}^n_\alpha\,N'\cdot\left(V'_1,\dots,V'_n\right)\right)_\alpha U'\ \overline{\pi}\ (\mathrm{D}^n_\alpha\,\nu{\downarrow}\cdot v_1{\downarrow},\dots,v_n{\downarrow})_\alpha\,U{\downarrow}.$$

  By contextuality of $\overline{\pi}$, $S'\ \overline{\pi}\ s{\downarrow}$.

- $s = (t)\,U$ and $S' = (T')\,U'$ with $t\ \pi_k\ T'$ and $U\ \overline{\pi_k}\ U'$. By induction hypothesis, $T'\ \overline{\pi}\ t\!\downarrow$ and $U'\ \overline{\pi}\ U\!\downarrow$.
  - If t has shape $t = \mathrm{D}^n\, x \cdot (v_1, \dots, v_n)$ or $t = \mathrm{D}^n\,(w)\,W \cdot (v_1, \dots, v_n)$, then $s\!\downarrow = (t\!\downarrow)\,U\!\downarrow$ and we conclude by contextuality.
  - If $t = \mathrm{D}^n\,\lambda x\, w \cdot (v_1, \dots, v_n)$, then

    $$T' = \mathrm{D}^{n-p}\,\lambda x \left(\frac{\partial^p W'}{\partial x^p} \cdot V'_I\right) \cdot V'_J$$

    with $w\ \pi_{k-1}\ W'$ and $v_i\ \pi_{k-1}\ V'_i$ for all i (we set $k-1 = 0$ if $k = 0$), and

    $$t\!\downarrow = \lambda x \left(\frac{\partial^n w\!\downarrow}{\partial x^n} \cdot (v_1\!\downarrow, \dots, v_n\!\downarrow)\right).$$

    By induction hypothesis (at height $k-1$) $W'\ \overline{\pi}\ v\!\downarrow$ and $V'_i\ \overline{\pi}\ v_i\!\downarrow$ for all i. Hence

    $$S' = \left(\mathrm{D}^{n-p}\,\lambda x \left(\frac{\partial^p W'}{\partial x^p} \cdot V'_I\right) \cdot V'_J\right) U'\ \overline{\pi} \left(\frac{\partial^n w\!\downarrow}{\partial x^n} \cdot (v_1\!\downarrow, \dots, v_n\!\downarrow)\right) [U\!\downarrow/x] = s\!\downarrow$$

    by lemmas 3.17 and lemmas 3.18.
  - If $t = \mathrm{D}^n\,\mu\alpha\,\nu \cdot (v_1, \dots, v_n)$, we use exactly the same technique.
- $s = \mathrm{D}^n\,\lambda x\, t \cdot (u_1, \dots, u_n)$ and $S' = \mathrm{D}^{n-p}\left(\lambda x\, \frac{\partial^p T'}{\partial x^p} \cdot U'_I\right) \cdot U'_J$ with $I \subseteq [1,n]$, $p = |I|$, $J = [1,n] \setminus I$, $t\ \pi_k\ T'$ and, for all $i \in [1,n]$, $u_i\ \pi_k\ U'_i$. By induction hypothesis, $T'\ \overline{\pi}\ t\!\downarrow$ and, for all $i \in [1,n]$, $U'_i\ \overline{\pi}\ u_i\!\downarrow$. By iteration of lemma 3.18, $\frac{\partial^p T'}{\partial x^p} \cdot U'_I\ \overline{\pi}\ \frac{\partial^p t\downarrow}{\partial x^p} \cdot u_I\!\downarrow$. Hence

  $$S' = \mathrm{D}^{n-p}\left(\lambda x\, \frac{\partial^p T'}{\partial x^p} \cdot U'_I\right) \cdot U'_J\ \overline{\pi}\ \lambda x \left(\frac{\partial^{n-p}}{\partial x^{n-p}} \left(\frac{\partial^p t\!\downarrow}{\partial x^p} \cdot u_I\!\downarrow\right) \cdot u_J\!\downarrow\right) = s\!\downarrow.$$

- $s = \mathrm{D}^n\,\mu\alpha\,\nu \cdot (u_1, \dots, u_n)$ and $S' = \mathrm{D}^{n-p}\,(\mu\alpha\,\mathrm{D}^p_\alpha\, N' \cdot U'_I) \cdot U'_J$ with $I \subseteq [1,n]$, $p = |I|$, $J = [1,n] \setminus I$, $\nu\ \pi_k\ N'$ and, for all $i \in [1,n]$, $u_i\ \pi_k\ U'_i$. By induction hypothesis, $N'\ \overline{\pi}\ \nu\!\downarrow$ and, for all $i \in [1,n]$, $U'_i\ \overline{\pi}\ u_i\!\downarrow$. By iteration of lemma 3.20, $\mathrm{D}^p_\alpha\, N' \cdot U'_I\ \overline{\pi}\ \mathrm{D}^p_\alpha\, \nu\!\downarrow \cdot u_I\!\downarrow$. Hence

  $$S' = \mathrm{D}^{n-p}\left(\mu\alpha\,\mathrm{D}^p_\alpha\, N' \cdot U'_I\right) \cdot U'_J\ \overline{\pi}\ \mu\alpha\left(\mathrm{D}^{n-p}_\alpha\,(\mathrm{D}^p_\alpha\, t\!\downarrow \cdot u_I\!\downarrow) \cdot u_J\!\downarrow\right) = s\!\downarrow.$$

- $s = \mathrm{D}^n\, t \cdot (u_1, \dots, u_n)$ and $S' = \mathrm{D}^n\, T' \cdot (U'_1, \dots, U'_n)$, where t is a variable or an application (otherwise, this is a subcase of one of the previous two cases, with $p = 0$), $t\ \pi_k\ T'$ and, for all $i \in [1,n]$, $u_i\ \pi_k\ U'_i$. By induction hypothesis, $T'\ \overline{\pi}\ T\!\downarrow$ and, for all $i \in [1,n]$, $U'_i\ \overline{\pi}\ u_i\!\downarrow$. Then the result holds by contextuality.

Now assume $S\ \overline{\pi_{k+1}}\ S'$: $S = \sum_{i=1}^n a_i s_i$ and $S' = \sum_{i=1}^n a_i S'_i$ where, for all $i \in [1,n]$, $s_i\ \pi_{k+1}\ S'_i$. We have just proven that this implies $S'_i\ \overline{\pi}\ s_i\!\downarrow$, for all i. By linearity of $\overline{\pi}$, we get $S' = \sum_{i=1}^n a_i S'_i\ \overline{\pi}\ \sum_{i=1}^n a_i s_i\!\downarrow = S\!\downarrow$. ☐

Proof of theorem 5.31

Proof. We prove the theorem by induction on the type derivation, along with the auxiliary property that if ν is a named simple term and

$$x_1 : A_1, \dots, x_m : A_m \vdash \nu \mid \alpha_1 : B_1, \dots, \alpha_n : B_n$$

then, assuming similar hypotheses,

$$(\nu\,[e_1, \dots, e_m/x_1, \dots, x_m])_{\alpha_1, \dots, \alpha_n}\,(\pi_1, \dots, \pi_n) \in \mathsf{R}\langle \mathcal{N}^\square \rangle.$$

For $i = 1, \dots, m$ we write

$$e_i = (u^{(i)}_1, \dots, u^{(i)}_{p_i}; T_i).$$

For $j = 1, \dots, n$ we write

$$\pi_j = f^{(j)}_1 \dots f^{(j)}_{r_j}$$

and for $k = 1, \dots, r_j$, we write

$$f^{(j)}_k = (w^{(j,k)}_1, \dots, w^{(j,k)}_{q_{j,k}}; V^{(j)}_k).$$

- *Variable:* $S = x_{i_0}$ for some i_0 and $A = A_{i_0}$. Write
$$S' = \left(x_{i_0}\left[e_1, \ldots, e_m / x_1, \ldots, x_m\right]\right)_{\alpha_1, \ldots, \alpha_n}(\pi_1, \ldots, \pi_n).$$
Assume some $q_{j,k}$ is non-zero: we then have
$$\mathrm{D}_{\alpha_j}\, x_{i_0} \cdot w_1^{(j,k)} = 0.$$
By lemma 2.18 and linearity of operations, we get $S' = 0$ and we conclude since $0 \in \mathsf{R}\langle A^*_{i_0}\rangle$. Now assume every $q_{j,k} = 0$: by lemma 2.13, we get
$$S' = x_{i_0}\left[e_1, \ldots, e_m / x_1, \ldots, x_m\right].$$
Assume there is $i \neq i_0$ such that $p_i \neq 0$: we then have
$$\frac{\partial x_{i_0}}{\partial x_i} \cdot u_1^{(i)} = 0.$$
By lemma 2.16 and linearity of operations, we get $S' = 0$ and we conclude. Now assume $i \neq i_0$ implies $p_i = 0$: by lemma 2.11, we get
$$S' = x_{i_0}\left[e_{i_0} / x_{i_0}\right].$$
If $p_{i_0} = 0$ then $S' = T_{i_0} \in \mathsf{R}\langle A^*_{i_0}\rangle$ by hypothesis. If $p_{i_0} = 1$ then $S' = u_1^{(i_0)} \in A^*_{i_0}$ by hypothesis. If $p_{i_0} \geq 2$ then $S' = 0$.
- *Application:* $S = (s)\, T$ with $x_1 : A_1, \ldots, x_m : A_m \vdash s : B \to A \mid \alpha_1 : B_1, \ldots, \alpha_n : B_n$ and $x_1 : A_1, \ldots, x_m : A_m \vdash T : B \mid \alpha_1 : B_1, \ldots, \alpha_n : B_n$. By lemmas 2.25 and 2.26, the term
$$\left(((s)\, T)\left[e_1, \ldots, e_m / x_1, \ldots, x_m\right]\right)_{\alpha_1, \ldots, \alpha_n}(\pi_1, \ldots, \pi_n)$$
is a sum of terms of the shape
$$\left(\mathrm{D}^h\, S' \cdot \left(T'_1, \ldots, T'_h\right)\right)\left(T\left[T_1, \ldots, T_m / x_1, \ldots, x_m\right]\right)_{\alpha_1, \ldots, \alpha_n}(\tilde{\pi}_1, \ldots, \tilde{\pi}_n)$$
where
  - $S' = \left(s\left[e'_1, \ldots, e'_m / x_1, \ldots, x_m\right]\right)_{\alpha_1, \ldots, \alpha_n}(\pi'_1, \ldots, \pi'_n)$, with $e'_i \preceq e_i$ for all i and $\pi'_j \preceq \pi_j$ for all j;
  - for all $k \in [1, h]$, T'_k is of the shape
  $$\left(T\left[e''_1, \ldots, e''_m / x_1, \ldots, x_m\right]\right)_{\alpha_1, \ldots, \alpha_n}\left(\pi''_1, \ldots, \pi''_n\right),$$
  with $e''_i \preceq e_i$ for all i and $\pi''_j \preceq \pi_j$ for all j;
  - for all $j \in [1, n]$, $\tilde{\pi}_j = V_1^{(j)} \ldots V_{r_j}^{(j)}$.

  By inductive hypothesis and proposition 5.30, we know that $S' \in \mathsf{R}\langle B^* \to A^*\rangle$, each $T'_k \in \mathsf{R}\langle B^*\rangle$ and
$$\left(T\left[T_1, \ldots, T_m / x_1, \ldots, x_m\right]\right)_{\alpha_1, \ldots, \alpha_n}(\tilde{\pi}_1, \ldots, \tilde{\pi}_n) \in \mathsf{R}\langle B^*\rangle.$$
Hence we conclude by definition of $B^* \to A^*$.
- *Derivative:* $A = B \to C$ and $S = \mathrm{D}^h\, s \cdot (t_1, \ldots, t_h)$ with
$$x_1 : A_1, \ldots, x_m : A_m \vdash s : B \to C \mid \alpha_1 : B_1, \ldots, \alpha_n : B_n$$
and
$$x_1 : A_1, \ldots, x_m : A_m \vdash t_k : B \mid \alpha_1 : B_1, \ldots, \alpha_n : B_n$$
for all $k \in [1, h]$. By lemmas 2.23 and 2.24, the term
$$\left(\left(\mathrm{D}^h\, s \cdot (t_1, \ldots, t_h)\right)\left[e_1, \ldots, e_m / x_1, \ldots, x_m\right]\right)_{\alpha_1, \ldots, \alpha_n}(\pi_1, \ldots, \pi_n)$$
is a sum of terms of the shape
$$\mathrm{D}^h\, S' \cdot \left(T'_1, \ldots, T'_h\right)$$
where

– $S' = (s\,[e'_1, \ldots, e'_m/x_1, \ldots, x_m])_{\alpha_1,\ldots,\alpha_n}\,(\pi'_1, \ldots, \pi'_n)$, with $e'_i \preceq e_i$ for all i and $\pi'_j \preceq \pi_j$ for all j;
– for all $k \in [1, h]$, T'_k is of the shape

$$\left(T\left[e''_1, \ldots, e''_m/x_1, \ldots, x_m\right]\right)_{\alpha_1,\ldots,\alpha_n}\left(\pi''_1, \ldots, \pi''_n\right),$$

with $e''_i \preceq e_i$ for all i and $\pi''_j \preceq \pi_j$ for all j.

By inductive hypothesis and proposition 5.30, $S' \in \mathsf{R}\langle B^* \to C^*\rangle$ and each $T'_k \in \mathsf{R}\langle B^*\rangle$; hence the conclusion by lemma 5.20.

- λ-*abstraction:* $A = B \to C$ and $S = \lambda x\, s$ with

$$x_1 : A_1, \ldots, x_m : A_m, x : B \vdash s : C \mid \alpha_1 : B_1, \ldots, \alpha_n : B_n.$$

We assume x is distinct from every x_i and does not occur free in any e_i nor π_j; then

$$((\lambda x\, s)\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha_1,\ldots,\alpha_n}\,(\pi_1, \ldots, \pi_n) = \lambda x\, S'$$

with

$$S' = \ (s\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha_1,\ldots,\alpha_n}\,(\pi_1, \ldots, \pi_n).$$

We show that $\lambda x\, S' \in \mathsf{R}\langle(B \to C)^*\rangle$ using the definition of $B^* \to C^*$: let $e = (u_1, \ldots, u_p; T)$ be a B^*-elimination, we have to prove $(\lambda x\, S')\, e \in \mathsf{R}\langle C^*\rangle$. For this purpose, we can apply lemma 5.16 since:

– C^* is a saturated subset of $\mathcal{N}$;
– e is an $\mathcal{N}$-elimination;
– $S' \in \mathsf{R}\langle\mathcal{N}\rangle$: take z any fresh variable, $z \in \mathcal{N}_0 \subseteq B^*$ so that

$$S'\,[z/x] = (s\,[e_1, \ldots, e_m, z/x_1, \ldots, x_m, x])_{\alpha_1,\ldots,\alpha_n}\,(\pi_1, \ldots, \pi_n) \in \mathsf{R}\langle C^*\rangle$$

by induction hypothesis; since $C^* \subseteq \mathcal{N}$ and $\mathcal{N}$ is closed under renaming, $S' \in \mathsf{R}\langle\mathcal{N}\rangle$.

Hence it remains to show that

– for all $I \subseteq [1, p]$, $\frac{\partial^k S'}{\partial x^k} \cdot u_I \in \mathsf{R}\langle C^*\rangle$ (with $k = |I|$);
– $S'\,[e/x] \in \mathsf{R}\langle C^*\rangle$.

Let $z_1, \ldots, z_m, z, \beta_1, \ldots, \beta_n$ be fresh variables and names, in particular distinct from $x_1, \ldots, x_m, x$ and $\alpha_1, \ldots, \alpha_n$ respectively, and not free in s nor e nor any e_i nor any π_j. If W is a term (or stack), we write $\widehat{W}$ for $W\,[z_1, \ldots, z_m, z, \beta_1, \ldots, \beta_n/x_1, \ldots, x_m, x, \alpha_1, \ldots, \alpha_n]$. Assume $I \subseteq [1, p]$ and $k = |I|$; we prove $\frac{\partial^k S'}{\partial x^k} \cdot u_I \in \mathsf{R}\langle C^*\rangle$. Since B^* is stable under renaming, and $z \in \mathcal{N}_0 \subseteq B^*$, $e' = (\widehat{u}_I; z)$ is a B^*-elimination. Moreover variables $x_1, \ldots, x_m, x$ and names $\alpha_1, \ldots, \alpha_n$ don't occur free in e'. Hence, by induction hypothesis,

$$S'\left[e'/x\right] = \left(s\left[e_1, \ldots, e_m, e'/x_1, \ldots, x_m, x\right]\right)_{\alpha_1,\ldots,\alpha_n}(\pi_1, \ldots, \pi_n) \in \mathsf{R}\langle C^*\rangle.$$

But $\frac{\partial^k S'}{\partial x^k} \cdot u_I = S'\,[e'/x]\,[x_1, \ldots, x_n, x, \alpha_1, \ldots, \alpha_n/z_1, \ldots, z_m, z, \beta_1, \ldots, \beta_n]$ and since C^* is closed under renaming, $\frac{\partial^k S'}{\partial x^k} \cdot u_I \in \mathsf{R}\langle C^*\rangle$. Now let's prove $S'\,[e/x] \in \mathsf{R}\langle C^*\rangle$. Since B^* is stable under renaming, $\widehat{e}$ is a B^*-elimination. Moreover, variables $x_1, \ldots, x_m, x$ and names $\alpha_1, \ldots, \alpha_n$ don't occur free in $\widehat{e}$. Hence, by induction hypothesis,

$$S'\,[\widehat{e}/x] = (s\,[e_1, \ldots, e_m, \widehat{e}/x_1, \ldots, x_m, x])_{\alpha_1,\ldots,\alpha_n}\,(\pi_1, \ldots, \pi_n) \in \mathsf{R}\langle C^*\rangle.$$

Since C^* is closed under renaming,

$$S'\,[e/x] = S'\,[\widehat{e}/x]\,[x_1, \ldots, x_m, \alpha_1, \ldots, \alpha_n/z_1, \ldots, z_m, \beta_1, \ldots, \beta_n] \in \mathsf{R}\langle C^*\rangle.$$

- *μ-abstraction:* $S = \mu\alpha\,\nu$ with

$$x_1 : A_1, \ldots, x_m : A_m \vdash \nu \mid \alpha : A, \alpha_1 : B_1, \ldots, \alpha_n : B_n.$$

We assume α is distinct from every α_j and does not occur free in any e_i nor π_j; then

$$((\mu\alpha\,\nu)\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha_1,\ldots,\alpha_n}\,(\pi_1, \ldots, \pi_n) = \mu\alpha\,N'$$

with

$$N' = \ (\nu\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha_1,\ldots,\alpha_n}\,(\pi_1, \ldots, \pi_n).$$

We show that $\mu\alpha\,N' \in \mathsf{R}\langle A^*\rangle$, using the property that $A^* = A^\perp \to \mathcal{N}$: let $\pi \in A^\perp$, we write

$$\pi = f_1 \ldots f_r$$

and for $k = 1, \ldots, r$, we write

$$f_k = (w_1^{(k)}, \ldots, w_{q_k}^{(k)}; V_k).$$

We have to prove $(\mu\alpha\,N')\,\pi \in \mathsf{R}\langle\mathcal{N}\rangle$. For this purpose, we can apply corollary 5.17 since:

  – $\mathcal{N}$ is saturated;
  – π is an $\mathcal{N}$-stack;
  – $N' \in \mathsf{R}\langle\mathcal{N}\rangle$: the empty stack $\varepsilon \in A^\perp$ so that

$$N' = \left(N'\right)_\alpha \varepsilon = (\nu\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha_1,\ldots,\alpha_n,\alpha}\,(\pi_1, \ldots, \pi_n, \varepsilon) \in \mathsf{R}\langle\mathcal{N}\rangle$$

    by induction hypothesis.

Hence it remains to show that

  – for all $k \in [1, r]$, for all $I \subseteq [1, q_k]$, $\mu\alpha\left(\mathrm{D}^g_\alpha\,(N')_\alpha\,f_1 \ldots f_{k-1} \cdot w_I^{(k)}\right) \in \mathsf{R}\langle\mathcal{N}\rangle$ (with $g = |I|$);
  – $(N')_\alpha\,\pi \in \mathsf{R}\langle\mathcal{N}\rangle$.

Let $z_1, \ldots, z_m, \beta, \beta_1, \ldots, \beta_n$ be fresh variables and names, in particular distinct from $x_1, \ldots, x_m$ and $\alpha, \alpha_1, \ldots, \alpha_n$ respectively, and not free in ν nor π nor any e_i nor any π_j. If W is a term (or stack), we write $\widehat{W}$ for $W\,[z_1, \ldots, z_n, \beta, \beta_1, \ldots, \beta_n/x_1, \ldots, x_m, \alpha, \alpha_1, \ldots, \alpha_n]$. Assume $k \in [1, r]$, $I \subseteq [1, q_k]$ and $g = |I|$; we prove $N'' = \mathrm{D}^g_\alpha\,(N')_\alpha\,f_1 \ldots f_{k-1} \cdot w_I^{(k)} \in \mathsf{R}\langle\mathcal{N}^\Box\rangle$. By lemma 3.11, this is implied by $(N'')_\alpha\,V_k \in \mathsf{R}\langle\mathcal{N}^\Box\rangle$. But $(N'')_\alpha\,V_k = (N')_\alpha\,\theta$, with $\theta = f_1 \ldots f_{k-1}(w_I^{(k)}; V_k)$. Since $A^\perp$ is closed under prefix, $\preceq$ and renaming, $\widehat{\theta} \in A^\perp$. Moreover variables $x_1, \ldots, x_m$ and names $\alpha, \alpha_1, \ldots, \alpha_n$ don't occur free in $\widehat{\theta}$. Hence, by induction hypothesis,

$$\left(N'\right)_\alpha \widehat{\theta} = (\nu\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha,\alpha_1,\ldots,\alpha_n}\left(\widehat{\theta}, \pi_1, \ldots, \pi_n\right) \in \mathsf{R}\langle\mathcal{N}^\Box\rangle.$$

But $(N')_\alpha\,\theta = \left((N')_\alpha\,\widehat{\theta}\right)[x_1, \ldots, x_n, \alpha, \alpha_1, \ldots, \alpha_n/z_1, \ldots, z_m, \beta, \beta_1, \ldots, \beta_n]$ and since $\mathcal{N}$ is closed under renaming, $(N')_\alpha\,\theta \in \mathsf{R}\langle\mathcal{N}^\Box\rangle$. Now let's prove $\mu\alpha\,(N')_\alpha\,\pi \in \mathsf{R}\langle\mathcal{N}\rangle$. Since $A^\perp$ is stable under renaming, $\widehat{\pi} \in A^\perp$. Moreover, variables $x_1, \ldots, x_m, x$ and names $\alpha_1, \ldots, \alpha_n$ don't occur free in $\widehat{\pi}$. Hence, by induction hypothesis,

$$\left(N'\right)_\alpha \widehat{\pi} = (s\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha,\alpha_1,\ldots,\alpha_n}\,(\pi, \pi_1, \ldots, \pi_n) \in \mathsf{R}\langle\mathcal{N}^\Box\rangle.$$

Since $\mathsf{R}\langle\mathcal{N}^\Box\rangle$ is closed under renaming,

$$\mu\alpha\left(N'\right)_\alpha \pi = \mu\alpha\left(\left(\left(N'\right)_\alpha \widehat{\pi}\right)[x_1, \ldots, x_m, \alpha, \alpha_1, \ldots, \alpha_n/z_1, \ldots, z_m, \beta, \beta_1, \ldots, \beta_n]\right) \in \mathsf{R}\langle\mathcal{N}\rangle.$$

- *Naming:* $A = B_{j_0}$ and $\nu = [\alpha_{j_0}]\,s$ for some $j_0 \in [1, n]$ with

$$x_1 : A_1, \ldots, x_m : A_m \vdash s : B_{j_0} \mid \alpha_1 : B_1, \ldots, \alpha_n : B_n.$$

Then by lemma 2.27,

$$(\nu\,[e_1, \ldots, e_m/x_1, \ldots, x_m])_{\alpha_1,\ldots,\alpha_n}\,(\pi_1, \ldots, \pi_n)$$

is a sum of terms of the shape

$$[\alpha_{j_0}]\left(\left(S'\right)\pi''_{j_0}\right)$$

with

$$S' = \left(s\left[e_1,\ldots,e_m/x_1,\ldots,x_m\right]\right)_{\alpha_1,\ldots,\alpha_n}\left(\pi'_1,\ldots,\pi'_n\right)$$

where $\pi'_j = \pi_j$ if $j \neq j_0$, $\pi'_{j_0} \preceq \pi_{j_0}$ and $\pi''_{j_0} \preceq \pi_{j_0}$. Hence π'_{j_0} and $\pi''_{j_0} \in B^{\perp}_{j_0}$. Directly from induction hypothesis, $S' \in \mathsf{R}\langle B^{*}_{j_0}\rangle$ and by definition of $B^{\perp}_{j_0}$ we have $(S')\,\pi''_{j_0} \in \mathsf{R}\langle\mathcal{N}\rangle$, hence the conclusion.

□

References

[Bar84] Henk Barendregt. *The Lambda-calculus*. North Holland, 1984.

[dG98] Philippe de Groote. An environment machine for the lambda-mu-calculus. *Mathematical Structures in Computer Science*, 8:637–669, 1998. Available at `http://www.loria.fr/~degroote/papers/mscs98.ps`.

[dLP95] Ugo de' Liguoro and Adolfo Piperno. Nondeterministic extensions of untyped λ-calculus. *Information and Computation*, 149-177(122), 1995.

[DP01] René David and Walter Py. lambda mu - calculus and Böhm's theorem. *J. Symb. Log.*, 66(1):407–413, 2001. Available at `http://www.lama.univ-savoie.fr/~david/ftp/bohm.ps`.

[Ehr01] Thomas Ehrhard. On Köthe sequence spaces and linear logic. *Mathematical Structures in Computer Science*, 12:579–623, 2001. Available at `http://iml.univ-mrs.fr/~ehrhard/pub/kothe.ps.gz`.

[Ehr04] Thomas Ehrhard. Finiteness spaces. Available at `http://iml.univ-mrs.fr/~ehrhard/pub/finite.ps.gz`, 2004.

[ER03] Thomas Ehrhard and Laurent Régnier. The differential lambda-calculus. *Theoretical Computer Science*, 309:1–41, 2003. Available at `http://iml.univ-mrs.fr/~ehrhard/pub/difflamb.ps.gz`.

[ER04] Thomas Ehrhard and Laurent Régnier. Uniformity and the taylor expansion of ordinary lambda-terms. Available at `http://iml.univ-mrs.fr/~ehrhard/pub/resources.ps.gz`, 2004.

[ER05] Thomas Ehrhard and Laurent Régnier. Böhm trees, Krivine machine and the Taylor expansion of ordinary lambda-terms. Available at `http://iml.univ-mrs.fr/~ehrhard/pub/bkt.ps.gz`, 2005.

[Gir87] Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987. Available at `http://iml.univ-mrs.fr/~girard/linear.pdf`.

[Kri90] Jean-Louis Krivine. *Lambda-calcul, types et modèles*. Masson, Paris, 1990.

[Lau98] Olivier Laurent. Réseaux de preuve polarisés : des réseaux pour LC. Mémoire de D.E.A. de logique et fondements de l'informatique, Université Paris VII, September 1998. Available at `http://www.pps.jussieu.fr/~laurent/dea.ps.gz`.

[Lau03] Olivier Laurent. Polarized proof-nets and λμ-calculus. *Theoretical Computer Science*, 290(1):161–188, January 2003. Available at `http://www.pps.jussieu.fr/~laurent/lmpn.ps.gz`.

[Par91] Michel Parigot. Free deduction: An analysis of "computations" in classical logic. In *RCLP*, pages 361–380, 1991.

[Par92] Michel Parigot. λμ-calculus: An algorithmic interpretation of classical natural deduction. In *LPAR '92: Proceedings of the International Conference on Logic Programming and Automated Reasoning*, pages 190–201, London, UK, 1992. Springer-Verlag.

[Par97] Michel Parigot. Proofs of strong normalisation for second order classical natural deduction. *J. Symb. Log.*, 62(4):1461–1479, 1997.

[Sau05] Alexis Saurin. Separation with streams in the λμ-calculus. In *LICS*, pages 356–365, 2005. Revised version available at `http://www.eleves.ens.fr/home/saurin/Recherche/Separation_Lambdamu.pdf`.

[SR98] Thomas Streicher and Bernhard Reus. Classical logic, continuation semantics and abstract machines. *J. Funct. Program.*, 8(6):543–572, 1998. Available at `http://www.mathematik.tu-darmstadt.de/~streicher/CONT/JFP.ps.gz`.

[Vau06] Lionel Vaux. λ-calculus in an algebraic setting. Available at `http://iml.univ-mrs.fr/~vaux/articles/alglam.ps.gz`, 2006.