Intuitionistic Differential Nets and Lambda-Calculus

Paolo Tranquilli [1]

Dipartimento di Matematica – Università Roma Tre

Laboratoire Preuves Programmes et Systèmes – Université Denis Diderot - Paris 7

Abstract

We define pure intuitionistic differential proof nets, extending Ehrhard and Regnier's differential interaction nets with the exponential box of Linear Logic. Normalization of the exponential reduction and confluence of the full one is proved. These results are directed and adjusted to give a translation of Boudol's untyped λ-calculus with resources extended with a linear-non linear reduction à la Ehrhard and Regnier's differential λ-calculus. Such reduction comes in two flavours: baby-step and giant-step β-reduction. The translation, based on Girard's encoding $A \to B \sim\, !A \multimap B$ and as such extending the usual one for λ-calculus into proof nets, enjoys bisimulation for giant-step β-reduction. From this result we also derive confluence of both reductions.

Key words: Lambda-calculus, differential interaction nets,
linear logic, proof nets, exponential reduction, confluence, normalization

1 Introduction

Twenty years ago Jean-Yves Girard introduced Linear Logic (LL, [12]) starting from a fine analysis of the coherent semantics he had introduced for system F. This logical framework has provided a new looking glass for the study of the essence of computation in general, and λ-calculus specifically. Particularly important for the background of this paper is the translation of pure and typed λ-calculus into Girard's proof nets, as studied by Danos and Regnier in their theses [3, 19]. It has proved to be a powerful tool to bring

Email address: `tranquil@mat.uniroma3.it` (Paolo Tranquilli).

[1] Partly supported by Università Italo-Francese (Programma Vinci 2007).

forth the study of both sides of the mapping, proof nets on one side and λ-calculus on the other. This translation comes in two forms: one, denoted by t°, which gives bijectively proof nets without exponential cuts, and another, $t^{\bullet}$, defined as the multiplicative normal form of t°, which quotients terms with an operational equivalence, the σ-equivalence, described in [20].

Recently Ehrhard has defined a semantics of topological vector spaces and continuous linear maps [7, 8] which fully employs some intuitions from linear algebra that may be already found in an "embryonic state" in coherent spaces [2]. Again from such semantical development the same author and Regnier presented extensions with syntactic differential operators for both Linear Logic [10] and λ-calculus [9]. One of the ideas supporting such endeavours is that taking the derivative of a function f and applying it to an argument (as derivatives give linear forms) can be seen from the computational and logical point of view as providing f with a single-use occurrence of that argument. The treatment of the subject can therefore rely on a line of research already present in λ-calculus. Starting from Boudol's work on λ-calculus with multiplicities [1], variants of λ-calculus where studied where arguments could have a limited availability. In [10] Ehrhard and Regnier introduce the link between the two approaches – a translation to their promotion-free differential interaction nets from the fragment of Boudol's calculus without infinitely available resources, the *resource calculus*.

The following are the two main contributions of this paper.

- We prove that pure intuitionistic differential proof nets with promotion are a "good" rewriting system – the exponential reduction is strongly normalizing, and the whole one is confluent.
- We use such results to fully develop the link between differential proof nets and a refined version of Boudol's full λ-calculus with resources. We establish between the two the same strong connection existing between proof nets and λ-calculus. A similar pairing can be found between polarized proof nets and $\lambda\mu$-calculus [15].

In the next section we will outline the story so far, pointing out the issues and the starting points that have motivated our research, and setting the goals for the following sections. Then, in Section 3, we define pure intuitionistic differential proof nets and prove the results we stated above, and refine them for the upcoming translation. In Section 4 we switch to λ-calculus, and present *full resource calculus*, which is Boudol's λ-calculus with resources enriched with the dynamics of Ehrhard and Regnier's differential λ-calculus.

[2] Namely, the operations on webs underlying tensor, dual or (direct) sum of coherent spaces are the sames done on bases in their counterparts of finite-dimensional vector spaces. In the spaces of [7, 8] such correspondence is exact, without being limited to finite dimension (a feature necessarily broken by exponential modalities).

Finally in Section 5 we define the translation from full resource calculus to differential proof nets, and show sequentialization and bisimulation.

Notation. We will denote sets of reduction rules with letters such as $\mathtt{m}$ or $\mathtt{e}$, and by $\xrightarrow{\mathtt{r}}$ ($\mathtt{r}$-reduction) the relation corresponding to rules $\mathtt{r}$, obtained by context closure. The relations $\xrightarrow{\mathtt{r}=}$, $\xrightarrow{\mathtt{r}+}$, $\xrightarrow{\mathtt{r}*}$ and $\equiv_{\mathtt{r}}$ are respectively the reflexive, transitive, reflexive-transitive and equivalence closures of $\xrightarrow{\mathtt{r}}$. An element u is **$\mathtt{r}$-normal** if there is no v with $u \xrightarrow{\mathtt{r}} v$. We write $u \xrightarrow{\mathtt{r}}\!\!\!\!\rightarrow v$ if $u \xrightarrow{\mathtt{r}*} v$ and v is $\mathtt{r}$-normal. Reduction $\xrightarrow{\mathtt{rs}}$ is the union of reductions $\xrightarrow{\mathtt{r}}$ and $\xrightarrow{\mathtt{s}}$. $R : u \xrightarrow{\mathtt{r}*} v$ or $u \xrightarrow{R} v$ denotes a given chain R of reduction steps from u to v, and $|R|$ denotes the length of R. The properties of confluence, its variants (local and strong) and of strong normalization are defined as usual.

$\mathcal{M}_{\mathrm{fin}}(X)$ is the set of finite multisets over X, i.e. functions $A : X \to \mathbb{N}$ with support $|A| < \omega$ finite. Depending on the context multisets will be presented either in additive or in multiplicative notation. In any case $\sum_{a\in A} D_a$ stands for a sum with multiplicities, i.e. $\sum_{a\in|A|} A(a) \cdot D_a$. For example cardinality is $\# A = \sum_{a\in A} 1$.

R will be a commutative semiring with unit, and $R\langle S\rangle$ is the R-module generated by S, i.e. the set of formal *finite* sums $\sum_{s\in S} c_s s$ over S with coefficients in R. We will usually have $R = \mathbb{N}$, and in such a case $\mathbb{N}\langle S\rangle = \mathcal{M}_{\mathrm{fin}}(S)$ and each sum can be written without coefficients.

2 State of the art

Our starting point is the pairing between resource calculus and Ehrhard and Regnier's differential interaction nets (DINs) given in [10], and the attempt at extending it to the same authors' differential λ-calculus [9]. We will skip over some definitions and technical points in this section. For a definition of pure DINs[3] one may refer to the next section, and take the promotion-free fragment of intuitionistic differential proof nets.

2.1 *Resource calculus and differential interaction nets*

Starting from different motivations various authors have studied resource calculi [1, 2, 13]. Ehrhard and Regnier give a presentation of Boudol's calculus with resources with a reduction borrowed from their differential λ-calculus, and a restriction to the linear fragment by ruling out infinitely

[3] They are called DR typed nets in [10].

available arguments. We present it here.

Given a denumerable set of variables $\mathbb{V}$ the set of simple terms Δ is defined by the following grammar:

$$\Delta ::= \mathbb{V} \mid \lambda\mathbb{V}.\Delta \mid \langle\Delta\rangle\,\Delta^{!},$$

where $\Delta^{!} := \mathcal{M}_{\mathrm{fin}}(\Delta)$, presented in multiplicative notation, is the set of *bags of arguments* [4]. This language is extended to $R\langle\Delta\rangle$, the set of terms, and the constructors of the grammar extended by multilinearity. We write $x \in t$ to mean "x free in t" as usual [5]. We define the 0-substitution by $t\,[x := 0] := 0$ if $x \in t$, and t otherwise. This is clearly the usual substitution with 0 if we take into account multilinearity. Moreover we have the linear substitution defined by

$$\frac{\partial y}{\partial x} \cdot u := \delta_{x,y} \cdot u, \qquad \frac{\partial \lambda y.s}{\partial x} \cdot u := \lambda y.\frac{\partial s}{\partial x} \cdot u \quad \text{with } y \notin u,$$
$$\frac{\partial \langle r\rangle\, A}{\partial x} \cdot u := \left\langle \frac{\partial r}{\partial x} \cdot u \right\rangle A + \langle r\rangle\,\frac{\partial A}{\partial x} \cdot u, \qquad \frac{\partial A}{\partial x} \cdot u := \sum_{v \in A} \left(\frac{\partial v}{\partial x} \cdot u \right) A/v,$$

where $\delta_{x,y} = 1$ if $x = y$, 0 otherwise. The notation reflects the fact that this substitution can be regarded as a partial derivative of a term in the direction of u. Strengthening such idea is the validity of Schwartz's lemma, in the sense that if $x \notin v$ and $y \notin u$ we have the commutation $\frac{\partial}{\partial x}\left(\frac{\partial t}{\partial y} \cdot v\right) \cdot u = \frac{\partial}{\partial y}\left(\frac{\partial t}{\partial x} \cdot u\right) \cdot v$. Restricting to $R = \mathbb{N}$, reduction is defined by

$$\langle \lambda x.s\rangle\, uA \;\beta_{\mathrm{bs}}\; \left\langle \lambda x.\frac{\partial s}{\partial x} \cdot u \right\rangle A, \qquad \langle \lambda x.s\rangle\, 1 \;\beta_{\mathrm{bs}}\; s\,[x := 0]\,,$$

first extended to simple terms and bags as a context closure and then on terms by linearity. One should notice that there is a choice regarding the term to be fetched from the bag, however Schwartz's lemma and linearity of substitution assure strong confluence, and even in this untyped setting strong normalization holds. This approach differs from Boudol's one, which defines a completely non-deterministic (therefore non confluent) lazy reduction. Here one keeps track of choices with sums, and moreover the reduction does not only substitute head variables. The bs in β_{bs} stands for *baby-step* β-reduction, as we can regard it as opposed to the reduction β_{gs}, *giant-step* β-reduction, that completely exhausts the redex. In our setting the two reductions are presented in Definition 19 [6].

The translation t° of this calculus can be regarded as a particular case of the one given in detail in Section 5. For now we can say that variables and

[4] They are called poly-terms in [10].
[5] We skip the subtleties involved with sums. A fine syntactical treatment of them can be found in [22].
[6] In [11] the two reductions are called *small-step* and *big-step*.

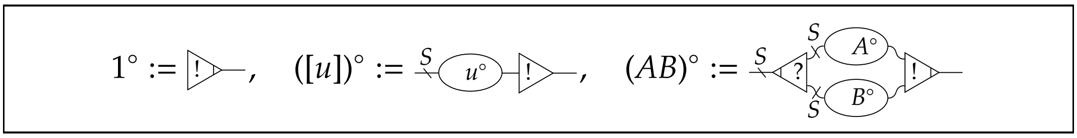


Figure 1: Rules to translate bags of arguments.

abstractions are treated in the same way as for λ-calculus. However, as DINs are defined with binary contractions and cocontractions, a bag is translated by writing it down as an iterated application of the binary merge operation on multisets, starting from singletons. This is shown in Figure 1. Application $\langle r\rangle A$ is translated by plugging A° on a tensor cut against the output port of r°, just like boxes are in the translation of the application of λ-calculus into proof nets. One should note that the translation of a bag A is different for each different way of writing A by means of binary merge operations. In [10] the solution is stated but not discussed, as the different nets are said to be equivalent modulo a notion left for future work, which is associativity of (co)contraction and neutrality of (co)weakening with respect to (co)contraction. Here we settle such notion by means of a reduction, and moreover we will also show we cannot really ignore the issue when boxes are around (Remark 3).

Given such an equivalence $\equiv_{\mathsf{a}}$, the rigorous statement of the simulation result is that

$$u\ \beta_{\mathrm{bs}}\ v \implies u^\circ \equiv_{\mathsf{a}} \xrightarrow{\mathtt{m}} \xrightarrow{\mathtt{e}*} \xleftarrow{\mathtt{m}} \equiv_{\mathsf{a}} v^\circ,$$

where $\mathtt{m}$ is the multiplicative reduction $⅋/\otimes$, and $\mathtt{e}$ is the exponential reduction $?/!$. We also have to rebuild the multiplicative redex by $\xleftarrow{\mathtt{m}}$. A better statement may be achieved by either considering giant-step reduction, for which the above result becomes

$$u\ \beta_{\mathrm{gs}}\ v \implies u^\circ \xrightarrow{\mathtt{m}} \overset{\mathtt{e}}{\twoheadrightarrow} \equiv_{\mathsf{a}} v^\circ,$$

or by adopting the translation $t^\bullet$ which normalizes multiplicative cuts, for which we would have

$$u\ \beta_{\mathrm{bs}}\ v \implies u^\bullet \equiv_{\mathsf{a}} \xrightarrow{\mathtt{e}*} \overset{\mathtt{m}}{\twoheadrightarrow} \equiv_{\mathsf{a}} v^\bullet.$$

Final a-conversion is needed to accommodate the arbitrary way in which $v^\circ/v^\bullet$ has been built. The initial one in β_{bs} is needed instead to fetch the argument from the bag that contains it, otherwise it might be buried by several cocontractions.

This problem with (co)contractions arises often in the translation of various calculi into nets. The order in which variable occurrences are identified and dummy variables are introduced is usually abstracted away in calculi, while respectively binary contractions and weakenings explicitly set it. Solutions proposed in LL include

- adopting a syntax which identifies contractions made at several exponential depths, as in [19] – for now it seems hard to apply it in differential nets with boxes, mainly because of the rule of codereliction against box;
- using such an identification as an equivalence relation, as hinted in [10] for DINs and investigated in [5] for LL proof nets – an elegant solution, though it is less so with respect to freely moving around weakenings, as it may generate infinite trees with weakened leaves;
- using it as a set of reductions, as in [6] – which is the way we are adopting here (Section 3.4).

2.2 *Differential λ-calculus and differential nets*

A natural direction of investigation arising from [9, 10] is the question whether differential λ-calculus can be translated into differential nets. The first problem which arises is that DINs are promotion-free, and though it is easy to define the extension, it has not yet been treated in the literature, other than by replacing boxes with their Taylor expansion, i.e. an infinite sum which therefore deprives the system of its finitary nature. In the next section we will thus introduce *differential nets* or DNs, dropping the "interaction" wording as Lafont's interaction net paradigm [14] is broken by the promotion cell. We will call differential *proof* nets or DPNs the DNs that are correct by the usual Danos-Regnier criterion [7] . As DPNs are not interaction nets anymore, fundamental results like confluence or normalization are far harder.

The second problem is related to the syntax of differential λ-calculus, which we briefly sketch here. Terms are defined by the grammar

$$\Lambda ::= \mathbb{V} \mid \lambda\mathbb{V}.\Lambda \mid (\Lambda)\, R\,\langle\Lambda\rangle \mid \mathrm{D}\,\Lambda \cdot \Lambda.$$

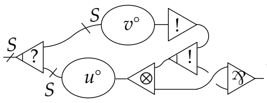
As application is linear in the function but not in the argument, sums must be kept in argument position. The construct $\mathrm{D}\,u \cdot v$ stands for taking the derivative of u linearly applied to v. From the computational point of view, it corresponds to providing u with a single-use instance of v. So, apart from the usual β-reduction, one defines the *linear reduction* $\mathrm{D}\,\lambda x.s \cdot u \to \lambda x.\frac{\partial s}{\partial x} \cdot u$. The linear substitution $\frac{\partial s}{\partial x} \cdot v$ is analogous to the one described for resource calculus, with particular care in handling the application (by means of a "linearization on the fly" similar to what we describe on page 25) [8] . Now if

[7] So here the neutral word "net" replaces the concept of proof structure.

[8] The syntax originally described in [9] has operators $\mathrm{D}_i\, u \cdot v$, standing for the derivative in the ith argument of u in the direction of v. It has already been remarked [21, Remark 1.4] that this conflicts with the intrinsic currying of λ-calculus. This conflict is highlighted even more when trying to translate in differential nets.

we try to give a translation in DPNs extending the one for λ-calculus, we would map the redex D λ to a multiplicative cut, so one chooses to represent D with a tensor cut against the main conclusion of the differentiated term. However in the reduct of linear reduction the λ is still present. One might therefore represent such a situation with

$$(\mathrm{D}_1\, u \cdot v)^\circ :=$$



,

where the rightmost ⅋ corresponds to a potential abstraction that gets there if D fires. This is disquieting: as opposed to λ-calculus' translation, there is not a "local" correspondence between the resulting net and the starting term. Whether a ⅋ is an actual λ or a "phantom" one due to a differentiation depends on what is around it. This could make a sequentialization proof hard if not impossible.

We therefore chose to look in another direction. The usual translation of λ-calculus into proof nets comes in two flavours t° and $t^\bullet$, both with strong properties. In particular t° is bijective on proof nets without exponential cuts (once one rules out exponential axioms) and enjoys bisimulation for β-reduction. So we looked for a calculus that would have both the translations with the same properties, and we arrived at a version of full Boudol's λ-calculus with resources. In fact just like resource calculus described by Ehrhard and Regnier in [10] is a an algebraic non-lazy version of the linear fragment of λ-calculus with resources, the *full resource calculus* we describe in Section 4 is the same for full Boudol's calculus. We may say that it is Boudol's calculus enriched with the dynamics of Ehrhard and Regnier's differential λ-calculus, which explains why such a strong link with differential nets can be found. After the next two sections, in Section 5, we will finally be able to define the translation and show sequentialization (i.e. surjectivity) and bisimulation of t°. The next stage, the translation $t^\bullet$, is left for future work.

3 Intuitionistic differential proof nets

Intuitionistic differential proof nets (or intuitionistic DPNs) are an extension of intuitionistic MELL proof nets with new rules (codereliction, cocontraction and coweakening). Due to our main interest here in λ-calculus, we will deal with a pure version of intuitionistic DPN. Typed and non intuitionistic versions are left for future work. Following the naming convention of [10] (though, as already explained, dropping "interaction"), *differential nets* will denote the nets freely built with the cells available, with no assumption about correctness/sequentializability, i.e. they take the role played by proof structures in MELL.

A **net** is given by the following data.

- A finite set P of **free ports**, also called **conclusions**.
- A finite set C of **cells**, to each of which is assigned a symbol, a **principal port** and a finite ordered sequence of **auxiliary ports**. The number of all these ports, which go by the collective name of **connected ports**, is called **arity** of the cell.
- A finite set W of **wires** which is the union of a partition of the set of ports into sets with 2 elements and some wires not related to any port (**deadlocks**).

Cells are typically graphically depicted as triangles with the principal port on a vertex and the auxiliary ones on the opposed side. A cell is said to be **commutative** if its auxiliary ports are to be considered an unordered set rather than a sequence [9] .

A **typing** is the assignment to all directed ports of a formula in a given language with duals. A directed port is a couple of a port and a direction – **incoming** or **outgoing** from the cell for connected ports, while on free ports incoming is given the meaning of outgoing from the net and vice versa. One imposes that if A is assigned to an outgoing port, then $A^\perp$ is assigned to the same incoming port and vice versa. Rules will be given for assigning types to ports of cells with a given symbol.

A net is **typable** with a given typing if for each wire between ports the outgoing type of one of its ports is equal to the incoming type of the other. If we assign a direction to any non deadlock wire, turning it into an ordered couple, its type is the outgoing type of its first port. We will call **interface** of a typed net the multiset of the outgoing types of its conclusion. At times, depending on the context, the same name will be used for set of conclusions. Deadlocks are voluntarily left out of the discussion.

Differential nets. The set DN_0 of pure 0-depth simple intuitionistic differential net (or 0-depth simple DNs for short) is the set of nets typable with formulas o, $!o$ and respective duals $\imath$, $?\imath$ with symbols, arities and typing rules defined in Figure 2, excluding the promotion cell. Then by induction the set DN_{k+1} of $k+1$-depth simple DNs is the set of nets built with all cells in Figure 2. To each promotion cell with n ports is associated an element π in $R\langle \mathrm{DN}_k \rangle$ where all addends have an interface of $n-1$ $?\imath$ and an o. This associated sum is called the content of the box and has a fixed correspondence

[9] One can give a more formal definition by defining an equivalence relation on nets and taking the equivalence classes thereafter.

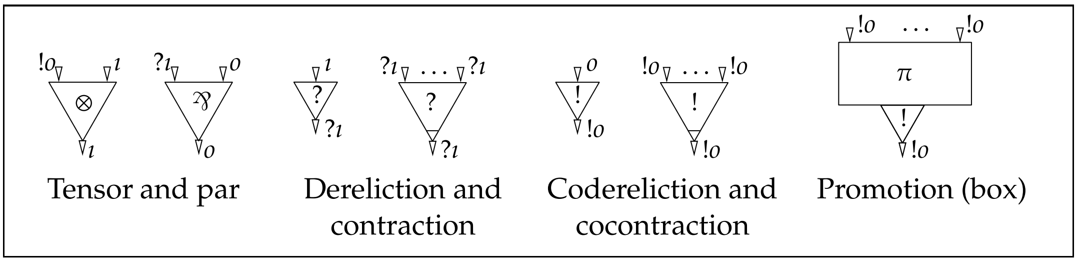


Figure 2: Cells for intuitionistic differential nets. Contractions and cocontractions are commutative and cannot have 2 ports.

between its $?\iota$-conclusions and the auxiliary ports of the box. The set of simple DNs is $\mathrm{DN} := \cup_{k\in\mathbb{N}}\mathrm{DN}_k$. Intuitionistic differential nets are elements of $R\langle\mathrm{DN}\rangle$ where all addends have the same interface. The (exponential) depth of a net π is the minimal k such that $\pi \in R\langle\mathrm{DN}_k\rangle$. The exponential depth of a cell in π is the number of boxes in which it is contained. From now on we will not write "pure" and "intuitionistic" anymore and we will often drop "differential", as these nets are the only ones present in this paper.

In fact the typing rules of cells implement the isomorphisms usually employed to interpret untyped λ-calculus via Girard's translation of the intuitionistic implication ($o \cong o \to o = !o \multimap o = ?\iota \parr o$). We will often omit these types in figures, as they can be easily derived. We will call n**-contraction** (resp. n**-cocontraction**) one which has $n+1$ ports. 0-(co)contractions are also called (co)weakenings. A wire is **exponential** if its type is $?\iota/!o$, and **multiplicative** otherwise. A **cut** is a wire which either connects two principal ports or a principal port and the auxiliary port of a box. An **axiom** is a wire which does not connect any principal or box auxiliary port.

Contexts. A **simple context** $\omega[\ ]$ is a simple differential net built with an additional special cell, the **hole**, which has an arbitrary arity and outgoing types, the sequence of which is called the **internal interface** of $\omega[\ ]$. We impose that the hole appears (syntactically) only once in $\omega[\ ]$: formally it means that either it appears once at exponential depth 0, or inductively there is one box which contains $a\psi[\ ] + \sigma$ with $a \neq 0$ and $\psi[\ ]$ simple context. Similarly, a differential context is $a\omega[\ ] + \pi$ with π differential net and $\omega[\ ]$ simple context [10] . Given a simple DN π and a context $\omega[\ ]$ such that the interface of π is equal to the internal interface of $\omega[\ ]$ we define $\omega[\pi]$ by substituting π for the hole, i.e. identifying the free ports of π with corresponding ports of the hole and then erasing them by merging wires which share such ports. In case π is a linear combination the sum is extended to the content of the box containing the hole, or the whole context if there is none. Given a relation

[10] It may be noted that for example $2[\ ] = [\ ] + [\ ]$ is considered a one-hole context. Doing differently when non-integer coefficients are around would be troublesome, and moreover the reduction defined as a context closure with this definition coincides with the one given in [10].

ρ on DNs its **context closure** is $\pi\ \tilde{\rho}\ \sigma$ iff there is a context $\omega[\ ]$ and two nets $\pi'\ \rho\ \sigma'$ such that $\pi = \omega[\pi']$ and $\sigma = \omega[\sigma']$.

Correct nets. Though DNs already have computational meaning, we define the **correctness criterion** following the Danos-Regnier one for LL proof nets [4]. Given a simple deadlock-free DN λ a **switching** of λ is an unoriented graph G with cells as nodes, obtained by deleting for every par and contraction the wires on all its auxiliary ports but one and converting all remaining wires as edges between the cells they connect. A **principal switching** is one that on $⅋$s always erases the exponential wire. A simple DN λ is said to be **correct**, or a simple **differential proof net**, or simple DPN for short, if it is deadlock-free, every switching G of λ is acyclic and with a number of connected components equal to the number of weakenings at depth 0 in λ plus one, and moreover if inductively every content of its boxes is correct. A DN is correct if it is a sum of simple DPNs. We speak of **differential modules** if we have only the acyclicity condition, and every box content is correct. This is the minimal correctness we need to be able to plug the module in a context and hope the result is correct: a cyclic net, or one which has incorrect box contents, gives incorrect nets no matter the context in which it is plugged.

Lemma 1 *A DPN net has exactly one o or $!o$ conclusion.*

PROOF (sketch). This proof is no different from what is done for LL intuitionistic proof nets. See for example [19]. The idea is to use paths in a principal switching, first to end up on a $o/!o$ conclusion, then to arrive at a contradiction if two such conclusions are supposed.

3.2 Dynamics: multiplicative, exponential, associative reductions

From now on we will assume $R = \mathbb{N}$. Though greatly interesting, other cases such as $\mathbb{Q}^+$ pose problems for normalization issues, not to speak of cases where R has an opposite -1 to the unit, where every term reduces to any other [21, 22]. In this setting sums may always be written without coefficients. We may also redefine contexts, ruling out the multiplication of the hole by a coefficient, and making the upcoming definition of reduction more atomic. This is left to personal taste, as the results do not change.

Figure 3 presents various sets of reduction rules on modules, which as already explained in Section 2 are to be extended by context closure to obtain the reduction relation. Note that the rules cover also the cases for (co)weakening. The $\mathtt{m}$-reduction is the **multiplicative** one, $\mathtt{e}$ is the **exponential** one, and $\mathtt{a}$ is the **associative** one, implementing associativity of (co)contraction and neutrality to it of (co)weakening. Remark 3 shows why we are dealing with $\mathtt{a}$-reduction together with the other more classical ones: $\mathtt{e}$-reduction

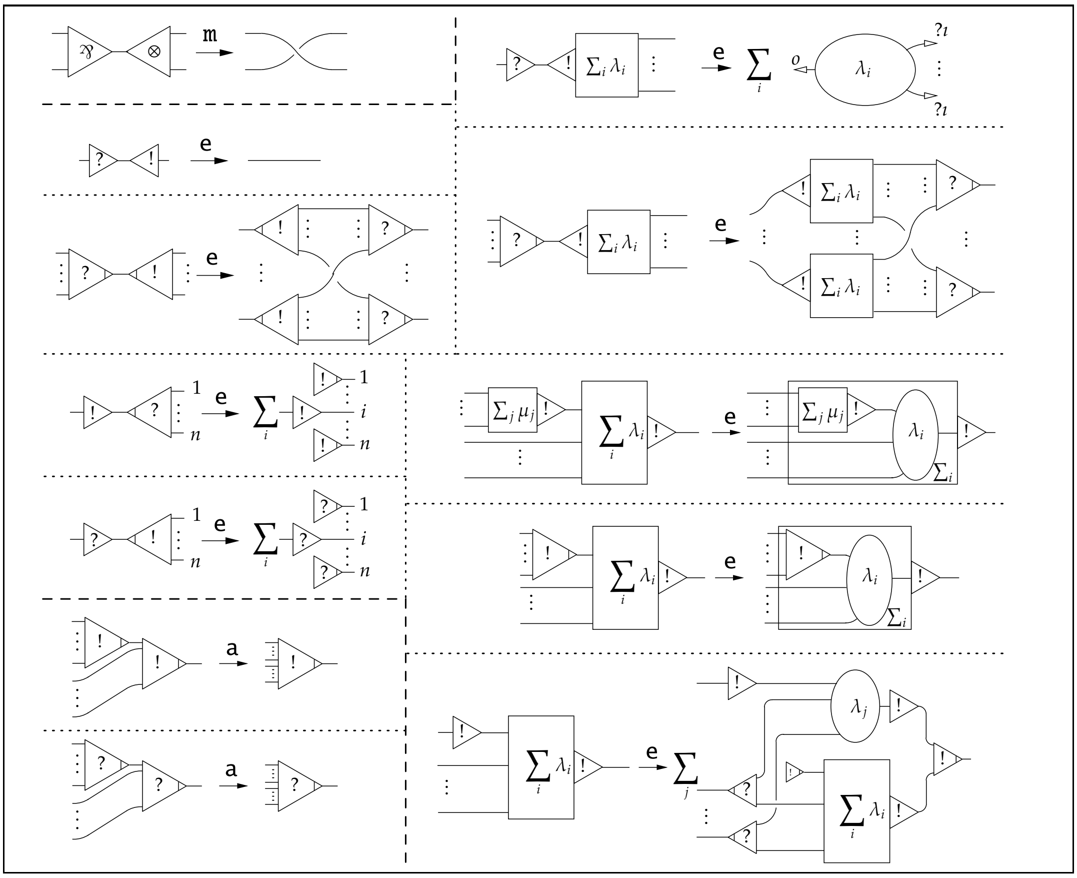


Figure 3: Reduction rules for differential nets. In a-reduction rules contractions or cocontractions in the reducts which come out to have 2 ports are a convention to denote a single connecting wire.

(and `me`-reduction) is not confluent without it. Reductions can be seen to preserve both typing and correctness. From now on all nets are to be considered correct.

Lemma 2 *The reduction* `ea` *is locally confluent.*

PROOF. As usual one checks the critical pairs. The ones that have not been covered in the literature about LL proof nets are easy, if somewhat long, to verify. We will show here one of the most interesting cases, codereliction vs box vs contraction, making the simplification that the box has two auxiliary ports and that the contraction is a 2-contraction. The two reductions are shown in Figure 4. In the end we arrive to two a-equivalent nets, which a-normalize to the same one. □

Remark 3 *The confluence diagram shown in Figure 4 proves also that* `e` *alone is not confluent, contrary to what happens in LL proof nets, where confluence of exponential (and general) reduction is independent of associativity.*

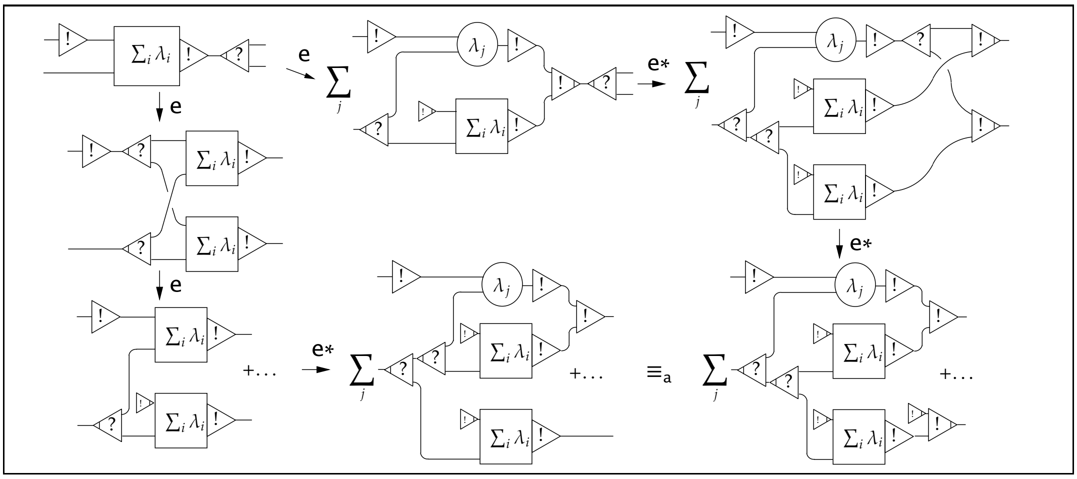


Figure 4: Confluence diagram for codereliction vs box vs contraction critical pair. The + . . . parts are the other addends in the sums which are symmetric.

3.3 *Strong normalization of exponential reduction*

We will now begin the most technical part of the paper: we will prove strong normalization of e first, and ea after that. It is crucial here that we do not have double exponential types: once an exponential is deleted, say for example by a dereliction against codereliction reduction, no new cut is exponential.

Sketch of the proof technique. We want to define a decreasing measure on the net. We start by assigning to each cut a natural number. After a cut is fired, the cuts created by the reduction have a lesser weight, though there may be many of them. Thus we employ the multiset of the weights of the cuts with multiset order. Another problem arises: sums make it so that when a reduction creates addends, there is a sort of global duplication of the net. This can be settled with multisets again: one takes the multiset of the multisets of weights given by the various addends, so that if all addends of the reduct have a multiset lower than the one of the redex we are done. This almost settles the issue, were not for promotion. Boxes can be duplicated, but fortunately there is a way to foresee how many copies of the boxes may be created. So we count the weights inside boxes as many times as these potential copies. Last problem: boxes contain sums, and when a box is duplicated and opened every copy may spawn a different addend. What we need is a way to combine every multiset in the multiset associated to a box with both everything that lies outside (including all the combinations of other boxes) and also a certain number of multisets of the same box depending on how many potential copies may be done. This "combinatorial monster" can be fortunately described by an operation on multisets that is in fact a multiplication with respect to multiset sum: the *convolution product* (Definition 4). So let us first introduce this abstract machinery on multisets.

Multisets. Let X me a well-ordered monoid $(X, <, 0, +)$ with $<$ compatible with the sum, and consider $\mathcal{M}_{\text{fin}}(X)$ with additive notation. For each $A \in \mathcal{M}_{\text{fin}}(X)$ we define $\max A := \max |A|$, with the convention that $\max \emptyset = 0$, and $A[a \mapsto 0] = A - A(a)[a]$. On $\mathcal{M}_{\text{fin}}(X)$ we can define an order in one of this two equivalent forms (inductive on $\# |A|$, and as a transitive-reflexive closure).

- $A \leq B$ iff $\max A \leq \max B$, and if $\max A = \max B$ then $A(\max A) \leq B(\max A)$, and if moreover $A(\max A) = B(\max A)$ then $A[\max A \mapsto 0] \leq B[\max A \mapsto 0]$;
- $\leq$ is the transitive-reflexive closure of $<_1$, where $A <_1 B$ iff there is $b \in B$ such that $A = B - [\, b\,] + K$ where $\max K < b$.

This is a well ordering on $\mathcal{M}_{\text{fin}}(X)$, a proof of which can be found in [16]. Moreover it is compatible with multiset sum, turning $(\mathcal{M}_{\text{fin}}(X), <, [\,], +)$ into a well-ordered monoid itself.

Definition 4 *The **convolution product** of two finite multisets A and B is*

$$(A * B)(z) := \sum_{x+y=z} A(x)B(y).$$

The support of $A * B$ is $|A| + |B| = \{\, x + y \mid x \in |A|, y \in |B| \,\}$ (and is therefore finite), and in fact we can see the product as a generalization of the sum on sets, i.e. we could write $A * B = [\, x + y \mid x \in A, y \in B \,]$ where we count multiplicities. This operation is commutative, associative, has $[0]$ as unit and $[\,]$ as absorbing element, and distributes over multiset sum. A less trivial property is compatibility with multiset order.

Proposition 5 *If $U \leq V$ then $U * W \leq V * W$.*

PROOF. Excluding the trivial case $W = [\,]$, we show that if $U <_1 V$ then $U * W < V * W$, which easily gives the result. We have

$$U*W = (V_0+K)*W = V_0*W+K*W, \quad V*W = (V_0+[a])*W = V_0*W+[a]*W,$$

with $\max K < a$. It is easy to see that

$$\max(K * W) = \max(|K| + |W|) = \max K + \max W < a + \max W = \max([a] * W),$$

which together with compatibility with the sum suffices to give what was looked for. $\square$

We define the **power** of a multiset V^k by iterated convolution product. Compatibility assures us this power is monotone increasing with respect to both V and k. We will use $\prod$ for finite convolution products. As hinted above, we will apply this machinery to finite multisets of finite multisets.

Measures on wires. We define four measures on exponential wires. Two of them will depend on exponential paths going against ? ports, the other two on paths going in the other direction. Let us fix for the subsequent definitions a module π. These measures are a generalization of a technique already employed in [17] for proving strong normalization in LL proof nets.

Definition 6 (exponential path) *An* ***exponential path*** *is a finite sequence (e_i) of exponential wires at the same exponential depth such that every e_i and e_{i+1} are on the ports of a same cell C, and if e_i is on an auxiliary port of C then e_{i+1} is on the principal one and viceversa.*

Because of typing all the cells in-between wires of such a path can only be (co)contractions or boxes, and if we orient all wires in the path in the direction of the path itself, all of them have he same type, whether $!o$ or $?\imath$. We thus distinguish accordingly between **!-paths** and **?-paths** respectively. Because of acyclicity exponential paths are non repeating (so no loops are possible and the length of paths is bounded). Fixing the starting wire e_0, **maximal** exponential paths can only end on conclusions of the whole module, (co)weakenings, (co)derelictions, and moreover ⊗s for !-paths, and pars or boxes without auxiliary ports for ?-paths.

We define the **!-measures** cd (**codereliction count**) and $\ell_!(e)$ (**!-length**) by induction on the exponential depth of e and the maximum length of maximal ?-paths starting from e. We will also use $\#_!(e)$ (**!-count**) for $1 + \mathrm{cd}(e)$. The definition is given by cases depending on the second port of the wire, directed with $?\imath$ type. For every incoming $?\imath$-typed conclusion x of the whole module (not of the content of a box) let us declare *variables* on $\mathbb{N}$ named $\mathrm{cd}(x)$ and $\ell_!(x)$. Such variables, called **!-variables**, are introduced so that we may regard all these measures as depending on the context in which the module is plugged, which will supply values for them.

- If e is on a codereliction, $\mathrm{cd}(e) := 1$ and $\ell_!(e) := 1$.
- If e is on a tensor, $\mathrm{cd}(e) := 0$ and $\ell_!(e) := 0$.
- If e is on a conclusion x of the whole module, then $\mathrm{cd}(e) := \mathrm{cd}(x)$ and $\ell(e) := \ell_!(x)$.
- If e is on a coweakening, $\mathrm{cd}(e) := 0$ and $\ell_!(e) := 1$.
- If e is on a contraction with principal port f (resp. if it is the conclusion of a simple net inside a box with corresponding auxiliary port f outside), $\mathrm{cd}(e) := \mathrm{cd}(f)$ and $\ell_!(e) := \ell_!(f)$.
- If e is on a cocontraction with auxiliary ports f_i, then

$$\mathrm{cd}(e) := \textstyle\sum_i \mathrm{cd}(f_i) \quad \text{and} \quad \ell_!(e) := 1 + \max_i\Big(\ell_!(e_i)\Big).$$

- If e is on a box with auxiliary ports f_i, then

$$\mathrm{cd}(e) := \textstyle\sum_i \mathrm{cd}(f_i) \quad \text{and} \quad \ell_!(e) := 1 + \mathrm{cd}(e) = 1 + \sum_i \mathrm{cd}(f_i).$$

We define the **?-measures** $\#_?(e)$ (**?-count**) and $\ell_?(e)$ (**?-length**) by induction on the exponential codepth of e (the depth of the module minus the depth of the wire) and the maximum length of maximal !-paths starting from e. Symmetrically to the ! case, the definition is given by cases depending on the second port of the !-oriented wire, and there are variables on $\mathbb{N}$ named $\#_?(x)$ and $\ell_?(x)$ (**?-variables**) for every incoming $!o$-typed conclusion x. These are the measures also appearing in [17].

- If e is on a dereliction or a weakening then $\#_?(e) := 1$ and $\ell_?(e) := 1$.
- If e is on a par, $\#_?(e) := 1$ and $\ell_?(e) := 0$.
- If e is on a conclusion x, $\#_?(e) := \#_?(x)$ and $\ell_?(e) := \ell_?(x)$.
- If e is on a cocontraction with principal port f, $\#_?(e) := \#_?(f)$ and $\ell_?(e) := \ell_?(f)$.
- If e is on a contraction with auxiliary ports f_i, then

$$\#_?(e) := \textstyle\sum_i \#_?(f_i) \quad \text{and} \quad \ell_?(e) := 1 + \max_i\Big(\ell_?(f_i)\Big).$$

- If e is on a box with principal port is p and content $\sum_i \lambda_i$, then

$$\#_?(e) := \#_?(p)\,\#_!(p)\max_i\Big(\#_?(e^{\lambda_i})\Big),\quad \ell_?(e) := 1 + \ell_?(p) + \mathrm{cd}(p) + \max_i\Big(\ell_?(e^{\lambda_i})\Big).$$

  where e^{λ_i} is the conclusion of λ_i corresponding to e.

We finally define $\ell(e) := \ell_?(e) + \ell_!(e)$ (**length**) and $\#(e) := \#_?(e)\,\#_!(e)$ (**count**). Whenever we want to specify in which module or net the measure is taken, we put it as a superscript, as in $\ell_?^{\pi}(e)$. We also naturally extend the measure on every port, as there is a unique wire connecting it. If we plug the module in a context, and the result is a module, we can calculate the missing measures and use them in place of the variables of the plugged module.

Measures on nets. We finally define the measure $|\pi|$ of a module, which will be a finite multiset of finite multisets of natural numbers. We will usually regard such measures as **relative**, i.e. dependent on the variables assigned on its conclusions. When finally measuring a net to be reduced, we will use the **absolute** measure, i.e. the relative one evaluated on the values 1 for $\ell_!$, $\ell_?$ and $\#_?$ and 0 for cd on all its conclusions. However we will not distinguish such measures with a different notation. The measure will be defined by induction on the exponential depth of the net. Given σ the content of a box in π, $|\sigma|_\pi$ denotes the relative measure $|\sigma|$ evaluated on the !-measures of the auxiliary ports of the box (there are no $!o$ conclusions). Given a set of wires W, let $\ell(W)$ be $[\,\ell(e) \mid e \in W\,]$, i.e. the multiset of lengths over W. For a simple module λ let $C_0(\lambda)$ (resp. $\mathcal{B}_0(\lambda)$) be the set of cuts (resp. boxes) at exponential depth 0 in λ. Given a box B, we denote by $\sigma(B)$ its content and by $\#(B)$ the count $\#(p) = \#_?(p)\,\#_!(p)$ on its principal port p.

Definition 7 (measure of a module) *In case $\pi = \sum_i \lambda_i$ is a sum of simple mod-*

ules then $|\pi| := \sum_i |\lambda_i|$. The measure of a simple module λ is defined as

$$|\lambda| := \Big[\, \ell(C_0(\lambda)) \,\Big] * \prod_{B \in \mathcal{B}_0(\lambda)} |\sigma(B)|_\lambda^{\#(B)} .$$

Notice that the first factor can be furthermore factorized in $\prod_{c \in C_0(\lambda)} \Big[\, [\ell(c)] \,\Big]$, and that the measure is monotone in all the measures on wires defined above.

Intuitive idea of the measures. ℓ measures the maximum number of steps before a single cut arrives to a stop if we follow just one of the possibly many children of the reduction, and this is done symmetrically in the two directions. $\#_?$ counts the maximum number of contraction branchings that can arrive on the wire, giving the number of box copies that can be created in the reduction. cd counts the coderelictions, and appears in all the other measures because they create contractions and cocontractions on their way. Also this count gives us $\#_!$ which is the number of linear copies of a box that can be made in the worst case. The elements of $|\pi|$, which are multisets as well, measure the net as if it was unfolded and boxes were opened, and from each one a single net was chosen. Box contents are however expanded with a power operation which makes potentially coexist together a number (given by the count # on the box) of nets fetched from the box.

In the following, given a simple module λ, let $C_?(\lambda)$ and $C_!(\lambda)$ be the set of the (incoming) $?\imath$ and $!o$ typed conclusions of λ respectively. Analogously, for a context $\omega[\ ]$ let $D_?(\omega)$ and $D_!(\omega)$ be the (outgoing) $?\imath$ and $!o$ typed ports of its hole. Note that !-measures of $C_!$ depend only on !-variables declared on $C_?$, while ?-measures of $C_?$ depend (monotonously) on both ?-variables of $C_!$ and !-variables of $C_?$ (more precisely the codereliction count). For simple modules λ, μ with the same interface we say that λ **can replace** μ (written $\lambda \preccurlyeq \mu$) if for f !-measure (resp. ?-measure) and $c \in C_!(\lambda) = C_!(\mu)$ (resp. $c \in C_?(\lambda) = C_?(\mu)$) we have $f^\lambda(c) \leq f^\mu(c)$ *pointwise* (they are functions in the variables declared on conclusions). Finally, a context $\omega[\ ]$ is said to be **admissible** if there is no exponential path connecting $D_?(\omega)$ to $D_!(\omega)$. The contexts in which the reduction rules of Figure 3 are plugged have to be admissible, otherwise a cycle would be formed.

Lemma 8 (replacement) *If $\omega[\ ]$ is admissible, $\lambda \preccurlyeq \mu$, $\omega[\lambda]$ and $\omega[\mu]$ are modules, then for each $f \in \{\mathrm{cd}, \ell_!, \#_?, \ell_?\}$ and e wire of $\omega[\ ]$, $f^{\omega[\lambda]}(e) \leq f^{\omega[\mu]}(e)$ pointwise.*

PROOF. Let $C_\varepsilon := D_\varepsilon(\omega) = C_\varepsilon(\lambda) = C_\varepsilon(\mu)$ for $\varepsilon = !, ?$ (after the modules are plugged in the hole D_ε and C_ε get identified). First note that !-measures on $C_?$ do not depend on the content of the hole. The only way to have a dependency, would be for it to depend on an !-measure on $C_!$, but that would break admissibility. So for $f_! \in \{\mathrm{cd}, \ell_!\}$, and for $c_? \in C_?$, $f_!^{\omega[\lambda]}(c_?) = f_!^{\omega[\mu]}(c_?)$. Having this values we can calculate $f_!$ on $c_! \in C_!$, and by hypothesis we get $f_!^{\omega[\lambda]}(c_!) \leq f_!^{\omega[\mu]}(c_!)$. Now this values can be used to calculate inside $\omega[\ ]$

the ?-measures $f_?$ on $C_!$, as values $f_?(c_?)$ do not appear in them because of admissibility. By monotonicity of such dependency, $f_?^{\omega[\lambda]}(c_!) \leq f_?^{\omega[\mu]}(c_!)$. We can then have the missing $f_?(c_?)$ by calculating them back inside the modules λ and μ, and again by monotonicity $f_?^{\omega[\lambda]}(c_?) \leq f_?^{\omega[\mu]}(c_?)$. We can conclude by applying one last time the argument of monotonicity: for all measures f and all wires e in $\omega[\ ]$, $f(e)$ depends (apart from the conclusion variables) monotonously on the values obtained above. □

We are now ready to prove the main lemma of this long proof, after which the strong normalization theorem will be within reach. A **terminal wire** is one between a conclusion and a non-auxiliary port. When plugging a module in a context, terminal wires are the only ones that can become cuts.

Lemma 9 (modularity) *Let $\pi = \omega[\lambda]$ and $\sigma = \omega\left[\sum \mu_i\right]$ be DPNs, where ω is a context, $\mu_i \preccurlyeq \lambda$ for $i = 1, \dots, n$ are simple modules. Let T_i be the set of terminal exponential wires of μ_i which were not terminal in λ. Suppose moreover that*

- *$n = 1$ and $[\ell^{\mu_1}(T_1)] * |\mu_1| < |\lambda|$ pointwise,*
- *or we can write $|\lambda| = [u] * X$ and $|\mu_i| = [v_i] * X_i$ with $X_i \leq X$ and $\ell^{\mu_i}(T_i) + v_i < u$ pointwise for every i,*

Then $|\sigma| < |\pi|$.

PROOF. Let $\varphi[\ ]$ be the simple context with its hole at depth 0, $\psi[\ ]$ the context, $a \neq 0$ the coefficient and χ the net such that $\omega[\ ] = \psi[a\varphi[\ ] + \chi]$ and $a\varphi[\ ] + \chi$ is either the content of the smallest box containing the hole or the whole $\omega[\ ]$ if none exists. We first prove that $\left|\varphi[\sum_i \mu_i]\right| = \sum_i \left|\varphi[\mu_i]\right| < \left|\varphi[\lambda]\right|$. If $n = 0$ (a case always covered by the second possibility in the hypotheses) this result is trivial, so take $n > 0$ in the following.

We can write $\left|\varphi[\lambda]\right| = |\lambda| * Y * Z$ (resp. $\left|\varphi[\mu_i]\right| = |\mu_i| * Y_i * Z_i$) where Y (resp. Y_i) is the part (in fact a multiset singleton) due to cuts on the interface between module and context, and Z (resp. Z_i) is the part due to the context $\varphi[\ ]$ itself. By the replacement lemma and monotonicity, $Z_i \leq Z$ and $Y_i \leq [\ell^{\mu_i}(T_i)] * Y$ pointwise for all i (all cuts counting for Y_i, if they are not cuts adding to Y, have become terminal during the replacement). The replacement lemma also assures that all pointwise inequalities listed in the hypotheses survive when the modules are plugged in the context φ. In case $n = 1$ we therefore have (by hypothesis)

$$\left|\varphi[\mu_1]\right| \leq |\mu_1| * [\,\ell(T_1)\,] * Y * Z < |\lambda| * Y * Z = \varphi[\lambda].$$

Otherwise, if $n > 1$, putting it together:

$$\left|\sum_i \varphi[\mu_i]\right| \leq \sum_i \left(|\mu_i| * [\,\ell(T_i)\,] * Y * Z\right) = \sum_i \left([v_i] * X_i * [\ell(T_i)] * Y * Z\right) \leq$$

$$\leq \sum_i ([\ell(T_i) + v_i] * X * Y * Z) = \Big(\sum_i [\ell(T_i) + v_i]\Big) * X * Y * Z =$$
$$= [\ell(T_1) + v_1, \dots, \ell(T_n) + v_n] * X * Y * Z < [u] * X * Y * Z = \big|\varphi[\lambda]\big|.$$

Let's return to $\omega[\ ] = \psi[a\varphi[\ ] + \chi]$. If $\psi[\ ] = [\ ]$, that is ω's hole is not contained in a box, we have nothing else to add, as the order is compatible with sum. If otherwise $\beta[\ ]$ is the smallest box containing $a\varphi[\ ] + \chi$ seen as a (simple) module with the hole being its content and $\psi'[\ ]$ is such that $\psi[\ ] = \psi'[\beta[\ ]]$, we may note that $\psi'[\ ]$ is admissible and that $\beta\big[a\sum_i \varphi[\mu_i] + \chi\big] \preccurlyeq \beta[a\varphi[\lambda] + \chi]$. So

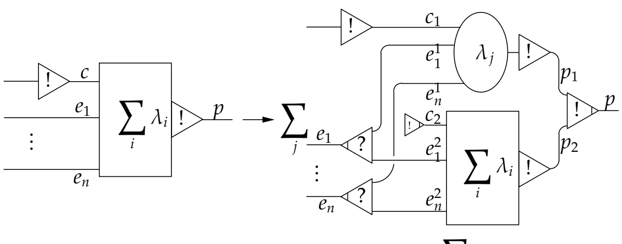
$$\Big|\psi\big[a\textstyle\sum_i \varphi[\mu_i] + \chi\big]\Big| = W' * \Big(a\textstyle\sum_i \big|\varphi[\mu_i]\big| + |\chi|\Big)^k,$$
$$\Big|\psi\big[a\varphi[\lambda] + \chi\big]\Big| = W * \Big(a\big|\varphi[\lambda]\big| + |\chi|\Big)^k$$

with k given by the product of the count # on all the boxes containing $a\varphi[\ ] + \chi$ (which does not depend on the content of the box), and $W' \leq W$ (the measures due to $\psi'[\ ]$) because of the replacement lemma. The same lemma assures us we can apply the strict pointwise comparison previously established on the measures $\sum_i \big|\varphi[\mu_i]\big|$ and $\big|\varphi[\lambda]\big|$, getting the final result. □

Theorem 10 *The reduction $\xrightarrow{\mathsf{e}}$ is strongly normalizing.*

PROOF. For each couple redex-reduct of $\xrightarrow{\mathsf{e}}$ as presented in Figure 3 we have to verify the hypotheses of the modularity lemma. In fact $\pi \xrightarrow{\mathsf{e}} \sigma$ means $\pi = \omega[\lambda]$ and $\sigma = \omega[\sum_i \mu_i]$ with $\lambda, \sum_i \mu_i$ a couple given by one of those rules and $\omega[\ ]$ an admissible context. If the modularity lemma applies, we get for absolute measures $|\sigma| < |\pi|$. By well-ordering we then have that there cannot be any infinite reduction. We will not show all of the cases, just the two most interesting (and hardest) cases.

Codereliction vs box.



π $\qquad$ $\sum_j \sigma_j$

In this case there is no new terminal wire. First we check the replacement hypothesis.

$$\mathrm{cd}^{\sigma_j}(p) = 1 + \mathrm{cd}^{\sigma_j}(p_2) = 1 + \textstyle\sum_h \mathrm{cd}^{\sigma_j}(e_h^2) = \mathrm{cd}^{\pi}(c) + \sum_h \mathrm{cd}(e_h) = \mathrm{cd}^{\pi}(p),$$
$$\ell_!^{\sigma_j}(p) = 1 + \max\Big(1, 1 + \textstyle\sum_h \mathrm{cd}(e_h)\Big) = 1 + \mathrm{cd}^{\pi}(c) + \sum_h \mathrm{cd}(e_h) = \ell_!^{\pi}(p),$$
$$\#_?^{\sigma_j}(e_h) = \#_?^{\sigma_j}(e_h^1) + \#_?^{\sigma_j}(p_2)\Big(1 + \textstyle\sum_k \mathrm{cd}(e_k)\Big)\max_i\Big(\#_?^{\sigma_j}(e_h^{2\lambda_i})\Big) \leq$$

$$
\begin{aligned}
&\leq \#_?(p)\max_i\Big(\#^\pi_?(e_h^{\lambda_i})\Big) + \#_?(p)\Big(1+\textstyle\sum_k \mathrm{cd}(e_k)\Big)\max_i\Big(\#^\pi_?(e_h^{\lambda_i})\Big) = \\
&= \#_?(p)\Big(2+\mathrm{cd}(e_k)\Big)\max_i\Big(\#^\pi_?(e_h^{\lambda_i})\Big) = \#^\pi_?(e_h),
\end{aligned}
$$

$$
\begin{aligned}
\ell^{\sigma_j}_?(e_h) &= 1+\max\Big(\ell^{\sigma_j}_?(e_h^1), 1+\max_i(\ell^{\sigma_j}_?(e_h^{2\lambda_i})) + \ell_?(p) + \textstyle\sum_k \mathrm{cd}(e_k)\Big) \leq \\
&\leq 1+\max\Big(\ell^\pi_?(e_h^{\lambda_j}), 1+\max_i(\ell^\pi_?(e_h^{\lambda_i}))\Big) + \ell_?(p) + \textstyle\sum_k \mathrm{cd}(e_k) \leq \\
&\leq 2+\max_i\Big(\ell^\pi_?(e_h^{\lambda_i})\Big) + \ell_?(p) + \textstyle\sum_k \mathrm{cd}(e_k) = \ell^\pi_?(e_h).
\end{aligned}
$$

We take the measures of the modules:

$$
|\pi| = [\,[\ell^\pi(c)]\,] * \Big(\textstyle\sum_i |\lambda_i|_\pi\Big)^{\#_?(p)\,\#_!(p)}, \qquad |\sigma_j| = [\delta_j] * |\lambda_j|_{\sigma_j} * \Big(\textstyle\sum_i |\lambda_i|_{\sigma_j}\Big)^{\#^{\sigma_j}_?(p_2)\,\#^{\sigma_j}_!(p_2)},
$$

where $\delta_j = [\ell^{\sigma_j}(c_1), \ell^{\sigma_j}(c_2)]$ if c_1 is a cut, $[\ell^{\sigma_j}(c_2)]$ otherwise. In any case, $\delta_j \leq [\ell^{\sigma_j}(c_1), \ell^{\sigma_j}(c_1)]$. First observe that the measure of the content inside the box is less in σ_j than in π as all measures on its border are the same apart from cd which is 1 less in σ_j (so $\#^{\sigma_j}_! \leq \mathrm{cd}(p)$), while the measure remains the same on the linear part λ_j. So:

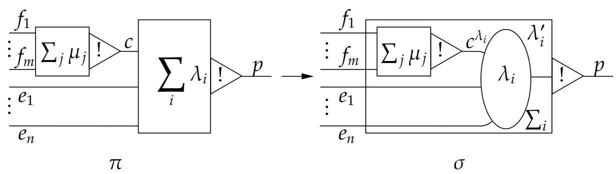
$$
|\lambda_j|_{\sigma_j} * \Big(\textstyle\sum_i |\lambda_i|_{\sigma_j}\Big)^{\#^{\sigma_j}_?(p_2)\,\#^{\sigma_j}_!(p_2)} \leq \Big(\textstyle\sum_i |\lambda_i|_\pi\Big)^{\#_?(p)} * \Big(\textstyle\sum_i |\lambda_i|_\pi\Big)^{\#_?(p)\,\mathrm{cd}(p)} = \Big(\textstyle\sum_i |\lambda_i|_\pi\Big)^{\#_?(p)\,\#_!(p)}.
$$

This settles the part $X_i \leq X$ in the hypotheses of the modularity lemma. Moreover:

$$
\begin{aligned}
\ell^{\sigma_j}(c_1) &= 1+\ell^{\sigma_j}_?(c_1) = 1+\ell^\pi_?(c^{\lambda_j}) \leq 1+\max_i\Big(\ell^\pi_?(c^{\lambda_i})\Big) < \ell^\pi_?(c) < \ell^\pi(c), \\
\ell^{\sigma_j}(c_2) &= 1+\ell^{\sigma_j}_?(c_2) = 1+\max_i\Big(\ell^{\sigma_j}_?(c_2^{\lambda_i})\Big) + \ell^{\sigma_j}_?(p_2) + \textstyle\sum_k \mathrm{cd}(e_k) < \\
&\quad < 1+\max_i\Big(\ell^\pi_?(c^{\lambda_i})\Big) + \ell^\pi_?(p) + 1 + \textstyle\sum_k \mathrm{cd}(e_k) = \ell^\pi(c),
\end{aligned}
$$

So $\delta_j \leq [\ell^{\sigma_j}(c_1), \ell^{\sigma_j}(c_2)] < [\ell^\pi(c)]$, which settles the $|C_i|_{\sigma_i} + v_i < u$ part of the hypotheses.

Box vs box.



Again there are no new terminal wires. Replacement hypothesis is satisfied:

$$
\begin{aligned}
\mathrm{cd}^\sigma(p) &= \textstyle\sum_k \mathrm{cd}(e_k) + \sum_h \mathrm{cd}(f_h) = \mathrm{cd}^\pi(p), \\
\ell^\sigma_!(p) &= 1+\mathrm{cd}^\sigma(p) = 1+\mathrm{cd}^\pi(p) = \ell^\sigma_!, \\
\#^\sigma_?(e_k) &= \#_?(p)\,\#^\sigma_!(p)\max_i\Big(\#^\sigma_?(e_k^{\lambda'_i})\Big) = \#_?(p)\,\#^\pi_!(p)\max_i\Big(\#^\pi_?(e_k^{\lambda_i})\Big) = \#^\pi_?(e_k), \\
\#^\sigma_?(f_h) &= \#^\sigma(p)\max_i\Big(\#^\sigma_?(f_h^{\lambda'_i})\Big) = \#^\pi(p)\max_i\Big(\#^\sigma_?(c^{\lambda'_i})\,\#^\sigma_!(c^{\lambda'_i})\max_j\Big(\#^\sigma_?(f_h^{\mu_j})\Big)\Big) = \\
&= \#^\pi(p)\max_i\Big(\#^\pi_?(c^{\lambda_i})\,\#^\pi_!(c)\Big)\max_j\Big(\#^\pi_?(f_h^{\mu_j})\Big) = \#^\pi(c)\max_j\Big(\#^\pi_?(f_h^{\mu_j})\Big) = \#^\pi_?(f_h),
\end{aligned}
$$

$$
\begin{aligned}
\ell^{\sigma}_{?}(e_k) &= 1 + \max_i\Big(\ell^{\sigma}_{?}(e_k^{\lambda'_i})\Big) + \ell_?(p) + \mathrm{cd}^{\sigma}(p) = \\
&= 1 + \max_i\Big(\ell^{\pi}_{?}(e_k^{\lambda_i})\Big) + \ell_?(p) + \mathrm{cd}^{\pi}(p) = \ell^{\pi}_{?}(e_k),\\
\ell^{\sigma}_{?}(f_h) &= 1 + \max_i\Big(\ell^{\sigma}_{?}(f_h^{\lambda'_i})\Big) + \ell_?(p) + \mathrm{cd}^{\sigma}(p) = \\
&= 1 + \max_i\Big(1 + \max_j\Big(\ell^{\sigma}_{?}(f_h^{\mu_j})\Big) + \ell^{\sigma}_{?}(c^{\lambda_i}) + \mathrm{cd}^{\sigma}(c^{\lambda_i})\Big) + \ell_?(p) + \mathrm{cd}^{\pi}(p) = \\
&= 1 + \max_j\Big(\ell^{\pi}_{?}(f_h^{\mu_j})\Big) + 1 + \max_i\Big(\ell^{\pi}_{?}(c^{\lambda_i})\Big) + \ell_?(p) + \mathrm{cd}^{\pi}(p) + \mathrm{cd}^{\pi}(c) = \\
&= \ell^{\pi}_{?}(f_h).
\end{aligned}
$$

Let us show $\ell^{\sigma}(c^{\lambda'_i}) < \ell^{\pi}(c)$, knowing that $\ell^{\sigma}_{!}(c^{\lambda'_i}) = \ell^{\pi}_{!}(p)$:

$$
\ell^{\sigma}_{?}(c^{\lambda'_i}) = \ell^{\pi}_{?}(c^{\lambda_i}) < 1 + \max_j\Big(\ell^{\pi}_{?}(c^{\lambda_j})\Big) + \ell_?(p) + \mathrm{cd}^{\pi}(p) = \ell^{\pi}_{?}(c).
$$

So if we let $\delta_i = [\ell^{\sigma}(c^{\lambda'_i})]$ if $c^{\lambda'_i}$ is a cut, $[\,]$ otherwise, and ε be $[\ell^{\pi}_{!}(c) + \max_j(\ell^{\pi}_{?}(c^{\lambda_j}))]$ we have $\delta_i \leq \varepsilon < [\ell^{\pi}(c)]$. Moreover $\#^{\sigma}(c^{\lambda'_i}) \leq \max_i(\#^{\pi}_{?}(c^{\lambda_i}))\,\#^{\pi}_{!}(c)$, $\#^{\sigma}(p) = \#^{\pi}(p)$, $|\lambda_i|_{\sigma} = |\lambda_i|_{\pi}$ and $|\mu_j|_{\sigma} = |\mu_j|_{\pi}$, so we get

$$
\begin{aligned}
|\sigma| &= \Big(\textstyle\sum_i |\lambda'_i|_{\sigma}\Big)^{\#^{\sigma}(p)} = \Big(\textstyle\sum_i\Big([\delta_i] * |\lambda_i|_{\sigma} * \Big(\textstyle\sum_j |\mu_j|_{\sigma}\Big)^{\#^{\sigma}(c^{\lambda'_i})}\Big)\Big)^{\#^{\sigma}(p)} \leq \\
&\leq \Big(\textstyle\sum_i\Big([\varepsilon] * |\lambda_i|_{\pi} * \Big(\textstyle\sum_j |\mu_j|_{\pi}\Big)^{\max_i(\#^{\pi}_{?}(c^{\lambda_i}))\,\#^{\pi}_{!}(c)}\Big)\Big)^{\#^{\pi}(p)} = \\
&= [\varepsilon]^{\#^{\pi}(p)} * \Big(\textstyle\sum_i |\lambda_i|_{\pi}\Big)^{\#^{\pi}(p)} * \Big(\textstyle\sum_j |\mu_j|_{\pi}\Big)^{\#^{\pi}(p)\max_i(\#^{\pi}_{?}(c^{\lambda_i}))\,\#^{\pi}_{!}(c)} = \\
&= [\#^{\pi}(p)\cdot\varepsilon] * \Big(\textstyle\sum_i |\lambda_i|_{\pi}\Big)^{\#^{\pi}(p)} * \Big(\textstyle\sum_j |\mu_j|_{\pi}\Big)^{\#^{\pi}_{?}(c)\,\#^{\pi}_{!}(c)} < \\
&< \Big[[\ell^{\pi}(c)]\Big] * \Big(\textstyle\sum_i |\lambda_i|_{\pi}\Big)^{\#^{\pi}(p)} * \Big(\textstyle\sum_j |\mu_j|_{\pi}\Big)^{\#^{\pi}(c)} = |\pi| \qquad \square
\end{aligned}
$$

Theorem 11 *The reduction $\overset{\mathtt{ea}}{\to}$ is strongly normalizing and confluent.*

PROOF. One has to check that $\overset{\mathtt{a}}{\to}$ does not increase the measure defined above, which is easy. Then one can take as measure $(|\pi|, k(\pi))$ where $k(\pi)$ simply counts all contractions and cocontractions in π. Newman's Lemma and Lemma 2 give confluence. □

We can now deal also with the $\mathtt{m}$ reduction, though working in the pure setting we clearly do not have normalization. An essay on the lemmas we use to prove confluence can be found in the introduction of [18].

Lemma 12 *If $\pi \overset{\mathtt{ea}}{\to} \sigma$ and $\pi \overset{\mathtt{m}}{\to} \tau$ there is v such that $\sigma \overset{\mathtt{m}*}{\to} v$ and $\tau \overset{\mathtt{ea}}{\to} v$.*

PROOF. $\mathtt{m}$-reductions leave $\mathtt{ea}$-redexes alone, while $\mathtt{ea}$-reductions can erase or duplicate an $\mathtt{m}$-redex, but cannot change it. So we can still perform the $\mathtt{ea}$-reduction in τ and close the diagram by performing $\mathtt{m}$-reductions on the copies of the $\mathtt{m}$-redex in σ. □

Theorem 13 *The reduction $\overset{\mathtt{mea}}{\to}$ is confluent.*

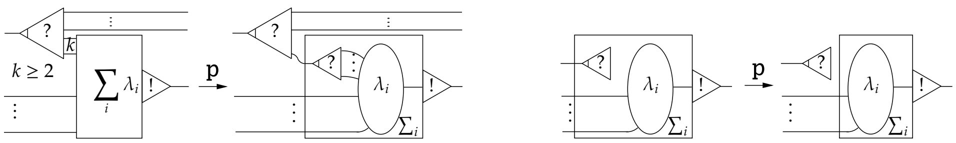


Figure 5: The push and pull rules. In the push rule $k \geq 2$ is required.

PROOF. By Huet's Lemma and the above one we get commutation of $\xrightarrow{\mathtt{ea}*}$ and $\xrightarrow{\mathtt{m}*}$. By confluence of ea and of m we finally employ Hindley-Rosen's Lemma and get the result.

3.4 Contractions, weakening and boxes: push and pull reductions

We will now fully tackle the problem with the order of identification of variables we discussed in Section 2. Already by means of a-reduction contractions made at the same exponential depth are merged and their order is forgotten. There remains to settle the order in which contractions (and weakenings) are made with respect to box borders. In an approach similar to [6], we will show that we can add two more reductions which do not ruin the properties proved in the previous section. These are the p-reductions (**push** and **pull**) presented in Figure 5. Similarly to a-reductions, if the outer contraction in the reduct of the push rule has one auxiliary port it must be considered a wire. Note how the two reductions work in opposite ways, though we cannot take any of them in the opposite direction. Pushing weakenings in boxes would be non deterministic and break confluence, pulling contractions from boxes would break strong normalization as boxes containing 0 could infinitely spawn contractions. From now on we will denote by c (for **canonical**) the combination of the a- and p-reductions. We will prove that c in itself is strongly normalizing and confluent, so we can speak of the unique **canonical form** $\mathrm{NF}^c(\pi)$ of π.

Lemma 14 *Reductions* $\xrightarrow{\mathtt{c}}$ *and* $\xrightarrow{\mathtt{ec}}$ *are locally confluent.*

PROOF. Straightforward, though long check of the new critical pairs. □

To prove strong normalization the approach used with a-reduction would fail, as the push reduction may increase the measure. We instead slightly complicate the definition of the measure in order to have one which does not increase on c, and then define another one strictly decreasing on c.

The push count. Let **generalized ?-paths** be the concatenations of ?-paths such that if ϕ and ψ get concatenated, the last wire of ϕ is the conclusion of a simple net inside a box and the first one of ψ is the wire on the corresponding auxiliary port of the box. In short words, we let generalized ?-paths "go out" of boxes. For every wire e on an auxiliary port of a box B, consider all the generalized ?-paths starting from e. For each such path E let $\mathrm{push}(E)$ be the number of contractions C along its way that have another generalized ?-path from an auxiliary port of B to an auxiliary port of C different from the one traversed by E. Write $\mathrm{push}(E)$ for such number, and define

$$\mathrm{push}(e) := \max\{\, \mathrm{push}(E) \mid E \text{ maximal gen. ?-path starting from } e \,\}.$$

Now redefine the ?-length by substituting the case for the auxiliary port of a box with

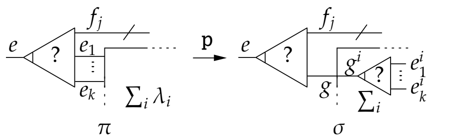
$$\ell_?(e) := 1 + \mathrm{push}(e) + \ell_?(p) + \mathrm{cd}(p) + \max_i\Big(\ell_?(e^{\lambda_i})\Big),$$

where p is the principal port of the box.

The rest of the definitions remains the same. We need to check that the push count does not increase in all e-reductions. This can be done by inspecting the reduction rules and noting how relevant generalized ?-paths persist from redexes to reducts with a lower or equal push count. Thus the measure $|\pi|$ still strictly decreases on e-reductions, as we have added a non increasing weight. Moreover we have the following result.

Lemma 15 *If $\pi \xrightarrow{\mathsf{a}} \sigma$ or $\pi \xrightarrow{\mathsf{p}} \sigma$ then $|\sigma| \leq |\pi|$.*

PROOF. After noticing that all a-reductions do not increase push counts, the only interesting case is the push reduction.



We have $\mathrm{push}^\pi(e_h) = 1 + \mathrm{push}^\sigma(g)$. Furtherly $\ell^\sigma_?(e^i_h) = \ell^\pi_?(e^{\lambda_i}_h)$ and, in case there is at least an f_j, by making maxima commute,

$$\begin{aligned}\ell^\sigma_?(e) &= 1 + \max\Big(\max_j\Big(\ell_?(f_j)\Big), 1 + \mathrm{push}^\sigma(p) + \max_i\Big(1 + \max_h(\ell^\sigma_?(e^i_h))\Big) + \ldots\Big) = \\ &= 1 + \max\Big(\max_j\Big(\ell_?(f_j)\Big), \max_h\Big(1 + \mathrm{push}^\pi(e_h) + \max_i(\ell^\pi_?(e^{\lambda_i}_h)) + \ldots\Big)\Big) = \ell^\pi_?(e),\end{aligned}$$

where the dots indicate the part about the omitted principal port. If there is no f_j then $g = e$ in σ, and $\ell_?(e)$ decreases by one. All other measures remain the same, and by monotonicity we get the result. □

Theorem 16 *The reductions $\xrightarrow{\mathsf{c}}$ and $\xrightarrow{\mathsf{ec}}$ are strongly normalizing and confluent.*

PROOF. Let $\mathrm{d}(\pi)$ be the depth of a net π, and $\mathrm{con}_0(\pi)$ and $\mathrm{coc}_0(\pi)$ be the sets of respectively contractions and cocontractions at exponential depth 0 in π. Moreover given a contraction cell C let $\mathrm{n}(C) := k$ if C is a k-contraction.

Define the multiset of natural numbers $\mathrm{p}(\pi)$ by induction on the depth of π. If π is a sum let $\mathrm{p}(\sum_i \lambda_i) := \sum_i \mathrm{p}(\lambda_i)$, if it is a simple net λ let

$$\mathrm{p}(\lambda) := \Big[\#\,\mathrm{coc}(\lambda) + \sum_{C\in\mathrm{con}_0(\lambda)} \mathrm{n}(C)3^{\mathrm{d}(\lambda)}\Big] * \prod_{B\in\mathcal{B}_0(\lambda)} \mathrm{p}\big(\sigma(B)\big).$$

Note that here the convolution product sums over $\mathbb{N}$. Finally let $\mathrm{aux}(\pi)$ be the total number of auxiliary ports of boxes in π. We now assign to each net π the measure $(|\pi|, \mathrm{p}(\pi), \mathrm{aux}(\pi))$, and show it decreases strictly for all reductions $\pi \xrightarrow{\mathsf{ec}} \sigma$. Confluence will follow from Newman's Lemma and Lemma 14. For p to decrease, it suffices that there is some simple net μ in the structure of π, in the sense that either μ is an addend of π or an addend of some box content, such that p decreases for μ, while the rest of π remains unchanged.

- If we e-reduce, then $|\sigma| < |\pi|$.
- If we a-reduce two cocontractions, then $|\sigma| \leq |\pi|$, and there is λ in π (containing the cocontractions at depth 0) and μ the corresponding simple net in σ, for which $\#\,\mathrm{coc}_0(\mu) < \#\,\mathrm{coc}_0(\lambda)$ and the rest is unchanged.
- If we a-reduce two contractions, then $|\sigma| \leq |\pi|$. If λ and μ are as above, $d = \mathrm{d}(\lambda) = \mathrm{d}(\mu)$ and the two contractions in λ are resp. n and k ones then the reduct is an $n + k - 1$-contraction (if any), and we have

$$\sum_{C\in\mathrm{con}_0(\mu)} \mathrm{n}(C)3^d = (n + k - 1)3^d + \ldots < n3^d + k3^d + \ldots = \sum_{C\in\mathrm{con}_0(\lambda)} \mathrm{n}(C)3^d$$

  while the rest is unchanged. The degenerate case $n + k - 1 = 1$ is trivial.
- If we p-reduce a push redex, then $|\sigma| \leq |\pi|$. If λ and μ are as above, D is the box of the redex, $\sum_i \lambda_i$ (resp. $\sum_i \mu_i$) is the content of D in λ (resp. in μ), $d + 1$ is the depth of λ and μ (all addends of D have $\leq d$), the contraction is an $n + k$ one with $k \geq 2$, then in μ the contraction left out (if any) is an $n + 1$-one and all addends in D get a pushed k-contraction. Summing up:

$$\begin{aligned}\mathrm{p}(\mu) &= [\ldots + (n+1)3^{d+1}\,] * \big(\textstyle\sum_i \mathrm{p}(\mu_i)\big) * \ldots = \\ &= [\ldots] * \Big(\textstyle\sum_i \big([\,(n+1)n3^{d+1}\,] * [\,k3^{\mathrm{d}(\mu_i)} + \ldots] * \ldots\big)\Big) * \ldots \leq \\ &\leq [\ldots] * \Big(\textstyle\sum_i \big([\,(n+1)3^{d+1} + k3^d\,] * \ldots\big)\Big) * \ldots\end{aligned}$$

  As $k \geq 2 > \frac{2}{3}$, $(n + 1)3^{d+1} + k3^d = (3n + 3 + k)3^d < (3n + 3k)3^d = (n + k)3^{d+1}$, we can continue the above chain of inequalities by

$$\begin{aligned}\mathrm{p}(\mu) &< [\ldots] * \Big(\textstyle\sum_i \big([\,(n+k)3^{d+1}\,] * \ldots\big)\Big) * \ldots = \\ &= [\ldots + (n+k)3^{d+1}\,] * \big(\textstyle\sum_i \mathrm{p}(\lambda_i)\big) * \ldots = \mathrm{p}(\lambda)\end{aligned}$$

- If we p-reduce a pull redex, then $|\sigma| \leq |\pi|$, and also $\mathrm{p}(\sigma) = \mathrm{p}(\pi)$, but $\mathrm{aux}(\sigma) < \mathrm{aux}(\pi)$. □

We can finally infer confluence of `mec` in the same way as we have done for `mea` (Theorem 13).

Theorem 17 *The reduction* $\overset{\texttt{mec}}{\to}$ *is confluent.*

4 Full resource calculus

In this section we will redefine Boudol's λ-calculus with resources [1] extending it with sums and two kinds of non lazy reduction. As nets presented in the previous section added promotion to DINs of [10], this will add infinitely available resources to the resource calculus described in the same paper and presented in Section 2, thus we call it the **full resource calculus**.

4.1 Statics: λ-calculus with resources

Let $\mathbb{V}$ be a countable set of variables, and let Δ_k be the increasing sequence of sets given by induction as $\Delta_0 := \mathbb{V}$, and Δ_{k+1} generated by the following grammar:

$$\Delta_{k+1} ::= \Delta_k \mid \lambda\mathbb{V}.\Delta_k \mid \langle\Delta_k\rangle\, \Delta^!_k.$$

$\Delta^!_k$, the k^{th} set of **bags of arguments**, is $\mathcal{M}_{\text{fin}}(\mathbb{A}_k)$, where furthermore $\mathbb{A}_k$, the k^{th} set of **arguments**, is generated by

$$\mathbb{A}_k ::= \Delta_k \mid (R\,\langle\Delta_k\rangle)^\infty.$$

Finally, the set Δ of **simple terms** and the set $\Delta^!$ of bags are $\Delta := \bigcup_{k\in\mathbb{N}} \Delta_k$ and $\Delta^! := \bigcup_{k\in\mathbb{N}} \Delta^!_k$. A **differential term**, or simply term, is an element of $R\,\langle\Delta\rangle$. We will also deal with $R\langle\Delta^!\rangle$, called **differential bags**. An argument of the form $(\sum_{t\in\Delta} c_t \cdot t)^\infty$ is called **boxed** or **exponential**, otherwise it is **linear**. Bags are multisets presented in multiplicative notation, and the above constructors are extended by multilinearity, all but the one for boxed argument. Given a bag A, its **linear part** $\mathcal{L}(A)$ (resp. **boxed** or **exponential part** $\mathcal{E}(A)$) is the multiset of its linear (resp. exponential) arguments. As usual terms are identical up to α-conversion. We write $x \in t$ to mean "x appearing free in t" for t term [11] . A **context** is a differential term or bag that uses a distinguished variable called its **hole** exactly once, similarly to what was done for nets on page 9 [12] . Classical terms of λ-calculus can be embedded in this calcu-

[11] With sums one should be more accurate with the definition, however with $R = \mathbb{N}$ many difficulties are set aside. Readers may refer to [22] for a more in-depth treatment of the subject.

[12] Again for example $[\,] + [\,] = 2[\,]$ is considered a one-hole context, though again if $R = \mathbb{N}$ we can safely rule out such possibilities and retain all the results.

lus by mapping arguments to a singleton bag with a corresponding boxed argument.

4.2 Dynamics: giant-step and baby-step β-reduction

Substitution $s\,[x := t]$ with $s, t \in R\,\langle\Delta\rangle$ is defined as usual, possibly applying the generalizations of constructors by multilinearity. **Linear substitution** $\frac{\partial}{\partial x}$ generalizes the one given in Section 2. Inductive rules are:

$$\frac{\partial y}{\partial x}\cdot t := \delta_{x,y}\cdot t, \qquad \frac{\partial \lambda y.u}{\partial x}\cdot t := \lambda y.\frac{\partial u}{\partial x}\cdot t \quad \text{with } y \notin t,$$

$$\frac{\partial \langle r\rangle\, A}{\partial x}\cdot t := \left\langle \frac{\partial r}{\partial x}\cdot t\right\rangle A + \langle r\rangle\,\frac{\partial A}{\partial x}\cdot t,$$

$$\frac{\partial A}{\partial x}\cdot t := \sum_{a\in A}\left(\frac{\partial a}{\partial x}\cdot t\right)A/a, \qquad \frac{\partial u^\infty}{\partial x}\cdot t := \left(\frac{\partial u}{\partial x}\cdot t\right)u^\infty.$$

The definition for applications and bags can be compacted into

$$\frac{\partial \langle r\rangle\, A}{\partial x}\cdot t = \left\langle \frac{\partial r}{\partial x}\cdot t\right\rangle A + \sum_{u\in\mathcal{L}(A)}\langle r\rangle\left(\frac{\partial u}{\partial x}\cdot t\right)A/u + \sum_{v^\infty\in\mathcal{E}(A)}\langle r\rangle\left(\frac{\partial v}{\partial x}\cdot t\right)A.$$

Note how the linear substitution operator distributes among linear terms, and extracts a linear copy from a boxed argument if needed [13] . This substitution is linear in both u and t, and if $x \notin u$ then $\frac{\partial u}{\partial x}\cdot t = 0$.

Non linear and linear substitutions enjoy the same properties found in [9]. In order to generalize them and define reduction, we employ one more substitution directly based on the regular one: the **partial substitution** of u for x in t is simply $t\,[x := x + u]$. Finally, in order to unify the notation, let the **generalized substitution** of a for x in t, with $a = u$ or $a = u^\infty$ an argument, be

$$\mathrm{S}_x\, t\cdot u := \frac{\partial t}{\partial x}\cdot u, \qquad \mathrm{S}_x\, t\cdot u^\infty := t\,[x := x + u]\,.$$

Using partial substitution instead of the regular one allows us to state the following generalized Schwartz's lemma.

Lemma 18 *For $t \in R\,\langle\Delta\rangle$, a, b arguments and x, y such that $y \notin a$ and $x \notin b$, we have* $\mathrm{S}_x(\mathrm{S}_y\, t\cdot b)\cdot a = \mathrm{S}_y(\mathrm{S}_x\, t\cdot a)\cdot b$.

PROOF. There are three combinations to check (partial-partial, linear-linear and linear-partial). The first one is trivial. As opposed to regular substitu-

[13] This reflects the derivation property of the exponential in calculus. Given $y = y(x)$ we have $\frac{\partial e^y}{\partial x} = \frac{\partial y}{\partial x}\cdot e^y$.

tion, we can also have $x = y$ ($a = u^\infty, b = v^\infty$):

$$t\,[x := x + v]\,[x := x + u] = t\,[x := x + u + v] = t\,[x := x + u]\,[x := x + v]\,.$$

The second is not much different from the proof of Schwartz's lemma in differential λ-calculus [9]. The third is by induction, where the inductive steps are trivial while the base case for variables is ($a = u, b = v^\infty$):

$$\frac{\partial (z\,[y := y + v])}{\partial x} \cdot u = \frac{\partial (z + \delta_{y,z} \cdot v)}{\partial x} \cdot u = \frac{\partial z}{\partial x} \cdot u = \delta_{x,z} \cdot u = \Big(\frac{\partial z}{\partial x} \cdot u\Big)\,[y := y + v]$$

where from $x \notin v, y \notin u$ we infer $\frac{\partial v}{\partial x} \cdot u = 0$ and $u = u\,[y := y + v]$. $\square$

If $A = [u_1, \ldots, u_n]$ is a bag of linear arguments such that $x \notin u_i$ we write

$$\frac{\partial^n t}{\partial x^n} \cdot A := \frac{\partial}{\partial x}\Big(\cdots\Big(\frac{\partial t}{\partial x} \cdot u_1\Big)\cdots\Big) \cdot u_n$$

which by the above lemma is well defined. More generally, given any bag $A = [a_1 \cdots a_{\# A}]$ and a variable $x \notin a_i$, we can define

$$\mathrm{S}_x^{\# A}\, t \cdot A := \mathrm{S}_x\,(\cdots(\mathrm{S}_x\, t \cdot a_1)\cdots) \cdot a_{\# A} = \Big(\frac{\partial^{\#\mathcal{L}(A)} t}{\partial x^{\#\mathcal{L}(A)}} \cdot \mathcal{L}(A)\Big)\Big[x := x + \sum_{u^\infty \in \mathcal{E}(A)} u\Big].$$

We are ready to define the reductions, which as foretold in Section 2 come in baby-step and giant-step form. Baby-step is more local and natural, and more close to the reduction defined for Boudol's calculus. However giant-step, which empties a bag altogether, is the reduction whose bisimulation result reflects the one for λ-calculus and proof nets.

Definition 19 (β_{gs} and β_{bs}) ***Giant-step β-reduction*** *(β_{gs} or $\xrightarrow{\mathsf{g}}$) is generated by*

$$\langle \lambda x.s\rangle\, A \xrightarrow{\mathsf{g}} \mathrm{S}_x^{\# A}\, s \cdot A\,[x := 0] = \Big(\frac{\partial^{\#\mathcal{L}(A)} s}{\partial x^{\#\mathcal{L}(A)}} \cdot \mathcal{L}(A)\Big)\Big[x := \sum_{u^\infty \in \mathcal{E}(A)} u\Big].$$

Baby-step β-reduction *(β_{bs} or $\xrightarrow{\mathsf{b}}$) is generated by*

$$\langle \lambda x.s\rangle\, aA \xrightarrow{\mathsf{b}} \langle \lambda x.\, \mathrm{S}_x\, s \cdot a\rangle\, A, \qquad \langle \lambda x.s\rangle\, 1 \xrightarrow{\mathsf{b}} s\,[x := 0]\,.$$

Partial substitutions break strong confluence, so we need more care in proving confluence. Later we will infer it for β_{gs} (Corollary 28). We here derive from it confluence of β_{bs}.

Lemma 20 *If $u\ \beta^*_{bs}\ v$ then there exist a term w such that $u, v\ \beta^*_{gs}\ w$.*

PROOF. By induction on the length of the reduction $u\ \beta^*_{\mathrm{bs}}\ v$. If it is zero, then take $w = u = v$ and we are done. Otherwise we have the following confluence diagram

$$\begin{tikzcd}
 & v' \arrow[r, "\mathrm{b}"] \arrow[dd, dashed, "\mathrm{g}*"'] \arrow[dr, dashed, "\mathrm{g}"'] \arrow[dr, phantom, shift left=3, "\text{(II)}"] & v \arrow[d, dashed, "\mathrm{g}*"] \\
u \arrow[ur, "\mathrm{b}*"] \arrow[dr, dashed, "\mathrm{g}*"'] \arrow[r, phantom, "\text{(I)}"] & \arrow[r, phantom, "\text{(III)}"] & w'' \arrow[d, dashed, "\mathrm{g}*"] \\
 & w' \arrow[r, dashed, "\mathrm{g}*"'] & w
\end{tikzcd}$$

We have (I) by inductive hypothesis, (II) is clear from the definition, as β_{gs}-reducing a redex before or after a single step of β_{bs} on the same redex is the same (we may have to β_{gs}-reduce in all addends possibly arisen), and (III) is confluence of β_{gs}. □

Theorem 21 *The baby-step β-reduction is confluent.*

PROOF. Suppose $u \; \beta^*_{\mathrm{bs}} \; v_1, v_2$. We get the following confluence diagram:

$$\begin{tikzcd}
 & v_1 \arrow[r, dashed, "\mathrm{g}*"] & w_1 \arrow[dr, dashed, "\mathrm{g}*"] & \\
u \arrow[ur, "\mathrm{b}*"] \arrow[dr, "\mathrm{b}*"'] \arrow[urr, dashed, "\mathrm{g}*"'] \arrow[drr, dashed, "\mathrm{g}*"] & & & s \\
 & v_2 \arrow[r, dashed, "\mathrm{g}*"'] & w_2 \arrow[ur, dashed, "\mathrm{g}*"'] &
\end{tikzcd}$$

The left triangles are from the above lemma, while the right square is simply confluence of β_{gs}. As β^*_{gs} is contained in β^*_{bs}, we get the result. □

5 Translation

We will now define the translation from terms and bags of full resource calculus to differential proof nets. In order to do so, we use **labelled nets**, i.e. correct nets with labels in $\mathbb{V}$ on the $?\iota$ conclusions. We draw all nets with the o/$!o$ conclusion right and the rest left, so types will be omitted. A wire with a bar on it stands for multiple wires (possibly none), and its label is the corresponding set of labels. In order to be able to erase or add dummy variables at will, nets are considered equal if they differ only for conclusions introduced by weakenings.

5.1 Statics: definition and sequentialization

Using the rules in Figure 6 for each t term (resp. bag or argument) we define t°, a labelled net with conclusions $?\iota, \ldots, ?\iota, o$ (resp. $!o$) where labels contain the free variables in t. The fact that t° is indeed correct is straightforward. Adding freely weakened conclusions is used in the definition. It is important to note that the translation is well defined with respect to equality modulo weakened conclusions because of pull reductions performed on boxes.

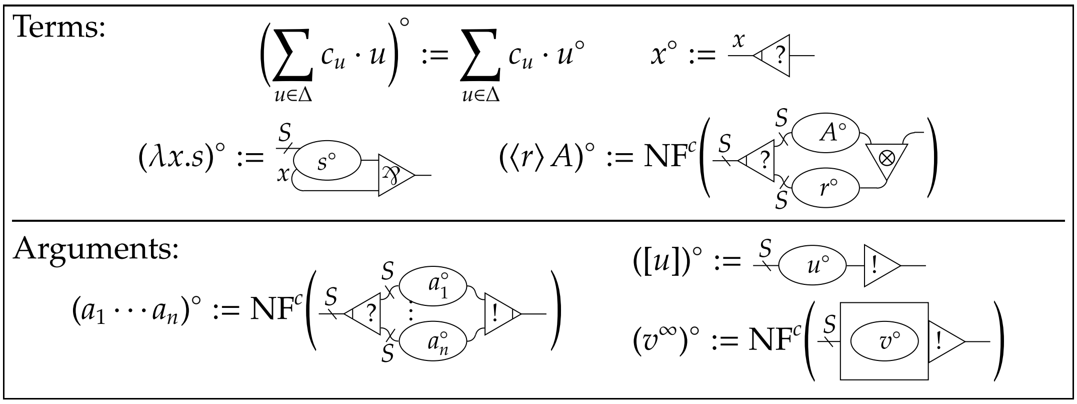


Figure 6: Inductive rules for the definition of t°. To remedy the lack of an explicit constructor, $[u]$ denotes a linear argument.

Remark 22 *For every term t its translation t° is* ec*-normal. Moreover each redex in t corresponds exactly to an* m*-redex in t°. So in fact t is normal iff t° is normal.*

Theorem 23 (sequentialization of ec-normal nets) *For every* ec*-normal and labelled net π with no exponential axiom* [14] *and no $\imath$ conclusion there is uniquely either a term t or a bag A such that $t^\circ = \pi$ (resp. $A^\circ = \pi$), modulo weakened conclusions.*

PROOF (sketch). One first takes a principal switching (see page 10) of every simple net in the net. We can erase all weakenings, which because of p-normality form each a connected component by themselves. The remaining connected component is a tree (it is acyclic) for which we choose the unique (Lemma 1) o/$!o$ conclusion as root. It is then easy to convert it to the syntactical tree of a term if the root is o, or of a bag if it is $!o$, by inductively doing the same for each box. The condition on exponential axioms assures that wires above the exponential port of a tensor, eventually forked by a single cocontraction, must end in coderelictions or boxes, i.e. linear or exponential arguments. Injectivity also depends on Lemma 1: we may compare two translations going up from the unique o/$!o$ conclusion. □

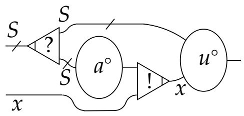
5.2 Dynamics: bisimulation

We want to show that reductions in the two systems are strongly linked by this translation. This is done in two steps, showing the two directions of bisimulation. First we have to state a substitution lemma.

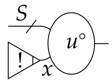
Lemma 24 (argument and 0 substitution) *Given an argument a, a simple term*

[14] Though this property is not stable under reduction (contraction vs cocontraction creates exponential axioms), one can prove all e-normal mec-reducts enjoy it.

or bag u and a variable $x \notin a$, we have that



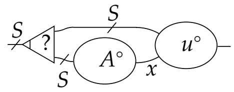
$$\overset{\mathtt{ec}}{\twoheadrightarrow} (\mathsf{S}_x(u) \cdot a)^\circ \quad \textit{and}$$



$$\overset{\mathtt{ec}}{\twoheadrightarrow} (u\,[x := 0])^\circ\,.$$

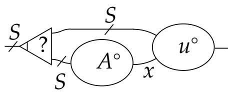
PROOF (sketch). It is an easy induction on u, which generalizes what can be already found in the literature. As in [6] the p-reduction is fundamental in the inductive step of boxed terms, to make trailing contractions enter the box. It can be noticed that this lemma implements the intuitions about the cells of differential nets as given in [10].

Lemma 25 (substitution) *If A is a bag of arguments and u is a simple term, then*

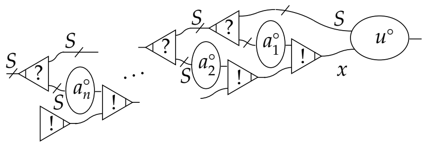


$$\overset{\mathtt{ec}}{\twoheadrightarrow} \left(\mathsf{S}_x^{\#A}\, u \cdot A\right)[x := 0]\,.$$

PROOF. If $A = a_1 \cdots a_n$ then, by expanding the cocontraction at the base of A° and the contractions on its variables, we have that



$$\equiv_{\mathsf{a}}$$



.

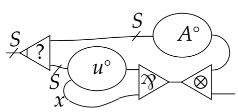
By a repeated application of Lemma 24 the above net gives as an ec-normal form $\left(S_x^{\#A} s \cdot A\,[x := 0]\right)^\circ$. Having used a-equivalence does not change the ec-normal form because of confluence. □

Note how the reduction on nets involved in the next theorem has a particular shape, so that even if the result is a logical equivalence it is not yet full bisimulation, which is truly achieved by the one after it.

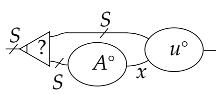
Theorem 26 (giant-step simulation) *$s\ \beta_{gs}\ t$ iff $s^\circ \overset{\mathtt{m}}{\to}\overset{\mathtt{ec}}{\twoheadrightarrow} t^\circ$.*

PROOF. First the only if part. Given a redex $\langle \lambda x.s\rangle\, A$, we have

$$(\langle \lambda x.s\rangle\, A)^\circ =$$



$$\overset{\mathtt{m}}{\to}$$



that because of the substitution lemma gives $(S_x^{\#A} s \cdot A\,[x := 0])^\circ$. Vice versa take any reduction $R : s^\circ \overset{\mathtt{m}}{\to} \pi \overset{\mathtt{ec}}{\twoheadrightarrow} t^\circ$, then let $s\ \beta_{\mathrm{gs}}\ r$ be the result of firing the redex corresponding to the multiplicative cut fired at the beginning of R. Because of the only if part, $s^\circ \overset{\mathtt{m}}{\to} \pi \overset{\mathtt{ec}}{\twoheadrightarrow} r^\circ$ (note π is the same as before). By uniqueness of normal form and injectivity of the translation we have $t = r$. □

Theorem 27 (giant-step bisimulation) *If $s^\circ \xrightarrow{\mathtt{mec}*} t^\circ$, then $s\ \beta^*_{gs}\ t$.*

PROOF. Let us first restrict the hypothesis and prove that if $u^\circ \overset{\mathtt{m}*}{\to}\overset{\mathtt{ec}}{\twoheadrightarrow} v^\circ$ then $u\ \beta^*_{\mathrm{gs}}\ v$. Let M be the sequence of multiplicative reductions in the reduction $u^\circ \overset{\mathtt{m}*}{\to} \pi \overset{\mathtt{ec}}{\twoheadrightarrow} v^\circ$, and let us reason by structural induction on u. If u is a variable there is no redex and the result is trivial. Also the abstraction case is easy, as all reductions in the net cannot touch the terminal ⅋-cell. Take therefore the case of an application $u = \langle r\rangle\, a_1 \cdots a_n$. If there is a reduction μ in M erasing the external tensor then it cannot create new m-redexes and we can safely shift it at the end of M. Let M' be M without μ if it exists, M itself otherwise. All reductions in M' happen either in r° or in either of the a_i°s (we need an a-conversion to really speak of such subnets, which however commutes with M'). We can therefore partition M' into $L : r^\circ \overset{\mathtt{m}*}{\to} \sigma$ and $N_i : a_i^\circ \overset{\mathtt{m}*}{\to} \tau_i$, and we can freely commute reductions which happen in different subnets. By ec-normalizing the results of all but μ, we get (via the sequentialization theorem) $r^\circ \overset{L}{\to}\overset{\mathtt{e}}{\twoheadrightarrow} q^\circ$ and $a_i^\circ \overset{N_i}{\to}\overset{\mathtt{e}}{\twoheadrightarrow} b_i^\circ$, where by inductive hypothesis $r\ \beta^*_{\mathrm{gs}}\ q$ and $a_i\ \beta^*_{\mathrm{gs}}\ b_i$. If μ is present then $q = \lambda x.w$ and we get by simulation $(\langle q\rangle\, B)^\circ \overset{\mu}{\to}\overset{\mathtt{ec}}{\twoheadrightarrow} (\mathrm{S}^n_x\, w \cdot B\,[x := 0])^\circ$, where $B = b_1 \cdots b_n$. Summing up, by applying N and L_i on the whole $(\langle r\rangle\, a_1 \cdots a_n)^\circ$ and commuting them and (possibly) μ back into their place in M, we get the same reduction chain we have started with. By uniqueness of ec-normal form and injectivity of translation, we get that either $v = \langle q\rangle\, B$ (if μ is not present) or $v = \mathrm{S}^n_x\, w \cdot B\,[x := 0]$ otherwise, and in both cases u reduces to it.

Let us proceed with the complete theorem. If $s^\circ = \pi_0 \overset{\mathtt{mec}}{\to} \pi_1 \overset{\mathtt{mec}}{\to} \ldots \overset{\mathtt{mec}}{\to} \pi_n = t^\circ$ is the reduction taken into account, let $s_i^\circ := \mathrm{NF}^{ec}(\pi_i)$ (existing by sequentialization), with $s_0 = s$ and $s_n = t$. Proving that $s_i^\circ \overset{\mathtt{m}*}{\to}\overset{\mathtt{ec}}{\twoheadrightarrow} s_{i+1}^\circ$ implies $s_i\ \beta^*_{\mathrm{gs}}\ s_{i+1}$ (as seen above), which ends the proof. If $\pi_i \overset{\mathtt{ec}}{\to} \pi_{i+1}$, then $\mathrm{NF}^{ec}(\pi_i) = \mathrm{NF}^{ec}(\pi_{i+1})$ and thus $s_i = s_{i+1}$. If $\pi_i \overset{\mathtt{m}}{\to} \pi_{i+1}$ we compose the following reduction diagram:

$$\begin{tikzcd}
\pi_i \arrow[r, "\mathtt{m}"] \arrow[d, twoheadrightarrow, "\mathtt{ec}"'] & \pi_{i+1} \arrow[d, dashed, "\mathtt{ec}*"'] \arrow[dr, twoheadrightarrow, "\mathtt{ec}"] & \\
s_i^\circ \arrow[r, dashed, "\mathtt{m}*"'] & \sigma_i \arrow[r, dashed, twoheadrightarrow, "\mathtt{ec}"'] & s_{i+1}^\circ
\end{tikzcd}$$

The left square is Lemma 12 for ec-reduction, while the right triangle is confluence to the ec-normal form. □

Corollary 28 *The reduction β_{gs} on terms is confluent.*

PROOF. Take $s\ \beta^*_{\mathrm{gs}}\ u, v$. By simulation $s^\circ \xrightarrow{\mathtt{mec}*} u^\circ, v^\circ$, so that by confluence of the mec reduction we further get $u^\circ, v^\circ \xrightarrow{\mathtt{mec}*} \pi$. If we take $t^\circ = \mathrm{NF}^{ec}(\pi)$, we have $u^\circ, v^\circ \xrightarrow{\mathtt{mec}*} t^\circ$ which by bisimulation gives $u, v\ \beta^*_{\mathrm{gs}}\ t$. □

Acknowledgements

The author would like to thank the Roman “lambdadiff” group, and above all its promoter Stefano Guerrini for his insight and motivation.

References

[1] G. Boudol. The lambda-calculus with multiplicities. Preliminary report, INRIA-Sophia Antipolis, 1993.

[2] G. Boudol, P.-L. Curien, and C. Lavatelli. A semantics for lambda calculi with resources. *Math. Structures Comput. Sci.*, 9(4):437–482, 1999.

[3] V. Danos. *La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul)*. Thèse de doctorat, Univ. Paris VII, 1990.

[4] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.

[5] R. Di Cosmo and S. Guerrini. Strong normalization of proof nets modulo structural congruences. In *Rewriting Techniques and Applications*, pages 75–89, 1999.

[6] R. Di Cosmo and D. Kesner. Strong normalization of explicit substitutions via cut elimination in proof nets. In *Logic In Comp. Science*, pages 35–46, 1997.

[7] T. Ehrhard. On köthe sequence spaces and linear logic. *Math. Structures Comput. Sci.*, 12:579–623, 2002.

[8] T. Ehrhard. Finiteness spaces. *Math. Structures Comput. Sci.*, 15(4):615–646, 2005.

[9] T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theor. Comput. Sci.*, 309(1):1–41, 2003.

[10] T. Ehrhard and L. Regnier. Differential interaction nets. *Theor. Comput. Sci.*, 364(2):166–195, 2006.

[11] T. Ehrhard and L. Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. To appear on *Theor. Comput. Sci.*, 2006.

[12] J.-Y. Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

[13] A. J. Kfoury. A linearization of the lambda-calculus and consequences. *Journal of Logic and Computation*, 10(3):411–436, 2000.

[14] Y. Lafont. From proof nets to interaction nets. In *Advances in Linear Logic*, volume 222 of *London Math. Soc. Lecture Note Ser.*, pages 225–247, 1995.

[15] O. Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Theor. Comput. Sci.*, 290(1):161–188, Jan. 2003.

[16] T. Nipkow. An inductive proof of the wellfoundedness of the multiset order, 1998. Available at `http://www4.informatik.tu-muenchen.de/~nipkow`.

[17] M. Pagani and L. Tortora de Falco. Strong normalization property for second order linear logic. Submitted to Theor. Comput. Sci., 2008.

[18] W. Py. *Confluence en $\lambda\mu$-calcul*. PhD thesis, Université de Savoie, July 1998. Available at `http://www.lama.univ-savoie.fr/~david/ftp/py.pdf`.

[19] L. Regnier. *Lambda-Calcul et Réseaux*. Thèse de doctorat, Univ. Paris VII, 1992.

[20] L. Regnier. Une équivalence sur les lambda-termes. *Theor. Comput. Sci.*, 126:281–292, 1994.

[21] L. Vaux. The differential $\lambda\mu$-calculus. *Theor. Comput. Sci.*, 379(1-2):166–209, 2007.

[22] L. Vaux. On linear combinations of λ-terms. In *Rewriting Techniques and Applications*, pages 374–388, 2007.