# Intuitionistic Differential Nets and Resource Lambda-Calculus

Paolo Tranquilli

*Dipartimento di Matematica – Università Roma Tre*
*Largo S.Leonardo Murialdo 1 – 00146 Roma – Italy*

**Abstract**

We define pure intuitionistic differential 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. Though interesting and independent on their own, these results are directed and adjusted to give a translation of Boudol's untyped $\lambda$-calculus with multiplicities extended with a linear-non linear reduction à la Ehrhrad and Regnier's differential $\lambda$-calculus. Such reduction comes in two flavours: baby-step and giant-step $\beta$-reduction. The translation, based on Girard's encoding $A \to B \sim\ !A \multimap B$ and as such extending the usual one for $\lambda$-calculus and proof nets, is in a sense injective and surjective and enjoys bisimulation for giant-step $\beta$-reduction, a result from which we also derive confluence of both the reductions.

*Key words:* Lambda-calculus, differential interaction nets, linear logic, proof nets

## 1 Introduction

Twenty years ago Jean-Yves Girard introduced Linear Logic (LL, [13]) starting from a fine analysis of the coherent semantics he had introduced for system F. This system has brought a new looking glass for the study of the essence of computation in general, and $\lambda$-calculus specifically. Particularly important for the background of this paper is the translation of pure and typed $\lambda$-calculus into Girard's proof nets, as studied by Danos and Regnier in their theses [5,23]. It has proved to be a powerful tool to bring forth the study of both sides of the mapping, proof nets on one side and $\lambda$-calculus

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

on the other. Examples can be seen in the analysis of a new operational equivalence [24], localization of $\beta$-reduction [7], or optimal reduction [14].

Recently Ehrhard has refined the coherent semantics by means of topological vector spaces and continuous linear maps [9,10], and again from such semantical refinement the same author and Regnier presented extensions with syntactic differential operators for both Linear Logic [12] and $\lambda$-calculus [11]. The key ingredients are a mathematical understanding of the actors involved, and the idea that linearity of differentiation along a certain direction is linked with the computational and logical meaning of linearity, which is using an argument or a hypothesis exactly once.

As such, the treatment of the subject can rely also on a line of research already present in $\lambda$-calculus. Starting from Boudol's work on $\lambda$-calculus with multiplicities [1] and going on in [2,15], variants of $\lambda$-calculus where studied where arguments could have a limited availability. In [12] 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*. We bring further this link, after introducing promotion in differential nets. Then the calculus that translates into them as $\lambda$-calculus does in proof nets turns out to be an extended version of full Boudol's $\lambda$-calculus with resources. Extended not in the constructs, but in the reduction rule, as non-deterministic lazy head reduction employed by Boudol is replaced by a non lazy one which moreover interprets non-determinism by means of a formal sums, in the style of Ehrhard and Regnier's differential $\lambda$-calculus.

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.

### *1.1 Notation*

We will denote sets of reduction rules with letters as $m$ or $e$, and by $\overset{e}{\to}$ ($e$-reduction) the corresponding relation. The relations $\overset{r=}{\to}, \overset{r+}{\to}, \overset{r*}{\to}$ and $\equiv_r$ are respectively the reflexive, transitive, reflexive-transitive and reflexive-transitive-symmetric closures of $\overset{r}{\to}$. Reduction $\overset{rs}{\to}$ is the union of reductions $\overset{r}{\to}$ and $\overset{s}{\to}$. A reduction $\overset{r}{\to}$ is

- strongly confluent if whenever $s \overset{r}{\to} u, v$ there is $w$ such that $u, v \overset{r}{\to} w$;
- confluent if $r^*$ is strongly confluent;
- locally confluent if whenever $s \overset{r}{\to} u, v$ there is $w$ such that $u, v \overset{r*}{\to} w$.

An element $u$ is $r$-normal if there is no $v$ with $u \xrightarrow{r} v$. We write $u \overset{r}{\twoheadrightarrow} v$ if $u \xrightarrow{e*} v$ and $v$ is $r$-normal. Whenever possible (there is always at least a reduction to an $r$-normal form and $r$ is confluent) we define $\mathrm{NF}^r(u)$ as the unique $r$-normal element such that $u \overset{r}{\twoheadrightarrow} \mathrm{NF}^r(u)$. $R : u \xrightarrow{r*} v$ or $u \xrightarrow{R} v$ denotes a given chain $R$ of reduction steps from $u$ to $v$. $|R|$ denotes the length of $R$, and $\xrightarrow{r}$ is strongly normalizing if all $R : u \xrightarrow{r} v$ are of finite length.

We denote by $\mathcal{M}_{\mathrm{fin}}(A)$ the set of finite multisets over $A$, equivalently seen as functions $A \to \mathbb{N}$ with finite support (which is denoted by $|A|$). Depending on where multisets are used, we will use either the additive (multiset union is $A + B$) or multiplicative (multiset union is $AB$) notation. In any case the notation $\sum_{a \in A} D_a$ stands for a sum with multiplicities, i.e. $\sum_{a \in |A|} A(a) \cdot D_a$. For example the cardinality $\#A$ of a multiset $A$ can be written as $\sum_{a \in A} 1$.

$R$ will be a commutative semiring with unit, and $R\langle S\rangle$ is the $R$-module generated by $S$, which is the set of formal finite sums over $S$ with coefficients in $R$, so that a generic element of $R\langle S\rangle$ is written as $\sum_{s \in S} c_s s$ with $c_s \in R$. Every time we write so, we will imply that the sum is finite, i.e. $\#\{s \in S \mid c_s \neq 0\} < +\infty$. We will usually have $R = \mathbb{N}$, and in such a case $\mathbb{N}\langle S\rangle$ is in fact $\mathcal{M}_{\mathrm{fin}}(S)$, and each sum can be written without coefficients, as for $U \in \mathbb{N}\langle S\rangle = \mathcal{M}_{\mathrm{fin}}(S)$ we can write $U = \sum_{u \in U} u$, counting the multiplicities as explained above.

## 2 State of the Art

Our starting point is the pairing between resource calculus and Ehrhard and Regnier's differential interaction nets (DIN) given in [12], and the attempt at extending it to the same authors' differential $\lambda$-calculus [11]. We will skip over some definitions and technical points in this section. For a definition of DINs one may refer to the next section, and take the promotion-free fragment of intuitionistic differential nets.

### *2.1 Resource calculus and differential interaction nets*

Starting from different motivations various authors have studied resource calculus [1,2,15], introduced as $\lambda$-calculus with multiplicities by Boudol. Ehrhard and Regnier present it with a reduction borrowed by their differential $\lambda$-calculus, and restrict it to the linear part, by ruling out infinitely available arguments. We present it here.

Given a denumerable set of variables $\mathbb{V}$ the set of simple terms $\Delta$ 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*, following [1] [1]. This language is extended to $R\langle\Delta\rangle$, the set of terms, and the constructors of the grammar extended by multilinearity, so that for example $\langle r\rangle (cu + dv)A = c\langle r\rangle uA + d\langle r\rangle vA$. The set of free variables of a simple term is defined as usual, and we say $x \in t$ to mean "$x$ free in $t$".

On this language 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 the multilinearity of constructors. Moreover we have the linear substitution defined by

$$\frac{\partial y}{\partial x} \cdot u := \delta_{x,y}, \quad \frac{\partial \lambda y.s}{\partial x} \cdot u := \lambda y.\frac{\partial s}{\partial x} \cdot u, \quad \frac{\partial \langle r\rangle A}{\partial x} \cdot u := \left\langle \frac{\partial r}{\partial x} \cdot u \right\rangle A + \langle r\rangle \frac{\partial A}{\partial x} \cdot u,$$

$$\frac{\partial A}{\partial x} \cdot u := \sum_{v \in A} \left( \frac{\partial v}{\partial x} \cdot u \right) A/v,$$

where $\delta_{x,y}$ is the Kronecker symbol, equal to 1 if $x = y$, 0 otherwise. The notation reflects the fact that this substitution can be regarded as a partial derivative of a term (which following the parallel in calculus would yield a linear form) 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

$$\frac{\partial}{\partial x} \left( \frac{\partial t}{\partial y} \cdot v \right) \cdot u = \frac{\partial}{\partial y} \left( \frac{\partial t}{\partial x} \cdot u \right) \cdot v.$$

Restricting to $R = \mathbb{N}$, reduction is defined by

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

first extended to simple terms and bags as a context closure and then on terms by linearity. One should note 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 the sum, and moreover the reduction does not substitute

[1] They are called poly-terms in [12].

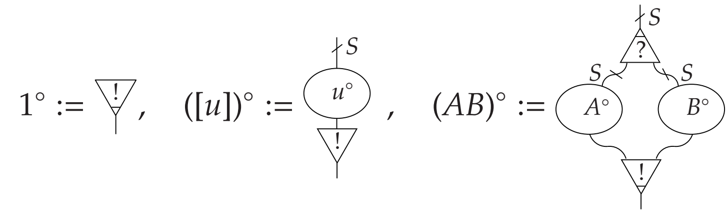



**Figure 1:** Rules to translate bags of arguments.

head variables only. The bs in $\beta_{\mathrm{bs}}$ stands for *baby-step* $\beta$-reduction, as in this paper we regard it as a fine-grained version of the reduction $\beta_{\mathrm{gs}}$, *giant-step* $\beta$-reduction, that completely exhausts the redex:

$$\langle \lambda x.s \rangle\, A \; \beta_{\mathrm{gs}} \; \frac{\partial^{\# A} s}{\partial x^{\# A}} \cdot A,$$

i.e. iterated linear substitution of $x$ with the terms in $A$, well defined because of Schwartz's lemma. More details will be given in the general setting in Section 4.

The translation of this calculus can be regarded as a particular case of the one given in Section 5. For now we can say that variables and abstraction are treated in the same way as one does for $\lambda$-calculus. However, as DINs are defined with a binary contraction and cocontraction, a bag is translated in two by two steps, as shown in Figure 1. A barred wire stands for multiple wires, $S$ labels the conclusions of the nets and is the set of free variables of the term/bag translated, where we possibly add needed dummy variables by means of weakenings. Application $\langle r \rangle\, A$ is translated by plugging $A^\circ$ on a tensor cut against the output port of $r^\circ$, just like boxes are in the translation of the application of $\lambda$-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 [12] 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, the rigorous statement of the simulation result is that

$$u \; \beta_{\mathrm{bs}} \; v \implies u^\circ \equiv_a \overset{m}{\rightarrow} \overset{e*}{\rightarrow} \overset{m}{\leftarrow} \equiv_a v^\circ,$$

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

$$u \; \beta_{\mathrm{gs}} \; v \implies u^\circ \overset{m}{\to}\overset{e*}{\twoheadrightarrow}\equiv_a v^\circ,$$

or by adopting the translation $t^\bullet$ which normalizes multiplicative cuts (see Section 6 for a sketched discussion on this translation and $\sigma$-equivalence), for which we would have

$$u \; \beta_{\mathrm{bs}} \; v \implies u^\bullet \equiv_a \overset{e*}{\to}\overset{m}{\twoheadrightarrow}\equiv_a v^\bullet.$$

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

This problem with (co)contractions is linked with one often arising 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 [23] – for now it seems hard to apply it in differential nets with boxes, we will see how the rule of codereliction against box introduces many difficulties;
- using such an identification as an equivalence relation, as hinted in [12] for DINs and investigated in [3,8] 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 [4] – which is is the way we are adopting here.

In Section 3.4 we will also address the issue with contraction and weakening order with respect to box borders as [4] does for LL.

### 2.2 *Differential $\lambda$-calculus and differential nets*

A natural direction of investigation arising from [12] and [11] is the question whether differential $\lambda$-calculus can be translated into differential nets. The first problem which arises is that DINs are promotion-free, and though from the syntactical point of view adding exponential boxes is easy, it has not yet been done in literature, other than by replacing boxes with their

Taylor expansion [2], i.e. an infinite sum which therefore deprives the system of its finitary nature. In next section we will thus introduce this system as *differential nets* or DNs, dropping the "interaction" wording because Lafont's interaction net paradigm [16] is broken by the promotion cell. This same reason makes fundamental results like confluence or normalization far harder. Therefore we will spend care in proving such results. As hinted in Section 6, the proof techniques used should cover much work in the way to settle normalization of propositional (first order) typed differential nets also.

The second problem arises from the particular syntax presented in [11]. We briefly sketch it here, assuming $R = \mathbb{N}$ and therefore dodging some of the difficulties. Simple terms are defined in this three forms:

- $\mathrm{D}_{i_1,\dots,i_n}\, x \cdot (u_1, \dots, u_n)$, imposing an invariance with respect to permutations $\sigma \in S_n$ so that

$$\mathrm{D}_{i_1,\dots,i_n}\, x \cdot (u_1, \dots, u_n) = \mathrm{D}_{i_{\sigma(1)},\dots,i_{\sigma(n)}}\, x \cdot (u_\sigma(1), \dots, u_{\sigma(n)});$$

- $\mathrm{D}^n_1\, \lambda x.s \cdot (u_1, \dots, u_n)$, with a similar invariance for permutations,
- $(u)v$ where $v$ is a differential term, i.e. a sum with coefficients in $\mathbb{N}$ over simple terms.

$\mathrm{D}_k\, u \cdot v$ stands for the differential operator relative to the $k^{\text{th}}$ argument of $u$ in the direction of $v$. The syntax makes it so that the commutation properties of such operators are internalized, by shifting them further inside the term. As an example:

$$\begin{aligned} \mathrm{D}_{1,2}(\lambda x_1, x_2, x_3.s)u \cdot (v_2, v_3) &= \Big(\mathrm{D}_{2,3}\, \lambda x_1, x_2, x_3.s \cdot (v_2, v_3)\Big)u = \\ &= \Big(\lambda x_1.\, \mathrm{D}_1(\lambda x_2.\, \mathrm{D}_1\, \lambda x_3.s \cdot v_3) \cdot v_2\Big)u, \end{aligned}$$

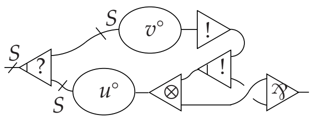
where one can see how the two $\mathrm{D}_k$s "shift" into position. These operations are taken as equalities, so that the substitution operator, which we here write as $t\,[x := v]$, is not anymore an almost blind copying of $v$ in every occurrence of $x$ as in $\lambda$-calculus, as after copying one must shift the differential operators as hinted by the example.

The new reduction rule is the linear one:

$$\mathrm{D}_1\, \lambda x.s \cdot v \to \lambda x. \frac{\partial s}{\partial x} \cdot v,$$

[2] If $R$ admits natural fractions, an exponential box has surprisingly enough the Taylor expansion of the exponential, i.e. a box containing $\pi$ is $\sum_{k=0}^{+\infty} \frac{\pi^k}{k!}$, where the power must be intended as $k$ copies of $\pi$ conveniently linked with cocontractions and contractions.

with $\frac{\partial s}{\partial x} \cdot v$ being a linear substitution similar to the one described for resource calculus, with particular care in handling the application, which is not linear in the argument. Now if we try to give a translation in nets extending the one for $\lambda$-calculus, $\mathrm{D}_1\,\lambda$ is a redex, so one chooses to represent $\mathrm{D}_1$ with a tensor cut against the main conclusion of the differentiated term. So in $\mathrm{D}_1\,\lambda$ is present, the corresponding ⅋ forms a multiplicative redex in the net. However in the reduct of linear reduction the $\lambda$ is still present. One might therefore represent such a situation with

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

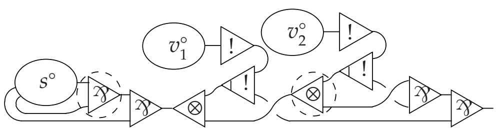


where the rightmost ⅋ corresponds to a potential abstraction that gets there if $\mathrm{D}_1$ fires. This could pose problems in sequentialization proofs: one should distinguish between ⅋s that correspond to actual abstractions and those that are justified by a differentiation.

Moreover the way in which differential operators shift in terms can be troublesome. Take the following term and its reduction:

$$\big(\lambda x.\, \mathrm{D}_{1,2}\, x \cdot (v_1, v_2)\big)\lambda y_1, y_2.s \to \mathrm{D}_{1,2}\, \lambda y_1, y_2.s \cdot (v_1, v_2) =$$
$$= \mathrm{D}_1(\lambda y_1.\, \mathrm{D}_1\, \lambda y_2.s \cdot v_2) \cdot v_1 \to \mathrm{D}_1\left(\lambda y_1, y_2.\left(\frac{\partial s}{\partial y_2} \cdot v_2\right)\right) \cdot v_1.$$

In nets the first reduct would be, after performing substitution via an appropriate substitution lemma:





where we are forgetting about contracted variables. The translation does not perform the operations that in the syntax are given as equalities. The result is that the marked cells are the ones corresponding to the couple $\mathrm{D}_1\,\lambda$ fired in the second reduction, whereas in the translation they are separated, and we do not get simulation.

We can see two solutions to this problem. One would be to directly employ the $t^\bullet$ translation that normalizes multiplicative cuts. This would probably abstract away the order of differentiation operators, and would by the way introduce $\sigma$-equivalence on differential calculus (see [23,24]). This would show how the two types of redex have a different syntactic nature: the classic "($\lambda$" that like in $\lambda$-calculus has a degree of imposed sequentiality that can be subsequently equated through $\sigma$-equivalence, and the differential "$\mathrm{D}_1\,\lambda$" which syntactically has $\sigma$-equivalence somehow built in.

We chose instead to investigate in another direction. In search for a calculus that would find an exact counterpart in DNs just like $\lambda$-calculus does in proof nets, we arrived at a version of full Boudol's $\lambda$-calculus with multiplicities. In fact just like resource calculus described by Ehrhard and Regnier in [12] is a an algebraic non-lazy version of the promotion-free fragment of $\lambda$-calculus with multiplicities, 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 $\lambda$-calculus, which explains why such a strong link with differential nets can be found. After the next two sections we will finally be able to define the translation in Section 5 and show sequentialization and bisimulation.

## 3 Intuitionistic differential nets

Intuitionistic differential nets (or iDN) are an extension of intuitionistic MELL proof nets with a differential operator. The main difference with respect to DIN (differential interaction nets) is the explicit presence of exponential boxes in the net. This in fact goes out of the interaction net language [16], as boxes have more than one active port for reduction. One of the main consequences of this is that strong confluence no longer holds, so that confluence and other results are a more delicate matter. Due to our main interest here in $\lambda$-calculus, we will deal only with a pure version of iDN. Typed and non intuitionistic versions are left for future work.

### *3.1 Statics: differential structures and correctness criterion*

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 certain number 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 indistinguishable and interchange-

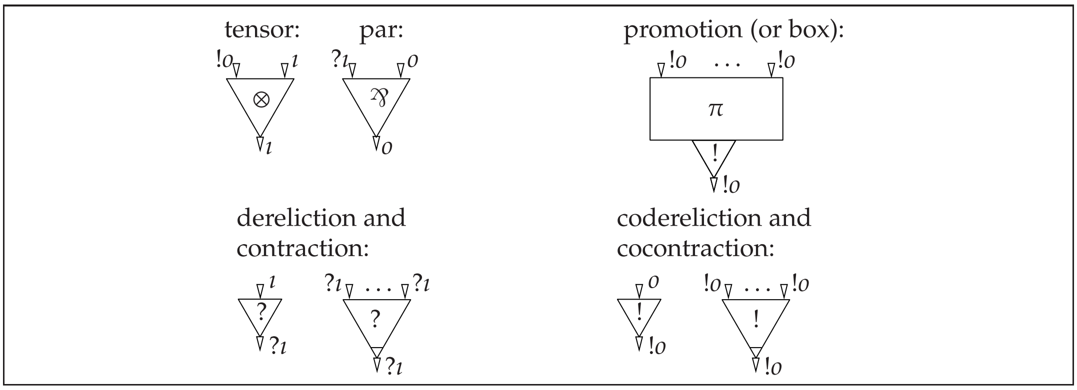



**Figure 2:** Cells for intuitionistic differential structures. Contractions and cocontractions are commutative and cannot have 2 ports.

able[3].

A typing is the assignment of a formula in a given language with duals to all directed ports. 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 (or the incoming type of its second). The sequence of the types of incoming free ports (which we recall is the type of wires directed towards the free ports and out of the net) is called interface. We are not much interested here in deadlocks, and they may be typed in any way.

**Differential structures.** The set $\mathrm{DS}_0$ of pure 0-depth simple intuitionistic differential structures (or 0-depth simple DSs for short) is the set of nets typable with formulas $o, !o$ and respective duals $\iota, ?\iota$ with symbols, arities and typing rules defined in Figure 2, without the promotion cell. Then by induction the set $\mathrm{DS}_{k+1}$ of $k+1$-depth simple DSs is the set of nets built with all cells in Figure 2. To each promotion cell with $n$ ports we must associate an element $\pi$ in $R\langle \mathrm{DS}_k \rangle$ with all addenda having an interface made of $n-1$ $?\iota$ and an $o$. This associated sum is called the content of the box and has a fixed correspondence between its $?\iota$-conclusions and the auxiliary ports of the box. $\mathrm{DS}_k$ is increasing, so we can define the set of simple DSs as $\mathrm{DS} := \cup_{k \in \mathbb{N}} \mathrm{DS}$. Intuitionistic differential structures are elements of $R\langle \mathrm{DS} \rangle$

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

where addenda have the same interface. Corresponding free ports in the addenda of a structure will be usually identified. The (exponential) depth of a net $\pi$ is the minimal $k$ such that $\pi \in R\langle \mathrm{DS}_k \rangle$. The exponential depth of a cell in $\pi$ is the number of boxes in which it is contained. As these nets are one of the main characters in this paper, we will from now on drop the "intuitionistic" nomenclature.

We will often omit types, as they can be easily derived from the cells involved. We will call $n$-contraction (resp. $n$-cocontraction) one which has $n+1$ ports. 0-contractions (resp. 0-cocontractions) are also called weakenings (resp. coweakenings). 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. A wire between the principal port of a (co)contraction and an auxiliary one of another (co)contraction is called an *associative redex*. A contraction which has more than one auxiliary port connected to auxiliary ports of the same box is called a *push redex*, while a box where all the addenda of the content have a weakening on a given conclusion is called a *pull redex*. An *axiom* is a wire which does not connect any principal or box auxiliary port.

**Contexts.** A simple context $\omega[\ ]$ is a simple differential structure built with an additional special node, the *hole*, which has an arbitrary but fixed arity and outgoing types, the sequence of which is called the internal interface of $\omega[\ ]$. We impose that the hole appears 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 $\pi$ differential structure and $\omega[\ ]$ simple context.

Given a simple structure $\pi$ and a context $\omega[\ ]$ such that the interface of $\pi$ is equal to the internal interface of $\omega[\ ]$ we define $\omega[\pi]$ by substituting $\pi$ for the hole, i.e. identifying the free ports of $\pi$ with corresponding ports of the hole and then erasing them by merging wires which share such ports. In case $\pi$ is a linear combination the sum is extended to the whole content of the box containing the hole, or the whole context is there is none. Formally, after having defined $\omega[\lambda]$ for simple nets, $\omega[\sum_i c_i\lambda_i] := \sum_i c_i\omega[\lambda_i]$ if the hole is at depth zero, otherwise $\omega[\sum_i c_i\lambda_i]$ is inductively the result of substituting the content $a\psi[\ ]+\sigma$ of the box containing the hole with $a\psi[\sum_i c_i\lambda_i]+\sigma$. Given a relation $\rho$ on structures 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 $\sigma = \omega[\sigma']$.

Though structures already have computational meaning, we define the *correctness criterion* following the Danos-Regnier one for LL proof nets [6].

**Correct nets.** Given a simple structure $\lambda$ a *switching* of $\lambda$ is an unoriented graph $G$ with cells as nodes, obtained by deleting for every par and contraction all wires on its auxiliary ports but one, and converting all remaining wires as edges between the cells connected by the wire. A *principal switching* is one that on $⅋$s always erases the exponential wire. A simple DS $\lambda$ is said to be *correct*, or a simple *differential net*, or simple DN for short, if

- every switching $G$ of $\lambda$ is acyclic and has a number of connected components equal to the number of weakenings at depth 0 in $\lambda$ plus one;
- inductively every content of a box is correct.

A sum of simple structures is correct if it is a sum of correct structures. 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.

We here state a lemma which will be used in Section 5.

**Lemma 1** *A correct 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 [23]. 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 take $R = \mathbb{N}$. Though greatly interesting, other cases such as $\mathbb{Q}^+$ pose problems for normalization issues [4], not to speak of cases where $R$ has subtraction, where one cannot even speak of a normal form [5]. In this setting sums may always be written without coefficients, as for $c \in \mathbb{N}$ $c \cdot \pi = \pi + \cdots + \pi$ $c$ times. Note that multiplicities still count. We may redefine contexts, ruling out the multiplication by a coefficient, and making the upcoming definition of reduction more atomic. This is left to personal taste, as the results do not change.

Reduction rules are defined in Figure 3, and take the form of ordered couples

[4] Take $\pi$ reducing in one step to $\pi'$. Then $\pi = \frac{1}{2}\pi + \frac{1}{2}\pi \to \frac{1}{2}\pi + \frac{1}{2}\pi' \to \frac{1}{4}\pi + \left(\frac{1}{4} + \frac{1}{2}\right)\pi' \to \ldots \to \frac{1}{2^k}\pi + \sum_{i=1}^{k} \frac{1}{2^k}\pi' \to \ldots$ Achilles and the turtle come to mind, and it is opinion of the author that there is a way to satisfactorily treat this "paradox".

[5] Taken a reduction $\pi \to \pi'$, then any net $\sigma$ reduces to $\sigma + \pi - \pi'$.

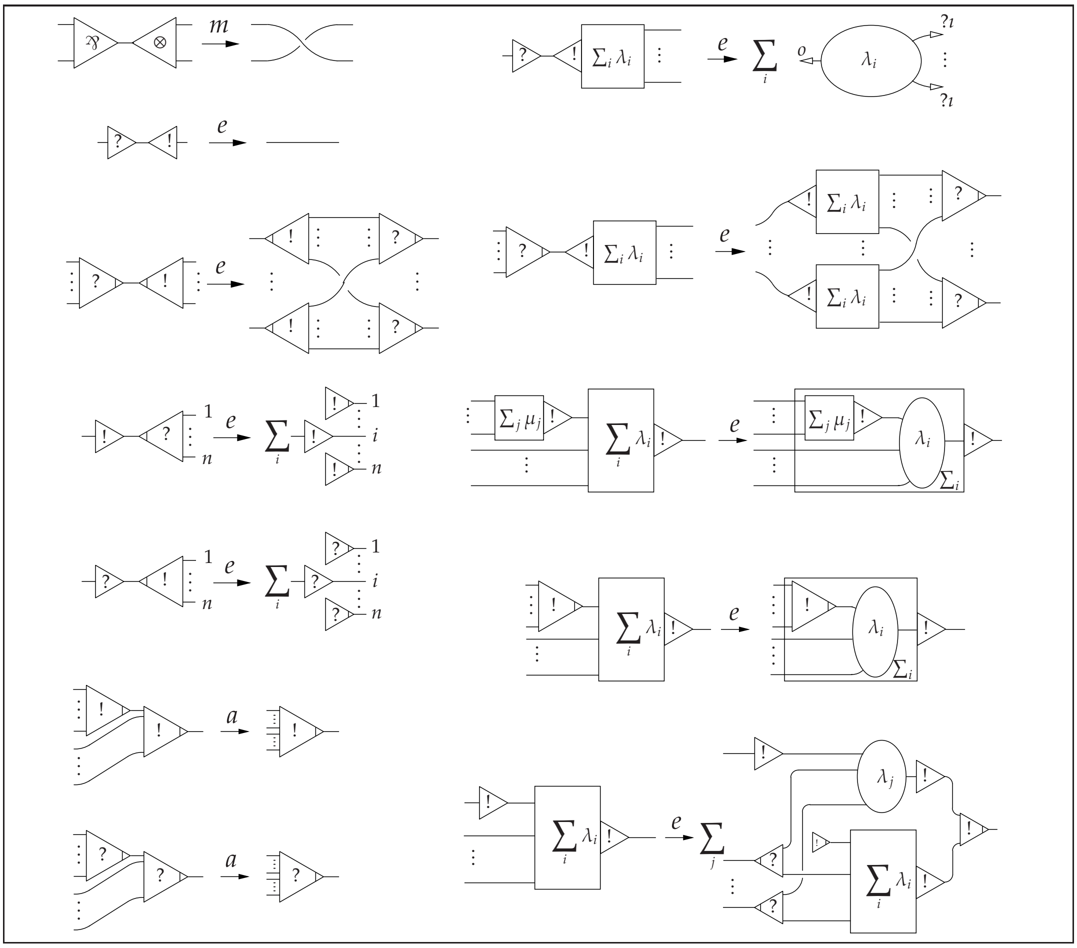


**Figure 3:** Reduction rules for differential structures.

of modules. They in turn define a relation on nets by context closure. In all reduction rules contraction or cocontraction cells in the reducts which come out to have 2 ports are a convention to denote a single connecting wire. Note also that the rules cover the cases for (co)weakening.

The reduction marked by $m$ is the *multiplicative* one, while $e$ denotes the *exponential* ones. These reduce all possible cuts. Moreover the *associative* reduction ($a$) reduces associative redexes. Reductions of push and pull redexes will be discussed later in Section 3.4. Remark 3 shows why we are dealing with $a$-reduction together with the other more classical ones: $e$-reduction (and $em$-reduction) is not confluent without it. Reductions can be seen to preserve both typing and correctness.

The rest of this section will be devoted to proving fundamental properties of these reductions. To sum up we will prove that $ea$ is strongly normalizing and confluent, and that $mea$ is confluent. For now it is immediate that $m$ and $a$ are strongly normalizing as they decrease the number of cells, and that they are also strongly confluent.

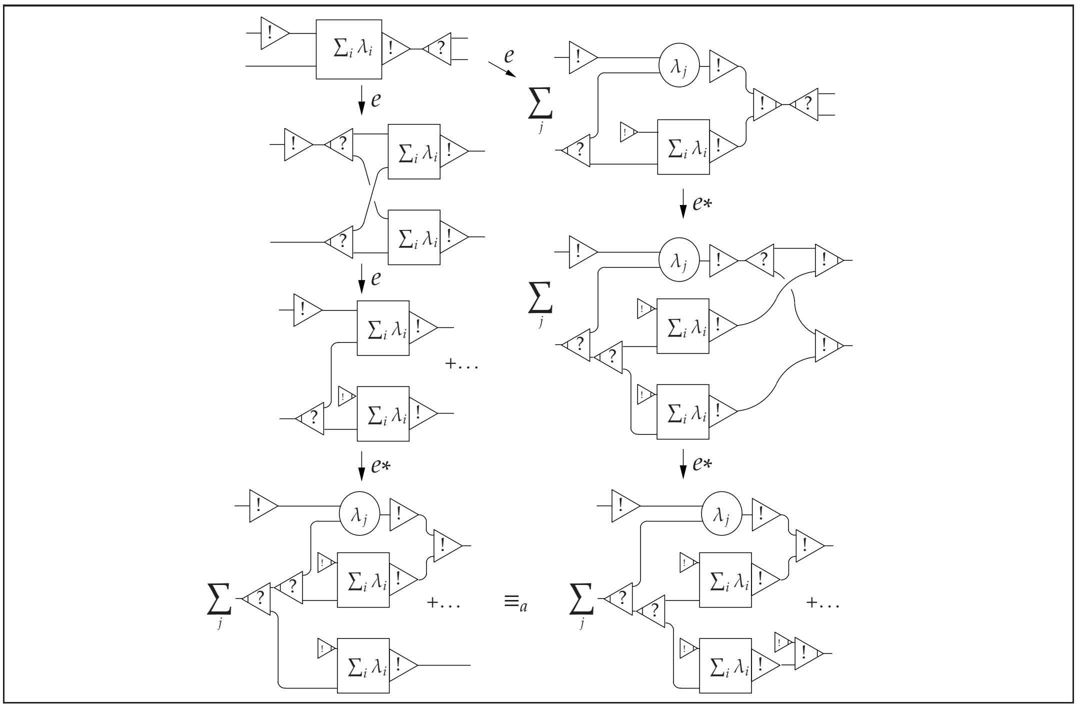


**Figure 4:** Confluence diagram for codereliction vs box vs contraction critical pair.

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

**PROOF.** As usual one checks the critical pairs. Some of them have been covered in the literature about LL proof nets. The other ones 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 only one other conclusion (apart from the one cut against the codereliction) and that the contraction is a 2-contraction. The two reductions are shown in Figure 4. The $+\dots$ parts are the other addenda in the sums which are completely symmetric. In the end we arrive to two $a$-equivalent (equivalent up to associativity) forms, which therefore normalize with $a$ to the same net. □

**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.*

### *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, the possible new cut is not exponential anymore and thus it is inactive from the point of view of this reduction [6] .

**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. When a cut fires 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 addenda, 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 addenda, so that if all addenda 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 done. So we count the weights inside boxes as many times as are these potential copies. Last problem: boxes contain sums, and when a box is duplicated and opened every copy may spawn a different addendum. 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 $A \setminus \{a\} := A[a \mapsto 0] = A - A(a)[a]$. 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 $A \setminus \{\max A\} \leq B \setminus \{\max B\}$;
- $\leq$ is the transitive and reflexive closure of $<_1$, where $A <_1 B$ iff there is $b \in B$ such that $A = B - [\, a \,] + K$ where $\max K < a$, i.e. all elements of $K$ are less than $a$.

This is a well ordering on $\mathcal{M}_{\text{fin}}(X)$, a proof of which can be found in [18]. Moreover it is compatible with multiset sum, turning $(\mathcal{M}_{\text{fin}}(X), <, [\,], +)$ into a well-ordered monoid itself.

[6] In a typed first order setting, we can use such hypothesis by considering the logical complexity of cut formulas. See Section 6 for a sketched discussion.

**Definition 4** *The* convolution product *of two finite multisets A and B is (in functional notation):*

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

The support of $A * B$ is $|A| + |B| = \{ x + y \mid x \in |A|, y \in |B| \}$ (and is therefore finite), and in fact the product can be seen as a generalization of that set operation to multisets, i.e. we could write $A * B = [ x + y \mid x \in A, y \in B ]$ where we count multiplicities. This operation enjoys some good and trivially provable properties: it is commutative, associative, has $[0]$ as unit and $[\,]$ as absorbing element, and distributes over multiset sum. A less trivial property is the following.

**Proposition 5** *The convolution product is compatible with multiset order, i.e. 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 $V = V_0 + [a]$ and $U = V_0 + K$, with $\max K < a$. Now

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

Now 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. □

We define the power of a multiset $V^k$ by iterated convolution product. Compatibility assures us this power is monotone increasing both with respect to $V$ and to $k$ (as every $V \neq [\,]$ is greater than the unity $[0]$). 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. The two directions give rise to a slightly different formal treatment. Let us fix for the subsequent definitions a module $\pi$.

**Definition 6 (!-path)** *An* !*-path is a sequence of exponential wires*, not necessarily at the same exponential depth, *such that*

- *either there is a cell C between $e_k$ and $e_{k+1}$, or $e_{k+1}$ is on an auxiliary port of a promotion cell, and $e_k$ is the corresponding wire in a simple net which is an addendum of the content of the same promotion cell.*

- *if $C$ is a cocontraction or a box, then $e_k$ is on its principal port and $e_{k+1}$ on an auxiliary one.*
- *if $C$ is a contraction, then $e_k$ is on an auxiliary port and $e_{k+1}$ on its principal port.*

Because of acyclicity there are no !-loops. Maximal !-paths can only end on conclusions of the whole module, coweakenings, coderelictions, or boxes without auxiliary ports. We define the *!-measures* cd (*codereliction count*) and $\ell_!(e)$ (*!-length*) by induction on the maximum length of maximal !-paths starting from $e$. The definition is given by cases depending on the ! port of the wire directed as in !-paths. For every incoming $?\iota$-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 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 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, or is the conclusion of a simple net inside a box, $\mathrm{cd}(e) := \mathrm{cd}(f)$ and $\ell_!(e) := \ell_!(f)$ where $f$ is the wire on the principal port of the contraction or on the corresponding auxiliary door of the box respectively.
- If $e$ is on a cocontraction, and $f_i$ are the wires on the auxiliary ports of the cell, then

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

- If $e$ is on a box, and $f_i$ are the wires on its auxiliary ports, then

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

We will also use $\#_!(e)$ (!-count) to mean $1 + \mathrm{cd}(e)$.

**Definition 7 (?-path)** *A ?-path is a sequence of exponential wires* at the same exponential depth *such that the reversed sequence is an !-path.*

Clearly there are no ?-loops, and maximal ?-paths end with either a conclusion, a weakening or a dereliction. We define the *?-measures* $\#_?(e)$ (*?-count*) and $\ell_?(e)$ (*?-length*) by induction on the exponential codepth of $e$ (the depth of the net 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 ? port of the wire, and there are assigned variables on $\mathbb{N}$ named $\#_?(x)$ and $\ell_?(x)$ for every incoming $!o$-typed conclusion $x$.

- If $e$ is on a dereliction or a weakening then $\#_?(e) := 1$ and $\ell_?(e) := 1$.
- If $e$ is on a conclusion $x$, $\#_?(e) := \#_?(x)$ and $\ell_?(e) := \ell_?(x)$.
- If $e$ is on a cocontraction, $\#_?(e) := \#_?(e)$ and $\ell_?(e) := \ell_?(f)$ where $f$ is the wire on the principal port of the cocontraction.
- If $e$ is on a contraction, and $f_i$ are the wires on the auxiliary ports of the cell, then

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

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

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

  where $e^{\lambda_i}$ denotes the conclusion in net $\lambda_i$ inside the box 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 ports rather then only wires, by calculating the corresponding measure on the unique wires connecting them.

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, getting again functions on the variables of the conclusions of the whole context. Note however that because of the feedback nature of the presence of cd in the definitions, ?-measures cannot be calculated on the internal interface of the context *before* plugging the module, while !-measures can.

**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 with a different notation the two. The measure will be defined by induction on the exponential depth of the net. Given $\sigma$ the content of a box in $\pi$, $|\sigma|_\pi$ denotes the relative measure $|\sigma|$ evaluated on the !-measures of the auxiliary ports of the box (there are no other exponential conclusions). Note that the measure of the content of a box is not independent of what is outside it. Given a set of wires $W$, let $|W|$ be $[\,\ell(e) \mid e \in W\,]$, i.e. the multiset of lengths over $W$. For a simple module $\lambda$ let $\mathcal{C}_0(\lambda)$ (resp. $\mathcal{B}_0(\lambda)$) be the set of cuts (resp. boxes) at exponential depth 0 in $\lambda$. Given a box $B$, we denote by $\sigma(B)$ its content and by $\#(B)$ the count $\#(p) = \#_?(p)\#_!(p)$ on its principal door $p$.

**Definition 8 (measure of a module)** *In case $\pi = \sum_i \lambda_i$ is a sum of simple modules $|\pi| := \sum_i |\lambda_i|$. The measure of a simple module $\lambda$ is defined as*

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

*Note that the first factor can be furthermore factorized in $\prod_{c \in \mathcal{C}_0(\lambda)} \Big[ \, [\ell(c)] \, \Big]$.*

Note that the measure is monotone in all the measures of wires defined above. In order to prove that the measure does not increase in other parts of the net, it will suffice to show that those measures do not increase.

**Intuitive idea of the measures.** $\ell$ 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. This resembles the idea of single threaded slices, a notion appearing in[21] in the case of LL with additives.However here we expand the measures given by the content of the box with a power operation which does nothing else than making potentially coexist together a number (given by the count # on the box) of nets fetched from the box, a coexistence that single threaded slices rule out.

In the following, given a simple module $\lambda$, let $C_?(\lambda)$ and $C_!(\lambda)$ be the set of the (incoming) $?\imath$ and $!o$ typed conclusions of $\lambda$ respectively. We see that $\mathrm{cd}_\lambda(y)$ with $y \in C_!$ is a function on just the variables $\mathrm{cd}(x)$ for $x \in C_?$, as cd is the only self defined measure. We say that $\mu$ *can replace* $\lambda$ if !-measures on $C_!$ and ?-measures on $C_?$ (which are functions on the variables of the ports) are pointwise less for $\mu$ than for $\lambda$.

The idea is that when (correctly) plugged in a context, the measures of the whole context do not increase between the two. However this is not really immediate, as values on the variables are not independent of what is plugged into the hole. Let us show what we mean with a little informal reasoning. Let the symbols ! and ? here be treated as functions that give ! and ? measures respectively, so that for example $!C_?$ is the collection of !-measures on $C_?$ ports. We will denote the various dependencies between the measures by function literals such as $F$, $G$, and we will put $\lambda$ and $\mu$ as subscript whenever something depends on whether we plug $\lambda$ or $\mu$. For example $!C_?$ does not depend on it, so we do not put any subscript. We can write

$$!_\lambda C_! = F_\lambda(!C_?) \geq F_\mu(!C_!) = !_\mu C_?,$$

where $F_\lambda$ and $F_\mu$ denote one of the dependencies that are the object of the comparison in the definition of "$\mu$ can replace $\lambda$". For $?C_?$ the discussion is more delicate, as $?C_!$ is not independent of the module plugged in, as it depends (let us say with function $H$) on $!C_!$, because of cd being present in all definitions. What is crucial is that all these dependencies are monotone increasing, and we have already seen $!_\lambda C_! \geq !_\mu C_!$, so:

$$?_\lambda C_? = G_\lambda(?_\lambda C_!, !C_?) = G_\lambda\Big(H(!_\lambda C_!), !C_?\Big) \geq$$
$$\geq G_\lambda\Big(H(!_\mu C_!), !C_?\Big) \geq G_\mu\Big(H(!_\mu C_!), !C_?\Big) = ?_\mu C_?.$$

We just had to strive a little more to be able to apply the pointwise comparison.

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

**Lemma 9 (modularity)** *Let $\pi = \omega[\lambda]$ and $\sigma := \omega\left[\sum \mu_i\right]$ be correct nets, where $\omega$ is a context, $\lambda$ and $\mu_i$ for $i = 1, \ldots, n$ are simple modules. Suppose that for every $i$ $\mu_i$ can replace $\lambda$, and let $C_i$ be the set of terminal exponential wires of $\mu_i$ which were not terminal in $\lambda$. Suppose moreover that*

- *$n = 1$ and $\Big[|C_1|_{\mu_1}\Big] * \big|\mu_1\big| < |\lambda|$ pointwise,*
- *or we can write*

$$|\lambda| = [u] * X, \quad \big|\mu_i\big| = [v_i] * X_i$$

  *so that we have that pointwise $X_i \leq X$ and $|C_i|_{\mu_i} + v_i < u$ for every $i$.*

*Then $|\pi'| < |\pi|$.*

**PROOF.** Let $\varphi[\ ]$ be the simple context with its hole at depth 0, $\psi[\ ]$ the context, $a$ the coefficient and $\chi$ the net such that $\omega[\ ] = \psi\Big[a\varphi[\ ] + \chi\Big]$ 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 $\big|\varphi[\sum_i \mu_i]\big| = \sum_i \big|\varphi[\mu_i]\big| < \big|\varphi[\lambda]\big|$. 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.

As previously remarked, by the hypothesis that $\mu_i$ can replace $\lambda$ we have that all ?-measures on $C_?(\lambda)$ and !-measures on $C_!(\lambda)$ decrease from $\varphi[\lambda]$ to $\varphi[\mu_i]$. Thus, because of monotonicity, the measure of what is outside the hole decreases also. We have to keep track of wires on conclusions in $\lambda$ and $\mu_i$ which may be cuts when plugged in the context. Such wires that are cuts in both situations have a lesser measure in $\mu_i$ because of the decrease of measures. The ones that are cuts in $\lambda$ but not in $\mu_i$ do not bother us. Finally the ones that are cuts for $\mu_i$ but not for $\lambda$ are contained in $C_i$. Summing up, we can write

$$\left|\varphi[\mu_i]\right| \leq \left[\, |C_i|_{\mu_i} \right] * \left|\mu_i\right| * Y_i, \qquad \left|\varphi[\lambda]\right| = |\lambda| * Y$$

with $Y_i \leq Y$ pointwise, which weight the parts of the context outside the hole.

In case $n = 1$ we have $[\,|C_i|\,] * |\mu_1| < |\lambda|$ pointwise by hypothesis. We can apply it here because of monotonicity, and we get what we were looking for. Otherwise, putting it together:

$$\begin{aligned}\left|\sum_i \varphi[\mu_i]\right| &\leq \sum_i \left(\left[\, |C_i|_{\mu_i} \right] * \left|\mu_i\right| * Y_i\right) \leq \sum_i \left(\left[|C_i|_{\mu_i}\right] * [v_i] * X_i * Y_i\right) \leq \\ &\leq \sum_i \left(\left[|C_i|_{\mu_i} + v_i\right] * X * Y\right) = \left(\sum_i \left[|C_i|_{\mu_i} + v_i\right]\right) * X * Y = \\ &= \left[|C_1|_{\mu_1} + v_1, \ldots, |C_n|_{\mu_n} + v_n\right] * X * Y.\end{aligned}$$

Now $u > v_i + |C_i|_{\mu_i}$ for any $i$, which implies $[|C_1|_{\mu_1} + v_1, \ldots, |C_n|_{\mu_n} + v_n] < [u]$ and so

$$\left|\sum_i \varphi[\mu_i]\right| \leq \left[|C_1|_{\mu_1} + v_1, \ldots, |C_n|_{\mu_n} + v_n\right] * X * Y < [u] * X * Y = \left|\varphi[\lambda]\right|.$$

Let's return to $\omega[\ ] = \psi\left[a\varphi[\ ] + \chi\right]$. If $\psi[\ ] = [\ ]$, that is $\omega$'s hole is not contained in a box, we have nothing else to add, as the order is compatible with sum. If otherwise $B$ is the smallest box containing $a\varphi[\ ]+\chi$, we first note by inspection of the definition that the ?-measures on the auxiliary doors of $B$ with $\chi + \sum_i \varphi[\mu_i]$ inside are less than the same measures with $\chi + \varphi[\lambda]$ instead (decrease of measures on $\varphi[\ ]$ conclusions). Because of this remark we can write

$$\left|\psi\left[a \sum_i \varphi[\mu_i] + \chi\right]\right| = Z' * \left(a \sum_i \left|\varphi[\mu_i]\right| + |\chi|\right)^k$$

and

$$\left|\psi\left[a\varphi[\lambda] + \chi\right]\right| = Z * \left(a \left|\varphi[\lambda]\right| + |\chi|\right)^k$$

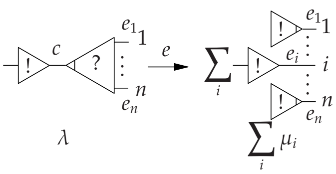
with $k$ given by the product of the count # on all the boxes containing $a\varphi[\ ]+\chi$ (which is equal between the two), and $Z' \leq Z$. We have that all !-measures on $C_?(\varphi[\ ])$ are the same (and $C_!$ is empty), so we can apply the pointwise comparison previously established on the measures $\sum_i \left|\varphi[\mu_i]\right|$ and $\left|\varphi[\lambda]\right|$, and we get the final result. □

2

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

**PROOF.** For each couple redex-reduct of $\xrightarrow{e}$ as presented in Figure 3 we have to verify the hypotheses of the modularity lemma. In fact $\pi \xrightarrow{e} \sigma$ 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. 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 a first easy example of the way the sum is dealt with and the two most interesting (and hardest) cases.

*Codereliction vs contraction.*




First, $\mu_i$ can replace $\lambda$, as

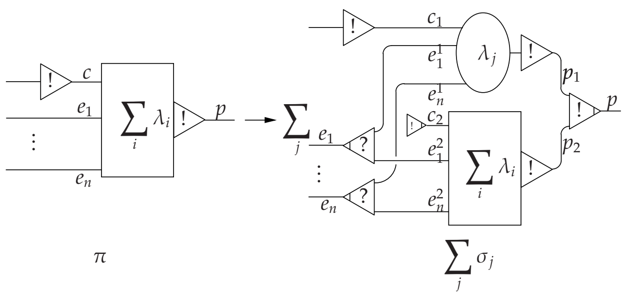
$$\mathrm{cd}^{\mu_i}(e_j) = \delta_{i,j} \leq 1 = \mathrm{cd}^{\lambda}(e_j), \quad \ell_!^{\mu_i}(e_j) = 1 = \ell_!^{\lambda}(e_j).$$

Then, $|\lambda| = \Big[\,[\ell(c)]\,\Big]$ and $\left|\mu_i\right| = \Big[\,[\,]\,\Big]$, but all $e_j$s have become terminal. Anyway however

$$\ell^{\mu_i}(e_j) = \ell_?(e_j) + 1 < 1 + \max_k\Big(\ell_?(e_k)\Big) + 1 = \ell^{\lambda}(c)$$

so that $[\ell(c)] > |C_i|_{\mu_i} + [\,]$ (as defined in the hypotheses of Lemma 9) which ends this case.

*Codereliction vs box.*




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

$$\mathrm{cd}^{\sigma_j}(p) = 1 + \mathrm{cd}^{\sigma_j}(p_2) = 1 + \textstyle\sum_h \mathrm{cd}^{\sigma_j}(e_h^2) = \mathrm{cd}^{\pi}(c) + \sum_h \mathrm{cd}(e_h) = \mathrm{cd}^{\pi}(p),$$

$$\ell_!^{\sigma_j}(p) = 1 + \max\Big(1, 1 + \textstyle\sum_h \mathrm{cd}(e_h)\Big) = 1 + \mathrm{cd}^\pi(c) + \sum_h \mathrm{cd}(e_h) = \ell_!^\pi(p),$$

$$\begin{aligned}
\#_?^{\sigma_j}(e_h) &= \#_?^{\sigma_j}(e_h^1) + \#_?^{\sigma_j}(p_2)\Big(1 + \textstyle\sum_k \mathrm{cd}(e_k)\Big)\max_i\Big(\#_?^{\sigma_j}(e_h^{2\lambda_i})\Big) \leq \\
&\leq \#_?(p)\max_i\Big(\#_?^\pi(e_h^{\lambda_i})\Big) + \#_?(p)\Big(1 + \textstyle\sum_k \mathrm{cd}(e_k)\Big)\max_i\Big(\#_?^\pi(e_h^{\lambda_i})\Big) = \\
&= \#_?(p)\Big(2 + \mathrm{cd}(e_k)\Big)\max_i\Big(\#_?^\pi(e_h^{\lambda_i})\Big) = \#_?^\pi(e_h), \\
\ell_?^{\sigma_j}(e_h) &= 1 + \max\Big(\ell_?^{\sigma_j}(e_h^1), 1 + \max_i(\ell_?^{\sigma_j}(e_h^{2\lambda_i})) + \ell_?(p) + \textstyle\sum_k \mathrm{cd}(e_k)\Big) \leq \\
&\leq 1 + \max\Big(\ell_?^\pi(e_h^{\lambda_j}), 1 + \max_i(\ell_?^\pi(e_h^{\lambda_i}))\Big) + \ell_?(p) + \textstyle\sum_k \mathrm{cd}(e_k) \leq \\
&\leq 2 + \max_i\Big(\ell_?^\pi(e_h^{\lambda_i})\Big) + \ell_?(p) + \textstyle\sum_k \mathrm{cd}(e_k) = \ell_?^\pi(e_h).
\end{aligned}$$

We take the measures of the modules:

$$|\pi| = \Big[\,[\ell^\pi(c)]\,\Big] * \Big(\textstyle\sum_i |\lambda_i|_\pi\Big)^{\#_?(p)\left(1+\mathrm{cd}(p)\right)},$$

$$\left|\sigma_j\right| = [\delta_j] * \left|\lambda_j\right|_{\sigma_j} * \Big(\textstyle\sum_i |\lambda_i|_{\sigma_j}\Big)^{\#_?^{\sigma_j}(p_2)(1+\mathrm{cd}^{\sigma_j}(p_2))},$$

where $\delta_j = [\ell^{\sigma_j}(c_1), \ell^{\sigma_j}(c_1)]$ 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 $\sigma_j$ than in $\pi$ as all measures on its border are the same apart from cd which is 1 less in $\sigma_j$, while the measure remains the same on the linear part $\lambda_j$. So:

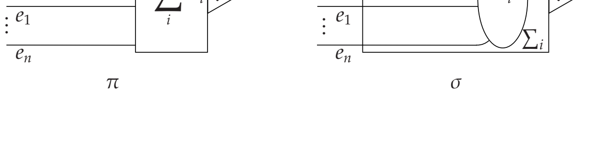
$$\begin{aligned}
&\left|\lambda_j\right|_{\sigma_j} * \Big(\textstyle\sum_i |\lambda_i|_{\sigma_j}\Big)^{\#_?^{\sigma_j}(p_2)(1+\mathrm{cd}^{\sigma_j}(p_2))} \leq \\
&\qquad\leq \Big(\textstyle\sum_i |\lambda_i|_\pi\Big)^{\#_?(p)} * \Big(\textstyle\sum_i |\lambda_i|_\pi\Big)^{\#_?(p)\,\mathrm{cd}(p)} = \Big(\textstyle\sum_i |\lambda_i|_\pi\Big)^{\#_?(p)\left(1+\mathrm{cd}(p)\right)}.
\end{aligned}$$

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

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

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, as

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

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 $\varepsilon$ be $[\ell^\pi_!(c) + \max_j\big(\ell^\pi_?(c^{\lambda_j})\big)]$ we have $\delta_i \le \varepsilon < [\ell^\pi(c)]$. Moreover $\#^\sigma(c^{\lambda'_i}) \le \max_i\big(\#^\pi_?(c^{\lambda_i})\big)\#^\pi_!(c)$, $\#^\sigma(p) = \#^\pi(p)$, $|\lambda_i|_\sigma = |\lambda_i|_\pi$ and $\left|\mu_j\right|_\sigma = \left|\mu_j\right|_\pi$, so we get

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

We end the section by stating and proving results which now come easily. First we can throw in the associative reduction.

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

**PROOF.** One has to check that $\overset{a}{\to}$ 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 $\pi$, which decreases in $a$-reductions. Lexicographic well-ordering does the rest. □

By Newman's Lemma and Lemma 2 we get

**Theorem 12** *The reduction $\stackrel{ea}{\to}$ is confluent.*

We can now briefly deal also with the $m$ reduction, though working in the pure setting we clearly cannot hope for normalization. An essay on the lemmas we use to prove confluence can be found in the introduction of [22].

**Lemma 13** *If $\pi \stackrel{ea}{\to} \sigma$ and $\pi \stackrel{m}{\to} \tau$ there is $\upsilon$ such that $\sigma \stackrel{m*}{\to} \upsilon$ and $\tau \stackrel{ea}{\to} \upsilon$.*

**PROOF.** $m$ reductions cannot erase an exponential cut or an associative redex, while an exponential reduction can erase or duplicate a multiplicative cut, but cannot change it. Anyway we can still perform the $e$-reduction in $\tau$ and close by performing the multiplicative reductions on the copies of the $m$-redex in $\sigma$. □

By Huet's Lemma we thus get commutation of $\stackrel{ea*}{\to}$ and $\stackrel{m*}{\to}$. By confluence of $ea$ and of $m$ (recall that $m$ is strongly confluent) we finally get by Hindley-Rosen's Lemma confluence of $\stackrel{eam}{\to}$.

### *3.4 Settling contractions and boxes: push and pull*

We have shown that pure intuitionistic differential nets with multiplicative exponential associative reduction are a "good" rewriting system. However we have yet to fully tackle the problem with the order of identification of variables we discussed in Section 2. The associative reduction that we had to add anyway to get confluence solves part of it: contractions made at the same exponential depth are merged and their order is forgotten. It remains to settle the order in which contractions (and weakenings) are made with respect to box enclosures. In an approach similar to [4], we will show that we can add two more reductions which do not ruin the properties proved in the previous section.

The $p$-reductions (*push* and *pull*) are presented in Figure 5. Similarly to the associative reduction, if the outer contraction in the reduct of the push rule has one auxiliary port it must be regarded as the notation for 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

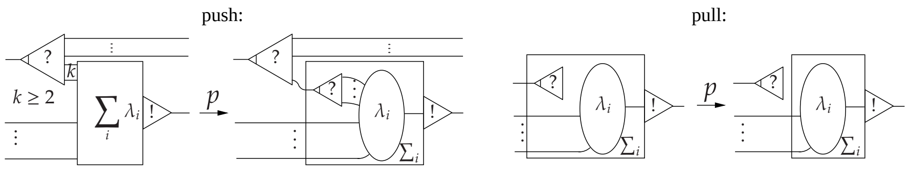



**Figure 5:** The push and pull rules. In the push rule $k \geq 2$ is required.

spawn contractions [7]. From now on we will denote by $c$ (for canonical) the combination of the associative, push and pull reductions. We will prove among other results that $c$ in itself is strongly normalizing and confluent, so we can speak of the unique *canonical form* $\mathrm{NF}^c(\pi)$ of $\pi$.

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

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

To prove strong normalization the approach used with the associative reduction would fail, as creating new contractions inside boxes may increase the measure. We instead slightly complicate the measure given in Definition 8 in order to have one which does not increase on both $a$ and $p$, and then define a measure which strictly decrease on $a$ and $p$ alone.

**The push count.** For every wire $e$ connected to an auxiliary port of a box $B$, consider all the !-paths starting from $e$. For each path $E$ of them count the number of contractions $C$ along its way that have another !-path from any auxiliary port of $B$ entering $C$ from an auxiliary port different than the one traversed by $E$. Say push($E$) is such number, and define

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

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

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

where $p$ is the principal port of the box.

The rest of the definitions remain the same, and we do not change the notation. In order to still have the result shown in Theorem 10 (exponential strong normalization) one has to check that the push count does not increase in all $e$-reductions. We briefly illustrate why it is so.

Dereliction against cocontraction, dereliction against box and codereliction against contraction cases clearly do not pose any problems. In contraction

[7] Note that this is different from what happens in LL.

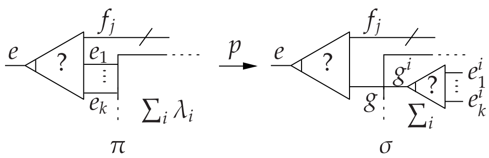
against cocontraction, every !-path $E$ traversing the redex persists exactly in the reduct, and if the contraction in the redex contributed to push($E$), then the only contraction traversed by $E$ in the reduct contributes too. The same can be said for !-paths traversing the redex of contraction against box, and paths $E$ starting from the auxiliary ports of the box (or starting somewhere inside it and going out of it) get duplicated in several copies in the reduct, each with the same push value as the new contraction they traverse surely does not contribute. Box against box and cocontraction against box are similarly easy.

For codereliction against box, there is one more !-path passing through the new cocontraction, however it ends immediately on the new codereliction, so it does increase neither the push count of other paths, nor the push count of some auxiliary ports with its own push count, as we take the maximum. Each !-path traversing the redex gets preserved in a unique !-path in the reduct with the same push count, as the new contraction does not add up to the count. Also !-paths starting from the box get preserved in the same way, and !-paths inside the box get copied also in the linear part, but anyway each copy gets the same push count, always because the new contractions cannot add up to any count.

So the measure $|\pi|$ still strictly decreases on $e$-reductions, as we have added a non increasing weight to lengths of cuts depending on auxiliary ports. Moreover we also have the following result.

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

**PROOF.** After noting that all $c$-reductions do not increase push counts, the only interesting case is the push reduction. Let us assign some names to wires (a barred wire stands for many wires, possibly none).




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

$$\ell^{\sigma}_{?}(e) = 1 + \max\Big(\max_j\Big(\ell_?(f_j)\Big), 1 + \text{push}^{\sigma}(p) + \max_i\Big(1 + \max_h(\ell^{\sigma}_{?}(e^i_h))\Big) + \ldots\Big) =$$

$$= 1 + \max\Big(\max_j\Big(\ell_?(f_j)\Big), \max_h\Big(1 + \text{push}^{\pi}(e_h) + \max_i(\ell^{\pi}_{?}(e^{\lambda_i}_h)) + \ldots\Big)\Big) = \ell^{\pi}_{?}(e),$$

where the dots indicate the part about the principal port omitted in the drawing which does not change. If there is no $f_j$ then $g = e$ in sigma, and the same calculations show that $\ell_?(e)$ decreases by one. In any case all other measures remain the same, and by monotonicity of the measure we get the result. □

**Theorem 16** *The reduction* $\overset{ec}{\to}$ *is strongly normalizing.*

**PROOF.** Let $\mathrm{d}(\pi)$ be the depth of a net $\pi$, and $\mathrm{con}_0(\pi)$ and $\mathrm{coc}_0(\pi)$ be the sets of respectively contractions and cocontractions at exponential depth 0 in $\pi$. Moreover given a contraction cell $C$ let $\mathrm{ar}(C) := n$ if $C$ is an $n$-contraction. Define the multiset of natural numbers $\mathrm{p}(\pi)$ by induction on the depth of $\pi$. If $\pi$ is a sum let $\mathrm{p}(\sum_i \lambda_i) := \sum_i \mathrm{p}(\lambda_i)$, if it is a simple net $\lambda$ let

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

Note that here the convolution product sums over $\mathbb{N}$. Moreover let $\mathrm{aux}(\pi)$ be the total number of auxiliary ports of boxes in $\pi$.

We now assign to each net $\pi$ the measure $\big(|\pi|\,, \mathrm{p}(\pi), \mathrm{aux}(\pi)\big)$, and show it decreases strictly for all reductions $\pi \overset{ec}{\to} \sigma$. For p to decrease, it suffices that there is some simple net $\mu$ in the structure of $\pi$, in the sense that either $\mu$ is an addendum of $\pi$ or an addendum of some box content, such that $p$ decreases for $\mu$, while the rest of $\pi$ remains unchanged.

- If we $e$-reduce, then $|\sigma| < |\pi|$.
- If we $a$-reduce a cocontraction associative redex, then $|\sigma| \le |\pi|$, and if $\lambda$ is the smaller simple net in $\pi$ containing the cocontractions and $\mu$ the corresponding simple net in $\sigma$, we have $\#\,\mathrm{coc}_0(\mu) < \#\,\mathrm{coc}_0(\lambda)$ and the rest is unchanged.
- If we $a$-reduce a contraction associative redex, then $|\sigma| \le |\pi|$. If $\lambda$ and $\mu$ are as above for the cocontraction case, $d = \mathrm{d}(\lambda) = \mathrm{d}(\mu)$ and the two contractions in $\lambda$ have ar equal to $n$ and $k$ then the merged contraction (if any) has ar equal to $n + k - 1$, and we have

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

  while the rest is unchanged. The degenerated case where $n + k - 1 = 1$ is trivial.
- If we $p$-reduce a push redex, then $|\sigma| \le |\pi|$. If $\lambda$ and $\mu$ 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 $\lambda$ (resp. in $\mu$), $d + 1$ is the depth of $\lambda$ and $\mu$ (so that all addenda of $D$ have depth less or equal than $d$), $n + k$ with $k \ge 2$ is ar of the contraction, then in $\mu$ the contraction left out (if any) has ar equal to $n + 1$ and all addenda in $D$ get

a pushed contraction of ar equal to $k$. Summing up:

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

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

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

- If we $p$-reduce a pull redex, then $|\sigma| \leq |\pi|$, and also $\mathrm{p}(\sigma) = \mathrm{p}(\pi)$, as 0-contractions simply do not contribute in any way to p. However trivially $\mathrm{aux}(\sigma) < \mathrm{aux}(\pi)$. □

We end the discussion by completing the confluence results exactly as done in the previous section.

**Theorem 17** *Reductions* $\xrightarrow{c}$, $\xrightarrow{ec}$ *and* $\xrightarrow{mec}$ *are confluent.*

**PROOF.** *c* and *ec* reductions are settled by Newman's Lemma and Lemma 14. As Lemma 13 is still valid substituting *ec* to *ea*, we get by Huet's and Hindley-Rosen's Lemmas that also *mec* is confluent. □

## 4 Full resource calculus

In this section we will redefine Boudol's $\lambda$-calculus with multiplicities [1] extending it with sums and two kinds of non lazy reduction. The substitutions employed are those found in differential $\lambda$-calculus [11], most notably the linear one. As nets presented in the previous section added promotion to DINs ofčitediffnet, 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: $\lambda$-calculus with multiplicities*

Let $\mathbb{V}$ be a countable set of variables, and let $\Delta_k$ be the increasing sequence of sets given by induction as $\Delta_0 := \mathbb{V}$, and $\Delta_{k+1}$ 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 $k^{\text{th}}$ set of *bags of arguments*, is $\mathcal{M}_{\text{fin}}(\mathbb{A}_k)$, where furthermore $\mathbb{A}_k$, the $k^{\text{th}}$ set of *arguments*, is generated by

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

Finally, the set $\Delta$ 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 $\left(\sum_{t\in\Delta} c_t \cdot t\right)^\infty$ is called *boxed* or *exponential*, otherwise it is *linear*.

As in Section 2, 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 $\alpha$-conversion, that is renaming of variables bound by $\lambda$. We write $x \in t$ to mean "$x$ appearing free in $t$" for $t$ term [8] . A *context* is a differential term or bag that uses a distinguished variable called its *hole* exactly once.

The resource calculus presented in Section 2 is clearly embedded in full resource calculus: it corresponds to the subset of terms that have no exponential arguments. Also classical terms of $\lambda$-calculus can be embedded in this calculus by the following mapping:

$$x^* := x, \quad (\lambda x.t)^* := \lambda x.t^*, \quad \big((s)t\big)^* := \langle s^* \rangle \, (t^*)^\infty.$$

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

In order to define an operational semantics we have to define the substitution operator, which as in differential $\lambda$-calculus takes different forms. 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:

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

[8] If in $R$ there is $-1$ then every differential term $t$ can be written as $t + x - x$ for every $x$, so we have to be accurate with the definition. One can circumvent this problem by using an intersection of the free variables of all the possible representations as sum of the term, but as we are mainly interested in $R = \mathbb{N}$ we do not investigate such a definition.

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

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

The definition for applications and bags can be compacted into

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

Note how the linear substitution operator distributes among linear terms, and extracts a linear copy from a boxed argument if needed. This reflects the derivation property of the exponential in calculus, where given $y = y(x)$ we have

$$\frac{\partial \mathrm{e}^y}{\partial x} = \frac{\partial y}{\partial x} \cdot \mathrm{e}^y.$$

This substitution is linear in both the derivated term $u$ and the argument $t$, and it is easy to state that if $x \notin u$, then $\frac{\partial u}{\partial x} \cdot t = 0$.

Non linear and linear substitutions enjoy the same properties found in [11], though due to the simpler syntax proofs are somewhat easier. We state the result needed in the definition of the reductions.

**Lemma 18 (Schwartz)** *For $t, u, v \in R\langle \Delta \rangle$, and $x, y$ such that $y \notin u$, then*

$$\frac{\partial}{\partial x} \left( \frac{\partial t}{\partial y} \cdot v \right) \cdot u = \frac{\partial}{\partial y} \left( \frac{\partial t}{\partial x} \cdot u \right) \cdot v + \frac{\partial t}{\partial y} \cdot \left( \frac{\partial v}{\partial x} \cdot u \right).$$

*In particular if also $x \notin v$, the second addendum is equal to 0 and we have the classic Schwartz's lemma about commutation of partial derivatives.*

**PROOF.** Standard induction on $t$. The case for application works because of the way we linearize on the fly exponential arguments. □

If $u_1, \ldots, u_n$ are such that $x \notin u_i$, we write

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

which by Schwartz's lemma is well defined regardless of the order in which we write $A$.

We will use a third type of substitution, directly derived from regular substitution. The *partial substitution of $u$ for $x$ in $t$* is $t\,[x := x + u]$. It is easy to derive from the properties of regular substitution that if $x \notin u, v$, we have

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

which is an analogous of Schwartz's lemma for this substitution. We have one further commutation property between linear and partial substitution.

**Lemma 19** *If $x \notin v$, then*

$$\left(\frac{\partial t}{\partial x} \cdot u\right)[y := y + v] = \frac{\partial t\,[y := y + v]}{\partial x} \cdot (u\,[y := y + v])\,.$$

**PROOF.** For a variable $t = z$ we have from the right member (as $\frac{\partial v}{\partial x}$ gives 0)

$$\begin{aligned}\frac{\partial z\,[y := y + v]}{\partial x} \cdot (u\,[y := y + v]) = \frac{\partial (z + \delta_{y,z} \cdot v)}{\partial x} \cdot (u\,[y := y + v]) = \\ = \frac{\partial z}{\partial x} \cdot (u\,[y := y + v]) + \delta_{y,z} \cdot 0 = (\delta_{x,z} \cdot u)\,[y := y + v] = \left(\frac{\partial z}{\partial x} \cdot u\right)[y := y + v]\,.\end{aligned}$$

Abstraction and application are straightforward. □

Let us define the *generalized substitution of $a$ for $x$ in $t$*, with $a$ an argument, as

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

Then, given any bag $A = a_1 \cdots a_{\#A}$ and a variable $x \notin A$, all the above lemmas permit us to define

$$\mathrm{S}_x^{\#A}\, t \cdot A := \mathrm{S}_x\,(\cdots(\mathrm{S}_x\, t \cdot a_1)\cdots) \cdot a_{\#A}.$$

If in particular we order the substitutions by doing first the linear ones, we obtain the equality

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

We are ready to define the reductions, which as foretold in Section 2 comes in baby-step and giant-step form.

**Definition 20 ($\beta_{\mathbf{gs}}$ and $\beta_{\mathbf{bs}}$)** Giant-step $\beta$-reduction *(denoted by $\beta_{gs}$ or $\xrightarrow{g}$) is generated by*

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

Baby-step $\beta$-reduction *(denoted by $\beta_{bs}$ or $\xrightarrow{b}$) is generated by*

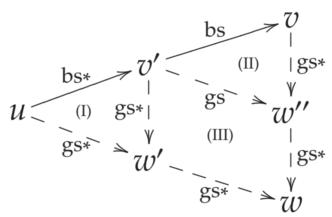
$$\langle \lambda x.s \rangle\, aA \xrightarrow{b} \langle \lambda x.\, \mathrm{S}_x\, s \cdot a \rangle\, A, \qquad \langle \lambda x.s \rangle\, 1 \xrightarrow{b} s\,[x := 0]\,.$$

Clearly there is only one way to $\beta_{\mathrm{gs}}$-reduce a redex, while there are as many ways as $\#A$ to $\beta_{\mathrm{bs}}$-reduce it. Partial substitutions break strong confluence, so we need more care for proving confluence. However, we will have it for giant-step reduction as a corollary of the bisimulation theorem in next section (Corollary 32). Supposing given that result, we will here derive from it confluence for baby-step reduction also.

**Remark 21** *Directly following the definition and commutation of substitutions we get that if* $u\ \beta_{bs}\ v$ *then there is a term* $w$ *such that* $u\ \beta_{gs}\ w$ *and* $v\ \beta^*_{gs}\ w$*, by just firing the same redex in* $u$ *and (if it still exists) in all addenda in* $v$*. Vice versa trivially* $\beta_{gs} \subset \beta^*_{bs}$*, which in turn implies* $\beta^*_{gs} \subset \beta^*_{bs}$*. The next lemma will generalize this.*

**Lemma 22** *If* $u\ \beta^*_{bs}\ v$ *then there exist a term* $w$ *such that* $u, v\ \beta^*_{gs}\ w$*.*

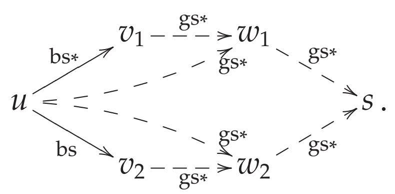
**PROOF.** By induction on the length of the reduction $u\ \beta^*_{\mathrm{bs}}\ v$. If it is zero, then take $w = u = v$ and we are done. Otherwise we have the following confluence diagram




We have (I) by inductive hypothesis, (II) by Remark 21, and (III) by confluence of $\beta_{\mathrm{gs}}$. □

**Theorem 23 (confluence of $\beta_{\mathrm{bs}}$)** *The baby-step β-reduction is confluent.*

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




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

## 5 Translation

We will now define the translation from terms and bags of full resource calculus to differential nets. In order to do so, we will label $?\imath$ conclusions of

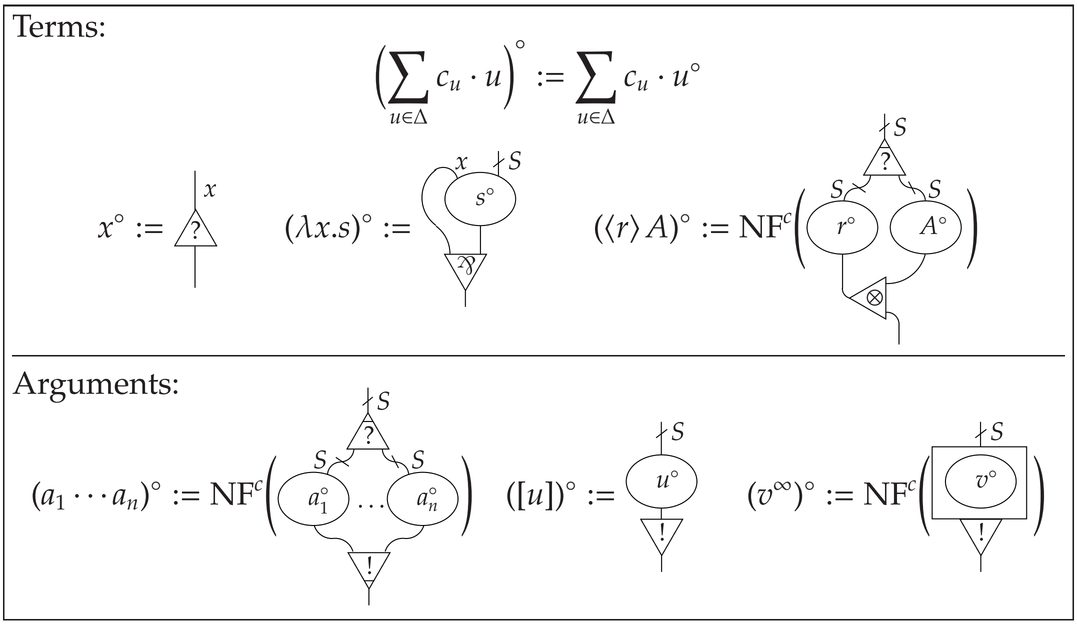



**Figure 6:** Inductive rules for the definition of $t^\circ$.

nets with different variables in $\mathbb{V}$. We will draw all nets with $?\imath$ conclusions above or left and the $o$/! conclusion down or right, and therefore wire types can and will be omitted. A wire with a bar on it will be a convention to indicate multiple wires (or possibly none), and its label will be the set of labels of the wires. In order to be able to erase or add dummy variables at will, nets will be considered equal if they differ only for conclusions introduced by weakenings.

### *5.1 Statics: definition and sequentialization*

Using the rules in Figure 5.1 for each $t$ term (resp. bag or argument) we define $t^\circ$, a labelled net with conclusions $?\imath, \dots, ?\imath, o$ (resp. $!o$) where the $?\imath$ ones are labelled by variables free in $t$. We are here denoting by $[u]$ the simple term $u$ regarded as a linear argument (in fact it is a singleton bag containing only a term). This is just to remedy the lack of an explicit constructor. Because of the equality up to weakened conclusions we may freely add conclusions, and this is used in the inductive definition. It is important to note that the translation is well defined with respect to such equality because of the pull reduction we perform on boxes. Without it when plugging nets equal modulo weakened conclusions in a box we would get different nets (even modulo weakened conclusions). Also the results we prove on bisimulation rely on the pull reduction, as weakened conclusions may appear inside boxes during the reduction.

In application, bag of arguments and boxed argument we normalize with respect to the canonical reduction in order to forget nesting and exponential depths of contractions. Due to confluence and strong normalization of $\overset{ec}{\to}$ (Theorems 16 and 17) we can freely *a* or *p*-convert (that is, pass to *a* or *p*-equivalent terms) if we have to *ec*-normalize somewhere in the future. In particular in such cases we may use the forms that in the inductive definition appear before *c*-normalization.

**Remark 24** *For every term t its translation $t^\circ$ is normal with respect to exponential and canonical reduction. Moreover it is easy to see that each redex in t corresponds exactly to a multiplicative cut in $t^\circ$. So in fact t is normal iff $t^\circ$ is normal.*

This translation is injective and surjective, once we restrict our scope to *ec*-normal nets *without exponential axioms*. These operationally have the same meaning of a boxed axiom with dereliction, i.e. they are the translation of a boxed variable [9] .

**Theorem 25 (sequentialization of *ec*-normal nets)** *For every ec-normal net $\pi$ without exponential axioms, with labels in $\mathbb{V}$ on every $?\imath$ 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 12) of every simple net in the net. Because of Lemma 1 we can take the connected component of the unique $o/!o$ conclusion, and because of correctness it is acyclic, and as such it is a tree, for which we choose the $o/!o$ 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: being the only such conclusion, we may compare two nets translating terms or bags by going up from it, and reason by induction on a term *t* different from a given term *u*, encountering either a different cell or a different label. □

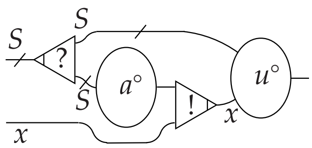
### *5.2 Dynamics: bisimulation*

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

[9] From the dynamic point of view, this notion is not stable under reduction, as the contraction against cocontraction reduction creates exponential axioms. However one can prove that a net without exponential axioms *e-normalizes* to a net with the same properties.

bisimulation. First we have to prove our version of the typical substitution lemma.

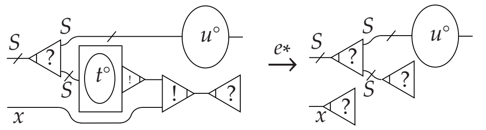
**Lemma 26 (argument substitution)** *Given an argument $a$, a simple term $u$ and a variable $x \notin a$, we have that*



$$\overset{ec}{\twoheadrightarrow} (\mathrm{S}_x(u) \cdot a)^\circ \,.$$

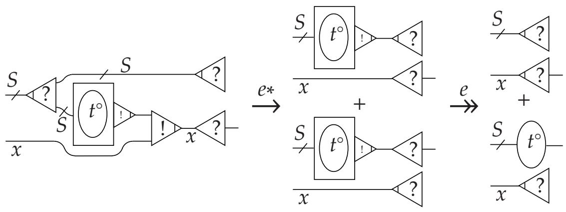
**PROOF.** Clearly we have to deal separately with the two possible types of substitution. When applying inductive hypothesis to box contents we can keep on ignoring weakened conclusions because the pull reduction brings them out.

For partial substitution, with $a = t^\infty$, let us deal first with the case in which $x \notin u$, which means that $x$ is introduced by a weakening. So:



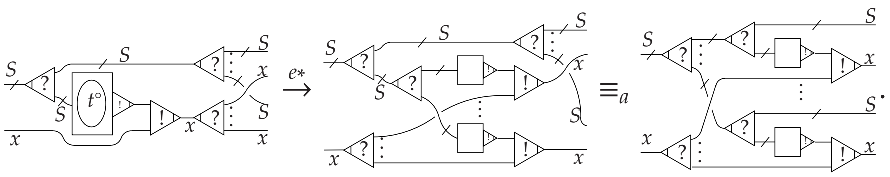
$$\overset{a}{\twoheadrightarrow} u^\circ = (u\,[x := x + t])^\circ \,.$$

So we can suppose that $x \in u$, and proceed by induction on $u$. If $u$ is a variable then it is $x$, and



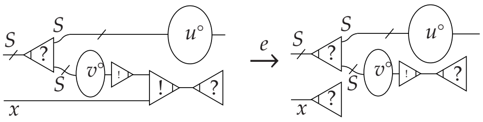
$$= x^\circ + t^\circ = (x\,[x := x + t])^\circ \,.$$

Applying the inductive hypothesis on an abstraction is trivial. For application and bags of arguments we work on the nets given in the inductive definition as they appear before $c$-normalization, as $c$-equivalence is flattened by $ec$-normalization. In order to be able to apply the inductive hypothesis, we simply have to show that contraction duplicates the whole construct leaving behind the appropriate contractions.



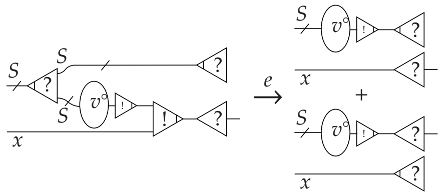
Lastly, skipping the trivial steps needed for linear argument, we need to handle boxed arguments. This is easy, as cocontraction and box simply enter $(t^\infty)^\circ$'s box, and the contractions on $S$ left behind enters because of the *push* rule.

For $a = v$ and linear substitution, again we first see what happens when $x$ is not free in $u$, in which case we have $\frac{\partial u}{\partial x} \cdot v = 0$. In fact:



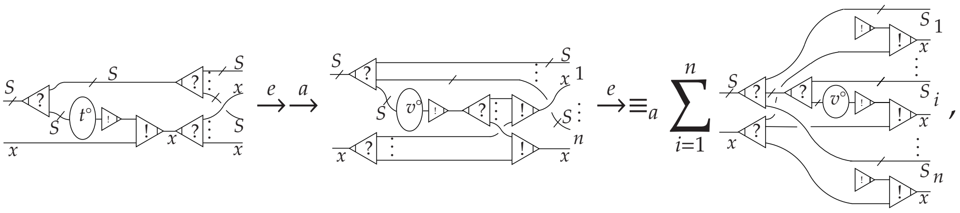
$$\overset{e}{\twoheadrightarrow} 0.$$

Next we check for the case $u = x$.

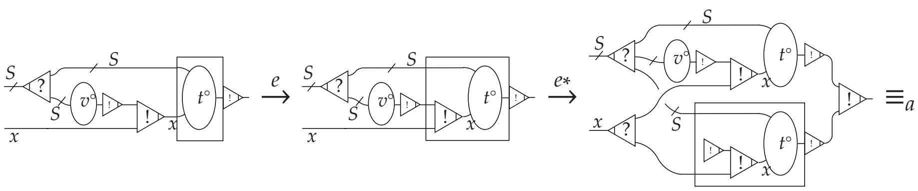


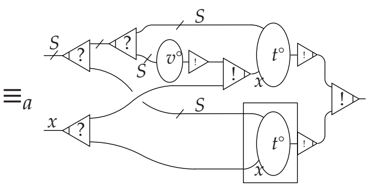
$$\overset{e}{\twoheadrightarrow} 0 + v^\circ = \left(\frac{\partial x}{\partial x} \cdot v\right)^\circ.$$

Now if we look at the definition of linear substitution for application, we again have to see what happens when cutting the construct against a contraction, and prove it reduces into a sum over the possible selections of the components of the application.



which is $a$-equivalent to what looked for. Lastly, again skipping linear application, in case of a boxed argument $t^\infty$, suppose for simplicity that $t$ is a simple term (the case for a sum follows straightforwardly). Then

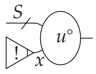




$$\overset{ec}{\to} \left(\left(\frac{\partial t}{\partial x} \cdot v\right) t^\infty\right)^\circ = \left(\frac{\partial t^\infty}{\partial x} \cdot v\right)^\circ.$$

by induction hypothesis. Associativity will take care of the leading cocontraction by merging it with the possible cocontraction of the bag containing $t^\infty$. □

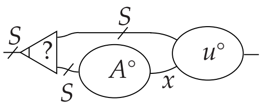
**Lemma 27 (0 substitution)** *Given a simple term u, we have that*




$$\xrightarrow{ec}\!\!\!\!\to (u\,[x := 0])^\circ\,.$$

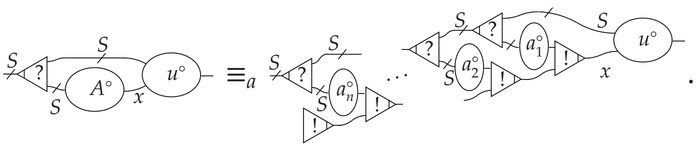
**PROOF.** Easy induction on $u$. The reductions involved are coweakening against weakening for case $x \notin u$, coweakening against dereliction for $u = x$, coweakening against contraction to apply inductive hypothesis in application and bag of arguments, and coweakening against box for an exponential argument. □

**Lemma 28 (substitution)** *If A is a bag of arguments and u is a simple term, then*




$$\xrightarrow{ec}\!\!\!\!\to \left(\mathrm{S}_x^{\#A}\, u \cdot A\right)[x := 0]\,.$$

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



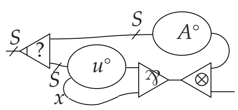

By a repeated application of Lemma 26 and a final application of Lemma 27 the above net gives as an *ec*-normal form $\left(S_x^{\#A} s \cdot A\,[x := 0]\right)^\circ$. Having used $a$-equivalence does not change the *ec*-normal form. □

We are ready for one direction of bisimulation. Note how the reduction 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.

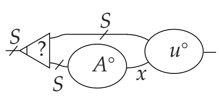
**Theorem 29 (giant-step simulation)** $s\ \beta_{gs}\ t$ *iff* $s^\circ \xrightarrow{m}\xrightarrow{ec}\!\!\!\!\to t^\circ$.

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

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




$$\xrightarrow{m}$$




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

The if implication of the above theorem can be generalized to the following lemma, which is in fact the hard part in proving the inverse direction of bisimulation.

**Lemma 30** *If* $s^\circ \overset{m*}{\to}\overset{ec}{\twoheadrightarrow} t^\circ$ *then* $s\ \beta^*_{gs}\ t$.

**PROOF.** Let $M$ be the sequence of multiplicative reductions in the reduction $R : s^\circ \overset{m*}{\to} \pi \overset{ec}{\twoheadrightarrow} t^\circ$. Let us reason by structural induction on $s$. If $s$ is a variable there is no redex and the result is trivial. Also passing to the induction hypothesis if $s = \lambda x.u$ is an abstraction is easy, as all reductions in the net cannot touch the terminal ⅋-cell which corresponds to the abstraction.

Take the case of an application $\langle r\rangle\, a_1 \cdots a_n$. First suppose that all reductions in $M$ happen either in $r^\circ$ or in either of the $a_i^\circ$s. Note we need an $a$-conversion to really speak of such subnets, which however commutes with $M$. We can partition $M$ into $L : u^\circ \overset{m*}{\to} \sigma$ and $N_i : a_i\circ \overset{m*}{\to} \tau_i$, and we can freely commute all reductions which happen in different subnets. By $ec$-normalizing the subnets we get (also employing the sequentialization theorem) $r^\circ \overset{L}{\to}\overset{e}{\twoheadrightarrow} s^\circ$ and $a_i^\circ \overset{N_i}{\to}\overset{e}{\twoheadrightarrow} b_i^\circ$. Applying these reductions on on the whole $(\langle r\rangle\, a_1 \cdots a_n)^\circ$ and commuting $L$ and $N_i$ back into their place in $M$, we get

$$(\langle r\rangle\, a_1 \cdots a_n)^\circ \overset{M}{\to}\overset{ec}{\twoheadrightarrow} (\langle s\rangle\, b_1 \cdots b_n)^\circ.$$

By inductive hypothesis $r\ \beta^*_{\mathrm{gs}}\ s$ and $a_i\ \beta^*_{\mathrm{gs}}\ b_i$ which implies the result.

Suppose now that $M$ reduces also a multiplicative cut outside of $r^\circ$ or either $a_i$. This means $s = \langle r\rangle\, A$ is itself a redex, with $r = \lambda x.u$. Reducing the corresponding multiplicative cut cannot create other multiplicative cuts, so that we can still partition $M$ into $L : u^\circ \overset{m*}{\to} \rho$, $N_i : a_i \overset{m*}{\to} \pi_i$ and the single reduction $\mu$ on the external cut. If we exclude $\mu$ from the reduction we have by a reasoning identical to the one above

$$(\langle \lambda x.u\rangle\, a_1 \cdots a_n)^\circ \overset{L}{\to}\overset{N_1}{\to} \ldots \overset{N_n}{\to}\overset{ec}{\twoheadrightarrow} (\langle \lambda x.v\rangle\, b_1 \cdots b_n)^\circ$$

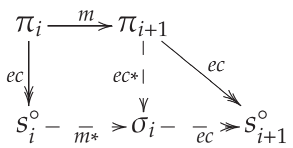
with $s\ \beta^*_{\mathrm{gs}}\ \langle \lambda x.v\rangle\, B$ (as $u\ \beta^*_{\mathrm{gs}}\ v$ and $a_i\ \beta_{\mathrm{gs}}\ b_i$).

Now in $(\langle \lambda x.v\rangle\, B)^\circ$ we execute $\mu$ and then $ec$-normalize, and by simulation theorem we get $\left(\left(\mathrm{S}^{\#A}_x\, v \cdot B\right)[x := 0]\right)^\circ$. By commuting all multiplicative reductions back into their place in $M$ and before the exponential-canonical ones, by unicity of $ec$-normal form and by injectivity of translation, we conclude that $\left(\mathrm{S}^{\#B}_x\, v \cdot B\right)[x := 0]$ (to which $s$ reduces) is equal to $t$. □

**Theorem 31 (giant-step bisimulation)** *If* $s^\circ \stackrel{mec*}{\longrightarrow} t^\circ$*, then* $s\ \beta^*_{gs}\ t$*.*

**PROOF.** Let $s^\circ = \pi_0 \stackrel{mec}{\to} \pi_1 \stackrel{mec}{\to} \ldots \stackrel{mec}{\to} \pi_n = t^\circ$ be the reduction taken into account, and consider $s_i^\circ := \mathrm{NF}^{ec}(\pi_i)$ (where we use the sequentialization theorem). By injectivity of translation $s_0 = s$ and $s_n = t$. We now prove that $s_i^\circ \stackrel{m*}{\to}\stackrel{e}{\twoheadrightarrow} s_{i+1}^\circ$, which by the above lemma implies $s_i\ \beta^*_{\mathrm{gs}}\ s_{i+1}$ and ends the proof.

If $\pi_i \stackrel{ec}{\to} \pi_{i+1}$, then $\mathrm{NF}^{ec}(\pi_i) = \mathrm{NF}^{ec}(\pi_{i+1})$ and therefore $s_i = s_{i+1}$. Suppose now $\pi_i \stackrel{m}{\to} \pi_{i+1}$. We can complete the reduction diagram in the following way:




The left confluence diagram is Lemma 13 for the *ec*-reduction, while the right triangle is simply confluence to the *ec*-normal form. The bottom side of the diagram is what we were looking for. □

**Corollary 32** *The reduction* $\beta_{gs}$ *on terms is confluent.*

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

## 6 Conclusion and future work

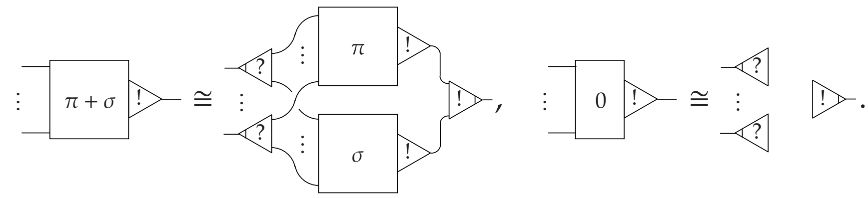
The main contributions of this paper are

- having defined pure intuitionistic differential nets, which are an extension of Ehrhard's and Regnier's differential interaction nets with LL's promotion box, and proved their "goodness" at being a rewriting system (i.e. confluence and normalization of part of its reductions);
- having fully developed the link between differential nets and a refined version of Boudol's $\lambda$-calculus with multiplicities hinted in Ehrhard's and Regnier's work, establishing between the two the same strong connection existing between proof nets and $\lambda$-calculus, and other calculi as well. We here cite as examples polarized proof nets with $\lambda\mu$-calculus [17] and stream nets with $\Lambda\mu$-calculus [20].

We now sketch some future developments or starting points for further work.

**Operational equivalence.** Just like $\lambda$-calculus and proof nets, one can find on full resource terms $\sigma$-equivalence [23,24] by equating terms having the same translation $t^\bullet$, this being defined as $\mathrm{NF}^m(t^\circ)$. This has not been treated here but should be, by proving sequentialization and (bi)simulation. We also expect to find in this translation the right simulation for the baby-step reduction, and to be able to give a translation of Ehrhard's and Regnier's differential $\lambda$-calculus.

**Exponential isomorphism.** The famous exponential isomorphism of LL and coherence spaces, for which $!A \otimes !B \cong !(A \,\&\, B)$ and $1 \cong !\top$, is clearly also valid in differential LL and finiteness spaces, where however $A \,\&\, B \cong A \oplus B$ is in fact the sum, and $\top \cong 0$. This isomorphisms spawn a specific feature of differential nets which is the following semantical and operational equivalence [10] :




In full resource calculus this translates to the equivalence on bags $(u+v)^\infty \cong u^\infty v^\infty$ and $0^\infty \cong 1$. This congruence should probably be investigated more from the syntactical and rewriting point of view.

**Propositional and second order differential MELL.** In this paper we have restricted our attention to pure intuitionistic DNs. However it is not hard to define a typed version, both in the propositional and second order fragment of LL formulas. The measure and techniques adopted in this paper can play a central role in proving strong normalization for the first order fragment of such a system. One should put more work on second order, and even more so if one wants to shift from usual correctness through acyclicity to correctness through visible acyclicity [19], which in Pagani's ongoing work is giving promising results in capturing the nature of differential nets and their main semantics, the finiteness spaces.

**Lazy head reduction in differential nets.** Viewing this paper from the perspective of $\lambda$-calculus with multiplicities, we can say we have given a translation that has ferried from differential nets to Boudol's calculus a more algebraic reduction, extending the non-deterministic and lazy head one adopted in his work. It could be fruitful to try the reverse procedure:

[10] In light of the parallel between boxes and the exponential $\mathrm{e}^x$ (see footnote 2 on page 7), this even more shows the link between the exponential isomorphism and the equalities $\mathrm{e}^{x+y} = \mathrm{e}^x\mathrm{e}^y$ and $\mathrm{e}^0 = 1$, which were one of the reasons in the first place for Girard to call the ! modality exponential.

define a non-deterministic and lazy head reduction on differential nets that restricted to the translation would reflect Boudol's one. Such a reduction would reduce only cuts at depth 0, and discard all addenda but one in cases where usually one has a sum, but should avoid taking choices that surely end up in a 0 addendum. The perception is that in the reduction of differential nets with their current syntax there are a lot of unnecessary reductions to 0 that pose difficulties to viewing it as a paradigmatic system for non-determinism. Maybe a different notion of reduction along the line of Boudol's one would help in this way.

## 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. 1993.

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

[3] R. D. Cosmo and S. Guerrini. Strong normalization of proof nets modulo structural congruences. *LNCS*, 1631:75–??, 1999.

[4] R. D. Cosmo and D. Kesner. Strong normalization of explicit substitutions via cut elimination in proof nets. In *LICS*, page 35. IEEE Computer Society, 1997.

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

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

[7] V. Danos and L. Regnier. Local and asynchronous beta-reduction (an analysis of Girard's execution formula). In *LICS*, pages 296–306, 1993.

[8] R. Di Cosmo, D. Kesner, and E. Polonovski. Proof nets and explicit substitutions. *MSCS*, 13(3):409–450, jun 2003.

[9] T. Ehrhard. On köthe sequence spaces and linear logic. *MSCS*, 12:579–623, 2002.

[10] T. Ehrhard. Finiteness spaces. *Mathematical. Structures in Comp. Sci.*, 15(4):615–646, 2005.

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

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

[13] J.-Y. Girard. Linear logic. *Th. Comp. Sc.*, 50:1–102, 1987.

[14] G. Gonthier, M. Abadi, and J.-J. Lévy. The geometry of optimal lambda reduction. In *Proceedings of the 19$^{th}$ Annual ACM Symposium on Principles of Programming Languages*, pages 15–26. Association for Computing Machinery, ACM Press, 1992.

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

[16] Y. Lafont. From proof nets to interaction nets. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*, pages 225–247. Cambridge University Press, 1995.

[17] O. Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Th. Comp. Sc.*, 290(1):161–188, Jan. 2003.

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

[19] M. Pagani. Acyclicity and coherence in multiplicative and exponential linear logic. volume 4207 of *Lecture Notes in Computer Science*, pages 531–545. Springer Berlin / Heidelberg, Sept. 2006.

[20] M. Pagani and A. Saurin. Translating $\Lambda\mu$-calculus into SIN. Submitted for publication, Dec. 2007.

[21] M. Pagani and L. Tortora de Falco. Strong normalization property for second order linear logic. Submitted for publication, 2007.

[22] 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`.

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

[24] L. Regnier. Une équivalence sur les lambda-termes. *Th. Comp. Sc.*, 126:281–292, 1994.