The Inverse Taylor Expansion Problem in Linear Logic

Michele Pagani
michele.pagani@pps.jussieu.fr

Christine Tasson
christine.tasson@pps.jussieu.fr

Laboratoire Preuves, Programmes et Systèmes
Université Paris Diderot – Paris 7

Abstract

*Linear Logic is based on the analogy between algebraic linearity (i.e. commutation with sums and scalar products) and the computer science linearity (i.e. calling inputs only once). Keeping on this analogy, Ehrhard and Regnier introduced Differential Linear Logic (*DiLL*) — an extension of Multiplicative Exponential Linear Logic with differential constructions. In this setting, promotion (the logical exponentiation) can be approximated by a sum of promotion-free proofs of* DiLL*, via Taylor expansion.*

We present a constructive way to revert Taylor expansion. Precisely, we define merging reduction *— a rewriting system which merges a finite sum of* DiLL *proofs into a proof with promotion whenever the sum is an approximation of the Taylor expansion of this proof. We prove that this algorithm is sound, complete and can be run in non-deterministic polynomial time.*

Introduction

In the 80's, Girard [7] introduced linear logic (LL) — a refinement of intuitionistic and classical logics. One particularity of LL is to be equipped with a pair of dual modalities (the *exponentials* ! and ?) which give a logical status to the operations of erasing and copying data.The idea is that linear proofs (i.e. proofs without exponentials) correspond to programs which call their arguments exactly once, whilst exponential proofs call their arguments at will. The study of LL contributed to unveil the logical nature of resource consumption and initiated a foundational comprehension of resource-related runtime properties of programs.

Linear logic makes an extensive use of jargon borrowed from vector spaces and analysis: linear, dual, exponential, etc. Indeed, at the very start of LL, there was the fundamental intuition that programs should be modeled as analytic functions and approximated by polynomials, representing bounded (although possibly non-linear) computations. This idea can be realized if one succeeds in interpreting a type as a collection A of bits of information and a datum of type A as a vector $\vec{a} = \sum_{a\in A} m_a a$, where each scalar m_a "counts" the multiplicity of the bit a in $\vec{a}$ (see [8]).

Interpreting formulæ of LL as vector spaces is not straightforward, because exponentials generate infinite dimensional spaces. For this reason, the vector spaces must be endowed with a topology yielding a suitable notion of converging sum [1]. In [9, 10] the fundamental intuition of LL becomes concrete. In these models, programs that use their arguments exactly once are interpreted as continuous linear functions and programs that can call their arguments infinitely often are analytic functions. Moreover, analytic functions can be approximated by polynomials through *Taylor expansion* [2]. This approach is possible thanks to the presence of a derivative operator. A natural question then arose: what is the meaning of such a derivative from the logical viewpoint? Ehrhard and Regnier answered to this question, introducing the *differential linear logic* (DiLL, [5]), and its functional fragment: the *differential λ-calculus* [4].

In LL, only the promotion rule introduces the ! modality. The semantics of this rule is the exponentiation of Ehrhard [2]. Operationally, the promotion creates inputs that can be called an unbounded number of times. In DiLL three more rules handle the ! modality (*codereliction*, *cocontraction* and *coweakening*) that are the duals of the LL rules dealing with the ? modality (*dereliction*, *contraction* and *weakening*). In particular, codereliction expresses in the syntax the semantical derivative: it makes available inputs of type $!A$ that must be called exactly once, so that executing a program f on a "coderelicted" input x amounts to calculate the best linear approximation of f on x. Notice that this imposes non-deterministic choices — if f is made of several subroutines each of them demanding for a copy of x, then there are different executions of f on x, depending on which subroutine is fed with the unique available copy of x. Thus we have a formal sum, where each addendum represents a possibility. This sum has a canonical mathematical interpretation — it corresponds to the sum obtained by computing the derivative of a non-linear function.

As expected, the Taylor expansion can be imported in the syntaxic realm by iterating differentiation [6]. A proof of LL can be approximated by finite sums of promotion-free proofs of DiLL. The principle is to decompose a program into a sum of purely "differential programs", all of them containing only bounded (although possibly non-linear) calls to inputs. Understanding the relation between a program and its Taylor expansion might be the starting point of renewing the logical approach to the quantitative analysis of computation started with the inception of LL.

A first question is to understand whether a finite sum of DiLL proofs approximates an LL proof. This paper tackles this question with an algorithm computing the proofs that are approximated by a *finite* DiLL sum. There are DiLL proofs that do not appear in the Taylor expansion of the same LL proof, in some sense they are not *coherent*. One should think to the addenda of a DiLL sum as parallel threads of a computation, the sum converges whenever these threads can be joined up into a sequential computation, represented by a LL proof. Our algorithm takes a finite sum $\sum_i \alpha_i$ of DiLL proofs as inputs, runs a rewriting reduction, namely the *merging reduction*, and returns a LL proof π or falls in a deadlock. We prove that this algorithm is complete (Th. 1) and sound (Th. 2): π is reached if, and only if, $\sum_i \alpha_i$ is in the Taylor expansion of π. The algorithm is non deterministic (a finite $\sum_i \alpha_i$ can appear in the Taylor expansion of several LL proofs) and can be run in non-deterministic polynomial time (Corollary 3).

The syntax of nets. We represent LL proofs as graphs called *ll-nets* (Def. 1). In [7] ll-nets are called *proof structures*. The distinction between proof structures and proof nets (the logically correct proof structures) plays no role in this paper: we will thus omit to speak of any correctness criterion. Besides, we consider only cut-free ll-nets. We adopt the syntax of [13] with generalized contractions and atomic axioms. In addition we have coweakenings, needed to define the informative order of Sect.2 and to state our main theorems (Th.1, 2). Concerning DiLL, we represent its proofs as *polynets*, which are sets of *simple nets* (Def.2). In general a *polynet* is a linear combination of simple nets with coefficients in a field of scalars. Since coefficients are irrelevant w.r.t. our questions, we omit them and define polynets as the sets which correspond to the supports of the linear combinations.

Outline. Section 1 defines the Taylor expansion of ll-nets into polynets (Def.5). In Section 2, we define *labelings* (Def.6), an equivalent but more local way to deal with boxes. We present our rewriting system, the terms, called *merging triples* (Def.13), and the reduction over them, called *merging reduction* (Def.12). We prove that the merging reduction is non-deterministically polynomial (Cor. 3), complete (Th. 1) and sound (Th. 2) with respect to the Taylor expansion.

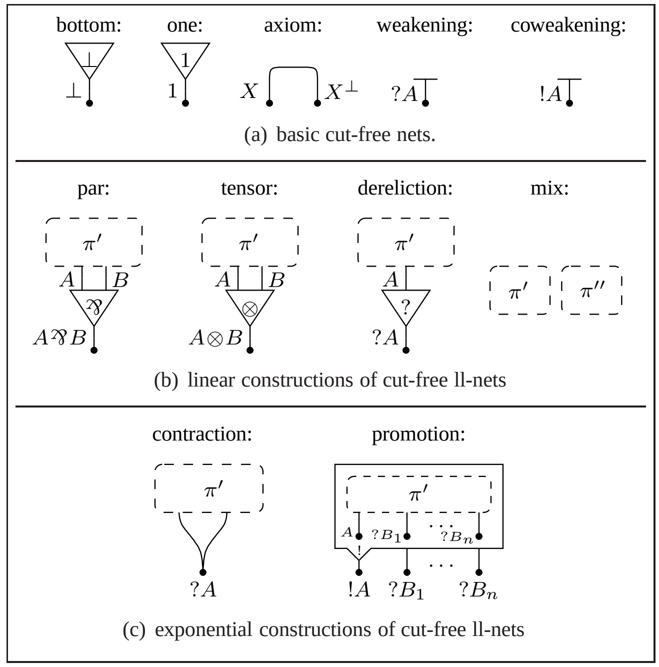


Figure 1: inductive definition of cut-free ll-nets.

1 Taylor expansion: from ll-nets to polynets.

We consider formulæ of propositional multiplicative exponential linear logic (MELL), generated by the grammar:

$$A, B := X \mid X^\perp \mid 1 \mid A \otimes B \mid \perp \mid A \parr B \mid !A \mid ?A\,,$$

where $X, X^\perp$ range over an enumerable set of propositional variables. The linear negation is involutive, i.e. $A^{\perp\perp} = A$, and defined through De Morgan laws $1^\perp = \perp$, $(A \otimes B)^\perp = A^\perp \parr B^\perp$ and $(!A)^\perp = ?A^\perp$.

Definition 1. The cut-free linear logic nets, **ll-nets**[1] for short, are inductively defined by the constructions drawn on Figures 1(a), 1(b) and 1(c), supposing that π' and π'' are cut-free ll-nets. They are finite hypergraphs made of (i) nodes labeled by MELL formulæ and called **ports**; (ii) directed hyperedges labeled by MELL connectives, depicted as triangles and named **cells**; (iii) directed hyperedges crossing ports labeled by a same exponential formula and named **structural wires**; (iv) undirected edges called **simple wires** and crossing two ports labeled by the same formula or (only in the axiom case) labeled by dual formulas.

A cell/stuctural wire c has a unique target, named the **principal port** of c, the sources, if any, are called the **auxiliary ports** of c. We adopt the convention of depicting the directed hyperedges with a top-to-bottom orientation.

A port of an ll-net π is **free** whenever it is not crossed by any cell. We require that π is given together with an **interface** $(p_i : A_i)_{i \leq n}$ enumerating its typed free ports. The

[1] This definition is kept informal: we refer to [5, 15] for precisions.

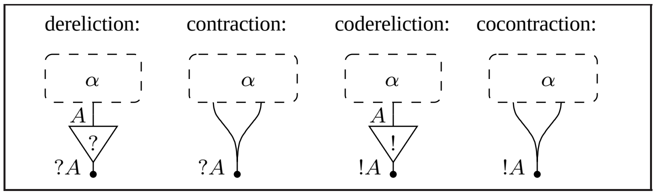
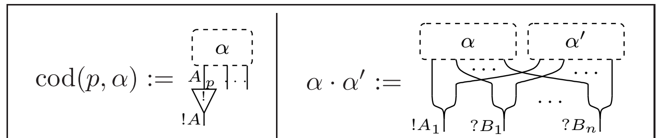


Figure 2: exponential constructions of cut-free simple nets.



(a) codereliction of a port

(b) product of two nets

Figure 3: codereliction and product of simple nets.

interfaces $(p_i : A_i)_{i\leq n}$ and $(q_i : B_i)_{i\leq m}$ are **paired** whenever $n = m$ and $A_i = B_i$.

In the contraction case (Fig. 1(c)), π' is a cut-free ll-net with at least two free ports p, q of type $?A$; to obtain the drawn ll-net, we equal p, q with a unique free port $r : ?A$ and merge the two (hyper)edges sharing r. In the promotion case, the ll-net π' is put into a **box**; this box is a cell labeled by a cut-free ll-net: its **contents**. Notice that given the box interface $(p_0 : !A, q_1 : ?B_1, \ldots, q_n : ?B_n)$, the interface of its contents is $(p'_0 : A, q'_1 : ?B_1, \ldots, q'_n : ?B_n)$ where the principal ports p_0 and p'_0 and the auxiliary ports q_i and q'_i match. We require moreover that:

$(*)$ any free port $q'_i : ?B_i$ of the contents of a box does not belong to a structural wire.[2]

The **depth** of an ll-net is the maximal number of nested boxes, it is defined by induction: the depth of any basic ll-net is 0, the depth of a box is $d+1$ whenever the depth of its contents is d, the depth of the mix of two ll-nets is the greatest depth of the used ll-nets and the other constructions do not change the depth. For any ll-net π, the set of boxes of depth 0 is denoted by $\mathrm{box}_0(\pi)$ and the set of boxes at any depth by $\mathrm{box}(\pi)$. We define similarly the sets of cells $\mathrm{cell}_0(\pi)$ and $\mathrm{cell}(\pi)$. Finally, we denote $c \in \pi$ if $c \in \mathrm{cell}(\pi)$. By extension, if b and b' are boxes, we will use $b \supseteq b'$ for $\mathrm{cell}(b) \supseteq \mathrm{cell}(b')$.

Notice that, for every $b, b' \in \mathrm{box}(\pi)$, $\mathrm{cell}(b)$ and $\mathrm{cell}(b')$ are either disjoint or included one into the other one. This means that $\supseteq$ is a tree-order over $\mathrm{box}(\pi)$, i.e. whenever b,b' have a sup then they are comparable.

As mentioned in the Introduction, boxes represent data that can be called infinitely often during the execution of a program. In DiLL new rules (cocontraction and codereliction) deal with !-formulæ but keep bounded the number of calls to the data. This allows to represent non-linear programs as *simple nets* where boxes are replaced by (co)contractions which explicitly give the number of calls to their contents.

Definition 2. The cut-free **simple nets** are inductively defined by the constructions depicted on Fig 1(a) and 1(b) and by the exponential constructions of Fig. 2. The cocontraction case is defined analogously to the contraction case. A **polynet** is a finite set of simple nets with paired interfaces.

[2] This condition is needed to have a canonical representation of ll-nets. It can be equivalently stated as: every q'_i is connected by a simple wire to a ?-cell or to an auxiliary port of another box.

Except for boxes and depth which have no meaning in the context of simple nets, we use the vocabulary of ll-nets. The word **net** will refer equally to ll-nets or simple nets.

The Taylor expansion decomposes an ll-net π into a set $\mathcal{T}(\pi)$ of simple nets; each simple net in $\mathcal{T}(\pi)$ represents an "instance" of π where every box has been replaced by a finite number of copies of its contents. Before giving the definition of $\mathcal{T}(\pi)$ (Def. 5), we need to introduce substitution (Def. 3), $\mathrm{cod}(p,\alpha)$ and product (Def. 4).

Definition 3. Let α, β and γ be three nets such that β and γ have paired interfaces $(p_i : A_i)$ and $(q_i : A_i)$. If β is a subnet of α then the **substitution** $\alpha[\gamma/\beta]$ is the net obtained from α by replacing β with γ. So, q_i replaces p_i and the wires sharing q_i are merged.

Definition 4. Let α be a simple net and p be a free port of α. We denote as $\mathrm{cod}(p,\alpha)$ the simple net obtained from α by adding a !-cell with auxiliary port p (Fig. 3(a)).

Let α and α' be two simple nets with paired interfaces resp. $(p : !A_1, q_1 : ?B_1, \ldots, q_n : ?B_n)$ and $(p' : !A_1, q'_1 : ?B_1, \ldots, q'_n : ?B_n)$. The **product** $\alpha \cdot \alpha'$ is the simple net resulting from the cocontraction of p and p' and the contractions of q_i and q'_i (Fig. 3(b)).

The product of simple nets is commutative, associative and its neutral element is the net only made of (co)weakenings and written as $!\mathbf{0}$.

Definition 5. The **Taylor expansion**[3] of an ll-net π is the set of simple nets $\mathcal{T}(\pi)$ defined by induction on the depth of π (Fig. 4(b)). We distinguish two cases according to whether π is a box b, or a generic ll-net:

$$\mathcal{T}(b) := \left\{ \prod_{j=1}^{k} \mathrm{cod}(p_j, \gamma_j) \;;\; \begin{array}{l} \text{where } k \in \mathbb{N},\ \gamma_j \in \mathcal{T}(\rho), \\ \rho \text{ is the contents of } b \text{ and} \\ p_j \text{ is the free port of } \gamma_j \\ \text{corresponding to the principal port of } b. \end{array} \right\},$$

$$\mathcal{T}(\pi) := \left\{ \pi[\beta_r/b_r]_{r\leq s} \;;\; \begin{array}{l} \text{where } \mathrm{box}_0(\pi) = \{b_r\}_{r\leq s}, \\ \rho_r \text{ is the contents of } b_r \text{ and} \\ \beta_r \in \mathcal{T}(b_r) \end{array} \right\}.$$

[3] Notice that the Taylor expansion defined by Ehrhard and Regnier [6] was defined in terms of sums of nets. Since we are only interesting in the support of these sums, our version deals with sets.

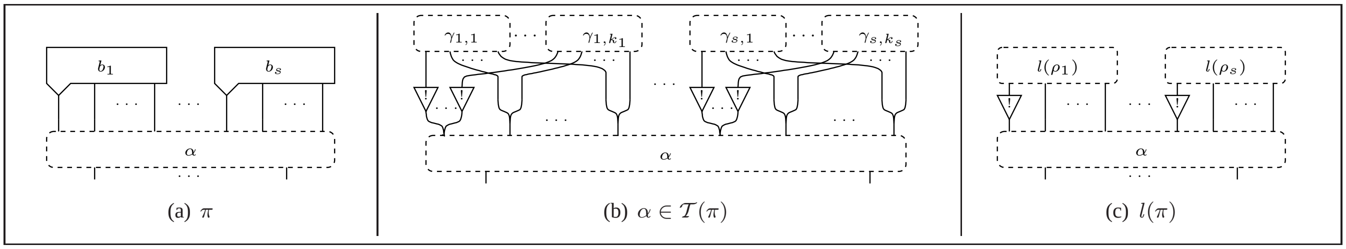


Figure 4: (a) a ll-net π s.t. $\text{box}_0(\pi) = \{b_r\}_{r\leq s}$; (b) the generic shape of a simple net $\alpha \in \mathcal{T}(\pi)$; (c) the linearization $l(\pi)$.

Not every polynet is the Taylor expansion of an ll-net. Indeed, simple nets appearing in the Taylor expansion of an ll-net π are coherent: their structure reflects the boxes of π. In Figures 5(c) and 5(d), we present an example of two incoherent simple nets $\alpha_i \in \mathcal{T}(\pi_i)$, $i = 1, 2$. However, π_1 and π_2 have the same linearization which is intuitively obtained by forgetting the contour line of boxes. More formally, the **linearization** (Fig. 4(c)) $l(\pi)$ of an ll-net π is inductively defined: $l(\pi) := \pi[\text{cod}(p_r, l(\rho_r))/b_r]_{r\leq s}$, where $\text{box}(\pi) = \{b_r\}_{r\leq s}$, ρ_r is the content of b_r and p_r is the principal free port of $l(\rho_r)$.

In the sequel $l(\pi)$ will play an important role, since it describes the structure of π except from the boxes outline. Indeed, it is a simple net of $\mathcal{T}(\pi)$, obtained by taking exactly one copy of every box of π.

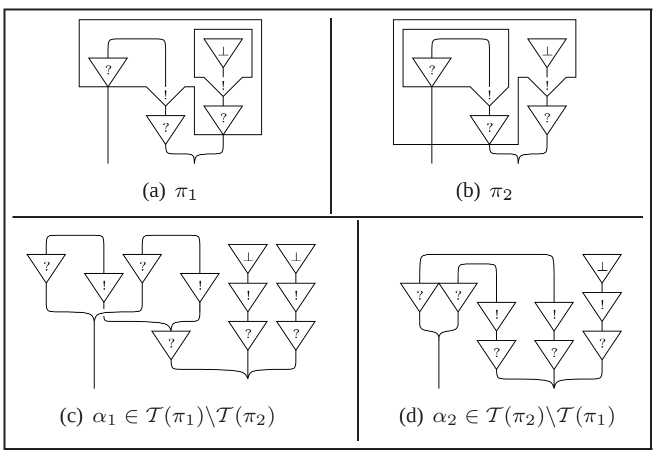


Figure 5: π_1 and π_2 with same linearization.

2 Reversing Taylor expansion:

In this section we present the merging reduction: our algorithm reversing the Taylor expansion. Given a finite polynet, the *initial state* (Def. 11, Fig. 8(b)) is obtained by plugging the simple nets into *counters* (Def. 10, Fig. 8(a)). Then these counters explore the simple nets, merge equal cells and draw boxes when it is possible. If the algorithm succeeds, then the result is an ll-net. On Fig. 9, we give the elementary reduction steps (**ers**) of the merging reduction.

2.1 An example.

Before going into more details, let us run our algorithm on an example. The rewriting is depicted step by step on Fig. 6. We draw in boldface the redex which is about to be reduced. The run we follow is successful and its result is the ll-net depicted on Fig. 5(a).

Initial state. Consider the polynet $\{\alpha_1, \alpha_2\}$, where $(p_i : ?1,\ q_i : ?!\bot)$ is the interface of α_i. The algorithm starts from the initial state depicted on Fig. 6(a). Two counters connect α_1 and α_2, one for $?1$ and one for $?!\bot$. There are two *tokens* $\mathtt{l}_1$ and $\mathtt{l}_2$ inside the counters and an *alphabet* $\{A\}$ containing an *address* $\mathtt{A} = \{\mathtt{l}_1, \mathtt{l}_2\}$ which is the set of tokens inside the counters (Def. 8).

First step. The only possibility is to apply a step $\mathtt{contr}$ to the right counter, setting $n_1 = 2, n_2 = 3$ and so $m = 2$ (see Fig. 9 for the notation). Indeed we need to choose how to distribute the three auxiliary ports of the contraction of α_2. It is a non-deterministic step of the merging algorithm: different choices may lead to non-confluent reductions. In this example, apart from the reduction we will pursue, one choice leads to the ll-net of Fig. 5(b), and the other ones fail, i.e. lead to nets with counters that are not further reducible.

Step 2. The ?-cells of the redex are merged into a unique ?-cell labelled with the address $\mathtt{A}$ (recall that it is the set of the tokens $\mathtt{l}_1, \mathtt{l}_2$ in the merging counter).

Step 3. The next redex is reduced by the "crucial" ers $!\mathtt{p}$. This step has "created" a box by adding three new tokens $\mathtt{l}_1^1, \mathtt{l}_2^1, \mathtt{l}_2^2$ and a new address $\mathtt{B} = \{\mathtt{l}_1^1, \mathtt{l}_2^1, \mathtt{l}_2^2\}$. The new tokens are associated with the coderelictions in the redex and they extend the old ones in a sense made precise in Def. 8: specifically $\mathtt{l}_1^1$ (resp. $\mathtt{l}_2^1, \mathtt{l}_2^2$) extends $\mathtt{l}_1$ (resp. $\mathtt{l}_2$). The address $\mathtt{B}$ represents a box associated with the !-cell labeled by $\mathtt{B}$ and resulting from the merging of the three coderelictions. The new address opens the possibility of applying ers of type $\xrightarrow{?\mathtt{p}}$ to the two counters inactive until now.

Step 4. While $\xrightarrow{!\mathtt{p}}$ ers creates a box adding a new address, and enters it via the principal port, a $\xrightarrow{?\mathtt{p}}$ ers enters a box already created using an address available in the alphabet (here $\mathtt{B}$) and via an auxiliary port. Notice also that a $\xrightarrow{?\mathtt{p}}$ ers can "consume" contractions (here, the counter increases the number of its auxiliary ports) but it does not merge ?-cells

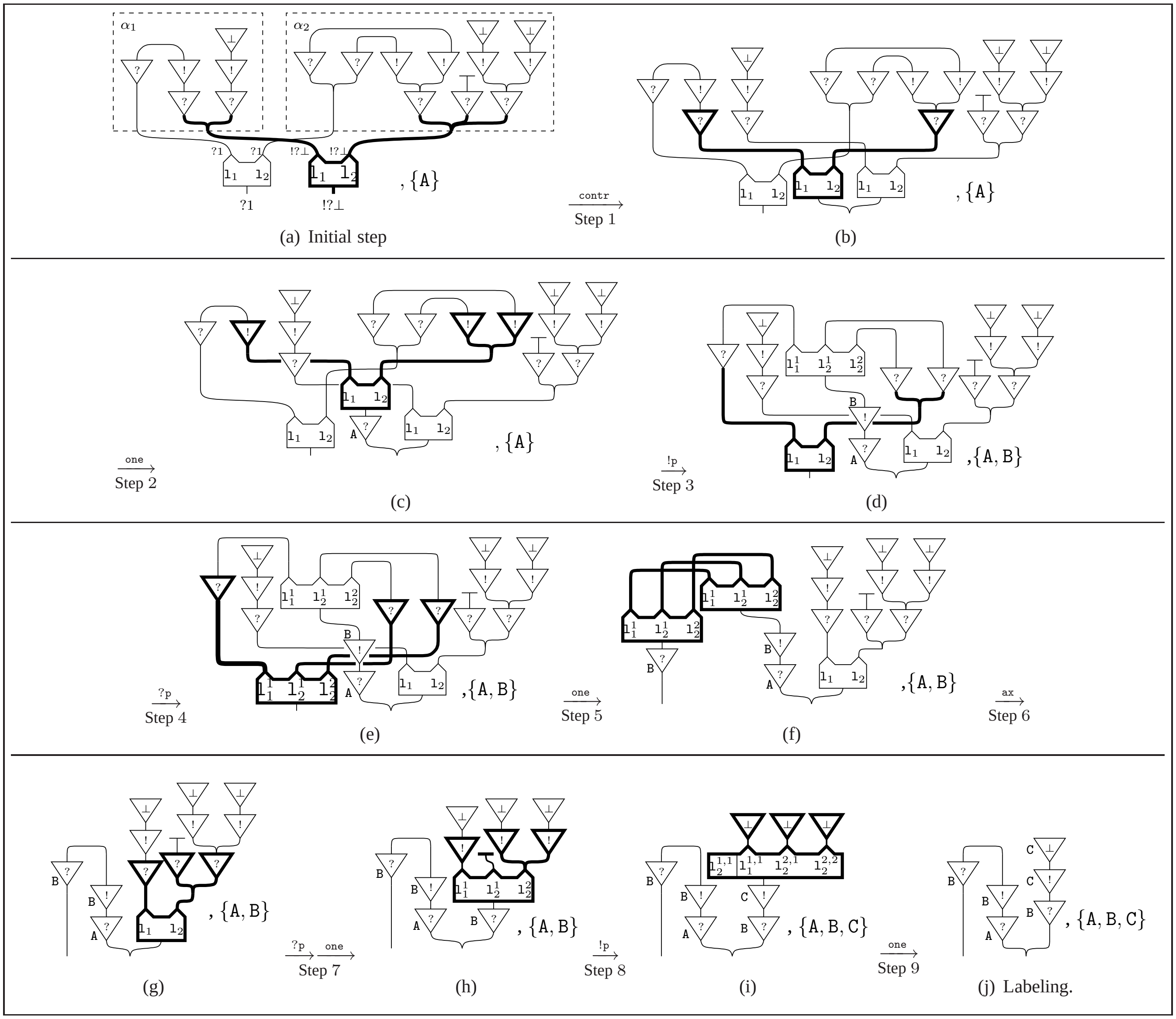


Figure 6: an example of reduction.

since these can belong to other boxes.

Step 5. The address stored in a counter after a number of $\xrightarrow{?\mathtt{p}}$ ers must be put down on a cell by a $\xrightarrow{\mathtt{one}}$ ers.

Step 6. Two counters meet and they share exactly the same address. Thus they can be eliminated by a $\xrightarrow{\mathtt{ax}}$ step.

Step 7. The ers $\xrightarrow{?\mathtt{p}}$ consumes contraction and the ers $\xrightarrow{\mathtt{one}}$ merges the $?$-cells into one $?$-cell labeled with $\mathtt{B}$.

Step 8. Remark that one port of the counter is wired to a coweakening. The $\xrightarrow{!\mathtt{p}}$ ers creates four more tokens $\mathtt{l}_1^{1,1}, \mathtt{l}_2^{1,1}, \mathtt{l}_2^{2,1}, \mathtt{l}_2^{2,2}$ associated with the coweakening/coderelictions of the redex. A new address $\mathtt{C}$ which is the set of new tokens appears. These tokens extend the old ones as hinted by the indices. The token $\mathtt{l}_2^{1,1}$, associated with the coweakening in the redex, is stored in a special basket that will be kept until the counter is erased (Def. 10).

Last step. The resulting net is a *labeling* (Def. 6). It has neither counter nor cocontraction and every cell is labelled by an address. It represents the ll-net drawn in Fig. 5(a).

In order to be as local as possible, our reduction cannot use boxes as they require to define their "frames" all in one go. Thus, we reconstruct the linearization $l(\pi)$ of an ll-net π and we represent the boxes by labeling the cells of $l(\pi)$ with addresses. A total labeling encodes exactly the boxes of π (Prop. 1). During the execution of the merging algorithm, the partial labeling is extended step by step up to a total function. The frames of the boxes of π are recovered from the addresses labeling the cells of $l(\pi)$.

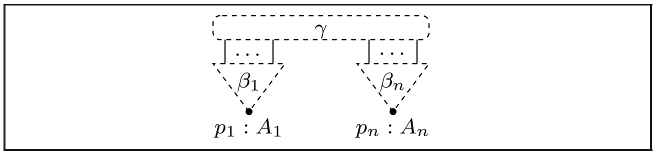


Figure 7: decomposition of a simple net.

2.2 Labeling

In our example the box associated with the !-cell labeled by $\mathtt{B}$ contains the cells labeled by $\mathtt{B}$ and everything above. Notice that the set of addresses is endowed with an order: $A \sqsubseteq B \sqsubseteq C$, which means that the box B contains the box C. Not every labeling is a correct boxing, we give conditions (Def. 6) on labelings sufficient to ensure the equivalence with ll-nets (Prop. 1).

For every cut-free net α there is only one decomposition of α into a subnet γ made of axioms and pairwise disjoint trees $(\beta_i)_{i\leq n}$ of cells and wires. The leaves of β_i can be units ($\perp$ or 1), (co)weakenings, or axioms in γ. We set $a \leq_\alpha b$ whenever a, b belong to the same tree and a is an ancestor of b. If α has more than one conclusion then there are several minimals with respect to $\leq_\alpha$. We introduce a **conclusion cell**[4] $\perp_\alpha$ set to be the minimum of $\leq_\alpha$.

Let $\mathcal{N}$ be an infinite set of **names**.

Definition 6. Let α be a cut-free simple net without cocontraction. Let $\mathfrak{L} : \{\perp_\alpha\} \cup \mathrm{cell}(\alpha) \to \mathcal{N}$ be a total function such that:

- $\mathfrak{L}$ is injective on $\mathrm{coder}(\alpha) \cup \{\perp_\alpha\}$;
- the codomain $\mathfrak{L}(\alpha)$ of $\mathfrak{L}$ is $\mathfrak{L}(\mathrm{coder}(\alpha) \cup \{\perp_\alpha\})$.

Let us define $\sqsubseteq^\circ_{\alpha,\mathfrak{L}}$ as the $\mathfrak{L}$ image of $\leq_\alpha$ onto $\mathfrak{L}(\alpha)$, that is:

$$\forall n, m \in \mathfrak{L}(\alpha),\ n \sqsubseteq^\circ_{\alpha,\mathfrak{L}} m \iff \exists c \leq_\alpha d,\ \begin{cases} n = \mathfrak{L}(c) \\ m = \mathfrak{L}(d). \end{cases}$$

Let us denote $\sqsubseteq_{\alpha,\mathfrak{L}}$ the transitive closure of $\sqsubseteq^\circ_{\alpha,\mathfrak{L}}$. The pair $(\alpha, \mathfrak{L})$ is called a **labeling** whenever

(i) $\sqsubseteq_{\alpha,\mathfrak{L}}$ is a partial tree-order, having $\mathfrak{L}(\perp_\alpha)$ as the minimum;

(ii) if $c \neq \perp_\alpha$ and c' is the predecessor cell of c, then either $\mathfrak{L}(c') = \mathfrak{L}(c)$ and c is not a !-cell, or $\mathfrak{L}(c)$ is the son of $\mathfrak{L}(c')$ and c is a !-cell, or finally c is a ?-cell;

(iii) for every axiom w, if one port of w is an auxiliary port of a cell c such that $\mathfrak{L}(c) \neq \mathfrak{L}(\perp_\alpha)$, then the other port of w is the auxiliary port of a cell c' and $\mathfrak{L}(c') = \mathfrak{L}(c)$.

From the order induced by the labeling, one can recover the contents of the box associated with a !-cell. Then a labeling and a box match if the contents of their boxes coincide.

[4]Formally, $\perp_\alpha$ is the set of free ports.

Definition 7. Let $(\alpha, \mathfrak{L})$ be a labeling. With any !-cell b of α, we associate the labeling $\mathrm{cont}(\alpha, \mathfrak{L}, b)$ corresponding to its contents. It is defined by the simple net

$$\{c \in \alpha \,|\, \mathfrak{L}(b) \sqsubseteq_{\alpha,\mathfrak{L}} \mathfrak{L}(c),\ c \neq b\},$$

and the labeling $\mathfrak{L}_{\text{cont}}(\perp) = \mathfrak{L}(b)$ and $\mathfrak{L}_{\text{cont}}(c) = \mathfrak{L}(c)$.

We say that a labeling $(\alpha, \mathfrak{L})$ is **equivalent** to an ll-net π and we write $(\alpha, \mathfrak{L}) \equiv \pi$ for short, whenever $\alpha = l(\pi)$ and

$$\forall b \in \mathrm{box}(\pi) \text{ with contents } \rho,\ \mathrm{cont}(\alpha, \mathfrak{L}, b) = l(\rho). \quad (1)$$

Proposition 1. *A labeling is equivalent to a unique cut-free ll-net and vice versa (up to a renaming).*

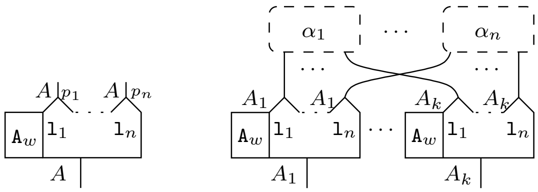
Proof. We prove that for any labeling, there is a unique equivalent ll-net by induction on the size of the simple net of the labeling (c.f. Annexe A, Prop. 6). We prove the converse (c.f. Annexe A, Prop. 7) by building a labeling candidate: the order $\sqsubseteq$ reflects the tree-order of boxes (c.f. Annexe A, Lem. 5), the labeling properties (Def 6) follow. □

Let us describe the labeling on Fig. 6. The set of names is $\mathcal{N} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}\}$. The algorithm starts with two different tokens $\mathtt{l}_1 = \{x\}$ and $\mathtt{l}_2 = \{y\}$ that are gathered in the address $\mathtt{A}$ which is the lowest element of the labeling. Each token corresponds to the lowest element of one of the simple nets. Step 3 introduces three new tokens $\mathtt{l}_1^1 = \{x, c_1\}$, $\mathtt{l}_2^1 = \{x, c_2\}$ and $\mathtt{l}_2^2 = \{x, c_3\}$ where c_1, c_2, c_3 correspond to the !-cells in boldface on 6(c). These tokens are gathered in the address $\mathtt{B} = \{\mathtt{l}_1^1, \mathtt{l}_2^1, \mathtt{l}_2^2\}$ which is used by Steps 4, 5 and 7. Finally, Step 8 introduces the tokens $\mathtt{l}_1^{1,1} = \{x, c_1, d_1\}$, $\mathtt{l}_2^{1,1} = \{x, c_2, w\}$, $\mathtt{l}_2^{2,1} = \{x, c_3, d_2\}$ and $\mathtt{l}_2^{2,2} = \{x, c_3, d_3\}$ where d_1, w, d_2, d_3 are the !-cells in boldface on 6(h). The address which corresponds is $\mathtt{B} = \{\mathtt{l}_1^{1,1}, \mathtt{l}_2^{1,1}, \mathtt{l}_2^{2,1}, \mathtt{l}_2^{2,2}\}$ is used in Step 9.

We are working with two orders: one is the order of addresses $\sqsubseteq$, encoding the structure of boxes; the other is the order of tokens (i.e. the set inclusion), encoding the structure of boxes in the simple nets appearing in the Taylor expansion. The merging reduction builds the token orders. At the same time, it checks that the boxes of each simple net are compatible and merges them inducing the addresses order. Notice that the order of addresses is an abstraction of the order of tokens, forgetting the cardinality of the latter.

2.3 Reduction

The most delicate task of merging reduction is to reconstruct a correct nesting of boxing, i.e. the order $\sqsubseteq$ of Def.6. This reconstruction is made step by step, using the set theoretical inclusion of the tokens, and the induced order $\trianglelefteq$ on addresses (Def.8): at the end of the process we will have $\trianglelefteq = \sqsubseteq$ and consenquently a ll-net.



(a) counter (b) initial state

Figure 8: counter and initial state.

Definition 8. Let X be an enumerable set called the **web**. A **token** is a finite set of elements in X. An **address** is a finite set of tokens. We set $\mathtt{l}, \mathtt{m}$ to range over tokens, $\mathtt{A}, \mathtt{B}$ to range over addresses, and $\mathcal{A}, \mathcal{B}$ to range over sets of addresses. The set-theoretical inclusion on tokens induces the following pre-order on addresses:

$$\mathtt{A} \trianglelefteq \mathtt{B} \iff \forall \mathtt{m} \in \mathtt{B}, \exists \mathtt{l} \in \mathtt{A}, \mathtt{l} \subseteq \mathtt{m}.$$

It is immediate to prove that $\trianglelefteq$ is a pre-order. However let us stress that $\trianglelefteq$ is not antisymmetric (consider $\mathtt{A} = \{\mathtt{l}, \mathtt{m}\}$, $\mathtt{B} = \{\mathtt{l}, \mathtt{m}'\}$, with $\mathtt{l} \subset \mathtt{m}, \mathtt{m}'$), nor tree-like (consider $\mathtt{A} = \{\mathtt{l}\}$, $\mathtt{A}' = \{\mathtt{m}\}$, $\mathtt{B} = \{\mathtt{l}, \mathtt{m}\}$, with $\mathtt{l}, \mathtt{m}$ disjoint) on the whole set of addresses. Indeed, *merging triple* (defined below) will handle sets of addresses on which $\trianglelefteq$ is a tree-order.

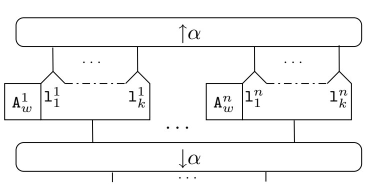
During the reduction, counters will go through the net and build a labeling.

Definition 9. A **counter** is a cell t with one principal port and $n \geq 1$ auxiliary ports. Every port of t is labeled by the same MELLformula. We consider counters as commutative cells: their auxiliary ports are interchangeable. Moreover, t is given with a labeling function λ_t which maps every auxiliary port to a token and t itself to an address (see Fig. 8(a)). We also require that for every ports p, q of t $\lambda_t(p), \lambda_t(q)$ are incomparable tokens and also are incomparable with every element of $\lambda_t(t)$.

In order to describe the partially labelled nets that appear during the reduction, we introduce triples.

Definition 10. We consider **triples** $(\alpha, \mathfrak{L}, \mathcal{A})$ made of

- an simple net α with counters which can be decomposed into two counter-free simple nets $\downarrow\alpha$ and $\uparrow\alpha$ joined by counters $t_1, \ldots, t_n$ as follows:



  We denote by $\rfloor\alpha$ resp. $\lceil\alpha$ the simple net made of $\downarrow\alpha$ resp. $\uparrow\alpha$ and the counters, and we identify the $\perp$ cells of α, $\downarrow\alpha$ and $\rfloor\alpha$;
- a set $\mathcal{A}$ of addresses and a function $\mathfrak{L}$ from cell($\rfloor\alpha$), $\perp_\alpha$ included, to $\mathcal{A}$, where for every counter t, $\mathfrak{L}(t) = \lambda_t(t) \cup \{\lambda_t(p) \mid p \text{ auxiliary port of } t\}$.

We will be interested in reductions beginning on an initial state made of counters linking the given simple nets.

Definition 11. Let $(\alpha_i)_{i\leq n}$ be a collection of simple nets with paired interfaces. Let $(\mathtt{l}_i)_{i\leq n}$ be tokens and $\mathtt{A}_w$ be an address such that the tokens in $\mathtt{A} = \mathtt{A}_w \cup \{\mathtt{l}_i\}_{i\leq n}$ are pairwise incomparable. The **initial state** associated with $(\alpha_i)_{i\leq n}$, $(\mathtt{l}_i)_{i\leq n}$, and $\mathtt{A}_w$ is the triple $(\alpha, \mathfrak{L}, \mathcal{A})$ where $\alpha = \mathrm{Init}_{\mathtt{A}_w}(\alpha_i, \mathtt{l}_i)_{i\leq n}$ is the simple net with counters pictured on Fig. 8(b) with $\mathfrak{L}(t) = \mathtt{A}$ for every counter t and $\mathcal{A} = \{\mathtt{A}\}$. In the sequel, when $\mathtt{A}_w$ is empty, we will often omit the subscript and write $\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}$.

Now we have all the ingredients to introduce a reduction $\xrightarrow{\mathtt{mrg}}$ on triples as the context closure of the binary relation $\mathtt{mrg}$ described in Fig. 9. In the interaction net paradigm [11], a redex is made of two cells wired by their principal ports. On the contrary, a merging redex is made of a counter whose *auxiliary* ports are linked to the principal ports of cells of simple nets. For this reason we represent the auxiliary ports of a counter as tips of triangles. It is important to notice that though the counters merge cells locally, the labeling process is global, whence the set of addresses appearing in the triples.

Definition 12. We consider the following unions of the **elementary reduction steps** (**ers** for short) in Fig. 9:

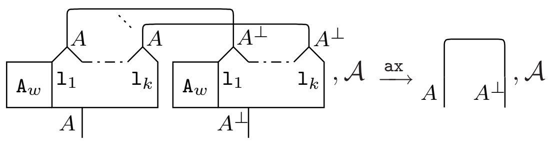
$$\begin{aligned} \mathtt{lnr} &:= \mathtt{ax} \cup \mathtt{zero} \cup \mathtt{one} \cup \mathtt{two} \cup (\mathtt{co})\mathtt{w} \cup \mathtt{contr}, \\ \mathtt{mrg} &:= \mathtt{lnr} \cup !\mathtt{p} \cup ?\mathtt{p}. \end{aligned}$$

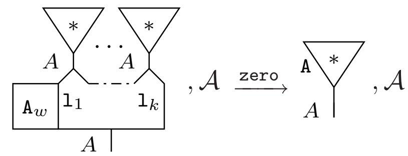
For $\mathtt{x}$ varying over $\mathtt{lnr}$, $\mathtt{mrg}$, we define the $\mathtt{x}$**-reduction** $(\alpha_1, \mathfrak{L}_1, \mathcal{A}_1) \xrightarrow{\mathtt{x}} (\alpha_2, \mathfrak{L}_2, \mathcal{A}_2)$ as the context closure of $\mathtt{x}$, more precisely it holds iff: $\exists \alpha_i' \subseteq \alpha_i$ such that $\alpha_2 = \alpha_1[\alpha_2'/\alpha_1']$, and $\forall c \in \alpha_2$, $\mathfrak{L}_2(c) = \mathfrak{L}_2'(c)$ if $c \in \alpha_2'$, otherwise (i.e. $c \in \alpha_1 \setminus \alpha_1'$) $\mathfrak{L}_2(c) = \mathfrak{L}_1(c)$, and finally $(\alpha_1', \mathfrak{L}_{1|\alpha_1'}, \mathcal{A}_1)\, \mathtt{x}\, (\alpha_2', \mathfrak{L}_2', \mathcal{A}_2)$.

We denote by $\xrightarrow{\mathtt{x}*}$ the reflexive and transitive closure of $\xrightarrow{\mathtt{x}}$. We say that $R : (\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{mrg}*} (\alpha', \mathfrak{L}', \mathcal{A}')$ is **successful** if α' is counter-free.

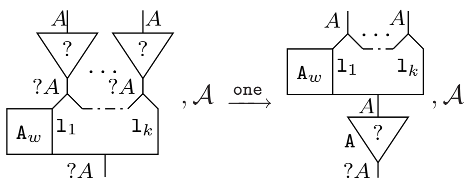
There are only three cases where a reduction falls into a deadlock, (i.e. a triple no further reducible but with counters): when two counters are linked by axioms and have different labels; when a counter is linked to one axiom and another cell; when there is no possible address to go through a contraction link for a $?\mathtt{p}$-ers.

Since we want the reduction to produce a labeling, we have to restrict the set of triples that we consider. So we introduce merging triples such that the result of a successful reduction (i.e. a counter-free merging triple) is a labeling.

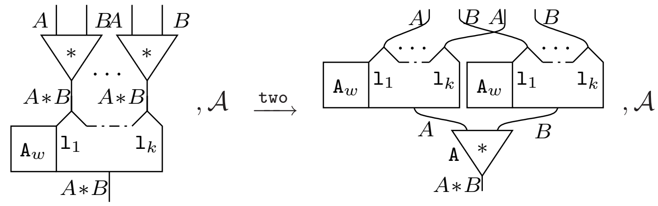




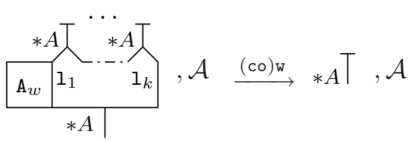
where $*$ is one between $1, \bot$; moreover $\mathtt{A} = \mathtt{A}_w \cup \{\mathtt{l}_i\}_{i \leq k}$ and $\mathtt{A} \in \mathcal{A}$.



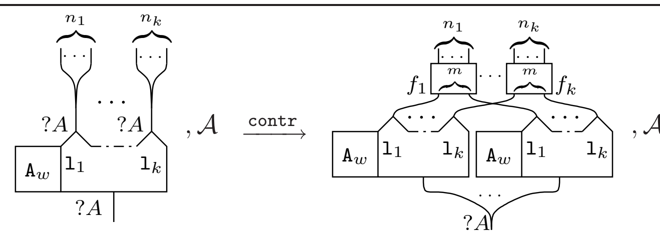
where $\mathtt{A} = \mathtt{A}_w \cup \{\mathtt{l}_i\}_{i \leq k}$ and $\mathtt{A} \in \mathcal{A}$.



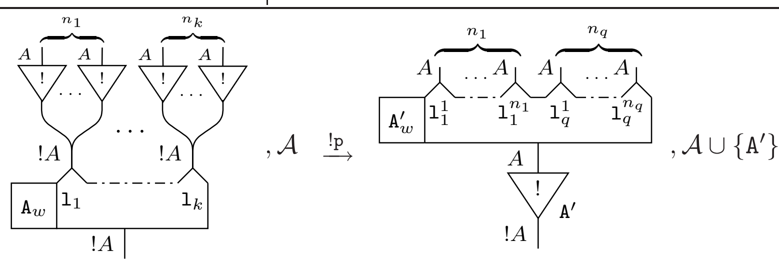
where $*$ is one between $⅋, \otimes$; moreover $\mathtt{A} = \mathtt{A}_w \cup \{\mathtt{l}_i\}_{i \leq k}$ and $\mathtt{A} \in \mathcal{A}$.



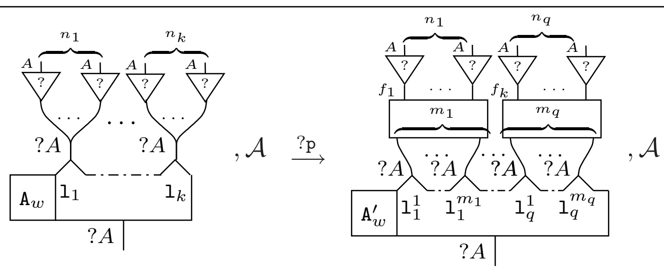
where $*$ is one between $?, !$.



where $m \geq 2$, and for each $i \leq k$, $m \leq n_i \geq 2$, and f_i is a surjection from n_i to m.



where $1 < q \leq k$ and for every $i \leq k$, $n_i = 0$ iff $i > q$; moreover, there exists a family $\{y\} \cup \{x_i^j\}_{i \leq q, j \leq n_i}$ of pairwise distinct elements of X which are fresh in $\mathcal{A}$ and such that $\mathtt{A}'_w = \mathtt{A}_w \cup \{\mathtt{l}_i \cup \{y\}\}_{q < i \leq k}$ and $\mathtt{l}_i^j = \mathtt{l}_i \cup \{x_i^j\}$.



where $1 < q \leq k$, and for every $i \leq k$, $n_i = 0$ iff $i > q$; for every $i \leq q$, f_i is a surjection from n_i to $m_i \leq n_i$; as for tokens and addresses, we ask that there exists a family $\{y\} \cup \{x_i^j\}_{i \leq q, j \leq n_i}$ of pairwise distinct elements of X such that $\mathtt{A}'_w = \mathtt{A}_w \cup \{\mathtt{l}_i \cup \{y\}\}_{q < i \leq k}$, $\mathtt{l}_i^j = \mathtt{l}_i \cup \{x_i^j\}$ and $\mathtt{A}' = \mathtt{A}'_w \cup \{\mathtt{l}_i^j\}_{i \leq q, j \leq m_i} \in \mathcal{A}$.

Figure 9: the elementary reduction steps (ers) of merging reduction; the net at left of an ers is the **redex**, that at right the **contractum** of the ers. In the $\xrightarrow{\mathtt{contr}}$, $\xrightarrow{\mathtt{?p}}$ ers, we present a bunch of contractions and wirings as a surjective function f from the auxiliary ports to the principal ones.

Then we prove that the reduction preserves the properties of the merging triples. Since the initial states are merging triples and the counter-free merging triples are labeling, we get the wanted result.

Definition 13. The triple $(\alpha, \mathfrak{L}, \mathcal{A})$ is called a **merging triple** if it satisfies

(i) for every counter $t \in \alpha$, the principal port of t is wired to a cell $c \in \downarrow\alpha$, $\bot_\alpha$ included, and $\mathfrak{L}(c) \trianglelefteq \mathfrak{L}(t)$; moreover, if $\mathfrak{L}(c) \neq \mathfrak{L}(t)$ then every auxiliary port of t is wired to a ?-cell $c \in\uparrow\alpha$, or a weakening;

(ii) $\forall \mathtt{A}, \mathtt{B} \in \mathcal{A}$, $(\mathtt{A} \sqsubseteq_{\downarrow\alpha, \mathfrak{L}} \mathtt{B} \iff \mathtt{A} \trianglelefteq \mathtt{B})$,

(iii) $(\downarrow\alpha, \mathfrak{L})$ is a labeling and $\mathfrak{L}(\downarrow\alpha) = \mathcal{A}$.

Notice that the initial state (Def. 11) is a merging triple.

As we wrote above $\trianglelefteq$ is not in general an order; however, if $(\alpha, \mathfrak{L}, \mathcal{A})$ is merging, then Cond. (ii), (iii) guarantees that $\trianglelefteq$ is a tree-order on $\mathcal{A}$.

It is very important to notice that the property of being a merging triple is stable under merging reduction (c.f. Prop. 8 of Annexe B). As a consequence, if a reduction $(\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{mrg}*} (\alpha', \mathfrak{L}', \mathcal{A}')$ is successful, i.e. α' is counter free, then $(\alpha', \mathfrak{L}') = (\downarrow \alpha', \mathfrak{L}')$ is a labeling and so represents an ll-net π (Prop. 1). In this case, we say that the reduction **leads to** π and we write

$$(\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{mrg}*} \pi.$$

Proposition 2. *The number of ers of any $\xrightarrow{\mathtt{mrg}}$-reduction starting from a merging triple $(\alpha, \mathfrak{L}, \mathcal{A})$ is polynomially bounded by the number of ports in α.*

Proof. c.f. Annexe B, Prop. 2 ☐

The ers of Fig. 9 are local and can be implemented on a Turing Machine in constant time. Thus Prop. 2 yields:

Corollary 3. *The runtime of any $\xrightarrow{\mathtt{mrg}}$-reduction starting from a merging triple $(\alpha, \mathfrak{L}, \mathcal{A})$ is polynomial in the number of ports in α.*

2.4 Completeness and soundness

We prove the completeness and soundness of merging reduction. The completeness ensures that simple nets coming from the Taylor expansion of a same ll-net can be merged. The soundness theorem proves the converse: if there is a successful reduction merging simple nets in an ll-net, then the formers are in the Taylor expansion of the latter.

Since our system considers *finite* subsets $\{\alpha_i\}_{i\leq n}$ of the Taylor expansion of an ll-net π, then some boxes of π can remain undefined from the merging of $\{\alpha_i\}_{i\leq n}$. Formally this means that the merging yields an ll-net that is the result of replacing some boxes of π with $!0$, i.e. with (co)weakenings.

Definition 14. We say that an ll-net π' is **less informative** than an ll-net π and we write $\pi' \ll \pi$, whenever there are boxes $(b_r)_{r\leq s}$ in $\mathrm{box}(\pi)$ such that $\pi' = \pi[!0/b_r]_{r\leq s}$.

It is easy to check that $\ll$ is an order. Intuitively, π' is the result of erasing some subroutines of π. In general a finite subset of $\mathcal{T}(\pi)$ does not have enough information to build π and we will rather build $\pi' \ll \pi$. However,

$$\pi' \ll \pi \Rightarrow \mathcal{T}(\pi') \subseteq \mathcal{T}(\pi). \tag{2}$$

Theorem 1 (Completeness)**.** *Let π be a ll-net, and let $\alpha_1, \dots, \alpha_n$ be simple nets in $\mathcal{T}(\pi)$. For any family $\{\mathtt{l}_i\}_{i\leq n} = \mathtt{A}$ of pairwise distinct tokens, there exists an ll-net $\pi_0 \ll \pi$ and a successful reduction that leads to π_0:*

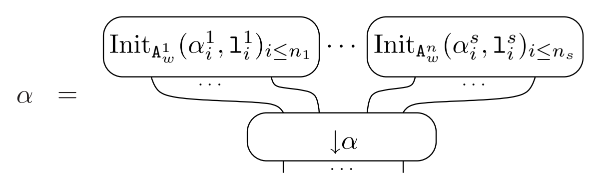
$$(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} \pi_0,$$

where $\mathfrak{L}_{\mathtt{A}}$ is, as usual, the constant function taking value $\mathtt{A}$.

Proof. The proof is by induction on the exponential depth of π; we split the induction step in two cases: if π is a box, we choose tokens extending the initial ones and gather them in an address $\mathtt{B}$. We use the induction case with $\mathtt{B}$ and conclude by context closure; in the general case, we make counter go through the linear part (c.f. Annexe C.1, Lem. 3) and stop at the entrance of boxes. We use the one box case and conclude by context closure (c.f. Annexe C.2, Th. 1). ☐

To prove the soundness theorem, we need a splitting lemma which decomposes $\uparrow\alpha$ in initial states (Def. 11).

Lemma 4 (Splitting)**.** *Let R be a successful reduction sequence from a merging triple $(\alpha, \mathfrak{L}, \mathcal{A})$ and s.t. no ers of R enters an address labeling a counter of α; then α can be split: there are suitable sequences $(\alpha_i^r)_{r\leq s,\, i\leq n_r}$ of simple nets, $(\mathtt{l}_i^r)_{r\leq s,\, i\leq n_r}$ of lists and $(\mathtt{A}_w^r)_{r\leq s}$ of addresses s.t.*



Proof. First we prove that since R is successful, two counters of the same connected component of $\uparrow\alpha$ have the same label (c.f. Annexe C.3, Lem. 5-6). With each label $\mathtt{A}$, we associate the subnet β made of cells connected to a counter labelled by $\mathtt{A}$. Then, two connected auxiliary ports of counters have the same label (Lem. 6). This allows to decompose β into an initial state (c.f. Annexe C.3, Lem. 4). ☐

Theorem 2 (Soundness)**.** *Let π be an ll-net, let $(\alpha_i)_{i\leq n}$ be a family of simple nets with the same interface and let*

$(\mathtt{l}_i)_{i \leq n} = \mathtt{A}$ *be a family of pairwise incomparable tokens. If there is a successful merging reduction leading to* π*:*

$$(\text{Init}(\alpha_i, \mathtt{l}_i)_{i \leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} \pi \tag{3}$$

then for every $i \leq n$, $\alpha_i \in \mathcal{T}(\pi)$.

Proof. The proof is by induction on the exponential depth of π. The main idea is to commute the ers in the reduction to π and to gather the ers that enter a box (c.f. Annexe B, Fact 3). Thanks to the splitting Lemma 4, we get an initial state for each box. Thus, we can apply the induction hypothesis (c.f. Annexe C.3, Th. 2). □

3 Perspectives

The merging reduction could have been presented differently, gathering all the non-deterministic choices in a single initial step and then performing the "deterministic" part of the reduction. This amounts to choose a labeling $\mathfrak{L}$ on the upper part $\uparrow \alpha$ of an initial state and to transform the merging reduction in a rewriting that checks deterministically whether $\mathfrak{L}$ is correct. However we have preferred the most local presentation of the reconstruction of the boxes.

In LL, the difficulty in reversing Taylor expansion lies in the uniformity of polynets approximations of an ll-net. A polynet is *uniform* whenever it appears in the Taylor expansion of the same ll-net, that is when there is a successful run of our algorithm. In λ-calculus, uniformity is characterized by a binary coherence relation [6]. In DiLL, the scene is far more complex. The uniformity is a hypercoherence: there are examples of non-uniform polynets made of simple nets pairwise (cf. Annexe D). This can be linked with the Gustave function. Ehrhard and Laurent have used the sum constructor in DiLL to model concurrent computing [3]. The uniformity question opens a logical approach to a typical problem of synchronization: how to link up multiple processes.

The informative order (2) (see Sect. 2.4) suggests a natural question: Is the Taylor expansion injective? Till now, the first author and Mazza [12] have shown the equivalence between the injectivity of the Taylor expansion and the injectivity of the LL relational semantics for cut-free LL nets. This is an open problem first addressed in [14]. Our results allow to deduce the injectivity from a confluence property of the merging reduction:

($*$) for every cut-free ll-net π', there are simple nets $(\alpha_i)_{i \leq n} \in \mathcal{T}(\pi')$ s.t. every successful reduction leads to π': $\text{Init}(\alpha_i)_{i \leq n} \xrightarrow{\mathtt{mrg}*} \pi'$.

First, ($*$) gives $\mathcal{T}(\pi') \subseteq \mathcal{T}(\pi) \Rightarrow \pi' \ll \pi$, the converse of (2). Let π' and π be ll-nets s.t. $\mathcal{T}(\pi') \subseteq \mathcal{T}(\pi)$. Assume ($*$): there are $(\alpha_i)_{i \leq n}$ in $\mathcal{T}(\pi')$ s.t. every succesful reduction leads to π'. Since $(\alpha_i)_{i \leq n} \subseteq \mathcal{T}(\pi)$, by completeness (Th. 1) there is $\pi_0 \ll \pi$ such that $\text{Init}(\alpha_i)_{i \leq n} \xrightarrow{\mathtt{mrg}*} \pi_0$. By ($*$), $\pi_0 = \pi'$ and $\pi' \ll \pi$. The other direction comes from (2). Second, two ll-nets $\pi' \neq \pi$ satisfy either $\pi' \not\ll \pi$ or $\pi' \not\ll \pi'$, hence $\mathcal{T}(\pi') \neq \mathcal{T}(\pi)$.

Acknowledgement. We are grateful to O. Laurent for having suggested counters travelling through nets and to T. Ehrhard and P.-L. Curien for useful discussions and hints.

References

[1] M. Barr. $*$-autonomous categories and linear logic. *Math. Struct. in Comput. Sci.*, 1(2):159–178, 1991.

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

[3] T. Ehrhard and O. Laurent. Interpreting a finitary pi-calculus in differential interaction nets. In *Concurrency Theory (CONCUR '07)*, volume 4703 of *Lecture Notes in Comput. Sci.*, pages 333–348. Springer, Sept. 2007.

[4] T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theor. Comput. Sci.*, 309(1-3):1–41, 2003.

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

[6] T. Ehrhard and L. Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.*, 2008.

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

[8] J.-Y. Girard. Normal functors, power series and λ-calculus. *Annals of Pure and Applied Logic*, 37(2):129–177, 1988.

[9] J.-Y. Girard. Coherent banach spaces: a continuous denotational semantics. *Theor. Comput. Sci.*, 3:227–275, 1996.

[10] R. Hasegawa. Two applications of analytic functors. *Theoretical Computer Science*, 272(1-2):113–175, 2002.

[11] Y. Lafont. Interaction nets. In *POPL '90: Proceedings*, pages 95–108, New York, NY, USA, 1990. ACM.

[12] D. Mazza and M. Pagani. The separation theorem for differential interaction nets. In *LPAR*, volume 4790 of *Lecture Notes in Comput. Sci.*, pages 393–407. Springer, 2007.

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

[14] L. Tortora de Falco. Obsessional experiments for linear logic proof-nets. *Math. Struct. Comput. Sci.*, 13(6):799–855, 2003.

[15] L. Vaux. *λ-calcul différentiel et logique classique : interactions calculatoires*. Thèse de doctorat, Université Aix-Marseille II, 2007.

A Labeling

Lemma 5. *Let $(\alpha, \mathfrak{L})$ be a labeling and b_1, b_2 be two distinct !-cells of α. We have:*

(i) If $\mathfrak{L}(b_2)$ is a son of $\mathfrak{L}(b_1)$, then $\mathrm{cont}(\alpha, \mathfrak{L}, b_2) \subset \mathrm{cont}(\alpha, \mathfrak{L}, b_1)$.

(ii) If both $\mathfrak{L}(b_1)$ and $\mathfrak{L}(b_2)$ are sons of $\mathfrak{L}(\bot_\alpha)$, then $\mathrm{cont}(\alpha, \mathfrak{L}, b_1)$ and $\mathrm{cont}(\alpha, \mathfrak{L}, b_2)$ are disjoint.

(iii) Every auxiliary conclusion of $\mathrm{cont}(\alpha, \mathfrak{L}, b_1) \cup \{b_1\}$, that is a minimal cell with respect to $\leq_\alpha$ different from b_1, is a ?-cell.

Proposition 6. *For every labeling $(\alpha, \mathfrak{L})$ there is a unique cut-free ll-net π such that $(\alpha, \mathfrak{L}) \equiv \pi$.*

Proof. We proceed by induction on the size of the codomain of $\mathfrak{L}$. If this codomain is $\{\mathfrak{L}(\bot_\alpha)\}$, then α has no !-cells, and we can take $\pi = \alpha$. Else let $(b_r)_{r\leq s}$ be the !-cells of α such that for each $r \leq s$, $\mathfrak{L}(b_r)$ is a son of $\mathfrak{L}(\bot_\alpha)$ in the tree-order $\sqsubseteq_{\alpha,\mathfrak{L}}$. We apply the induction hypothesis on every $\mathrm{cont}(\alpha, \mathfrak{L}, b_r)$, obtaining a unique equivalent cut-free ll-net ρ_r: let b'_r be a box with contents ρ_r. Thanks to the above Lemma 5, the only possible ll-net

$$\pi = \alpha[b'_r/\mathrm{cont}(\alpha, \mathfrak{L}, b_r) \cup \{b_r\}]_{r\leq s}$$

is a well defined cut-free ll-net such that $\pi \equiv (\alpha, \mathfrak{L})$. ☐

The following proposition shows the converse.

Proposition 7. *For every cut-free ll-net π, up to renaming, there is a unique labeling $(\alpha, \mathfrak{L}) \equiv \pi$.*

Proof. Let π be a cut-free ll-net. First we define $(l(\pi), \mathfrak{L})$. For every $c \in l(\pi)$, we denote by c^π the unique cell in π corresponding to c. Let $\mathfrak{L}$ be an embedding of $\mathrm{coder}(l(\pi)) \cup \{\bot_{l(\pi)}\}$ into $\mathcal{N}$. This is the only time in this proof we make a choice: every other choice will generate the same labeling up to renaming. We extend $\mathfrak{L}$ to any other cell c of $l(\pi)$. If c^π has depth 0 in π, then $\mathfrak{L}(c) = \mathfrak{L}(\bot_{l(\pi)})$. If b is the $!d$-cell of $l(\pi)$ associated with the box b^π containing c^π at depth 0, then $\mathfrak{L}(c) = \mathfrak{L}(b)$. It remains to show that $(l(\pi), \mathfrak{L})$ is a labeling equivalent to π. This will be a simple consequence of the remark that for every $c_1, c_2 \in l(\pi)$ we have

$$\forall b^\pi \in \mathrm{box}(\pi),\ (c_1^\pi \in b^\pi \Rightarrow c_2^\pi \in b^\pi) \iff \mathfrak{L}(c_1) \sqsubseteq_{l(\pi),\mathfrak{L}} \mathfrak{L}(c_2). \quad (4)$$

Let us first prove (4). For $i \in \{1, 2\}$, let b_i^π be the boxes containing c_i^π at depth 0 and b_i be the !-cells corresponding to b_i^π in $l(\pi)$. By definition we have $\mathfrak{L}(c_i) = \mathfrak{L}(b_i)$.
First assume that for every box $b^\pi \in \mathrm{box}(\pi)$, $c_1^\pi \in b^\pi$ implies $c_2^\pi \in b^\pi$. If $b_1^\pi = b_2^\pi$, then $\mathfrak{L}(c_1) = \mathfrak{L}(c_2)$. If $b_1^\pi \neq b_2^\pi$, thanks to the box nesting, the sets $\mathrm{cell}(b_1^\pi)$ and $\mathrm{cell}(b_2^\pi)$ are either disjoint or one is contained in the other. Since $c_1^\pi \in b_1^\pi$, we have $c_2^\pi \in b_1^\pi$. Therefore, $\mathrm{cell}(b_2^\pi) \subsetneq \mathrm{cell}(b_1^\pi)$ and $b_2^\pi \in \mathrm{box}(b_1^\pi)$. Moreover, the principal port of b_2 in $l(\pi)$ is associated with the principal port of the box b_2^π. Due to the constraints on auxiliary cells of boxes (see $*$ Def. 1), there is $d \leq_{l(\pi)} b_2$, such that either $d = b_1$ or d^π is contained at depth 0 in the contents of b_1^π. We have $\mathfrak{L}(b_1) = \mathfrak{L}(d)$ by definition of $\mathfrak{L}$ and $\mathfrak{L}(d) \sqsubseteq_{l(\pi),\mathfrak{L}} \mathfrak{L}(b_2) = \mathfrak{L}(c_2)$, by definition of $\sqsubseteq$.
Second, assume that $\mathfrak{L}(c_1) \sqsubseteq_{l(\pi),\mathfrak{L}} \mathfrak{L}(c_2)$. Since $\sqsubseteq$ is the transitive closure of $\sqsubseteq^\circ$, there are two sequences $(d_i)_{i\leq k}, (e_i)_{i\leq k}$ of cells in $l(\pi)$ such that $d_1 = c_1$, $e_k = c_2$ and $\mathfrak{L}(d_1) \sqsubseteq^\circ \mathfrak{L}(d_2) = \mathfrak{L}(e_2) \sqsubseteq^\circ \mathfrak{L}(e_3) = \mathfrak{L}(d_3) \ldots \sqsubseteq^\circ \mathfrak{L}(e_k)$. For every $i \leq k$, $\mathfrak{L}(d_i) = \mathfrak{L}(e_i)$, hence there exists $b_i^\pi \in \mathrm{box}(\pi)$ such that $d_i, e_i \in b_i^\pi$. Because of box nesting, $\forall b^\pi \in \mathrm{box}(\pi)$, $d_i^\pi \in b^\pi \iff e_i^\pi \in b^\pi$. Besides, for every i odd, $d_i \leq_{l(\pi)} d_{i+1}$, $e_{i+1} \leq_{l(\pi)} e_{i+2}$. Since $l(\pi)$ is a cut-free simple net, we deduce that for each box b^π, we have $d_i^\pi \in b^\pi \Rightarrow d_{i+1}^\pi \in b^\pi$ and $e_{i+1}^\pi \in b^\pi \Rightarrow e_{i+2}^\pi \in b^\pi$. Combining these implications, we get $\forall b^\pi, c_1^\pi = d_1^\pi \in b^\pi \Rightarrow c_2^\pi = e_2^\pi \in b^\pi$.

We are now in a position to show that $(l(\pi), \mathfrak{L})$ is a labeling (see Def. 6). From (4) and the nesting of boxes in π, we deduce that $\sqsubseteq_{l(\pi)}$ is a tree-like order with $\mathfrak{L}(\bot_{l(\pi)})$ as a minimal element (Cond. (i)); from (4) and the condition on the conclusions of the contents of a box in π, we deduce Cond. (ii) and (iii).

It remains to check that $(\alpha, \mathfrak{L}) \equiv \pi$. Let $b^\pi \in \mathrm{box}\,\pi$, and b be the !-cell of $l(\pi)$ associated with b^π: by (4) we have $c \in l(\rho)$ iff $c \in \mathrm{cont}(l(\pi), \mathfrak{L}, b)$. ☐

B Reduction

We give a sequence of simple facts over the labelling, that will be useful for proving soundness and completeness.

It is immediate that $(\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{mrg}} (\alpha', \mathfrak{L}', \mathcal{A}')$ entails that $(\alpha, \mathfrak{L}, \mathcal{A} \cup \mathcal{B}) \xrightarrow{\mathtt{mrg}} (\alpha', \mathfrak{L}', \mathcal{A}' \cup \mathcal{B})$. The converse does not hold in general, however we have:

Fact 1. *If $(\alpha, \mathfrak{L}, \mathcal{A} \cup \mathcal{B}) \xrightarrow{\mathtt{mrg}} (\alpha', \mathfrak{L}', \mathcal{A}' \cup \mathcal{B})$ and $\mathfrak{L}(\alpha) \subseteq \mathcal{A}$, $\mathfrak{L}'(\alpha') \subseteq \mathcal{A}'$ then $(\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{mrg}} (\alpha', \mathfrak{L}', \mathcal{A}')$.*

Fact 2. *For every reduction $R : (\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{mrg}*} (\alpha', \mathfrak{L}', \mathcal{A}')$, the alphabet $\mathcal{A}'$ consists of $\mathcal{A}$ and the addresses introduced by the ers of R.*

Fact 3 (Postponement). *Let $(\alpha_1, \mathfrak{L}_1, \mathcal{A}_1) \xrightarrow{\mathtt{x}} (\alpha_2, \mathfrak{L}_2, \mathcal{A}_2) \xrightarrow{\mathtt{y}} (\alpha_3, \mathfrak{L}_3, \mathcal{A}_3)$ for two ers $\mathtt{x}, \mathtt{y}$ of $\mathtt{mrg}$. If the counter in the redex of $\mathtt{y}$ is not in the contractum of $\mathtt{x}$ and $\mathtt{y}$ uses an address in $\mathcal{A}_1$ or introduces a new address, then there is $(\alpha_3, \mathfrak{L}_3, \mathcal{A}_3)$ such that the following diagram* *commutes:*

$$\begin{array}{ccc}(\alpha_1,\mathfrak{L}_1,\mathcal{A}_1) & \xrightarrow{\mathtt{x}} & (\alpha_2,\mathfrak{L}_2,\mathcal{A}_2)\\ {\scriptstyle \mathtt{y}}\,\vdots & & \downarrow{\scriptstyle \mathtt{y}}\\ (\alpha_4,\mathfrak{L}_4,\mathcal{A}_4) & \overset{\mathtt{x}}{\dashrightarrow} & (\alpha_3,\mathfrak{L}_3,\mathcal{A}_3)\end{array}$$

Finally, the next fact states that linear reduction transports initial configurations.

Fact 4. *If* $(\mathrm{Init}_{\mathtt{A}_w}(\alpha_i,\mathtt{l}_i),\mathfrak{L}_{\mathtt{A}},\{\mathtt{A}\}) \xrightarrow{\mathrm{lnr}*} (\alpha',\mathfrak{L}',\mathcal{A}')$*, then* $\underline{\uparrow}\alpha'$ *is an initial configuration that is there are* $(\alpha'_i)_{i\leq n}$ *such that* $\underline{\uparrow}\alpha' = \mathrm{Init}_{\mathtt{A}_w}(\alpha'_i,\mathtt{l}_i)_{i\leq n}$*, and* $\mathfrak{L}' = \mathfrak{L}_{\mathtt{A}}$*,* $\mathcal{A}' = \{\mathtt{A}\}$*.*

Proposition 8. *If* $(\alpha,\mathfrak{L},\mathcal{A})$ *is a merging triple and* $(\alpha,\mathfrak{L},\mathcal{A}) \xrightarrow{\mathtt{mrg}} (\alpha',\mathfrak{L}',\mathcal{A}')$ *then* $(\alpha',\mathfrak{L}',\mathcal{A}')$ *is a merging triple.*

Proof. The proof splits in several cases, depending on the type of the ers performed in $(\alpha,\mathfrak{L},\mathcal{A}) \xrightarrow{\mathtt{mrg}} (\alpha',\mathfrak{L}',\mathcal{A}')$ (see Figures 9). In all cases we will deduce that $(\alpha',\mathfrak{L}',\mathcal{A}')$ meets the conditions (i)-(iii) of the definition of merging triple (Def. 13), assuming that these conditions hold in $(\alpha,\mathfrak{L},\mathcal{A})$. In the sequel, (i)-(iii) refer to the properties (i)-(iii) of Definition 6).

Case i ($\mathtt{ax}$)**.** Assume $(\alpha,\mathfrak{L},\mathcal{A}) \xrightarrow{\mathtt{ax}} (\alpha',\mathfrak{L}',\mathcal{A}')$ and let t_1, t_2 be the two counters of α erased by the $\mathtt{ax}$-ers and w be the axiom created in α'.

By definition, $\mathfrak{L}(t_1) = \mathfrak{L}(t_2)$, $\mathcal{A} = \mathcal{A}'$ and $\mathfrak{L}$, $\mathfrak{L}'$ take the same values on the same cells. Since $(\alpha',\mathfrak{L}',\mathcal{A}')$ has no new counter, it clearly meets Cond. (i). Besides, the only difference between $\downarrow\alpha$ and $\downarrow\alpha'$ is in the axiom w created in $\downarrow\alpha'$: this means that $\sqsubseteq_{\downharpoonleft\alpha,\mathfrak{L}}$ and $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'}$ are the same order on $\mathcal{A}$, and so $(\alpha',\mathfrak{L}')$ satisfies Cond. (ii) and (i), (ii). Proving (iii) is subtler: we have to check that it holds for the new axiom w. Suppose that one port of w in $\downarrow\alpha'$ is an auxiliary port of a cell c such that $\mathfrak{L}'(c) \neq \mathfrak{L}'(\bot_\alpha)$, then in $\overline{\downarrow}\alpha$ the principal port of one the two counters t_1, t_2, say w.l.o.g. of t_1, is wired to the auxiliary port of c. By Cond. (i) we have $\mathfrak{L}(c) = \mathfrak{L}(t_1)$. This means that $\mathfrak{L}(t_2) = \mathfrak{L}(t_1) \neq \mathfrak{L}(\bot_\alpha)$, and so, again by Cond. (i), we deduce that the principal port of t_2 is wired in $\downarrow\alpha$ to a cell c' and $\mathfrak{L}(c') = \mathfrak{L}(t_2) = \mathfrak{L}(c)$. We conclude by remarking that c and c' are the two cells wired by w in $\downarrow\alpha'$ and $\mathfrak{L}'(c) = \mathfrak{L}'(c')$.

Case ii ($\mathtt{zero}, \mathtt{one}, \mathtt{two}, \mathtt{contr}$)**.** Assume $(\alpha,\mathfrak{L},\mathcal{A}) \xrightarrow{\mathtt{one}} (\alpha',\mathfrak{L}',\mathcal{A}')$, let t be the counter involved in the $\mathtt{one}$-ers, where we adopt the convention of denoting with the same letter t both the counter in the redex and its residue in the contractum, and let c' be the $?d$-cell created in α'.

By definition, $\mathfrak{L}(t) = \mathfrak{L}'(c') = \mathfrak{L}'(t)$, $\mathcal{A} = \mathcal{A}'$ and $\mathfrak{L}$, $\mathfrak{L}'$ take the same values on the same cells but t, c'. Cond. (i) is immediate, since for the counters different from t nothing changes, while the principal port of t is wired to c' in $\overline{\downarrow}\alpha'$ and $\mathfrak{L}'(t) = \mathfrak{L}(c')$.

Now we prove that $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'}$ is equal to $\trianglelefteq$ on $\mathcal{A}$ (Cond. (ii)), supposing that the latter is equal to $\sqsubseteq_{\downharpoonleft\alpha,\mathfrak{L}}$. Since $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'}$ is the transitive closure of $\sqsubseteq^\circ_{\downharpoonleft\alpha',\mathfrak{L}'}$ (Def. 6) and since we are supposing that $\trianglelefteq$ is equal to the transitive closure of $\sqsubseteq^\circ_{\downharpoonleft\alpha,\mathfrak{L}}$, it suffices to prove $\sqsubseteq^\circ_{\downharpoonleft\alpha,\mathfrak{L}} \subseteq \sqsubseteq^\circ_{\downharpoonleft\alpha',\mathfrak{L}'} \subseteq \trianglelefteq$ on $\mathcal{A}$. The first inequality is immediate. Let us show $\sqsubseteq^\circ_{\downharpoonleft\alpha',\mathfrak{L}'} \subseteq \trianglelefteq$. The only pairs in $\sqsubseteq^\circ_{\downharpoonleft\alpha',\mathfrak{L}'}$ which might not be in $\sqsubseteq^\circ_{\downharpoonleft\alpha,\mathfrak{L}}$ are of the form $\mathfrak{L}'(d) \sqsubseteq^\circ_{\downharpoonleft\alpha',\mathfrak{L}'} \mathfrak{L}'(c')$ for a cell $d \leq_{\downharpoonleft\alpha'} c'$. By Cond. (i) there is a cell c in $\downarrow\alpha$ wired to the principal port of t and such that $\mathfrak{L}(c) \trianglelefteq \mathfrak{L}(t)$. In $\downarrow\alpha'$ the cell c is wired to the principal port of c', so $d \leq_{\downharpoonleft\alpha'} c'$ means $d = c'$ or $d \leq_{\downharpoonleft\alpha} c$. In the first case obviously $\mathfrak{L}'(d) = \mathfrak{L}'(c')$, otherwise we have $\mathfrak{L}(d) \trianglelefteq \mathfrak{L}(c)$ (by the supposed equivalence $\trianglelefteq = \sqsubseteq_{\downharpoonleft\alpha,\mathfrak{L}}$ on $\mathcal{A}$), and so $\mathfrak{L}'(d) \trianglelefteq \mathfrak{L}'(c')$ by $\mathfrak{L}'(d) = \mathfrak{L}(d)$ and $\mathfrak{L}(c) \trianglelefteq \mathfrak{L}(t) = \mathfrak{L}'(c')$ (recall that $\trianglelefteq$ is transitive). We conclude that $\sqsubseteq^\circ_{\downharpoonleft\alpha',\mathfrak{L}'} \subseteq \trianglelefteq$ and so $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'} = \trianglelefteq$.

Cond.(iii) follows immediately from $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'} = \trianglelefteq = \sqsubseteq_{\downharpoonleft\alpha,\mathfrak{L}}$ on $\mathcal{A}$ and from the fact that the only cell created in $\downarrow\alpha'$ is an exponential cell (hence Lab.(ii) holds).

The cases $(\alpha,\mathfrak{L},\mathcal{A}) \xrightarrow{\mathtt{x}} (\alpha',\mathfrak{L}',\mathcal{A}')$, for $\mathtt{x}$ among $\mathtt{zero}$, $\mathtt{two}$, $\mathtt{contr}$, are easy variants. The only notable difference is in proving Lab. (ii) in the $\mathtt{two}$-ers case: in that case the cell c' created by the ers is not exponential as it is in the other cases. However we remark that whenever $\mathfrak{L}(c') \neq \mathfrak{L}(\bot_{\alpha'})$, we have $\mathfrak{L}(t) \neq \mathfrak{L}(\bot_\alpha)$ and thus by Cond. (i) on $(\alpha,\mathfrak{L},\mathcal{A})$, the principal port of t is wired to a cell $c \in \downarrow\alpha$ such that $\mathfrak{L}(c) = \mathfrak{L}(t)$. This means that the principal port of c' is wired to c in $\downarrow\alpha'$, which give also (ii).

Case iii ($!\mathtt{p}$)**.** Assume $(\alpha,\mathfrak{L},\mathcal{A}) \xrightarrow{!\mathtt{p}} (\alpha',\mathfrak{L}',\mathcal{A}')$, let t be the counter involved in the $!\mathtt{p}$-ers, where, as in the former case, t denotes the counter in both the redex and the contractum, and let c' be the $!d$-cell created in α'.

On every cell different from t and c' the two labelings $\mathfrak{L}$ and $\mathfrak{L}'$ coincide, while we have $\mathfrak{L}(t) \triangleleft \mathfrak{L}'(t) = \mathfrak{L}'(c')$. Similarly to the previous case, one can prove Cond. (i) and that $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'}$ is equal to $\trianglelefteq$ on $\mathcal{A}'$ (Cond. (ii)). In particular notice that from the hypothesis $\mathfrak{L}(\downarrow\alpha) = \mathcal{A}$, one deduces $\mathfrak{L}'(\downarrow\alpha') = \mathcal{A} \cup \{\mathfrak{L}'(c')\}$.

As for Cond. (iii), Lab. (ii)-(iii) are straightforward, the cell c' being exponential and the net $\downarrow\alpha'$ having no new axioms with respect to $\downarrow\alpha$. However Lab. (i) is subtle since from $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'} = \trianglelefteq$ on $\mathcal{A}'$ we cannot deduce that $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'}$ is a tree-order: from the hypothesis $\sqsubseteq_{\downharpoonleft\alpha,\mathfrak{L}} = \trianglelefteq$ on $\mathcal{A}$ we do not know anything on the relation $\trianglelefteq$ w.r.t. the address $\mathfrak{L}'(c')$. However since the definition of the $!\mathtt{p}$-ers requires that $\mathfrak{L}'(c')$ is obtained from $\mathfrak{L}(t)$ by elements of X fresh in $\mathcal{A}$, we can deduce for every address $\mathtt{A} \in \mathcal{A}$ that $\mathfrak{L}'(c') \not\trianglelefteq \mathtt{A}$, and $\mathtt{A} \trianglelefteq \mathfrak{L}'(c')$ iff $\mathtt{A} \trianglelefteq \mathfrak{L}(t)$. We conclude that $\trianglelefteq$, hence $\sqsubseteq_{\downharpoonleft\alpha',\mathfrak{L}'}$, stays antisymmetric and tree-like on $\mathcal{A}'$, which proves (i).

Case iv ($?\mathtt{d}$)**.** Assume $(\alpha,\mathfrak{L},\mathcal{A}) \xrightarrow{!\mathtt{p}} (\alpha',\mathfrak{L}',\mathcal{A}')$, and let t

be the counter (both in α and α') involved in the $?\mathtt{d}$-ers. By definition, $\mathcal{A}' = \mathcal{A}$ and $\mathfrak{L}$, $\mathfrak{L}'$ take the same values on the same cells but on the counter t, where we have $\mathfrak{L}(t) \triangleleft \mathfrak{L}'(t)$ and $\mathfrak{L}'(t) \in \mathcal{A}$. Cond. (i) still holds, since the auxiliary ports of t are wired to ?-cells also in α'. Cond. (ii) and (iii) are immediate, since $(\downarrow\alpha, \mathfrak{L})$ and $(\downarrow\alpha', \mathfrak{L}')$ denote the same labeling.

□

Proposition 2. *The number of ers of any* $\xrightarrow{\mathtt{mrg}}$*-reduction starting from a merging triple* $(\alpha, \mathfrak{L}, \mathcal{A})$ *is polynomially bounded by the number of ports in* α.

Proof. Given a counter c, we set ($\#$ is the set cardinality):

$$\mathsf{width}(c) := 1 + (\#\{p \text{ port}\ ;\ p \in \alpha \setminus c \text{ and } p >_\alpha c\})^2$$
$$\mathsf{depth}(c) := \#\{b \text{ !-cell}\ ;\ b \in\downarrow\alpha \text{ and } \mathfrak{L}(c) \not\subseteq \bigcup \mathcal{A}\},$$

then set $|c| := \mathsf{width}(c) + \mathsf{depth}(c)$, and $|(\alpha, \mathfrak{L}, \mathcal{A})|$ as the sum of the $|c|$ for every counter $c \in \alpha$. One can check that $|(\alpha, \mathfrak{L}, \mathcal{A})|$ shrinks under any ers (the square in $\mathsf{width}(c)$ is needed for $\xrightarrow{\mathtt{contr}}$, and depth is needed for $\xrightarrow{?\mathtt{p}}$). □

C Completeness and soundness

C.1 The linear case

We start by studying the properties of the linear reduction.

Lemma 3 (Linear reduction)**.** *Let π be an ll-net, let $b_1, \ldots, b_s$ be the $s \geq 0$ boxes at depth 0 in π, and let ρ_r be the content of b_r, for each $r \leq s$. Let $(\alpha_i)_{i \leq n}$ be a family of simple nets with the same interface as π, let $(\mathtt{l}_i)_{i\leq n} = \mathtt{A}$ be a family of pairwise incomparable tokens, and let for each $r \leq s$, $(\gamma_{r,i})_{i \leq n}$ be a family of simple nets with the same interface of b_r.*

The following two conditions are equivalent:

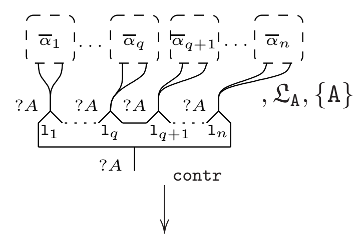
1. *for each $i \leq n$, $\alpha_i = \pi[\gamma_{r,i}/b_r]_{r\leq s}$,*
2. *there is a reduction:*

$$(\mathrm{Init}\,(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{lnr}*} (\pi[\mathrm{Init}\,(\gamma_{r,i}, \mathtt{l}_i)_{i\leq n}/b_r]_{r\leq s}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}).$$

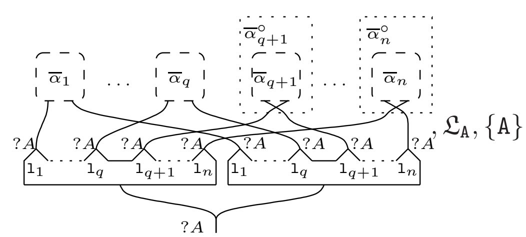
Proof. The proof is by induction on the construction cases of an ll-net π, as given in Definition 1. Contraction is the only delicate case, the other being straightforward and left to the reader. Let π be the result of contracting two free ports p', p'' of an ll-net π, i.e. π (resp. π') has interface $(p : ?A, \Gamma)$ (resp. $(p' : ?A, p'' : ?A, \Gamma)$), where Γ denotes the remaining free ports of the interface. Indeed π can also be the result of contracting p' and p'' in the ll-net π'' swapping the ports p', p'' of π', i.e. with interface $(p'' : ?A, p' : ?A, \Gamma)$. This is the delicate point: π' and π'' are different ll-nets, but contracting the free ports p', p'' yields in both cases the same ll-net π, since contraction is commutative. Now, let us prove the equivalence between the conditions 1 and 2 of the Lemma.

Assume 1 and let us deduce 2. By definition of substitution each simple net α_i has interface $(p : ?A, \Gamma)$, and it can be obtained by contracting two free ports $p' : ?A$ and $p'' : ?A$ of a simple net $\overline{\alpha}_i$ such that $\overline{\alpha}_i = \pi'[\gamma_{r,i}/b_r]_{r\leq s}$ or $\overline{\alpha}_i = \pi''[\gamma_{r,i}/b_r]_{r\leq s}$. The point is that we cannot choose which one between π', π'' yields $\overline{\alpha}_i$. Let us enumerate the family $\{\alpha_i\}_{i\leq n}$ so that there is a $q \leq n$ s.t. for every $i \leq q$, $\overline{\alpha}_i = \pi'[\gamma_{r,i}/b_r]_{r\leq s}$, and for every i, $q < i \leq n$, $\overline{\alpha}_i = \pi''[\gamma_{r,i}/b_r]_{r\leq s}$. For every i, $q < i \leq n$, let $\overline{\alpha}_i^\circ$ be the simple net obtained from $\overline{\alpha}_i$ by swapping p', p'', so that $\overline{\alpha}_i^\circ$ has interface $(p' : ?A, p'' : ?A, \Gamma)$ and $\overline{\alpha}_i^\circ = \pi'[\gamma_{r,i}/b_r]_{r\leq s}$. By inductive hypothesis we know that: $(\mathrm{Init}\,(\overline{\alpha}_1, \mathtt{l}_1, \ldots, \overline{\alpha}_q, \mathtt{l}_q, \overline{\alpha}^\circ_{q+1}, \mathtt{l}_{q+1}, \ldots, \overline{\alpha}^\circ_n, \mathtt{l}_n), \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{lnr}*} (\pi'[\mathrm{Init}\,(\gamma_{r,i}, \mathtt{l}_i)_{i\leq q}/b_r]_{r\leq s}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\})$ Besides we have (by omitting the counter on Γ for clarity):

$$\mathrm{Init}\,(\alpha_i, \mathtt{l}_i)_{i\leq n} =$$



$$\mathrm{Init}\,(\overline{\alpha}_1, \mathtt{l}_1, \ldots, \overline{\alpha}_q, \mathtt{l}_q, \overline{\alpha}^\circ_{q+1}, \mathtt{l}_{q+1}, \ldots, \overline{\alpha}^\circ_n, \mathtt{l}_n)_{i\leq n} =$$



from which we conclude. Notice that in the above step it is crucial to be able to swap the free ports of $\overline{\alpha_{q+1}}, \ldots, \overline{\alpha_n}$ so to apply the inductive hypothesis. This justifies our definition of the `contr` step.

The proof that 2 implies 1 is symmetric and left to the reader. □

The following is straightforward from Lemma 3 (with $s = 0$):

Proposition 4 (linear reduction)**.** *Let $(\alpha_i)_{i\leq n}$ be a family of simple nets with the same interface and without !-cells,* *let* $\mathtt{A} = (\mathtt{l}_i)_{i\leq n}$ *be a family of pairwise incomparable tokens. There is a labeling* $(\alpha, \mathfrak{L})$ *and a successful reduction sequence*

$$(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\alpha, \mathfrak{L}, \mathcal{A}),$$

iff for every $i \leq n$, $\alpha_i = \alpha$, $\mathfrak{L} = \mathfrak{L}_{\mathtt{A}}$ *and* $\mathcal{A} = \{\mathtt{A}\}$.

C.2 Completeness

Theorem 1 (Completeness)**.** *Let* π *be a ll-net, and let* $\alpha_1, \dots, \alpha_n$ *be simple nets in* $\mathcal{T}(\pi)$*. For any family* $\{\mathtt{l}_i\}_{i\leq n} = \mathtt{A}$ *of pairwise distinct tokens, there exist a labeling* $(\alpha, \mathfrak{L})$ *and a merging reduction sequence*

$$(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\alpha, \mathfrak{L}, \mathcal{A}),$$

where $\mathfrak{L}_{\mathtt{A}}$ *is, as usual, the constant function taking value* $\mathtt{A}$*. Moreover, there are boxes* $b_o \in \mathrm{box}(\pi)$ *for* $0 \leq r \leq s$ *such that*

$$(\alpha, \mathfrak{L}) \equiv \pi[!0/b_r]_{r\leq s}.$$

Proof. The proof is by induction on the exponential depth of π; we split the induction step in two cases, the one where π is equal to a box $!\rho$ and the general case. This splitting recalls the cases of Taylor expansion (Definition 5).

Case i (no box)**.** If π has no boxes, then $\mathcal{T}(\pi) = \{\pi\}$ (see Def. 5) and by Lemma 3 (case $s = 0$) we have $(\mathrm{Init}(\pi, \dots, \pi), \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\pi, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\})$, where $\mathfrak{L}_{\mathtt{A}}$ is the constant function taking value $\mathtt{A}$. Clearly $(\pi, \mathfrak{L}_{\mathtt{A}}) \equiv \pi$.

Case ii (one box)**.** If π is a box of contents ρ, then each $\alpha_i = \prod_{j=1}^{k_i} \mathrm{cod}(A, \gamma_{i,j})$, where A is the label of the principal free port of ρ, k_i is an integer and $\forall j \leq k_i, \gamma_{i,j} \in \mathcal{T}(\rho)$. Notice that it might be $k_i = 0$, i.e. $\alpha_i = l(!0)$ is made of a coweakening and some weakenings. Without loss of generality we can enumerate the α_i's so that every $i \leq q$ has $k_i > 0$ and every i included between $q+1$ and n has $k_i = 0$. We suppose moreover $q > 0$, i.e. at least $k_1 > 0$: the case $q = 0$ is straightforward.

We consider an element $y \in X$ fresh in $\mathtt{A}$, and a family $\{x_i^j\}_{i\leq q, j\leq k_i}$ of pairwise distinct elements of X which are fresh in $\mathtt{A}, y$. By means of them we define $\mathtt{m}_{i,j} = \mathtt{l}_i \cup \{x_i^j\}$ for every $i \leq q$ and $j \leq k_i$, and $\mathtt{m}_i = \mathtt{l}_i \cup \{y\}$ for every i, $q < i \leq n$. Finally let $\mathtt{B}_w = \{\mathtt{m}_i \ ; \text{ for } q < i \leq n\}$ and $\mathtt{B} = \mathtt{B}_w \cup \{\mathtt{m}_{i,j} \ ; \text{ for } i \leq q, j \leq k_i\}$, and notice that $\mathtt{A} \lhd \mathtt{B}$. By induction hypothesis we have:

$$(\mathrm{Init}(\gamma_{i,j}, \mathtt{m}_{i,j})_{\substack{i\leq q\\ j\leq k_i}}, \mathfrak{L}_{\mathtt{B}}, \{\mathtt{B}\}) \xrightarrow{\mathtt{mrg}*} (\beta, \mathfrak{L}', \mathcal{B}), \quad (5)$$

where $(\beta, \mathfrak{L}') \equiv \rho[!0/b_r]_{r\leq s}$, for $s \geq 0$ boxes in $\mathrm{box}(\rho)$. Moreover, by simple computations we have

$$(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{!p} \xrightarrow{?p*} (\mathrm{cod}(p, \mathrm{Init}_{\mathtt{B}_w}(\gamma_{i,j}, \mathtt{m}_{i,j})_{\substack{i\leq q\\ j\leq k_i}}), \mathfrak{L}'', \{\mathtt{A}, \mathtt{B}\}),$$

where $\mathfrak{L}''$ labels only two cells, the conclusion cell, with $\mathtt{A}$, and the $!d$-cell of principal port p, with $\mathtt{B}$.

As we have remarked after the Definition 12 of merging reduction, one can always extend the set of addresses and keep $\mathtt{mrg}$-ers: this means in particular that (5) still holds when replacing $\{\mathtt{B}\}$ and $\mathcal{B}$ with resp. $\{\mathtt{A}, \mathtt{B}\}$ and $\{\mathtt{A}\} \cup \mathcal{B}$. We conclude by context closure:

$$(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\mathrm{cod}(p, \beta), \mathfrak{L}''', \mathcal{B}),$$

where $\mathfrak{L}'''_{|\beta} = \mathfrak{L}''$, so $(\mathrm{cod}(p, \beta), \mathfrak{L}''') \equiv \pi[!0/b_r]_{r\leq s}$.

Case iii (otherwise)**.** Assume π has $s' \geq 1$ boxes at exponential depth 0, i.e. $\mathrm{box}(\pi) = \{b'_r, r \leq s'\}$, where we use the ' to distinguish these boxes from the ones mentioned in the statement of the theorem. For each $r \leq s'$ let ρ_r be the contents of b'_r. By definition of Taylor expansion (Def. 5), each α_i is equal to $\pi[\gamma_{r,i}/b'_r]_{r\leq s'}$, with $\gamma_{r,i} \in \mathcal{T}(b_r)$ for every $r \leq s'$. Fix now the index $r \leq s'$ and let vary $i \leq n$: by the previous case we have

$$(\mathrm{Init}(\gamma_{r,i}, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\beta_r, \mathfrak{L}_r, \mathcal{A}_r), \quad (6)$$

where $\mathfrak{L}_{\mathtt{A}}$ is, as usual, the constant function taking value $\mathtt{A}$ and we have $(\beta_r, \mathfrak{L}_r) \equiv b_r[!0/b_{r,i}]_{i\leq k_r}$, for some $k_r \geq 0$ boxes in $\mathrm{box}(b_r)$. By the way, remark that we might have $k_r = 1$ and $b_{r,1} = b_r$, so $(\beta_r, \mathfrak{L}_r) \equiv !0$. Notice also that we can suppose that for different $r \leq s'$ the sets of addresses $\mathcal{A}_r$ share only $\mathtt{A}$. By Lemma 3 we have:

$$(\mathrm{Init}(\pi[\gamma_{r,i}/b'_r]_{r\leq s'}, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\pi[\mathrm{Init}(\gamma_{r,i}, \mathtt{l}_i)_{i\leq n}/b'_r]_{r\leq s'}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}).$$

Finally, by context closure and (6), we conclude:

$$(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\pi[\beta_r/b'_r]_{r\leq s'}, \cup_{r\leq s'} \mathfrak{L}_r, \cup_{r\leq s'} \mathcal{A}_r),$$

with $(\pi[\beta_r/b'_r]_{r\leq s'}, \cup_{r\leq s'} \mathfrak{L}_r) \equiv \pi[!0/b_{r,i}]_{\substack{r\leq s'\\ i\leq k_r}}$, since for every $r \leq s'$ we have $(\beta_r, \mathfrak{L}_r) \equiv !\rho_r[!0/b_{r,i}]_{i\leq k_r}$.

□

C.3 Soundness

We need some other notations: The ers $!\mathtt{p}$ and $?\mathtt{p}$ are called **boxing** ers. They are the only ones changing an addresses $\mathtt{A}$ associated with a counter into $\mathtt{A}'$; we say that the ers **enters** into the new address $\mathtt{A}'$. Besides, these ers require that the family $\{y\} \cup \{x_i^j\}_{i\leq q, j\leq n_i}$ of elements of X added to the new addresses are pairwise distinct. Thus, the new labeling meets the conditions of the definition of counter (see Def. 13).

We first prove the splitting Lemma 4. For this, we need two other lemmas.

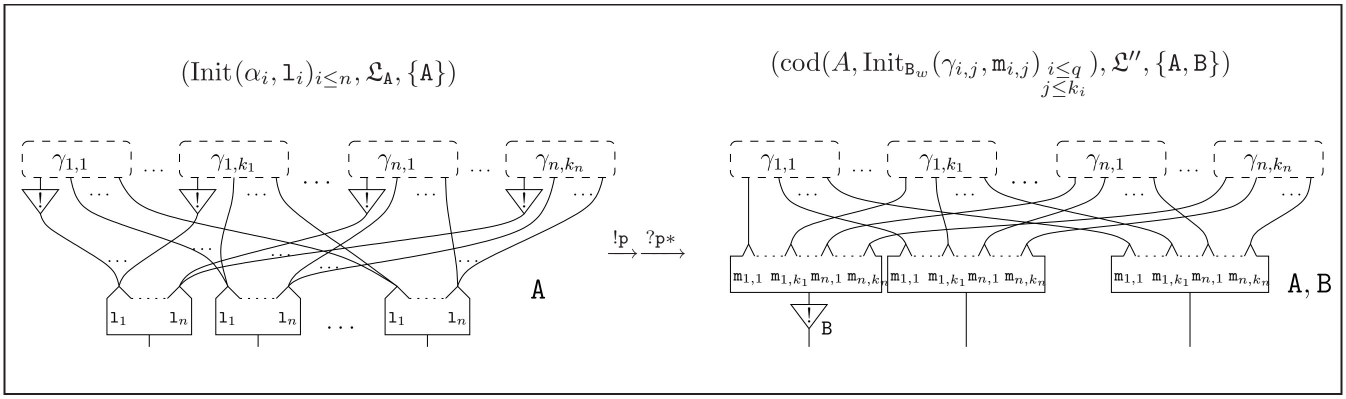


Figure 10: case one box

Lemma 5. *Let R be a successful reduction sequence from a merging triple $(\alpha, \mathfrak{L}, \mathcal{A})$; let $\beta_1, \ldots, \beta_n$ be the trees of the canonical decomposition of the simple net $\uparrow\alpha$, as in Figure 7. Let p, q be two distinct free ports of $\uparrow\alpha$, let β_p (resp. β_q) be the β_i tree with root p (resp. q) and let t (resp. u) be the counter having p (resp. q) as auxiliary port. If β_p and β_q share an axiom then:*

(i) t and u are different counters;

(ii) both pairs $\mathfrak{L}(t)$, $\mathfrak{L}(u)$ and $\lambda_t(p)$, $\lambda_u(q)$ are in the same order with respect to $\trianglelefteq$ for the former and $\subseteq$ for the latter;

(iii) if $\mathfrak{L}(t) < \mathfrak{L}(u)$, then R enters the $\mathfrak{L}(u)$, if $\mathfrak{L}(t) < \mathfrak{L}(u)$ then R enters $\mathfrak{L}(u)$.

Proof. We proceed by induction on the length of R. Assume that R starts with $(\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{mrg}} (\alpha', \mathfrak{L}', \mathcal{A}')$, then the suffix R' of R defines a successful reduction sequence starting from $(\alpha', \mathfrak{L}', \mathcal{A}')$. The lemma does not follow immediately from the induction hypothesis applied to R' only in case one of the two counters t, u is involved in the ers $(\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{mrg}} (\alpha', \mathfrak{L}', \mathcal{A}')$. Under this hypothesis the proof splits in several cases, depending on the type of the ers.

If $(\alpha, \mathfrak{L}, \mathcal{A}) \xrightarrow{\mathtt{ax}} (\alpha', \mathfrak{L}', \mathcal{A}')$, then the only wires which were not already in $\uparrow\alpha'$, are axioms wiring $t \neq u$. Notice that the rule $\xrightarrow{\mathtt{ax}}$ forces $\mathfrak{L}(t) = \mathfrak{L}(u)$ and the labels of any two wired auxiliary ports to be equal.

If the first reduction step of R is $\xrightarrow{\mathtt{!p}}$ or $\xrightarrow{\mathtt{?p}}$ (see Def. 12), let t be the counter in α involved in the reduction, and t' be the counter in the result α' of the reduction (see Figure 9): t' has one port p' which is a residue of p and such that its $\beta'_{p'}$ tree shares an axiom with β'_q. We have $\mathfrak{L}(t) \triangleleft \mathfrak{L}'(t')$ and $\lambda_t(p) \subset \lambda_{t'}(p')$. By induction hypothesis, $t' \neq u$ in α', hence $t \neq u$ in α. Besides, either $\mathfrak{L}'(t') \trianglelefteq \mathfrak{L}'(u) = \mathfrak{L}(u)$ and $\lambda_{t'}(p') \subseteq \lambda_u(q)$ (the labeling $\mathfrak{L}_u$ of u is the same in α and α'), then $\mathfrak{L}(t) \triangleleft \mathfrak{L}(u)$ and $\lambda_t(p) \subset \lambda_u(q)$; or $\mathfrak{L}(u) \trianglelefteq \mathfrak{L}'(t)$. In the latter case, we have $\lambda_u(q) \subseteq \lambda_{t'}(p')$ by induction hypothesis; moreover, by the definition of the ers $\lambda_t(p) \subset \lambda_{t'}(p')$ and the elements in $\lambda_{t'}(p') \setminus \lambda_t(p)$ are fresh in $\mathcal{A} \ni \mathfrak{L}(u), \mathfrak{L}(t)$. We conclude $\lambda_u(q) \subseteq \lambda_t(p)$.

The other cases are easier and left to the reader. □

Lemma 6. *Let R be a successful reduction sequence from a merging triple $(\alpha, \mathfrak{L}, \mathcal{A})$, and such that no ers of R enters an address labeling a counter of α; if p, q are free ports of the same connected component of $\uparrow\alpha$, then:*

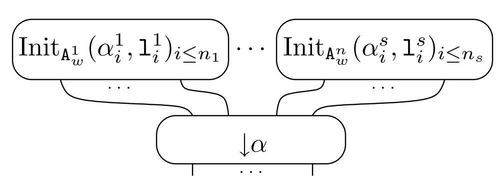
$$\mathfrak{L}(t) = \mathfrak{L}(u) \qquad \textit{and} \qquad \lambda_t(p) = \lambda_u(q). \tag{7}$$

Proof. Let $\beta_1, \ldots, \beta_n$ be the trees of the canonical decomposition of the simple net $\uparrow\alpha$, as in Figure 7. The two free ports p and q belong to a same connected component iff there are $\beta_{i_1}, \ldots, \beta_{i_k}$ trees such that p (resp. q) is the root of β_{i_1} (resp. β_{i_k}) and for every $j < k$, β_{i_j} and $\beta_{i_{j+1}}$ shares at least one axiom. We prove Equation (7) by induction on the number k.

If $k = 1$, then the two auxiliary port are equal, and the lemma is satisfied. Else, let us denote p' the root of β_{i_2}, and t' the counter with auxiliary port p'. Since β_{i_1} and β_{i_2} share an axiom, we apply Lemma 5 to p and p': since R does not enter the addresses $\mathfrak{L}(t)$ and $\mathfrak{L}(t')$, then $\mathfrak{L}(t) = \mathfrak{L}(t')$ and $\lambda_t(p) = \lambda_{t'}(p')$. Finally, we apply the induction hypothesis to p', q and get $\mathfrak{L}(t') = \mathfrak{L}(u)$ and $\lambda_{t'}(p') = \lambda_u(q)$. □

Lemma 4 (Splitting)**.** *Let R be a successful reduction sequence from a merging triple $(\alpha, \mathfrak{L}, \mathcal{A})$, and such that no ers of R enters an address labeling a counter of α; then α can be split: there are suitable sequences $(\alpha_i^r)_{r\leq s,\, i\leq n_r}$, of simple nets, $(\mathtt{l}_i^r)_{r\leq s,\, i\leq n_r}$ of lists, and $(\mathtt{A}_w^r)_{r\leq s}$ of addresses such that*

$\alpha =$



Proof. Since R is successful, two counters of the same connected component of $\uparrow\alpha$ have the same label (Lem. 5- 6). With each label $\mathtt{A}$, we associate the subnet β made of cells connected to a counter labelled by $\mathtt{A}$. We are left to decompose the subnet $\uparrow\beta$ which has only one type of counter. Again, thanks to Lem. 6, two auxiliary ports of counters which are connected have the same label. Hence, for every $\mathtt{l} \in \mathtt{A}$, for any counter of β and above each auxiliary port labelled by $\mathtt{l}$, there is a subnet of $\uparrow\beta$ whose conclusions are auxiliary ports labelled by $\mathtt{l}$. With $\mathtt{l}$, we associate the net made of the subnets above each auxiliary port labelled by $\mathtt{l}$. So we have split $\uparrow\beta$ in disconnected components and got an initial state whose label is $\mathtt{A}$. □

Theorem 2 (Soundness)**.** *Let π be a ll-net, let $(\alpha_i)_{i\leq n}$ be a family of simple nets with the same interface and let $(\mathtt{l}_i)_{i\leq n} = \mathtt{A}$ be a family of pairwise incomparable tokens. If there is a labeling $(\alpha, \mathfrak{L}) \ll \pi$, a set of addresses $\mathcal{A}$ and a merging reduction sequence*

$$R = (\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_\mathtt{A}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\alpha, \mathfrak{L}, \mathcal{A}), \qquad (8)$$

then for every $i \leq n$, $\alpha_i \in \mathcal{T}(\pi)$.

Proof. Since $\pi_0 \ll \pi$ implies that $\mathcal{T}(\pi) \subseteq \mathcal{T}(\pi)'$, we only need to prove the theorem when $(\alpha, \pi) \equiv \pi$.

The proof is by induction on the exponential depth of π; we split the induction step in two cases, the one where π is equal to one box and the general case. Notice this splitting recalls the cases of Taylor expansion (Def. 5) and the cases of the proof of Theorem 1. In general notice that $(\alpha, \mathfrak{L}) \equiv \pi$ entails $\alpha = l(\pi)$ (Def. 7).

Case i (no box)**.** If π has no boxes, then $l(\pi) = \pi$, hence $\alpha = \pi$, and $\mathcal{T}(\pi) = \{\alpha\}$. In particular α has no !-cell. Since $(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_\mathtt{A}, \{\mathtt{A}\}) \xrightarrow{\mathtt{mrg}*} (\alpha, \mathfrak{L}, \mathcal{A})$, we deduce that every α_i has no !-cell, and we conclude by Proposition 4.

Case ii (one box)**.** Assume that π is reduced to only one box b whose content is ρ. Let p be the principal port of ρ in $l(\pi) = \pi[\mathrm{cod}(p, \rho)/b]$. Since $\pi \equiv (\alpha, \mathfrak{L})$ is a labeling, $\alpha = \mathrm{cod}(p, l(\rho))$, that is $l(\rho) = \mathrm{cont}(\alpha, \mathfrak{L}, \mathfrak{L}(b)) = \{c \in \alpha \,|\, \mathfrak{L}(b) \sqsubseteq_{\alpha,\mathfrak{L}} \mathfrak{L}(c)\} \backslash \{b\}$ and $\rho \equiv (l(\rho), \mathfrak{L}_{|l(\rho)})$ where $\mathfrak{L}(b)$ is the labeling of the !-cell associated with b in α. In the sequel, we will denote $\mathtt{B} = \mathfrak{L}(b)$ and $\mathcal{B} = \mathfrak{L}(l(\rho)) \cup \{\mathtt{B}\}$.
Since the only cell of α labelled by $\mathtt{A}$ is $\bot$, R does not use it. Moreover, $\mathtt{B}$ is a lower bound of $\mathcal{B}$, hence we can apply Fact 3 and decompose R into two reductions: one, R_B, made only of ers producing $\mathtt{B}$ and the other R' which does not produce any $\mathtt{B}$

$$(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_\mathtt{A}, \{\mathtt{A}\}) \xrightarrow[R_\mathtt{B}]{!\mathtt{p}?\mathtt{p}*} (\alpha', \mathfrak{L}', \{\mathtt{A}, \mathtt{B}\}) \xrightarrow[R']{\mathtt{mrg}*} (\alpha, \mathfrak{L}, \mathcal{A}), \qquad (9)$$

Notice that R_B begins with $!\mathtt{p}$ (see Fig. 9) $\mathtt{B}$. This means that there are numbers such that there are $0 \leq q \leq n$ and $(k_i)_{i\leq q}$ such that $\mathtt{B} = \{\mathtt{m}_i^j = \mathtt{l}_i \cup \{x_i^j\} \;;\; i \leq q, j \leq k_i\} \cup \{\mathtt{m}_i = \mathtt{l}_i \cup \{y\} \;;\; q < i \leq n\}$, where y is an element of X fresh in $\mathtt{A}$, and $\{x_i^j\}_{i\leq q, j\leq k_i}$ is a family of pairwise distinct elements of X fresh in $\mathtt{A} \cup \{y\}$.
Because $\downarrow \mathrm{Init}(\alpha_i, l_i)_{i\leq n}$ has no cell (except $\bot$) and due to the shape of $R_\mathtt{B}$, the simple net $\downarrow \alpha'$ is made of only one $!d$-cell labelled by $\mathtt{B}$, and the counters of $\mathbin{\rotatebox[origin=c]{180}{$\perp$}}\alpha'$ are also labelled by $\mathtt{B}$. So we have $\alpha' = \mathrm{cod}(p, \mathbin{\rotatebox[origin=c]{180}{$\perp$}}\alpha')$ on the one side of R' and $\alpha = \mathrm{cod}(p, l(\rho))$ on the other side (where we abusively identify the auxiliary port of the redex and the reduct). Moreover $\mathcal{A} = \mathcal{B} \cup \{\mathtt{A}\}$, $\mathfrak{L}'(\alpha') \subseteq \{\mathtt{B}\}$ and $\mathfrak{L}(l(\rho)) \subseteq \mathcal{B}$, hence Fact 1 implies that $(\mathbin{\rotatebox[origin=c]{180}{$\perp$}}\alpha', \mathfrak{L}', \{\mathtt{B}\}) \xrightarrow[R'']{\mathtt{mrg}*} (l(\rho), \mathfrak{L}'_{|l(\rho)}, \mathcal{B})$. Notice that both sides are merging triples thanks to the restriction of the set of addresses to the codomain of the labeling.
Since R'' is successful and does not produce $\mathtt{B}$, we can apply the splitting lemma (Lemma 4) and decompose $\uparrow \alpha'$ into a family $(\gamma_{i,j})_{i\leq q,\, j\leq k_i}$ of simple nets such that $\mathbin{\rotatebox[origin=c]{180}{$\perp$}}\alpha' = \mathrm{Init}_{\mathtt{B}_w}(\gamma_{i,j}, \mathtt{m}_{i,j})_{i\leq q,\, j\leq k_i}$. We are now ready to apply the induction hypothesis to $(\mathrm{Init}_{\mathtt{B}_w}(\gamma_{i,j}, \mathtt{m}_{i,j})_{i\leq q,\, j\leq k_i}, \mathfrak{L}_\mathtt{B}, \{\mathtt{B}\}) \xrightarrow{\mathtt{mrg}*} (l(\rho), \mathfrak{L}'|_{l(\rho)}, \mathcal{B})$ and get $\forall i \leq n,\ j \leq k_i,\ \gamma_{i,j} \in \mathcal{T}(\rho)$.
Because $\alpha' = \mathrm{cod}(p, \mathrm{Init}_{\mathtt{B}_w}(\gamma_{i,j}, \mathtt{m}_{i,j})_{i\leq q,\, j\leq k_i})$ and due to the shape of R_B, we have $\forall i \leq n,\ \alpha_i = \prod_{j=1}^{k_i} \mathrm{cod}(p, \gamma_{i,j})$. Applying the definition of Taylor expansion, we can conclude $\alpha_i \in \mathcal{T}(\pi)$.

Case iii (Otherwise)**.** Assume that, at exponential depth 0, π has $s \geq 1$ boxes $(b_r)_{r\leq s}$ whose respective contents are $(\rho_r)_{r\leq s}$. For each $r \leq s$, let p_r be the principal port of ρ_r in $\alpha = l(\pi) = \pi[\mathrm{cod}(p_r, l(\rho_r))/b_r]_{r\leq s}$. Since $\pi \equiv (\alpha, \mathfrak{L})$ is a labeling, for every $r \leq s$, $l(\rho_r) = \mathrm{cont}(\alpha, \mathfrak{L}, \mathfrak{L}(b_r)) = \{c \in \alpha \,|\, \mathfrak{L}(b_r) \sqsubseteq_{\alpha,\mathfrak{L}} \mathfrak{L}(c)\} \backslash \{b_r\}$ and $\rho_r \equiv (l(\rho_r), \mathfrak{L}_{|l(\rho_r)})$ where $\mathfrak{L}(b_r)$ is the labeling of the !-cell associated with b_r in α. In the sequel, we will denote $\mathtt{B}_r = \mathfrak{L}(b_r)$ and $\mathcal{B}_r = \mathfrak{L}(l(\rho_r)) \cup \{\mathtt{B}_r\}$ for each $r \leq s$. Notice that $\mathcal{A} = \biguplus_{r\leq s} \mathcal{B}_r \cup \{A\}$. Moreover, the boxes are disjoint, hence $\mathcal{B}_r$ and $\mathcal{B}_{r'}$ are also disjoint.
Now, let us consider the reduction R defined by (8): remark that along the reduction, the set of addresses is increasing, hence $\mathtt{A}$ is in the set of addresses of every merging triple occurring in R. We can apply Fact 3 several times and anticipate the ers using $\mathtt{A}$ before all other ers, thus transforming R into

$$(\mathrm{Init}(\alpha_i, \mathtt{l}_i)_{i\leq n}, \mathfrak{L}_\mathtt{A}, \{\mathtt{A}\}) \xrightarrow[R']{\mathtt{lnr}*} (\alpha', \mathfrak{L}_\mathtt{A}, \{\mathtt{A}\}) \xrightarrow[R'']{\mathtt{mrg}*} (\alpha, \mathfrak{L}, \mathcal{A}).$$

where R' is made of the linear ers of R using the address $\mathtt{A}$, while R'' is made of the ers of R using or introducing some address in $\biguplus_{r=1}^{s}\mathcal{B}_r$. Since R'' has no ers using $\mathtt{A}$, then $\downarrow\alpha'$ is the subnet of α made of the cells of α with label $\mathtt{A}$, i.e. $\alpha' = \pi[\gamma_r/b_r]_{r\leq s}$, for some simple nets γ_r whose conclusions are auxiliary ports of the counters. Moreover R' is linear, thanks to Fact 4, each $\gamma_r = \mathrm{Init}(\gamma_{i,r}, \mathtt{1}_i)_{i\leq n}$ for some simple nets $\gamma_{i,r}$. Finally, by Lemma 3, $\forall i \leq n$, $\alpha_i = \pi[\gamma_{i,r}/b_r]_{r\leq s}$.

Let $r \leq s$. We focus on showing $\gamma_{i,r} \in \mathcal{T}(!\rho_r)$ for every $i \leq s$. Notice that every ers of R'' using an address $\mathtt{B}$ in $\mathcal{B}_r$ can be anticipated before any ers using or introducing an address $\mathtt{B}'$ in $\mathcal{B}_{r'}$, for $r' \neq r$, since neither the latter can introduce $\mathtt{B}$ (otherwise $\mathtt{B} = \mathtt{B}' \in \mathcal{B}_{r'}$), nor its contractum can have the redex of the former (otherwise $\mathtt{B}' \trianglelefteq \mathtt{B}$, which would imply $\mathtt{B} \in \mathcal{B}_{r'}$). So we can apply Fact 3 and transform R'' into R''_r which uses $\mathcal{B}_r$ and R''' does not use any address in $\mathcal{B}_r$

$$(\pi[\mathrm{Init}(\gamma_{i,r}, \mathtt{1}_i)_{i\leq n}/b_r]_{r\leq s}, \mathfrak{L}_{\mathtt{A}}, \{\mathtt{A}\}) \xrightarrow[R''_r]{\mathtt{mrg}*} (\alpha_r, \mathfrak{L}_r, \{\mathtt{A}\} \uplus \mathcal{B}_r) \xrightarrow[R''']{\mathtt{mrg}*} (\alpha, \mathfrak{L}, \mathcal{A})$$

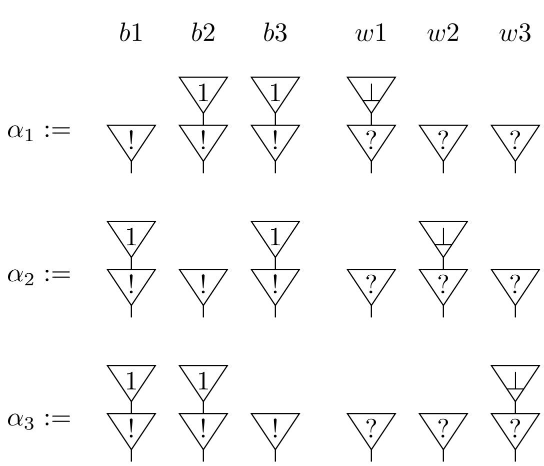
with $\alpha_r = \pi[l(!\rho_r)/b_r][\mathrm{Init}(\gamma_{i,r'}, \mathtt{1}_i)_{i\leq n}/b_{r'}]_{r'\leq s,\, r'\neq r}$. Due to Def. 12, R''_r is the reduction $(\mathrm{Init}(\gamma_{i,r}, \mathtt{1}_i)_{i\leq n}, \mathfrak{L}_{\mathtt{B}}, \{\mathtt{B}_r\}) \xrightarrow{\mathtt{mrg}*} (l(!\rho_r), \mathfrak{L}_r, \mathcal{B}_r)$ in a context. We are now ready to apply (case ii) to $\pi = !\rho_r$: $\forall i \leq n$, $\gamma_{i,r} \in \mathcal{T}(!\rho_r)$.

We have shown that for any $r \leq s$, for every $i \leq n$, $\gamma_{i,r} \in \mathcal{T}(!\rho)$. Recall that for every $i \leq n$, $\alpha_i = \pi[\gamma_{i,r}/b_r]_{r\leq s}$. Finally, by definition of Taylor expansion (see Def. 5), $\forall i \leq n$, $\alpha_i \in \mathcal{T}(\pi)$.

□

D Hypercoherence

Consider the following three simple nets, with sequent conclusion $!1, !1, !1, ?\bot, ?\bot, ?\bot$.



Each of them has three !-cells, named resp. $b1, b2, b3$, and three ?-cells, named resp. $w1, w2, w3$. We merge (co)weakening, (co)contraction and (co)dereliction in a unique cell ? (resp. !).

Now, every pair of simple nets between $\alpha_1, \alpha_2, \alpha_3$ is coherent, in the sense that there is a Taylor expansion of a ll-net which contains both elements of the pair:

α_1, α_2 is contained in the Taylor expansion of the ll-net:

$w1$	auxiliary port of	$b2$
$w2$	auxiliary port of	$b1$
$w3$	weakening	

α_2, α_3 is contained in the Taylor expansion of the ll-net:

$w1$	weakening	
$w2$	auxiliary port of	$b3$
$w3$	auxiliary port of	$b2$

α_1, α_3 is contained in the Taylor expansion of the ll-net:

$w1$	auxiliary port of	$b3$
$w2$	weakening	
$w3$	auxiliary port of	$b1$

However there is no ll-net whose Taylor expansion contains all three $\alpha_1, \alpha_2, \alpha_3$. Suppose such a ll-net π does exist. Then by $\alpha_1, \alpha_2 \in \mathcal{T}(\pi)$ we have that $b1$ has $w2$ as an auxiliary port; but then by $\alpha_3 \in \mathcal{T}(\pi)$ we should have $w2$ a dereliction in α_3 and not a weakening, as it is. We thus conclude such a ll-net π cannot exist.