Proof nets, Garbage, and Computations [1]

Stefano Guerrini [a] Simone Martini [c] Andrea Masini [b]

[a] *Institut de Mathématiques de Luminy - 163, Avenue de Luminy - Case 907 - 13288 Marseille Cedex 9, France - email:* `stefanog@iml.univ-mrs.fr`

[b] *Dipartimento di Informatica, Università di Pisa - Corso Italia, 40 - I-56100, Pisa, Italy - email:* `masini@di.unipi.it`

[c] *Dipartimento di Matematica e Informatica, Università di Udine - Via delle Scienze, 206 - I-33100, Udine, Italy - email:* `martini@dimi.uniud.it`

We study the problem of local and asynchronous computation in the context of multiplicative exponential linear logic (MELL) proof nets. The main novelty is in a complete set of rewriting rules for cut-elimination in presence of weakening (which requires garbage collection). The proposed reduction system is strongly normalizing and confluent. The proof techniques are new, all based on pure syntactical reasonings.

Key words: Linear logic; typed lambda-calculus; cut-elimination; sharing graphs; proof nets.

1 Introduction

Cut-elimination (or normalization) is the logical description of the computational process of (term) reduction, central to most of the literature on lambda-calculus and related functional languages. From the pioneering description of beta-reduction in terms of normalization of natural deduction proofs (dating back to the sixties, by Curry, Prawitz, and Howard), this logical interpretation has been extended to a variety of functional languages and formal logical systems.

The arrival on the scene of linear logic [Gir87] gave a good momentum to this research area, for the stress on resource conscious computations. Using the key idea that the type constructor for the function space may be understood as

[1] Extended version of [GMM97].

a modal operator (explaining the process of duplicating and/or erasing input data) followed by a linear function, it was soon discovered that the lambda-calculus may be faithfully embedded into linear logic, thus allowing the use of linear logic computations (in the form of proof net cut-elimination) to perform (or study) lambda-reduction. A crucial step was the discovery [GAL92] that Lamping's graph-reduction algorithm [Lam90] for optimal lambda-reduction (in the sense of Lévy, [Lév78]) could be interpreted as a way of performing proof net cut-elimination in a distributed and local way. The (global) concept of proof net box is replaced with information distributed on the graph (brackets, croissants, and indices). Cut-elimination is performed with a set of completely local graph-rewriting rules, main part of which are those manipulating the information added to the graph to (dynamically) reconstruct the boxes. The (potential) sharing (expressed with new nodes) of common subgraphs is the key to optimal reduction. Cut-elimination in these *sharing graphs* is based on three main ideas. First, in the reduction of a logical cut involving duplication of information, the duplication is not actually performed. It is instead indicated in a lazy way by the introduction of specific new nodes (*fans*). Second, new reduction rules are added to incrementally perform the required duplication. Third, there is a mechanism to recognize when this process of incremental and distributed duplication is over.

Sharing graphs have been revisited from different perspectives: a categorical interpretation (and new notation) [Asp95]; their extension to other logical systems [AL96]; their relations to the geometry of interaction [ADLR94]; a new notation ensuring better properties (in particular, that the normal form of a sharing graph be a proof net) [GMM96,Gue97]. For a detailed presentation of the connections between sharing graphs and 'optimal reductions' of λ-calculus we refer the reader to [AG98].

All these approaches differ in the specific way the bookkeeping information is coded into the sharing graph. However, they agree on their focus on what in [GMM96] we called *restricted* proof nets: weakening is not allowed. There are at least two reasons for this. First, the problems that weakening raises during the reduction of an arbitrary proof net do not show up during the reduction of a lambda-term (better: of the proof net corresponding to a lambda-term), even if weakening *is* allowed in the term. Second, the usual syntax for proof nets do not seem to allow for *any* solution to those problems (see Section 2).

We propose in this paper a set of completely local and asynchronous graph rewriting rules for cut-elimination in proof nets for the Multiplicative Exponential Linear Logic (with weakening and contraction but without constants), MELL. The proposed rules are proved strongly normalizing and confluent. Moreover, the normal form of a sharing graph is a proof net. This generalizes to MELL the results of [GMM96].

As in [GMM96], these results rely on a sharp distinction between logic and control. In standard sharing graphs, the nodes used to control the reduction process (fans, brackets, and croissants) have in fact also a static role, to introduce logical formulas. In our approach, instead, new information, in the form of indexes over formulas, are responsible for the static correctness (that is, for *logic*), while the *control* nodes (muxes) are responsible only for the duplication and reindexing during cut-elimination. This logic vs. control separation is rooted in our previous work on indexed systems for linear logic [MM95].

To treat weakening we exploit a well known *permutability* result: In the sequent calculus formulation of MELL, the weakening rule permutes with all the other rules, and hence it can be pushed upwards, to the axioms. Axioms may then be formulated as

$$\vdash \mathrm{p}, \mathrm{p}^{\perp}, ?\Gamma,$$

dropping an explicit weakening rule. When expressed in a suitable proof net setting, this idea always generates connected proof nets, allowing a local graph-rewriting cut-elimination. The approach may be seen as a specialization of that of Banach [Ban95].

In our setting, the cut-elimination of a box against a weakening may be performed in two (ideal) phases: first, a marking of the box to erase, keeping intact its logical structure; second, the actual erasing of the box, with the reorganization of its (secondary) doors as weakenings.

Together with the extension to the case with weakening, the main novelty of this paper w.r.t. [GMM96] is the proof technique for the technical results. As in [GMM96], we exploit the decomposition of rules for boxes in two steps, duplication and rearrangement of box nesting, that allows to reduce to the case of the so-called unshared reductions (*i.e.*, to a case in which duplication is performed globally, while box rearrangement is done step by step). But, instead to resort to some kind of algebraic semantics derived from the so-called Geometry of Interaction, the proof of the main properties of the unshared case are given by means of standard syntactical techniques (namely, by means of a direct analysis of confluence and strong normalization of the rewriting system).

The structure of the paper is as follows. Section 2 discusses the problems that weakening raises, informally introducing the techniques used in the sequel. Section 3 sets the stage with definitions and the relations with more usual formulations of proof nets. Section 4 introduces the rewriting rules. Section 5 proves the main properties of the reduction systems. In Section 6 we discuss some relations with optimality and possible extensions.

2 Weakening in Proof Net Reduction

Weakening in linear logic can produce boxes whose contents are disconnected, and, more subtly, such boxes can be generated by the cut-elimination procedure, even starting from proof nets whose boxes are connected. The crucial case (for cut-elimination) is that of a box whose principal door has as premise a weakening link. A more general situation is depicted in Figure 1 (left) (weakening boxes are not shown). The net σ is a correct proof net. The dotted region on the left is built starting from some weakenings and provides the principal door of a box. In a sequent proof, we would first construct the proof σ; then we would proceed with the weakenings; finally we would build π. We shall call the dotted region comprising π a *weakening isle*, for it is a separate connected component of the net. Moreover, a weakening isle is *not* a proof net by itself, *e.g.*, the global correctness of the net in Figure 1 is guaranteed by the presence of the proof net σ.

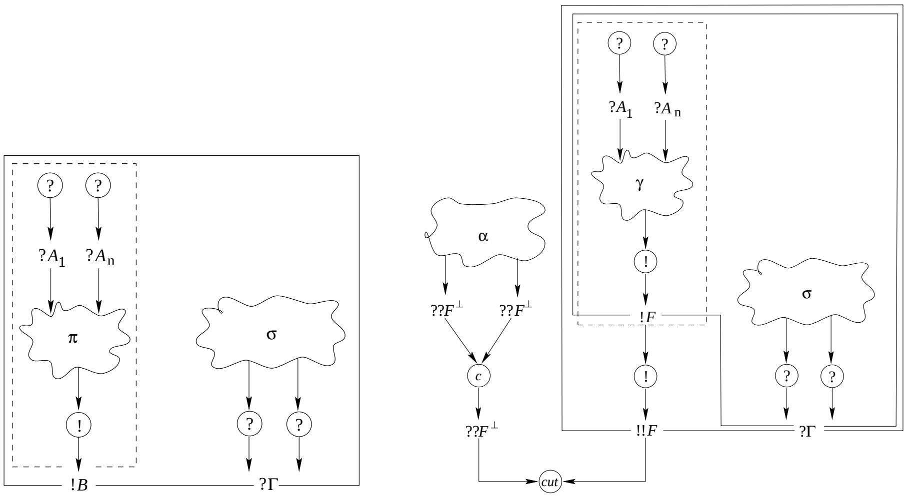


Fig. 1. A weakening isle.

Cut now the principal door of the box against a contraction, as shown in Figure 1 (right). The reduction of the cut causes the (global) duplication of the box, and the replacement of the cut with two cuts. But in sharing graphs boxes are not explicit. They may be reconstructed by means of the auxiliary information (brackets, indices, etc.) *through a graph exploration starting from their principal door*. And then we are lost. No matter which rules will be devised to rewrite the cut, if these rules are to be completely local, the σ part of the graph will never be affected by them and, hence, will not be duplicated. The result of any local rewriting of Figure 1 (right), then, cannot be much different from the graph shown in Figure 2. That graph is not the intended reduct of the cut; moreover is not a proof net, and it cannot be made into one by adding exponential boxes. There are two weakening isles, but only one copy of the proof net σ, which cannot validate both isles.

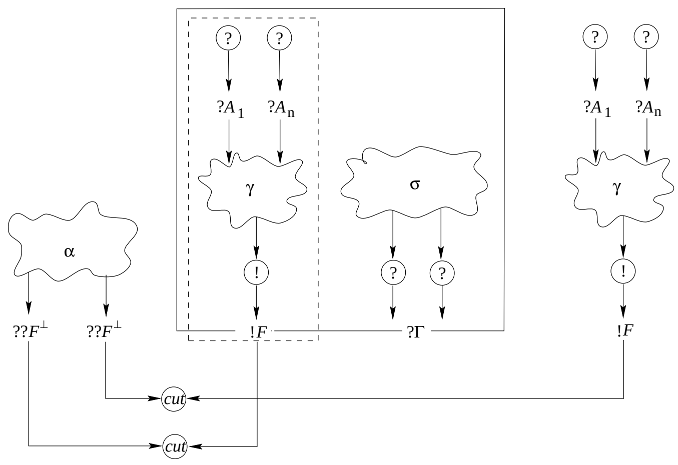


Fig. 2. A mistake.

To solve the connected components problem we change the definition of the axiom link. Beside the dual atoms p and $\mathrm{p}^\perp$, we attach a list of weakening formulas to an axiom link. There is no explicit weakening link. In this way a proof net is always connected and there is hope for a local exploration of its boxes. The reduction of the cut of Figure 1 (b) may now be done in the standard sharing graph way: duplicate (and move) the cuts and add a link (a fan, or a *mux* in our terminology) indicating the sharing of the two boxes. The actual duplication will be done incrementally, making the mux travel inside the box. Since any weakening formula is explicitly connected to some axiom, the mux will eventually visit all the box, ensuring the duplication of σ.

This formulation of weakening in proof nets is a variant of the technique introduced by Banach [Ban95]. To prove the so-called sequentialization theorem (that an acyclic and connected proof-structure comes indeed from a sequent linear logic proof and it is thus a proof net) for MELL, Banach introduced the notion of *probe*, an arc pointing 'back' from a weakening link to any other link of the net, thus guaranteeing connectedness. In the example of Figure 1 (a), a probe would connect each ? link on top left to one link (anyone is fine) of σ. However, this approach remains too liberal (in the choice of the target link of a probe) for the purpose of a distributed cut-elimination rewriting algorithm. In fact, this freedom is not necessary. Our formulation forces the target of a probe to always be an axiom contained in the same weakening box.

Besides the 'weakening isle' case, the other important situation involving weakening is that of a weakening formula (with its probe connecting to an axiom) cut against the principal door of a box, whose reduction is the erasing of the box and the 'relocation' of its secondary doors into weakenings. In our approach this will happen in two ideal phases. First (*mark*), weakening and cut are replaced with a mux (connected through the probe to an axiom), which will explore the box, marking the links for deletion, but preserving the logical structure. This mux will stop its marking at the border of the box, like any other mux. Second (*sweep*), starting from the marked axioms, the box will be erased, reducing it to a special 'garbage collector' link, which will collect all the secondary doors of the box. At the end, these secondary doors will be transformed into weakenings, with probes toward the axiom connected to the original weakening.

3 Proof Nets and ℓ-nets

3.1 Leveled Structures

As in [GMM96], we shall assume that nets are hypergraphs in which the hypernodes are indexed formulas and the hyperedges are links corresponding to the rules of MELL, to a sharing operator named mux (in some sense, the extension of contraction) and to an unsharing operator named demux (which is the complement of a mux and therefore a sort of duplication). In addition to these links, we shall have a new link (the *garbage collector*, or simply collector), that will implement the garbage collecting process. Let us be more formal.

(i) An *indexed formula* $X^n, Y^n, W^n, \ldots$ is a formula—either a MELL formula $A, B, C, \ldots$ or the extra-logical constant $\varnothing$, read *dummy*—whose superscript n, named the *level* of the formula, is a natural number. (We shall not be pedantic on this issue, but in the following we will use 'formula' as a synonym of 'occurrence of an indexed formula'.)

(ii) A *link* $u, v, w, \ldots$ (see Figure 3) is an oriented hyperedge (*i.e.*, a pair of disjoint sets of indexed formulas, respectively named the source and the target of the link) labeled by a symbol from the denumerable alphabet of labels $\mathbb{L}$.

$$
\begin{aligned}
\mathbb{L} ::= {} & \mathsf{ax} \mid \mathsf{cut} \mid \parr \mid \otimes \mid ! \mid \mathsf{d} \mid \mathsf{w} \mid \bullet \\
& \mid \mathsf{mux}[0] \mid \mathsf{mux}[1] \mid \ldots \mid \mathsf{mux}[n] \mid \ldots \\
& \mid \mathsf{demux}[0] \mid \mathsf{demux}[1] \mid \ldots \mid \mathsf{demux}[n] \mid \ldots \\
& \mid \mathsf{gc}[0] \mid \mathsf{gc}[1] \mid \ldots \mid \mathsf{gc}[n] \mid \ldots
\end{aligned}
$$

Where the natural number i of a label $\mathsf{mux}[i]$, $\mathsf{demux}[i]$ or $\mathsf{gc}[i]$ is named the *threshold* of the corresponding link.

(iii) The formulas in the source of a link u are the *premises* of u; the formulas in the target of u are the *conclusions* of u. Each formula is conclusion of exactly one link and premise of at most one link.

(iv) The number, the levels and the shape of the premises/conclusions of a link must accord with the patterns in Figure 3, where p is an atomic formula and A, B are MELL formulas (therefore, $A, B \neq \varnothing$). In particular, we remark that:

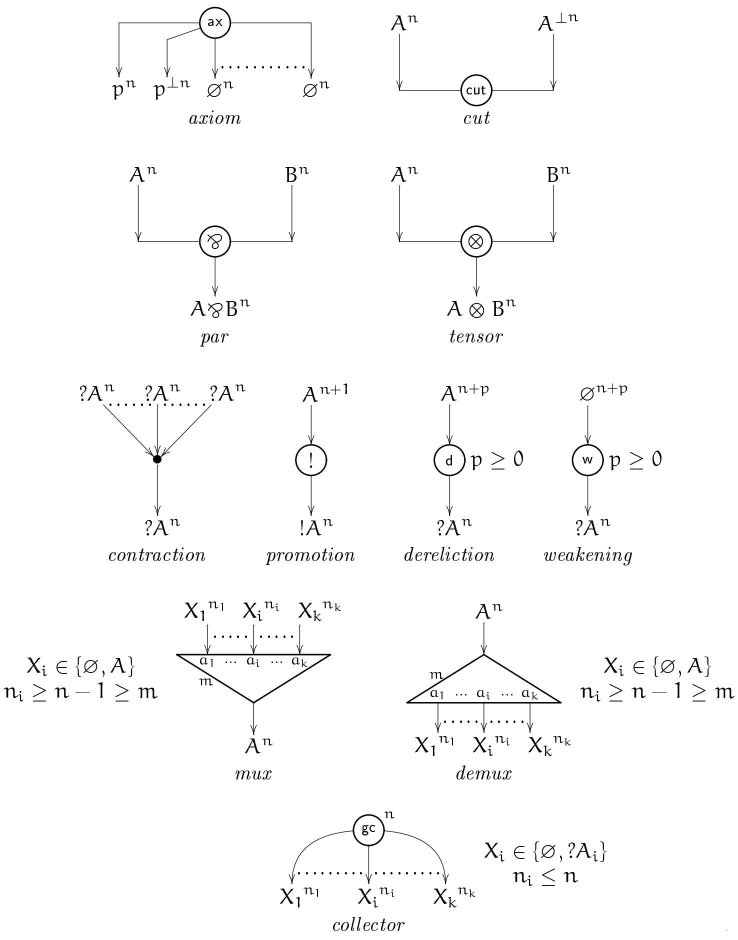


Fig. 3. Links.

(a) In addition to its *principal conclusions* p and $p^\perp$, an axiom may have an arbitrary number of conclusions $\varnothing$.
(b) The contraction has always $k > 1$ premises.
(c) A collector has $k > 0$ conclusions.
(d) A mux has $k > 0$ premises and only one conclusion. Moreover, any premise X_i of the mux is either $\varnothing$ or equal to the conclusion A of the mux. A demux is obtained from a mux exchanging the role of premises and conclusions.

(v) The only exceptions to Figure 3 are the so-called *erased* links, in which all the premises and conclusions are equal to $\varnothing$. A link with at least a premise/conclusion different from $\varnothing$ is said *alive*. Therefore, for any alive link u we have that:
(a) When u is an axiom, the link u has two *principal conclusions* that are atomic formulas, while all the other conclusions of the link are $\varnothing$.
(b) When u is a cut, a par, a tensor, a promotion, a dereliction or a contraction link no premise/conclusion of u is equal to $\varnothing$.
(c) When u is a collector, all the conclusions of u different from $\varnothing$ are why-not formulas $?A$.

(d) When $\mathtt{u}$ is a weakening, its conclusion is not $\varnothing$, while its premise is always $\varnothing$.
(e) When $\mathtt{u}$ is a (de)mux, its (premise) conclusion is not $\varnothing$.

(vi) We shall refer to both dereliction and weakening as why-not (?) links. Therefore, especially in picture, we will use the label ? to denote a link whose label can be d or w. It is indeed immediate that an alive d link is a ? whose premise is not $\varnothing$, while an alive w link is a ? whose premise is $\varnothing$.

(vii) In accordance with the choice of a $\mathtt{k}$-ary version of contraction, we assume that no premise of a contraction link can be the conclusion of another contraction link. In other words, a tree of contraction links is always 'contracted' into a unique contraction link with the root of the tree as conclusion and with the leaves of the tree as premises.

(viii) The premises and conclusions of any link but the contraction are assumed to be distinguishable (*i.e.*, the source and the target of the link are ordered sets). Namely,
(a) cut, $⅋$ and $\otimes$ have a left and a right premise;
(b) if the link $\mathtt{u}$ is an axiom or a collector, we will have the 1st, the 2nd, ... , the $\mathtt{k}$th conclusion of $\mathtt{u}$;
(c) we remind that the premises of a contraction are indistinguishable (in fact, the source of a contraction is not ordered).

(ix) Given a (de)mux $\mathtt{u}$ with $\mathtt{k}$ (conclusions) premises, we say that $\mathtt{u}$ has $\mathtt{k}$ *ports* or that $\mathtt{u}$ is $\mathtt{k}$-ary. Unary (de)muxes are also called (negative) *lifts*. Each port of $\mathtt{u}$ is named by a label chosen over a denumerable alphabet

(x) Each port of a (de)mux is of kind *identity* or of kind *garbage*. In the first case, the (conclusion) premise connected to that port is equal to the (premise) conclusion of the (de)mux; in the second case, such a (conclusion) premise is always $\varnothing$. Let us remark that the converse is not true, for in an erased mux $\mathtt{u}$ also the premises connected to the ports of kind identity are $\varnothing$; nevertheless, also in this case such premises are equal to the conclusion of the mux.

(xi) The difference between the level of the formula at a given port $\mathtt{a}$ and the conclusion of a mux $\mathtt{u}$ (the premise of a demux $\mathtt{u}$) is the *offset* of the port $\mathtt{a}$. We shall see that the offset (and the kind) is a property of the port, that is, it is invariant under reduction.

(xii) The nodes that are not the premise of any link are the *net conclusions*.

Definition 1 ($\mathsf{s}\ell$-structure) *An* $\mathsf{s}\ell$-structure *(sharing leveled structure) of links is a finite connected hypergraph whose nodes are indexed formulas and whose hyperedges are links.*

With respect to the usual presentation of proof nets, $\mathsf{s}\ell$-structures have two additional types of link (mux/demux and gc) and a new constant ($\varnothing$).

The $\varnothing$ is used (at first) for introducing weakening formulas. During the cut-elimination process, moreover, $\varnothing$ will be used to mark those parts of the structure that have to be discarded (because cut against a weakening).

Muxes (introduced in [Gue96,GMM96]) are responsible for the processes of:

(i) reindexing of formulas (that is, the local re-computation of boxes during reduction);
(ii) local (lazy) duplication;
(iii) marking of garbage.

The garbage collector link is designed to collect the garbage, that is, to remove from the net those nodes which have been marked $\varnothing$.

It should be clear already from these short remarks that muxes and garbage collectors are not necessary to represent a proof net. They are a sort of intermediate structures used for 'implementing' cut elimination. According to this we may define the so-called leveled structures of links.

Definition 2 (proof ℓ-structure) *A* proof ℓ-structure *of links is an* $\mathsf{s}\ell$-*structure without (de)muxes and garbage collector links.*

A proof ℓ-structure *does not contain garbage*, when all its links are alive.

Remark 3 The key technical point of our approach is avoiding nullary premise links to introduce weakening formulas. In fact, a $\varnothing$ premise always connect a weakening to an axiom. More precisely, in a proof ℓ-structure G which does not contain garbage, any weakening is immediately below an axiom, for $\varnothing$ cannot be the conclusion of any other kind of link. More generally, when G contains garbage or G is an $\mathsf{s}\ell$-structure, the premise of the weakening may be the conclusion of some erased link. Nevertheless, the key idea is that while collecting the erased links in the structure, this $\varnothing$ formula will eventually become the conclusion of some axiom.

3.2 Decoration

It is well known (see [Gir87], pag. 63) that proof nets may be formulated with *several weakenings in the same weakening box.* With this formulation, it is easy to show (a trivial induction) that by suitable permutations any proof net can be transformed into an equivalent one in which each weakening box contains exactly one axiom link as interior. Let PN be the set of proof nets with such a structure. Since any weakening box contains exactly one axiom link, we may forget the boxes and simply record the weakening formulas with each axiom. Only exponential boxes survive in PN.

We will now show how to associate to each $\mathsf{P} \in \mathsf{PN}$ a (unique!) proof ℓ-structure $\mathcal{D}[\mathsf{P}]$, the *decoration* of P. The proof ℓ-net $\mathcal{D}[\mathsf{P}]$ is obtained by applying the following steps:

(i) assign to each node of P a level corresponding to the number of exponential boxes containing that node;
(ii) connect each weakening $?\mathsf{A}$ to an axiom α belonging to the weakening box of $?\mathsf{A}$ by means of a node labeled $\varnothing$;
(iii) set the level of this $\varnothing$ premise to the number of exponential boxes containing α.

Definition 4 *A proof ℓ-structure* S *is a* proof ℓ-net *iff* $\mathsf{S} = \mathcal{D}[\mathsf{P}]$ *for some* $\mathsf{P} \in \mathsf{PN}$.

By means of indexes it is possible to recover exponential boxes:

Definition 5 *Let* S *be a proof ℓ-structure and let* $!\mathsf{A}^{\mathsf{k}}$ *be the conclusion of an $!$-link* $\mathfrak{l}$. *The* box *of* $\mathfrak{l}$ *is a subset of links* $\mathsf{box}_{\mathsf{S}}[\mathfrak{l}]$ *of* S *verifying the following properties:*

(i) $\mathfrak{l} \in \mathsf{box}_{\mathsf{S}}[\mathfrak{l}]$*;*
(ii) the subnet N *induced by* $\mathsf{box}_{\mathsf{S}}[\mathfrak{l}]$ *is connected;*
(iii) each premise of a link in $\mathsf{box}_{\mathsf{S}}[\mathfrak{l}]$ *belongs to* N*;*
(iv) the formula $!\mathsf{A}^{\mathsf{k}}$*, that is a conclusion of* N*, is the* principal door *of the box;*
(v) any other conclusion $\mathsf{C}^{\mathfrak{m}}$ *of* N *is an* auxiliary door *of the box; all the auxiliary conclusions of a box are $?$-formulas,* i.e., $\mathsf{C}^{\mathfrak{m}} = ?\mathsf{B}^{\mathfrak{m}}$.
(vi) $\mathfrak{m} \leq \mathsf{k}$*, for each auxiliary door* $?\mathsf{B}^{\mathfrak{m}}$*;*
(vii) $\mathfrak{m} > \mathsf{k}$*, for each* $\mathsf{C}^{\mathfrak{m}}$ *internal to* N*,* i.e., $\mathsf{C}^{\mathfrak{m}}$ *is a formula in* N*, but it is not a conclusion of* N.

We denote by $\mathsf{BOX}[\mathsf{S}]$ the set of boxes of S. We remark that, because of the definition of ℓ-structure, boxes are connected.

3.3 Standard beta reduction

At a first level, cut elimination for proof ℓ-nets (in analogy with λ-calculus we shall call it β-reduction) may be seen as the natural generalization of classical proof net cut-elimination. The only difference is that the cut elimination rule for the exponential modalities works on indexes too (let us remark that we merge in a unique rule the cases of contraction, dereliction and weakening). The schema of the rule is given in Figure 4 (the boxes shown in the figure are computed according to Definition 5). Since in the following we will implement β-reduction in a shared way, in the following we will refer to the rule in Figure 4, plus the rule for multiplicative connectives, as *standard beta reduction*, or β_s. Moreover, we will denote by $G \twoheadrightarrow_s G'$ the fact that G reduces to G' by application of a β_s-rule.

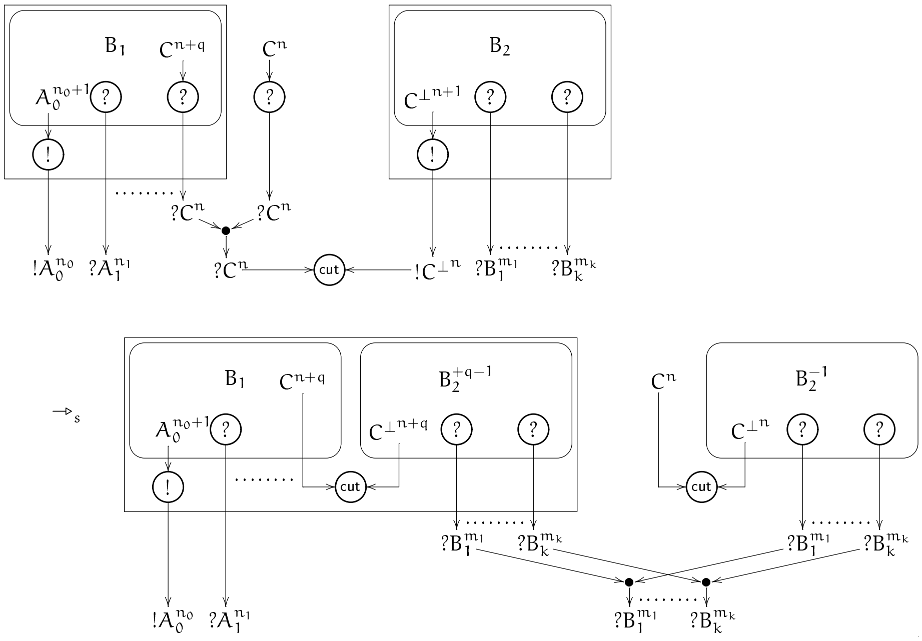


Fig. 4. Standard β-rule.

4 Sharing reduction

The new information that we added to proof nets allows a clear notion of local and asynchronous computation as a fully distributed execution of the standard cut-elimination process (as defined by [Gir87]). The reduction procedure is described by rules of three kinds: logical reduction (β_l), bookkeeping or mux propagation (π), garbage collection ($\sigma_\varnothing$).

4.1 Logical reductions

The *logical rules* β_l in Figure 5 implement a local version of the standard cut-elimination process. The only difference with the standard process is when an exponential cut is reduced. In this case, no duplication, reindexing, or erasing of boxes is done. Such operations are only indicated by the introduction of suitable muxes.

Cuts involving links whose premises/conclusions are formulas of type $\varnothing$ are not redexes. That corresponds to the intuition that there is no reason to reduce a cut in a subnet that will be erased.

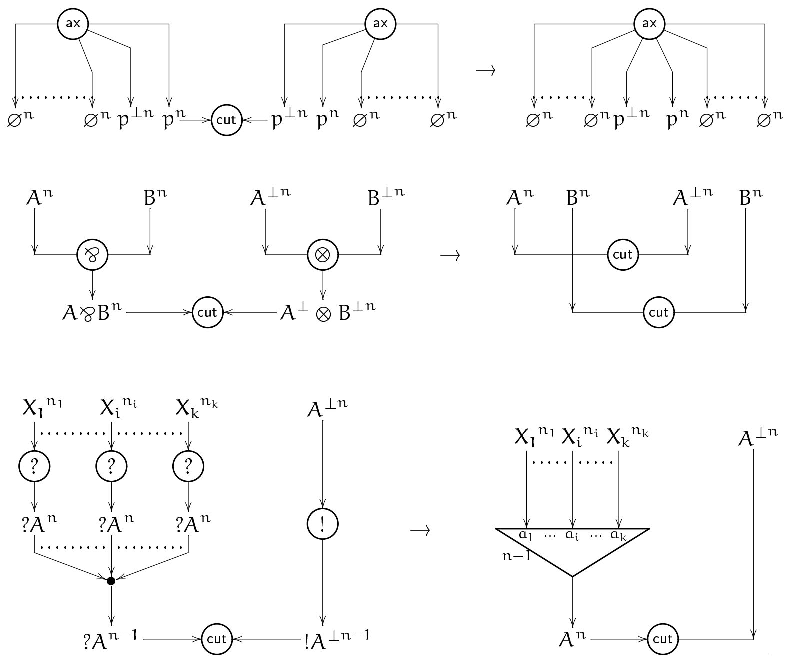


Fig. 5. Cut elimination rules: β_l.

Remark 6 The last rule of Figure 5 (elimination of an exponential cut) is the rule that creates muxes. In the new mux inserted by this rule, a port a_i is of kind *garbage* when the corresponding formula X_i was the premise of a w link, otherwise, when X_i was the premise of a d link, a_i is of kind *identity*.

4.2 Mux propagation rules

The *bookkeeping* or *propagation rules* (π) are responsible for the incremental reindexing and duplication/erasing of a box. Muxes will travel along the net, instantiating (according to the mux arity) and reindexing all the links they find. The general pattern for these rules is given in Figure 6, where $\star$ stands for any label but mux.

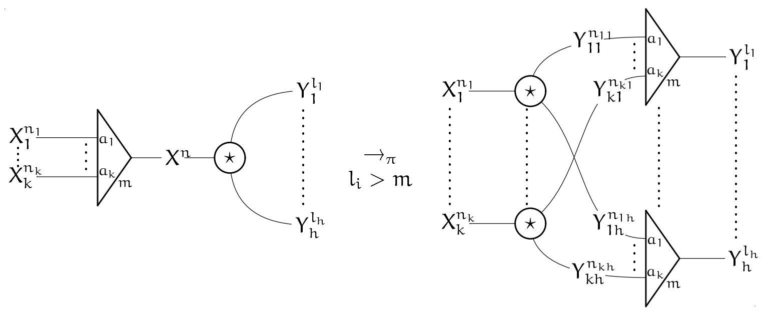
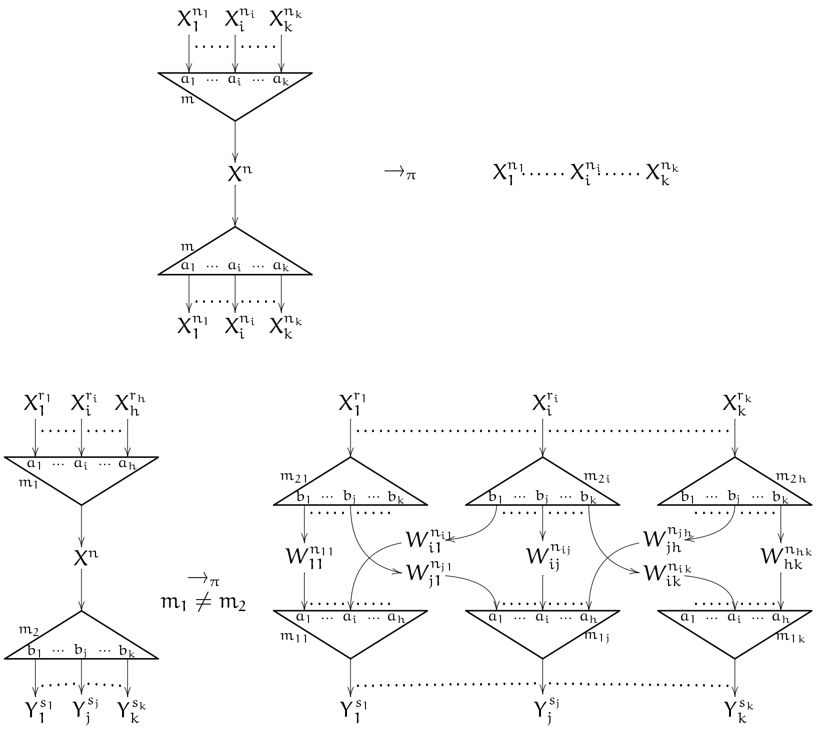


Fig. 6. Mux propagations.

When a mux faces another mux, see Figure 7, we must distinguish two cases.

(i) In the first case, the so called *mux annihilation* (on the top in Figure 7), the two muxes face with the same threshold. This denotes that that pair of facing muxes had been generated by the same exponential cut and that muxes' job is complete. According to this, the name, the kinds and the offsets of the ports of the two muxes must match, denoting in this way that each mux has created the same number of instances, increasing the levels by the same offset. Formulas connected to matching ports correspond then to instances of the same formula and can be merged, causing the pair of matching muxes to disappear.

(ii) In the second case, the so called *mux swap* (on the bottom in Figure 7), the two muxes face with different thresholds. In this case they duplicate each other and continue their task after swapping their positions. The formulas W_s are determined using the kinds of the ports. The offsets are preserved. The new values of the thresholds are computed according to the fact that 'the mux u_l with a lower threshold is operating inside a box that contains the scope of the mux u_h with the higher threshold'. Therefore, the threshold of any instance of u_l is unchanged, while the threshold of the ith instances of u_h is increased by the offset of the ith port of u_l. According to the general pattern of the mux propagation rules, we may say that u_l operates on u_h (duplicating the link and increasing its threshold).



$$n_{ij} = r_i + s_j - n$$
$$m_{1j} = m_1 + \delta \cdot (s_j - n_j) \qquad m_{2i} = m_2 + (1 - \delta) \cdot (r_i - n_i)$$
$$\delta = \texttt{if } m_1 > m_2 \texttt{ then } 1 \texttt{ else } 0$$

Fig. 7. Mux interactions.

Remark 7 In the case of mux swap, the pair of facing muxes might even be generated by the same cut. The remarkable point for the application of the rule is that their thresholds are different. An unabridged discussion on this issue can be found in [Lam90,Asp95,AG98]. Here, we just remark that this is the essential reason because of which muxes and the other control nodes have an index that changes along reduction. In fact, a static label assigned to each mux at its creation would not be able to properly decide when to apply annihilation or swap.

The propagation of muxes ends when they annihilate or when they reach an auxiliary door of the box corresponding to their scope. In this case, see Figure 8, the mux disappears, absorbed by a ?–contraction pair. The result is a new ? link for each port of the mux. Namely, for each port of kind identity we have a new dereliction, while for each port of kind garbage we have a new weakening.

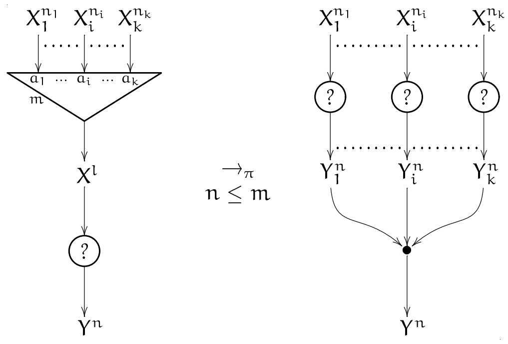


Fig. 8. Mux absorption: ? link case.

Remark 8 In the lhs of the ? absorption rule we recognize that the mux has reached the border of its scope from the fact that the level of the ? conclusion is greater to or equal than the threshold of the mux. This corresponds to the intended interpretation of the threshold: a mux cannot operate on formulas or links whose level is lower or equal than its threshold (see the discussion on the mux swap rule also).

4.3 Example

Let us consider the first $s\ell$-net represented in Figure 9 (call it S_1). In S_1 there is a lift with a port of kind identity marked by a. Such a lift will perform a reindexing process. After the execution of a π step we obtain the $s\ell$-net S_2 (in the middle of Figure 9). Let us consider the lift with $\varnothing^9$ as premise (call it L). It is important to highlight that L *does not behave like an eraser*, even if the L premise is a $\varnothing$ node. In order to make a lift an eraser it is necessary that, at the creation time (because of an exponential cut involving weakening), it had the port of kind *garbage*. After two propagation steps $S_2 \to_\pi \to_\pi S_3$ (where S_3

is the $s\ell$-net on the right hand side of Figure 9), the lift L has performed a correct reindexing, but no erasing.

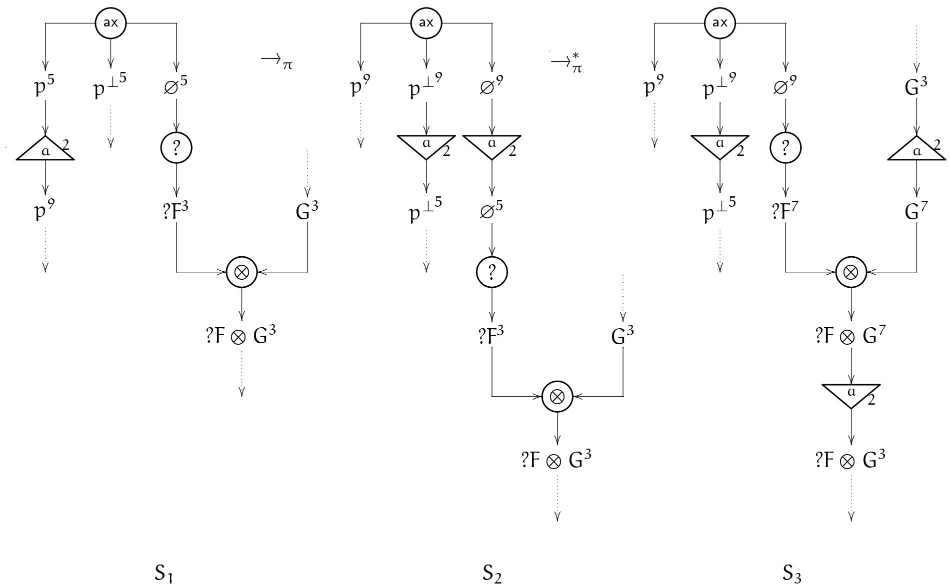


Fig. 9. Example: a non erasing lift.

4.4 Erasing of links

It remains to be discussed how the bookkeeping rules handle the marking of boxes to be erased (because cut against weakenings). Here come to the stage the *kinds* on the ports of the muxes.

First, let us recall that in the last rule of Figure 5, for any port a_i of kind garbage the formula X_i is $\varnothing$. Then, let us consider Figure 6. When $X_i = \varnothing$ and a_i is of kind garbage, we stipulate that $Y_{i1} = \ldots = Y_{ih} = \varnothing$, that is, the ith copy of the duplicated link has been erased. Otherwise, when a_i is an identity port, the formula $Y_{i1} = Y_1, \ldots, Y_{ih} = Y_h$. The same convention applies to all the other rules involving a mux propagation (Figures 7 and 13).

4.5 Example

Let us consider the three $s\ell$-nets S_1, S_2 and S_3 depicted (in the given order) in Figure 10. The example shows how a mux with two ports, one of kind identity and one of kind garbage, can be activated.

In S_1 the left premise of the cut (namely, $?(G^\perp \parr (D^\perp \otimes E^\perp))^6$) is obtained by means of the contraction of the conclusions of a weakening and of a dereliction.

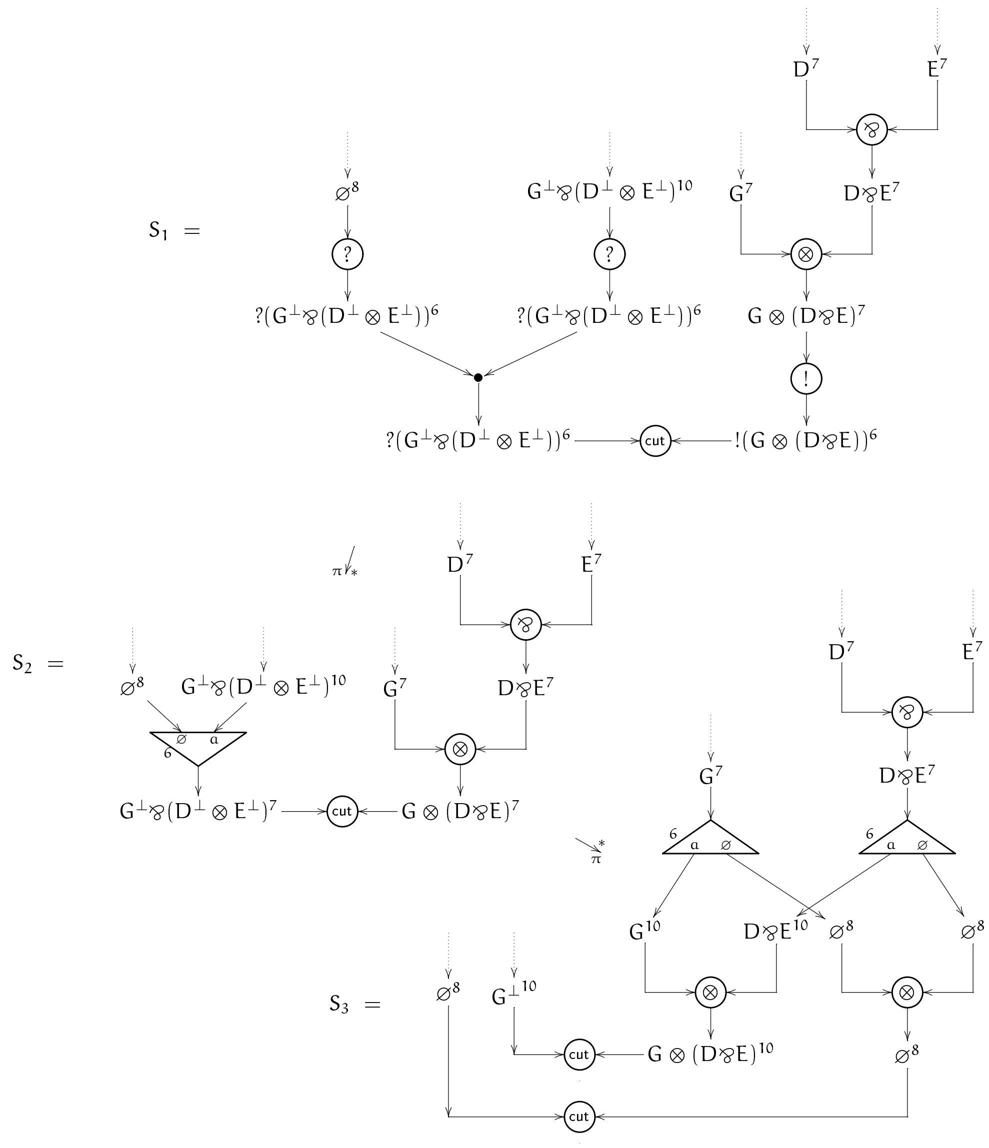


Fig. 10. Example: reindexing plus erasing.

As a consequence, the reduction step $S_1 \rightarrow_{\beta_\iota} S_2$ activates a mux link with a port of kind *garbage* (denoted by $\varnothing$), corresponding to the node introduced by the weakening, and another port of kind identity (denoted with a), corresponding to the node introduced by the dereliction. Starting from S_2 the mux must perform (in parallel) the following two processes: (*i*) duplication with reindexing; (*ii*) erasing (*i.e.*, marking by $\varnothing$).

Note that, after the two π reduction steps $S_2 \rightarrow_\pi\rightarrow_\pi S_3$, the mux has duplicated the links cut and $\otimes$. At the same time, one of the two duplicated subnets has all the nodes marked by $\varnothing$.

Let us analyze the reindexing process performed by the mux. In the reduction steps $S_2 \rightarrow_\pi\rightarrow_\pi S_3$ the mux increments by 1 (from level 7 to level 8) the nodes of the erased subnet and by 3 (from level 7 to level 10) the nodes of the other subnet.

4.6 Garbage Collection

During its propagation, a mux with a garbage port creates an erased instance of the net that it visits. The collection of the garbage left by such a mux is demanded to the gc link. In particular, garbage collection is demanded to the $\sigma_{\varnothing}$ rules in Figure 11, 12, 13 and 14, which correspond to the implementation of a parsing algorithm for garbage subnets (see [GM96] for the problem of parsing proof nets).

Garbage collection starts from erased axioms, transforming them into gc links, see Figure 11.

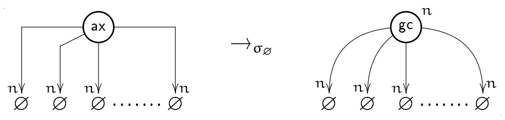


Fig. 11. Sweep rules: start collecting.

Subsequently (see Figure 12), the gc link keeps 'eating' the dummy marked net, collapsing it into a single gc link and collecting all the secondary doors of the box to be deleted (see the last rule of Figure 12).

Note that gc does not delete muxes. The interaction between muxes and gc links is regulated by the rule in Figure 13. This rule sums up the general mux propagation rule and the absorption. The kary demux connected to the conclusion X_0 of the garbage collector u creates k copies $u_1, \dots u_k$ of u. This implies that some kind of contraction must be inserted between the hth conclusion of u_l and the formula X_h that was the hth conclusion of u. We can distinguish two cases, corresponding to the ith and the jth conclusion in the picture, respectively.

Absorption The level of A_i is $n_i \leq m$. In this case, the demux is absorbed and replaced by a real contraction. This agrees with the constraints on the shape of the formulas near to a contraction. In fact, it is easy to convince oneself that A_i is either a $\varnothing$ or a why-not formula $?B_i$. In the first case, we have inserted an erased contraction, that does not cause any trouble; in the second case, we have a correct case of contraction.

Propagation The level of A_j is $n_j > m$. In this case, we have a standard propagation of the mux. The jth conclusion A_{lj} of the lth garbage collection link is lifted and (possibly) changed according to the kind of the port a_l.

Garbage collection ends when the box to be erased has been collapsed into a unique gc node. The conclusions of this node are the secondary doors of

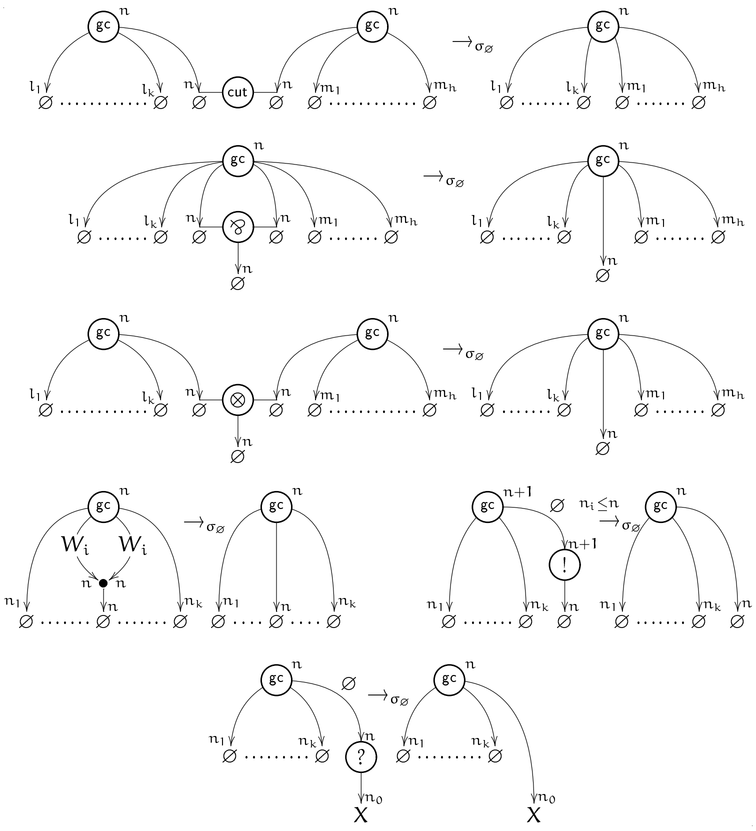


Fig. 12. Sweep rules.

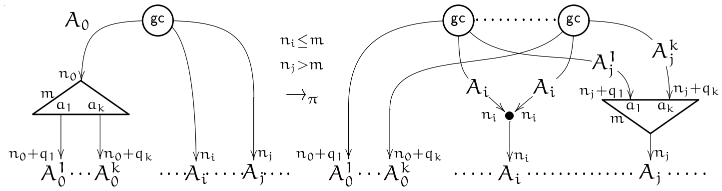


Fig. 13. Mux propagation/absorption: gc link case.

the box, plus a single $\varnothing$ conclusion, cut against another $\varnothing$ (coming from the original weakening). This configuration and the corresponding rewriting are depicted in Figure 14. Let us observe that the gc link disappears and that the secondary doors of the collected box are transformed into weakenings.

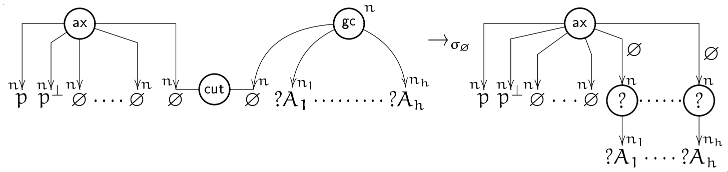


Fig. 14. Sweep rules: end collecting.

4.7 Example

In Figures 15 and 16 we have depicted a complete sequence of reductions involving *erasing* (mark process) and *garbage collection* (sweep process).

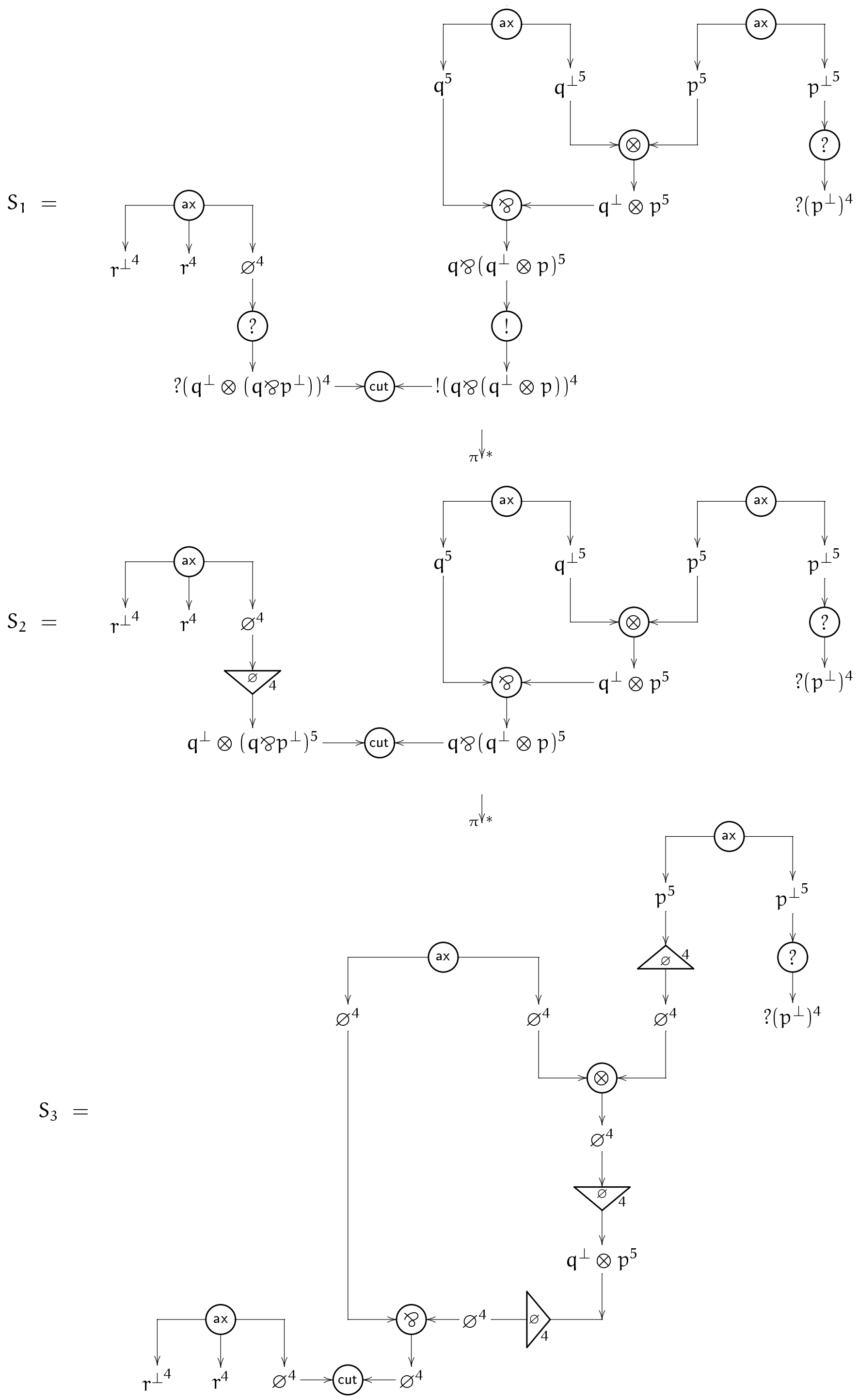


Fig. 15. Mark and sweep: starting

The $s\ell$-net S_1 contains a redex given by a cut against weakening (with conclusion $?(\mathsf{q}^{\perp} \otimes (\mathsf{q} \parr \mathsf{p}^{\perp}))^4$. After the reduction step $\mathsf{S}_1 \to_{\beta_\iota} \mathsf{S}_2$ we obtain the $s\ell$-net S_2 with a new lift link with port of kind *garbage* (denoted by a $\varnothing$). The π rule allows us to propagate the lift as in the reduction sequence $\mathsf{S}_2 \to_\pi \to_\pi \to_\pi \mathsf{S}_3$. In the $s\ell$-net S_3 we are now ready to start the sweep process. Note that we can start sweeping even if the lifts have not yet completed the marking. The reduction steps $\mathsf{S}_3 \to_{\sigma_\varnothing} \to_\pi \to_{\sigma_\varnothing} \mathsf{S}_4$ propagate a lift and activate the sweep process (rewriting the two axioms of S_3 as gc link). The $s\ell$-net S_4 has three redexes involving: (*i*) the collection of the $\otimes$ link, (*ii*) the absorption of a lift, (*iii*) the annihilation of the two complementary lifts. Performing the corresponding reduction steps *in any order* we have $\mathsf{S}_4 \to^* \mathsf{S}_5$. The $s\ell$-net S_5 does not contains lift links but a gc link only. In two sweep steps we can complete the garbage collection; namely, $\mathsf{S}_5 \to_{\sigma_\varnothing} \mathsf{S}_6 \to_{\sigma_\varnothing} \mathsf{S}_7$.

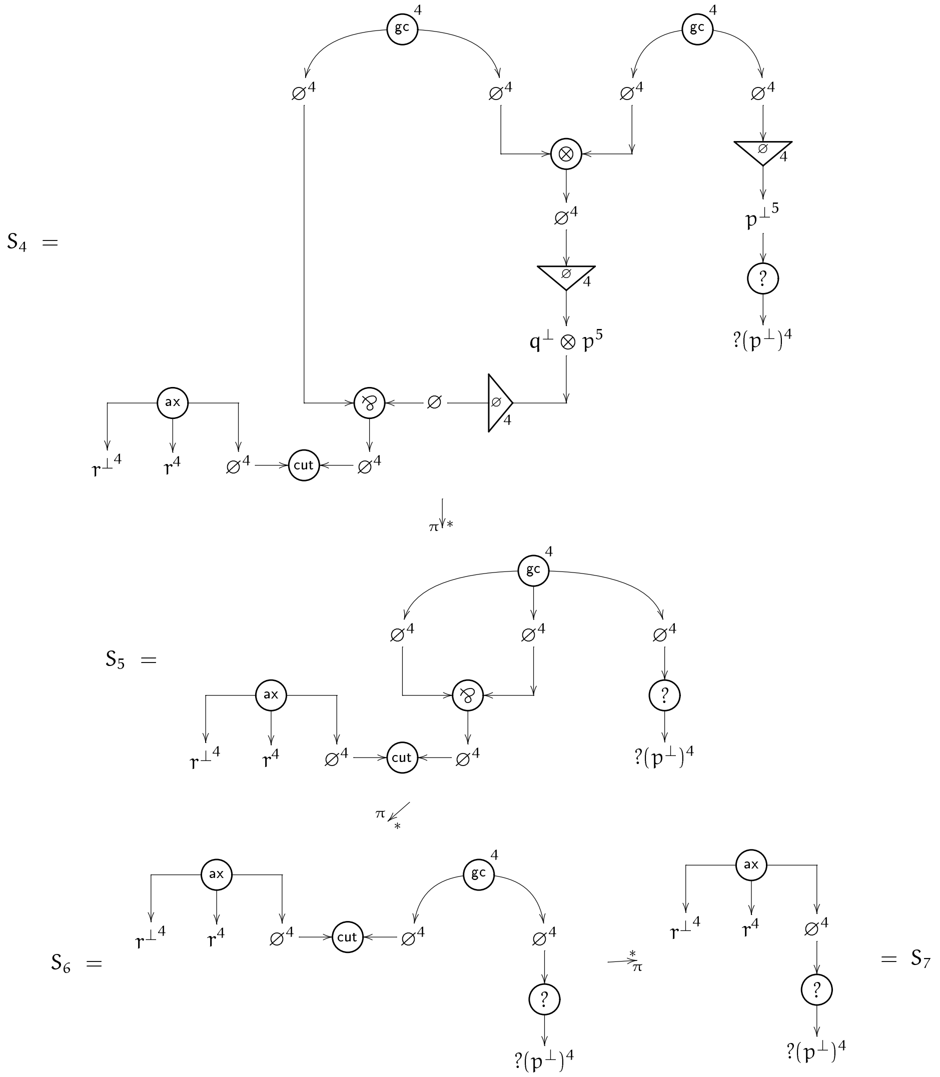


Fig. 16. Mark and sweep: ending

5 Results

We may summarize the main result of the paper as:

> cut elimination (with garbage collection) in proof nets may be performed in a completely local and asynchronous way.

In more details, let us say that the $s\ell$-structure G is a *proof $s\ell$-net* when $\mathsf{N} \to^* \mathsf{G}$, for some proof ℓ-net N. We will show that, for any proof $s\ell$-net G:

Theorem 40 The $\pi + \sigma_\varnothing$ rules are strongly normalizing and confluent on G. Moreover, the unique $\pi + \sigma_\varnothing$ normal form $\mathcal{R}(\mathsf{G})$ of G, named the *readback* of G, is a proof ℓ-net.

Theorem 41 Any standard reduction of a proof ℓ-net N can be simulated by a β_u-reduction followed by a read-back reduction, where by readback reduction we mean a sequence of π+$\sigma_\varnothing$ rules ending with a proof ℓ-net.

Theorem 42 The sharing reduction is correct. More precisely, if N is a proof ℓ-net s.t. $\mathsf{N} \to^*_{\mathsf{G}}$, then $\mathsf{N} \twoheadrightarrow^*_s \mathcal{R}\mathsf{G}$.

Theorem 44 The sharing reduction of G is strongly normalizing.

Theorem 45 The sharing reduction of G is confluent.

Corollary 46 The $\beta_l + \pi + \sigma_\varnothing$ normal form of a proof ℓ-net N is unique and coincides with its β_s normal form.

The main difference between the proof techniques that we have used here an the ones given in [GMM96,Gue97] is that we do not resort to any kind of algebraic semantics derived from the so-called Geometry of Interaction. In fact, the main properties of the unshared reductions (see subsection 5.4) are all proved by means of standard syntactical techniques. Namely, by a direct analysis of confluence and strong normalization of the rewriting system.

5.1 Notations

The proof of the properties summarized above requires a detailed analysis of several reduction strategies and of their properties. This will also require the introduction of a large amount of notations.

To help the reader in getting through to the more technical part of the paper, we summarize in Figure 17 the main reduction relations that we will use in the following section and the corresponding arrows. As usual, we shall use starred arrows to denote the transitive and reflexive closure of the corresponding reduction relation; moreover, in diagram construction, we shall use a dotted arrow to denote the existence of at least one reduction corresponding to such an arrow.

arrow	reduction
$\rightharpoondown_s$	β_s (standard)
$\rightharpoondown_l$	β_l (local)
$\rightharpoondown_u$	β_u (unshared)
$\rightarrow_\alpha$	$\alpha \in \{\pi, \sigma_\varnothing, \sigma_\bullet, \sigma\}$
$\Rightarrow$	$\pi + \sigma_\varnothing$
$\Rightarrow_\bullet$	$\pi + \sigma$
$\rightarrow$	$\pi + \sigma_\varnothing + \beta_l$
$\rightarrow_\bullet$	$\pi + \sigma + \beta_l$
$\rightarrow_u$	$\pi + \sigma_\varnothing + \beta_u$

Fig. 17. Reduction relations.

Instead, in Figure 18, we list some predicates that will be defined and used in the following sections.

predicate	definition
$\mathsf{CORR}(G)$	$G \Rightarrow_\bullet^* \mathsf{net}_\Gamma$.
$\mathsf{DF}_\pi(G)$	G is deadlock free.
$\mathsf{SN}_\alpha(G)$	G is strongly normalizing w.r.t. a set α of rewriting rules.
$\mathcal{R}_\pi(G)$	π normal form of G;
$\mathcal{R}(G)$	$\pi + \sigma_\varnothing$ normal form of G.

Fig. 18. Useful predicates.

5.2 Parsing

We extend $s\ell$-structures to parsing $s\ell$-structures by introducing a new link called net. A net is a link with k conclusions and no premises. There is no restriction on the shape and number of the conclusions of net links. However, because of the definition of parsing, a net link corresponds always to a correct subnet of the initial net.

Definition 9 (parsing) *Let us denote by $\sigma_\bullet$ the set of rewriting rules derived from the sweep rules $\sigma_\varnothing$ by:*

(i) Replacing each gc *link by a* net *link.*
(ii) Relaxing the condition on the type of the formulae involved in the redexes, i.e.*, the dummy nodes are replaced by ordinary* MELL *formulae according to the usual constraints established by the types of the links.*

We denote by σ *the union of* $\sigma_{\varnothing}$ *and* $\sigma_{\bullet}$*.*

For a detailed discussion of parsing see [GM96]. We only stress that a proof ℓ-structure is a proof ℓ-net iff it contracts by the parsing rules to a structure formed of a net link only.

Theorem 10 *A proof* ℓ*-structure* G *with conclusions* Γ *is a proof* ℓ*-net iff* $\mathsf{G} \to^{*}_{\sigma_{\bullet}} \mathsf{net}_{\Gamma}$*, where* net_{Γ} *denotes a structure formed of a single* net *link with conclusions* Γ*.*

5.3 Deadlock

A mux/demux is *deadlocked* when its principal port—its conclusion in the case of a mux, or its premise in the case of a demux—is connected to any port of another link but a secondary port of a mux and does not form a π redex.

In particular, a mux with threshold n is deadlocked when its conclusion is the premise of a non-complementary demux (i.e. a demux with the same threshold n but with different port names), or when its conclusion is the premise of an of-course link whose conclusion is at level n. These configurations are called *deadlocks*, for it is immediate that no reduction can remove them.

An $s\ell$-structure is *deadlock free* when none of its reducts contains a deadlock.

5.4 The unshared case

Following the mainstream of [Gue96,GMM96] the proofs of the results split in two main parts:

(i) The definition and analysis of the unshared case.
(ii) The lifting to the shared case via a simulation lemma stating that each shared reduction induces a corresponding unshared one.

5.4.1 Unshared ℓ-structures and unshared reduction

In unshared structures, $u\ell$-structures for short, boxes are still present and used to define the unshared beta reduction β_u, in which, even if box reindexing is performed by unary muxes, box duplication is done in a global way.

An *unshared* $\mathsf{s}\ell$*-structure* is a proof $\mathsf{s}\ell$-structure in which all the muxes are *lifts*. The definition of unshared $\mathsf{s}\ell$-structure boxes is similar to the one for standard proof net boxes, with the main difference that here they may contain lift links also. Therefore, let U be an unshared $\mathsf{s}\ell$-structure; a *box* of U is a subset of the links in U s.t.:

(i) the corresponding subnet B is connected (this is consistent with the fact that weakening formulas are connected to axioms);
(ii) B has no premises, *i.e.*, B is an unshared $\mathsf{s}\ell$-structure apart for the fact that the level of its conclusions is not necessarily equal to 0;
(iii) there is a conclusion of B, its *principal port*, that is the conclusion of an !-link (the principal link of B);
(iv) all the other conclusions of B, its *auxiliary ports*, are conclusions of a ?-link, of a gc-link or (when U is a parsing structure) of a net-link (named the auxiliary links of B).

A *box assignment* is a map that associates a box to each !-link (or better, to each conclusion of an !-link) with the constraint that two boxes never overlap in a non-trivial way. Namely, if B_1 and B_2 are two boxes, then $\mathsf{B}_1 \cap \mathsf{B}_2 \neq \emptyset$ implies that $\mathsf{B}_1 \subset \mathsf{B}_2$ or $\mathsf{B}_2 \subset \mathsf{B}_1$.

Definition 11 *A* proof $\mathsf{u}\ell$-structure *is an unshared* $\mathsf{s}\ell$*-structure* U *plus a box assignment. Moreover, the level of all the conclusions of* U *is equal to* 0.

The main purpose for the introduction of $\mathsf{u}\ell$-structures is for defining an unshared version of the β-rule, named β_{u}-rule (the corresponding rewriting will be denoted by $\mathsf{U} \rightharpoonup_{\mathsf{u}} \mathsf{U}'$), in which duplication is separated from reindexing, see Figure 19. In the β_{u}-rule, boxes are globally duplicated, while reindexing is delegated to the lifts introduced in the contractum. We stress that there is no constraint of the names of such lifts (*e.g.*, in Figure 19, we do not ask $a_1 \neq a_2$).

Garbage collection, parsing and π-reduction extend in a natural way to $\mathsf{u}\ell$-structures (even if, the unshared π-rules correspond to the particular case in which all the muxes in Figure 6, 7, 8 and 13 are lifts). The only relevant novelty is that the unshared version of such reductions must take into account the presence of boxes. In particular, we have that:

(i) Lift propagations does not change the border of a box; in other words, when $\mathsf{U} \rightarrow_{\pi} \mathsf{U}'$ because of a lift propagating through a link at the border of a box B, the image of B in U' is a box B' with the same principal and auxiliary links.
(ii) During garbage collection or parsing, we cannot contract a redex s.t. there is a box splitting it in two disjoint (non-empty) parts (*e.g.*, a gc-inside a

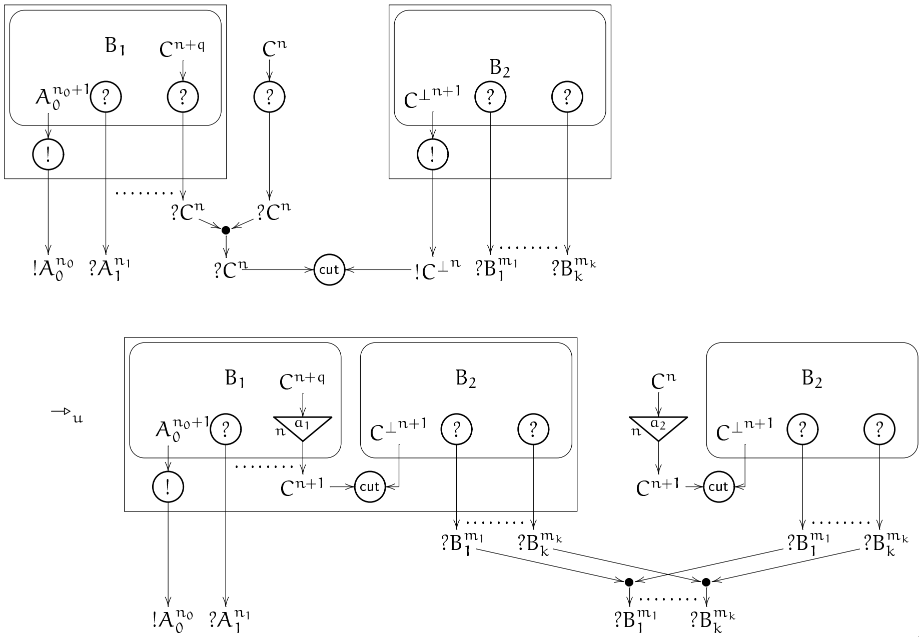


Fig. 19. Unshared β-rule

box B and some ?-link outside B). Indeed, we remark that in a structure 'properly' indexed, these situations are automatically forbidden.

(iii) A box disappear when its principal port has been collected or parsed.

We define unshared reduction as the union of β_u, π and $\sigma_\varnothing$ and we will denote by $U \to_u U'$ an unshared rewriting (note that both U and U' are $u\ell$-structures).

A proof ℓ-net is a particular case of $u\ell$-structure. According to this, we will define proof $u\ell$-nets as the result of unshared reductions starting from a proof ℓ-net.

Definition 12 ($u\ell$-net) *A $u\ell$-net U is a $u\ell$-structure for which there exists a proof ℓ-net N s.t. $N \to_u^* U$.*

The key issue on $u\ell$-nets is that the $\pi + \sigma_\varnothing$-normal form of a $u\ell$-net is an ℓ-net. But, in order to be able to prove this fact (subsection 5.4.5), we need to study in details the π-reduction of $u\ell$-structures.

A first trivial observation is that the π-normal form of a deadlock free $u\ell$-structure, if any, is lift free.

Fact 13 *Any deadlock free $u\ell$-structure U in π normal form is lift free.*

Proof. Let us start remarking that, since a $u\ell$-structure is connected, no $u\ell$-structure can contain a 'vicious cycle' as the one in Figure 20.

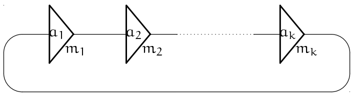


Fig. 20. Vicious cycle.

Then, let U be a deadlock free $u\ell$-structure that contains a lift. By the previous argument on vicious cycles, there is at least a lift u whose principal port is not connected to the auxiliary port of another lift. Since all the conclusions of U are at level 0 and the threshold of any lift is strictly lower than the level of its principal port, we see that the principal port of u cannot be a conclusion of U. Therefore, since U is deadlock free, the lift u is part of a redex, that is, U is not in π normal form. $\square$

5.4.2 *Lift permutation equivalence*

For the sake of the forthcoming proofs, let us introduce the lift permutation in Figure 21 (in this figure and in some of the following we shall omit the names of the formulas between the links; in such figures, we shall rather draw formula levels only).

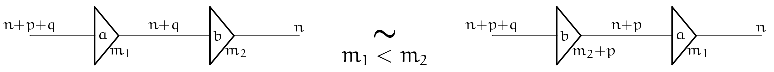


Fig. 21. Mux permutation rule.

The permutation in Figure 21 defines the equivalence $\sim$ on $u\ell$-structures.

The first remark is that lift permutation is sound w.r.t. π-reduction, at least in the relevant case in which deadlock freeness holds. In fact, we have the following two lemmas:

(i) permuting lifts does not block the computation, Lemma 14;
(ii) π-reduction is locally confluent modulo $\sim$, Lemma 15 (that Lemma is the key step for proving confluence of π).

Let $\mathsf{DF}_\pi(G)$ denotes that G is deadlock free.

Lemma 14 *Let* U_1, U_2 *be two* $u\ell$*-structures s.t.* $U_1 \sim U_2$*. If* $U_1 \to_\pi U_1''$ *and* $\mathsf{DF}_\pi(U_1)$*, then there exist* $U_1'' \to_\pi^* U_1'$ *and* $U_2 \to_\pi^* U_2'$ *s.t.* $U_1' \sim U_2'$*.*

Proof. By definition of $\sim$, there exists a $k \geq 0$ and two sets of sequences of lifts $T_1', \dots, T_k' \subset U_1$ and $T_1'', \dots, T_k'' \subset U_2$ s.t.: (*i*) $T_i' \cap T_j' \neq \emptyset$ or $T_i'' \cap T_j'' \neq \emptyset$ implies $i = j$; (*ii*) $T_i' \sim T_i''$, for $i = 1, \dots, k$; (*iii*) $U_1 \backslash (\bigcup_{i=1}^k T_i') = U_2 \backslash (\bigcup_{i=1}^k T_i'')$. We distinguish two cases:

(i) $\mathsf{U}_1 \to_\pi \mathsf{U}_1''$ is a lift interaction – W.l.o.g., we assume that the two lifts in the redex are the heads of the sequences T_1' and T_2' and that at least one of them has been permuted in T_1'' or in T_2'', otherwise there is nothing to prove. According to this, the redex reduced in U_1' has no correspondence in U_2.
First of all, let us note that each sequence of lifts is a sort of compound link whose conclusion (premise) corresponds to the head of the sequence, and whose premise (conclusion) corresponds to the tail the sequence. Moreover, since $\mathsf{DF}_\pi(\mathsf{U}_1)$, we see by induction on the length of T_1' and T_2', that there is a reduction sequence simulating a swap rule of such compound links. Such a rule generalizes, for the case of sequences of lifts, the bottom rule in Figure 7. In fact, referring to the l.h.s., the mux (respectively demux) with threshold m_1 (respectively m_2) is a sequence of positive (respectively negative) lifts; therefore, since $\mathsf{h} = \mathsf{k} = 1$, on the r.h.s. there are a sequence of negative lifts S_2' on the top of a sequence of positive lifts S_1'. The new sequences S_1' and S_2' are the residual of T_2' and T_1' after the compound swap; they are essentially the same of T_1' and T_2' apart for some reindexing of the thresholds and for some pairs of lifts that annihilated during the reduction. (As a matter of fact we have annihilation as a particular case, when S_1' and S_2' are empty. This corresponds to the case in which T_1' and T_2' are equal but for the orientation of their links, that are opposite.)
By induction on the size of T_1' and T_2' (that coincides with the size of T_1'' and T_2''), we see that the pair T_1'' and T_2'' reduces in a similar way and that, after that compound reduction, we get two sequences S_2'' and S_1'' s.t. $\mathsf{S}_2'' \sim \mathsf{S}_2'$ and $\mathsf{S}_1'' \sim \mathsf{S}_1'$.
Summing up, after the reduction $\mathsf{U}_1 \to_\pi \mathsf{U}_1''$ we can complete the swap of the sequences T_1' and T_2', obtaining $\mathsf{U}_1'' \to_\pi^* \mathsf{U}_1'$, and execute the swap of T_1'' and T_2'' on U_2, obtaining $\mathsf{U}_2'' \to_\pi^* \mathsf{U}_2'$. By the previous argument on the compound reduction of the sequences T_1' and T_2' (T_1'' and T_2''), we conclude that $\mathsf{U}_1' \sim \mathsf{U}_2'$.

(ii) $\mathsf{U}_1 \to_\pi \mathsf{U}_1''$ is a mux propagation or absorption. Let us assume that the sequences T_1' and T_1'' only are involved in the reduction. In this case also, we can define a generalized mux propagation/absorption rule and conclude by means of an argument similar to the one used above.

□

Lemma 15 *Let* $\mathsf{U}_1 \sim \mathsf{U}_2$ *be two* $\mathsf{u}\ell$*-structures s.t.* $\mathsf{DF}_\pi(\mathsf{U}_i)$*, for* $i = 1, 2$*. If* $\mathsf{U}_1 \to_\pi \mathsf{U}_1''$ *and* $\mathsf{U}_2 \to_\pi \mathsf{U}_2''$*, then there exist* $\mathsf{U}_1'' \to_\pi^* \mathsf{U}_1'$ *and* $\mathsf{U}_2'' \to_\pi^* \mathsf{U}_2'$ *s.t.* $\mathsf{U}_1' \sim \mathsf{U}_2'$*.*

Proof. Let us assume that T_i' and T_j'' denote the same sequences of lifts introduced in the proof of Lemma 14. We see that the two reductions $\mathsf{U}_1'' \to_\pi^*$

U_1' and $U_2'' \to_\pi^* U_2'$ can be easily found when the redexes reduced by $U_1 \to_\pi U_1''$ and $U_2 \to_\pi U_2''$ do not overlap, or more precisely, when they do not form a critical pair in terms of the sequences T_i' and T_j'' in which they are involved.
To simplify the analysis of critical pairs, let us restrict to the case in which the sequences T_i' and T_j'' are formed by a single lift. In practice, and w.l.o.g., this amounts to take $U = U_1 = U_2$.
A first case of critical pair is given in Figure 22, in which $U \to_\pi U_1''$ moves the lift with the port a, while $U \to_\pi U_2''$ moves the lift with the port b. Since $\mathsf{DF}_\pi(U)$, U_1'' and U_2'' cannot contain a deadlock, that is, $a = b$. As a consequence, U_1'' and U_2'' differ for two pairs of complementary lifts in different positions. By means of the annihilation of that pairs of lifts, we get local confluence, *i.e.*, $U_1'' \to_\pi U'$ and $U_2'' \to_\pi U'$.

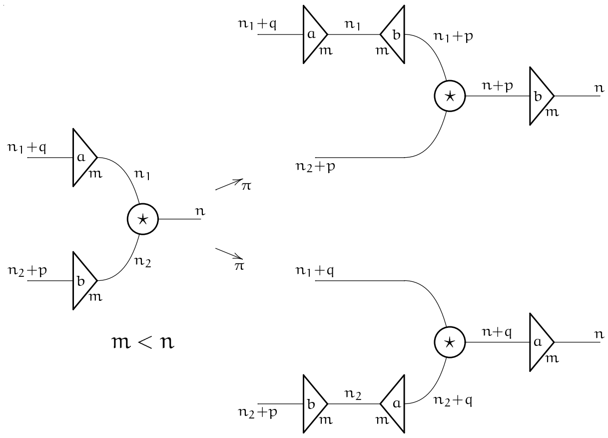


Fig. 22. Critical pair: two lifts whose thresholds are equal.

A second case of critical pair is the one in Figure 23. The main difference w.r.t. the first case is that the thresholds of the two lifts are different. Assuming as above that $U \to_\pi U_1''$ moves the lift with the port a, while $U \to_\pi U_2''$ moves the lift with the port b, we see that these reductions can be continued as in Figure 23, *i.e.*, $U_1'' \to_\pi U_1'$ and $U_2'' \to_\pi U_2'$, where the $u\ell$-structures U_1' and U_2' at the end of the reductions differ. By definition of lift permutation, we see that $U_1' \sim U_2'$.
A similar reasoning applies to the case in which one or both the lifts are absorbed.
By means of an easy iteration of the above arguments, the result generalizes to the case in which the two lifts are replaced by sequences of lifts. □

Remark 16 The proof of Lemma 15 shows the reason for the introduction of the lift permutation equivalence.

Our goal is to prove confluence of the π-rules, at least when the π-rules are strong normalizing—which is indeed the relevant case (see Lemma 25). But, because of the critical pair in Figure 23, we are not yet able to strengthen the result of Lemma 15 obtaining 'real' local confluence, *i.e.*, that $U_1 = U_2$

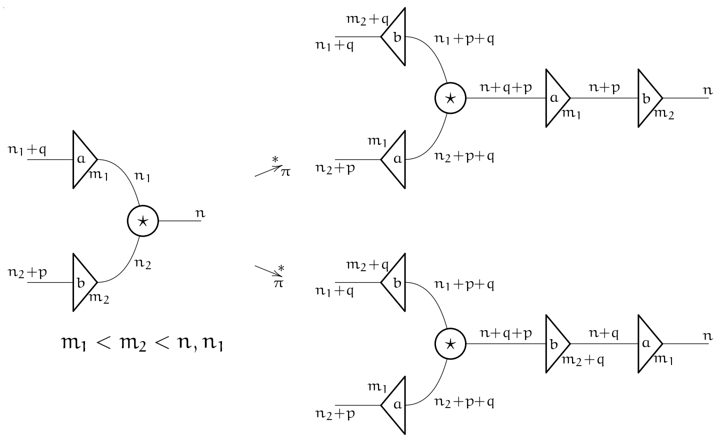


Fig. 23. Critical pairs: two lifts whose thresholds differ.

implies $U_1' = U_2'$ also. Instead, exploiting the result of Lemma 15, we must content ourselves with confluence modulo permutation of lifts.

Let $\mathsf{SN}_\alpha(G)$ denotes that G is strongly normalizing w.r.t. a set α of rewriting rules.

Lemma 17 *Let* $U_1 \sim U_2$ *be two* $u\ell$*-structures s.t.* $\mathsf{DF}_\pi(U_i)$ *and* $\mathsf{SN}_\pi(U_i)$, *for* $i = 1, 2$. *If* $U_1 \to^*_\pi U_1''$ *and* $U_2 \to^*_\pi U_2''$, *then there are* $U_1'' \to^*_\pi U_1'$ *and* $U_2'' \to^*_\pi U_2'$ *s.t.* $U_1' \sim U_2'$.

Proof. This is essentially the Newman's Lemma for $u\ell$-structures equated modulo $\sim$. As in the proof of Newman's Lemma, we take the maximal length $\#(U)$ of a reduction $U \to^*_\pi U''$ and we exploit that $\#(U_1), \#(U_2) < \omega$. More precisely, the proof is by induction on $\#(U_1, U_2) = \max(\#(U_1), \#(U_2))$.
In the base case, U_1 and U_2 are in π-normal form. Therefore, $U_1 = U_2$ (by Fact 13) and confluence holds trivially.
In the induction case, let us assume that, for $i = 1, 2$, $U_i \to^1_\pi V_i' \to^*_\pi U_i''$, where $\to^1_\pi$ denotes a reduction whose length is at most one (see Figure 24 for a graphic representation of the reductions in the proof), and $V_i' \neq U_i$ for at least one value of i. Moreover, we shall assume w.l.o.g. that $V_1' \neq U_1$ and that $V_2' = U_2$ only if $U_2 = U_2''$.
By Lemma 14 (when $V_2' = U_2$) or Lemma 15 (when $V_2' \neq U_2$), we see that there exists $V_1' \to^*_\pi V_1$ and $V_2' \to^*_\pi V_2$ s.t. $V_1 \sim V_2$. Hence, $V_i' \to^*_\pi U_i''$ and $V_i' \to^*_\pi V_i$, for $i = 1, 2$. Now, let us remark that:
(*i*) Since $\#(V_1') < \#(U_1, U_2)$, there are $U_1'' \to^*_\pi W_1''$ and $V_1 \to^*_\pi W_1'$ s.t. $W_1' \sim W_1''$ (by induction hypothesis).
(*ii*) When $V_2' \neq U_2$, we have $\#(V_2') < \#(U_1, U_2)$, then the induction hypothesis applies as for V_1', otherwise, $U_2 = V_2' = U_2''$. In both the cases, there are $U_2'' \to^*_\pi W_2''$ and $V_2 \to^*_\pi W_2'$ s.t. $W_2' \sim W_2''$ (in particular, when $U_2'' = U_2$, we have $W_2' = W_2'' = V_2$).
Thus, applying the induction hypothesis to $V_1' \to^*_\pi W_1'$ and $V_2' \to^*_\pi W_2'$ (we

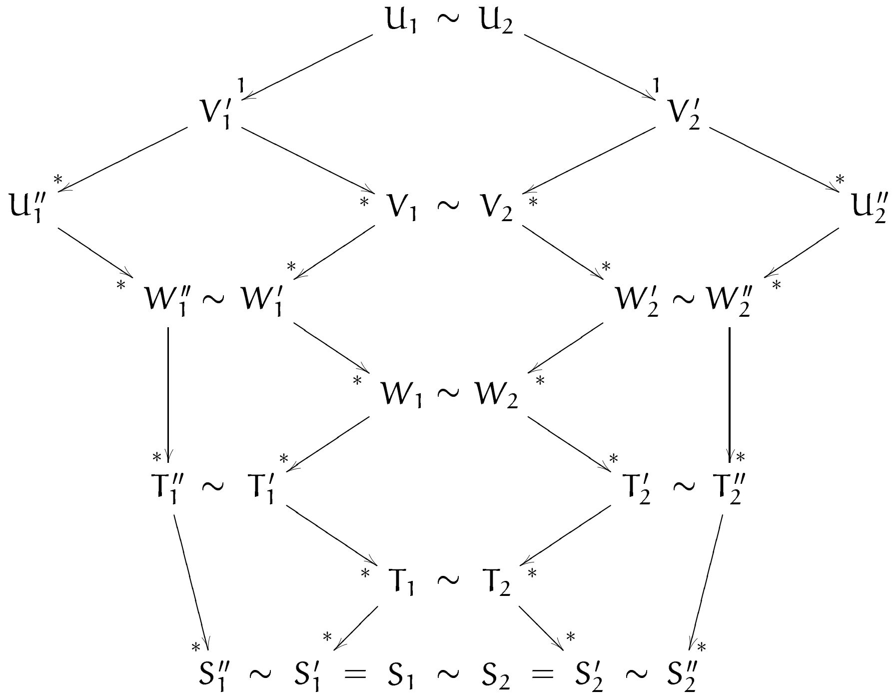


Fig. 24. Confluence modulo $\sim$.

remark that $\#(V_1, V_2) < \#(U_1, U_2)$), there are $W_1' \to_\pi^* W_1$ and $W_2' \to_\pi^* W_2$ s.t. $W_1 \sim W_2$.
Summing up, in order to conclude the proof, we are left to show that:
When, for $i = 1, 2$, there are $U_i \to_\pi^* W_i' \to_\pi^* W_i$ and $U_i'' \to_\pi^* W_i''$ s.t. $W_i' \sim W_i''$; then, there are $W_1' \to_\pi^* U_1'$ and $W_2' \to_\pi^* U_2'$ s.t. $U_1' \sim U_2'$ also.
By Lemma 14, there are $W_i \to_\pi^* T_i'$ and $W_i'' \to_\pi^* T_i''$ s.t. $T_i' \sim T_i''$. Moreover, since $\#(W_1, W_2) \leq \#(V_1, V_2) < \#(U_1, U_2)$, there are $T_1' \to_\pi^* T_1$ and $T_2' \to_\pi^* T_2$ s.t. $T_1 \sim T_2$ (by induction hypothesis).
We have got a new pair $T_1 \sim T_2$ with the same characteristics of $W_1 \sim W_2$, and for which $\#(T_1, T_2) < \#(U_1, U_2)$, when $T_1 \neq T_1'$ or $T_2 \neq T_2'$.
The above construction can then be iterated in the following way (we assume to start from $W_1 \sim W_2$):
(*i*) If $W_1 = W_1'$ and $W_2 = W_2'$, then stop.
(*ii*) Otherwise (in this case $\#(W_1, W_2) < \#(U_1, U_2)$), let us take, for $i = 1, 2$, T_i, T_i', T_i'' as above. Then, let us repeat the construction after replacing T_1 for W_1 and T_2 for W_2.
Since $\#(W_1, W_2) > \#(T_1, T_2) > 0$, we eventually end up with a pair $S_1 \sim S_2$, s.t. $S_i = S_i' \sim S_i''$ and $U_i'' \to_\pi^* S_i''$, for $i = 1, 2$.
Therefore, taking $U_1' = S_1''$ and $U_2' = S_2''$, we conclude. $\square$

For the relevant case in which $U_1 = U_2 = U$, the hypotheses on deadlock freeness imply 'real' confluence indeed. As a consequence, the π normal form of U is unique.

Let $\mathcal{R}_\pi(\mathsf{G})$ denotes the π normal form of G (if any).

Lemma 18 *Let* $\mathfrak{U}$ *be a* $\mathfrak{u}\ell$*-structure. If* $\mathsf{DF}_\pi(\mathfrak{U})$ *and* $\mathsf{SN}_\pi(\mathfrak{U})$*, then* $\mathcal{R}_\pi(\mathfrak{U})$ *exists unique and is lift free.*

Proof. Let $\mathcal{R}_\pi(\mathfrak{U})$ be a π normal form of $\mathfrak{U}$. By Lemma 17, $\mathcal{R}_\pi(\mathfrak{U}) \sim \mathsf{N}$, for any other π normal form N of $\mathfrak{U}$. But, since $\mathcal{R}_\pi(\mathfrak{U})$ is lift free (by Fact 13), we have $\mathcal{R}_\pi(\mathfrak{U}) = \mathsf{N}$. $\square$

Strong normalization and deadlock freeness are the key assumptions for Lemma 18. However, strong normalization is indeed the only issue that we need—and for the $\mathfrak{u}\ell$-structures in which we are interested, it will be ensured by Lemma 25. In fact, we can start observing that, provided SN_π, deadlock freeness is invariant under π-reduction and under lift permutation.

Lemma 19 *Let* $\mathfrak{U}$ *be a* $\mathfrak{u}\ell$*-structures s.t.* $\mathsf{SN}_\pi(\mathfrak{U})$*.*

(i) If $\mathfrak{U} \to_\pi \mathfrak{U}'$*, then* $\mathsf{DF}_\pi(\mathfrak{U})$ *iff* $\mathsf{DF}_\pi(\mathfrak{U}')$*.*
(ii) If $\mathfrak{U} \sim \mathfrak{U}'$ *and* $\mathsf{SN}_\pi(\mathfrak{U}')$*, then* $\mathsf{DF}_\pi(\mathfrak{U})$ *implies* $\mathsf{DF}_\pi(\mathfrak{U}')$*.*

Proof.

(i) The direction $\mathsf{DF}_\pi(\mathfrak{U})$ implies $\mathsf{DF}_\pi(\mathfrak{U}')$ is trivial. Therefore, let us assume $\neg\mathsf{DF}_\pi(\mathfrak{U})$. There is $\mathfrak{U} \to_\pi^* \mathfrak{U}_\mathrm{d}$ s.t. $\mathfrak{U}_\mathrm{d}$ contains a deadlock. Since a deadlock is not removed by any π-reduction and $\mathsf{SN}_\pi(\mathfrak{U})$ (by hypothesis), we choose $\mathfrak{U}_\mathrm{d}$ s.t. it is in normal form. By Lemma 17, $\mathfrak{U}_\mathrm{d} \sim \mathcal{R}_\pi(\mathfrak{U}')$. That is, any normal form of $\mathfrak{U}'$ contains at least a lift. Thus, by Fact 13, we conclude that $\neg\mathsf{DF}_\pi(\mathfrak{U})$ implies $\neg\mathsf{DF}_\pi(\mathfrak{U}')$.

(ii) Let us assume $\neg\mathsf{DF}(\mathfrak{U})$. Any π-normal form N of $\mathfrak{U}$ contains a deadlock. By Lemma 24, we see that $\mathsf{N} \sim \mathsf{N}'$ for any π-normal form N' of $\mathfrak{U}'$. As a consequence, also N' contains a deadlock.

$\square$

Then, in order to ensure deadlock freeness of a $\mathfrak{u}\ell$-structure that strongly normalizes under π-reduction, it suffices to find a π-reduction ending with a lift free structure.

Corollary 20 *Let* $\mathfrak{U}$ *be a* $\mathfrak{u}\ell$*-structure s.t.* $\mathsf{SN}_\pi(\mathfrak{U})$*. If* $\mathcal{R}_\pi(\mathfrak{U})$ *is lift free, then* $\mathsf{DF}_\pi(\mathfrak{U})$*. Moreover,* $\mathcal{R}_\pi(\mathfrak{U})$ *is the unique* π *normal form of* $\mathfrak{U}$*.*

Proof. Since $\mathcal{R}_\pi(\mathfrak{U})$ is lift free, $\mathsf{DF}_\pi(\mathcal{R}_\pi(\mathfrak{U}))$ holds trivially. But, since $\mathfrak{U} \to_\pi^* \mathcal{R}_\pi(\mathfrak{U})$, we have $\mathsf{DF}_\pi(\mathfrak{U})$ (by Lemma 19). The uniqueness of $\mathcal{R}_\pi(\mathfrak{U})$ is a consequence of Lemma 18. $\square$

5.4.3 Correctness

We are not interested in any generic $\mathfrak{u}\ell$-structure. On the contrary, we focus on 'correct' $\mathfrak{u}\ell$-structures, that is, on those $\mathfrak{u}\ell$-structures that can be seen as the 'unfolding' of an $s\ell$-structure arising along the reduction of a proof ℓ-net.

Generalizing the notion of parsing introduced in Section 5.2, we give a notion of *correctness.*

Definition 21 (correctness) *The $\mathfrak{u}\ell$-structure $\mathfrak{U}$ is* correct*, written* $\mathsf{CORR}(\mathfrak{U})$*, when* $\mathfrak{U} \Rightarrow_\bullet^* \mathsf{net}_\Gamma$.

Such a definition of correctness is justified by the following standardization lemma.

Lemma 22 *For each reduction sequence $\rho : \mathfrak{U} \Rightarrow_\bullet^* \mathfrak{U}'$, there is a corresponding standard reduction sequence $\rho' : \mathfrak{U} \to_\pi^* \mathfrak{U}_1 \to_{\sigma_\varnothing}^* \mathfrak{U}_2 \to_{\sigma_\bullet}^* \mathfrak{U}'$. Moreover, when ρ does not contain any $\sigma_\bullet$ rule, then $\rho' : \mathfrak{U} \to_\pi^* \mathfrak{U}_1 \to_{\sigma_\varnothing}^* \mathfrak{U}'$.*

Proof. Let us start observing that σ and π commutes according to the following scheme: if $\mathfrak{U} \to_\sigma \mathfrak{U}_1 \to_\pi \mathfrak{U}_0$, then there exist $\mathfrak{U}_2$ s.t. $\mathfrak{U} \to_\pi^* \mathfrak{U}_2 \to_\sigma \mathfrak{U}_0$. (Since the proof of the previous observation proceed by a simple case analysis, we shall limit to draw one of that cases, see Figure 25.)
Now, ρ can be decomposed in a unique way in three reductions $\rho_0 : \mathfrak{U} \Rightarrow_\bullet^* \mathfrak{U}_1$, $\rho_1 : \mathfrak{U}_1 \to_\pi^* \mathfrak{U}_2$ and $\rho_2 : \mathfrak{U}_2 \to_\sigma^* \mathfrak{U}_3$ s.t. ρ_1 and ρ_2 are of maximal length. Let us define $\#(\rho) = \langle s, |\rho_1| \rangle$, where s is the number of σ reductions in ρ_0. We prove by induction on $\#(\rho)$ that there is $\rho'' : \mathfrak{U} \to_\pi^* \mathfrak{U}_1 \to_\sigma^* \mathfrak{U}'$. In the base case, *i.e.*, when ρ_0 is empty, there is nothing to prove, for the reduction has the required shape. When ρ_0 is not empty, we see that the last reduction of ρ_0 must be a σ-rule and that ρ_1 must be non empty too. Permuting the last rule of ρ_0 with the first rule of ρ_1 we get then a new reduction ρ'' s.t. $\#(\rho'') < \#(\rho)$. Thus, by induction hypothesis, we conclude.
To conclude the proof, we remark that any reduction $\mathfrak{U}_1 \to_\sigma^* \mathfrak{U}'$ can be decomposed in $\mathfrak{U}_1 \to_{\sigma_\varnothing}^* \mathfrak{U}_2 \to_{\sigma_\bullet}^* \mathfrak{U}'$ (by the fact that σ is trivially confluent). $\square$

In particular, we have the following corollary.

Corollary 23 $\mathsf{CORR}(\mathfrak{U})$ *implies* $\mathfrak{U} \to_\pi^* \mathcal{R}_\pi(\mathfrak{U}) \to_{\sigma_\varnothing}^* \mathcal{R}(\mathfrak{U}) \to_{\sigma_\bullet}^* \mathsf{net}_\Gamma$.

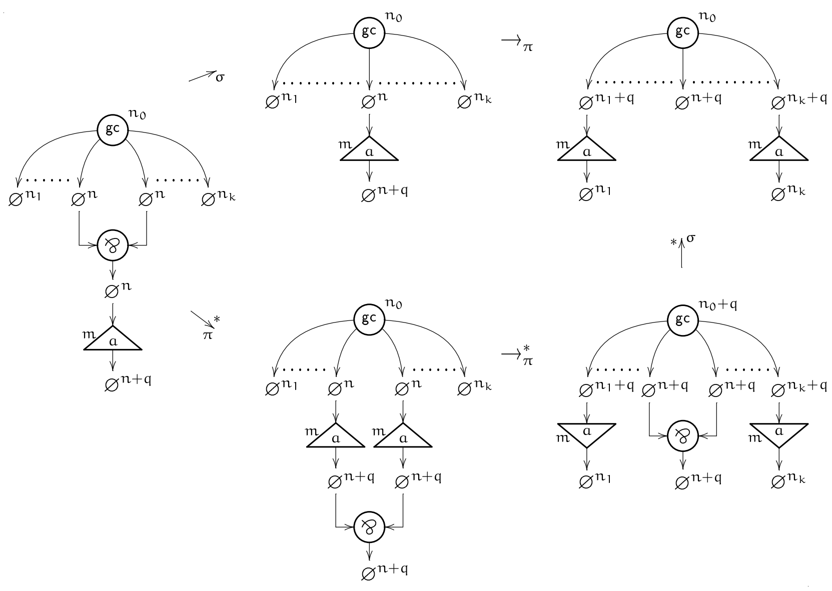


Fig. 25. π–σ permutation.

Corollary 23 can be summarized by the following considerations.

For any correct $\mathfrak{u}\ell$-structure U, we have that:

(i) The read-back of U is a proof ℓ-net.
(ii) In order to prove the uniqueness of the read-back of U, we need to show that $\mathsf{CORR}(U)$ implies $\mathsf{DF}_\pi(U)$ and $\mathsf{SN}_\pi(U)$; but, because of Corollary 20, this amounts to prove that $\mathsf{CORR}(U)$ implies $\mathsf{SN}_\pi(U)$ only.
(iii) The computation of the read-back U can be decomposed in two phases: first, the propagation of the lifts; second, the collection of the garbage created during the first phase. However, this way to proceed is not compulsory. In fact, we shall see that any $\pi + \sigma_\varnothing$ reduction will eventually end with the same ℓ-net $\mathcal{R}(U)$, independently from the order in which π and $\sigma_\varnothing$ are applied.

5.4.4 Strong normalization of π-reduction

As already done for deadlock freeness, the key issue is that the existence of a lift free π normal form implies the strong normalization of π-reduction. Before proving this result, we need to introduce some structures.

Let $\mathcal{W}_0$ be the set of pairs $\langle m, q\rangle \in \mathbb{Z}^2$ s.t. $m \geq 0$ and $q \geq -1$. A *weight* is a class of strings over $\mathcal{W}_0$ that are equivalent modulo the congruence induced by the equation:

$$\langle m_2, q_2\rangle \langle m_1, q_1\rangle \approx \langle m_1, q_1\rangle \langle m_2 + q_1, q_2\rangle \qquad \text{when } m_1 < m_2$$

i.e., $[\alpha] = \{\alpha' \mid \alpha' \approx \alpha\}$, for $\alpha \in \mathcal{W}_0^*$. Using the notations $1 = [\epsilon] = \{\epsilon\}$ and

$\langle m, q\rangle = [\langle m, q\rangle] = \{\langle m, q\rangle\}$, the monoid of weights $\mathcal{W} = \mathcal{W}_0^*/\approx$ is the monoid generated by the pairs in $\mathcal{W}_0$ by means of the composition rule

$$\langle m_1, q_1\rangle \circ \langle m_2, q_2\rangle \circ \cdots \circ \langle m_k, q_k\rangle = [\langle m_1, q_1\rangle\, \langle m_2, q_2\rangle\, \cdots\, \langle m_k, q_k\rangle]$$

That is, $\alpha_1 \circ \alpha_2 = [s_1\, s_2]$, where $s_i \in \alpha_i$, for $i = 1, 2$.

Since $\approx$ equates strings with the same length, we see that the *length of a weight* is well-defined and that $|\alpha| = |\langle m_1, q_1\rangle \circ \cdots \circ \langle m_k, q_k\rangle| = k$, for $\langle m_1, q_1\rangle \circ \cdots \circ \langle m_k, q_k\rangle \in \alpha$.

The set $\mathcal{W}[n, m]$ of the weights ranging from n to m, provided that $0 \leq n \leq m$, is defined by

$$\begin{aligned}\mathcal{W}[n, m] = \{\langle m_1, q_1\rangle \circ \cdots \circ \langle m_k, q_k\rangle \mid \\ n \leq m_i < m + q_1 + \cdots + q_i, \text{ for } i = 1, \dots, k\}\end{aligned}$$

In particular, let us note that $W[0, 0] = \{1\}$.

Given a parsing $u\ell$-structure U (*i.e.*, U may contain net links too). A $\mathcal{W}$-*decoration* of U is a weight assignment that associates to each formula X^n of U an element of $\mathcal{W}[0, n]$ s.t., for each link ν and each pair of indexed formulas $X_1^{n_1}$ and $X_2^{n_2}$ that are a premise or a conclusion of ν, the weights α_1 and α_2 respectively assigned to $X_1^{n_1}$ and $X_2^{n_2}$ verify the following constraints:

(i) If ν is an axiom, par, tensor, contraction or of-course link, then $\alpha_1 = \alpha_2$.
(ii) If ν is a why-not, a net, or a garbage collection link, and $n_1 \leq n_2$, then $\alpha_2 = \alpha_1 \cdot \alpha_0$, with $\alpha_0 \in \mathcal{W}[0, n_1]$ and $\alpha_1 \in \mathcal{W}[n_1, n_2]$,
(iii) If ν is a positive (negative) lift with threshold m and offset q, and $X_2^{n_2}$ is its conclusion (premise), then $\alpha_2 = \langle m, q\rangle \circ \alpha_1$.

Let $\mathcal{D}$ be a $\mathcal{W}$-decoration for U, the weight of U w.r.t. $\mathcal{D}$ is the pair $\|U\|_{\mathcal{D}} = \langle h_0, h_1\rangle$ defined by:

(i) $h_0 = \Sigma_\nu|\alpha_\nu|$, for ν ranging over the set of the links that are not lifts, and where α_ν is the weight assigned by $\mathcal{D}$ to the premise/conclusion of ν whose level is maximum;
(ii) $h_1 = \Sigma_\nu|\alpha_\nu|$, for ν ranging over the set of the lifts in U, and where α_ν is the weight assigned by $\mathcal{D}$ to the formula at the principal port of ν (the conclusion of ν when ν is a positive lift, the premise when ν is a negative lift).

Using the lexicographic order, we shall say that $\|U\|_{\mathcal{D}} = \langle h_0, h_1\rangle \leq \langle h_0, h_1\rangle = \|U'\|_{\mathcal{D}'}$, when $h_0 < h_0'$, or $h_0 = h_0'$ and $h_1 < h_1'$.

The previous machinery is justified by the following fact.

Fact 24 *Let* $\mathcal{U}$ *be a* $\mathfrak{u}\ell$*-structure. For any* $\mathcal{U} \Rightarrow_{\bullet} \mathcal{U}'$*, we have that:*

(i) $\mathcal{U}$ *has a* $\mathcal{W}$*-decoration iff* $\mathcal{U}'$ *has a* $\mathcal{W}$*-decoration.*
(ii) If $\mathcal{D}$ *is a* $\mathcal{W}$*-decoration of* G*, then there exists a* $\mathcal{W}$*-decoration of* G' *such that* $\|\mathcal{U}'\|_{\mathcal{D}'} < \|\mathcal{U}\|_{\mathcal{D}}$*.*

Proof. By case analysis. □

From which follows immediately the main result concerning read-back and correctness.

Lemma 25 *Let* $\mathcal{U}$ *be a* $\mathfrak{u}\ell$*-structure.* $\mathsf{CORR}(\mathcal{U})$ *implies:*

(i) $\mathsf{SN}_{\pi+\sigma}(\mathcal{U})$ *and therefore* $\mathsf{DF}_{\pi}(\mathcal{U})$ *also.*
(ii) $\mathcal{R}_{\pi}(\mathcal{U})$ *and* $\mathcal{R}(\mathcal{U})$ *exist unique and are lift free.*

Proof. By hypothesis, $\mathcal{U} \Rightarrow_{\bullet}^{*} \mathsf{net}_{\Gamma}$. Therefore, since net_{Γ} has a trivial $\mathcal{W}$-decoration, we can conclude that $\mathcal{U}$ has a $\mathcal{W}$-decoration (by the first item of Fact 24). By the second item of Fact 24, $\mathcal{U}$ has no infinite $\pi + \sigma$-reduction. By definition, $\mathcal{R}_{\pi}(\mathcal{U})$ is lift free. Thus, by Corollary 20, we conclude $\mathsf{DF}_{\pi}(\mathcal{U})$. By Corollary 20, we have that $\mathcal{R}_{\pi}(\mathcal{U})$ is the unique π-normal form of $\mathcal{U}$. The uniqueness of $\mathcal{R}(\mathcal{U})$ follows then from Lemma 22 and from the fact that $\sigma_{\varnothing}$ is (trivially) confluent. □

We also remark that correctness is invariant under lift permutation.

Fact 26 *Let* $\mathcal{U}_1 \sim \mathcal{U}_2$ *be two* $\mathfrak{u}\ell$*-structures.* $\mathsf{CORR}(\mathcal{U}_1)$ *implies* $\mathsf{CORR}(\mathcal{U}_2)$*.*

Proof. Let us proceed by induction on the length $\#(\mathcal{U}_1)$ of the maximal π-reduction of $\mathcal{U}_1$. In the base case, $\mathcal{U}_1$ is lift free. Thus, in this case there is nothing to prove, for $\mathcal{U}_1 = \mathcal{U}_2$. When $\mathcal{U}_1 \to_{\pi} \mathcal{U}_1''$. By Lemma 14, there are $\mathcal{U}_1'' \to_{\pi}^{*} \mathcal{U}_1'$ and $\mathcal{U}_2 \to_{\pi}^{*} \mathcal{U}_2'$ s.t. $\mathcal{U}_1' \sim \mathcal{U}_2'$. Since $\#(\mathcal{U}_1') < \#(\mathcal{U}_1)$, we conclude $\mathsf{CORR}(\mathcal{U}_2')$ and $\mathsf{CORR}(\mathcal{U}_2)$ by induction hypothesis. □

5.4.5 Correctness and unshared nets

We aim at proving that any $\mathfrak{u}\ell$-net is correct, see Lemma 30 (we remind that a $\mathfrak{u}\ell$-net is the result of an unshared reduction starting with a proof ℓ-net). By the way, for we have already seen that correctness is invariant under $\pi + \sigma_{\varnothing}$, we are left to prove that $\beta_{\mathfrak{u}}$ preserves correctness (see Lemma 29).

Let U be a $\mathsf{u}\ell$-structure s.t. $\mathsf{U} \twoheadrightarrow_{\mathsf{u}} \mathsf{V}$. We want to reduce the problem of proving $\mathsf{CORR}(\mathsf{U})$ implies $\mathsf{CORR}(\mathsf{V})$ to the case in which U is lift free, *i.e.*, it is a proof ℓ-net. Because of this, we need to analyze how and when we can permute β_{u} and π-reduction.

The key idea is to treat the links in the β_{u}-redex under analysis as a rigid configuration that can be removed by the execution of the corresponding β_{u}-reduction only.

For instance, let us assume that one of the ?-links in the β_{u}-redex of U might absorb a lift above it. Executing such an absorption first, we get $\mathsf{U} \rightarrow_{\pi} \mathsf{U}' \twoheadrightarrow_{\mathsf{u}} \mathsf{U}''$. After the β_{u}-reduction, we get instead $\mathsf{U} \twoheadrightarrow_{\mathsf{u}} \mathsf{V}$; where V and U'' differ substantially, for a pair of lifts in V corresponds to a unique lift in U''. As a matter of fact, there is no chance to find a naive permutation of a pair of redexes like that.

Another case that deserves some care is when a π-propagation $\mathsf{U} \rightarrow_{\pi}^{*} \mathsf{U}'$ hides the β_{u}-redex of U. This situation arises when a lift above a ?-link in the β_{u}-redex propagates through the ?-link and stops before completing the propagation through the whole β_{u}-redex. However, in this case, it is not difficult to see that it suffices to consider U' as a non-observable situation, assuming instead that the reduction must continue with $\mathsf{U}' \rightarrow_{\pi}^{*} \mathsf{U}''$, until reaching a U'' in which the β_{u}-redex is visible again.

The latter situation is indeed the key case of critical pair between a π and β_{u}-rule. In Figure 26 (to simplify the picture we have omitted the formulas between the links), we have drawn a case in which a pair of contracted ?-links in cut with the principal door of a box B is below a pair of equal lifts. In accord with the previous considerations, we see that $\mathsf{U} \rightarrow_{\pi}^{*} \mathsf{U}_1 \twoheadrightarrow_{\mathsf{u}} \mathsf{V}_1$ and that $\mathsf{U} \rightarrow_{\mathsf{u}} \mathsf{V} \sim \mathsf{V}' \rightarrow_{\pi}^{*} \mathsf{V}_1$. Moreover, after the propagation of the two lifts with ports $\mathfrak{b}_1$ and $\mathfrak{b}_2$, we have $\mathsf{V}_1 \rightarrow_{\pi}^{*} W_1$ and $\mathsf{V} \rightarrow_{\pi}^{*} W_2$ with $W_1 \sim W_2$ (W_2 is obtained from W_1 by permuting the two lifts with port $\mathfrak{a}$ and the lifts with port $\mathfrak{b}_1$ and $\mathfrak{b}_2$).

To complete the analysis of this kind of critical pair, we have to explain why we can assume w.l.o.g. that above the two ?-links in the β_{u}-redex there are two lifts with the same port, threshold and offset. The first step is to observe that, restricting the application of the π-rules to a substructure S of a correct $\mathsf{u}\ell$-structure, we can move all the lifts to the border of S.

Fact 27 *Let S be a sub-structure of a correct $\mathsf{u}\ell$-structure U. There is a π-*

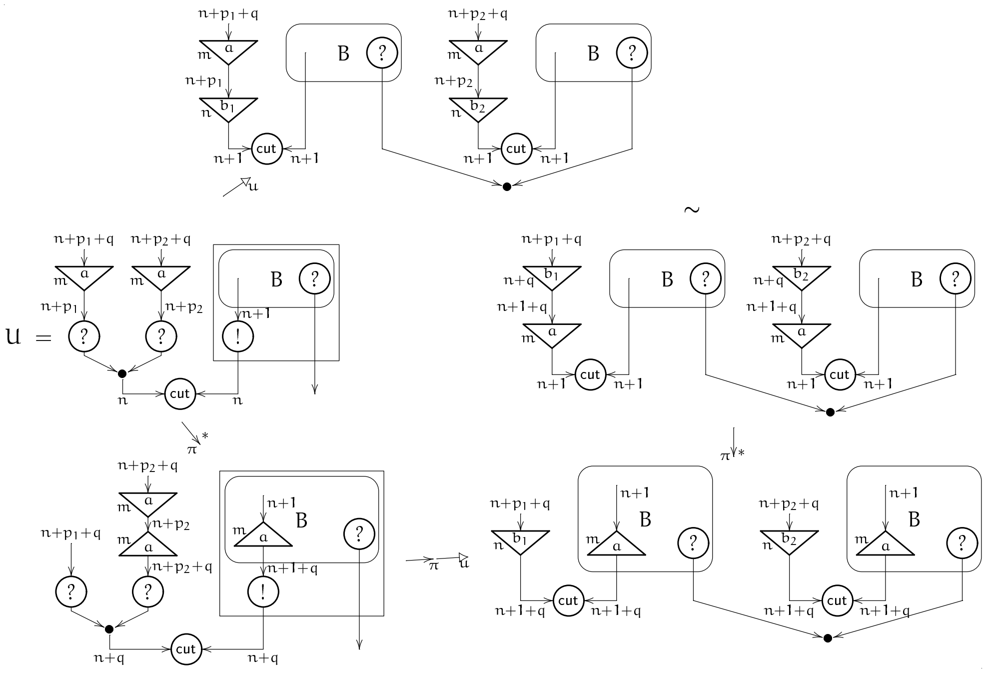
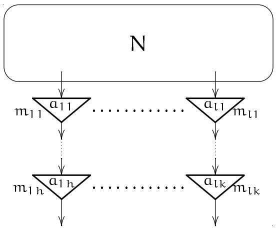


Fig. 26. π–β_{u} critical pair.

reduction $\mathsf{S} \to_{\pi}^{*} \mathsf{S}'$ *s.t.* S' *has the following shape:*



where N *is a lift free* ℓ*-structure. Moreover, if* S'' *is another* π*-normal form of* S*, then* $\mathsf{S}' \sim \mathsf{S}''$*.*

Proof. The only difference w.r.t. Fact 13 is that the conclusions of S may have a level greater than 0. If this is the case, a normal form S' of S may contain sequences of positive lifts pointing to its conclusions (note that this does not contradict deadlock freeness).
Now, let us remark that, in the proof of Lemma 24, we have never used the fact that all the conclusions of a $\mathsf{u}\ell$-structure have a level equal to 0. Therefore, Lemma 24 can be extended to the case in which $\mathsf{U}_1 \sim \mathsf{U}_2$ are sub-structures of two $\mathsf{u}\ell$-structures V_1 and V_2 s.t. $\mathsf{CORR}(\mathsf{V}_1)$ and $\mathsf{CORR}(\mathsf{V}_2)$. In particular, taking $\mathsf{S} = \mathsf{U}_1 = \mathsf{U}_2$ and $\mathsf{V}_1 = \mathsf{V}_2 = \mathsf{U}$, we see that, for any $\mathsf{S} \to_{\pi}^{*} \mathsf{S}''$, there is $\mathsf{S}'' \to_{\pi}^{*} \mathsf{T}'$, with $\mathsf{T}' \sim \mathsf{S}'$ (since S' is in π-normal form). Hence, when S'' also is in π-normal form, we must have $\mathsf{S}' \sim \mathsf{S}''$. $\square$

Now, let us see what happens when the substructure S of U is a box and two of its auxiliary ports are contracted by a link c. In the resulting sub-structure S', the box surrounding S is transformed into a box surrounding the lift free structure N. Therefore, $N \Rightarrow^* \mathsf{net}_\Gamma$ (we remind that any box of a proof net is a proof net at its turn). According to this, above the two premises of c that were connected to two ports of S, there are now two sequences of lifts T_1 and T_2 equivalent by lift permutation, otherwise, we would have an infinite $\pi + \sigma$ reduction of U. Namely, after reducing N to a net link u we could propagate T_1 and T_2; if $T_1 \not\sim T_2$, T_1 (respectively T_2) would be replaced by a sequence of negative lifts T_2' (respectively T_2) pointing towards u and s.t. $T_1' \not\sim T_2'$; again, after the propagation of T_1' and T_2' we would obtain a sequence of positive lifts T_1'' in the place of T_1' and a sequence of positive lifts T_2'' in the place of T_2' s.t. $T_1'' \not\sim T_2''$; and so on forever.

The previous argument, plus the fact that analyzing π-reduction we can work modulo lift permutation, explains why the shape chosen for the β_u-redex in Figure 26 is w.l.o.g. in the case in which the two ?-links in the $u\ell$-structure U are in the same box. The case in which such two links belong to two distinct boxes B_1 and B_2 is similar. Also in this case, there is a sub-structure S containing both B_1 and B_2 and s.t.: *(i)* $S \to_\pi^* S'$ as in Fact 27; *(ii)* $N \Rightarrow^* \mathsf{net}_\Gamma$ (where N is the lift free part of S'); *(iii)* $S \cap B = \emptyset$; *(iv)* if u is the contraction link contracting the two auxiliary ports of B_1 and B_2, then $u \notin S$. The analysis proceed then as in the case in which $B_1 = B_2$, there is a $\pi + \sigma$ reduction of U contracting S to S' and then the lift free part of S' to a net-link, and so on.

What showed for the case in Figure 26 can be extended to the following general rule of permutation.

Lemma 28 *Let* U_0 *be a correct* $u\ell$*-structure s.t.* $r_0 : U_0 \twoheadrightarrow_u V_0$. *Let*

$$U_0 \to_\pi^* U_1' \sim U_1 \to_\pi^* U_2 \to_\pi^* \cdots U_k' \sim U_k \to_\pi^* U_{k+1}' \sim U_{k+1}$$

be a reduction s.t.:

(i) $\mathsf{SN}_\pi(U_i)$, *for any* $i = 1, 2, \ldots, k+1$*;*
(ii) no residual of a ?-link in the β_u*-redex* r_0 *is involved in a lift absorption;*
(iii) there is a β_u*-redex* $r_i : U_i \to_u V_i$ *that is the residual of the initial redex* r_0*, for* $i = 1, \ldots, k+1$.

Then, for any pair of indexes $i, j \leq k+1$*, there is a pair of* $u\ell$*-structures* $W_{ij} \sim W_{ji}$ *s.t.* $V_i \to_\pi^* W_{ij}$.

Proof. Let us start showing that, when the property hold for two pairs (i, j) and (j, l), then it holds for (i, l) also. By hypothesis, there are $W_{ij} \sim W_{ji}$ and $W_{jl} \sim W_{lj}$ s.t. $V_i \to_\pi^* W_{ij}$, $V_l \to_\pi^* W_{lj}$ and $V_j \to_\pi^* W_{ji}, W_{jl}$. Let N_x be the

π-normal forms of W_x. By Lemma 17, we have that $N_{ij} \sim N_{ji} \sim N_{jl} \sim N_{lj}$. Therefore, there are $V_i \to_\pi^* W_{ij} \to_\pi^* W_{il}$ and $V_l \to_\pi^* W_{lj} \to_\pi^* W_{li}$ s.t. $W_{il} \sim W_{li}$ (*e.g.*, we could take $W_{il} = N_{ij}$ and $W_{li} = N_{lj}$).
Hence, we suffice to prove the case $k = 1$ only. The proof proceed by induction on the length of the reduction $\rho : U_0 \to_\pi^* U_1'$ and by case analysis w.r.t. the last π-redex p in that reduction (*i.e.*, $\rho = \rho' p$).
When the ancestor in U_0 of p does not form a critical pair with r_0. We easily see that the redex r_1 has an image r_i'' in the the $u\ell$-structure U_1'' s.t. $\rho' : U_0 \to_\pi^* U_1''$. Moreover, $U_1'' \twoheadrightarrow_u V_1'' \to_\pi V_1' \sim V_1$, where $r_1' : U_1' \twoheadrightarrow_u V_1'$ is the image of r_1 in U_1'. By the induction hypothesis on the length of ρ, we have that V_0 and V_1'' can be reduced to two $u\ell$-structures that are equivalent modulo lift permutation. By confluence modulo $\sim$, the same is true for V_1'' and V_1, and then for V_0 and V_1 also.
The case in which p completes a π-propagation through the redex r_0, we can assume w.l.o.g. that there is a sequence of reductions τ at the end of ρ, *i.e.*, $\rho = \rho''\tau$ with $\rho'' : U \to_\pi^* U_1''$, s.t. τ moves a lift from the premise of a ?-link of the residual r_0'' of r_0 in U_1'' to the premise of the !-link of r_0''. By the analysis of the example in Figure 26, also in this case we have that $U_1'' \twoheadrightarrow_u V_1'' \to_\pi^* V_1'$ s.t. $V_1' \to_\pi^* W_1'$ and $V_1 \to_\pi^* W_1$, with $W_1' \sim W_1$. As above, by the induction hypothesis on the length of ρ, V_0 and V_1'' can be reduced to two $u\ell$-structures that are equivalent modulo lift permutation and, by confluence modulo $\sim$, the same is true for V_1'' and V_1, and then for V_0 and V_1.
The analysis of the other critical pairs between a π and a β_u-rule is similar. $\square$

The proof of Lemma 28 is the keystone for proving that the β_u-rule preserves correctness.

Lemma 29 *Let* U *be a* $u\ell$*-structure s.t.* $U \twoheadrightarrow_u V$. $\mathsf{CORR}(U)$ *implies* $\mathsf{CORR}(V)$.

Proof. We start remarking that $\mathsf{SN}_\pi(U)$ implies $\mathsf{SN}_\pi(V)$. In fact, any $\mathcal{W}$-decoration of U induces a $\mathcal{W}$-decoration of V (the decoration of V is obtained by assigning to each formula of V the weight of its ancestor in U). Therefore, by Fact 24, we conclude that there is no infinite π-reduction of V.
We show then that $\mathsf{DF}_\pi(U)$ implies $\mathsf{DF}_\pi(V)$, by reducing such a proof to the analysis of the lift free case.
The crucial observation is that there is a sequence of reductions as the ones in Lemma 28 s.t. the last $u\ell$-structure $\widehat{U}$ obtained in this way cannot be reduced any farther using that technique. Namely, for any $U' \sim \widehat{U}$, $p : U' \to_\pi U''$ implies that p is an absorption involving a ?-link in the residual $\widehat{r}$ of r. By Lemma 28, $\widehat{r} : \widehat{U} \twoheadrightarrow_u \widehat{V} \to_\pi^* \widehat{W}$ and $V \to_\pi^* W$, for some $W \sim \widehat{W}$. Therefore, by Lemma 19, $\mathsf{DF}_\pi(\widehat{V})$ implies $\mathsf{DF}_\pi(V)$.
Now, let us analyze in more details the shape of $\widehat{U}$. By construction, we see that $\widehat{U} \to_\pi^* \mathcal{R}_\pi(U)$ by a sequence of absorptions at the ?-links in the redex $\widehat{r}$.

Therefore, the only lifts in $\widehat{\mathsf{r}}$ are arranged in sequences above the ?-links in $\widehat{\mathsf{r}}$ that, no matter how we try to permute them, can move from that position by means of an absorption only. After the reduction of $\widehat{\mathsf{r}}$, each sequence of lifts S in $\widehat{\mathsf{U}}$ give raise in $\widehat{\mathsf{V}}$ to a sequence of lifts T whose head is a lift introduced while reducing $\widehat{\mathsf{r}}$; moreover, such a lift is the head of any sequence $\mathsf{T}' \sim \mathsf{T}$. Because of the previous considerations and by inspection of the π-rules, we remark the following points:
(*i*) Any sequence T is a sort of compound link s.t., whenever the head of T can propagate through a link or is absorbed by a port, also the rest of the chain propagates through such a link or is absorbed by such a port;
(*ii*) Let us assume that we choose the names of the lifts introduced by $\widehat{\mathsf{r}}$ s.t. they are all distinct and that we reduce $\widehat{\mathsf{V}}$ in accord treating the sequences T as a rigid structure. We see that, along the reduction, two facing sequences T' and T'' that are not deadlocked are residual of the same initial sequence T. Thus, they annihilate.
(*iii*) Let W be the $\mathsf{u}\ell$-structure s.t. $\mathcal{R}_\pi(\widehat{\mathsf{U}}) = \mathcal{R}_\pi(\mathsf{U}) \rightarrowtriangle_{\mathsf{u}} W$, by the contraction of the same cut in $\widehat{\mathsf{r}}$. Each sequence of lifts T in $\widehat{\mathsf{V}}$ correspond to a unique lift in W. Because of this, we see that, for any π-reduction of $\widehat{\mathsf{V}}$ there is an isomorphic π-reduction of W. Thus, $\mathsf{DF}_\pi(\widehat{\mathsf{V}})$ iff $\mathsf{DF}_\pi(W)$.
Summing up, we left to prove that $\mathcal{R}_\pi(\mathsf{U}) \rightarrowtriangle_{\mathsf{u}} W$ implies $\mathsf{CORR}(W)$. For the sake of simplicity we will refer to the case depicted in Figure 27.

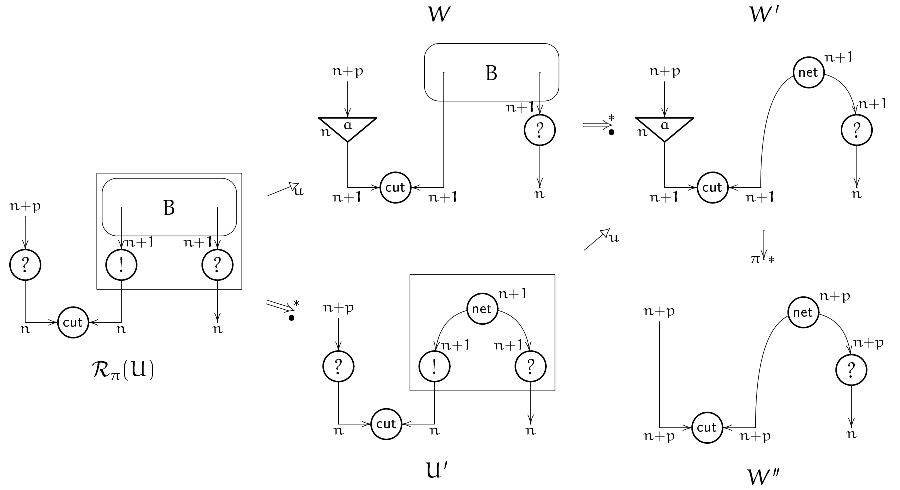


Fig. 27. β_{u}-rule: the lift free case.

Let us take $\mathcal{R}_\pi(\mathsf{U})$ as in Figure 27. By $\mathsf{CORR}(\mathcal{R}_\pi(\mathsf{U}))$, we see that there is a parsing reduction contracting the interior of the box B to a net-link (U'). Then, by means of a β_{u}-rule (W') followed by the a sequence of π-propagations, we reach a parsing $\mathsf{u}\ell$-structure (W'') that is clearly correct (in Figure 27, we have drawn the relevant parts of the structures involved in the reduction only; anyhow, we can easily see that, whichever is the shape of $\mathcal{R}_\pi(\mathsf{U})$, $\mathsf{CORR}(\mathcal{R}_\pi(\mathsf{U}))$ implies $\mathsf{CORR}(W'')$). Summing up, we have $\mathcal{R}_\pi(\mathsf{U}) \to^*_\bullet \mathsf{U}' \rightarrowtriangle_{\mathsf{u}} W' \to^*_\pi W'' \to^*_\bullet \mathsf{net}_\Gamma$. Now, since $\mathcal{R}_\pi(\mathsf{U}) \rightarrowtriangle_{\mathsf{u}} W \to^*_\bullet W'$ also, we conclude

$\mathsf{CORR}(W)$ by Lemma 22. The extension to the general case is straightforward. $\square$

Finally, we can conclude that any $\mathfrak{u}\ell$-net is correct.

Lemma 30 *Let* U *be a* $\mathfrak{u}\ell$*-structure and* N *be a proof* ℓ*-net. If* $\mathsf{N} \to_{\mathfrak{u}}^{*} \mathsf{U}$*, that is,* U *is a* $\mathfrak{u}\ell$*-net, then* $\mathsf{CORR}(\mathsf{U})$*.*

Proof. Correctness is trivially preserved by π and $\sigma_{\varnothing}$. Therefore, by $\mathsf{CORR}(\mathsf{N})$ and by Lemma 29, we conclude. $\square$

5.4.6 Dead and alive parts of a $\mathfrak{u}\ell$*-net*

By inspection of the proof of Lemma 29, we see that $\beta_{\mathfrak{u}}$-reduction behaves as expected on $\mathfrak{u}\ell$-nets, at least in the case in which the $\mathfrak{u}\ell$-net is garbage free. Namely, let us assume that the $\mathfrak{u}\ell$-net U does not contain any garbage,*i.e.*, $\mathcal{R}_{\pi}(\mathsf{U}) = \mathcal{R}(\mathsf{U})$, and that $\mathsf{U} \twoheadrightarrow_{\mathfrak{u}} \mathsf{V}$ by a reduction that does not involve any weakening. Then $\mathcal{R}(\mathsf{U}) \twoheadrightarrow_{s} \mathcal{R}(\mathsf{V})$.

The situation is no longer as direct when the $\mathfrak{u}\ell$-net U contains a garbage collection link, a dead axiom or some lift with a port of kind garbage. In fact, in U might contain some $\beta_{\mathfrak{u}}$-redex that, although they have not been erased yet, should be considered as dead because belonging to a box that has been cut against a weakening. Therefore, we need a more formal definition of what we mean by dead part of a $\mathfrak{u}\ell$-net.

Given a $\mathfrak{u}\ell$-net U there is a natural embedding of $\mathcal{R}_{\pi}(\mathsf{U})$ into U associating each link of $\mathcal{R}_{\pi}(\mathsf{U})$ to a corresponding link of U, s.t. any pair of links in $\mathcal{R}_{\pi}(\mathsf{U})$ connected by a premise/conclusion A corresponds to a pair of links in U connected by a path crossing lifts only. By inspection of the $\sigma_{\varnothing}$ rules, we see that $\mathcal{R}(\mathsf{U})$ is a subnet of $\mathcal{R}_{\pi}(\mathsf{U})$. The sub-structure $\mathsf{U}^{(a)}$ of U associated to $\mathcal{R}(\mathsf{G})$ via the previous embedding (*i.e.*, it is the smallest $s\ell$-structure $\mathsf{U}^{(a)} \subset \mathsf{U}$ s.t. the image of any link in $\mathcal{R}(\mathsf{U})$ is contained in $\mathsf{U}^{(a)}$) is the *alive part* of U. Its complement $\mathsf{U}^{(d)}$ is instead the *dead part* of U. Correspondingly, we shall say *dead* any reduction of U internal to $\mathsf{U}^{(d)}$, and *alive* any reduction of U internal to $\mathsf{U}^{(a)}$.

The relevant point about the dead reductions of a $\mathfrak{u}\ell$-net U is that they do not change the read-back of U. While, alive reductions change the read-back of U in accord with the standard proof net reduction.

Fact 31 *Let* U *be a* $\mathfrak{u}\ell$*-net s.t.* $\mathsf{r} : \mathsf{U} \twoheadrightarrow_{\mathfrak{u}} \mathsf{V}$*.*

(i) If r *is in the alive part* $\mathsf{U}^{(a)}$ *of* U*, then* $\mathcal{R}(\mathsf{U}) \rightarrowtriangle_s \mathcal{R}(\mathsf{V})$*.*
(ii) If r *is in the dead part* $\mathsf{U}^{(d)}$ *of* U*, then* $\mathcal{R}(\mathsf{U}) = \mathcal{R}(\mathsf{V})$*.*

Proof. By inspection of the proof of Lemma 29 and by the permutation properties between π and β_u-rule (Lemma 28). In particular, referring to the lift free case to which the proof of Lemma 29 has been reduced, we see that:

(i) When r is in $\mathcal{R}_\pi(\mathsf{U})^{(d)}$, the part modified by the reduction of r is in $\mathcal{R}_\pi(\mathsf{V})^{(d)}$. Thus, in this case, $\mathcal{R}(\mathsf{U}) = \mathcal{R}_\pi(\mathsf{U})^{(a)} = \mathcal{R}_\pi(\mathsf{V})^{(a)} = \mathcal{R}(\mathsf{V})$.
(ii) When r is in $\mathcal{R}_\pi(\mathsf{U})^{(a)} = \mathcal{R}(\mathsf{U})$, we have instead $\mathcal{R}(\mathsf{U}) = \mathcal{R}_\pi(\mathsf{U})^{(a)} \rightarrowtriangle_s \mathcal{R}_\pi(\mathsf{V})^{(a)} = \mathcal{R}(\mathsf{V})$.

□

5.4.7 Main results

We have now all the technical preliminaries required for stating the properties of unshared reductions. These properties will be extended to $s\ell$-structures in the next section.

Theorem 32 (existence and uniqueness of readback) *For any proof* $u\ell$*-net* U*.*

(i) The $\pi + \sigma_\varnothing$ *rules are strongly normalizing and confluent on* U*.*
(ii) $\mathcal{R}(\mathsf{U})$ *is unique and is a proof* ℓ*-net.*

Proof. By Lemma 30, we have $\mathsf{CORR}(\mathsf{U})$. Therefore, by Lemma 25, we have $\mathsf{SN}_{\pi+\sigma_\varnothing}(\mathsf{U})$ and that $\mathcal{R}(\mathsf{U})$ is unique and lift free. Moreover, since $\mathsf{CORR}(\mathsf{U})$ implies $\mathsf{CORR}(\mathcal{R}(\mathsf{U}))$, we see that $\mathcal{R}(\mathsf{U}) \Rightarrow^* \mathsf{net}_\Gamma$. That is, $\mathcal{R}(\mathsf{U})$ is a proof ℓ-net. □

Theorem 33 (coherence) *Let* U *be a proof* $u\ell$*-net. Then*

$$\begin{tikzcd}
\mathsf{N} \arrow[r, "*"', "u"] \arrow[dr, dotted, "{s\,*}"'] & \mathsf{U} \arrow[r, "u"] \arrow[d, Rightarrow, "*"] & \mathsf{V} \arrow[d, Rightarrow, "*"] \\
& \mathcal{R}(\mathsf{U}) \arrow[r, dotted, "{*\,s}"] & \mathcal{R}(\mathsf{V})
\end{tikzcd}$$

where N *is any proof* ℓ*-net s.t.* $\mathsf{N} \to_u^* \mathsf{U}$*. Moreover,* $\mathcal{R}(\mathsf{U}) \rightarrowtriangle_s \mathcal{R}(\mathsf{V})$*, when* $\mathsf{r} : \mathsf{U} \rightarrowtriangle_u \mathsf{V}$ *and* r *is alive, while* $\mathcal{R}(\mathsf{U}) = \mathcal{R}(\mathsf{V})$*, in all the other cases.*

Proof. Let us start proving that $\mathsf{r} : \mathsf{U} \to_u \mathsf{V}$ implies $\mathcal{R}(\mathsf{U}) \rightarrowtriangle_s^* \mathcal{R}(\mathsf{V})$. When, r is a π, $\sigma_\varnothing$, or a dead β_u-reduction (see Fact 31), we have $\mathcal{R}(\mathsf{U}) = \mathcal{R}(\mathsf{V})$. When

r is an alive β_u-reduction, we have instead $\mathcal{R}(U) \twoheadrightarrow_s \mathcal{R}(V)$ (by Fact 31). Then, by induction on the length of $\rho : N \to_u^* U$, we conclude. □

Theorem 34 (standard strategy) *Any standard reduction of a proof ℓ-net N can be simulated by a β_u-reduction followed by a normalizing read-back reduction. That is,*

$$\begin{tikzcd} & U \arrow[d, Rightarrow, "*"] \\ N \arrow[ur, dotted, "u"] \arrow[r, "s"'] & \mathcal{R}(U) \end{tikzcd}$$

Proof. It is an immediate corollary of Theorem 33. □

In absence of dead reductions, Theorem 33 would immediately imply strong normalization of unshared reduction (for β_s-reduction of proof ℓ-nets is strong normalizing). This is indeed what we have already exploited in the case without weakening, see [GMM96]. Here, we also need to prove that we cannot go on forever reducing inside a dead part.

Theorem 35 (strong normalization) *There is no infinite unshared reduction of a proof $u\ell$-net.*

Proof. By the definition of $u\ell$-net, it suffices to prove that there is no infinite unshared reduction of any proof ℓ-net N.
Let $\rho : U_0 \to_u U_1 \to_u \cdots \to_u U_k$ be an unshared reduction of the ℓ-net $N = U_0$. W.l.o.g. we can assume that ρ does not contain $\sigma_\varnothing$-rules. In fact, given a reduction $\rho_\varnothing$ that contains $\sigma_\varnothing$-rules, we can transform $\rho_\varnothing$ into another reduction ρ in which we never perform garbage collection, while each propagation through a gc-link is replaced by a suitable sequence of π-rules (through the garbage substructure collected by that gc-link). Moreover, any reduction containing an infinite number of $\sigma_\varnothing$-rules must also contain an infinite number of π and β_u-rules.
We define now a modified version of standard and unshared β, let us call them $\beta_s^{\not w}$ and $\beta_u^{\not w}$. The idea is that such rules does not erase cuts involving a weakening link. In this way, the reduced reduction never contains garbage, for we never introduce lifts whose port is of kind garbage.
Namely, the standard definition given in Figure 4 can be seen as the simultaneous execution of three steps:
(i) the creation of a new instance of the box B_2 (the one whose principal port is in r) for each ?-link above the contraction in the redex r;

(*ii*) the reindexing of the instances of B_2 corresponding to ?-links of type dereliction (because of a cut between such a dereliction link and the !-link at the principal port of the corresponding instance of B_2);
(*iii*) the erasing of the instances of B_2 corresponding to each ?-link of type weakening (because of a cut between such a weakening link and the !-link at the principal port of the corresponding instance of B_2).
Then, given a standard reduction $r : N \rightarrowtriangle_s M$, we define the $\beta_s^{\not{w}}$ reduction $r : N \overset{\not{w}}{\rightarrowtriangle}_s M'$ of the proof ℓ-net N as the result of the above procedure in which the third step has not been executed. Such a reduction corresponds to a legal reduction according to the usual notion of MELL cut-elimination. (The postposition of weakening cuts to the end of the reduction is a standard proof technique that allows to get rid of weakenings, *e.g.*,, see [Reg92].)
A similar reasoning applies to the β_u-rule. Also in this case, the rule decomposes in three steps, but, while the first one is as above, the other two are:
(*ii*) the substitution of each ?-link of type dereliction by a corresponding lift whose port is of kind identity;
(*iii*) the substitution of each ?-link of type weakening by a corresponding lift of kind garbage (*i.e.*, whose port is of kind garbage).
Also in this case, given an unshared reduction $r : U \rightarrowtriangle_s V$, we define the $\beta_u^{\not{w}}$ reduction $r : U \overset{\not{w}}{\rightarrowtriangle}_u V'$ of the $u\ell$-net U as the result of the execution of the first two steps given for β_u. In this way, no lift of kind garbage is ever introduced along the reduction and no dead part is ever created, *i.e.*, if U is a $u\ell$-net obtained by an unshared reduction in which we have used $\beta_u^{\not{w}}$ instead of β_u, then $\mathcal{R}_\pi(U) = \mathcal{R}(U)$.
We can now remark that, the reduction ρ given at the beginning of the proof induces a corresponding reduction $\rho^{\not{w}}$ in which any β_u is replaced by a corresponding $\beta_u^{\not{w}}$, while any propagation of a lift of kind garbage is simply removed. In fact, for the premise of a weakening is always a formula of type $\varnothing$, the execution of a cut involving a weakening, and the following insertion of a lift of kind garbage, cannot create new redexes; it may create new cuts whose formulas are both of type $\varnothing$, but such cuts are not redexes according to our definition of β-reduction.
Therefore, corresponding to ρ, we have a reduction

$$\rho^{\not{w}} : N = V_0 \overset{\not{w}}{\rightarrowtriangle}_u V_1 \to_\pi^* W_1 \overset{\not{w}}{\rightarrowtriangle}_u V_2 \to_\pi^* W_2 \cdots \overset{\not{w}}{\rightarrowtriangle}_u V_h \to_\pi^* W_h$$

with a number h of $\beta_u^{\not{w}}$-rules equal to the number of β_u-rules contained in ρ. By $\mathcal{R}_\pi(V_i) = \mathcal{R}(V_i)$, we see that $\mathcal{R}(V_i) \overset{\not{w}}{\rightarrowtriangle}_s \mathcal{R}(V_{i+1}) = \mathcal{R}(W_{i+1})$, for $i = 0, \ldots, h-1$. That is, we have a proof ℓ-net reduction

$$\rho_s^{\not{w}} : N \overset{\not{w}}{\rightarrowtriangle}_s \mathcal{R}(V_1) \overset{\not{w}}{\rightarrowtriangle}_s \mathcal{R}(V_2) \cdots \overset{\not{w}}{\rightarrowtriangle}_s \mathcal{R}(V_h)$$

Moreover, by the strong normalization property of MELL proof nets, there is a finite K (depending on U only) s.t. $h \leq K$.
Summing up, ρ cannot contain an infinite number of β_u-reduction. Moreover,

since $\mathsf{SN}_\pi(\mathsf{U}_i)$ for any i (by Theorem 32), ρ cannot contain an infinite sequence of π-rules. That is, there is no infinite unshared reduction of N. $\square$

Theorem 36 (confluence) *For any proof* $\mathsf{u}\ell$*-net* U *such that* $\mathsf{U} \to_u^* \mathsf{U}_1$ *and* $\mathsf{U} \to_u^* \mathsf{U}_2$*, there exists a* $\mathsf{u}\ell$*-net* U_0 *such that* $\mathsf{U}_1 \to_u^* \mathsf{U}_0$ *and* $\mathsf{U}_2 \to_u^* \mathsf{U}_0$*.*

Proof. Let us consider the following diagram,

$$\begin{tikzcd}
 & & \mathsf{N} \arrow[d, dotted, "*"] \arrow[ddll, dotted, "*"'] \arrow[ddrr, dotted, "*"] & & \\
 & & \mathsf{U} \arrow[dl, "*"'] \arrow[dr, "*"] & & \\
\mathcal{R}(\mathsf{U}_1) \arrow[ddrr, dotted, "*"'] & \mathsf{U}_1 \arrow[l, Rightarrow, dotted, "*"'] \arrow[dr, dotted, "*"'] & & \mathsf{U}_2 \arrow[r, Rightarrow, dotted, "*"] \arrow[dl, dotted, "*"] & \mathcal{R}(\mathsf{U}_2) \arrow[ddll, dotted, "*"] \\
 & & \mathsf{G}_0 \arrow[d, dotted, "*"] & & \\
 & & \mathsf{N}_0 & &
\end{tikzcd}$$

in which N and N_0 are proof ℓ-nets. The inner diamond in the diagram corresponds to the statement of the theorem. The commutativity of the upper half of the diagram follows from Theorem 33. The commutativity of the external diamond is a consequence of confluence of β_s-reduction. To get the remaining arrows, let us note that, by Theorem 34, we have $\mathsf{U}_1 \to_u^* \mathcal{R}(\mathsf{U}_1) \to_u^* \mathsf{N}_0$ and $\mathsf{U}_2 \to_u^* \mathcal{R}(\mathsf{U}_2) \to_u^* \mathsf{N}_0$. So, $\mathsf{U}_0 = \mathsf{N}_0$ is a possible choice for completing the construction. $\square$

5.5 The shared case

First of all, we need the notion of sharing morphism.

Definition 37 (sharing morphism) *A sharing morphism* M *is a surjective graph homomorphism* $\mathsf{M} : \mathsf{U} \to \mathsf{G}$*, where* U *is a* $\mathsf{u}\ell$*-structure and* G *is an* $\mathsf{s}\ell$*-structure, that preserves the link and node labels (*i.e.*, the type of the links and the formula and level of the nodes) and the names of the link ports, and is injective when restricted to the conclusions of* G *(*i.e.*,* U *and* G *have the same conclusions).*

The key idea is to simulate unshared reductions by means of usual shared reductions.

Lemma 38 (simulation) *Let* U *be a* $u\ell$*-net and let* $M : U \to G$ *be a sharing morphism. For each reduction* $r : G \to G'$ *there exists a non empty unshared reduction sequence* ρ *and a sharing morphism* M' *s.t.*

$$\begin{tikzcd} G \arrow[r, "r"] & G' \\ U \arrow[u, "M"] \arrow[r, dotted, "+", "\rho"'] & U' \arrow[u, dotted, "M'"'] \end{tikzcd}$$

Proof. The counterimage $M^{-1}(r)$ in U of a redex r of G is a set of redexes that may contain only a case of critical pair: k lifts with the same threshold that points to the premises of the same k-ary contraction c. (We remind that contraction premise ports are the only ones that we cannot distinguish.)
Therefore, let us assume to have one of such a critical pair. Since U is a $u\ell$-net, we have $\mathsf{CORR}(U)$ and, therefore, $\mathsf{DF}_\pi(U)$ also. This implies, that the k lifts in the critical pair are equal (see Figure 22) and that, no matter which redex of the pair is executed first, after some reduction we reach a $u\ell$-net U' in which the lifts at the premises of c have been removed and replaced by a positive lift at the conclusion of c (by the way, U and U' differ for the level of the premises and of the conclusion of c also, since the propagation of the lifts has properly increased that level).
We also remark that, by $\mathsf{CORR}(U)$, U cannot contain pairs of deadlocked facing muxes. Therefore, when r is a mux annihilation, all the redexes in $M^{-1}(r)$ are mux annihilations also.
Hence, let us execute in any order the redexes of U in $M^{-1}(r)$. The $u\ell$-net U' is the one obtained after closing any critical pair in $M^{-1}(r)$, according to what previously said.
The sharing morphism between U' and G' maps any residual of a link ν of U into the residual of $M(\nu)$. □

Given the simulation lemma, the reformulation of the results in subsection 5.4.7 is an easy exercise. By the way, also in this case, the relevant structures in which we are interested are the ones arising along the reduction of a proof ℓ-net.

Definition 39 (proof $s\ell$**-net)** *An* $s\ell$*-structure* G *is a* proof $s\ell$-net *when* $N \to^* G$*, for some proof* ℓ*-net* N.

Theorem 40 (existence and uniqueness of readback) *For any proof* $s\ell$*-net* G.

(i) The $\pi + \sigma_\varnothing$ *rules are strongly normalizing and confluent on* G.
(ii) $\mathcal{R}(G)$ *is unique and is a proof* ℓ*-net.*

Theorem 41 (standard strategy) *Any standard reduction of a proof* ℓ*-net* N *can be simulated by a* β_u*-reduction followed by a normalizing read-back reduction. That is,*

$$\begin{tikzcd}
 & \mathsf{G} \arrow[d, Rightarrow, "*"] \\
\mathsf{N} \arrow[ur, -Triangle] \arrow[r, ->>] & \mathcal{R}(\mathsf{G})
\end{tikzcd}$$

Theorem 42 (coherence) *Let* G *be an* $s\ell$*-net. Then*

$$\begin{tikzcd}
\mathsf{N} \arrow[r, ->>, "*"] \arrow[dr, dotted, ->>, "*"'] & \mathsf{G} \arrow[r, -Triangle] \arrow[d, Rightarrow, "*"] & \mathsf{G}_1 \arrow[d, Rightarrow, "*"] \\
 & \mathcal{R}(\mathsf{G}) \arrow[r, dotted, ->>, "*"] & \mathcal{R}(\mathsf{G}_1)
\end{tikzcd}$$

where N *is any proof* ℓ*-net s.t.* $\mathsf{N} \rightarrow^*_u \mathsf{G}$.

Remark 43 The previous result differs from the analogous of [GMM96] in the fact that we cannot ensure that to a β_l contraction of G corresponds a non-empty β_s reduction of $\mathcal{R}(\mathsf{G})$. In fact, G may not be garbage free, since it may contain part of a box to be erased. Hence, the contraction of a cut in such a part has no correspondence in $\mathcal{R}(\mathsf{G})$. As a consequence, the proof of the strong normalization property becomes more involved.

Theorem 44 (strong normalization) *There is no infinite reduction of a proof* $s\ell$*-net.*

Theorem 45 (confluence) *For any proof* $s\ell$*-net* G *such that* $\mathsf{G} \rightarrow^* \mathsf{G}_0$ *and* $\mathsf{G} \rightarrow^* \mathsf{G}_1$*, there exists an* $s\ell$*-net* G_0 *such that* $\mathsf{G}_1 \rightarrow^* \mathsf{G}_0$ *and* $\mathsf{G}_2 \rightarrow^* \mathsf{G}_0$.

Corollary 46 (unique normal form) *The* $\beta_l + \pi + \sigma_\varnothing$ *normal form of a proof* ℓ*-net* N *is unique and coincides with its* β_s *normal form.*

6 Conclusions

We have presented a distributed and local graph-rewriting algorithm for MELL proof nets. The relations of this work with the ideas and techniques developed for the optimal reduction of interaction nets have been discussed in the Introduction. One might wonder why there is no statement on this subject in the paper. The crucial reason is that, in presence of weakening, the very notion of optimal reduction for proof nets is highly an open question.

It is clear that the mux propagation rule, when applied with a duplicating mux facing the premise of a logical node, would duplicate any redex in its scope, thus destroying any optimality. It is easy to split the propagation rule in two rules, one for when the mux faces a premise (call it the *dup* rule) and one for when the mux faces the conclusion (the *odup* rule). Let now π_{opt} be the subset of π including *odup* but not *dup* (except for axioms, cuts, and gc).

Theorem 47 *Let* G *be a proof* $s\ell$*-net and* N *be its* $\beta + \pi + \sigma$ *normal form. Let* G' *be a* $\beta + \pi_{opt} + \sigma$ *normal form of* G*, then* $\mathcal{R}(\mathsf{G}') = \mathsf{N}$*.*

Therefore, normalization of proof $s\ell$-nets may be performed in two distinct steps: first 'optimal' reduction ($\beta + \pi_{opt} + \sigma$), then read-back reduction. However, this theorem only warrants that no duplication of redexes is done, but nothing is said about the *need* of such redexes (that is, whether these redexes belong to a part of the net which will not be erased). We are missing, in other words, an effective strategy ensuring the reduction of only needed cuts (like the left-most-outermost in the case of λ-calculus). Let us note, incidentally, that even for sharing graphs for λ-calculus, the meaning of an optimal reduction in presence of weakening is still a non completely understood subject.

All the results of this paper hold for full MELL, *i.e.*, MELL with the two constant $\perp$ and $\mathsf{1}$. The basic ideas in order to treat $\perp$ and $\mathsf{1}$ are the following. The constant $\mathsf{1}$ is introduced by means of a new axiom link, treated as all the other axioms. The $\perp$ constant is treated like weakening formulas, namely: (1) there is a *bottom* link with a $\varnothing$ premise and the $\perp$ constant as conclusion; (2) the concept of box is extended in order to allow $\perp$ as secondary door (such an extension do not change the expressive power of the logic, [GM96]). All the results stated here extend rather simply to full MELL. As a treatment of constants should have produced an increase of notation without any gain from a computational point of view, we have chosen to drop $\perp, \mathsf{1}$.

It would be interesting to develop a semantics of the reductions of this papers, along the lines of the geometry of interaction [Gir89], or of the consistent path semantics for sharing graphs of [ADLR94]. For the weakening-free case, see [Gue96].

References

[ADLR94] Andrea Asperti, Vincent Danos, Cosimo Laneve, and Laurent Regnier. Paths in the lambda-calculus: three years of communications without understanding. In *Proceedings of 9th Annual Symposium on Logic in Computer Science*, pages 426–436, Paris, France, July 1994. IEEE.

[AG98] Andrea Asperti and Stefano Guerrini. *The Optimal Implementation of Functional Programming Languages*. Cambridge University Press, 1998.

[AL96] Andrea Asperti and Cosimo Laneve. Interaction system II: The practice of optimal reductions. *Theoretical Computer Science*, 159(2):191–244, 3 June 1996.

[Asp95] Andrea Asperti. Linear logic, comonads and optimal reductions. *Fundamenta infomaticae*, 22:3–22, 1995.

[Ban95] Richard Banach. Sequent reconstruction in LLM—A sweepline proof. *Annals of Pure and Applied Logic*, 73:277–295, 1995.

[GAL92] Georges Gonthier, Martín Abadi, and Jean-Jacques Lévy. Linear logic without boxes. In *Proceedings of 7th Annual Symposium on Logic in Computer Science*, pages 223–234, Santa Cruz, CA, June 1992. IEEE.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[Gir89] Jean-Yves Girard. Towards a geometry of interaction. Number 92 in Contemporary Mathematics, pages 69–108. AMS, 1989.

[GM96] Stefano Guerrini and Andrea Masini. Parsing MELL proof nets. Technical report, Dipartimento di Informatica, Pisa, 1996.

[GMM96] S. Guerrini, S. Martini, and A. Masini. Coherence for sharing proof nets. In Harald Ganzinger, editor, *Proceedings of the 7th International Conference on Rewriting Techniques and Applications (RTA-96)*, volume 1103 of *Lecture Notes in Computer Science*, pages 215–229, New Brunswick, NJ, USA, 1996. Springer-Verlag. Extended abstract.

[GMM97] Stefano Guerrini, Simone Martini, and Andrea Masini. Proof nets, garbage, and computations. In P. De Groote and J. R. Hindley, editors, *Typed lambda calculi and applications: Third International Conference on Typed Lambda Calculi and Applications, TLCA '97*, volume 1210 of *Lecture Notes in Computer Science*, Berlin, 1997. Springer-Verlag.

[Gue96] Stefano Guerrini. *Theoretical and Practical Issues of Optimal Implementations of Functional Languages.* Phd thesis, Dipartimento di Informatica, Pisa, 1996. TD-3/96.

[Gue97] Stefano Guerrini. A general theory of sharing graphs. IRCS Report 97–04, Institute for Reasearch in Cognitive Science, University of Pennsylvania, Philadelphia, March 1997. Submitted for publication: invited submission to a special issue of Theoretical Computer Science.

[Lam90] John Lamping. An algorithm for optimal lambda calculus reduction. In *Proceedings of Seventeenth Annual ACM Symposyum on Principles of Programming Languages*, pages 16–30, San Francisco, California, January 1990. ACM.

[Lév78] Jean-Jacques Lévy. *Réductions Correctes et Optimales dans le lambda-calcul.* PhD Thesis, Université Paris VII, 1978.

[MM95] S. Martini and A. Masini. On the fine structure of the exponential rule. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 197–210. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.

[Reg92] Laurent Regnier. *Lambda-Calcul et Reseaux.* Phd Thesis, Université Paris 7, January 1992.