Coherence for sharing proof-nets [1]

Stefano Guerrini [a,2] Simone Martini [c,3] Andrea Masini [b,3]

[a] *Dipartimento di Scienze dell'Informazione - Università Roma I, 'La Sapienza' Via Salaria, 113 - I-00198, Roma, Italy - Email:* `guerrini@dsi.uniroma1.it`

[b] *Dipartimento di Matematica, Università di Trento - Via Sommarive 14 - I-38050, Povo-Trento, Italy - email:* `masini@science.unitn.it`

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

Sharing graphs are an implementation of linear logic proof-nets in which a redex is never duplicated. In their usual formulation, sharing graphs present a problem of coherence: if the proof-net N reduces by standard cut-elimination to N', then, by reducing the sharing graph of N we do *not* obtain the sharing graph of N'. We solve this problem by changing the way the information is coded into sharing graphs and introducing a new reduction rule (*absorption*). The rewriting system is confluent and terminating. The proof exploits an algebraic semantics for sharing graphs.

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

1 Introduction

Implementations of functional languages based on graph rewriting need sophisticated techniques to control the runtime duplication of subgraphs. From a theoretical point of view, we know after [Lév78] that, given a normalizable λ-term, there is an optimal (in the number of beta-reductions) reduction strategy to reach the normal form. Since, however, it is a parallel strategy

[1] Extended and revised version of [GMM96].

[2] Partially supported by: HCM Project CHRX-CT93-0046 and CNR GNSAGA.

[3] Partially supported by: MURST COFIN 97 'Tecniche formali per la specifica, l'analisi, la verifica, la sintesi e la trasformazione di sistemi software' and ERB 4061 PL 97-0244 'LINEAR'.

(counting as a single step the simultaneous reduction of several redexes, those belonging to the same *family*), how to implement this strategy remained open until Lamping [Lam90] introduced his *sharing graphs*.

Sharing graphs are based on three main ideas. First, any time a duplication seems required (e.g., when a bound variable appears several times in the body of a term), it is not actually performed; it is instead indicated (in a somewhat lazy way) by specific (new) nodes in the graph (*fans*, in Lamping's terminology). Second, special reduction rules are added to perform the actual duplication in a controlled way (a redex will be never duplicated). Finally (and non trivially), there is a way to mark the boundary of the subgraph where duplication has to happen (again new nodes, the *brackets*). The reduction then proceeds in a distributed and asynchronous way, firing locally those reduction rules that apply. The crucial properties to show are then: (i) this asynchronous process terminates (if the term has a normal form); (ii) the normal form is (a possibly shared representative of) the same we would reach doing the reduction in the standard way; and (iii) no useless duplication is ever done (i.e., optimality of beta-reduction).

Following Lamping's breakthrough, several papers generalized and improved his result. First, Gonthier, Abadi, and Lévy [GAL92a,GAL92b] realized that Lamping's method was in fact a way to reduce linear logic proof-nets [Gir87] and that the information needed to mark the boundary of the subgraph to be duplicated was a local and distributed representation of the (global) notion of (linear logic) 'box'. Asperti showed how the same problems might be approached from a categorical point of view [Asp95b], and Asperti and Laneve generalized the theory to the 'interaction systems' [AL94,AL96]. The relations with the geometry of interaction are investigated in [ADLR94].

Sharing graphs present a problem of coherence. Suppose that the proof-net (or lambda-term) N reduces by standard cut-elimination (beta-reduction) to N'. Then, by reducing the sharing graph corresponding to N, we do *not* obtain the sharing graph corresponding (in the given translation) to N'. The recovering of the proof-net N' is instead obtained by the so-called *read-back* process, a semantically based procedure *external* to the reduction system, which essentially computes the equivalence quotient of all the sharing graphs representing the same proof-net (term). A first contribution towards the solution of this problem is the notion of safeness in [Asp95a]: some additional reductions can be performed when certain safety conditions hold, allowing a further simplification of the net.

We adopt a different approach. The main contribution of this paper is a solution to the coherence problem (for restricted proof-nets, see below) obtained by changing the way the information is coded into sharing graphs. This is achieved via two technical tools: (i) a new reduction rule (*absorption*) allowing a simplification of the net in some critical cases; (ii) a clear separation of the logical and control information in the representation of a net. The logical information takes the form of levels on the formulas of the proof-net; control is expressed by unifying fans and brackets into one single node (*mux*). It is this separation to allow the formulation of the absorption reduction and to enforce coherence.

Our results, like those of most of the literature, hold for restricted proof-nets, where weakening is not allowed. It should be clear that any approach to cut-elimination based on a *local* graph exploration may work only on connected components. If the syntax allows, during reduction, the creation of distinct components out of a single connected graph, then any local approach is bound to fail. This is why we ban weakening from our logic (see [GAL92b] also). A different solution might be to change the logic in order to allow weakening; e.g., we might restrict to the image of the implicative fragment of intuitionistic logic (that is, of typed λ-calculus) inside linear logic (see [Gue98]) or, we might allow weakening introduction at the axioms only (see [GMM97]).

The insight needed to introduce our new techniques came from the proof theory of modal logics. In the context of proof-nets, the already mentioned notion of box is necessary to ensure soundness of the introduction of a modal connective (the of-course '!') and to allow the proper reduction of the proof-net during the cut-elimination process. A box is a global, explicitly given notion: each occurrence of an of-course connective in the proof-net 'comes together' with a certain subgraph, its box. In [MM95]—applying to linear logic ideas and techniques previously developed for modal logic, see [MM96]—we discovered that a different, straightforward approach was possible, labeling with natural number indexes the formulas of the proof-net. The approach of [MM95], moreover, allowed a clear recognition, at any time, of the boundary of the box. This suggested our new, simple absorption rule. The approach has been applied to the optimal reduction of lambda-terms in [Gue98], where the main algebraic techniques necessary to prove its correctness are developed. A generalization of the technique and detailed proofs may be found in Guerrini's thesis [Gue96] or in [Gue99].

2 Formulas, levels, and exponentials: from natural deduction to proof-nets

In [MM95] we have presented an approach to the linear logic modality *of-course* in a natural deduction setting.

In the proof-theory of modal logics there is a long tradition, starting from Kripke, devoted to *indexed* systems where formulas are suitably decorated in order to enforce the context constraints on the rules of the various logics. The approach followed in [MM95] is to index usual linear formulas with natural numbers. The formula A indexed with n, say the *level* of A, is denoted by A^n.

Levels allow the formulation of introduction/elimination rules for ! without explicit reference to the shape of the context:

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ A^{k+1}\end{array}}{!A^k}\,!\mathcal{I}\quad {\scriptstyle k\geq\#\Gamma} \qquad\qquad \dfrac{\begin{array}{c}\Gamma\\ \vdots\\ !A^{k}\end{array}}{A^{k+v}}\,!\mathcal{E}_{v\geq 0}$$

where $\#\Gamma$ denotes the maximum level of the formulas in Γ; $\#\Gamma = 0$ when Γ is empty.

It is worth comparing the two exponential rules with the rules for universal quantification:

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ A\end{array}}{\forall x.A}\,\forall\mathcal{I}\quad {\scriptstyle x\notin FREE(\Gamma)} \qquad\qquad \dfrac{\begin{array}{c}\Gamma\\ \vdots\\ \forall x.A\end{array}}{A[t/x]}\,\forall\mathcal{E}$$

Indeed, as the introduction of ! decrements the level of the conclusion of exactly one, so the introduction of $\forall$ binds exactly one variable. The side condition $k \geq \#\Gamma$, is analogous to the usual constraint that x be not free in the active premises of the derivation. Again, as the elimination of ! raises the level of the conclusion of an arbitrary increment, so the elimination of $\forall$ allows the introduction of a new term t with an arbitrary number (possibly zero) of new free variables. This analogy has been a leading idea of the 2-sequents approach and keeps holding when we consider reduction of proofs.

In such linear, natural deduction proofs, exponential redexes and their reductions may be defined as follows:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ A^k\end{array}}{!A^{k-1}}\,!\mathcal{I}}{A^{k-1+j}}\,!\mathcal{E} \qquad \text{reduces to} \qquad \begin{array}{c}[j-1]_{k-1}\mathcal{D}\\ A^{k-1+j}\end{array}$$

where the (meta) notation $[n]_k\mathcal{D}$ means the result of incrementing by n all the levels greater than k in the deduction $\mathcal{D}$. Formally:

absorption:

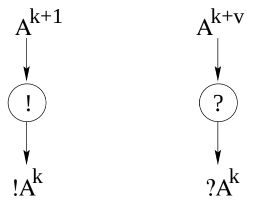
$$\text{If } v \leq i: \qquad [n]_i \begin{pmatrix} \mathcal{D} \\ \alpha^v \end{pmatrix} = \begin{matrix} \mathcal{D} \\ \alpha^v \end{matrix}$$

reindexing:

$$\text{If } v > i: \qquad [n]_i \begin{pmatrix} \mathcal{D} \\ \alpha^v \end{pmatrix} = \begin{matrix} [n]_i\mathcal{D} \\ \alpha^{v+n} \end{matrix}$$

The side condition on $!\mathcal{I}$ ensures correctness of the reduction. Under the analogy 'modalities are quantifiers', this process of *reindexing* corresponds to *substitution* in first-order logic, the absorption case corresponding to a test on the freeness of the involved variable. (For a rigorous treatment of the first order case see [TvD88].)

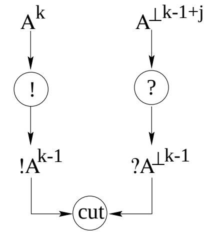
Let us now move towards a proof-net framework. We build proof-structures as usual, but we label (hyper)nodes with indexed formulas. As usual, (natural deduction) introduction rules of a connective $*$, become $*$-links in proof-structures; elimination rules of $*$ become $*^\perp$-links, where $*^\perp$ is the dual of $*$. In particular, $!\mathcal{I}$ introductions become ! links, while $!\mathcal{E}$ eliminations become ? links:



The other multiplicative links are given as usual, with the restriction that all the formulas involved in a link must have the same level (in the case of natural deduction, this is not true for the $\otimes$ and $\mathbf{1}$ elimination rules).

Levels allow the elimination of the global concept of *box* as a primitive notion; indeed, the boxes of a net can be reconstructed by means of its levels. In fact, given an ! link with conclusion $!A^k$, its associated box is the connected subnet B with conclusions $\Gamma, !A^k$ (we remind that $!A^k$ is the principal door, while the formulas in Γ are the auxiliary or secondary doors of B): all the internal nodes (formulas) of B have a level greater than k; all the secondary doors (conclusions) Γ have a level $\leq k$ (indeed, for technical reasons, the formal definition of box that we shall use later slightly differs from the previous one, see Definition 5 and Remark 6)—we remark that the *level constraint on secondary doors* corresponds *exactly* to the side condition of the $!\mathcal{I}$ rule.

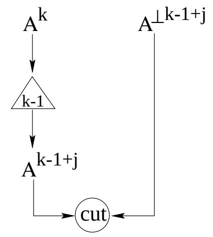
Since we have (implicit) boxes, the reduction of an exponential cut



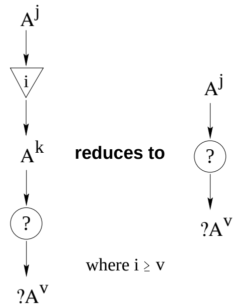
(1)

may be performed as usual, although in general, that might involve *reindexing* a subnet. In fact, after the elimination of an exponential cut, the interior of a box is moved inside some other boxes, increasing thus the box-nesting-depth (the levels) of the formulas in the box. This operation closely corresponds to what we indicated as $[n]_k\mathcal{D}$ in the natural deduction setting. The general situation of an exponential cut (contraction included) is depicted in Figure 3, where the notation $\Pi[k_i - k]$ means that all the levels of the subnet Π have been increased by $k_i - k$.

In this *standard exponential cut-elimination*, reindexing and duplicating a subnet is thought of as a single, global (meta) operation. In this paper, following the sharing graph approach, we will internalize that process by means of explicit operators (links). Thus, to reduce the exponential cut in (1), we introduce a new link named *lift* and we rewrite the cut to



Lifts mimic (at the object level) the reindexing operation: they reindex the box associated to the ! link eliminated reducing an exponential cut by means of local rewriting rules. To constrain lifts to the interior of their boxes, an *absorption rule* is introduced to stop lift propagation:



Observe that the constraint on the absorption rule is *exactly* the same as that of the natural deduction case.

In nets with contractions, the duplication process too may be handled in a 'lazy' way, similarly to reindexing. In full generality, therefore, we introduce a new link named *mux* in charge of both duplication and reindexing.

3 Leveled nets, proof-nets, reduction

3.1 Structures and nets

We introduce in this section the net concepts that we will use in the sequel. The most standard notions are those of restricted (because weakening is not allowed) proof ℓ-structure and proof ℓ-net (Definitions 3 and 4), though given here as hypergraphs (consistently with the presentation of [Gir87]) and with levels instead of boxes—from which the ℓ in the name. Proof ℓ-structures are special cases of *$s\ell$-structures* (sharing leveled structures of links, Definition 1), that may contain additional links in charge of duplication: *muxes* and their duals *demuxes*. (A mux correspond, in Lamping's approach, to several fans and brackets, see Remark 2.) By a *formula*, we mean a multiplicative-exponential linear logic formula; an *indexed* formula is a formula decorated with a non-negative integer, the *level* of the formula.

Definition 1 *An $s\ell$-*structure *is a finite connected hypergraph whose nodes are labeled with indexed formulas and whose hyperedges (also called* links*) are labeled from the set* $\{\mathsf{cut}, \mathsf{ax}, \parr, \otimes, !, ?\} \cup \{\mathsf{mux}[i] |\ i \geq 0\} \cup \{\mathsf{demux}[i] |\ i \geq 0\}$*; the integer i in a (de)mux is the* threshold *of the link. Allowed links and nodes are drawn in Figure 1. The source nodes of a link are its* premises*; the target nodes are its* conclusions*. Premises and conclusions are assumed to be distinguishable (i.e., we have the* left/right *premise of a link, the i-th conclusion, and so on), with the exception of the premises of the ?-links. In an $s\ell$-structure, each node must be conclusion of exactly one link and premise of at most one link; those nodes that are not premises of any link are the* net conclusions*. Unary (de)muxes are also called* lifts.

We assume that $s\ell$-structure axioms have atomic conclusions. Such a restriction does not decrease the expressive power of $s\ell$-structures. However, it would be possible to have a more economic representation of nets, allowing non-atomic axioms, even if in that case, we should forbid axioms whose conclusions are exponential formulas (see [Gue96]).

Remark 2 Figure 2 states the correspondence between our $s\ell$-structures and

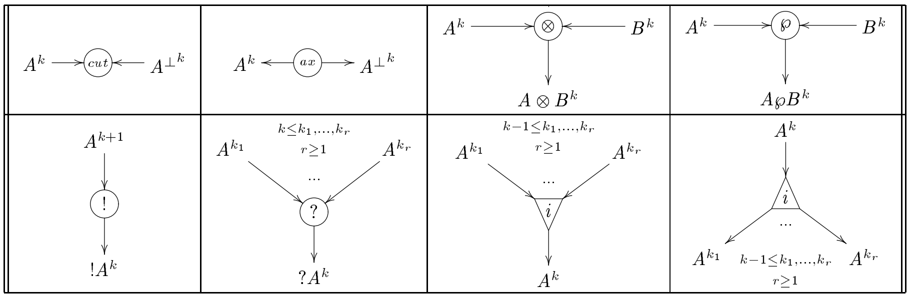


Fig. 1. $s\ell$-structure links

the nets of [GAL92b,Asp95b] (see Remark 7 also). A (de)mux with n auxiliary doors corresponds (in Asperti's notation) to a tree of fans with n leaves, followed by chains of brackets closed at the top by a croissant—one chain for each leaf. The length of a chain is the offset of the corresponding door (i.e., the difference between the level of the formula assigned to such a door and the one assigned to the principal door of the mux) increased by 1. The top of Figure 2 shows the binary case (the triangle on the right side of the equivalence is then a fan and not a mux). A ?-link with a conclusion at level k corresponds to a bracket with an index equal to k (the Gonthier index would be 0) followed by a configuration analogous to that of a mux with threshold k and conclusion at level $k+1$ (cf. the $\triangleright_{\mathsf{exp}}$ rule). The corresponding binary case is drawn at the bottom-left of Figure 2. An !-link is just a bracket indexed as the bottom bracket of a corresponding ?-link.

Definition 3 *A* proof ℓ-structure *is an $s\ell$-structure that does not contain (de)muxes.*

Let $\mathbb{PN}$ be the set of proof-nets à la Girard. We associate to each $P \in \mathbb{PN}$ a (unique!) proof ℓ-structure $\mathcal{D}[P]$, the *decoration* of P, in the following way: the level of any node of $\mathcal{D}[P]$ is equal to the number of exponential boxes containing that node in P.

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

Definition 5 *Let S be a proof ℓ-structure and let A^k be a premise of an !-link; the* box *of A^k is the unique sub-hypergraph $bx_S[A^k]$ of S verifying the following properties:*

(i) $A^k \in bx_S[A^k]$ *(A^k is the* principal door *of $bx_S[A^k]$);*
(ii) $bx_S[A^k]$ *is a proof ℓ-net;*
(iii) each net conclusion of $bx_S[A^k]$ different from the principal door is a premise, in S, of a ?-link with conclusion at level $j < k$ (such ?-premises are the auxiliary *or* secondary doors *of the box);*
(iv) for each $B^j \in S$, if $B^j \in bx_S[A^k]$, then $j \geq k$.

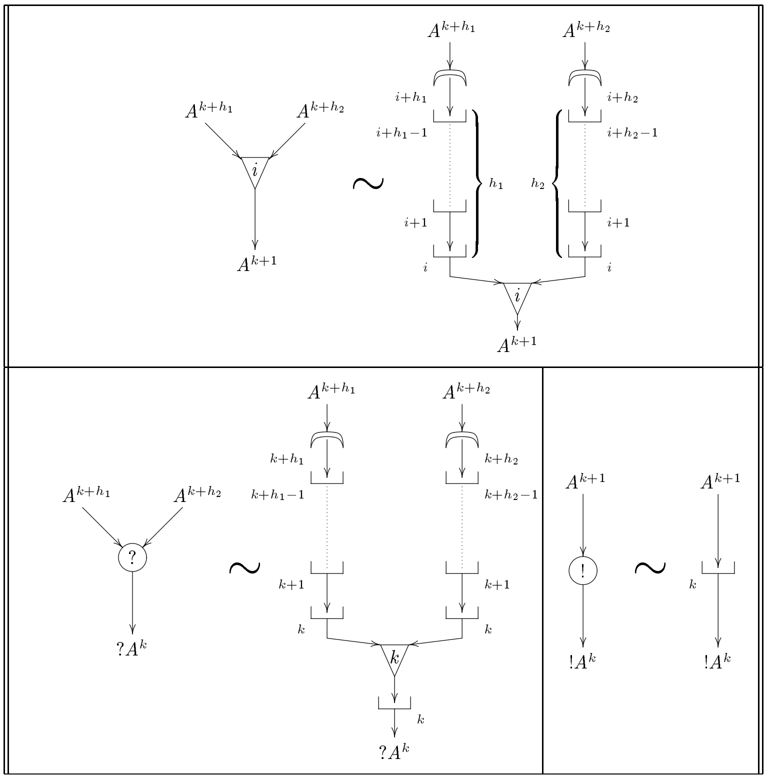


Fig. 2. Correspondence between $s\ell$-structures and sharing graphs

We denote by $BX[S]$ the set of boxes of S. Because of the definition of $s\ell$-structure, boxes are connected.

Remark 6 According to Definition 5, the ! and ? links bounding a box are not included into it. This choice is consistent with the inclusion of contraction into ? links, for otherwise we would loose the box nesting property (i.e., two boxes are either disjoint or nested). By the way this is just a matter of presentation (for instance, in [Gue99], where there is an explicit contraction link, ! and ? links belong to their boxes).

3.2 Reduction

The $s\ell$-structures may be used to implement a local and asynchronous version of the standard cut-elimination for proof-nets (as defined in [Gir87]). The elimination of *propositional* cuts (i.e., tensor/par and axiom/cut) is directly mirrored in the corresponding rules. Figure 3 shows how to perform *standard exponential cut-elimination.* Observe that, first, the box Π is (globally) duplicated; second, after the reduction, the different copies of Π may have been put inside some other boxes (this happens when the ?-node is a secondary door of another box). The notation $\Pi[k_i - k]$ means that all the levels of Π have been increased by $k_i - k$.

Levels and (de)muxes are designed to take care in a local way of both these aspects of the exponential reduction: multiple premises handle (incremental) duplication, while the threshold handles the (incremental) *reindexing* of the box (i.e., the re-computation of the levels of its nodes).

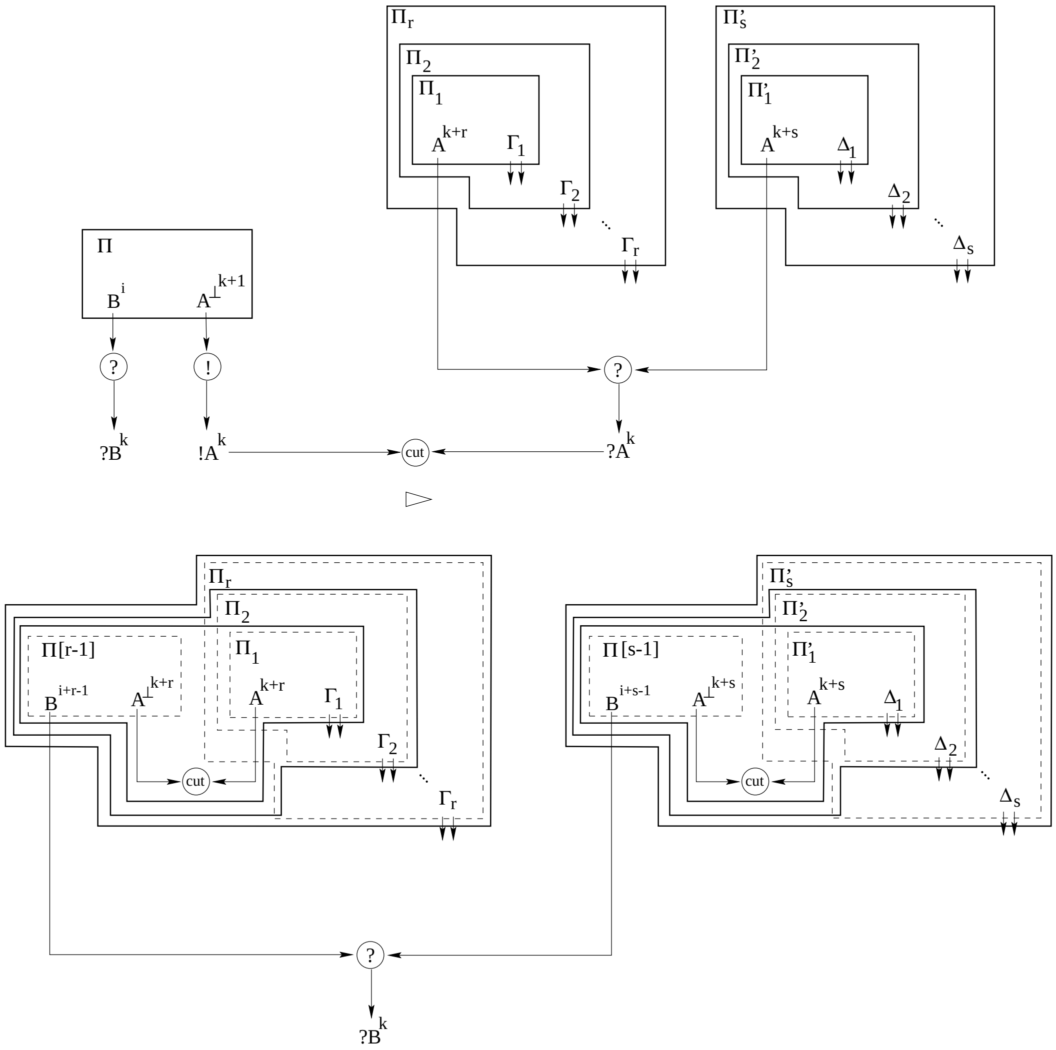


Fig. 3. Exponential cut reduction

We distinguish the rules in two kinds: the logical (or β) rules (Figure 4), where interaction happens through a cut-link (corresponding to a logical cut-elimination step); and the π rules (Figures 5 to 9), when one of the interacting nodes is a mux/demux (corresponding to a step of incremental duplication and/or reindexing). In the figures, we do not list the symmetric cases of those shown (e.g., *the* $\triangleright_{\mathsf{dup}}$ rules where the interaction happens through another premise of the ? link); moreover, $*$ stands for $\otimes$ or $\invamp$.

The set $\pi_{\mathsf{opt}} = \pi - \triangleright_{\mathsf{dup}}$ contains the only rules allowed during an *optimal* reduction (see Section 3.4). We stress the presence of the *absorption* rule ($\triangleright_{\mathsf{abs}}$), corresponding to the case when the mux reaches the border of a box

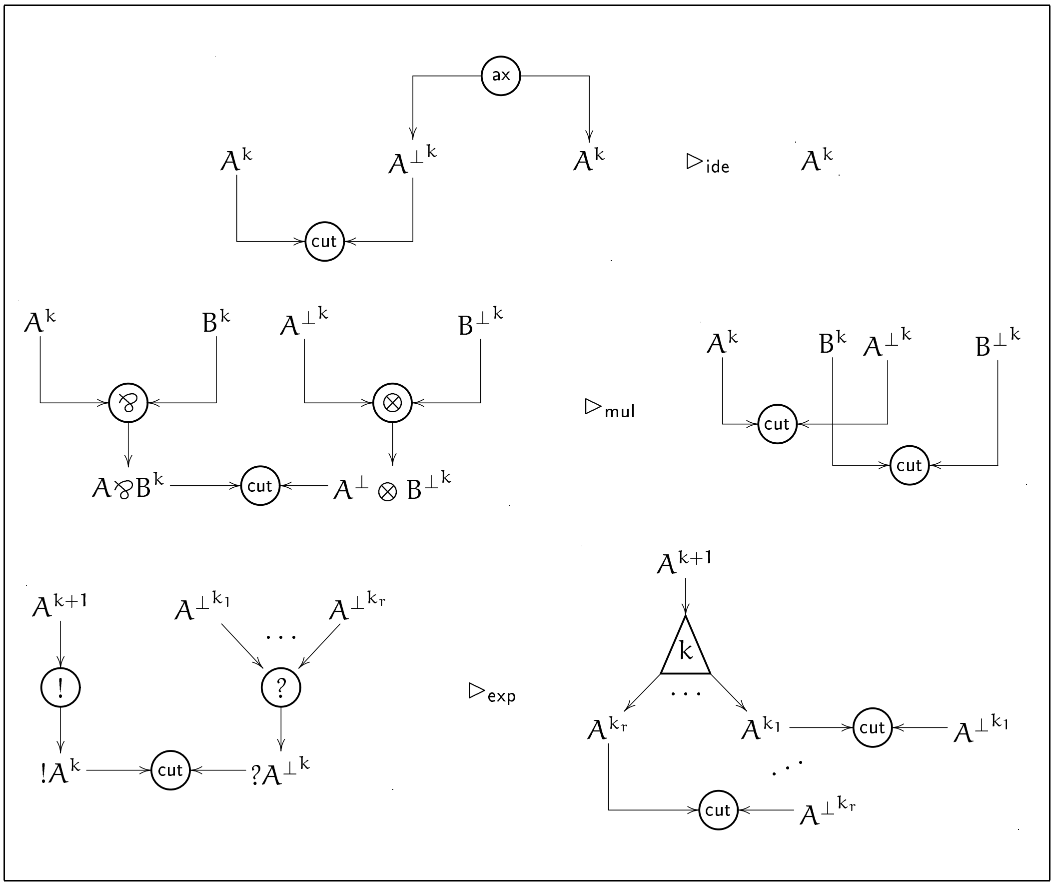


Fig. 4. Logical (or β) rules.

(through one of its secondary doors) and has therefore exhausted its job. It is motivated by the proof theoretical work in [MM95,MM96] (see also Section 7) and it is a special case of a safe reduction [Asp95a].

Remark 7 Any rule of π_{opt}, but $\rhd_{\mathsf{abs}}$, is admissible with respect to the reductions of [GAL92b] and the translation of Remark 2. The fact that $\rhd_{\mathsf{abs}}$ is not valid if one uses the notation of [GAL92b] depends on the choice to unify logical and control information in the same nodes; in fact, this makes impossible to recognize in a local way when a bracket configuration corresponds to a secondary door of a box (see also Section 7). If one sticks to the notation of [GAL92b], the solution is that indicated in [Asp95a]: add another tag to each node, to record its 'safeness'.

Remark 8 Interactions between muxes are allowed only between pairs in which the conclusion of a mux is the premise of a demux—in interaction nets terminology, the mux and the demux are connected through their principal doors (see π_{swap} and π_{anh}). Correspondingly, a non-identity logical link interacts with a demux when its conclusion (i.e., its principal door) is the premise of that demux (compare π_{odup} with π_{swap}). Generalizing the rules presented in [Asp95b], a mux may interact with a logical link (see π_{dup}) when its conclusion

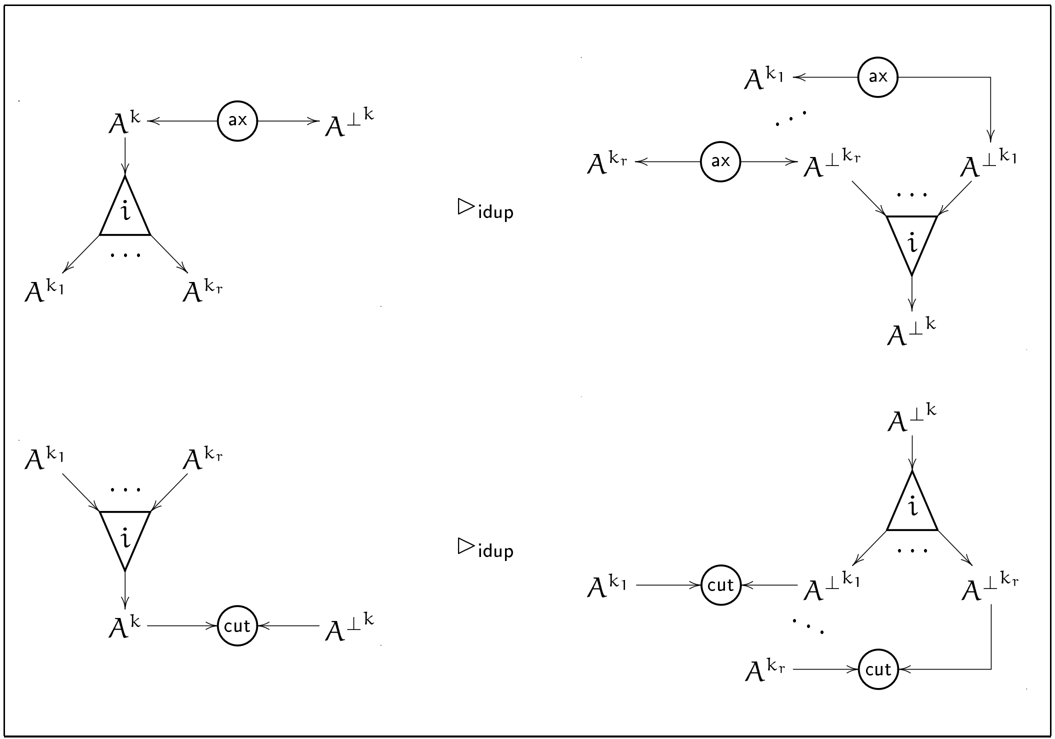


Fig. 5. Duplication rules: axiom and cut.

is a premise of that logical link. Identity links are straight connections between their formulas, their only purpose is to invert an orientation. Thus, a cut-link interacts with a mux when one of its premises is a conclusion of that mux, and vice versa for the complementary case of an axiom link and a demux (see π_{idup}).

Remark 9 It is impossible for a mux to reach a net conclusion. In fact, let i be the threshold of a (de)mux and let A^k be its (premise)conclusion. The relation $k > i \geq 0$ is invariant under reduction and any net conclusion has level 0.

3.3 An example

Figures 10, 11 and 12 depict a simple example that focuses on reindexing performed by muxes, that is, the core of our proposal.

The net on the left-hand side of Figure 10, call it G_1, is a restricted proof ℓ-net. Boxes are not really necessary—they are displayed to stress the relationship between our restricted proof ℓ-nets and classical proof-nets (erasing the levels of G_1 we get a proof-net à la Girard). G_1 contains two cuts that can be reduced by applying (twice) $\triangleright_{\mathsf{exp}}$.

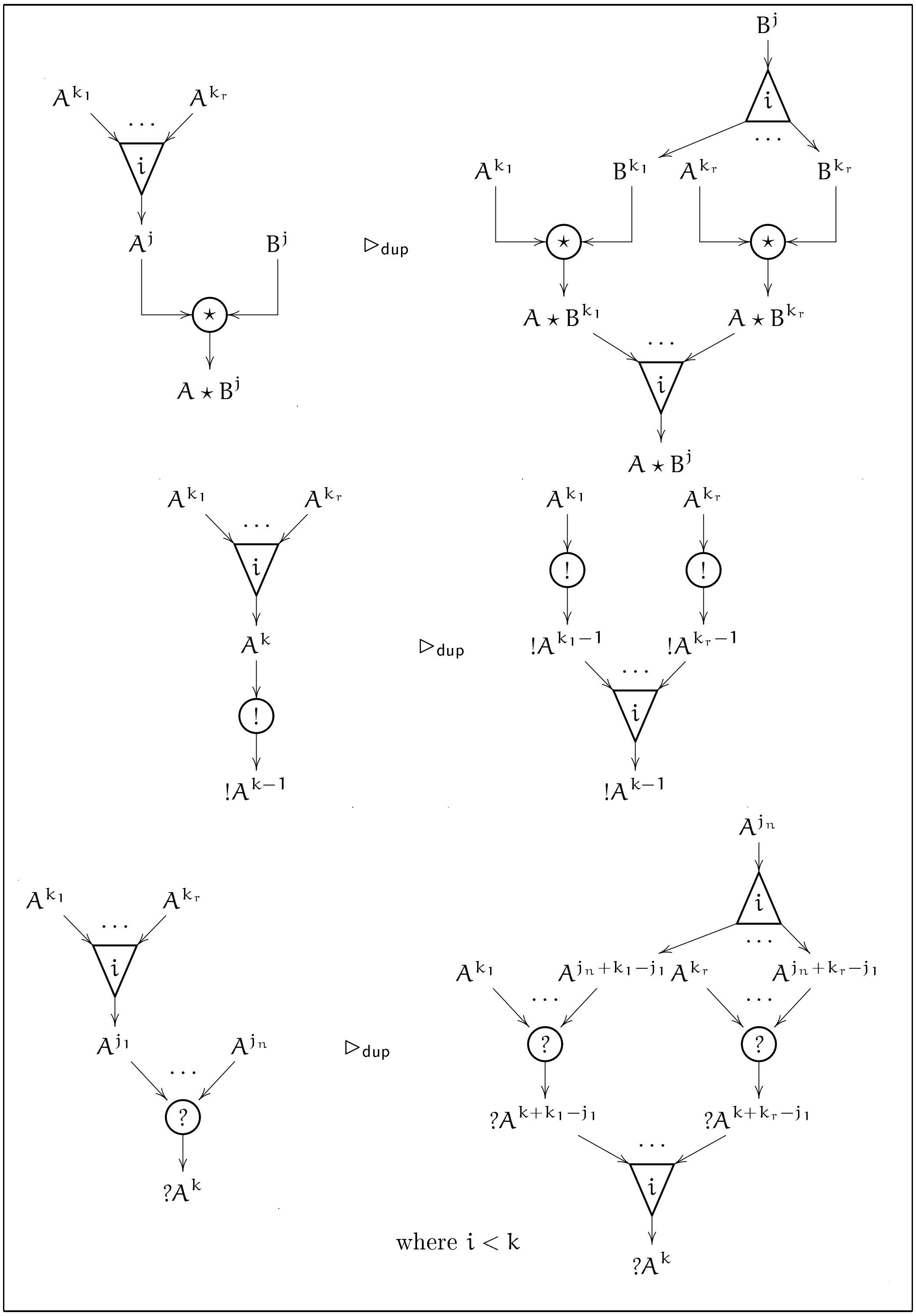


Fig. 6. Duplication rules: non-optimal duplication ($*$ stands for $\otimes$ or $\parr$).

The right-hand side of Figure 10, call it G_2, is the net after the execution (in any order) of the two exponential cuts. These reductions inserted two demuxes: one with threshold 0 and one with threshold 1.

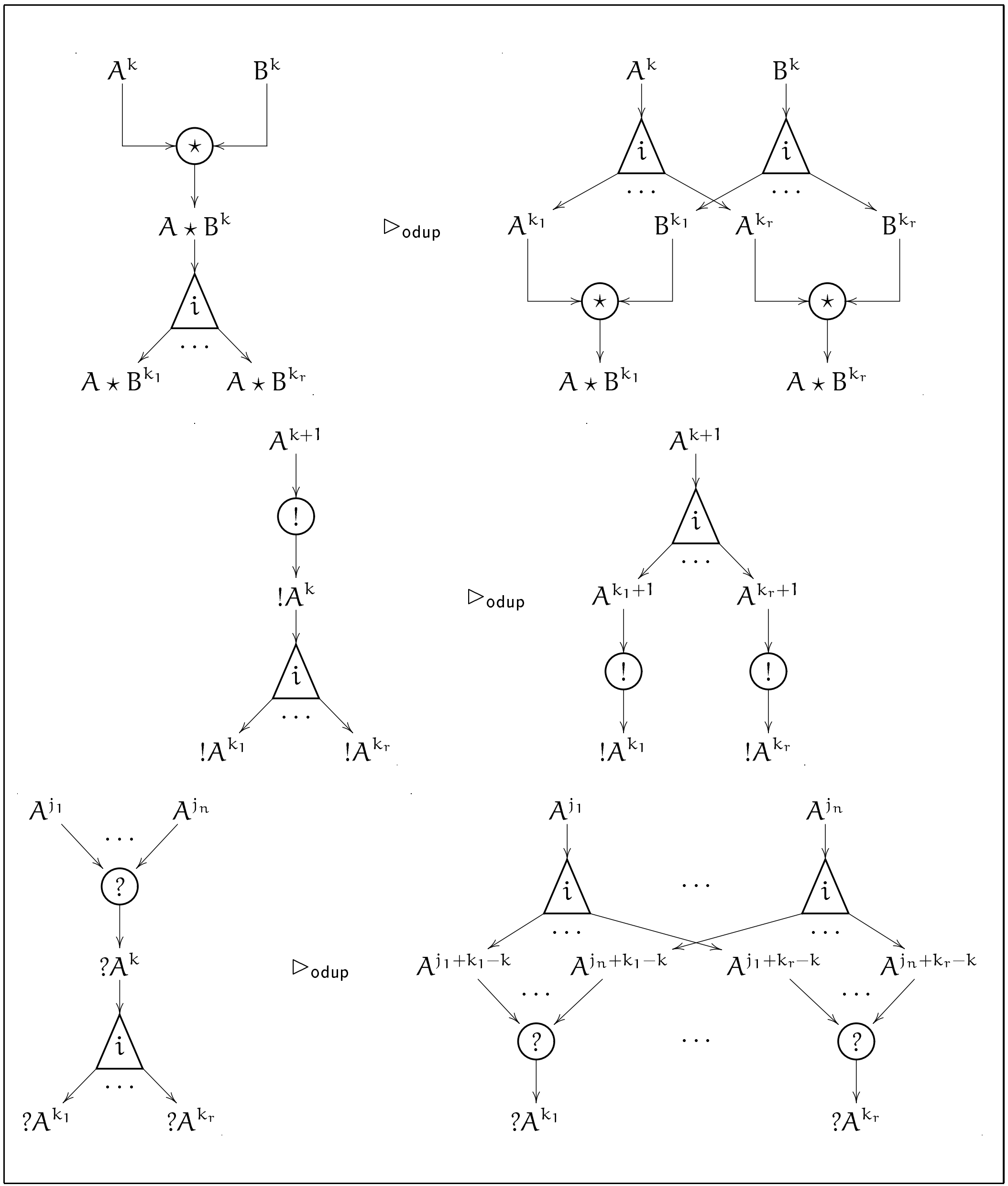
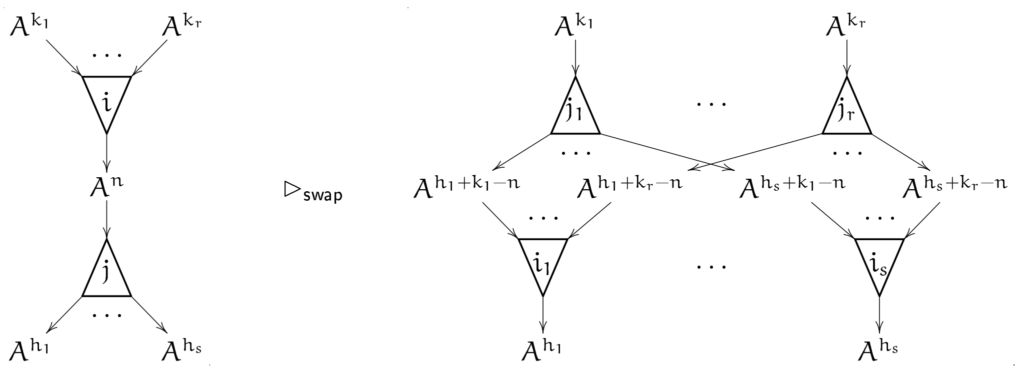


Fig. 7. Duplication rules: optimal duplication ($*$ stands for $\otimes$ or $\bindnasrepma$).

The left-hand side of Figure 11, call it G_3, shows the result of a propagation of the mux with threshold 0 by executing one $\rhd_{\mathsf{odup}}$ and one $\rhd_{\mathsf{idup}}$.

G_3 contains a redex given by two facing muxes. Since the thresholds are different, we apply $\rhd_{\mathsf{swap}}$ (but not $\rhd_{\mathsf{anh}}$). The result of this reduction is the right-hand side, call it G_4, of Figure 10. Note the change of threshold in one of the two muxes after the swap (the mux that before the swap had the lower threshold).

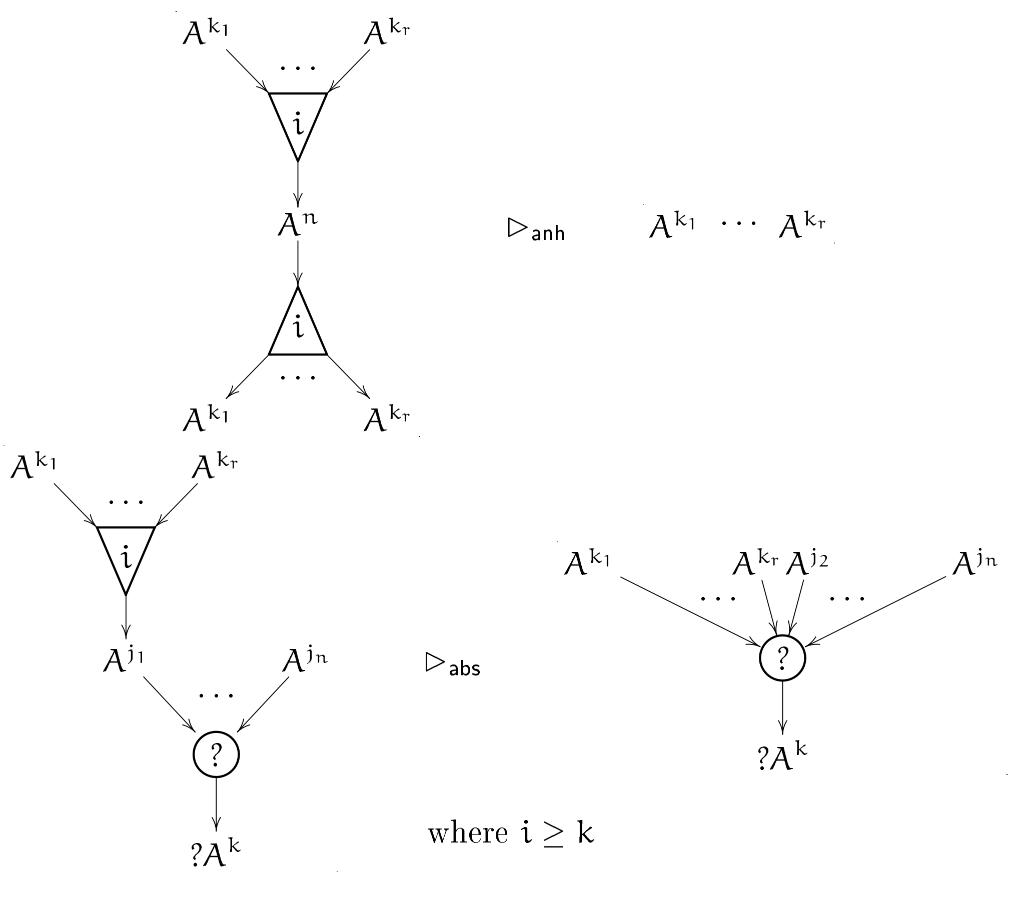
Now, the muxes of G_4 can freely propagate. The result is the net G_5 on the left-hand side of Figure 12. In G_5, the muxes are above the secondary doors



where $i_1 = \cdots = i_s = i$, and $j_1 = j + k_1 - n, \ldots, j_r = j + k_r - n$ when $i < j$,

$i_1 = i + h_1 - n, \ldots, i_s = i + h_s - n$ and $j_1 = \cdots = j_r = j$, when $i > j$.

Fig. 8. Duplication rules: swap.



where $i \geq k$

Fig. 9. Simplification rules

of the boxes (w.r.t. the original net G_1) involved in the reductions. The side condition of rule $\triangleright_{\mathsf{abs}}$ holds and the result of its application is the net on the right-hand side of Figure 12, say G_6. The boxes drawn on G_6 are obtained by applying Definition 5 (note that G_6 does not contain lifts). We see that G_6 is the net we would have got by applying the standard global reduction to eliminate the exponential cuts in G_1.

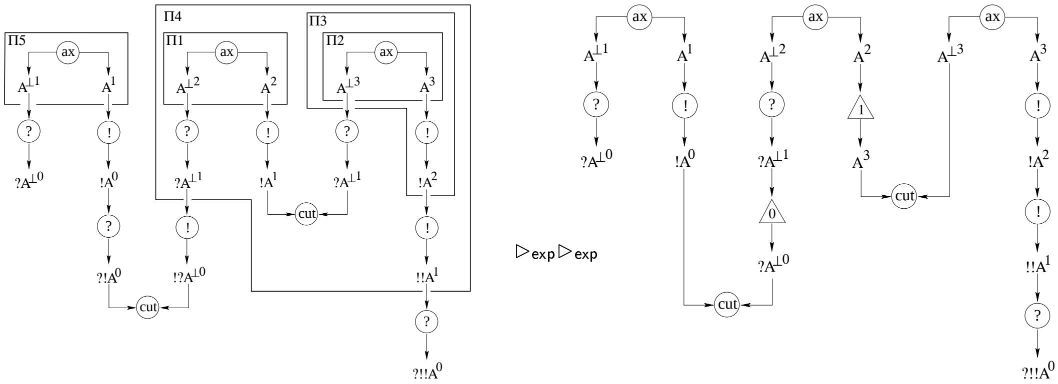


Fig. 10. Example: Exponential reductions

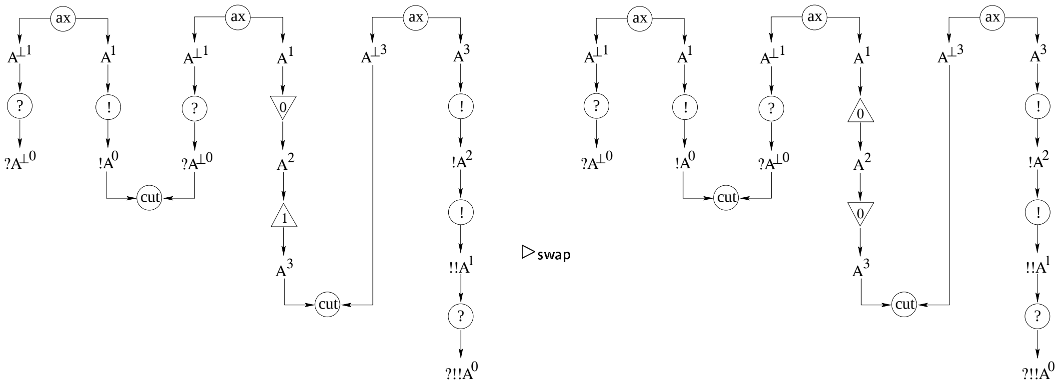


Fig. 11. Example: Swap reduction

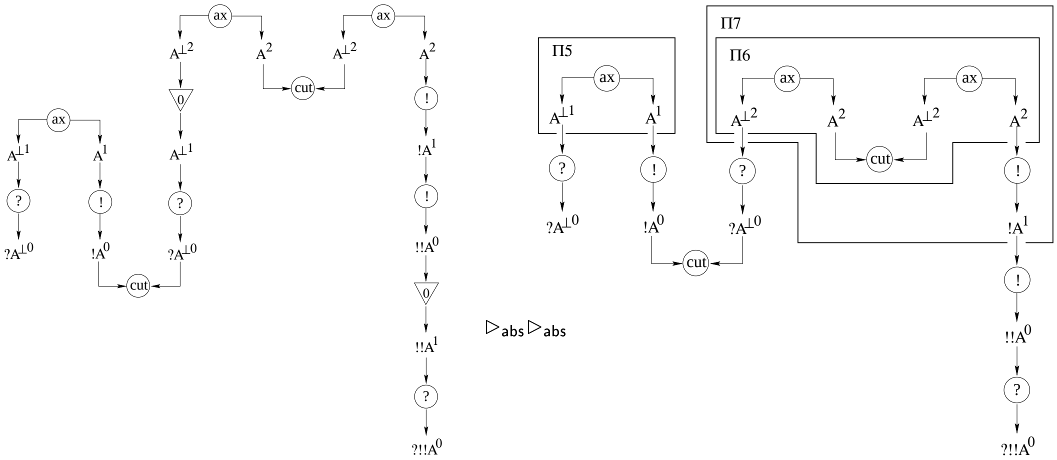


Fig. 12. Example: Absorption

Optimality for β reduction of λ-calculus was defined and studied by Lévy [Lév78,Lév80]. Analogous analysis may be given for proof-nets (see [GAL92b], or [AL94]). By a suitable labeling of (standard) proof-nets, a Lévy labeled rewriting system for proof-nets is defined. In it, as in the λ-calculus case, residuals of a cut have the same label and new labels appear only when new cuts are created during reduction. Starting from a labeled proof-net N in which all nodes have different labels, two cuts (not necessarily belonging to the same reduct of N) are in the same *Lévy family* iff they have the same label. A *family reduction* is a sequence of parallel rewritings $R_1 R_2 \ldots$ s.t. all the cuts in R_i are in the same family. A *complete* reduction is a sequence of rewritings where at each step *all* the cuts of the same family are reduced (i.e., if r and r' are two cuts in the same family, then $r \in R_i$ implies $r' \in R_i$). Finally, a *call-by-need reduction* of N is a sequence of rewritings in which at least a needed cut is reduced at every step (a cut is needed when it, or more precisely a residual of it, appears in every reduction sequence starting from N). Lévy's main argument [Lév80] is that the optimal cost of the reduction of a λ-term is the number of β reductions of a call-by-need complete family reduction (in the λ-calculus case, the left-most-outer-most strategy is call-by-need). We assume the same measure (β contractions) for proof-nets.

Remark 10 Any redex of a restricted proof ℓ-net is *needed*. This is not surprising, since without weakenings no redex belongs to a subgraph that will be erased. Therefore, any restricted ℓ-net reduction strategy is call-by-need.

To conclude these notes on efficiency, we stress that the solution to the coherence problem presented in this paper is motivated by pure proof theoretical considerations. We have not studied the efficiency of our approach compared with other approaches. Finding a good measure for the computational complexity of asynchronous and local reductions in proof-nets (and λ-calculus) is an important open problem, outside the scope of the present paper (e.g., [AM98,LM96,ACM00]).

4 Coherence

We state in this section our main result, that the reduction rules $\beta + \pi$ solve the coherence problem for $s\ell$-structures. The proofs are not trivial, since the rules may be fired in any order (logical and non-logical reductions will be in general interleaved), and they will be presented in the following sections. The proof strategy, analogous to that used in [Gue98] for the λ-calculus, is to simulate $s\ell$-structures over restricted proof ℓ-nets and it will require the introduction of an algebraic semantics for $s\ell$-structures, here restricted to the essential (for a detailed presentation we refer the reader to [Gue99]).

Let an $s\ell$-structure G be *correct* iff there exists a restricted proof ℓ-net N s.t. $N \rhd^* G$; informally, an $s\ell$-structure is correct if it represents a restricted proof ℓ-net.

Theorem 11 *Let G be a correct $s\ell$-structure.*

(i) The π rules are strongly normalizing and confluent on G. The π normal form of G is a restricted proof ℓ-net.
(ii) The $\beta+\pi$ rewriting rules are strongly normalizing and confluent on G. The $\beta+\pi$ normal form of G is a restricted proof ℓ-net.
(iii) The π normal form of G reduces by standard cut-elimination to its $\beta+\pi$ normal form.

The third item of Theorem 11 ensures the soundness of the system. The result can be even stated in a stronger way, as in the following Theorem 13 (further, $\rhd_{\mathsf{std}}$ denotes a standard cut-elimination reduction).

Definition 12 *The* read-back $\mathcal{R}(G)$ *of a correct $s\ell$-structure G is the π normal form of G.*

Theorem 13 *Let G be a correct $s\ell$-structure and N be a restricted proof ℓ-net s.t. $N \rhd^* G$. Then $N \rhd^*_{\mathsf{std}} \mathcal{R}(G)$.*

According to Section 3.4, there is a strategy minimizing the number of $\rhd_\beta$ rules.

Theorem 14 *The $\beta+\pi_{\mathsf{opt}}$ rewriting rules are Lévy optimal.*

Confluence of $\beta+\pi$ implies thus the following.

Theorem 15 *Let G be a correct $s\ell$-structure and N be its $\beta+\pi$ normal form. Let N_o be a $\beta+\pi_{\mathsf{opt}}$ normal form of G, then $N_o \rhd^*_\pi N$.*

By Theorem 15, normalization of correct $s\ell$-structures may be performed in two distinct steps: first optimal reduction ($\beta+\pi_{\mathsf{opt}}$), then read-back reduction (π).

The following sections will give the proofs of the previous statements. The technical core of the approach is an algebraic semantics of $s\ell$-structures presented in [Gue96,Gue99], to which we refer the reader for more insight.

The proof goes as follows. In Section 5, the main tool is the notion of *$u\ell$-structure*, whose muxes and demuxes are all unary (single premise; they are lifts, in the terminology of Definition 1). Over $u\ell$-structures we define a reduction with global duplication (for contractions) but local reindexing. Then, we assign an algebraic semantics to $u\ell$-structures and we exploit the semantics to prove confluence and strong normalization of the $u\ell$-structure reduction.

By using the notion of *sharing morphism*, then, we prove in Section 6 that any $s\ell$-structure has a *least shared instance*, which is a $u\ell$-structure. Finally, we prove that reduction of $s\ell$-structures may be simulated over reduction of $u\ell$-structures. By a simple argument, this simulation establishes the results.

5 Unshared structures and their reduction

5.1 Sharing morphisms

Definition 16 *An* s-morphism *(sharing morphism) is a surjective homomorphism $M : G_0 \to G_1$ of $s\ell$-structures which is injective when restricted to the net conclusions and that preserves the labeling of the nodes/links (i.e., the type of the links, the levels and the formulas of the nodes) and the names of the doors to which the nodes are connected.*

Let $M : G_0 \to G_1$. The $s\ell$-structure G_1 is equal in all respects to G_0 but for the number of premises of (de)muxes and ? links (e.g., a k-ary mux may be mapped to one with $k' \geq k$ premises). Furthermore, since any node (link) of G_1 is an image of at least a node (link) of G_0, we may say that 'G_0 is a less-shared-instance of G_1.' Thus, we will write $G_0 \preceq G_1$ to denote that there is at least an s-morphism from G_0 to G_1 (and $M : G_0 \preceq G_1$ to explicit that M is one of such s-morphisms). Unfortunately, not all the less-shared-instances definable in this way can be considered a 'correct' unfolding of G_1. In fact, let us assume that G_1 contains a pair of binary muxes l_1 and l_2 forming an annihilating redex (a redex for the π_{anh} rule) and that G_0 contains two unary muxes l'_1 and l'_2 s.t. $M(l'_i) = l_i$, for $i = 1, 2$. The annihilation rule for the muxes l_1 and l_2 suggests us that the label of the unique door of l'_1 and l'_2 must coincide, otherwise G_0 would contain a deadlock that was not present in G_1.

The reader may see [Gue99] for an unabridged discussion of how to obtain the correct unsharings of a (general) $s\ell$-structure. Here, we proceed by assuming further that $s\ell$-structures are correct, that is, obtained along the reduction of a restricted proof ℓ-net.

We define in this section a notion of reduction living midway between standard proof-net reduction (global duplication, global box reindexing) and $s\ell$-structure reduction (local duplication, local box reindexing).

Let an $s\ell$-structure be *unshared*, when all (de)muxes are (negative) lifts, that is, have a unique (conclusion)premise. A *$u\ell$-structure* U is an unshared $s\ell$-structure $\widetilde{U}$ for which a box assignment is given (that is, a map associating a box to each ! link of the net) in accordance with the usual constraints on boxes (i.e., the box nesting condition holds and the secondary doors of each box are conclusions of ? links).

The multiplicative β and the π rules apply unchanged to $u\ell$-structures (though the π rules always with unary muxes). The β rule for the exponential cut is instead reformulated. In such unshared version of the β rule the boxes are duplicated without altering their levels; the consistency of the level assignment is achieved by the introduction of a lift at the principal door of each duplicated box (see Figure 13).

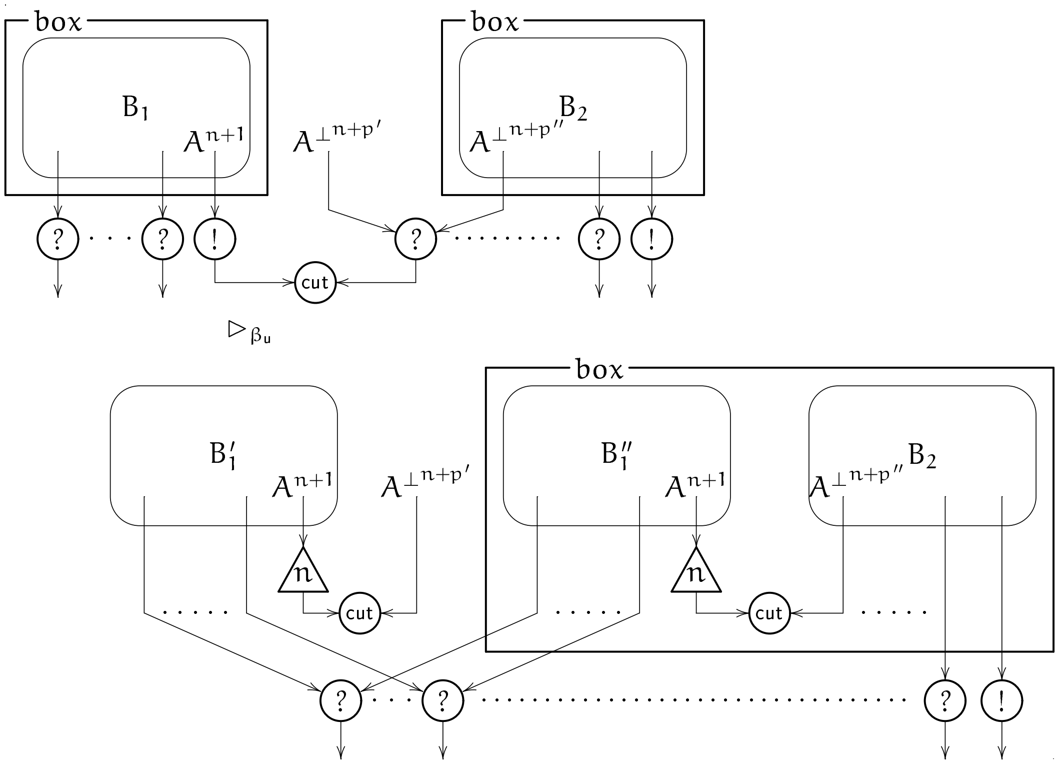


Fig. 13. The β_u rule.

Further, we will write $U \triangleright_{\mathsf{u}} U'$ to denote any (unshared) reduction of a $u\ell$-structure, and in particular we will write $G \triangleright_{\beta_{\mathsf{u}}} G'$ in the case of an unshared exponential β reduction.

Definition 17 *The set of the* correct $u\ell$-structures *is the smallest set closed under* $\triangleright_{\mathsf{u}}$ *that contains the* $u\ell$*-structures obtainable from a restricted proof* ℓ*-net assigning boxes according to Definition 5.*

Remark 18 As for restricted proof ℓ-nets, also $u\ell$-structure boxes can be avoided and computed using levels (see Proposition 23). However, the presence of lifts makes the definition of boxes more complex: it requires the introduction of the algebraic semantics that we will briefly present in section 5.3. The possibility to compute boxes justifies why in the following we will sometimes identify a correct $u\ell$-structure U with its underlying unshared $s\ell$-structure $\widetilde{U}$.

Before stating the key properties of $u\ell$-structures, let us note that there is a direct way to associate a restricted proof ℓ-net to a correct $u\ell$-structure U. In fact, let N be the net obtained erasing the levels of U and removing its lifts (by merging their premise and conclusion nodes); if N is a proof-net à la Girard, we define $\mathcal{R}_u(U) = \mathcal{D}[N]$ the *read-back* of U. It is worth noting that such a read-back is invariant under π reduction and is well-behaved w.r.t. β_{u}.

Fact 19 *Let* U *be a correct* $u\ell$*-structure for which* $\mathcal{R}_u(U)$ *is defined.*

(i) If $U \triangleright_\pi U'$*, then* $\mathcal{R}_u(U) = \mathcal{R}_u(U')$*.*
(ii) If $U \triangleright_{\beta_{\mathsf{u}}} U'$*, then* $\mathcal{R}_u(U')$ *is defined and* $\mathcal{R}_u(U) \triangleright_{\mathsf{std}} \mathcal{R}_u(U')$ *by the standard* β*-reduction of the corresponding cut.*

In general, $\mathcal{R}_u$ is a partial map from $u\ell$-structures to restricted proof ℓ-nets, but by induction on the definition of correct $u\ell$-structure and by the previous fact, we see that correct $u\ell$-structures are a relevant case.

Fact 20 *If* U *is a correct* $u\ell$*-structure, then* $\mathcal{R}_u(U)$ *is always defined.*

Further, we will also see that the read-back of a correct $u\ell$-structure corresponds to its unique π normal form $\mathcal{R}(U)$, which is indeed an a posteriori justification for the name given to these functions.

5.3 Solutions of correct $u\ell$*-structures*

For a complete presentation of the material in this subsection, we refer the reader to [Gue99].

A *lifting operator* is a triple of integers $\mathcal{L}[m, q, a]$ s.t. $m \geq 0$, $q \geq 1$, and $a \geq 0$; m is the threshold and q is the offset of the operator. The monoid of the lifting sequences LSeq is the free monoid generated by the formal product of lifting operators modulo the equivalence:

$$\mathcal{L}[m_2, q_2, a_2] \cdot \mathcal{L}[m_1, q_1, a_1] = \mathcal{L}[m_1, q_1, a_1] \cdot \mathcal{L}[m_2 + q_1, q_2, a_2] \qquad (\mathsf{SW})$$

when $m_1 < m_2$.

Let $n_0 \leq n_1$. A *lifting sequence* from n_0 to n_1 is a formal product of lifting operators $\mathcal{H} = \prod_{0<i\leq k} \mathcal{L}[m_i, q_i, a_i]$, in which, $n_0 \leq m_i < n_1 + \sum_{0<j<i} q_j$, for $i = 1, 2, \ldots, k$. The set $\mathsf{LSeq}[n_0, n_1]$ is the family of the lifting sequences from n_0 to n_1. (It is direct to check that the definition of $\mathsf{LSeq}[n_0, n_1]$ is sound w.r.t. to the (SW) equivalence.) The *global offset* $||\mathcal{H}||$ of a lifting sequence $\mathcal{H}$ is the sum of the offsets of the lifting operators in $\mathcal{H}$.

Fact 21 *Let $n_1 \leq n \leq n_2$.*

(i) If $\mathcal{H} \in \mathsf{LSeq}[n_1, n_2]$, then $n_1 \leq n_2 + ||\mathcal{H}||$.
(ii) If $\mathcal{H}_1 \in \mathsf{LSeq}[n_1, n]$ and $\mathcal{H}_2 \in \mathsf{LSeq}[n, n_2]$, then $\mathcal{H}_2 \cdot \mathcal{H}_1 \in \mathsf{LSeq}[n_1, n_2]$.
(iii) For every $\mathcal{H} \in \mathsf{LSeq}[n_1, n_2]$, there exists a unique pair $\mathcal{H}_1 \in \mathsf{LSeq}[n_1, n]$, $\mathcal{H}_2 \in \mathsf{LSeq}[n, n_2]$, s.t. $\mathcal{H}_2 \cdot \mathcal{H}_1 = \mathcal{H}$.

A *lifting assignment* for a $u\ell$-structure U is a map $\mathcal{A}$ from the nodes of U to LSeq s.t.:

(i) $\mathcal{A}(v) \in \mathsf{LSeq}[0, n]$, where n is the level of v;
(ii) $\mathcal{A}(v_2) = \mathcal{A}(v_1)$, if v_1 and v_2 are conclusion/premise nodes of the same multiplicative or identity link (that is, the type of the link is in $\{\mathsf{ax}, \mathsf{cut}, \parr, \otimes\}$);
(iii) $\mathcal{A}(v_2) = \mathcal{H} \cdot \mathcal{A}(v_1)$, for some $\mathcal{H} \in \mathsf{LSeq}[n_1, n_2]$, if v_1 and v_2 are respectively the conclusion and a premise of an exponential link (that is, an ! or a ? link), and n_1 and n_2 are the levels of v_1 and v_2, respectively;
(iv) $\mathcal{A}(v_2) = \mathcal{L}[m, q, a] \cdot \mathcal{A}(v_1)$, if v_1 and v_2 are respectively the (conclusion) premise and (premise) conclusion of a (negative) lift with threshold m, door offset q, and door name a. (The name of a door is an index assigned to the door to distinguish it. The offset q of a door is the difference between the level of its formula and the level of the (de)mux (premise) conclusion; note that $q \geq -1$.)

Let $\mathcal{S}$ be a map from the ! links of a $u\ell$-structure U to LSeq. We say that $\mathcal{S}$ is an *internal state* of U, when $\mathcal{S}(l) \in \mathsf{LSeq}[n, n+1]$, being n the level of the conclusion of l.

Let $\mathcal{A}$ be a lifting assignment for a $u\ell$-structure. To each ! link l whose conclusion is at level n, the assignment associates the lifting sequence $\mathcal{H}_l \in \mathsf{LSeq}[n, n+1]$ s.t. $\mathcal{A}(v_2) = \mathcal{H}_l \cdot \mathcal{A}(v_1)$, where v_2 is the premise of l and v_1 its conclusion (by Fact 21); that is, the assignment corresponds to a unique internal state $\mathcal{S}(l) = \mathcal{H}_l$ to the $u\ell$-structure U. Vice versa, given an internal state $\mathcal{S}$ of the $u\ell$-structure U, we say that U has a solution for $\mathcal{S}$ if there is a lifting assignment (the $\mathcal{S}$-solution of U) s.t., for every ! link l, $\mathcal{A}(v_2) = \mathcal{S}(l) \cdot \mathcal{A}(v_1)$, where v_2 is the premise of l and v_1 its conclusion. Exploiting the fact that a $u\ell$-structure is connected, and that for any lifting assignment $\mathcal{A}$ we have $\mathcal{A}(v) = 1$ when v is a conclusion of U (since $\mathsf{LSeq}[0, 0] = \{1\}$), we see indeed that, for every internal state, there is at most one $\mathcal{S}$-solution. When, moreover, the $u\ell$-structure is correct, this solution exists.

Lemma 22 *A correct $u\ell$-structure has an $\mathcal{S}$-solution for any internal state $\mathcal{S}$.*

PROOF. The proof is by induction on the definition of $u\ell$-structure. In the base case the $u\ell$-structure U is obtained by assigning the boxes to a restricted proof ℓ-net. In the induction case there exists a $u\ell$-structure U' s.t. $U' \rhd_{\mathsf{u}} U$. For the sake of the proof we also prove at the same time that: if two states $\mathcal{S}_1$ and $\mathcal{S}_2$ differ only for their value on the ! link l, then the corresponding solutions coincide on the vertices that are not contained in the box of l.

(base) Let $\mathcal{S}$ be an internal state of U. Let us take the sequence of internal states $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_k = \mathcal{S}$, defined in this way: $\mathcal{S}_i(l) = \mathcal{S}(l)$ if the level of the conclusion of l is lower than i, and $\mathcal{S}_i(l) = 1$ otherwise. Note that this implies, $\mathcal{S}_0 = \mathcal{I}$ (with $\mathcal{I}(l) = 1$, for every l). Since U does not contain lifts, we immediately see that the $\mathcal{I}$-solution is: $\mathcal{A}_0(v) = 1$ for any v. Hence, let B^i_l be the box of an ! link l whose conclusion has level i. We inductively define a sequence of assignments by $\mathcal{A}_{i+1}(v) = \mathcal{S}(l) \cdot \mathcal{A}_i(v)$ if v is in the box B^i_l for some l, and $\mathcal{A}_{i+1}(v) = \mathcal{A}_i(v)$, otherwise. The assignments are well-defined (for two boxes at the same level are disjoint); moreover, every $\mathcal{A}_i$ is an $\mathcal{S}_i$-solution. By inspection of the way in which we get $\mathcal{A}_k = \mathcal{A}$, we conclude.

($U' \rhd_{\pi_{\mathsf{u}}} U$) Let $\mathcal{S}$ be an internal state of U. We show how to build an $\mathcal{S}$-solution $\mathcal{A}$ of U, given an $\mathcal{S}'$-solution $\mathcal{A}'$ of U', where $\mathcal{S}'$ is derived from $\mathcal{S}$. In all the cases $\mathcal{S} = \mathcal{S}'$, but in a duplication involving an ! link. The way in which $\mathcal{A}$ will be defined also proves the independence of $\mathcal{A}(v)$ from the value of $\mathcal{S}(l)$ when v is not in the box of l, provided that the analogous property holds for $\mathcal{A}'$ and $\mathcal{S}'$. We have several cases according to the π rule applied.

(annihilation) Let v_0 be the node between the lifts, and let v_1 and v_2 be the outer premise and conclusion of the pair of lifts. We have $\mathcal{A}'(v_0) = \mathcal{L}[m, q, a] \cdot \mathcal{A}'(v_1) = \mathcal{L}[m, q, a] \cdot \mathcal{A}'(v_2)$, being $\mathcal{L}[m, q, a]$ the triple associated to the facing lifts; then $\mathcal{H} = \mathcal{A}'(v_1) = \mathcal{A}'(v_2)$. We take $\mathcal{A}(v) = \mathcal{A}'(v)$, if v has not been involved in the reduction, and $\mathcal{A}(v) = \mathcal{H}$, if v is the node that replaces the annihilated pair of lifts.

(swap) In this case, $\mathcal{A}'(v_0) = \mathcal{L}[m_1, q_1, a_1] \cdot \mathcal{A}'(v_1) = \mathcal{L}[m_2, q_2, a_2] \cdot \mathcal{A}'(v_2)$, with $m_1 < m_2$. By the properties of the lifting sequences, we see that $\mathcal{A}'(v_0) = \mathcal{L}[m_1, q_1, a_1] \cdot \mathcal{L}[m_2 + q_1, q_2, a_2] \cdot \mathcal{H} = \mathcal{L}[m_2, q_2, a_2] \cdot \mathcal{L}[m_1, q_1, a_1] \cdot \mathcal{H}$, for some $\mathcal{H}$. Thus, if w_0 is the node of U between the swapped lifts, $\mathcal{A}(w_0) = \mathcal{H}$; the other assignments are unchanged.

(duplication) The identity and multiplicative link cases are trivial. Therefore, let us consider the case of an ! link and a lift pointing to its premise—the other exponential link cases being similar. This case and its complementary one in which a demux points the conclusion of an ! link are the only cases in which $\mathcal{S}' \neq \mathcal{S}$, as we will see in the following.

Let us assume that the lift points to the premise v_0 of the ! link l, that v_1 is the premise of the lift, and that v_2 is the conclusion of the ! link. For any $\mathcal{S}'$-solution $\mathcal{A}'$, we have that $\mathcal{A}'(v_0) = \mathcal{L}[m, q, a] \cdot \mathcal{A}'(v_1) = \mathcal{S}'(l) \cdot \mathcal{A}'(v_2)$, with $\mathcal{S}'(l) \in \mathsf{LSeq}[n, n+1]$ and $m < n$. By a simple induction on the length of $\mathcal{S}'(l)$, we see that $\mathcal{S}'(l) \cdot \mathcal{L}[m, q, a] = \mathcal{L}[m, q, a] \cdot \mathcal{S}'(l)^{+q}$, where $\mathcal{H}^{+q}$ means that all the thresholds in $\mathcal{H}$ have been increased by q. Then, $\mathcal{A}'(v_0) = \mathcal{L}[m, q, a] \cdot \mathcal{S}'(l)^{+q} \cdot \mathcal{H} = \mathcal{S}'(l) \cdot \mathcal{L}[m, q, a] \cdot \mathcal{H}$, for some $\mathcal{H}$. As in the swap case, we take $\mathcal{A}(w_0) = \mathcal{H}$ for the conclusion w_0 of the image of l in U, and we leave the other assignments unchanged. The map $\mathcal{A}$ is an $\mathcal{S}$-solution, being $\mathcal{S}$ the internal state that differs from $\mathcal{S}'$ only for its value in l, i.e., $\mathcal{S}(l) = \mathcal{S}'(l)^{+q}$. Since any internal state of U can be obtained in this way, we conclude.

($U' \rhd_{\beta_{\mathsf{u}}} U$) Let us consider the linear case (the ? link is unary), the extension to the general case being trivial. Let l be the ! link involved in the reduction; let w_1, v_1 be the premise and the conclusion of the ? link; and let w_2, v_2 be the premise and the conclusion of l. Let $\mathcal{A}'$ be the $\mathcal{S}'$-solution of U'. At the ? link we have that $\mathcal{A}'(w_1) = \mathcal{S}_? \cdot \mathcal{A}'(v_2)$, for some $\mathcal{S}_? \in \mathsf{LSeq}[n, n+p]$, where n is the level of v_1 and v_2, and $n + p$ is the level of w_1. Let us take the internal state $\mathcal{S}''$ obtained from $\mathcal{S}'$ just changing its value in l, that is, $\mathcal{S}''(l) = \mathcal{L}[n, p-1, a] \cdot \mathcal{S}_?$. We get a new solution $\mathcal{A}''$. By the induction hypothesis, we have that $\mathcal{A}''(v_1) = \mathcal{A}'(v_1)$ and $\mathcal{A}''(w_1) = \mathcal{A}'(w_1)$. Hence, $\mathcal{A}''(w_2) = \mathcal{L}[n, p-1, a] \cdot \mathcal{S}_? \cdot \mathcal{A}''(v_2) = \mathcal{L}[n, p-1, a] \cdot \mathcal{A}''(w_1)$, which justifies the replacement of the ! and ? links in the redex by a lift whose triple is $\mathcal{L}[n, p-1, a]$. $\square$

5.4 Recovering the boxes of a correct uℓ-structure

Let U be a *uℓ*-structure. The internal state $\mathcal{I}$ which associates the empty lifting sequence to each ! link of U (i.e., $\mathcal{I}(l) = 1$, for any l) is the quiescence internal state of U. The corresponding $\mathcal{I}$-solution $\mathcal{Q}$ (if any) is said the *quiescence solution* of U.

Let n be the level of a node v of a correct *uℓ*-structure (thus admitting a quiescence solution) and let $\mathcal{Q}(v) = \mathcal{L}[m_1, q_1, a_1] \cdots \mathcal{L}[m_k, q_k, a_k]$. The *actual level* of v is the sum $n + ||\mathcal{Q}(v)||$. Namely, the actual level of a node v is the level of v increased by the sum of the offsets of the lifting operators that the quiescence solution assigns to v.

Proposition 23 *Let U be a correct $u\ell$-structure. If $\mathcal{R}_l(U)$ is the $s\ell$-structure obtained from the unshared $s\ell$-structure of U by erasing its lifts and by associating to each node its actual level, then $\mathcal{R}_l(U) = \mathcal{R}_u(U)$.*

PROOF. First of all we have to prove that $\mathcal{R}_l(U)$ is well-defined. In fact, let $\mathcal{A}$ be the $\mathcal{S}$-solution of U. We have that $\ell(v) + ||\mathcal{A}(v)|| \geq 0$, for every node v (being $\ell(v)$ or $\ell_U(v)$ the level of the node v in U) and, when the nodes v_1 and v_2 are connected to the same link e:

(i) $\ell(v_1) + ||\mathcal{A}(v_1)|| = \ell(v_2) + ||\mathcal{A}(v_2)||$, when e is a multiplicative or identity link—it follows from $\ell(v_1) = \ell(v_2)$ and $\mathcal{A}(v_1) = \mathcal{A}(v_2)$;

(ii) $\ell(v_1) + ||\mathcal{A}(v_1)|| \leq \ell(v_2) + ||\mathcal{A}(v_2)||$, when v_1 and v_2 are respectively the conclusion and the premise of a ? or of an ! link—it follows from $\ell(v_2) + ||\mathcal{A}(v_2)|| = \ell(v_2) + ||\mathcal{S}|| + ||\mathcal{A}(v_1)||$, with $\mathcal{S} \in \mathsf{LSeq}[\ell(v_1), \ell(v_2)]$, and then $\ell(v_2) + ||\mathcal{A}(v_2)|| \geq \ell(v_1) + ||\mathcal{A}(v_1)||$ (by Fact 21);

(iii) $\ell(v_1) + ||\mathcal{A}(v_1)|| = \ell(v_2) + ||\mathcal{A}(v_2)||$, when v_1 and v_2 are the conclusion and the premise of a (negative) lift—it follows from $\ell(v_1) + ||\mathcal{A}(v_1)|| = \ell(v_1) + ||\mathcal{L}[m, q, a] \cdot \mathcal{A}(v_2)|| = \ell(v_p) + q + ||\mathcal{A}(v_2)|| = \ell(v_2) + ||\mathcal{A}(v_2)||$.

In particular, in the case of the quiescence solution, the previous equations imply that: (i) the actual levels of the multiplicative and identity links are sound; (ii) for any ? link the difference between the actual levels of its premise v_1 and of its conclusion v_2 may differ from $\ell(v_1) - \ell(v_2)$, but it remains positive in any case—the number of boxes closed by a ? link may vary, but cannot become negative; (iii) the actual level of the premise of an ! link is equal to the actual level of its conclusion plus 1; (iv) the actual levels of any premise and conclusion of a (negative) lift coincide. From which we conclude that the definition of $\mathcal{R}_l(U)$ is correct.

The rest of the proof is by induction on the definition of correct $u\ell$-structure.

(base) By hypothesis, $\widetilde{U} = \mathcal{R}_u(U)$ (being $\widetilde{U}$ the net underlying U). The quiescence solution of U is $\mathcal{Q}(v) = 1$, for every node v. Then $\ell(v)$ is the actual level of every node v and $\mathcal{R}_u(U) = \mathcal{R}_l(U)$.

($U' \rhd_{\pi_\mathsf{u}} U$) Immediate, by the definition of the $\mathcal{S}$-solution $\mathcal{A}$ from the $\mathcal{S}'$-solution $\mathcal{A}'$ given in the corresponding case of Lemma 22. In fact, for every v in U which is a copy of a node of U', we see that $\ell_{U'}(v) + ||\mathcal{A}'(v)|| = \ell_U(v) + ||\mathcal{A}(v)||$.

($U' \rhd_{\beta_\mathsf{u}} U$) Let us assume that the ? link $l_?$ involved in the reduction has only one premise only; the extension to the n-ary case is immediate. Let $l_!$ be the ! link involved in the reduction. We have to prove that the actual level of any v contained in the box of $l_!$ is increased by the difference Q between the actual level of $v_?$ (the premise of $l_?$) and the actual level of $v_!$ (the premise of $l_!$). By the proof of Lemma 22, such an actual level can be found by computing the solution $\mathcal{A}$ of U' for the internal state $\mathcal{S}(l) = 1$, when $l \neq l_!$, and $\mathcal{S}(l_!) = \mathcal{L}[n, p-1, a] \cdot \mathcal{S}_?$, where $\mathcal{S}_?$ is imposed by the assignment at the nodes connected to $l_?$, and $\mathcal{L}[n, p-1, a]$ corresponds to the lift inserted by the β_u rule. By easy computation, we see that $Q = ||\mathcal{L}[n, p-1, a] \cdot \mathcal{S}_?||$. Let $\mathcal{S}$ and $\mathcal{S}'$ be internal states that differ only for their value in $l_!$ and let $\mathcal{A}$ and $\mathcal{A}'$ be the respective solutions. Again by inspection of the proof of Lemma 22, we see that:

(i) $||\mathcal{A}(v)|| - ||\mathcal{S}(e_!)|| = ||\mathcal{A}'(v)|| - ||\mathcal{S}'(e_!)||$, when v is in the box $l_!$;

(ii) $||\mathcal{A}(v)|| = ||\mathcal{A}'(v)||$, otherwise.

The second item has been explicitly shown proving Lemma 22. To prove the first item, let us start noticing that, when $\mathcal{S}$ and $\mathcal{S}'$ differ for their values in l_1 and l_2, there exists $\mathcal{S}''$ which differs from $\mathcal{S}$ for its value in l_1 and differs from $\mathcal{S}'$ for its value in l_2. This trivial consideration allows to use the induction of Lemma 22 to see that $||\mathcal{A}'(v)|| = ||\mathcal{A}(v)|| - \delta_1(v)(||\mathcal{S}(l_1)|| - ||\mathcal{S}'(l_1)||) - \delta_2(v)(||\mathcal{S}(l_2)|| - ||\mathcal{S}'(l_2)||)$, where $\delta_i(v) = 1$ if v is in the box of l_i, and $\delta_i(v) = 0$ otherwise, for $i = 1, 2$. Hence, as in our case we have $\mathcal{S}(l_!)$ differing from $\mathcal{I}$ for its value in $l_!$ only, and $||\mathcal{S}(l_!)|| = Q$, we conclude that the actual level of every node in the box of $l_!$ is increased by Q. □

Corollary 24 *If U is a correct $u\ell$-structure with no lifts, then $\mathcal{R}_u(U) = \widetilde{U}$.*

PROOF. The map which associates 1 to each node is the quiescence solution of U. Then, $\widetilde{U} = \mathcal{R}_l(U) = \mathcal{R}_u(U)$. □

This corollary shows the soundness of the approach that uses lifting operators. Indeed, it was not immediate that the boxing computed during the reduction and the one induced by the levels coincide on the result of a computation.

5.5 On the solutions of correct $u\ell$-structures

Before applying the results obtained so far to the unshared reductions, let us summarize some remarks that we can infer from the proofs in the last two subsections.

5.5.1 Scope of a lift

Let us assume that $\mathcal{Q}$ is the quiescence solution of U, that l is a lift whose corresponding triple is $\mathcal{L}[m, q, a]$ and that v is the conclusion of l. We see that $\mathcal{Q}(v)$ contains $\mathcal{L}[m, q, a]$. Since we defined the actual level of a node as the sum of its level plus the offsets of the lifting operators assigned to it, this means that v is in the scope of the reindexing operator corresponding to l. In other words, the offset q of l contributes to determine the actual level of v. More in general, we can say that v is in the scope of a lift l when the triple of l, or a suitable transformation of it, appears in the lifting sequence that the quiescence solution assigns to v. Then, $\mathcal{Q}(v) = \mathcal{L}[m_1, q_1, a_1] \cdots \mathcal{L}[m_k, q_k, a_k]$ expresses that v is in the scope of k reindexing operators. Such an interpretation has a direct correspondence in the fact that after a π rule involving l, the length of the lifting sequence assigned to the image of the conclusion v of l decreases—for v is no longer in the scope of the reindexing operator of l. The latter is the base property that will allow us to prove that the π rules are strongly normalizing (Lemma 25).

5.5.2 *Independence property*

The exponential links are global boundaries for the scope of the reindexing operators: a lift with threshold m is absorbed by a ? link $l_?$ whose conclusion is at level $n \leq m$. Note, that an analogous situation for the ! link is without meaning instead (and will be shown unreachable), for it would correspond to stop reindexing a box at its principal door (note that we have no absorption rule for an ! link). After the execution of a β_u rule involving $l_?$, the boundary corresponding to $l_?$ disappears, and the scope of the lifts that would have been absorbed by $l_?$ spreads over the box of the ! link $l_!$ cut with $l_?$. Then, after the β_u rule, the lifting operators corresponding to those lifts must be assigned to the nodes in the box of $l_!$. The internal states of a $u\ell$-structure model this behavior. If n is the level of the conclusion of $l_!$, the ! link $l_!$ may force an arbitrary reindexing to each node of its box, with the proviso that it has to operate on the levels above n only. (As a consequence, a lift with threshold n cannot reach the premise of $l_!$, since otherwise we would not get a solution for any internal state.) Summarizing, while the behavior of an ! link is independent of the context, the ? links may only erase the lifting operators originated inside the boxes that they close, and the reindexing operators forced by the ! link at the principal doors of such boxes. We remark that this corresponds to the 'property of independence' that Lamping proved in [Lam90] for the sharing graphs implementing the λ-calculus.

5.5.3 *Deadlock-freeness*

The existence of a quiescence solution for a $u\ell$-structure U implies the absence of deadlocks for the reindexing operators. Namely, it is not possible that a (negative) lift gets stuck without the possibility to reindex its (premise) conclusion. In fact, it is not possible to have pairs of facing lifts with the same threshold and different triples; moreover, we have already seen that it is impossible that a lift might be stopped by an ! link. To conclude, let us note that it is indeed impossible to have a lift whose conclusion is a conclusion of U. In fact, by inspection of the rules, we see that: (i) the β_u rule inserts a negative lift with threshold n whose premise is at level $n+1$; (ii) the property '$m < n$, where m is the threshold of a (negative) lift, and n the level of its (premise) conclusion' is invariant under π reduction. Thus, we cannot have a lift pointing to a conclusion of U, for the conclusions of U have level 0. Such a deadlock-freeness is the key property that we will use to prove that the π normal form of a correct $u\ell$-structure is a restricted proof ℓ-net (Lemma 25).

5.6 Properties of the unshared reductions

Lemma 25 *Let U be a correct $u\ell$-structure.*

(i) There is no infinite π reduction of U.
(ii) The restricted proof ℓ-net $\mathcal{R}_u(U)$ is the unique π normal form of U.

PROOF.

(i) Let us consider the following two measures: (a) the sum k_1 of the length of the lifting sequences assigned by the quiescence solution $\mathcal{Q}$ to the conclusion of any logical link of U; (b) the sum k_2 of the length of the lifting sequences assigned by $\mathcal{Q}$ to the principal node of any lift. Any π_{dup} rule decreases k_1 but may increase k_2. All the other π rules decrease at least one of the previous measures. Hence, each π rule decreases the combined measure (k_1, k_2) (w.r.t. the lexicographic order). From which we get that the π rules are strongly normalizing over correct $u\ell$-structures.

(ii) Let $U \rhd_\pi^* U'$. By Lemma 22 both U and U' admit a quiescence solution, and then do not contain deadlocked lifts. As a consequence, any π normal form of U does not contain lifts (since the conclusions of a $u\ell$-structure have always level 0 we cannot have lifts pointing to them). Thus, let N be a normal form of U. We have that $\widetilde{N} = \mathcal{R}_u(N)$ (Corollary 24) and, by the invariance of the read-back under π (Fact 19), $\widetilde{N} = \mathcal{R}_u(U)$. $\square$

Corollary 26 *The reduction rules $\pi + \beta_u$ are strongly normalizing and confluent on correct $u\ell$-structures. The unique $\pi + \beta_u$ normal form of a correct $u\ell$-structure U is the standard normal form of the restricted proof ℓ-net $\mathcal{R}_u(U)$.*

6 Simulation

6.1 Correctness of $s\ell$-structure reduction

We may now simulate π and β reductions of $s\ell$-structures by unshared $u\ell$-structure reductions.

Let us say that a correct $s\ell$-structure G has a *complete unsharing* when:

(i) There exists a correct $u\ell$-structure U s.t. $M : U \preceq G$;
(ii) If $\mathcal{A}$ is a solution of U and $M : U \preceq G$, then $M(v) = M(v')$ and $\mathcal{A}(v) = \mathcal{A}(v')$ implies $v = v'$.

We will also say that U is a *least-shared-instance* of G, written $U \lll G$. The fact that this is the correct notion of unfolding we were looking for will be shown proving the existence of a unique least-shared-instance for any correct $s\ell$-structure (see Corollary 31). For the moment, let us note that such an interpretation is sound at the level of restricted proof ℓ-nets, for a restricted proof ℓ-net has no (proper) less-shared-instances.

Fact 27 *Let N be a restricted proof ℓ-net. If $U \lll N$ or $N \lll U$, then $N = U$.*

PROOF.

($U \lll N$) Since N is a restricted proof ℓ-net, U does not contain lifts. Thus, the quiescence solution $\mathcal{Q}$ of U assigns 1 to each node. By this, we have that $M : U \preceq N$ is injective, for $M(v_1) = M(v_2)$ implies $v_1 = v_2$. Since M is surjective by definition, we conclude $N = U$.

($N \lll U$) Analogous. □

The following simulation properties (Lemma 28 and Lemma 30) show that the $\lll$ is well-behaved w.r.t. the reduction of correct $s\ell$-structures.

Lemma 28 *Let G_0 be a correct $s\ell$-structure and let $U_0 \lll G_0$, for some U_0. For any $G_0 \rhd_\pi G_1$, there exists $U_0 \rhd_\pi^+ U_1$ s.t. $U_1 \lll G_1$.*

PROOF. Let M be the s-morphism between U_0 and G_0 and let r be a redex of G_0. The counterimage $M^{-1}(r)$ of r is a set of redexes that may contain a case of critical pair only: two lifts pointing to the premises of the same ? link. If the redex r is a duplication, the algebraic semantics (remember that U_0 is correct) allows to prove that such two lifts must be equal and then that such a critical pair is confluent. Hence, let us execute in any order the redexes of

U_0 in $M^{-1}(r)$ (closing as previously stated the critical pairs present in it); the result is U_1. The s-morphism between U_1 and G_1 maps every residual of a link v of U_0 into the residual of $M(v)$. $\Box$

As a corollary of the previous lemma, we can lift Lemma 25 to the $s\ell$-structures.

Lemma 29 *Let G be a correct $s\ell$-structure s.t. $U \lll G$, for some U.*

(i) There is no infinite π reduction of G.
(ii) G has a unique π normal form $\mathcal{R}(G) = \mathcal{R}_u(U)$.

PROOF.

(i) By Lemma 25, U strongly normalizes by π reduction to the restricted proof ℓ-net $\mathcal{R}_u(U)$. By Lemma 28, the existence of an infinite π reduction of G would contradict that there are no infinite π reductions of U.
(ii) For any π normal form $\mathcal{R}(G)$, we have $\mathcal{R}_u(U) \lll \mathcal{R}(G)$ (by Lemma 28), and thus $\mathcal{R}_u(U) = \mathcal{R}(G)$ (by Fact 27). $\Box$

The next step is the simulation of the $s\ell$-structure β reduction by a corresponding β_u reduction.

Lemma 30 *Let G_0 be a correct $s\ell$-structure for which there exists U_0 s.t. $U_0 \lll G_0$. For any $G_0 \rhd_\beta G_1$, there exists $U_0 \rhd^+_{\beta_u} U_1$ s.t. $U_1 \lll G_1$.*

Sketch of the proof. Let $M_0 : U_0 \preceq G_0$ and let r be a β redex of G_0. The unshared reduction corresponding to the reduction of r is a development of the set of redexes $M_0^{-1}(r)$ (a development of a set of β redexes of a proof-net is analogous to a development of a set of β redexes for the λ-calculus). The s-morphism M_1 between the $u\ell$-structure U_1 obtained in this way and G_1 maps any residual of a link l of U_0 to the residual of its image $M_0(l)$ (see the detailed proof given in [Gue96] for the λ-calculus case or see [Gue99]). To prove that $M_1(v) = M_1(v')$ and $\mathcal{A}(v) = \mathcal{A}(v')$ implies $v = v'$, note that in the unary case the property holds immediately. In fact, by inspection of the proof of Lemma 22 we see that, if $U_0 \rhd_{\beta_u} U'' \rhd_{\beta_u} U'$, any assignment $\mathcal{A}'$ of U' is obtained from an assignment $\mathcal{A}$ of U_0 and that, for any pair of nodes s.t. $M(v) = M(v')$, it is impossible to have $\mathcal{A}'(v) = \mathcal{A}'(v')$ if $\mathcal{A}(v) \neq \mathcal{A}(v')$. So, let $U_0 \rhd_{\beta_u} U'$ be a reduction involving a k-ary ? link. The principal door of the i-th instance of the duplicated box is replaced by $\mathcal{L}[n, q_i, a_i]$, with $a_i = a_j$ only if $i = j$. Now, let v_i be the i-th instance of the node v; for the s-morphism M' induced by the reduction, we have $M'(v_i) = M'(v)$, for $i = 1, 2, \dots, k$. But, as the lifting sequence $\mathcal{A}'(v_i)$ contains $\mathcal{L}[n, q_i, a_i]$ (again by inspection of the proof of Lemma 22), we conclude that $\mathcal{A}'(v_i) = \mathcal{A}'(v_j)$ iff $i = j$. $\square$

Corollary 31 *Any correct $s\ell$-structure G has a (unique) least-shared-instance.*

PROOF. The existence of a complete unsharing follows from Lemma 28 and Lemma 30. The Uniqueness is irrelevant for the proof of the main theorems (it suffices the result of Fact 27), so for its proof we refer the reader to [Gue99].

6.2 Proofs of the main theorems

Theorem 32 (Theorem 11) *Let G be a correct $s\ell$-structure.*

(i) The π rules are strongly normalizing and confluent on G. The π normal form of G is a restricted proof ℓ-net.
(ii) The $\beta + \pi$ rewriting rules are strongly normalizing and confluent on G. The $\beta + \pi$ normal form of G is a restricted proof ℓ-net.
(iii) The π normal form of G reduces by standard cut-elimination to its $\beta + \pi$ normal form.

PROOF.

(i) By Corollary 31, G has a least-shared-instance U. By Lemma 29 and Lemma 28 we have the strong normalization and that $\mathcal{R}(G) = \mathcal{R}_u(U)$ is a restricted proof ℓ-net ($\mathcal{R}_u(U)$ is a restricted proof ℓ-net by definition).
(ii) Let us assume that $G \rhd_\beta^+ G_1 \rhd_\pi^* G_2$ and that $U \lll G$. Again by Lemma 29 and Lemma 28, plus Lemma 30, there exists a corresponding unshared reduction $U \rhd_{\beta_\mathsf{u}}^+ U_1 \rhd_\pi^* U_2$, s.t. $U_i \lll G_i$, $\mathcal{R}_u(U) = \mathcal{R}(G)$, and $\mathcal{R}_u(U_i) = \mathcal{R}(G_i)$, for $i = 1, 2$. Moreover, as $\mathcal{R}_u(U) \rhd_{\mathsf{std}}^+ \mathcal{R}_u(U_1) = \mathcal{R}_u(U_2)$ (by Fact 19), $\mathcal{R}(G) \rhd_{\mathsf{std}}^+ \mathcal{R}(G_2)$. Thus, let us decompose a reduction of G in an alternating sequence of a non-empty β reduction and of a finite (by Lemma 29) number of π rewritings. Since each element of such a sequence corresponds to a non-empty sequence of β_{std} rewritings, the alternating sequence cannot be infinite, for otherwise we would have an infinite (standard) reduction of a proof-net. Let N be a normal form of G. The confluence of $\beta + \pi$ is shown by proving the uniqueness of N. In fact, by Corollary 26 the unique $\beta_u + \pi$ normal form of U is the standard normal form of $\mathcal{R}_u(U)$, that is, a restricted proof ℓ-net N_u. But by the simulation lemmas, $N_u \lll N$ and then $N = N_u$ (by Fact 27).
(iii) From the last considerations in the previous item. In fact, $\mathcal{R}(G) = \mathcal{R}_u(U)$ and N is the standard normal form of $\mathcal{R}_u(U)$. $\square$

Theorem 33 (13) *Let G be a correct $s\ell$-structure and N be a restricted proof ℓ-net s.t. $N \rhd^* G$. Then $N \rhd^*_{\mathsf{std}} \mathcal{R}(G)$.*

PROOF. By the simulation lemmas (Lemma 29 and Lemma 28), $N \rhd^* U$, where U is the least-shared-instance of G. By Fact 19, $N \rhd^*_{\mathsf{std}} \mathcal{R}_u(U) = \mathcal{R}(G)$. □

Theorem 34 (Theorem 14) *The $\beta + \pi_{\mathsf{opt}}$ rewriting rules are Lévy optimal.*

PROOF. According to the interpretation of the algorithm in terms of brackets and croissants (Remark 2 and Figure 2), we see that the π_{opt} rules correspond to a particular optimal reduction strategy (see Remark 7, also). □

Theorem 35 (Theorem 15) *Let G be a correct $s\ell$-structure and N be its $\beta + \pi$ normal form. Let N_o be a $\beta + \pi_{\mathsf{opt}}$ normal form of G, then $N_o \rhd^*_\pi N$.*

PROOF. By inspection of the π_{opt} rules we see that: if $\mathcal{R}(G)$ contains a β redex r, then there exists $G \rhd^*_{\mathsf{opt}} G'$ s.t. the image of r in G' is a redex. Then, $\mathcal{R}(N_o)$ is the $\beta + \pi$ normal form of G. □

7 Conclusions

We have presented in this paper a solution to the coherence problem for the sharing graph representation of (restricted) proof ℓ-nets and their computations. This result has been made possible by a change in the representation of the nets. As discussed in Remarks 2 and 7, there is a rather simple correspondence between our approach (levels on formulas and only one kind of control nodes—(de)muxes) and the one established in the literature (levels on nodes, two kinds of control nodes—fans and brackets). This shift of notation, however, is crucial and responds to a deep conceptual issue: separating logic from control. The level of a formula, indeed, is a logical information, necessary to ensure not only the correctness of the reduction, but even the static correctness of a net. This has been clear since our previous work on leveled approaches to modal and linear proof theory [MM95,MM96]. In that work, what we have called here the reindexing of a box is a meta-level operation (i.e., 'control'), expressed in a formalism external to the logic itself. The situation is the exact analogous to substitution in first order logic: variables and side-conditions on them are a logical concept; the substitution of a term for a variable is a control operation, necessary during the cut-elimination procedure. In the case of this paper, levels belong to logic (and as such are essentials for the static correctness of a net) and (de)muxes and their reduction rules belong to control. It is this separation to make coherence possible. In the standard approach, instead, logic and control are blurred together. Brackets, fans and indexes represent, depending on context, box nesting (i.e., levels), or logical nodes (the why-not), or control nodes. There is more uniformity of notation, but the price to be paid is the difficulty to recognize in a local way the border of boxes, that is, to eventually guarantee coherence. A different solution is that of the safe reductions of [Asp95a], of which our absorption is a special case.

It remains to address the problem of full proof-nets, where weakening is allowed. Weakening in linear logic can produce boxes whose contents are disconnected. Such boxes can be also generated by the cut-elimination procedure, even starting from proof-nets whose boxes are connected. The crucial case is that of a box whose principal door has as premise a weakening link, and hence it needs a separate component S (that must be a proof-net) to be a valid conclusion of the box. This separate component yields the secondary doors of the box. Now, any attempt to reindex/duplicate the box through its principal door will not reach the disconnected net S. Observe that this problem is shared by all the approaches proposed so far, as any local graph rewriting procedure cannot deal with disconnected components. There is a simple way to bypass this problem, e.g., by restricting the proof-net syntax to generate interaction systems (this means for example to be able to code typed λ-calculus, intuitionistic linear logic and so on). A solution to the general case, however, calls for an extension of the proof-net syntax in order to avoid the formation of disconnected boxes, see [GMM97].

Acknowledgments

We are happy to thank Andrea Asperti, for the many discussions, Laurent Regnier, for many detailed remarks on the subject of this paper, and an anonymous referee, for help in improving presentation.

References

[ACM00] Andrea Asperti, Paolo Coppola, and Simone Martini. (Optimal) Duplication is not elementary recursive. In *Proceedings of the 27th ACM SIGPLAN-SIGACT on Principles of Programming Languages, POPL 2000*, pages 96–107, Boston, Massachusetts, January 2000.

[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.

[AL94] Andrea Asperti and Cosimo Laneve. Interaction systems I: The theory of optimal reductions. *Mathematical Structures in Computer Science*, 4:457–504, 1994.

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

[AM98] Andrea Asperti and Harry G. Mairson. Parallel beta reduction is not elementary recursive. In *Proceedings of the 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '98*, pages 303–315, San Diego, CA, 1998. ACM Press.

[Asp95a] Andrea Asperti. $\delta \circ !\varepsilon = 1$: Optimizing optimal λ-calculus implementations. In Jieh Hsiang, editor, *Rewriting Techniques and Applications, 6th International Conference, RTA-95*, volume 914 of *Lecture Notes in Computer Science*, pages 102–116, Kaiserslautern, Germany, 1995. Springer-Verlag.

[Asp95b] Andrea Asperti. Linear logic, comonads and optimal reductions. *Fundamenta infomaticae*, 22:3–22, 1995.

[GAL92a] Georges Gonthier, Martín Abadi, and Jean-Jacques Lévy. The geometry of optimal lambda reduction. In *Proceedings of Nineteenth Annual ACM Symposyum on Principles of Programming Languages*, pages 15–26, Albequerque, New Mexico, January 1992.

[GAL92b] 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.

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

[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*, pages 181–195, Berlin, 1997. Springer-Verlag. The full paper will appear in TCS.

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

[Gue98] Stefano Guerrini. Sharing-graphs, sharing-morphisms and (optimal) λ-graph reductions. In J. Ginzburg, Z. Khasidashvili, E. Vogel, J. J. Lévy, and E. Vallduví, editors, *The Tbilisi Symposium on Language, Logic, and Computation: Selected Papers*, Studies in Logic, Language, and Computation. CSLI Publications, 1998.

[Gue99] Stefano Guerrini. A general theory of sharing graphs. *Theoretical Computer Science*, 227:99–151, 1999.

[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.

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

[Lév80] Jean-Jacques Lévy. Optimal reductions in the lambda-calculus. In Jonathan P. Seldin and J. Roger Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 159–191. Academic Press, 1980.

[LM96] Julia L. Lawall and Harry G. Mairson. Optimality and inefficiency: What isn't a cost model of the lambda calculus? In *Proceedings of the 1996 ACM SIGPLAN International Conference on Functional Programming*, pages 92–101, Philadelphia, Pennsylvania, 1996.

[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.

[MM96] Simone Martini and Andrea Masini. A computational interpretation of modal proofs. In H. Wansing, editor, *Proof theory of Modal Logics*, pages 213–241. Kluwer, 1996.

[TvD88] Anne S. Troelstra and Dirk van Dalen. *Constructivism in Mathematics, Volume II.* North-Holland, 1988.