Decorating Proof-Nets with Lambda-Terms

Olivier . Laurent @ens-lyon.fr

November 14, 2015

1 Simply Typed λ-Calculus with Pairs

Given a countable set of λ-variables x, y, ..., the *terms* of the λ-calculus (or λ-*terms*) with pairs are:

$$t, u ::= x \mid \lambda x.t \mid t\,u \mid \langle t, u\rangle \mid \pi_1\,t \mid \pi_2\,t$$

where λ is a binder for x in $\lambda x.t$ and, in most cases, λ-terms are considered up to α-renaming of bound variables (denoted $=_\alpha$). In $\lambda x.\_$, x is not considered as an occurrence of x.

We take into account two different notions of substitution in the λ-calculus:

- the traditional capture-avoiding substitution $t[^u/_x]$ where free occurrences of x in t are replaced by u but with α-renaming allowed in t in such a way that no free variable of u becomes bound in $t[^u/_x]$;
- the naive syntactic (capture-allowing) substitution $t\{^u/_x\}$ where free occurrences of x in t are replaced by u without α-renaming in t so that free occurrences of variables in u become bound when x occurs in the scope of binders for these variables.

For example, we have $(\lambda x.x)[^y/_x] = \lambda x.x = (\lambda x.x)\{^y/_x\}$ (no free occurrence of x in $\lambda x.x$) and $(\lambda y.x)[^y/_x] =_\alpha \lambda d.y \neq_\alpha \lambda y.y = (\lambda y.x)\{^y/_x\}$, and also $(\lambda y.t)\{^u/_x\} = \lambda y.(t\{^u/_x\})$ if $x \neq y$.

The dynamics of λ-terms is described through the β-reduction relation, denoted $\to_\beta$, which is the congruence generated by:

$$(\lambda x.t)\,u \to_\beta t[^u/_x]$$
$$\pi_1\,\langle t, u\rangle \to_\beta t$$
$$\pi_2\,\langle t, u\rangle \to_\beta u$$

We assume given a countable set of ground types α, β, ... The simple types of the λ-calculus with pairs are:

$$\tau, \sigma ::= \alpha \mid \tau \to \sigma \mid \tau \times \sigma$$

Typing judgements are of the shape $\Gamma \vdash t : \tau$ where Γ is a finite list of pairs (x, τ) where the xs are distinct λ-variables and the τs are simple types. The typing rules of the simply typed λ-calculus with pairs are:

$$\frac{}{\Gamma, x : \tau, \Delta \vdash x : \tau}\ var \qquad \frac{\Gamma, x : \tau \vdash t : \sigma}{\Gamma \vdash \lambda x.t : \tau \to \sigma}\ abs \qquad \frac{\Gamma \vdash t : \tau \to \sigma \qquad \Gamma \vdash u : \tau}{\Gamma \vdash t\,u : \sigma}\ app$$

$$\frac{\Gamma \vdash t : \tau \qquad \Gamma \vdash u : \sigma}{\Gamma \vdash \langle t, u\rangle : \tau \times \sigma}\ pair \qquad \frac{\Gamma \vdash t : \tau \times \sigma}{\Gamma \vdash \pi_1\,t : \tau}\ proj_1 \qquad \frac{\Gamma \vdash t : \tau \times \sigma}{\Gamma \vdash \pi_2\,t : \sigma}\ proj_2$$

We consider two subsets of the set of λ-terms, the *neutral terms* and the *results*:

$$n ::= x \mid n\,r \mid \pi_1\,n \mid \pi_2\,n$$
$$r, s ::= n \mid \lambda x.r \mid \langle r, s\rangle$$

Lemma 1 (Normal Forms)
A result is a normal form.

PROOF: More generally, neutral terms and results are normal forms. This comes from the fact that λ-abstractions and pairs are not neutral terms, while the argument of a projection or the first argument of an application in this grammar is necessarily a neutral term. □

2 Intuitionistic Linear Logic

2.1 IMELL

Given a countable set of atomic formulas X, Y, ..., formulas of IMELL are given by:

$$A, B ::= X \mid A \otimes B \mid A \multimap B \mid !A$$

Sequents are intuitionistic: $\Gamma \vdash A$ where Γ is a list of formulas, and the rules are:

$$\frac{}{A \vdash A}\, ax \qquad \frac{\Gamma \vdash A \qquad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}\, cut \qquad \frac{\varsigma \text{ permutation} \qquad \Gamma \vdash A}{\varsigma(\Gamma) \vdash A}\, ex$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \otimes R \qquad \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} \otimes L$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \multimap R \qquad \frac{\Gamma \vdash A \qquad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C} \multimap L$$

$$\frac{\Gamma \vdash B}{\Gamma, !A \vdash B}\, !w \qquad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\, !c$$

$$\frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}\, !d \qquad \frac{\Gamma, !!A \vdash B}{\Gamma, !A \vdash B}\, !! \qquad \frac{\Gamma \vdash A}{!\Gamma \vdash !A}\, !f$$

We will often let the (ex) rule implicit in derivations.

2.2 Decorating IMELL

Given a proof π in IMELL, a *decoration* of π is obtained by labelling left-hand side formulas with λ-variables and right-hand side formulas with λ-terms in such a way that, for a sequent $\Gamma \vdash A$, the formulas of Γ are labelled with different λ-variables and rules are decorated as follows:

$$\frac{}{x : A \vdash x : A}\, ax \qquad \frac{\Gamma \vdash u : A \qquad \Delta, x : A \vdash t : B}{\Gamma, \Delta \vdash t[{}^{u}/_{x}] : B}\, cut \qquad \frac{\Gamma \vdash t : A}{\varsigma(\Gamma) \vdash t : A}\, ex$$

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash u : B}{\Gamma, \Delta \vdash \langle t, u \rangle : A \otimes B} \otimes R \qquad \frac{\Gamma, x_1 : A, x_2 : B \vdash t : C}{\Gamma, x : A \otimes B \vdash t[{}^{\pi_1\, x}/_{x_1}, {}^{\pi_2\, x}/_{x_2}] : C} \otimes L$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \multimap B} \multimap R \qquad \frac{\Gamma \vdash u : A \qquad \Delta, x : B \vdash t : C}{\Gamma, \Delta, y : A \multimap B \vdash t[{}^{y\, u}/_{x}] : C} \multimap L$$

$$\frac{\Gamma \vdash t : B}{\Gamma, x : !A \vdash t : B}\, !w \qquad \frac{\Gamma, x_1 : !A, x_2 : !A \vdash t : B}{\Gamma, x : !A \vdash t[{}^{x}/_{x_1}, {}^{x}/_{x_2}] : B}\, !c$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma, x : !A \vdash t : B}\, !d \qquad \frac{\Gamma, x : !!A \vdash t : B}{\Gamma, x : !A \vdash t : B}\, !! \qquad \frac{\Gamma \vdash t : A}{!\Gamma \vdash t : !A}\, !f$$

Lemma 2 (Unique Decoration)
Given a decoration of Γ *by distinct* λ*-variables, there exists a unique decoration (up to renaming of* λ*-variables and* α*-equivalence) of each* IMELL *proof of* $\Gamma \vdash A$*. Moreover, if* $\vec{x} : \Gamma \vdash t : A$ *is such a decoration then* $\vec{y} : \Gamma \vdash t[\vec{y}/\vec{x}] : A$ *as well.*

PROOF: Simple induction on IMELL proofs. In the (*cut*) case for example, one can notice that the choice of x has no impact on the λ-term in the conclusion. □

2.3 Girard's Translation

Given a function $(\_)^\star$ from ground types of the λ-calculus to (atomic) formulas of IMELL, we extend it to arbitrary simple types by *Girard's translation*:

$$(\tau \to \sigma)^\star = !\tau^\star \multimap \sigma^\star \qquad\qquad (\tau \times \sigma)^\star = !\tau^\star \otimes !\sigma^\star$$

The key case is the translation of $\_ \to \_$ where the ! connective allows us to reuse (or forget) the argument of an implication. This corresponds to the presence of structural rules on the left side of sequents in intuitionistic logic.

Proposition 1 (Girard's Translation)
Given a typing derivation π *with conclusion* $\Gamma \vdash t : \tau$ *in the simply typed* λ*-calculus with pairs, there exists a derivation* $\pi^\star$ *in* IMELL *with conclusion* $!\Gamma^\star \vdash \tau^\star$*, whose decoration has conclusion* $!\Gamma^\star \vdash t : \tau^\star$ *(when, given* $x : \sigma$ *in* Γ*,* $!\sigma^\star$ *in* $!\Gamma^\star$ *is labelled with* x*).*

PROOF: We define $\pi^\star$ by induction on π:

$$\frac{}{\Gamma, x : \tau, \Delta \vdash x : \tau}\, var \qquad \mapsto \qquad \dfrac{\dfrac{\dfrac{}{x : \tau^\star \vdash x : \tau^\star}\, ax}{x : !\tau^\star \vdash x : \tau^\star}\, !d}{\vec{x} : !\Gamma^\star, x : !\tau^\star, \vec{y} : !\Delta^\star \vdash x : \tau^\star}\, !w$$

$$\frac{\Gamma, x : \tau \vdash t : \sigma}{\Gamma \vdash \lambda x.t : \tau \to \sigma}\, abs \qquad \mapsto \qquad \frac{\vec{x} : !\Gamma^\star, x : !\tau^\star \vdash t : \sigma^\star}{\vec{x} : !\Gamma^\star \vdash \lambda x.t : !\tau^\star \multimap \sigma^\star} \multimap R$$

$$\frac{\Gamma \vdash t : \tau \to \sigma \qquad \Gamma \vdash u : \tau}{\Gamma \vdash t\, u : \sigma}\, app$$

$$\mapsto \qquad \dfrac{\vec{x_1} : !\Gamma^\star \vdash t[\vec{x_1}/\vec{x}] : !\tau^\star \multimap \sigma^\star \qquad \dfrac{\dfrac{\dfrac{\vec{x_2} : !\Gamma^\star \vdash u[\vec{x_2}/\vec{x}] : \tau^\star}{\vec{x_2} : !!\Gamma^\star \vdash u[\vec{x_2}/\vec{x}] : !\tau^\star}\, !f}{\vec{x_2} : !\Gamma^\star \vdash u[\vec{x_2}/\vec{x}] : !\tau^\star}\, !! \qquad \dfrac{}{x : \sigma^\star \vdash x : \sigma^\star}\, ax}{\vec{x_2} : !\Gamma^\star, y : !\tau^\star \multimap \sigma^\star \vdash y\,(u[\vec{x_2}/\vec{x}]) : \sigma^\star} \multimap L}{\dfrac{\vec{x_1} : !\Gamma^\star, \vec{x_2} : !\Gamma^\star \vdash t[\vec{x_1}/\vec{x}]\, u[\vec{x_2}/\vec{x}] : \sigma^\star}{\vec{x} : !\Gamma^\star \vdash t\, u : \sigma^\star}\, !c}\, cut$$

$$\frac{\Gamma \vdash t : \tau \qquad \Gamma \vdash u : \sigma}{\Gamma \vdash \langle t, u\rangle : \tau \times \sigma}\, pair \qquad \mapsto \qquad \dfrac{\dfrac{\dfrac{\vec{x_1} : !\Gamma^\star \vdash t[\vec{x_1}/\vec{x}] : \tau^\star}{\vec{x_1} : !!\Gamma^\star \vdash t[\vec{x_1}/\vec{x}] : !\tau^\star}\, !f}{\vec{x_1} : !\Gamma^\star \vdash t[\vec{x_1}/\vec{x}] : !\tau^\star}\, !! \qquad \dfrac{\dfrac{\vec{x_2} : !\Gamma^\star \vdash u[\vec{x_2}/\vec{x}] : \sigma^\star}{\vec{x_2} : !!\Gamma^\star \vdash u[\vec{x_2}/\vec{x}] : !\sigma^\star}\, !f}{\vec{x_2} : !\Gamma^\star \vdash u[\vec{x_2}/\vec{x}] : !\sigma^\star}\, !!}{\vec{x_1} : !\Gamma^\star, \vec{x_2} : !\Gamma^\star \vdash \langle t[\vec{x_1}/\vec{x}], u[\vec{x_2}/\vec{x}]\rangle : !\tau^\star \otimes !\sigma^\star} \otimes R}{\vec{x} : !\Gamma^\star \vdash \langle t, u\rangle : !\tau^\star \otimes !\sigma^\star}\, !c$$

$$\frac{\Gamma \vdash t : \tau \times \sigma}{\Gamma \vdash \pi_1\, t : \tau}\, proj_1 \qquad \mapsto \qquad \dfrac{\vec{x} : !\Gamma^\star \vdash t : !\tau^\star \otimes !\sigma^\star \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{x_1 : \tau^\star \vdash x_1 : \tau^\star}\, ax}{x_1 : !\tau^\star \vdash x_1 : \tau^\star}\, !d}{x_1 : !\tau^\star, x_2 : !\sigma^\star \vdash x_1 : \tau^\star}\, !w}{x : !\tau^\star \otimes !\sigma^\star \vdash \pi_1\, x : \tau^\star} \otimes L}{\vec{x} : !\Gamma^\star \vdash \pi_1\, t : \tau^\star}\, cut$$

$$\frac{\Gamma \vdash t : \tau \times \sigma}{\Gamma \vdash \pi_2\, t : \sigma}\, proj_2 \qquad \mapsto \qquad \dfrac{\vec{x} : !\Gamma^\star \vdash t : !\tau^\star \otimes !\sigma^\star \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{x_2 : \sigma^\star \vdash x_2 : \sigma^\star}\, ax}{x_2 : !\sigma^\star \vdash x_2 : \sigma^\star}\, !d}{x_1 : !\tau^\star, x_2 : !\sigma^\star \vdash x_2 : \sigma^\star}\, !w}{x : !\tau^\star \otimes !\sigma^\star \vdash \pi_2\, x : \sigma^\star}\, \otimes L}{\vec{x} : !\Gamma^\star \vdash \pi_2\, t : \sigma^\star}\, cut$$

□

2.4 From IMELL to MELL

Formulas of IMELL can be represented as formulas of MELL by defining the linear implication connective through:

$$A \multimap B = A^\perp \parr B$$

It is natural to distinguish two sub-classes of formulas of MELL: those which correspond to the representation of an IMELL formula and their duals. In this way, the O entry of the following grammar exactly represents the image of IMELL formulas in MELL (and I corresponds to their duals):

$$\begin{array}{rcccccccc} O & ::= & X & | & O \otimes O & | & I \parr O & | & !O \\ I & ::= & X^\perp & | & I \parr I & | & O \otimes I & | & ?I \end{array}$$

Formulas from the O entry are called *output formulas*, those from the I entry are called *input formulas*. Note that A is an output formula (resp. input formula) if and only if $A^\perp$ is an input formula (resp. output formula).

Any proof in IMELL of a sequent $\Gamma \vdash A$ is then translated into a proof of $\vdash \Gamma^\perp, A$ in MELL, where $\Gamma^\perp$ is made of input formulas only and A is seen as an output formula (to make things simpler, we identify $A \multimap B$ and $A^\perp \parr B$).

We can see the converse is true:

Lemma 3 (Conservativity)
If π is a proof of $\vdash \Gamma$ in MELL *which contains output and input formulas only, then $\Gamma = \mathcal{I}, O$ (up to permutation) where $\mathcal{I}$ contains input formulas only, O is an output formula, and π is the translation of an* IMELL *proof of $\mathcal{I}^\perp \vdash O$.*

Proof: Simple induction on π. We can consider for example the case of the $(?w)$ rule:

$$\frac{\begin{array}{c}\pi' \\ \vdash \Gamma\end{array}}{\vdash \Gamma, ?A}\, ?w$$

By assumption $?A$ must be an input or an output formula so that $?A = ?I$ is an input formula. We thus have $\Gamma = \mathcal{I}, O$ by induction hypothesis and we have a proof π'_0 of $\mathcal{I}^\perp \vdash O$ in IMELL such that $\pi' = {\pi'_0}^\star$. We can then build π_0:

$$\frac{\begin{array}{c}\pi'_0 \\ \mathcal{I}^\perp \vdash O\end{array}}{\mathcal{I}^\perp, !I^\perp \vdash O}\, !w$$

with $\Gamma, ?A = \mathcal{I}, ?I, O$, up to permutation, and ${\pi_0}^\star = \pi$.

Another important case is the $(\otimes)$ rule:

$$\frac{\begin{array}{cc}\pi' & \pi'' \\ \vdash \Gamma, A & \vdash \Delta, B\end{array}}{\vdash \Gamma, \Delta, A \otimes B} \otimes$$

Since $A \otimes B$ is either an output formula or an input formula, we must have both A and B output or one of them is output and the other one is input. In the first case, we simply apply twice the induction hypothesis which proves that both Γ and Δ contain input formulas only and we can build:

$$\frac{\Gamma^\perp \vdash A \qquad \Delta^\perp \vdash B}{\Gamma^\perp, \Delta^\perp \vdash A \otimes B} \otimes R$$

If A is an output formula and B is input (the converse being similar), by induction hypothesis, Γ contains input formulas only and $\Delta = \mathcal{I}, O$. We can then build:

$$\frac{\Gamma^\perp \vdash A \qquad \mathcal{I}^\perp, B^\perp \vdash O}{\Gamma^\perp, \mathcal{I}^\perp, A \multimap B^\perp \vdash O} \multimap L$$

where $(A \multimap B^\perp)^\perp = A \otimes B$ is an input formula. □

3 I/O-Proof-Nets

3.1 From IMELL to Proof-Nets

Based on the relation between IMELL and MELL described in Section 2.4, one can consider the restriction of MELL proof-structures where every formula is either an output formula or an input formula. We call them *ι/o-proof-structures*. As a restriction of MELL proof-structures, they come with the notion of DR-acyclicity from the Danos-Regnier correctness criterion. We can define a new *ι/o-orientation* on the edges of these *ι/o*-proof-structures: upwardly if the edge is labelled with an input formula (they are called *input* edges) and downwardly if it is labelled with an output formula (called *output* edges). This leads also to the distinction of two different kinds of $\otimes$-nodes and $\parr$-nodes depending on the associated formulas:

We define a specific correctness condition on *ι/o*-proof-structures.

Definition 1 (*ι/o*-graph)
The *ι/o-graph* of an *ι/o*-proof-structure is the directed graph obtained by forgetting the box structure (keeping only the nodes), disconnecting the input premise (the left premise) of each $\parr^o$-node (thus creating new edges with no source node), and by endowing the edges with the *ι/o*-orientation.

An *ι/o-path* (resp. *ι/o-cycle*) is a path (resp. cycle) in the *ι/o*-graph.

Lemma 4 (*ι/o*-acyclicity)
*If an ι/o-proof-structure is DR-acyclic then its ι/o-graph is a directed acyclic graph (we say that the ι/o-proof-structure is ι/o-*acyclic*).*

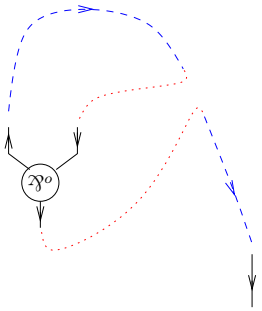
PROOF: Let $\mathcal{R}$ be an ι/o-proof-structure, and let us assume its ι/o-graph contains a directed cycle with respect to the ι/o-orientation. We consider a minimal (with respect to inclusion) such cycle ρ. We focus on the nodes at minimal depth. They are all contained in the same box (or at depth 0). Given a $\parr^\iota$-node or a $?c$-node belonging to ρ, by minimality, ρ contains one incoming edge of the node and one outgoing edge of the node. Thus it cannot contain the two premises of such a node. We can now build a DR-cycle from ρ:

- replace each part of ρ contained in a deeper box by a single node connecting the two doors of the box used for going inside the box and outside the box;
- erase the input premise of each $\parr^o$-node;
- erase, for each $\parr^\iota$-node and $?c$-node, a premise not used by ρ (which must exist as remarked just before).

This gives a DR-cycle thus a contradiction. □

Definition 2 (ι/o-correctness)
An ι/o-proof-structure is *ι/o-correct* (or is an *ι/o-proof-net*) if:

- it has exactly one output conclusion;
- its is DR-acyclic (thus ι/o-acyclic by Lemma 4);
- any ι/o-path, starting from the input premise of a $\parr^o$-node and ending in the output conclusion of the proof-structure, crosses the $\parr^o$-node (from its output premise to its output conclusion).

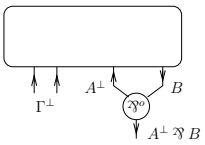


Starting from a proof π of $\Gamma \vdash A$ in IMELL, and by going through MELL, we obtain a proof-structure $\overline{\pi}$ with conclusions $\Gamma^\perp$ and A.

Proposition 2 (Translation)
If π is a proof in IMELL*, $\overline{\pi}$ is an ι/o-proof-net.*

PROOF: By induction on the definition of $\overline{\pi}$. The key case is:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \multimap R \quad \mapsto \quad \frac{\vdash \Gamma^\perp, A^\perp, B}{\vdash \Gamma^\perp, A^\perp \parr B} \parr \quad \mapsto$$

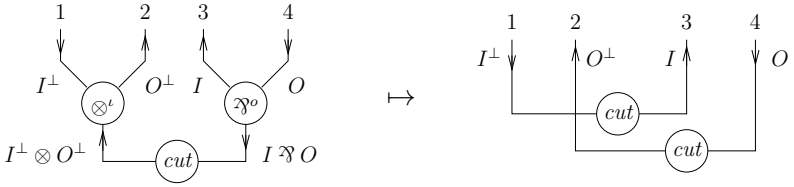


Since any ι/o-path ending in $A^\perp \parr B$ must go through B, the result is immediate. □

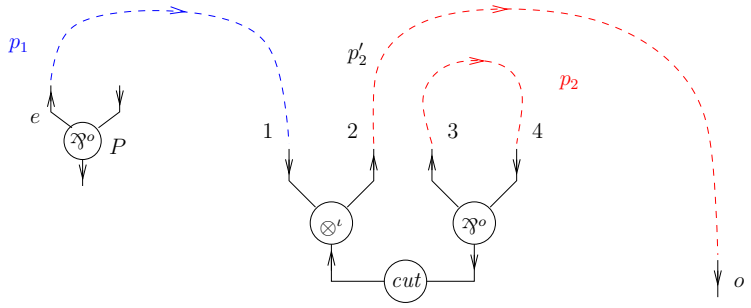
Proposition 3 (Cut Elimination)
If $\mathcal{R}$ is an ι/o-proof-net and if $\mathcal{R}$ reduces to $\mathcal{R}'$ by cut elimination then $\mathcal{R}'$ is an ι/o-proof-net.

PROOF: First notice that the number of output conclusions is not modified by cut elimination. Second, we already know that DR-acyclicity is preserved under cut elimination. It remains to prove the preservation of the third condition of ι/o-correctness.

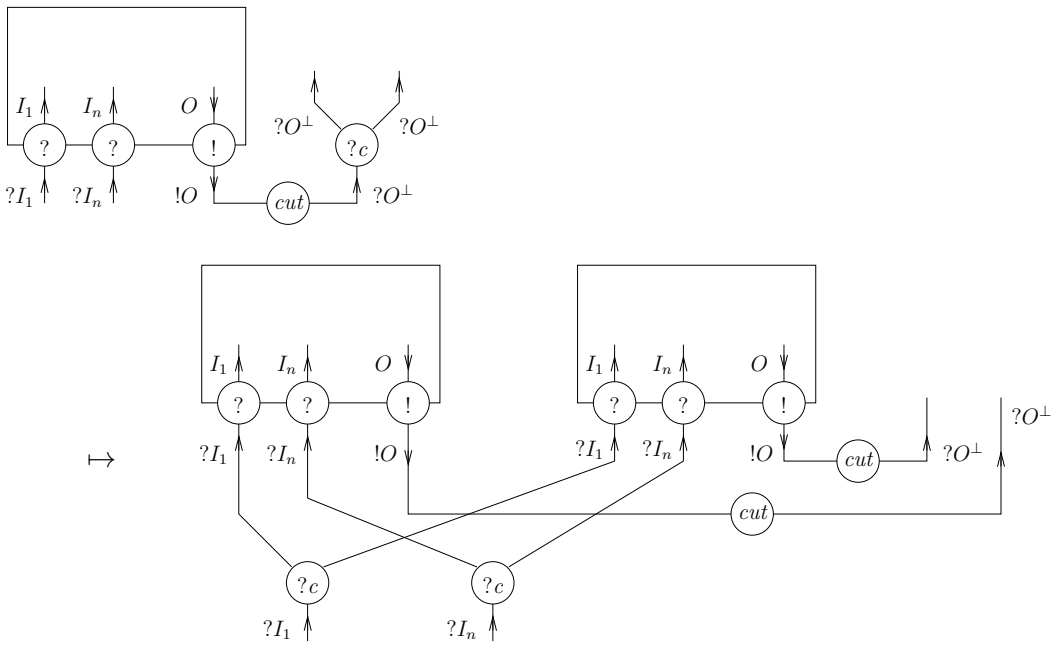
The main case is the following one:



Assume we have an ι/o-path p in $\mathcal{R}'$ from the input premise e of a $\parr^o$-node P to the output conclusion o of $\mathcal{R}'$. We build from it a similar ι/o-path in $\mathcal{R}$. If p uses the cut $2-4$, we modify it into p' in $\mathcal{R}$ which uses the path from 4 to 2 instead of this cut, thus going through P by ι/o-correctness of $\mathcal{R}$ (so that p goes through P as well). If it uses the cut $1-3$ then we can decompose p into a path p_1 from e to 1 and a path p_2 from 3 to o. By ι/o-correctness of $\mathcal{R}$, this path p_2 goes through the $\parr^o$-node with premises 3 and 4 in $\mathcal{R}$ and thus p_2 reaches 4. Let p'_2 be the suffix of p_2 starting from 2. By concatenating p_1 and p'_2, we obtain an ι/o-path in $\mathcal{R}$ which goes from e to o thus, by ι/o-correctness of $\mathcal{R}$, this ι/o-path goes through P, so that P belongs to p (since p_1 and p'_2 are contained in p).



We also consider the $?c$-case.

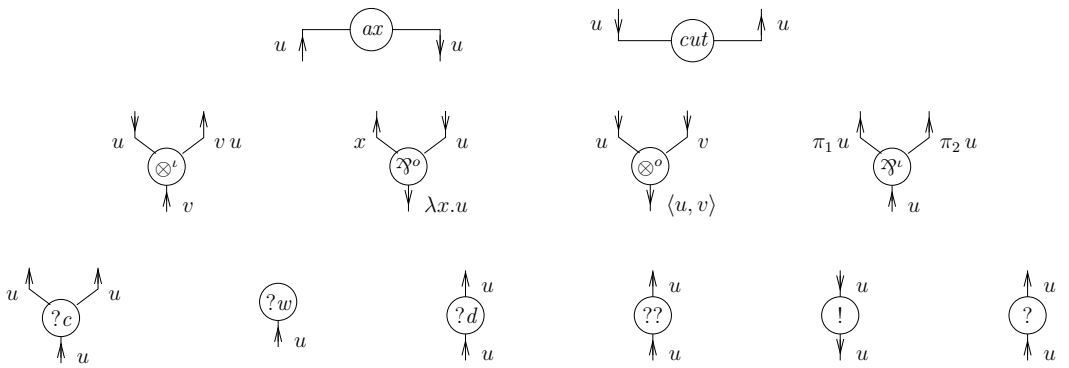


Since main doors are the only output conclusions of boxes, an ι/o-path cannot exit twice the same box without being a cycle. The condition is then preserved for $⅋^o$-nodes outside the copied box since a path which enters copies of the box must exit them to reach the output conclusion of $\mathcal{R}'$ and it cannot enter both copies otherwise this would contradict the ι/o-acyclicity of $\mathcal{R}$. Concerning copied $⅋^o$-nodes, if we have a path p from the input premise of a copy P' of a $⅋^o$-node P to the conclusion of $\mathcal{R}'$, thanks to ι/o-acyclicity of $\mathcal{R}$, it must exit its copy of the box, cannot enter it again, cannot enter the other one, and thus corresponds to a path in $\mathcal{R}$ which must go through P, so that p goes through P'. □

Thanks to the ι/o-acyclicity property of ι/o-proof-nets (Lemma 4), it is possible to define a partial order relation on the edges of such a proof-structure: $e_1 \preccurlyeq e_2$ if there is an ι/o-path from e_1 to e_2. By finiteness of the proof-structures, this partial order is well-founded. The input conclusions and the input premises of $⅋^o$-nodes are the minimal edges. The input conclusions of $?w$-nodes and the unique output conclusion of the ι/o-proof-net are the maximal edges.

3.2 Decorating I/O-Proof-Nets

A *decoration* of an ι/o-proof-structure is a function from edges to λ-terms (except when it is mentioned, λ-terms are not considered up to α-equivalence in decorations) which satisfies the following local constraints:



Proposition 4 (Uniqueness)
Given an ι/o-acyclic ι/o-proof-structure, if we fix a labelling of its input conclusions by λ-terms and a labelling of the input premises of its $⅋^o$-nodes by λ-variables, there exists a unique decoration compatible with this labelling.

Proof: Relying on the ι/o-acyclicity, we can work by induction on the well-founded order $\preccurlyeq$. The input conclusions and the input premises of $⅋^o$-nodes are the minimal edges. We can see that, in the ι/o-graph, the labels of the outgoing edges of a given node are uniquely defined from the labels of its incoming edges (and of the input premise for $⅋^o$-nodes) through the local constraints coming from the definition of decoration. This allows us to apply the induction over the rank in $\preccurlyeq$. □

When the labelling of the input conclusions of an ι/o-proof-net $\mathcal{R}$ and of the input premises of its $⅋^o$-nodes is fixed, we denote by $\underline{\mathcal{R}}$ the λ-term labelling the output conclusion of $\mathcal{R}$ in this unique decoration.

We now prove a few technical properties concerning decorations and $\underline{\mathcal{R}}$ to be used in Propositions 5 and 6.

Lemma 5 (Accessibility)
Given a decorated ι/o-acyclic ι/o-proof-structure $\mathcal{R}$, if x occurs (resp. occurs freely) in the label

of an edge e, there exists an input conclusion or an input premise of $\parr^o$-node c with an ι/o-path p from c to e (thus in particular $c \preccurlyeq e$) such that x occurs (resp. occurs freely) in the label of every edge of p.

PROOF: By induction on $\preccurlyeq$. □

Lemma 6 (Substitution)
Given an ι/o-acyclic ι/o-proof-structure $\mathcal{R}$ with an associated decoration d, if we replace the label v associated by d to an input conclusion c by u, the uniquely generated decoration d' of $\mathcal{R}$ is such that the label of each edge e in d' is the same as its label in d if $c \not\preccurlyeq e$, and can be obtained from its label in d by replacing some sub-terms v with u otherwise.

PROOF: By induction on $\preccurlyeq$. □

Lemma 7 (Variable Substitution)
Given an ι/o-acyclic ι/o-proof-structure $\mathcal{R}$ with an associated decoration d such that a given input conclusion c is labelled with a λ-variable x which does not occur freely in the label of any other input conclusion or input premise of $\parr^o$-node, if we replace the label x by u on c, the uniquely generated decoration d' of $\mathcal{R}$ is such that the label of each edge e in d' is $t\{^u/_x\}$ where t is the label of e in d and $t\{^u/_x\}$ denotes the substitution of free occurrences of x in t by u with possible capture of free occurrences of λ-variables of u (see Section 1).

PROOF: By induction on $\preccurlyeq$ (in fact, by Lemma 5, if $c \not\preccurlyeq e$ then $t\{^u/_x\} = t$).

If e is an input conclusion, we consider the cases $e = c$ and $e \neq c$ and the result is immediate.

If e is an input premise of a $\parr^o$-node, its label is unchanged.

If the e is the output conclusion of a $\parr^o$-node, by induction hypothesis, if the label of its output premise is t in d, it becomes $t\{^u/_x\}$ in d'. Let y be the label of the input premise, the label of e in d' is $\lambda y.(t\{^u/_x\}) = (\lambda y.t)\{^u/_x\}$ since $x \neq y$.

The other cases are similar or easy. □

If $\mathcal{R}$ be an ι/o-proof-net, a decoration of $\mathcal{R}$ is called *simple* if the input conclusions and the input premises of $\parr^o$-nodes are all labelled with distinct λ-variables.

Lemma 8 (Label Building)
Let $\mathcal{R}$ be an ι/o-proof-net and d be a simple decoration of $\mathcal{R}$:

- *if $u\,v$ occurs in the label of an edge e, there exists a $\otimes^\iota$-node with input premise e' labelled $u\,v$ and such that $e' \preccurlyeq e$;*
- *if $\lambda x.t$ occurs in the label of an edge e, there exists a $\parr^o$-node with conclusion e' labelled $\lambda x.t$ and such that $e' \preccurlyeq e$.*

PROOF: By induction on $\preccurlyeq$ using that minimal edges are labelled with variables. □

Lemma 9 (Proof-Net Decoration)
Let $\mathcal{R}$ be an ι/o-proof-net and d be a simple decoration of $\mathcal{R}$, the value of $\underline{\mathcal{R}}$ does not depend, up to α-equivalence, on the labels of the input premises of $\parr^o$-nodes. It only depends on the labelling of the input conclusions by d.

PROOF: Let d_1 and d_2 be two simple decorations of $\mathcal{R}$ with the same labelling of the input conclusions and which differ only on the labelling of one premise c of a $\parr^o$-node P by x_1 and x_2 respectively. If x_1 occurs in $\underline{\mathcal{R}}_1$ (the label of the output conclusion o of $\mathcal{R}$ in d_1) then, by Lemma 5 and by ι/o-correctness, x_1 cannot occur freely in $\underline{\mathcal{R}}_1$ since any path from c to o goes through P and thus x_1 is bound in $\underline{\mathcal{R}}_1$ so that $\underline{\mathcal{R}}_1 =_\alpha \underline{\mathcal{R}}_2$. □

Up to α-equivalence, one can modify the decoration of an IMELL proof in such a way that bound λ-variables and the λ-variables labelling the context in the conclusion are all different.

Proposition 5 (Translation of Decorations)
The translation $\overline{(\_)}$ from IMELL *to ι/o-proof-nets maps such a decoration to a simple decoration and preserves the λ-terms decorating the conclusions.*

PROOF: By induction on the proof in IMELL.

We consider the case where the last rule of the proof is a (*cut*) rule. By induction hypothesis, we have ι/o-proof-nets $\mathcal{R}_1$ and $\mathcal{R}_2$ with simple decorations d_1 and d_2 giving to their conclusions labels $\vec{y}, u$ and $\vec{x}, x, t$. Using Lemma 7, if we replace x by u in d_2, we obtain a decoration d_2' with labels $\vec{x}, u, t\{^u/_x\}$ for the conclusions of $\mathcal{R}_2$ (since $x \notin \vec{x}$). By introducing a *cut*-node between $\mathcal{R}_1$ and $\mathcal{R}_2$ and by using the labels from d_1 and d_2, we build a simple decoration with labels $\vec{y}, \vec{x}, t\{^u/_x\}$ for the conclusions. Since the free λ-variables of u are among $\vec{y}$ and are different from any bound λ-variable of t, we have $t\{^u/_x\} = t[^u/_x]$.

The other cases are similar. □

3.3 Cut Elimination

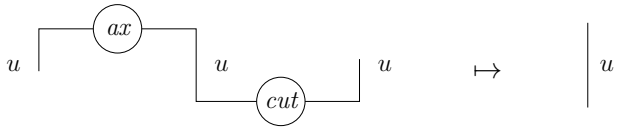
Proposition 6 (Reduction of Decorations)
Let $\mathcal{R}$ be an ι/o-proof-net with a simple decoration, we have:

- *if $\mathcal{R}$ reduces to $\mathcal{R}'$ by an ax-step or an exponential step then $\underline{\mathcal{R}} =_\alpha \underline{\mathcal{R}'}$;*
- *if $\mathcal{R}$ reduces to $\mathcal{R}'$ by a $\otimes/\parr$-step then $\underline{\mathcal{R}} \to_\beta^* \underline{\mathcal{R}'}$;*

where $\underline{\mathcal{R}'}$ is obtained by means of any simple decoration with the same labels on the input conclusions as for the decoration of $\mathcal{R}$.

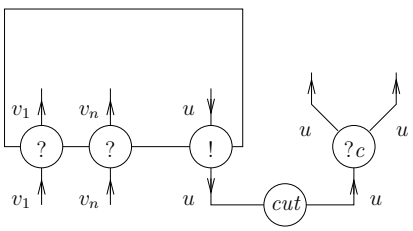
PROOF: Note first that, thanks to Lemma 9, it is meaningful to compare $\underline{\mathcal{R}}$ and $\underline{\mathcal{R}'}$ when we assume the labels of the input conclusions to be the same since the labels of the input premises of $\parr^o$-nodes do not matter up to α-equivalence. We call o the output conclusion of $\mathcal{R}$ and $\mathcal{R}'$.

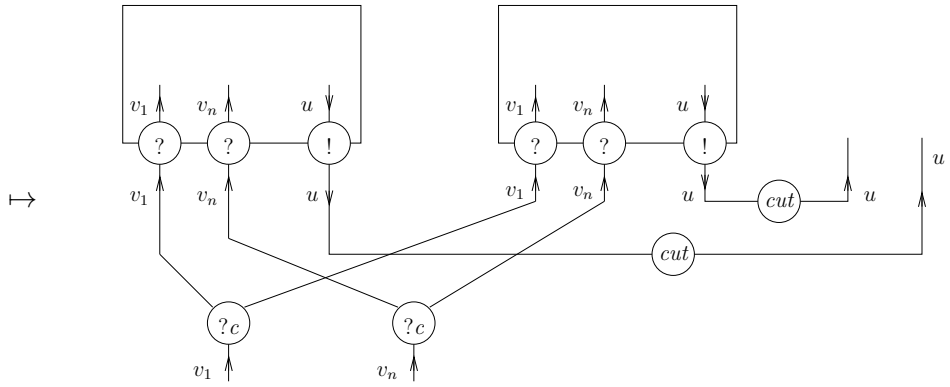
The *ax*-step is immediate:



thus no label (including the label of o) is modified.

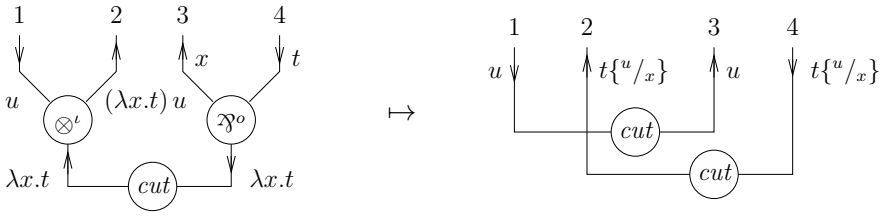
The case of exponential steps has no real impact on the decorations since the λ-terms labelling the edges are all the same around a given exponential node. We just have to check that we can preserve the fact that the decoration is simple. We focus on the $?\,c$-step:





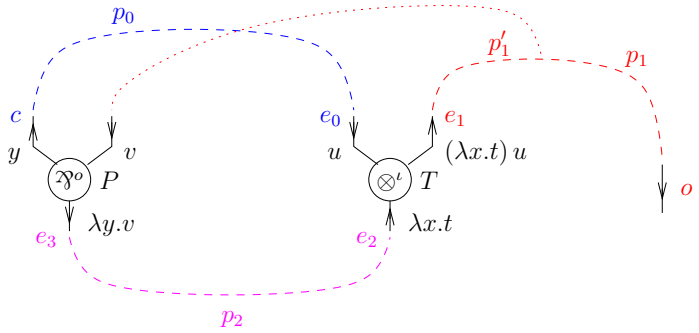
Starting from the decoration d of $\mathcal{R}$, we consider the decoration d' of $\mathcal{R}'$ with the same labels on input conclusions and, for each input premise of $\parr^o$-node, we give it the same label as the one given by d to its unique antecedent in $\mathcal{R}$. Since the minimal edges (with respect to $\preccurlyeq$) of a box are its auxiliary doors and the input premises of its $\parr^o$-nodes, by Proposition 4, the labels of the main doors of the two copies of the box are both equal to the label of the corresponding edge in d. This means that the labels of o in d and d' are the same. We now want to turn d' into a simple decoration. We consider a $\parr^o$-node P of $\mathcal{R}$ which is copied into P_1 and P_2. Let x be the label of the input premise of P in d, we define d'' to be the decoration obtained by labelling the input premise of P_2 with a fresh λ-variable x_2 (the input conclusions and the other input premises of $\parr^o$-nodes keeping the same labels) thanks to Proposition 4. Let e_2 be the output conclusion of the main door of the copy b_2 of the box which contains P_2. Since e_2 is the only output conclusion of b_2, the only edges outside b_2 which may have a modified label are those bigger than e_2 with respect to $\preccurlyeq$. If $e_2 \not\preccurlyeq o$, then the label of o is not modified. If $e_2 \preccurlyeq o$ and if x_2 occurs in the label of e_2, by ι/o-correctness, it is bound. This proves that the labels of e_2 in d' and d'' are α-equivalent, and then that the labels of any edge outside b_2 in d' and d'' are α-equivalent (in particular for o). By renaming this way one of the two copies of each input premise of a $\parr^o$-node, we finally obtain a simple decoration which gives the same label as d' to o up to α-equivalence.

The key case is the $\otimes^\iota/\parr^o$-case:



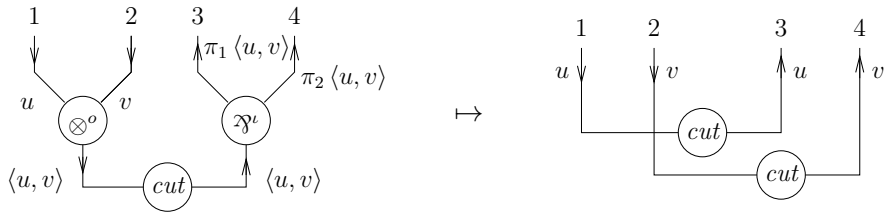
Let us first remark that x is not free in u: otherwise we would have $3 \preccurlyeq 1$ (by Lemma 5 in $\mathcal{R}$) which is impossible by ι/o-acyclicity of $\mathcal{R}'$ (Proposition 3 and Lemma 4). By removing the $\otimes^\iota$-node, the $\parr^o$-node and the *cut*-node in $\mathcal{R}$, we obtain an ι/o-acyclic ι/o-proof-structure $\mathcal{R}_0$ equipped with a decoration d_0 labelling 1 with u, 2 with $(\lambda x.t)\, u$, 3 with x and 4 with t. We replace x by u and we apply Lemma 7 (since x is not free in $(\lambda x.t)\, u$), we obtain a decoration d'_0 of $\mathcal{R}_0$ which labels 1 with $u\{^u/_x\} = u$ (since x is not free in u), 2 with $(\lambda x.t)\, u$ (since x is not free in $(\lambda x.t)\, u$), 3 with u and 4 with $t\{^u/_x\}$. We cannot have x occurring freely in $\underline{\mathcal{R}}$ which is the label of o by ι/o-correctness of $\mathcal{R}$, so that the label of o in d'_0 is $\underline{\mathcal{R}}$. We now apply Lemma 6 by replacing $(\lambda x.t)\, u$ with $t\{^u/_x\}$ in the label of 2 in d'_0 to obtain a decoration d''_0. By ι/o-acyclicity of $\mathcal{R}$, we have $2 \not\preccurlyeq 1$

and $2 \not\preccurlyeq 4$ so that d_0'' labels 1 with u, 2 with $t\{^u/_x\}$, 3 with u and 4 with $t\{^u/_x\}$. Let v be the label of o in d_0'', v is obtained from $\underline{\mathcal{R}}$ by replacing some $(\lambda x.t)\,u$ with $t\{^u/_x\}$. If we add two *cut*-nodes connecting 1 and 3, and 2 and 4 in $\mathcal{R}_0$, we obtain $\mathcal{R}'$ and d_0'' is a simple decoration of $\mathcal{R}'$ compatible with the labelling of d on the input conclusions with $v =_\alpha \underline{\mathcal{R}'}$ (Lemma 9). In order to conclude that $\underline{\mathcal{R}} \to_\beta^* \underline{\mathcal{R}'}$, we prove that if $(\lambda x.t)\,u$ occurs in $\underline{\mathcal{R}}$, $t\{^u/_x\} = t[^u/_x]$. If it is not the case, a free occurrence of a λ-variable y in u should have a binder in t. By Lemma 8 applied twice in $\mathcal{R}$, there exists a $\otimes^\iota$-node T with output premise e_0 labelled u, input premise e_1 labelled $(\lambda x.t)\,u$, conclusion e_2 labelled $\lambda x.t$ with $e_1 \preccurlyeq o$ (we note p_1 an associated ι/o-path from e_1 to o), and a $\parr^o$-node P with input premise c labelled y and conclusion e_3 labelled $\lambda y.v$ with $e_3 \preccurlyeq e_2$ (we note p_2 an associated ι/o-path from e_3 to e_2). By Lemma 5, we have an ι/o-path p_0 from c to e_0, which cannot contain P since y is free in every edge of p_0.



Since $p_0 T p_1$ is an ι/o-path from c to o, by ι/o-correctness, it goes through P thus P belongs to p_1. If we decompose p_1 into $p_1' P p_1''$, we can build an ι/o-cycle $p_2 T p_1' P$ contradicting the ι/o-acyclicity of $\mathcal{R}$.

Finally, in the $\otimes^o/\parr^\iota$-case:



By removing the $\otimes^o$-node, the $\parr^\iota$-node and the *cut*-node in $\mathcal{R}$, we obtain an ι/o-acyclic ι/o-proof-structure $\mathcal{R}_0$ equipped with a decoration d_0 labelling 1 with u, 2 with v, 3 with $\pi_1\langle u,v\rangle$ and 4 with $\pi_2\langle u,v\rangle$. We apply Lemma 6 by replacing $\pi_1\langle u,v\rangle$ with u in the label of 3 in d_0 to obtain a decoration d_0'. By ι/o-acyclicity of $\mathcal{R}$, we have $3 \not\preccurlyeq 1$ and $3 \not\preccurlyeq 2$. This means d_0' labels 1 with u, 2 with v, 3 with u and 4 with $\pi_2\langle u,v\rangle$. We apply Lemma 6 again by replacing $\pi_2\langle u,v\rangle$ with v in the label of 4 in d_0' to obtain a decoration d_0''. d_0'' labels 1 with u, 2 with v, 3 with u and 4 with v. The label t of o in d_0'' is obtained from $\underline{\mathcal{R}}$ by replacing some $\pi_1\langle u,v\rangle$ with u and then some $\pi_2\langle u,v\rangle$ with v thus $\underline{\mathcal{R}} \to_\beta^* t$. If we add two *cut*-nodes connecting 1 and 3, and 2 and 4 in $\mathcal{R}_0$, we obtain $\mathcal{R}'$ and d_0'' is a simple decoration of $\mathcal{R}'$ compatible with the labelling of d on the input conclusions so that $t =_\alpha \underline{\mathcal{R}'}$ (Lemma 9). □

3.4 The λ-Calculus and Proof-Nets

Theorem 1 (Simulation)
If π is a typing derivation with conclusion $\Gamma \vdash t : A$ in the simply typed λ-calculus with pairs,

and if $\overline{\pi^\star}$ reduces to $\mathcal{R}$, then $t \to_\beta^* \underline{\mathcal{R}}$.

PROOF: Putting together Propositions 1 and 5, the ι/o-proof-net $\mathcal{R}_\pi = \overline{\pi^\star}$ is such that $\underline{\mathcal{R}_\pi}$ is t. So that, using Proposition 6, if $\mathcal{R}_\pi$ reduces to $\mathcal{R}$ then $t \to_\beta^* \underline{\mathcal{R}}$. □

Lemma 10 (Normal Forms)
If $\mathcal{R}$ is a cut-free ι/o-proof-net, $\underline{\mathcal{R}}$ is a normal form (in simple decorations).

PROOF: Let d be a simple decoration of $\mathcal{R}$, we can prove by induction on $\preccurlyeq$ that input edges are labelled with neutral terms and output edges are labelled with results. Since $\underline{\mathcal{R}}$ is the label of an output edge, we conclude with Lemma 1. □

This means that, given a typing derivation π with conclusion $\Gamma \vdash t : A$, the normal form of t can be obtained as $\underline{\mathcal{R}}$ where $\mathcal{R}$ is the normal form of $\overline{\pi^\star}$.

$$\begin{array}{ccc} t & \mapsto & \overline{t^\star} \\ {\scriptstyle\beta}\downarrow_* & & \downarrow_* \\ \underline{\mathcal{R}} & \mapsfrom & \mathcal{R} \\ \downarrow\!\!\downarrow & & \downarrow\!\!\downarrow \end{array}$$

Thanks to P. Fermé, A. Grospellier and D. Rouhling for their useful comments and suggestions.