# Decorating Proof-Nets with Lambda-Terms

Olivier . Laurent @ens-lyon.fr

October 26, 2015

## 1 Simply Typed $\lambda$-Calculus with Pairs

Given a countable set of $\lambda$-variables $x$, $y$, ..., the *terms* of the $\lambda$-calculus (or *$\lambda$-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 $\lambda$ is a binder for $x$ in $\lambda x.t$ and $\lambda$-terms are considered up to $\alpha$-renaming of bound variables.

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

$$\begin{aligned}
&(\lambda x.t)\, u \rightarrow_\beta t[{}^{u}/_{x}] \\
&\pi_1\, \langle t, u\rangle \rightarrow_\beta t \\
&\pi_2\, \langle t, u\rangle \rightarrow_\beta u
\end{aligned}$$

We assume given a countable set of ground types $\alpha$, $\beta$, ... The simple types of the $\lambda$-calculus with pairs are:

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

Typing judgements are of the shape $\Gamma \vdash t : \tau$ where $\Gamma$ is a finite partial function from $\lambda$-variables to simple types. The typing rules of the simply typed $\lambda$-calculus with pairs are:

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

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

We consider two subsets of the set of $\lambda$-terms, the *neutral terms* and the *results*:

$$\begin{aligned}
n &::= x \mid n\, r \mid \pi_1\, n \mid \pi_2\, n \\
r, s &::= n \mid \lambda x.r \mid \langle r, s\rangle
\end{aligned}$$

**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 $\lambda$-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. $\square$

## 2 Intuitionistic Linear Logic

### 2.1 IMELL

Given a countable set of atomic formulas $X$, $Y$, ..., formulas of $\mathsf{IMELL}$ are given by:

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

Sequents are intuitionistic: $\Gamma \vdash A$ where $\Gamma$ is a list of formulas.

The rules are:

$$\frac{}{A \vdash A} \qquad \frac{\Gamma \vdash A \qquad \Delta, A \vdash B}{\Gamma, \Delta \vdash B} \qquad \frac{\Gamma \vdash A}{\varsigma(\Gamma) \vdash A}$$

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

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

### 2.2 Decorating IMELL

Given a proof $\pi$ in $\mathsf{IMELL}$, a *decoration* of $\pi$ is obtained by labelling left-hand side formulas with $\lambda$-variables and right-hand side formulas with $\lambda$-terms in such a way that, for a sequent $\Gamma \vdash A$, the formulas of $\Gamma$ are labelled with different $\lambda$-variables and the following rules are satisfied:

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

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

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

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

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

**Lemma 2**
*Given a decoration of $\Gamma$ by $\lambda$-variables, there exists a unique decoration (up to renaming of $\lambda$-variables and $\alpha$-equivalence) of each* $\mathsf{IMELL}$ *proof of $\Gamma \vdash A$.*

Proof: Simple induction on $\mathsf{IMELL}$ proofs. In the (*cut*) case for example, one can notice that the choice of $x$ has no impact on the $\lambda$-term in the conclusion. □

### 2.3 Girard's Translation

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

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

**Proposition 1**
*Given a typing derivation $\pi$ with conclusion $\Gamma \vdash t : \tau$ in the simply typed $\lambda$-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 $x : \tau$ in $\Gamma$, $\tau^\star$ in $\Gamma^\star$ is labelled with $x$).*

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

$$\dfrac{}{\Gamma, x : \tau \vdash x : \tau}\ \mathit{var} \quad\mapsto\quad \dfrac{\dfrac{\dfrac{}{x : \tau^\star \vdash x : \tau^\star}}{x : !\tau^\star \vdash x : \tau^\star}}{\vec{x} : !\Gamma^\star, x : !\tau^\star \vdash x : \tau^\star}$$

$$\dfrac{\Gamma, x : \tau \vdash t : \sigma}{\Gamma \vdash \lambda x.t : \tau \to \sigma}\ \mathit{abs} \quad\mapsto\quad \dfrac{\vec{x} : !\Gamma^\star, x : !\tau^\star \vdash t : \sigma^\star}{\vec{x} : !\Gamma^\star \vdash \lambda x.t : !\tau^\star \multimap \sigma^\star}$$

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

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

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

$$\dfrac{\Gamma \vdash t : \tau \times \sigma}{\Gamma \vdash \pi_1\, t : \tau}\ \mathit{proj}_1 \quad\mapsto\quad \dfrac{\vec{x} : !\Gamma^\star \vdash t : !\tau^\star \otimes !\sigma^\star \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{x_1 : \tau^\star \vdash x_1 : \tau^\star}}{x_1 : !\tau^\star \vdash x_1 : \tau^\star}}{x_1 : !\tau^\star, x_2 : !\sigma^\star \vdash x_1 : \tau^\star}}{x : !\tau^\star \otimes !\sigma^\star \vdash \pi_1\, x : \tau^\star}}{\vec{x} : !\Gamma^\star \vdash \pi_1\, t : \tau^\star}$$

$$\dfrac{\Gamma \vdash t : \tau \times \sigma}{\Gamma \vdash \pi_2\, t : \sigma}\ \mathit{proj}_2 \quad\mapsto\quad \dfrac{\vec{x} : !\Gamma^\star \vdash t : !\tau^\star \otimes !\sigma^\star \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{x_2 : \sigma^\star \vdash x_2 : \sigma^\star}}{x_2 : !\sigma^\star \vdash x_2 : \sigma^\star}}{x_1 : !\tau^\star, x_2 : !\sigma^\star \vdash x_2 : \sigma^\star}}{x : !\tau^\star \otimes !\sigma^\star \vdash \pi_2\, x : \sigma^\star}}{\vec{x} : !\Gamma^\star \vdash \pi_2\, t : \sigma^\star}$$

□

## 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 $\mathsf{IMELL}$ formulas in $\mathsf{MELL}$:

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

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

We can see the converse is true:

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

PROOF: Simple induction on $\pi$. □

# 3 I/O-Proof-Nets

## 3.1 From IMELL to Proof-Nets

Based on the relation between $\mathsf{IMELL}$ and $\mathsf{MELL}$ described in Section 2.4, one can consider the restriction of $\mathsf{MELL}$ proof-structures where every formula is either an $O$ formula or an $I$ formula. We call them *$\iota/o$-proof-structures*. We can define a new *$\iota/o$-orientation* on the edges of these $\iota/o$-proof-structures: upwardly if the edge is labelled with an $I$ formula (they are called *input* edges) and downwardly if it is labelled with an $O$ 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 $\iota/o$-proof-structures.

**Definition 1** ($\iota/o$-graph)
The *$\iota/o$-graph* of an $\iota/o$-proof-structure is the directed graph obtained by forgetting the box structure (keeping only the nodes), disconnecting the input premise of each $\parr^o$-node, and by endowing the edges with the $\iota/o$-orientation.

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

PROOF: Let $\mathcal{R}$ be an $\iota/o$-proof-structure, and let us assume it contains a directed cycle with respect to the $\iota/o$-orientation. We consider a minimal (with respect to inclusion) such cycle. 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 the cycle, by minimality the cycle contains one input edge of the node and one output edge of the node. Thus it cannot contain the two premises of such a node. By replacing each part of the cycle 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, and by erasing the input premise of each $\parr^o$-node, and by erasing for each $\parr^\iota$-node and $?c$-node a premise not used by the cycle (which must exist as remarked just before), one obtains a DR-cycle, a contradiction. □

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

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

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

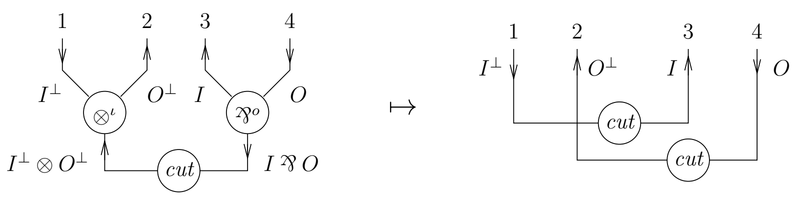
**Proposition 2**
*If $\pi$ is a proof in* $\mathsf{IMELL}$*, $\overline{\pi}$ is an $\iota/o$-proof-net.*

PROOF: By induction on the definition of $\overline{\pi}$. □

**Proposition 3**
*If $\mathcal{R}$ is an $\iota/o$-proof-net and if $\mathcal{R}$ reduces to $\mathcal{R}'$ by cut elimination then $\mathcal{R}'$ is an $\iota/o$-proof-net.*

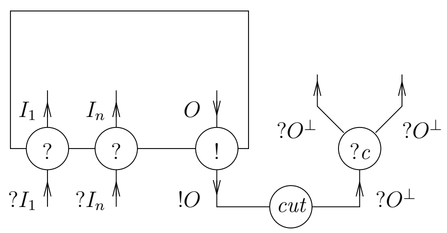
PROOF: First notice that the number of output conclusions is not modified by cut elimination. Second, we already know that DR-acyclicity is preserved under cut elimination. It remains to prove the preservation of the third condition of $\iota/o$-correctness.

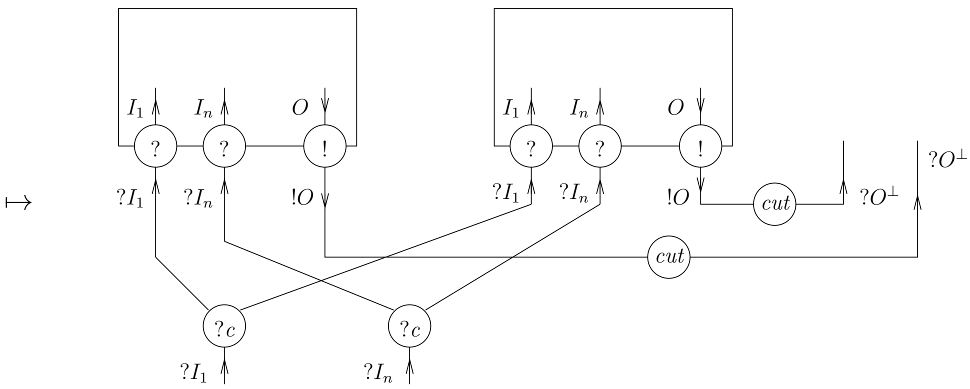
The main case is the following one:




Assume we have a path $p$ in $\mathcal{R}'$ from the input premise $e$ of a $⅋^o$-node $P$ to the output conclusion $c$ of $\mathcal{R}'$. We build from it a similar path in $\mathcal{R}$. If $p$ uses the cut $2-4$, we modify $p$ so that it uses the path from 4 to 2 in $\mathcal{R}$. 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 $c$. By $\iota/o$-correctness of $\mathcal{R}$, this path $p_2$ goes through the $⅋^o$-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 a path in $\mathcal{R}$ which goes from $e$ to $c$ thus, by $\iota/o$-correctness of $\mathcal{R}$, this path goes through $P$, so that $P$ belongs to $p$.

We also consider the $?c$-case.



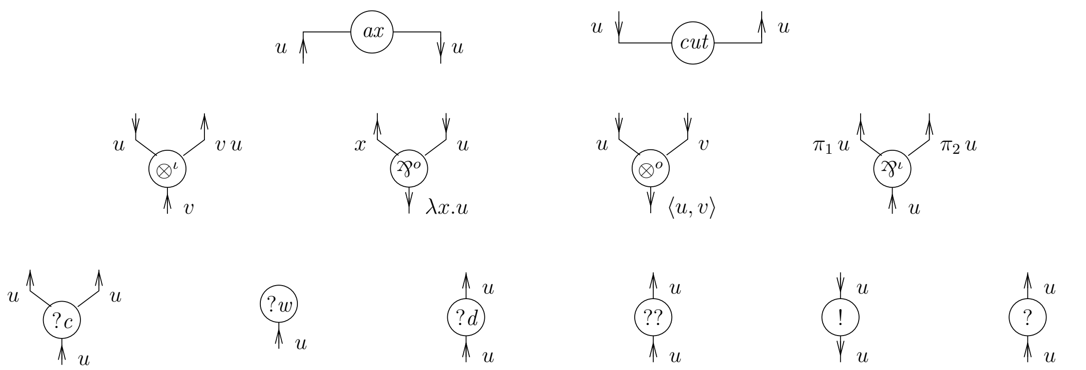


Since main doors are the only output conclusions of boxes, a path in an $\iota/o$-graph cannot exit twice the same box without being a cycle. The condition is then preserved for $\parr^o$-nodes outside the copied box since the considered paths which enter copies of the box must exit them to reach the output conclusion of $\mathcal{R}'$ and cannot enter both copies otherwise this would contradict the $\iota/o$-acyclicity of $\mathcal{R}$. Concerning copied $\parr^o$-nodes, if we have a path $p$ from the input premise of a copy $P'$ of a $\parr^o$-node $P$ to the conclusion 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 $P'$. $\square$

Thanks to the $\iota/o$-acyclicity property of $\iota/o$-proof-nets (Lemma 4), it is possible to define a partial order relation on the nodes (and edges) of such a proof-structure: $n_1 \preccurlyeq n_2$ if there is a path from $n_1$ to $n_2$ in its $\iota/o$-graph (and similarly for edges). By finiteness of the proof-structures, this partial order is well-founded. The input conclusions and the input premises of $\parr^o$-nodes are the minimal edges. The input conclusions of $?w$-nodes and the unique output conclusion of the $\iota/o$-proof-net are the maximal edges.

## 3.2 Decorating I/O-Proof-Nets

A *decoration* of an $\iota/o$-proof-structure is a function from edges to $\lambda$-terms which satisfies the following local constraints:



**Proposition 4**
*Given an $\iota/o$-acyclic $\iota/o$-proof-structure, if we fix a labelling of its input edges by $\lambda$-terms and a labelling of the input premises of its $\parr^o$-nodes by $\lambda$-variables, there exists a unique decoration compatible with this labelling. Moreover the label of an edge only depends on the labels of the edges which are smaller with respect to $\preccurlyeq$.*

Proof: Relying on the $\iota/o$-acyclicity, we can work by induction on the well-founded order $\preccurlyeq$.

The input conclusions and the input premises of $\parr^o$-nodes are the minimal edges. We can see that, in the $\iota/o$-graph, the labels of the outgoing edges of a given node are uniquely defined from the labels of its incoming edges through the local constraints coming from the definition of decoration. This allows us to apply the induction over $\preccurlyeq$. $\square$

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

**Lemma 5** (Substitution)
*Given an $\iota/o$-acyclic $\iota/o$-proof-structure $\mathcal{R}$ with an associated decoration $d$, if we replace the label $v$ associated by $d$ to an input conclusion $c$ by $u$ (with the free $\lambda$-variables of $u$ included in those of $v$), 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 $\preccurlyeq$. $\square$

**Lemma 6** (Accessibility)
*Given a decorated $\iota/o$-acyclic $\iota/o$-proof-structure $\mathcal{R}$, the $\lambda$-variable $x$ occurs (freely or not) in the label of an edge $e$ if and only if $x$ occurs in the label of an input conclusion or of an input premise of $\parr^o$-node $c$ such that $c \preccurlyeq e$. If $x$ occurs freely in the label of an edge $e$, $x$ occurs freely in the label of an input conclusion or of an input premise of $\parr^o$-node $c$ such that $c \preccurlyeq e$.*

Proof: By induction on $\preccurlyeq$. $\square$

**Lemma 7** (Variable Substitution)
*Given an $\iota/o$-acyclic $\iota/o$-proof-structure $\mathcal{R}$ with an associated decoration $d$ such that a given input conclusion $c$ is labelled with a $\lambda$-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 $\lambda$-variables of $u$.*

Proof: By induction on $\preccurlyeq$ (in fact, by Lemma 6, 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. $\square$

If $\mathcal{R}$ be an $\iota/o$-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 $\lambda$-variables.

**Lemma 8** (Proof-Net Decoration)
*Let $\mathcal{R}$ be an $\iota/o$-proof-net and let us fix a labelling of its input conclusions with distinct $\lambda$-variables, the value of $\underline{\mathcal{R}}$ does not depend on the labels of the input premises of $\parr^o$-nodes in a simple decoration (up to $\alpha$-equivalence).*

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 6 and by $\iota/o$-correctness, any path from $c$ to $o$ goes through $P$ and thus $x_1$ is bound in $\underline{\mathcal{R}}_1$ so that $\underline{\mathcal{R}}_1 =_\alpha \underline{\mathcal{R}}_2$. □

**Lemma 9**
*If $t$ is the label of an edge in a simple decoration of an $\iota/o$-proof-net, $t$ cannot contain a binding $\lambda x.\_$ in the scope of another binding $\lambda x.\_$ for the same $\lambda$-variable.*

Proof: Since there is no $\lambda$ in the labels of the minimal edges, the only place where $\lambda x.\_$ can be introduced is through the unique $\parr^o$-node $P$ whose input premise is labelled $x$. But if the label of the output premise of $P$ contains some $\lambda x.\_$, we have a contradiction with Lemma 4. □

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

**Proposition 5**
*The translation $\overline{(\_)}$ from IMELL to $\iota/o$-proof-nets maps such a decoration to a simple decoration and preserves the $\lambda$-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 $\iota/o$-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 $\lambda$-variables of $u$ are among $\vec{y}$ and are different from any bound $\lambda$-variable of $t$, we have $t\{^u/_x\} = t[^u/_x]$.

The other cases are similar. □

### 3.3 Cut Elimination

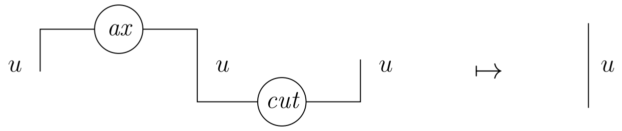
**Proposition 6**
*Let $\mathcal{R}$ be an $\iota/o$-proof-net with a simple decoration, we have:*

- *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 a simple decoration with the same labels on the input conclusions as for the decoration of $\mathcal{R}$.*

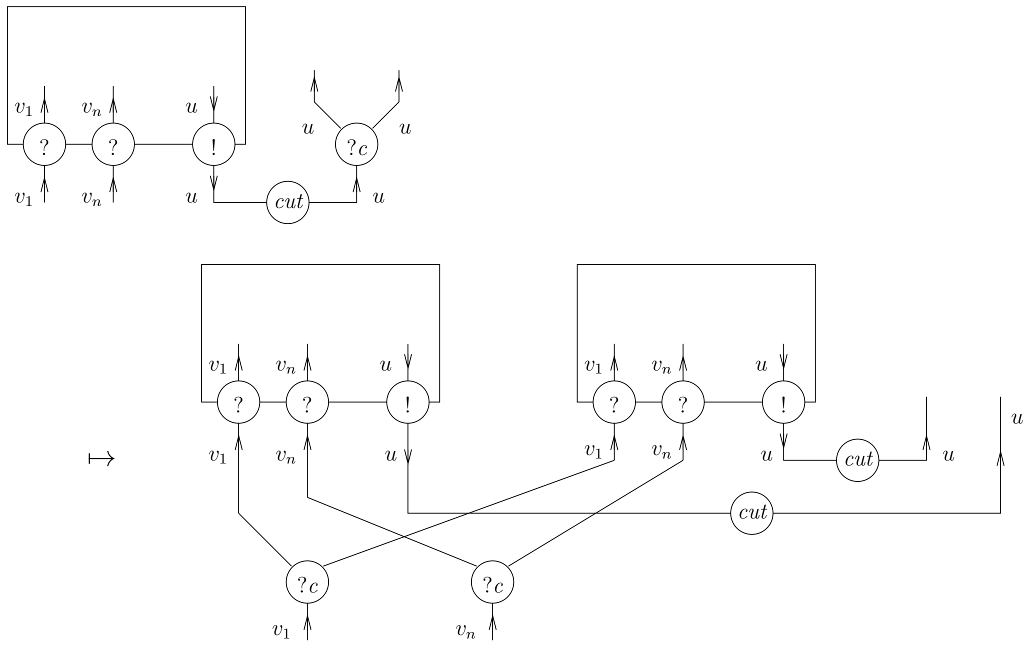
Proof: Note first that, thanks to Lemma 8, it is meaningful to compare $\underline{\mathcal{R}}$ and $\underline{\mathcal{R}'}$ when we assume the labels of the input conclusions to be the same since the labels of the input premises of $\parr$-nodes do not matter. We call $o$ the output conclusion of $\mathcal{R}$ and $\mathcal{R}'$.

The *ax*-step is immediate:



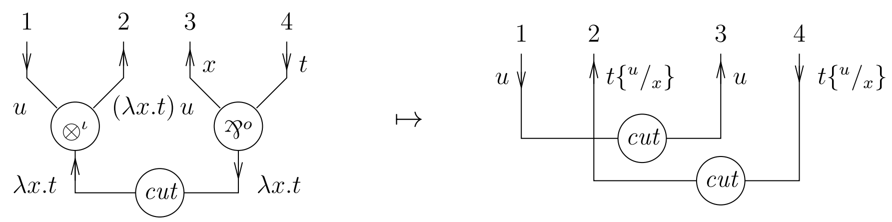
thus no label (including the label of $o$) is modified.

The case of exponential steps has no real impact on the decorations since the $\lambda$-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:



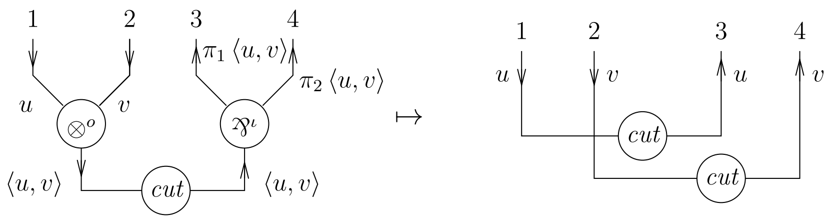
Starting from the decoration $d$ of $\mathcal{R}$, we consider the decoration $d'$ of $\mathcal{R}'$ with the same labels on input conclusions and, for each input premise of $\parr^o$-node, we give it the same label as the one given by $d$ to its unique antecedent in $\mathcal{R}$. Since the minimal edges (with respect to $\preccurlyeq$) of a box are its auxiliary doors, 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 $d''$. 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 $\lambda$-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 its label is not modified. If $e_2 \preccurlyeq o$ and if $x_2$ occurs in the label of $e_2$, by $\iota/o$-correctness, it is bound. This proves that the labels of $e_2$ in $d'$ and $d''$ are $\alpha$-equivalent, and then that the labels of any edge outside $b_2$ in $d'$ and $d''$ are $\alpha$-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 $\alpha$-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 6 in $\mathcal{R}$) which is impossible by $\iota/o$-acyclicity of $\mathcal{R}'$. By removing the $\otimes^\iota$-node, the $\parr^o$-node and the *cut*-node in $\mathcal{R}$, we obtain an $\iota/o$-acyclic $\iota/o$-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 $\iota/o$-correctness of $\mathcal{R}$, so that the label of $o$ in $d_0'$ is $\underline{\mathcal{R}}$. We now apply Lemma 5 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 $\iota/o$-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 so that $v = \underline{\mathcal{R}'}$ (Proposition 4 and Lemma 8). 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 $\lambda$-variable $y$ in $u$ should have a binder in $t$. But $y$ cannot be free in $\underline{\mathcal{R}}$ by Lemma 6 and $\iota/o$-correctness of $\mathcal{R}$, so that we must have $\underline{\mathcal{R}} = \ldots \lambda y.(\ldots(\lambda x.t)\,u \ldots)\ldots$ with $\lambda y.\_$ occurring in $t$ which contradicts Lemma 9.

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 $\iota/o$-acyclic $\iota/o$-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 5 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 $\iota/o$-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 5 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 $t = \underline{\mathcal{R}'}$ (Proposition 4 and Lemma 8). □

### 3.4 The $\lambda$-Calculus and Proof-Nets

**Theorem 1**
*If $\pi$ is a typing derivation with conclusion $\Gamma \vdash t : A$ in the simply typed $\lambda$-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 $\iota/o$-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 $\iota/o$-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. $\square$

This means that, given a typing derivation $\pi$ 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 $\overline{\pi^\star}$.

$$\begin{array}{ccc}
t & \mapsto & \overline{t^\star} \\
{}_\beta \downarrow_* & & \downarrow_* \\
\underline{\mathcal{R}} & \leftarrow\!\shortmid & \mathcal{R} \\
\not\downarrow & & \not\downarrow
\end{array}$$