Electronic Notes in Theoretical Computer Science 50 No. 2 (2001) – Proc. BOTH 2001
URL: http://www.elsevier.nl/locate/entcs/volume50.html 15 pages

On intuitionistic proof nets with additional rewrite rules and their approximations

Satoshi Matsuoka

Department of Electrical and Computer Engineering, Faculty of Engineering, Nagoya Institute of Technology, Gokiso, Showa-Ku, 466-8555, Japan.

Abstract

First we present a proof nets system with eight additional rewrite rules, which concerns ordering of introductions of exponential-links and are only applied to normal forms of proof nets in the usual sense. We show that the reduction relation generated by these eight rewrite rules is strong normalizing and confluent. Second we propose an simply judged equality on intuitionistic proof nets based on the notion of the main path of an intuitionistic proof net. The notion is an analogue of Böhm-trees in λ-calculus.

1 Introduction

The notion of proof nets has been introduced in [Gir87]. The proof nets are a "complete" representation of the notion of proofs of Linear Logic, which means that any proof of Linear Logic has the corresponding proof net and any proof net the corresponding proofs of Linear Logic. Proof nets also has a clean operational semantics based on graph rewriting reduction. In the multiplicative exponential fragment of proof nets, the graph rewriting reduction is strong normalizing [Gir87] and confluent [Laf95]. Proof nets can thus be a computational system.

But the equality based on the graph rewriting reduction in [Laf95] makes unnecessary distinctions between normal proof nets. The distinctions are due to ordering of introductions of exponential links in proof nets. The graph rewriting system in [DK97] introduces three rewrite rules called cw, fusion, and push in order to overcome the defect. However for our purpose, i.e., higher order pre-unification on intuitionistic proof nets [Mat00a,Mat00b,Mat01a], the three rewrite rules are not sufficient: higher order pre-unification algorithm, which is the central part of higher order programming languages like [NM88], is constructed based on the notion of approximations to terms like Böhm-trees (see [SG89]). But in the system in [DK97] the notion of approximations to intuitionistic proof nets is complex, it is hopeless to construct a simple unification algorithm based on the notion. For example, in [DK97] the left side of Figure 1 is a normal proof net and the right side of Figure 1 is the approximation to the normal proof net, which is a subgraph obtained by eliminating some sub-proof-nets of the normal proof net. But such a complex combination of c-links and d-links makes it difficult to define an equality on such approximations. Moreover !-boxes make the situation worse.

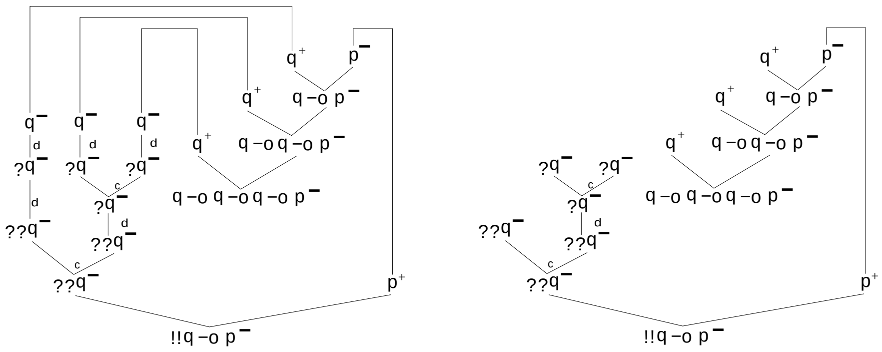


Fig. 1. a normal proof net in Di Cosmo and Kesner's system and its approximation

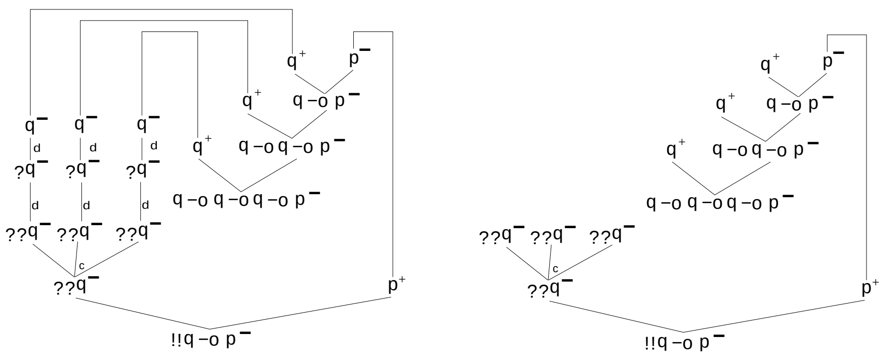


Fig. 2. a normal proof net in our system and its approximation

In this paper, we present a graph rewriting system on proof nets, which retains cw and fusion rewrite rules and adds six rewrite rules. The basic idea of the system is to lift d-links inside !-boxes and to push out c-links and w-links outside !-boxes. In our system, the left side of Figure 1 is no more normal form and it reduces to the left side of Figure 2. The approximation to the proof net is the right side of Figure 2 and does not include any d-links. This makes it simple to define an equality on the approximations to normal proof nets. These rewrite rules are only applied to normal forms of proof nets in the usual sense because when these rewrite rules are incorporated into the standard rewrite rules of Linear Logic, the system violates the Church-Rosser property. But we can show that the reduction relation generated by only these rewrite rules is strong normalizing and confluent. Moreover, we propose the notion of an equality based on the rewriting rules. We have succeeded to construct a simple higher order pre-unification algorithm on intuitionistic proof nets based on the equality (see [Mat01a]). The main features of the higher order pre-unification algorithm are twofold: First one can reason about terms with sharing on higher-order compound terms as well as constants and variables in the same spirit as *Optimal Reduction* on *lambda*-calculus. Second, one may reduce the number of partial bindings, which are approximations of solutions, occurring in a sequence of transformations of equations on higher-order terms.

2 IIMELL proof nets

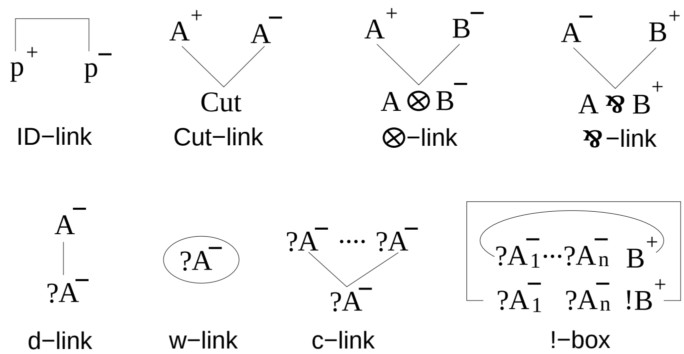
In this section, we introduce proof nets for implicational intuitionistic multiplicative exponential fragment of Linear Logic (for short IIMELL).

Definition 2.1 (MELL formulas) *MELL formulas (or simply formulas) (F) are inductively constructed from atomic formulas (P) and logical connectives:*

- $P \equiv p|q|r|\cdots$
- $F \equiv P|F \otimes F|F \parr F|!F|?F.$

Definition 2.2 (polarized formulas) *An IIMELL formula is a pair* $\langle A, pl\rangle$ *where* A *is a MELL formula and* pl *is an element of* $\{+, -\}$. $+$ *and* $-$ *are called Danos-Regnier polarity. A formula* $\langle A, pl\rangle$ *can be written as* A^{pl}. *An IIMELL formula is called polarized formula. A formula with* $+$ *(resp.* $-$*) polarity is called* $+$*-formula or positive formula (resp.* $-$*-formula or negative formula).*

In the following we list the links which we use in this paper. We call these links *IIMELL links.*



In IMELL links above,

(i) each of p^+ and p^- of ID-link is called a conclusion of the link;

(ii) each of A^+ and A^- of Cut-link is called a premise of the link;

(iii) each of A^+ and B^- of $\otimes$-link is called a premise of the link and $A \otimes B^-$ the conclusion of the link;

(iv) each of A^- and B^+ of $\parr$-link is called a premise of the link and $A \parr B^+$ the conclusion of the link;

(v) A^- of d-link is called a premise of the link and $?A^-$ the conclusion of the link;

(vi) $?A^-$ of w-link is called the conclusion of the link;

(vii) each $?A^-$ in the upper formula occurrences $?A^-, \ldots, ?A^-$ of c-link is called a premise of the link and the lower formula occurrence $?A^-$ is called the conclusion of the link;

(viii) each of conclusion formula occurrences $?A_1^-, \ldots, ?A_n^-, B^+$ of the maximal proof net inside !-box is called a premise of the !-box and each of formula occurrences $?A_1^-, \ldots, ?A_n^-, B^+$ of !-box is called a conclusion of the !-box.

For simplicity we restrict conclusions of ID-links to atomic formulas. Figure 3 shows the definition of *IIMELL proof nets*: the definition of IIMELL proof nets is inductively defined. In an IIMELL proof net Θ, an IIMELL formula that is not a premise of some link is called a *conclusion formula* of Θ. Any IIMELL proof net has exactly one positive conclusion formula. Figure 4, Figure 5

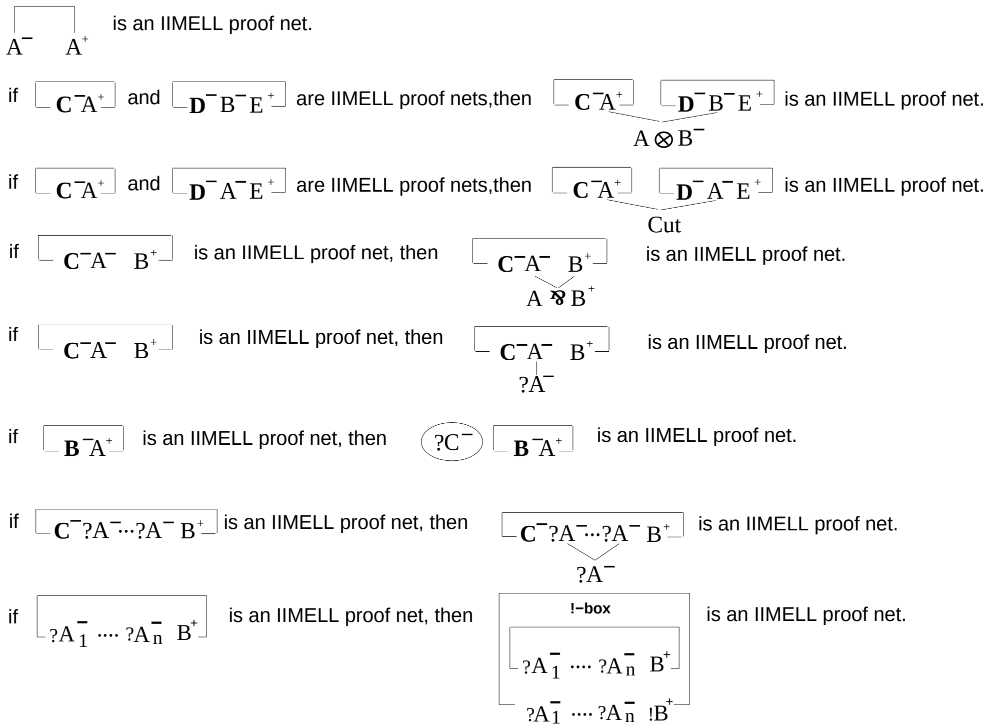


Fig. 3. the definition of IIMELL proof nets

and Figure 6 show ID, multiplicative, dereliction, contraction, weakening and of-course rewrite rules. We call the reduction relation defined by these six rewrite rules *the standard reduction*, which is denoted by $\twoheadrightarrow_{\mathrm{std}}$. The one-step reduction of $\twoheadrightarrow_{\mathrm{std}}$ is denoted by $\rightarrow_{\mathrm{std}}$. Figure 7 and Figure 8 show c-d, d-w, c-w, c-del, fusion, c-outside, w-outside, and d-inside rewrite rules. c-w and fusion rewrite rules have been introduced in [DK97]. The rest are new. We define $\twoheadrightarrow_{\mathrm{ex}}$ as the reduction relation on normal proof nets of $\rightarrow_{\mathrm{std}}$ generated by these eight rewrite rules, which is called *the extended reduction*. This definition is well-defined because these eight rewrite rules never create

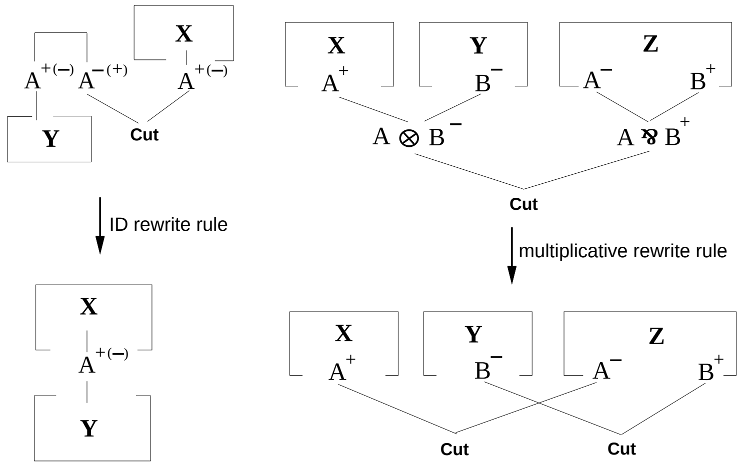


Fig. 4. ID and multiplicative rewrite rules

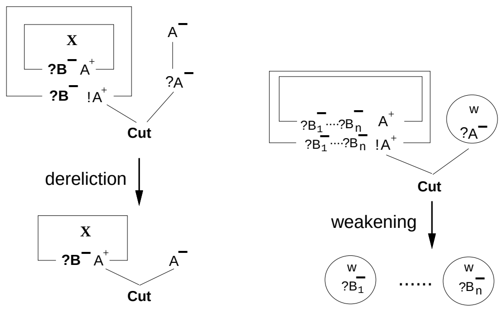


Fig. 5. exponential (dereliction and weakening) rewrite rules

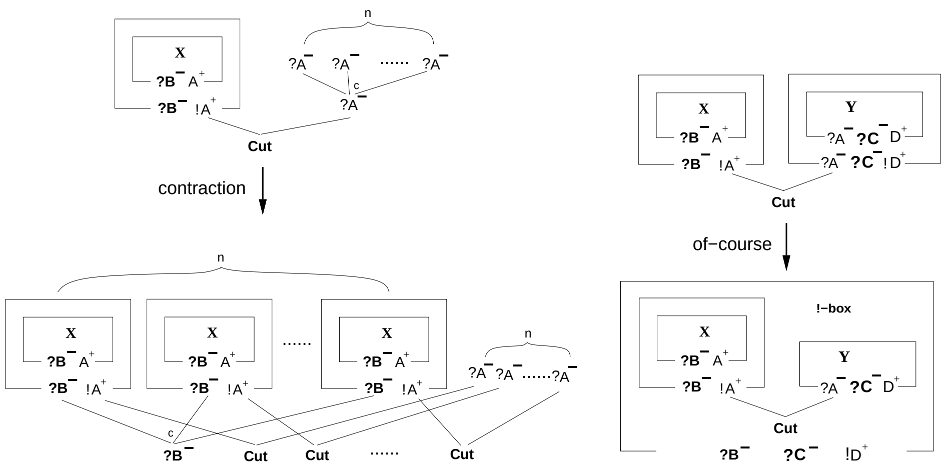


Fig. 6. Exponential rewrite rules (contraction and of-course)

any Cut-links. The one-step reduction of $\twoheadrightarrow_{\mathrm{ex}}$ is denoted by $\rightarrow_{\mathrm{ex}}$.

Proposition 2.3 *Let Θ_1 be an IIMELL proof net. If $\Theta_1 \rightarrow_{\mathrm{ex}} \Theta_2$, then Θ_2 is also an IIMELL proof net.*

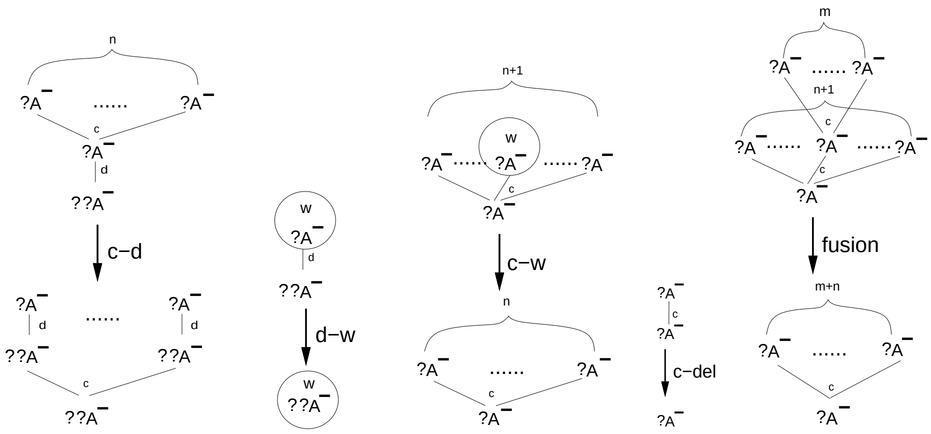


Fig. 7. c-d, d-w, c-w, c-del, and fusion reductions

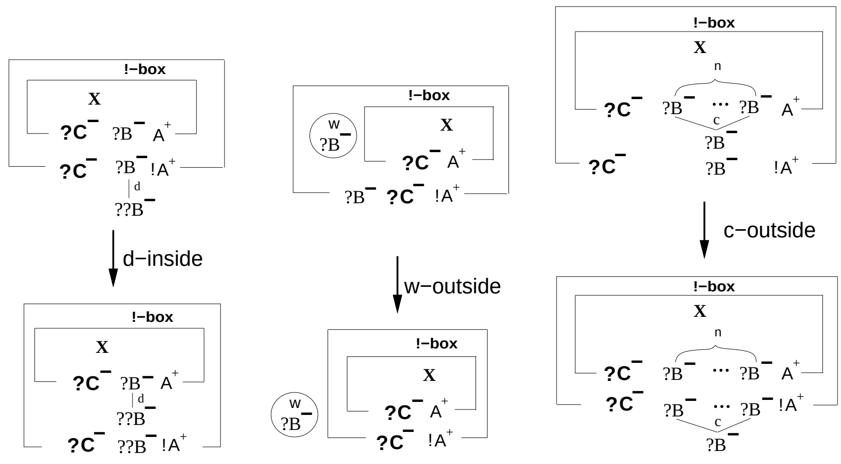


Fig. 8. d-inside, w-outside, and c-outside

Proof. Induction on the construction of IIMELL proof net Θ_1 and an easy argument on permutations of links. □

3 Strong Normalization and Confluence of $\rightarrow_{\mathrm{ex}}$

Lemma 3.1 (weak confluence) $\rightarrow_{\mathrm{ex}}$ *is weak confluent: when* $\Theta \rightarrow_{\mathrm{ex}} \Theta_1$ *and* $\Theta \rightarrow_{\mathrm{ex}} \Theta_2$*, there is a proof net* Θ_3 *such that* $\Theta_1 \twoheadrightarrow_{\mathrm{ex}} \Theta_3$ *and* $\Theta_2 \twoheadrightarrow_{\mathrm{ex}} \Theta_3$*.*

Proof. We have six critical pairs. But it is easy to see that each of these pairs has a confluent point. □

A proof of strong normalization for $\rightarrow_{\mathrm{std}}$ can be seen in [Gir87]. In order to prove the strong normalization of $\rightarrow_{\mathrm{ex}}$, we need some notions.

Definition 3.2 (depth) *For a link* L *in a proof net* Θ*, the depth of* L *in* Θ $\mathrm{d}_\Theta(L)$ *is the number of !-boxes that include* L *in* Θ*. The depth of* Θ $\mathrm{d}(\Theta)$ *is the maximum of* $\mathrm{d}_\Theta(L)$ *for any link* L *in* Θ*. The co-depth* $\text{co-d}_\Theta(L)$ *is* $\mathrm{d}(\Theta) - \mathrm{d}_\Theta(L)$*.*

Definition 3.3 *An alternative sequence of links and IIMELL formulas in* Θ

$$L_1, A_1^{pl_1}, \ldots, A_{i-1}^{pl_{i-1}}, L_i, A_i^{pl_i}, L_{i+1}, A_{i+1}^{pl_{i+1}}, \ldots, A_{k-1}^{pl_{k-1}}, L_k, A_k^{pl_k}$$

is a downward path if A_i *that is one of the conclusions of* L_i *is a premise of* L_{i+1} *for* $1 \leq i \leq k-1$*, where if* L_i *is a !-box, then either* $A_{i-1}^{pl_{i-1}} = A^+$ *and* $A_i^{pl_i} = !A^+$ *or* $A_{i-1}^{pl_{i-1}} = ?B^-$ *and* $A_i^{pl_i} = ?B^-$*. Then we also say that the downward path is from* $A_1^{pl_1}$ *to* $A_k^{pl_k}$*. A link* L *is upper than a link* L' *if there is a downward path* s, L, s', L', s''*, where* s, s' *and* s'' *are sequences of links. Then we also say* L' *is lower than* L*.*

Definition 3.4 *Let* Θ *be a proof net. Then*

- $\text{c-num}(\Theta)$ *is the number of c-links in* Θ*.*
- $\text{w-num}(\Theta)$ *is the number of w-links in* Θ*.*
- *For a contraction-link* L *in* Θ*,* $\text{cd}_\Theta(L)$ *is the number of d-links that are lower than* L*.*
- *For a weakening-link* L *in* Θ*,* $\text{wd}_\Theta(L)$ *is the number of d-links that are lower than* L*.*

By using the notions above, we define the weight of a proof net Θ.

Definition 3.5 (weight)

$$\begin{aligned}\text{weight}(\Theta) = \Big\langle & \text{c-num}(\Theta) + \text{w-num}(\Theta) \\ & + \sum_{L:\text{c-link in }\Theta} \text{cd}_\Theta(L) + \sum_{L:\text{w-link in }\Theta} \text{wd}_\Theta(L) \\ & + \sum_{L:\text{c-link in }\Theta} \text{d}_\Theta(L) + \sum_{L:\text{w-link in }\Theta} \text{d}_\Theta(L), \sum_{L:\text{d-link in }\Theta} \text{co-d}_\Theta(L) \Big\rangle\end{aligned}$$

Theorem 3.6 *If* $\Theta \rightarrow_{\text{ex}} \Theta'$*, then* $\text{weight}(\Theta) > \text{weight}(\Theta')$*.*

Proof.

(i) When $\Theta \rightarrow_{\text{c-d}} \Theta'$, then $\sum_{L:\text{d-link in }\Theta'} \text{co-d}_{\Theta'}(L)$ may increase, but
$$\sum_{L:\text{c-link in }\Theta} \text{cd}_\Theta(L) > \sum_{L:\text{c-link in }\Theta'} \text{cd}_{\Theta'}(L).$$

(ii) When $\Theta \rightarrow_{\text{d-w}} \Theta'$, then $\sum_{L:\text{w-link in }\Theta} \text{wd}_\Theta(L) > \sum_{L:\text{w-link in }\Theta'} \text{wd}_{\Theta'}(L)$.

(iii) When $\Theta \rightarrow_{\text{c-w}} \Theta'$, then $\text{w-num}(\Theta) > \text{w-num}(\Theta')$.

(iv) When $\Theta \rightarrow_{\text{c-del}} \Theta'$, then $\text{c-num}(\Theta) > \text{c-num}(\Theta')$.

(v) When $\Theta \rightarrow_{\text{fusion}} \Theta'$, then $\text{c-num}(\Theta) > \text{c-num}(\Theta')$.

(vi) When $\Theta \rightarrow_{\text{d-inside}} \Theta'$, then $\sum_{L:\text{d-link in }\Theta} \text{co-d}_\Theta(L) > \sum_{L:\text{d-link in }\Theta'} \text{co-d}_{\Theta'}(L)$.

(vii) When $\Theta \rightarrow_{\text{c-outside}} \Theta'$, then $\sum_{L:\text{c-link in }\Theta} \text{d}_\Theta(L) > \sum_{L:\text{c-link in }\Theta'} \text{d}_{\Theta'}(L)$.

(viii) When $\Theta \rightarrow_{\text{w-outside}} \Theta'$, then $\displaystyle\sum_{L:\text{w-link in }\Theta} \mathrm{d}_\Theta(L) > \sum_{L:\text{w-link in }\Theta'} \mathrm{d}_{\Theta'}(L)$.

In each case above, $\mathrm{weight}(\Theta) > \mathrm{weight}(\Theta')$. □

Corollary 3.7 *$\rightarrow_{\text{ex}}$ is strong normalizing.*

Theorem 3.8 *$\rightarrow_{\text{ex}}$ is confluent.*

Proof. By Lemma 3.1, Corollary 3.7, and Newman's lemma. □

4 An equality $=_{\text{ex}}$ on IIMELL proof nets

We make abbreviations as follows:

- $A \otimes B^-$ as $A \multimap B^-$ and $A \otimes ?B^-$ as $A \multimap !B^-$;
- $A \invamp B^+$ as $A \multimap B^+$ and $?A \invamp B^+$ as $!A \multimap B^+$;
- $A_1 \multimap (\cdots \multimap (A_n \multimap B)\cdots)$ as $A_1 \multimap \cdots \multimap A_n \multimap B$.

Figure 9 shows the general form of normal IIMELL proof nets in the sense of $\rightarrow_{\text{ex}}$. The proof net Θ has depth n. The formula $A(p)$ is $!p$ or $A' \multimap p$ for some formula A'. The link L_A is !-box or $\invamp$-link. The formula $B(p)$ is $?p$ or $B' \multimap p$ for some formula B'. The link L_B is d-link or $\otimes$-link.
Each formula A_i $(0 \le i \le n)$ has the form

$$\begin{aligned} &A_{i1} \multimap \cdots \multimap A_{im(i)} \multimap !(A_{i+11} \multimap \cdots \multimap A_{i+1m(i+1)} \\ &\qquad \multimap !(\cdots !(A_{n1} \multimap \cdots \multimap A_{nm(n)} \multimap p)\cdots))^+. \end{aligned}$$

Each formula B_i $(0 \le i \le n)$ the form

$$\begin{aligned} &B_{i11} \multimap \cdots \multimap B_{i1\ell(i1)} \multimap !(B_{i21} \multimap \cdots \multimap B_{i2\ell(i2)} \\ &\quad \multimap !(\cdots \multimap !(B_{ik(i)1} \multimap \cdots \multimap B_{ik(i)\ell(ik(i))} \\ &\quad \multimap !(\cdots \multimap !(B_{n11} \multimap \cdots \multimap B_{n1\ell(n1)} \multimap !(B_{n21} \multimap \cdots \multimap B_{n2\ell(n2)} \\ &\quad \multimap !(\cdots \multimap !(B_{nk(n)1} \multimap \cdots \multimap B_{nk(n)\ell(nk(n))} \multimap p)\cdots)))\cdots))\cdots))^-. \end{aligned}$$

The formula $B^+_{i_1 i_2 i_3}$ $(0 \le i_1 \le n, 1 \le i_2 \le k(i_1), 1 \le i_3 \le \ell(i_1 i_2))$, which is a subformula of B_0^+, is called an *imperial positive subformula* of B_0^+. Each imperial positive subformula is the positive conclusion formula of an imperial sub-proof-net (which is defined below).

Definition 4.1 (main paths) *Let s_1 be the reverse of the downward path from $A(p)^+$ to A_0^+, that is,*

$$A_0^+, L_1, \ldots, L_2, A(p)^+, L_A$$

and s_2 be the downward path from p^- to B_0^-, that is,

$$L_{\text{ID}}, p^-, L_B, B(p)^-, L_3, \ldots, L_4, B_0^-.$$

The main path of Θ is the concatenation of s_1, p^+ (which is the positive conclusion of L_{ID}), and s_2, that is,

$$A_0^+, L_1, \ldots, L_2, A(p)^+, L_A, p^+, L_{\text{ID}}, p^-, L_B, B(p)^-, L_3, \ldots, L_4, B_0^-.$$

Then we also say that the main path of Θ *is from* A_0^+ *to* B_0^-. *The positive subpath of the main path is* $A_0^+, L_1, \ldots, L_2, A(p)^+, L_A, p^+$ *and the negative subpath* $p^-, L_B, B(p)^-, L_3, \ldots, L_4, B_0^-$.

We call B_0^-, which is neither a premise of a $\otimes$-link nor the premise of a d-link, *the head-formula* of the main path of Θ. A negative formula occurrence C^- that has the form p^- or the form $B_1 \multimap B_2{}^-$ is *linear-discharged* if C^- is a premise of a $⅋$-link. A negative formula occurrence C^- that has the form $?B^-$ is *nonlinear-discharged* if there is a downward path from the occurrence $?B^-$ to the formula occurrence $!B \multimap A^+$ for some positive formula A^+,

$$L', ?B^-, L_{!1}, ?B^-, \ldots, ?B^-, L_{!k}, ?B^-, L_c, ?B^-, L_{⅋}, !B \multimap A^+,$$

where L' is some link, each $L_{!i}(1 \leq i \leq k, k \geq 0)$ is a !-box, L_c is a c-link and, $L_{⅋}$ is a $⅋$-link. In the both linear and nonlinear cases when $C \multimap A^+$ is the conclusion of the $⅋$-link, we say that the formula occurrence $C \multimap A^+$ is the *discharged point* of C^-. Generally discharge has the form of Figure 10. A negative formula occurrence that has the form $?B^-$ is *absorbed* if there is a downward path from the occurrence $?B^-$ to the formula occurrence $A \multimap !B^-$ for some positive formula A^+,

$$L', ?B^-, L_{!1}, ?B^-, \ldots, ?B^-, L_{!k}, ?B^-, L_c, ?B^-, L_{\otimes}, A \multimap !B^-,$$

where L' is some link, each $L_{!i}(1 \leq i \leq k, k \geq 0)$ is a !-box, L_c is a c-link, and $L_{\otimes}$ is a $\otimes$-link. We say that the formula occurrence $A \multimap !B^-$ in the downward path is the *absorbed point* of $?B^-$. Generally absorption has the form of Figure 10.

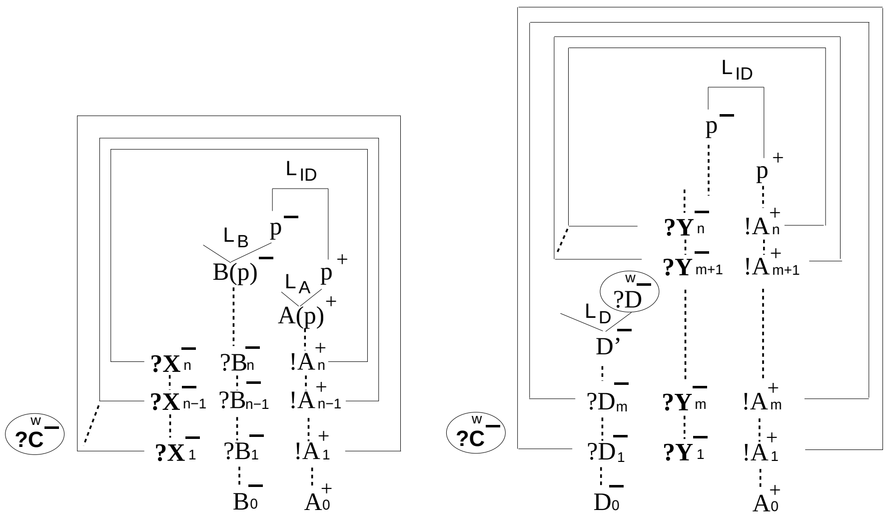


Fig. 9. the general form of normal IIMELL proof nets and that of dummy paths

The notion of absorption of ?-formulas is important: both proof nets in Figure 11 represent a term $fx(fxy)$ in usual functional notation. The left proof net uses absorption but the right not. The left proof net is more compact than the right, since it shares some links. Hence the use of absorption

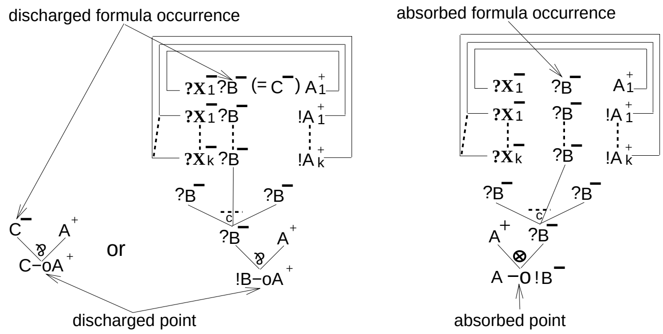


Fig. 10. the general form of discharge and that of absorption

makes representations of proofs or programs more efficient. But both standard lambda calculus and linear lambda calculus like [CP97] do not have the absorption. This is one of advantages of the syntax of proof-nets over linear lambda calculus.

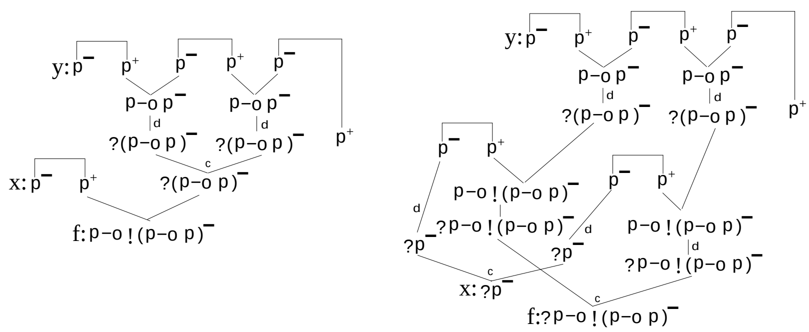


Fig. 11. an example which shows a usefulness of absorption of ?-formulas

Definition 4.2 (imperial sub-proof-nets) *Let Θ be a normal IIMELL proof nets. We inductively define imperial sub-proof-nets of Θ as follows:*

(i) *Let B^+ be an imperial positive subformula of the main path of Θ. The maximal sub-proof-net among the sub-proof-nets of Θ that have B^+ as the positive conclusion formula is an imperial sub-proof-net of Θ. We call such an imperial sub-proof-net a direct imperial sub-proof-net;*

(ii) *Let Θ' be an imperial sub-proof-net of Θ. An imperial sub-proof-net of Θ' is also an imperial sub-proof-net of Θ.*

A downward path beginning from a w-link to a negative conclusion formula, which we call a *dummy path*, has a similar form to main paths. Figure 9 shows the general form of such dummy paths. The formula D' is $?D$, $??D$, or $D'' \multimap D$ for some formula D''. The link L_D is !-box, d-link, c-link, or $\otimes$-link. Each formula D_i $(0 \leq i \leq m)$ the form

$$\begin{aligned}
&D_{i11} \multimap \cdots \multimap D_{i1v(i1)} \multimap !(D_{i21} \multimap \cdots \multimap D_{i2v(i2)} \\
&\quad \multimap !(\cdots \multimap !(D_{iu(i)1} \multimap \cdots \multimap D_{iu(i)v(iu(i))} \\
&\quad \multimap !(\cdots \multimap !(D_{m11} \multimap \cdots \multimap D_{m1v(m1)} \multimap !(D_{m21} \multimap \cdots \multimap D_{m2v(m2)}
\end{aligned}$$

$$\multimap !(\cdots \multimap !(D_{mu(m)1} \multimap \cdots \multimap D_{mu(m)v(mu(m))} \multimap p)\cdots)))\cdots))\cdots))^{-}.$$

We call $D^+_{i_1i_2i_3}$ $(0 \leq i_1 \leq n, 1 \leq i_2 \leq u(i_1), 1 \leq i_3 \leq v(i_1i_2))$, which is a subformula of D^+_0, an *dummy imperial positive subformula* of D^+_0. We also define the *dummy imperial sub-proof-nets* of $D^+_{i_1i_2i_3}$ in a similar fashion to imperial sub-proof-nets. But these dummy imperial sub-proof-nets are not imperial sub-proof-nets, because each dummy imperial sub-proof-net is above a dummy path. Basically the equality $=_{\mathrm{ex}}$ defined below ignores these dummy imperial sub-proof-nets.

Definition 4.3 (c-elim paths) *Let a normal IIMELL proof net be Θ. The c-elim path of the main path of Θ is the sequence obtained by eliminating all the occurrences $L_c, ?B^-$ for a c-link L_c and a negative formula $?B^-$ from the main path. Then the c-elim path does not include any c-links.*

We define $\equiv$ on IIMELL proof nets as follows: for IIMELL proof nets Θ^1 and Θ^2, $\Theta^1 \equiv \Theta^2$ if Θ^1 and Θ^2 are the same IIMELL proof nets exactly. On the other hand, $=_{\mathrm{ex}}$ is defined on normal IIMELL proof nets:

Definition 4.4 ($=_{\mathrm{ex}}$) *Let Θ^1 and Θ^2 be two normal IIMELL proof nets with the same positive conclusion formula.*
$\Theta^1 =_{\mathrm{ex}} \Theta^2$ is inductively defined on the number of the main paths of the imperial sub-proof-nets in Θ^1 and Θ^2.

(i) *the case where neither Θ^1 nor Θ^2 has any imperial sub-proof-nets: $\Theta^1 =_{\mathrm{ex}} \Theta^2$ if the following two conditions are satisfied:*
 (a) *The c-elim path of the main path of Θ^1 must be the same as that of Θ^2. Then both head-formulas of the main paths of Θ^1 and Θ^2 are the same and the head-formula must be $\overbrace{?\cdots?}^{s}p^-$ for some MELL formula p and $s \geq 0$.* **(coincidence of main paths)**
 (b) *The head-formula of Θ^1 $\overbrace{?\cdots?}^{s}p^-$ is discharged in Θ^1 iff the head-formula of Θ^2 $\overbrace{?\cdots?}^{s}p^-$ is discharged in Θ^2. Then if both head-formulas are discharged, then both head-formulas must have the same discharged point.* **(coincidence of head states)**

(ii) *otherwise: if Θ^1 and Θ^2 have different numbers of the main paths of the imperial sub-proof-nets, then $\Theta^1 \neq_{\mathrm{ex}} \Theta^2$. Otherwise, $\Theta^1 =_{\mathrm{ex}} \Theta^2$ if the following conditions are satisfied:*
 (a) *The c-elim path of the main path of Θ^1 must be the same as that of Θ^2.* **(coincidence of main paths)**
 (b) *The head-formula of Θ^1 is discharged in Θ^1 iff the head-formula of Θ^2 is discharged in Θ^2. Moreover, if both head-formulas of Θ^1 and Θ^2 are discharged, then both head-formulas must have the same discharged point.* **(coincidence of head states)**
 (c) *For each imperial positive subformula $B^+_{i_1i_2i_3}$ $(0 \leq i_1 \leq n, 1 \leq i_2 \leq k(i_1), 1 \leq i_3 \leq \ell(i_1i_2))$ of the head-formula of Θ^1 and Θ^2,*

$e_{\Theta^1}(B^+_{i_1 i_2 i_3}) =_{\mathrm{ex}} e_{\Theta^2}(B^+_{i_1 i_2 i_3})$, *where* $e_\Theta(B^+)$ *is the imperial sub-proof-net with the positive conclusion formula* B^+. **(inductive condition)**

(d) *Let* Θ'^1 *and* Θ'^2 *be an imperial sub-proof-net of* Θ^1 *and that of* Θ^2 *respectively that have the same positive conclusion formula* C^+ *whose position in* Θ^1 *is the same as* Θ^2. *The head-formula of the main path of* Θ'^1 *is discharged (resp. absorbed) iff the head-formula of the main path of* Θ'^2 *is discharged (resp. absorbed). Moreover, if both head-formulas are discharged (resp. absorbed) in both* Θ^1 *and* Θ^2, *then both head-formulas must have the same discharged (resp. absorbed) point.* **(coincidence of discharge and absorption)**

(e) *Let* Θ'^1 *(resp.* Θ''^1*) and* Θ'^2 *(resp.* Θ''^2*) be an imperial sub-proof-net of* Θ^1 *and that of* Θ^2 *that have the same positive conclusion formula* C'^+ *(resp.* C''^+*) whose position in* Θ^1 *is the same as* Θ^2 *and whose head-formulas of the form* $?B'^-_0$ *(resp.* $?B''^-_0$*) are neither discharged nor absorbed. Then, in* Θ^1 $?B'^-_0$ *and* $?B''^-_0$ *are premises of a c-link iff in* Θ^2 $?B'^-_0$ *and* $?B''^-_0$ *are premises of a c-link.* **(coincidence of free ports)**

Remark 4.5 [the difference between $\equiv$ and $=_{\mathrm{ex}}$] The two relations $\equiv$ and $=_{\mathrm{ex}}$ are different even if we restrict $\equiv$ to normal IIMELL proof nets: The left hand side Θ^1 and the right hand side Θ^2 of Figure 12 satisfy $\Theta^1 =_{\mathrm{ex}} \Theta^2$ by Definition 4.4. But obviously $\Theta^1 \not\equiv \Theta^2$.

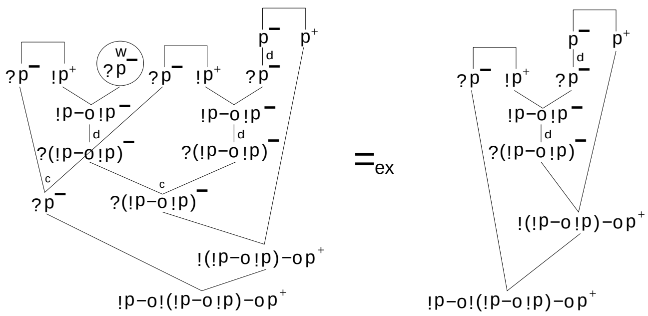


Fig. 12. an example of $\equiv \neq =_{\mathrm{ex}}$

Remark 4.6 [coincidence of main paths] Note that it may happen that although the head-formula B^{1-}_0 of the main path of Θ^1 and the head-formula B^{2-}_0 of the main path of Θ^2 are the same formula, B^{1-}_0 and B^{2-}_0 have different indices depending on Θ^1 and Θ^2. For example, in Figure 13 although two IIMELL proof nets have $B_0 = ?(q \multimap !p)^-$, when indices are considered, the left proof net has $B_0 = ?(B_{011} \multimap !p)$ and the right $B_0 = ?(B_{111} \multimap !p)$.

5 Concluding remarks

The eight rewrite rules of $\to_{\mathrm{ex}}$ are only applied to normal forms of proof nets in the usual sense because when these rewrite rules are incorporated into the

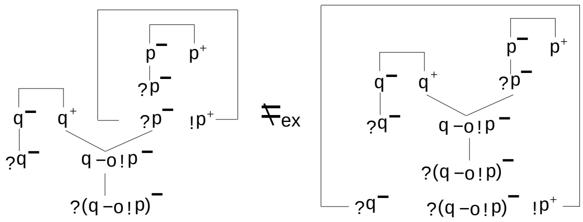


Fig. 13. two IIMELL proof net with different indices

standard rewrite rules of Linear Logic, the system violates the Church-Rosser property. Figure 14 shows a counterexample. As shown in [DK97] and [DG99], the reduction relation generated by the standard rewrite rules of Linear Logic, fusion, cw, c-del, c-outside, and w-outside rewrite rules has the Church-Rosser property. But when c-d rewrite rule is applied to a proof net, a new fusion redex or c-outside redex may be created and when d-w rewrite rules is applied to a proof net, a new c-w or w-outside redex may be created. Therefore in $\rightarrow_{\mathrm{ex}}$ we need fusion, cw, c-del, c-outside, and w-outside rewrite rules.
The proof nets system in [DR95] combines c-link, d-link and w-link into one *why not* link. In this system we need no more d-inside and c-outside rewrite rules since a why not link can cross several !-boxes. In addition, to the system in [DR95] we can add a commutative rewrite rule on why not links similar to the c-d rewrite rule. Figure 15 shows the rewrite rule. The Church-Rosser property does not hold in the extended system in the same way as Figure 14. On the other hand, our additional eight rewrite rules are compatible with classical proof nets. An interesting question comes up: what are approximations to classical proof nets? We have not found the answer to the question yet.

References

[CP97] I. Cervesato and F. Pfenning. Linear Higher-Order Pre-Unification. LICS'97, 1997.

[DK97] R. Di Cosmo and D. Kesner. Strong Normalization of Explicit Substitutions via Cut Elimination in Proof Nets. LICS'97, 1997.

[DG99] R. Di Cosmo and S. Guerrini. Strong normalization of proof nets modulo structural congruences. RTA'99, 1999.

[DR95] V. Danos and L. Regnier. Proof-nets and the Hilbert space. *Advances in Linear Logic, London Mathematical Society Lecture Notes Series 222*, pp 307-328, Cambridge University Press, 1995.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Hue75] G. Huet. A Unification Algorithm for Typed λ-Calculus. *Theoretical Computer Science*, 1:27–57, 1975.

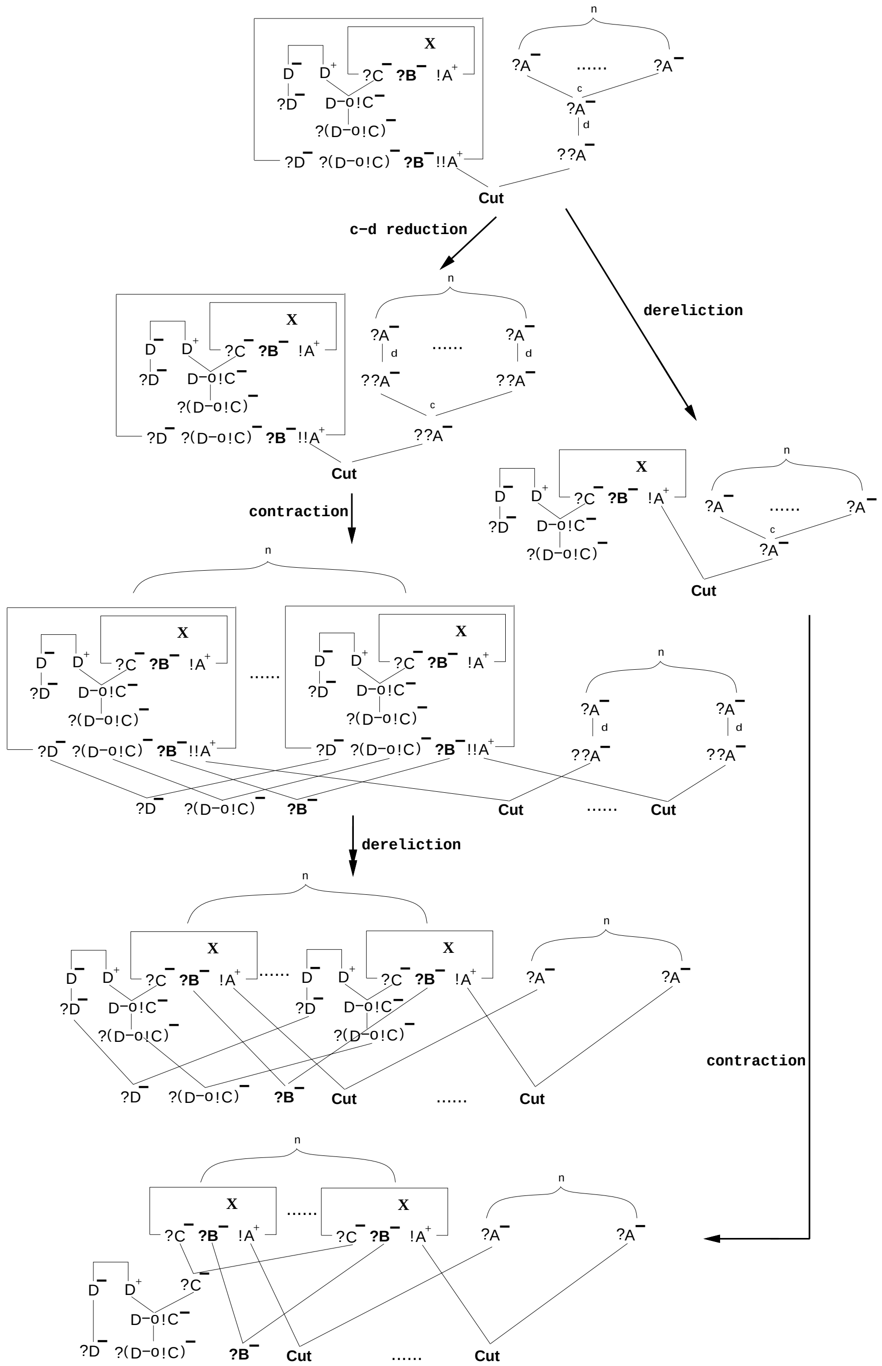


Fig. 14. A counterexample which shows that the Church-Rosser property does not hold

[Laf95] Y. Lafont. From Proof-Nets to Interaction Nets. *Advances in Linear Logic, London Mathematical Society Lecture Notes Series 222*, pp 225-247, Cambridge University Press, 1995.

[Mat00a] S. Matsuoka. Some Applications of Linear Logic to Automated Theorem Proving and Logic Programming (in Japanese). Dissertation, Department of

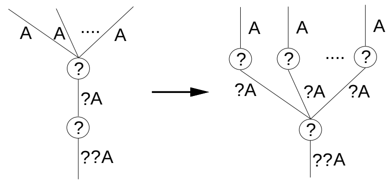


Fig. 15. the c-d rewrite rule in Danos-Regnier's nets

Electrical and Computer Engineering, Faculty of Engineering, Nagoya Institute of Technology, March, 2000.

[Mat00b] S. Matsuoka. Higher Order Pre-Unification on Intuitionistic Proof Nets (Extended Abstract). Proceedings of the 4th Asian Symposium on Computer Mathematics, pp 138-141, World Scientific, 2000.

[Mat01a] S. Matsuoka. Higher Order Pre-Unification on Intuitionistic Proof Nets with Exponentials. Preparation, 2001.

[NM88] G. Nadathur and D. Miller. An Overview of λProlog. In *Fifth International Logic Programming Conference*, pages 810-827, MIT Press, 1988.

[SG89] W. Snyder and J.H. Gallier. Higher Order Unification Revisited: Complete Sets of Transformations. *Journal of Symbolic Computation*, 8:101-140, 1989.