*Electronic Notes in Theoretical Computer Science 50 No. 2 (2001) { Proc. BOTH 2001 URL:* [*http://www.elsevier.nl/locate/entcs/volume50.html*](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 con- cerns 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 con uent. 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 Bohm-trees in -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 mul- tiplicative exponential fragment of proof nets, the graph rewriting reduction is strong normalizing [Gir87] and con uent [Laf95]. Proof nets can thus be a computational system.

*But the equality based on the graph rewriting reduction in [Laf95] makes un-*

*necessary distinctions between normal proof nets. The distinctions are due to* ordering of introductions of exponential links in proof nets. The graph rewrit- ing 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-uni cation on intuitionistic proof nets [Mat00a,Mat00b,Mat01a], the three rewrite rules are not suÆcient: higher order pre-uni cation algorithm, which is the central part of higher order programming languages like [NM88], is con- structed based on the notion of approximations to terms like Bohm-trees (see [SG89]). But in the system in [DK97] the notion of approximations to intu- itionistic proof nets is complex, it is hopeless to construct a simple uni cation algorithm based on the notion. For example, in [DK97] the left side of Figure 1

*c 2001 Published by Elsevier Science B. V. Open access under* [*CC BY-NC-ND license.*](http://creativecommons.org/licenses/by-nc-nd/3.0/)

*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 diÆcult to de ne an equality on such approximations. Moreover !-boxes make the situation worse.

q+ p**−**

q+

q+ p**−**

q −o p **−**

q**−** q**−**

d

d

?q**−** ?q**−**

q**−**

d

?q**−**

q+

q −o q −o p **−**

d

??q**−**

?q**−**

??q**−**

c

q −o q −o q −o p **−**

d

c

??q**−**

p+

q+ q −o p **−**

?q**−**

?q**−**

q+ q −o q −o p **−**

!! q −o p **−**

??q**−**

c

?q**−**

d

??q**−**

c

??q**−**

q −o q −o q −o p **−**

p+

!! q −o p **−**

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

!! q −o p **−**

??q**−** ??q**−** ??q**−**

p+

q+

q+ p**−**

q −o p **−**

q+

q −o q −o p **−**

q −o q −o q −o p **−**

!! q −o p **−**

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

c

??q**−**

q+

q+ p**−**

q −o p **−**

q**−** q**−**

d d

?q**−** ?q**−**

d

d

q**−**

d

?q**−**

d

q+

q −o q −o p **−**

q −o q −o q −o p **−**

??q**−** ??q**−** ??q**−**

c

??q**−**

p+

*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 de ne 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 con uent. Moreover, we propose the notion of an equality based on the rewriting rules. We have succeeded to construct a simple higher order pre-uni cation algorithm on intuitionistic proof nets based on the equality (see [Mat01a]). The main features of the higher order

*pre-uni cation 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 multi-* plicative exponential fragment of Linear Logic (for short IIMELL).

*De nition 2.1 (MELL formulas) MELL formulas (or simply formulas)*

*(F) are inductively constructed from atomic formulas (P) and logical connec-* tives:

*P pjqjrj*

*F P jF F jF OF j!F j?F:*

*De nition 2.2 (polarized formulas) An IIMELL formula is a pair hA; pli* where A is a MELL formula and pl is an element of f+; g. + and are called Danos-Regnier polarity. A formula hA; pli can be written as Apl. 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.

A+ A

p+

p

Cut

A+ B



A B

A B+

A

B

+

ID−link

A

?A

Cut−link

?A

−link

?A .... ?A

?A

−link

?A1...?An B+

?A1 ?An !B+

d−link w−link c−link !−box

*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 -link is called a premise of the link and A B the conclusion of the link;*

*(iv) each of A and B+ of O-link is called a premise of the link and AOB+* 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 ;::: ; ?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 ;::: ; ?A ; B+ of the maximal*

*1 n*

*proof net inside !-box is called a premise of the !-box and each of formula*

*occurrences ?A ;::: ; ?A ; B+ of !-box is called a conclusion of the !-box.*

*1 n*

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

is an IIMELL proof net.

A A+

**D** B E +



**C** A+ **D** B E +

A B

if and

**C** A+

are IIMELL proof nets,then is an IIMELL proof net.

if and

**C** A+

**D** A E +

are IIMELL proof nets,then

**C** A+

**D** A E +

Cut

is an IIMELL proof net.

if is an IIMELL proof net, then

**C** A

B+

if is an IIMELL proof net, then

**C** A

B+

**C** A B+

A B+

**C** A B+

?A

**B** A+

?C

is an IIMELL proof net.

is an IIMELL proof net.

if

**B** A+

is an IIMELL proof net, then is an IIMELL proof net.

if is an IIMELL proof net, then is an IIMELL proof net.

**C** ?A ...?A B+

**C** ?A ...?A B+

?A

**!−box**

?A ?A

**−**

1

....

**−** +

n

B

?A ?A

**−**

1

....

**−** +

n

!B

if is an IIMELL proof net, then is an IIMELL proof net.

?A ?A

**−**

1

....

**−** +

n

B

*Fig. 3. the de nition of IIMELL proof nets*

*and Figure 6 show ID, multiplicative, dereliction, contraction, weakening and* of-course rewrite rules. We call the reduction relation de ned by these six rewrite rules the standard reduction, which is denoted by !!std. The one- step reduction of !!std is denoted by !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 de ne !!ex as the reduction relation on normal proof nets of !std generated by these eight rewrite rules, which is called the extended reduction. This de nition is well-de ned because these eight rewrite rules never create

A+( ) A (+)

**X**

A+( )

**Y**

**Cut** B



**Y**

B

A B

A

**X**

A+

**Z**

A

B+

+

ID rewrite rule



**Cut**

multiplicative rewrite rule



**X**

A+

**Y**

B

**Z**

A

B+

**Cut Cut**

**X**

A+( )

**Y**

*Fig. 4. ID and multiplicative rewrite rules*

A**−**

**X**

**?B−** A+

**?B−** ! A+

**−**



?B1**−** ?B n**−**

**−**.... **−** A+

?B1 ?Bn ! A+

w

?A**−**

dereliction

?A

**Cut**



weaken

**X**

**?B−** A+

A**−**

**Cut**

**Cut**



ing

......

w

?B**−**

n



?B**−**

w

1

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

?A**−**

n

?A**−**

**X**

**?B−** A+

**?B**

**−**

+

! A

. ?A**−**

**Cut**



c

?A**−**

**X**

**?B−** A+

**?B−** ! A+

**Y**

?A **?C**

**−**

**−**

D

+

?A**− ?C−**! D+

contraction

**Cut**

of−course



n

**?B−**



**X**

**?B−** A+

**X**

**?B−** A+

......

**X**

**?B−** A+

n

**?B−**

?A ?A ?A

**− −**

**−**

! A

+

**?B−**

! A

+

**?B−**

! A

+

c

**Cut Cut**

......

**Cut**

**!−box**

**X**

**?B−** A+

**?B−** ! A+

**Y**

?A**− ?C−**D+

**Cut**

**?B− ?C −** !D+

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

*any Cut-links. The one-step reduction of !!ex is denoted by !ex.*

*Proposition 2.3 Let 1 be an IIMELL proof net. If 1 !ex 2, then 2 is* also an IIMELL proof net.

?A**−**

?A**−**

n

......

c

?A**−**

d

??A**−**



c−d

......

?A**−**

?A**−**

w

?A**−**

d

??A**−**

n+1

?A**−**

w

?A**−**...... ?A**−** ?A

c



c−w

n



d−w

??A**−**

w

**−**

?A**−**

c



m



c

n+1

c

?A**−**...... ?A**−** ?A**−**

?A**−** ?A**−**

?A**−**



fusion

m+n

d d

?A**− − −**

??A**−**

c

**−**

??A**−**

?A**−**

......

c

**−**

?A**−**

c−del

?A ?A

c

**−**

??A

?A ?A**−** ?A

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

**!−box**



**!−box**

**?C−**

?B**−**

?B**−** ! A+

**X**

n

**?C−** ?B**−** ... ?B**−** +

c

A

**X**



?B**−**

w

**!−box**

**X**

**?C** A

**−** +

?B**− ?C−** ! A+

**?C−** ?B**−** +

A

**?C−** ?B**−** ! A+

d

??B**−**



d−inside

w−outside

**!−box**

**?C**

**X**

**−**

?B A

d

**−**

+

**?C−** ??B**−** ! A+

??B**−**



?B**−**

w

**!−box**

**X**

**?C**

**−**

A

+

**?C−** ! A+

c−outside



c



**!−box**

**X**

n

**?C−**

**?C−**

?B**−** ... ?B**−**

A

+

?B**−** ... ?B**−** ! A+

?B**−**

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

*Proof. Induction on the construction of IIMELL proof net 1 and an easy* argument on permutations of links. *2*

# *3 Strong Normalization and Con uence of !ex*

*Lemma 3.1 (weak con uence) !ex is weak con uent: when !ex 1* and !ex 2, there is a proof net 3 such that 1 !!ex 3 and 2 !!ex 3.

*Proof. We have six critical pairs. But it is easy to see that each of these pairs* has a con uent point. *2*

*A proof of strong normalization for !std can be seen in [Gir87]. In order* to prove the strong normalization of !ex, we need some notions.

*De nition 3.2 (depth) For a link L in a proof net , the depth of L in d(L) is the number of !-boxes that include L in . The depth of d( )* is the maximum of d(L) for any link L in . The co-depth co-d(L) is d( ) d(L).

*De nition 3.3 An alternative sequence of links and IIMELL formulas in*

*L ; Apl1 ;::: ; Apli 1 ;L ; Apli ;L ; Apli+1 ;::: ; Aplk 1 ;L ; Aplk*

*1 1 i 1 i* *i*

*i+1*

*i+1*

*k 1 k k*

*is a downward path if Ai that is one of the conclusions of Li is a premise*

*of L*

*i 1*

*i+1*

*for 1 i k 1, where if Li*

*is a !-box, then either Apli 1*

*= A+*

*and Apli*

*i*

*=!A+ or Apli 1*

*=?B and Apli*

*=?B . Then we also say that the*

*downward path is from Apl1 to Aplk . A link L is upper than a link L0 if there is*

*i 1*

*i*

*1 k*

*a downward path s; L; s0; L0 ; s00, where s; s0 and s00 are sequences of links. Then*

*we also say L0 is lower than L.*

*De nition 3.4 Let be a proof net. Then*

*c-num( ) is the number of c-links in .*

*w-num( ) is the number of w-links in .*

*For a contraction-link L in , cd(L) is the number of d-links that are* lower than L.

*For a weakening-link L in , wd(L) is the number of d-links that are* lower than L.

*By using the notions above, we de ne the weight of a proof net .*

*De nition 3.5 (weight)*

*weight( ) = Dc-num( ) + w-num( )*

*+* *X*

*L:c-link in*

*cd(L)+* *X*

*L:w-link in*

*wd(L)*

*+* *X*

*L:c-link in*

*d(L)+* *X*

*L:w-link in*

*d(L);* *X*

*L:d-link in*

*co-d(L)E*

*Theorem 3.6 If !ex 0, then weight( ) > weight( 0).*

*Proof.*

*(i) When !c-d 0, then* *X*

*L:d-link in 0*

*co-d0 (L) may increase, but*

*X*

*L:c-link in*

*cd(L) >* *X*

*L:c-link in 0*

*cd0 (L).*

*(ii) When !d-w 0, then* *X*

*L:w-link in*

*wd(L) >* *X*

*L:w-link in 0*

*wd0 (L).*

*(iii) When !c-w 0, then w-num( ) > w-num( 0).*

*(iv) When !c-del 0, then c-num( ) > c-num( 0).*

*(v) When !fusion 0, then c-num( ) > c-num( 0).*

*(vi) When !d-inside 0, then* *X*

*L:d-link in*

*co-d(L) >* *X*

*L:d-link in 0*

*co-d0 (L).*

*(vii) When !c-outside 0, then* *X*

*L:c-link in*

*d(L) >* *X*

*L:c-link in 0*

*d0 (L).*

*(viii) When !w-outside 0 , then* *X*

*L:w-link in*

*d(L) >* *X*

*L:w-link in 0*

*d0 (L).*

*In each case above, weight( ) > weight( 0).* *2*

*Corollary 3.7 !ex is strong normalizing.* Theorem 3.8 !ex is con uent.

*Proof. By Lemma 3.1, Corollary 3.7, and Newman's lemma.* *2*

*4 An equality =ex on IIMELL proof nets We make abbreviations as follows:*

*A B as A Æ B and A ?B as A Æ!B ;*

*AOB+ as A Æ B+ and ?AOB+ as !A Æ B+;*

*A1 Æ( Æ(An Æ B) ) as A1 Æ Æ An Æ B.*

*Figure 9 shows the general form of normal IIMELL proof nets in the sense of*

*!ex. The proof net has depth n. The formula A(p) is !p or A0 Æ p for some* formula A0 . The link LA is !-box or O-link. The formula B(p) is ?p or B0 Æ p for some formula B0. The link LB is d-link or *link.*

*Each formula Ai (0 i n) has the form*

*Ai1 Æ Æ Aim(i) Æ !(Ai+11 Æ Æ Ai+1m(i+1)*

*Æ!( !(An1 Æ Æ Anm(n) Æ p) ))+: Each formula Bi (0 i n) the form*

*Bi11 Æ Æ Bi1`(i1) Æ!(Bi21 Æ Æ Bi2`(i2)*

*Æ!( Æ!(Bik(i)1 Æ Æ Bik(i)`(ik(i))*

*Æ!( Æ!(Bn11 Æ Æ Bn1`(n1) Æ!(Bn21 Æ Æ Bn2`(n2)*

*Æ!( Æ!(B Æ Æ B Æ p) ))) )) )) :*

*nk(n)1 nk(n)`(nk(n))*

*The formula B+ (0 i1 n; 1 i2 k(i1); 1 i3 `(i1i2)), which is*

*i1i2i3*

*a subformula of B+, is called an imperial positive subformula of B+. Each*

*0 0*

*imperial positive subformula is the positive conclusion formula of an imperial*

*sub-proof-net (which is de ned below).*

*De nition 4.1 (main paths) Let s1 be the reverse of the downward path* from A(p)+ to A+, that is,

*0*

*A+; L1;::: ; L2; A(p)+; LA*

*0*

*and s2 be the downward path from p to B , that is,*

*0*

*L ; p ;L*

*; B(p) ;L ;::: ;L ; B :*

*ID B*

*3 4 0*

*The main path of is the concatenation of s1, p+ (which is the positive con-* clusion of LID), and s2, that is,

*A+;L ;::: ;L ; A(p)+;L ; p+;L*

*; p ;L*

*; B(p) ;L ;::: ;L ; B :*

*0 1 2*

*A ID B*

*3 4 0*

*Then we also say that the main path of is from A+ to B . The positive sub-*

*0 0*

*path of the main path is A+; L1;::: ; L2; A(p)+; LA; p+ and the negative subpath*

*0*

*p ;L ; B(p) ;L ;::: ;L ; B .*

*B 3 4* *0*

*We call B , which is neither a premise of a -link nor the premise of a* d-link, the head-formula of the main path of . A negative formula occurrence C that has the form p or the form B Æ B is linear-discharged if C is a premise of a O-link. A negative formula occurrence C that has the form

*0*

*1 2*

*?B is nonlinear-discharged if there is a downward path from the occurrence*

*?B to the formula occurrence !B Æ A+ for some positive formula A+,*

*L0 ; ?B ; L!1; ?B ;::: ; ?B ; L!*

*k*

*; ?B ;L ; ?B ;L*

*; !B Æ A+;*

*where L0 is some link, each L!i(1 i k; k 0) is a !-box, Lc is a c-link*

*c*

*O*

*and, LO*

*is a O-link. In the both linear and nonlinear cases when C Æ A+ is*

*the conclusion of the O-link, we say that the formula occurrence C Æ 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 Æ!B for some positive formula A+,

*L0 ; ?B ;L ; ?B ;::: ; ?B ;L ; ?B ;L ; ?B ;L*

*;A Æ!B ;*

*!1 !k c*

*where L0 is some link, each L!i(1 i k; k 0) is a !-box, Lc is a c-link,* and L is a -link. We say that the formula occurrence A Æ!B in the downward path is the absorbed point of ?B . Generally absorption has the form of Figure 10.

L ID

**?Y** n

−

p +

!An

!A+

+

**?Y**−m+1

L D −

?D−

w

m+1

D’

?D−

m

1

**?Y**−m

**?Y**−1

!A+

m

?D−

!A 1

+

D−

0

A+

0

p−

w

**?C**

−

L B

p +

A(p)+

A

**?X**−n ?B−n

!A+

n

**?X**

−

L ID

p−

B(p)− L

n−1

!A

+

n−1

n−1 ?B

−

+

1

!A 1

**?X**−1

?B−

w

**?C**

−

B0 A0

*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 f x(f xy) in usual functional notation. The left proof net uses absorption but the right not. The left proof net is more com- pact than the right, since it shares some links. Hence the use of absorption

discharged formula occurrence absorbed formula occurrence

**?X**

**− −**

1

?B C A1

(= )

**−**

+

**?X−**1 ?B**−** !A +

1

**?X−**k ?B**−**

?B**−**

c **−**

!A +

k

C**−** A+

?B**−**

+

**?X−**1

**?X−**1

**?X−**k

?B**−**

?B**−**

?B**−**

?B**−**

?B**−**

A+

1

!A +

1

!A +

k

A ?B

+

c **−**

C−oA + or

?B A

+

!B−oA

A −o ! B**−**

discharged point absorbed point

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

*makes representations of proofs or programs more eÆcient. But both stan-* dard 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.

y: p**−** p+

p**−** p+ p**−**

y: p**−** p+

p**−** p+ p**−**

p−o p**−**

p−o p**−**

p−o p**−**

p−o p**−**

d

**− −** p+

d

d

?( p−o p )**−**

d

?( p−o p )**−** p+

c

p**−** p+

?( p−o p ) ?( p−o p )

x: p**−** p+

?( p−o p )**−**

p−o

d

! ( p−o

p )**−**

p**−** p+

f: p−o ! ( p−o p )**−**

**−**? p−o ! ( p−o p )**−** d p−o ! ( p−o p )**−**

?p ?p**−** p **−**

d

x:?p**−**

c

? −o ! ( p−o p )

c

f:? p−o ! ( p−o p )**−**

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

*De nition 4.2 (imperial sub-proof-nets) Let be a normal IIMELL* proof nets. We inductively de ne imperial sub-proof-nets of as follows:

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

*(ii) Let 0 be an imperial sub-proof-net of . An imperial sub-proof-net of 0* is also an imperial sub-proof-net of .

*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 D0 is ?D, ??D, or D00 Æ D for some formula D00. The link LD is !-box, d-link, c-link, or -link. Each formula Di (0 i m) the form

*Di11 Æ Æ Di1v(i1) Æ!(Di21 Æ Æ Di2v(i2)*

*Æ!( Æ!(Diu(i)1 Æ Æ Diu(i)v(iu(i))*

*Æ!( Æ!(Dm11 Æ Æ Dm1v(m1) Æ!(Dm21 Æ Æ Dm2v(m2)*

*Æ!( Æ!(D Æ Æ D Æ p) ))) )) )) :*

*mu(m)1 mu(m)v(mu(m))*

*We call D+ (0 i1 n; 1 i2 u(i1); 1 i3 v(i1i2)), which is a*

*i1i2i3*

*subformula of D+, an dummy imperial positive subformula of D+. We also*

*0*

*de ne the dummy imperial sub-proof-nets of D+*

*i1i2i3*

*0*

*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 =ex de ned below ignores these dummy imperial sub-proof-nets.

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

*We de ne on IIMELL proof nets as follows: for IIMELL proof nets 1 and*

*2, 1 2 if 1 and 2 are the same IIMELL proof nets exactly. On the* other hand, =ex is de ned on normal IIMELL proof nets:

*De nition 4.4 (=ex) Let 1 and 2 be two normal IIMELL proof nets with* the same positive conclusion formula.

*1 =ex 2 is inductively de ned on the number of the main paths of the* imperial sub-proof-nets in 1 and 2.

*(i) the case where neither 1 nor 2 has any imperial sub-proof-nets: 1 =ex*

*2 if the following two conditions are satis ed:*

*(a) The c-elim path of the main path of 1 must be the same as that of*

*2. Then both head-formulas of the main paths of 1 and 2 are the*

*s*

*same and the head-formula must be ?z } | ?{p for some MELL formula*

*p and s 0. (coincidence of main paths)*

*s*

*(b) The head-formula of 1*

*s*

*?z } | ?{p is discharged in 1 i the head-*

*formula of 2 z? } | ?{p is discharged in 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 1 and 2 have di erent numbers of the main paths of the* imperial sub-proof-nets, then 1 6=ex 2. Otherwise, 1 =ex 2 if the following conditions are satis ed:

*(a) The c-elim path of the main path of 1 must be the same as that of*

*2. (coincidence of main paths)*

*(b) The head-formula of 1 is discharged in 1 i the head-formula of 2* is discharged in 2. Moreover, if both head-formulas of 1 and 2 are discharged, then both head-formulas must have the same discharged point. (coincidence of head states)

*(c) For each imperial positive subformula B+ (0 i1 n; 1 i2*

*i1i2i3*

*k(i1); 1 i3 `(i1i2)) of the head-formula of 1 and 2,*

*+ + +*

*e1 (Bi i i ) =ex e2 (Bi i i ), where e(B*

*) is the imperial sub-proof-*

*1 2 3 1 2 3*

*net with the positive conclusion formula B+. (inductive condition)*

*(d) Let 01 and 02 be an imperial sub-proof-net of 1 and that of 2* respectively that have the same positive conclusion formula C+ whose position in 1 is the same as 2. The head-formula of the main path of 01 is discharged (resp. absorbed) i the head-formula of the main path of 02 is discharged (resp. absorbed). Moreover, if both head- formulas are discharged (resp. absorbed) in both 1 and 2, then both head-formulas must have the same discharged (resp. absorbed) point. (coincidence of discharge and absorption)

*(e) Let 01 (resp. 001) and 02 (resp. 002) be an imperial sub-proof-net* of 1 and that of 2 that have the same positive conclusion formula

*C0+ (resp. C00+) whose position in 1 is the same as 2 and whose* head-formulas of the form ?B0 (resp. ?B00 ) are neither discharged

*0 0*

*nor absorbed. Then, in 1 ?B0 and ?B00 are premises of a c-link*

*0*

*0*

*i in 2 ?B0 and ?B00 are premises of a c-link. (coincidence of*

*0 0*

*free ports)*

*Remark 4.5 [the di erence between and =ex] The two relations and*

*=ex are di erent even if we restrict to normal IIMELL proof nets: The left* hand side 1 and the right hand side 2 of Figure 12 satisfy 1 =ex 2 by De nition 4.4. But obviously 1 6 2.

p**−** p+ p**−** p+



? p

**−** ! p+ p

? **−**

w

d

? p

**−** ! p+ ? p**−**

! p−o ! p **−** ! p−o ! p **−**

d d

?(! p−o ! p)

**−**

?(! −o ! )

p p

**−**

c

? p**−**

?(! p−o ! p)**−**

! (! p−o ! p) −o p+

c

=ex

d

? p**−** ! p+ ? p**−**

! p−o ! p **−**

d

?(! p−o ! p)**−**

! (! p−o ! p) −o p+

! p−o!(! p−o ! p) −o p+ ! p−o!(! p−o ! p) −o p+

*Fig. 12. an example of 6==ex*

*Remark 4.6 [coincidence of main paths] Note that it may happen that al-*

*though the head-formula B1 of the main path of 1 and the head-formula*

*0*

*B2 of the main path of 2 are the same formula, B1 and B2 have di erent*

*0 0 0*

*indices depending on 1 and 2. For example, in Figure 13 although two*

*IIMELL proof nets have B0 =?(q Æ!p) , when indices are considered, the left* proof net has B0 =?(B011 Æ!p) and the right B0 =?(B111 Æ!p).

# *5 Concluding remarks*

*The eight rewrite rules of !ex are only applied to normal forms of proof nets* in the usual sense because when these rewrite rules are incorporated into the

q**−**

?q**−**

p**−**

? p**−**

q+ ? p**−**

q −o ! p **−**

p+

! p+

=ex

?(q −o ! p)**−**

*Fig. 13. two IIMELL proof net with di erent indices*

q**−**

?q**−**

?q**−**

p**−** p+

q+ ? p**−**

q −o ! p **−**

?(q −o ! p)**−**

?(q −o ! p)**−** ! p+

*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

*!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 clas- sical 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-Uni cation.*

*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 Uni cation Algorithm for Typed -Calculus. Theoretical Computer Science, 1:27{57, 1975.*

?A**−**

D**−** D+

**X**

?C**− ?B−** ! A+

?D**−** D−o ! C**−**

?( D−o ! C)**−**

?D**−** ?( D−o ! C)**− ?B−**!! A+

n

......

c

A**−**

?

d

?A**−**

??A**−**

**Cut**

**c−d reduction**



D**−** D+

**X**

?C**− ?B−** ! A+

?D**−**

D−o ! C**−**

?( D−o ! C)**−**

?A**−**

d

??A**−**

.....

?D**−** ?( D−o ! C)**− ?B−**!! A+

c

??A**−**

**dereliction**

**−**

d

**−**

n

. ?A

??A

**Cut**

n

D**−** D+

**X**

?C**− ?B−** ! A+

**−**

**−**

D**−**

D−o ! C**−**

?( D−o ! C)**−**

?A

......

c

?A**−**

?A

**contraction**



?

n

**Cut**



?C**− ?B−** ! A+

**−**

D**−** D+

**X**

?C**− ?B−** ! A+

?D**−** D−o ! C**−**

?( D−o ! C)**−**

?D**−** ?( D−o ! C)**− ?B−**!! A+

**X**

**Cut**

......

**Cut**

?D**−** D−o ! C**−**

?( D−o ! C)

?D**−** ?( D−o ! C)**− ?B−**!! A+

?A**−**

d

??A**−**

?A**−**

d

??A**−**

n

n

**X**

D**−** D+ ?C**− ?B−** ! A+ ...... D**−** D+

**X**

?C**− ?B−** ! A+

?A**−**

?A**−**

?D**−** D−o ! C**−** ?D**−** D−o ! C**−**

?( D−o ! C)**−** ?( D−o ! C)**−**

?D**−** ?( D−o ! C)**− ?B− Cut Cut**

?D**−** ?( D−o ! C)**− ?B−**

D**−** D+

**dereliction**

n

......

n

n

**X**

?C**− ?B−** ! A+

......

**X**

?C**− ?B−** ! A+ ?A**−**

?A**−**

D**−** D+ ?C**−**

D−o ! C**−**

?D**−** ?( D−o ! C)**− ?B−**

**Cut**

. **Cut**



**contraction**

*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*



A

A A

?

?A

?

??A



A A

? ?

A

.... ?

?A ?A ?A

?

??A

*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-Uni cation on Intuitionistic Proof Nets (Extended Abstract). Proceedings of the 4th Asian Symposium on Computer Mathematics, pp 138-141, World Scienti c, 2000.*

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

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

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