プログラミング 12－4
（1996. 11. 1）

Nondeterministic Linear Logic

Satoshi Matsuoka

Department of Electrical and Computer Engineering, Faculty of Engineering,
Nagoya Institute of Technology, Gokiso, Showa-Ku, 466, Japan.
matsuoka@juno.ics.nitech.ac.jp

Abstract. In this paper, we introduce Linear Logic with a nondeterministic facility, which has a self-dual additive connective. In the system proof net technology is available in a natural way. The important point is that nondeterminism in the system is expressed by the process of normalization, not by proof search. Moreover we can incorporate the system into Light Linear Logic and Elementary Linear Logic developed by J.-Y.Girard recently: Nondeterministic Light Linear Logic and Nondeterministic Elementary Linear Logic are defined in a very natural way.

1 Introduction

So far (untyped or typed) lambda calculi with the facility of nondeterminism have been studied: recently e.g., in [Aba94, DCLP93]. For example, in [DCLP93] nondeterminism is represented by using union type, while parallelism by using intersection type: this means that nondeterminism corresponds to the logical connective "or" and parallelism to "and". Further this means that nondeterminism and parallelism are dual notions each other. Basically other researchers similarly classify nondeterminism and parallelism. In this paper, we advocate that nondeterminism and parallelism are not dual notions. For this we use the framework of Linear Logic [Gir87]. In Linear Logic, usual logical connectives are classified into two: multiplicative and additive connectives. Our advocacy is as follows:

- Nondeterminism $\Longrightarrow$ Additive.
- Parallelism $\Longrightarrow$ Multiplicative.

Already it has been pointed out that the multiplicative connectives are deeply related to parallelism since the appearance of [Gir87]. Here we point out that the additive connectives are deeply related to nondeterminism. We incorporate nondeterminism facility into the framework of Linear Logic by introducing new additive connective $\triangle$ (nondeterministic with), which is self-dual. In the framework, nondeterminism is represented by reduction of cut between two $\triangle$: by the reduction of $\triangle$ from one proof net two proof nets are obtained. By using $\triangle$ we can define Nondeterministic Light Linear Logic and Nondeterministic Elementary Linear Logic in a very natural way. Our advocacy has not been advocated before as far as we know. Also I believe that such a classification contributes to studies w.r.t. relationship between Linear Logic and Process Calculus.

2 The System

The system NDMALL is usual MALL (the multiplicative additive fragment of Linear Logic) with $\triangle$ (nondeterministic with). The connective has arity 2 (hence in NDMALL $A\triangle B$ is accepted as a formula if A and B are NDMALL formulas). The negation of $A\triangle B$ is defined as follows:

$$(A\triangle B)^{\perp} \equiv_{\mathrm{def}} A^{\perp}\triangle B^{\perp}$$

The inference rules for NDMALL are the same as MALL except for the following rule:

$$\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \triangle B} \text{ (NDWITH)}$$

The notion of proofs (in sequent calculus) of NDMALL is defined in usual manner. Obviously the connective $\triangle$ belongs to additives. In practice, the connective does not occur in conclusions of NDMALL proofs: if it occurs in them, then it behaves like & in completely the same manner. Hence we can assume that $\triangle$ does not occur in cut free NDMALL proofs. We omit cut elimination procedure for $\triangle$ in NDMALL sequent calculus. But we will introduce it in NDMALL proof nets in the following section.

3 NDMALL proof nets

First we shall define NDMALL proof structures, which are basically the same as them in [Gir95b] except for connective $\triangle$. Simply by formulas we mean NDMALL formulas. Note that to each $\triangle$-link L an eigenweight p_L is assigned.

Definition 1. A link L is an $n+m$-tuple of formulas with a type: $\dfrac{P_1, \ldots, Pn}{Q_1, \ldots, Q_m} L$
The type of a link is either ID, Cut, generalized axiom, $\otimes$, $\wp$, &, $\oplus_1$, $\oplus_2$, or $\triangle$. To each type, n, a number of its premises and m, a number of its conclusions are assigned($m, n \geq 0$, $m + n \neq 0$). The links with ID, Cut, generalized axiom, $\otimes, \wp, \&, \oplus_1, \oplus_2$, and $\triangle$ as types have the following forms:

$$\text{ID-links } \frac{}{A \quad A^{\perp}} \quad \text{Cut links } \frac{A \quad A^{\perp}}{\text{Cut}} \quad \text{generalized axiom-links } \frac{}{A_1 \cdots A_n}$$

$$\text{times } \frac{A \quad B}{A \otimes B} \quad \text{par } \frac{A \quad B}{A \wp B} \quad \text{with } \frac{A \quad B}{A \& B} \quad \text{plus} \quad \oplus_1 \ \frac{A}{A \oplus B} \quad \oplus_2 \ \frac{B}{A \oplus B}$$

$$\text{nondeterministic with } \frac{A \quad B}{A \triangle B}$$

Definition 2. To any &-link or $\triangle$-link L with $A \& B$ or $A \triangle B$ as its conclusion, we associate an eigenweight p_L, which is a boolean variable. The intuitive meaning of p_L is the choice $\{l/r\}$ between the premises A and B: $+p_L$ stands for the selection "left", i.e., A and $-p_L$ stands for the selection "right", i.e., B. We use $\epsilon.p_L$ to speak of $+p_L$ or $-p_L$.

Definition 3. A triple $\Theta = (V, E, w)$ is a proof structure if

- (V, E) is a pair such that V is a multiset of formulas and E is a multiset of links between formulas occurring in V.
- w is a function such that
  (i) For each formula A in V, a weight $w(A)$, i.e., a non-zero element of the boolean algebra generated by the eigenweights $p_1, \ldots, p_n$ of the &-links or $\triangle$-links of Θ;
  (ii) For each link L in E, a weight $w(L)$, i.e., a non-zero element of the boolean algebra generated by the eigenweights $p_1, \ldots, p_n$ of the &-links or $\triangle$-links of Θ.

Moreover, the following conditions must be satisfied:

(a) Each formula in V is the premise of at most one link and the conclusion of at least one link. The formulas which are not premises of some link are called the conclusions of Θ;

(b) $w(A) = \sum_{L \text{ has } A \text{ as the conclusion}} w(L)$;
(c) If A is a conclusion of Θ, then $w(A) = 1$;
(d) If u is any weight occurring in Θ, then u is a monomial $\epsilon_1.p_{L_1} \cdots \epsilon_n.p_{L_n}$ of eigenweights and negations of eigenweights;
(e) If u is a weight occurring in Θ and containing $\epsilon.p_L$ then $u \leq w(L)$;
(f) If L is any non ID-link, with premises A and/or B then
  - if L is any of $\otimes, \wp$ and Cut, then $w(L) = w(A) = w(B)$;
  - if L is a $\oplus_1$-link, then $w(L) = w(A)$;
  - if L is a $\oplus_2$-link, then $w(L) = w(B)$;
  - if L is a &-link, then $w(A) = w(L) \cdot p_L$ and $w(B) = w(L) \cdot \neg p_L$ (hence $w(L) = w(A) + w(B)$);
  - if L is a Δ-link, then $w(A) = w(L) \cdot p_L$ and $w(B) = w(L) \cdot \neg p_L$ (hence $w(L) = w(A) + w(B)$).
(g) For any $A \in V$, if the links whose conclusion is A are $L_1, \ldots, L_m$ then for each $1 \leq i, j \leq m$, whenever $i \neq j$, then $w(L_i) \neq w(L_j)$.

Definition 4. Let ϕ be a valuation for a proof structure $\Theta = (V, E, w)$, i.e. a function from the set of eigenweights of Θ to $\{0, 1\}$, which is extended to a function (still denoted ϕ) from the weights of Θ to $\{0, 1\}$. A pair $\phi(\Theta) = (V_0, E_0)$ is the slice by ϕ if V_0 is the restriction to the formulas A in V such that $\phi(w(A)) = 1$ and E_0 is the restriction of E by V_0 where the definition of &-links and Δ-links is changed such that they have exactly one premise and one conclusion.

The definition of the dependencies of the weights and the formulas in proof structures on an eigenweight is the same as that of [Gir95b].

Definition 5. Let ϕ be a valuation of Θ, let p_L be an eigenweight; we say that the weight w (in Θ) depends on p_L (in $\phi(\Theta)$) iff $\phi(w) \neq \phi_L(w)$, where the valuation ϕ_L is defined by:

- $\phi_L(p_L) = \neg(\phi(p_L))$;
- $\phi_L(p_{L'}) = \phi(p_{L'})$ if $L' \neq L$.

A formula A of Θ is said to depend on p_L (in $\phi(\Theta)$), if A is the conclusion of a link L' such that $\phi(w(L')) = 1$ and $\phi_L(w(L')) = 0$.

Definition 6. A switching $\mathcal{S} = (\phi_{\mathcal{S}}, \text{select}_\wp, \text{select}_\&, \text{select}_\Delta)$ of a proof structure Θ consists in:

- A choice of a valuation $\phi_{\mathcal{S}}$ for Θ;
- A function $\text{select}_\wp$ from the set of all $\wp$-links L of $\phi_{\mathcal{S}}(\Theta)$ to $\{l, r\}$ whose element represents a choice for premises of a $\wp$-link.
- A selection $\text{select}_\&$ for each &-link L of $\phi_{\mathcal{S}}(\Theta)$ a formula $\text{select}_\&(L)$, the jump of L, depending on p_L in $\phi_{\mathcal{S}}(\Theta)$. There is always a normal choice of jump for L, namely the premise A of L such that $\phi_{\mathcal{S}}(w(A)) = 1$.
- A selection select_Δ for each Δ-link L of $\phi_{\mathcal{S}}(\Theta)$ a formula $\text{select}_\Delta(L)$, the jump of L, depending on p_L in $\phi_{\mathcal{S}}(\Theta)$. There is always a normal choice of jump for L, namely the premise A of L such that $\phi_{\mathcal{S}}(w(A)) = 1$.

Definition 7. Let $\mathcal{S}$ be a switching of a proof structure Θ;
the graph $\Theta_{\mathcal{S}} = (V_{\mathcal{S}}, E_{\mathcal{S}})$ corresponding to $\mathcal{S}$ consists in:

- the vertices $V_{\mathcal{S}}$ is V_0 of $\phi_{\mathcal{S}}(\Theta) = (V_0, E_0)$;
- the edges $E_{\mathcal{S}}$ are consists of:
  1. the edge between the conclusions for any ID-link of $\phi_{\mathcal{S}}(\Theta)$;

2. the edge between the premises for any Cut-link of $\phi_S(\Theta)$;
3. the edge between the conclusion and the premise for any $\oplus$-links of $\phi_S(\Theta)$;
4. the edges between the left premise and the conclusion, and between the right premise and the conclusion for any $\otimes$-link of $\phi_S(\Theta)$;
5. the edge between the the premise (left or right) selected by $\text{select}_{\wp}(L)$ and the conclusion of any $\wp$-links L of $\phi_S(\Theta)$;
6. the edge between the jump $\text{select}_{\&}(L)$ of L and the conclusion for any &-link L.
7. the edge between the jump $\text{select}_{\Delta}(L)$ of L and the conclusion for any Δ-link L.

Definition 8. A proof structure Θ is said to be a proof net when for all switching $\mathcal{S}$, the graph Θ_S is connected and acyclic.

The removal of a link of a proof structure Θ in NDMALL is defined in the same manner as [Gir95b] except for Δ-links. Here the definition of the removal for Δ-links is only added.

Definition 9. • If L is a Δ-link with premises A and B such that $w(L)$=1 and L is a conclusion of Θ, and $\Gamma, A\Delta B$ is the set of conclusions of Θ. The removal of L consists in first removing the conclusion $A\Delta B$ and the link L (to get Θ') and then forming two proof structures Θ_A and Θ_B:
  - ⋆ In Θ' make the substitution $p_L = 1$, and keep only those links L' whose weight is still non-zero, together with the premises and conclusions of such links: the result is by definition Θ_A, a proof structure with conclusions Γ, A.
  - ⋆ In Θ' make the substitution $p_L = 0$, and keep only those links L' whose weight is still non-zero, together with the premises and conclusions of such links: the result is by definition Θ_B, a proof structure with conclusions Γ, B.

Definition 10. A proof structure Θ is sequentializable when it can be reduced, by iterated removal of terminal links, to identity links.

The proof of the following theorem is completely the same as that of [Gir95b] which uses empire for each valuation and each formula, since in fixed proof nets Δ-links behave in the same manner as &-links. However the behavior of Δ-link in cut elimination is different from that of & which is defined in the next section.

4 Lazy Cut Elimination in NDMALL

Definition 1. A cut-link L is ready if

- $w(L) = 1$ and;
- If the premises of L are A and $A^{\perp}$ then both A and $A^{\perp}$ are the conclusion of exactly one link.

Definition 2 (lazy cut elimination). Let L_0 be a ready cut in a proof net Θ, whose premises $B\Delta C$ and $B^{\perp}\Delta C^{\perp}$ are the respective conclusions of links L and L'. Then we define the contractums Θ' and Θ'' of redex Θ when reducing L_0 in Θ.

- If L is a Δ-link (with premises B and C) and L' is a Δ-link (with premises $B^{\perp}$ and $C^{\perp}$), then Θ' and Θ'' are obtained in three steps (the reduction is called Δ-reduction):
  how to get Θ' (resp. Θ''):
  First we remove in Θ the formulas $B\Delta C$ and $B^{\perp}\Delta C^{\perp}$ as well as L_0, L and L'; then we replace the eigenweights p_L and p'_L by 1 (resp. 0) and keep only those

formulas and links that still have a nonzero weight: therefore B(resp. C) and $B^{\perp}$ (resp. $C^{\perp}$) remain with weight 1 whereas C (resp. B) and $C^{\perp}$ (resp. $B^{\perp}$) disappears; finally we add a cut between B (resp. C) and $B^{\perp}$ (resp $C^{\perp}$), and then get Θ' (resp. Θ'').

Proposition 3. *If Θ' is obtained from a proof net Θ by lazy cut elimination, then Θ' is a proof net and has the same conclusions as Θ.*

Proof. Similar as [Gir95b].

Proposition 4. *By lazy cut elimination, any MALL proof net is reduced to a unique normal form (which contains ready cuts) in linear time of its size.*

5 Nondeterministic Light Linear Logic

In [Gir95c], it is shown that (1) any p-time Deterministic Turing Machine are representable in Light Linear Logic (for short LLL) and (2) under the condition of bounded depth any LLL proof net is reduced to a normal form in p-time of its size. In this section we show that it is shown that (1') any p-time Nondeterministic Turing Machine are representable in Nondeterministic Light Linear Logic (for short NDLLL) and (2') under the condition of bounded depth any NDLLL proof net is reduced to a normal form by lazy cut elimination in p-time of its size. The system NDLLL is obtained from LLL by adding the inference rule (NDWITH) in Section 2. It is not difficult to show (2') if we follow Girard's proof for LLL, since any ND-MALL proof net is reduced a normal form by lazy cut elimination in linear time of its size (Proposition 4) and Δ connective does not cooperate with exponential reduction.
In order to prove (1'), we only show any move (transition) relation of Nondeterministic Turing Machine is representable in NDMALL. Let a Nondeterministic Turing Machine be M. Let Σ be the set of the symbols used in M and Q be the set of the states used in M. Let p be the number of the symbols used in M, i.e., the cardinal of Σ and q be the number of the states used in M, i.e., the cardinal of Q. The move relation R of M is represented as a subset of $(\Sigma \times Q) \times (\Sigma \times Q \times \{\leftarrow, \rightarrow\})$. Then it is sufficient to represent the move function by a NDLLL proof net with $\mathbf{bool}^{p\times q} \multimap \mathbf{bool}^{p\times q\times 2}$ as the conclusion, where $\mathbf{bool}^{k} = \forall X.\S(\overbrace{X\&\cdots\&X}^{k} \multimap X)$. Since it is obvious that the set $(\Sigma \times Q)$ is represented by $\mathbf{bool}^{p\times q}$ and $(\Sigma \times Q \times \{\leftarrow, \rightarrow\})$ by $\mathbf{bool}^{p\times q\times 2}$ and moreover, it is not difficult to construct proof nets with $\mathbf{bool}^{p} \otimes \mathbf{bool}^{q} \multimap \mathbf{bool}^{p\times q}$ as the conclusion and with $\mathbf{bool}^{p\times q\times 2} \multimap \mathbf{bool}^{p} \otimes \mathbf{bool}^{q} \otimes \mathbf{bool}^{2}$ as the conclusion by using a general version of D in Section 11.3 in [GLT89]. Let m be $\max\{|\{(y,t,d) : (x,s,(y,t,d)) \in R\}| : x \in \Sigma, s \in Q\}$. The following NDLL proof corresponds to the intended proof

net:

$$\begin{array}{ccc} \langle 1 \rangle & & \langle p\times q \rangle \\ \vdots & \cdots & \vdots \\ \overbrace{X\&\cdots\&X}^{p\times q\times 2} \vdash X\Delta\cdots\Delta X & \vdots & \overbrace{X\&\cdots\&X}^{p\times q\times 2} \vdash X\Delta\cdots\Delta X \end{array}$$

$$\dfrac{\dfrac{\cdots}{\overbrace{X\&\cdots\&X}^{p\times q\times 2} \vdash \overbrace{(X\Delta\cdots\Delta X)\&\cdots\&(X\Delta\cdots\Delta X)}^{p\times q}}(\&) \qquad \dfrac{X\vdash X \quad \cdots \quad X\vdash X}{\underbrace{\overbrace{X\Delta\cdots\Delta X}}_{m} \vdash X}(\Delta)}{\overbrace{\overbrace{(X\Delta\cdots\Delta X)}^{m}\&\cdots\&\overbrace{(X\Delta\cdots\Delta X)}^{m}}^{p\times q} \multimap \overbrace{(X\Delta\cdots\Delta X)}^{m} \vdash \overbrace{X\&\cdots\&X}^{p\times q\times 2}\multimap X}$$

$$\dfrac{\overbrace{\overbrace{(X\Delta\cdots\Delta X)}^{m}\&\cdots\&\overbrace{(X\Delta\cdots\Delta X)}^{m}}^{p\times q} \multimap \overbrace{(X\Delta\cdots\Delta X)}^{m} \vdash \overbrace{X\&\cdots\&X}^{p\times q\times 2}\multimap X}{\S(\overbrace{\overbrace{(X\Delta\cdots\Delta X)}^{m}\&\cdots\&\overbrace{(X\Delta\cdots\Delta X)}^{m}}^{p\times q} \multimap \overbrace{(X\Delta\cdots\Delta X)}^{m}) \vdash \S(\overbrace{X\&\cdots\&X}^{p\times q\times 2}\multimap X)}(\S)$$

$$\dfrac{\dfrac{\dfrac{\S(\cdots) \vdash \S(\overbrace{X\&\cdots\&X}^{p\times q\times 2}\multimap X)}{\mathbf{bool}^{p\times q} \vdash \S(\overbrace{X\&\cdots\&X}^{p\times q\times 2}\multimap X)}(\exists)}{\mathbf{bool}^{p\times q} \vdash \mathbf{bool}^{p\times q\times 2}}(\forall)}{\vdash \mathbf{bool}^{p\times q} \multimap \mathbf{bool}^{p\times q\times 2}}(\multimap)$$

From what precedes the following theorem is proved.

Theorem 1. *Any p-time Nondeterministic Turing Machine are representable in Nondeterministic Light Linear Logic.*

It is obvious that in the context of Elementary Linear Logic, the same theorem is proved.

References

[Aba94] M. Abadi. A Semantics for Static Type Inference in a Nondeterministic Language. *Information and Computation*, 109:300–306, 1994.

[DCLP93] M. Dezani-Ciancaglini, U. de'Liguoro and A. Piperno. Filter Models for a Parallel and Non Deterministic λ-calculus. LNCS 711, 1993.

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

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. Proofs and Types. Cambridge University Press, 1989.

[Gir95b] J.-Y. Girard. Proof-nets: the parallel syntax for proof-theory. In Ursini and Agliano, editors, *Logic and Algebra, New York, Marcel Dekker*, 1995.

[Gir95c] J.-Y. Girard. Light Linear Logic. Available by ftp anonymous on **lmd.univ-mrs.fr**, in **pub/girard**, 1995.

This article was processed using the LaTeX macro package with LLNCS style