An Introduction to Linear Logic: Expressiveness and Phase Semantics

Mitsuhiro Okada
Department of Philosophy, Keio University

Abstract

We first point out some nature of linear logic, in comparison with traditional logics, in Introduction (§1), then give the syntax and the intuitive meaning of the syntax in §2. We give phase semantics of linear logic and a phase semantic proof for the completeness and cut-elimination theorems (at once) in §3. We formulate a concurrent process calculus as a conservative subsystem of linear logic and give some examples of concurrent process specification by linear logic in §4. In §5 we give a simplified phase semantics and the completeness for a fragment of linear logic, in which some concurrent process models such as Petri nets can be formulated.

1 Introduction

In the traditional logic, either in the classical logic or the intuitionistic logic, the following logical inference is admitted:

$$\frac{C \longrightarrow A \quad C \longrightarrow B}{C \longrightarrow A \wedge B} \ .$$

Here, $A \longrightarrow B$ is read as "If A then B" and $\wedge$ is read as "and". Then, the above inference says: From the two promises "If C then A" and "If C then B", one can conclude "If C then A and B". This is obviously true for the usual mathematical reasoning. For example,

$$\frac{f(x) < a \longrightarrow b < x \quad f(x) < a \longrightarrow x < c}{f(x) < a \longrightarrow b < x \text{ and } x < c} \ .$$

However, when we try to apply this inference rule to the two promises; "If one has one dollar then one gets a chocolate package" and "If one has one dollar then one gets a candy package", then one may have the following inference:

$$\frac{\text{one has \$1} \longrightarrow \text{one gets a chocolate} \quad \text{one has \$1} \longrightarrow \text{one gets a candy}}{\text{one has \$1} \longrightarrow \text{one gets a chocolate and a candy}} \ .$$

A naive reading of this inference leads to a wrong conclusion "If one has one dollar then one gets both a chocolate package and a candy package". In fact, the following are implicitly assumed when the traditional logical rules are applied to some statements; (1) the statements are independent of temporality, i.e., the logic can treat only "eternal" knowledge which is independent of time. (2) the logical implication "$\longrightarrow$" is independent of any consumption relation or any causal relation. These assumptions are appropriate when we treat the ordinary mathematics. Hence, the traditional logical inferences can be used for mathematical reasoning in general. However, when we would like to treat concurrency-sensitive matters or the resource-consumption relation we need to be careful to apply the logical inference rules. The above example gives one of such situations. In particular, when we would like to study the mathematical structures of information or computation in computer science, information science, etc., we often need to elaborate the traditional logical inferences since concurrency-sensitive setting and concepts for consumption of computational resources or of resources for information processing are often very essential in computer science and the related fields. Linear logic proposed by Girard is considered one of the basic logical systems which would provide a logical framework for such a new situation occurring in computer science and its related fields [1] . For example, instead of the traditional logical connective $\wedge$ ("and"), linear logic provides two different kinds of logical connectives $\otimes$ and &, where $A \otimes B$ means "A and B hold in parallel (at the same time)" while $A \,\&\, B$ means "either A or B can be chosen to hold (as you like) but only one of them at once". The traditional logical implication $\longrightarrow$ is replaced by the linear implication $\multimap$, where $A \multimap B$ means "by the consumption of A, B holds". The logical inference rule for $\otimes$ and & are;

$$\frac{C \multimap A \quad D \multimap B}{C, D \multimap A \otimes B}\ , \qquad \frac{C \multimap A \quad C \multimap B}{C \multimap A \,\&\, B}\ .$$

When we apply "If one has \$1 then one gets a chocolate package" and "If one has \$1 then one gets a candy package" to the two promises of the left inference rule for $\otimes$, then we can conclude "one has $\{\$1, \$1\} \multimap$ one gets (a chocolate $\otimes$ a candy)", which means "If one has two \$1's (namely, \$2) then one gets both a chocolate package and a candy package at the same time", and if we apply the same two promises to the right rule for &, we can conclude "one has $\$1 \multimap$ one gets (a chocolate & a candy)", which means "If one has \$1 then one gets either one of a chocolate package and a candy package as you like".

On the other hand, the infinite amount of a resource of A is expressed as $!A$, with the help of modal operator ! in linear logic. ($!A$ is such a resource that one can consume A as many times as one wants without any loss of $!A$.) By using this modal operator ! one can express the traditional logical truth (i.e., eternal truth) inside the framework of linear logic. Hence, linear logic contains the traditional logic (with the help of modal operator), and linear logic is considered a *refined*

[1] There are some other approaches in which the traditional first order logic is refined in order to capture actions and changes of states. Situation calculus proposed by J. McCarthy[22] is one of such approaches.

form, or a *fine-grained* of the traditional logic, rather than a logic *different* from the traditional logic.

The following table shows some correspondences between the logical connectives of linear logic and the traditional logical connectives/rules.

Linear Logic	Traditional Logics
$\otimes$ (multiplicative "and") & (additive "and")	$\wedge$ ("and")
$\parr$ (multiplicative "or") $\oplus$ (additive "or")	$\vee$ ("or")
$A^{\perp}$ (linear negation)	$\neg A$ (traditional negation)
$\multimap$ (linear implication)	$\rightarrow$ (traditional implication)
$!A$ ("of-course" modality) $?A$ ("why-not" modality)	structural rules (weakening and contraction)
1 ("one") $\top$ ("top")	$\top$ ("top")
$\perp$ ("bottom") 0 ("zero")	$\perp$ ("bottom")

As seen in the above example, the traditional logical connective $\wedge$ ("and") splits into two different connectives, $\otimes$, &, in linear logic, and the traditional logical connective $\vee$ ("or") into two different connectives, $\parr$, $\oplus$. The traditional structural inference rules (which are explicit in Gentzen's Sequent Calculus formulation of traditional logics) are now represented by the modality connectives !, ?. The meaning of these logical connectives of linear logic will be given with the list of linear logic inference rules in the next section.

2 Syntax of Linear Logic

2.1 Syntax of Linear Logic and Naive Operational Semantics

We introduce the vocabularies of the language of Linear Logic as follows;

(1) logical connectives; $\otimes$ ("multiplicative-and" or "tensor-product"),
$\parr$ ("multiplicative-or" or "par"),
& ("additive-and" or "with"),
$\oplus$ ("additive-or" or "plus"),
$A^{\perp}$ (the "linear negation" of A),
$\multimap$ ("linear implication"),
! ("bang" or "of course"), ? ("why not")

(2) logical constants; 0 , 1 , $\top$, $\perp$

(3) propositional variables; $P, Q, R \ldots, P_0, P_1, P_2, \ldots$

The formulas are constructed from logical constants and/or propositional variables by logical connectives, as follows;

(1) If A is a logical constant or propositional variable, then A is a formula.
(2) if A is a formula, then so is $(A^{\perp})$.
(3) if A and B are formulas, then so are $(A \otimes B)$, $(A \parr B)$, $(A \multimap B)$, $(A\&B)$, $(A \oplus B)$.

The outermost parenthesis is often deleted. For example, $(P \otimes 1) \multimap (\perp \parr (Q \oplus R))$ is a formula. We use meta-symbols $A, B, \ldots$ to express formulas. Finite sequence of formulas (possibly the empty sequence) are denoted by $\Gamma, \Delta, \ldots$. When Γ is $A_1, \ldots, A_n$, by $!\Gamma$ we mean $!A_1, \ldots, !A_n$, by $?\Gamma$ we mean $?A_1, \ldots, ?A_n$, and by $\Gamma^{\perp}$ we mean $A_1^{\perp}, \ldots, A_n^{\perp}$. $((\cdots((A_1 \otimes A_2) \otimes A_3) \otimes \cdots) \otimes A_n)$ is abbreviated by $A_1 \otimes \cdots \otimes A_n$. We may also delete some parentheses of a formula if there is no ambiguity from the context. A sequent is an expression of the form $\Gamma \vdash \Delta$.

We introduce the notion of inference rule. There are two kinds of inference rules;

$$\frac{\Gamma_1 \vdash \Delta_1}{\Gamma' \vdash \Delta'} \quad , \qquad \frac{\Gamma_1 \vdash \Delta_1 \quad \Gamma_2 \vdash \Delta_2}{\Gamma' \vdash \Delta'} \quad .$$

The former has only one upper sequent $\Gamma_1 \vdash \Delta_1$, while the latter has two upper sequents $\Gamma_1 \vdash \Delta_1$ and $\Gamma_2 \vdash \Delta_2$. Both have only one lower sequent $\Gamma' \vdash \Delta'$. We also consider a special kind of inference rules for which there is no upper sequent. Such a special kind of inference rule is called an "axiom sequent".

The following are the inference rules for (Classical) Linear Logic.

Definition 1 (Inference rules for Classical Linear Logic) Below, A and B represent arbitrary formulas and Γ, Δ, Γ', Δ' represent arbitrary (finite) sequence of formulas, including the case of empty sequence.

- **Axiom sequent**

  Logical axiom sequent

$$A \vdash A$$

  Logical constants

$$\vdash 1 \qquad\qquad \Gamma \vdash \Delta, \top$$

$$\perp \vdash \qquad\qquad 0, \Gamma \vdash \Delta$$

- **Rules for constants**

$$\frac{\Gamma \vdash \Delta}{1, \Gamma \vdash \Delta} \qquad\qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \perp}$$

- **Exchange**

Exchange-left

$$\frac{\Gamma, A, B, \Gamma' \vdash \Delta}{\Gamma, B, A, \Gamma' \vdash \Delta}$$

Exchange-right

$$\frac{\Gamma \vdash \Delta, A, B, \Delta'}{\Gamma \vdash \Delta, B, A, \Delta'}$$

- **Cut-rule**

$$\frac{\Gamma \vdash \Delta, A \quad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

- **Multiplicative**

$\otimes$-left

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

$\parr$-left

$$\frac{A, \Gamma \vdash \Delta \quad B, \Gamma' \vdash \Delta'}{A \parr B, \Gamma, \Gamma' \vdash \Delta, \Delta'}$$

$\otimes$-right

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma' \vdash \Delta', B}{\Gamma, \Gamma' \vdash \Delta, \Delta', A \otimes B}$$

$\parr$-right

$$\frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \parr B}$$

- **Additive**

&-left

$$\frac{A, \Gamma \vdash \Delta}{A \,\&\, B, \Gamma \vdash \Delta}$$

$$\frac{B, \Gamma \vdash \Delta}{A \,\&\, B, \Gamma \vdash \Delta}$$

&-right

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \,\&\, B}$$

$\oplus$-left

$$\frac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \oplus B, \Gamma \vdash \Delta}$$

$\oplus$-right

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, A \oplus B}$$

$$\frac{\Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \oplus B}$$

- **Linear Implication**

$\multimap$-left

$$\frac{\Gamma \vdash \Delta, A \quad B, \Gamma' \vdash \Delta'}{A \multimap B, \Gamma, \Gamma' \vdash \Delta, \Delta'}$$

$\multimap$-right

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

- **Linear Negation**

$^\perp$-left

$$\frac{\Gamma \vdash \Delta, A}{A^\perp, \Gamma \vdash \Delta}$$

$^\perp$-right

$$\frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, A^\perp}$$

- **Modality**

?-left (promotion)

$$\frac{A, !\Gamma \vdash ?\Delta}{?A, !\Gamma \vdash ?\Delta}$$

!-right (promotion)

$$\frac{!\Gamma \vdash ?\Delta, A}{!\Gamma \vdash ?\Delta, !A}$$

!-left
(dereliction-left)

$$\frac{A, \Gamma \vdash \Delta}{!A, \Gamma \vdash \Delta}$$

?-right
(dereliction-right)

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, ?A}$$

(contraction-left)

$$\frac{!A, !A, \Gamma \vdash \Delta}{!A, \Gamma \vdash \Delta}$$

(contraction-right)

$$\frac{\Gamma \vdash \Delta, ?A, ?A}{\Gamma \vdash \Delta, ?A}$$

(weakening-left)

$$\frac{\Gamma \vdash \Delta}{!A, \Gamma \vdash \Delta}$$

(weakening-right)

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, ?A}$$

A proof is a tree structure where each leaf (i.e., each top node) is an instance of axiom sequent and each inner node is an instance of inference rule. A sequent $\Gamma \vdash \Delta$ is said "provable" if there exists a proof whose last sequent (i.e. the root) is $\Gamma \vdash \Delta$. We says "A is provable" if sequent $\vdash A$ is provable.

The following are examples of a proof of $(A \otimes B)^{\perp} \multimap ((A^{\perp}) \parr (B^{\perp}))$ and a proof of $(!A) \otimes (!B) \vdash !(A \otimes B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash A, A^{\perp}}\,{\perp}\text{-}right}{\vdash A^{\perp}, A}\,exchange\text{-}right \quad \dfrac{\dfrac{B \vdash B}{\vdash B, B^{\perp}}\,{\perp}\text{-}right}{\vdash B^{\perp}, B}\,exchange\text{-}right}{\vdash A^{\perp}, B^{\perp}, A \otimes B}\,\otimes\text{-}right}{(A \otimes B)^{\perp} \vdash A^{\perp}, B^{\perp}}\,{\perp}\text{-}left}{(A \otimes B)^{\perp} \vdash (A^{\perp}) \parr (B^{\perp})}\,\parr\text{-}right}{\vdash (A \otimes B)^{\perp} \multimap ((A^{\perp}) \parr (B^{\perp}))}\,\multimap\text{-}right$$

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{!A \vdash A}\,!\text{-}left \quad \dfrac{B \vdash B}{!B \vdash B}\,!\text{-}left}{!A, !B \vdash A \otimes B}\,\otimes\text{-}right}{!A, !B \vdash !(A \otimes B)}\,!\text{-}right}{(!A) \otimes (!B) \vdash !(A \otimes B)}\,\otimes\text{-}left$$

Exercise 1 The following are provable in (Classical) Linear Logic. Here, $A \equiv B$ means $(A \multimap B) \otimes (B \multimap A)$. Give a proof for each.

(1)	$\vdash A^{\perp\perp} \equiv A$	(Double negation law)
(2)	$\vdash A \parr A^{\perp}$	(Excluded middle)
(3)	$\vdash (A \otimes B)^{\perp} \equiv (A^{\perp}) \parr (B^{\perp})$	(De Morgan law)
(4)	$\vdash (A \parr B)^{\perp} \equiv (A^{\perp}) \otimes (B^{\perp})$	(De Morgan law)
(5)	$\vdash (A\&B)^{\perp} \equiv (A^{\perp}) \oplus (B^{\perp})$	(De Morgan law)
(6)	$\vdash (A \oplus B)^{\perp} \equiv (A^{\perp})\&(B^{\perp})$	(De Morgan law)
(7)	$\vdash (!A)^{\perp} \equiv ?(A^{\perp})$	(De Morgan law)
(8)	$\vdash (?A)^{\perp} \equiv !(A^{\perp})$	(De Morgan law)
(9)	$\vdash A \otimes (B \oplus C) \equiv (A \otimes B) \oplus (A \otimes C)$	(Distributive law)
(10)	$\vdash A \parr (B\&C) \equiv (A \parr B)\&(A \parr C)$	(Distributive law)
(11)	$\vdash (!A) \otimes (!B) \equiv !(A\&B)$	
(12)	$\vdash (?A) \parr (?B) \equiv ?(A \oplus B)$	
(13)	$\vdash A \multimap B \equiv A^{\perp} \parr B$	

In particular, when Δ consists of at most one formula, a sequent $\Gamma \vdash \Delta$ is sometimes called an intuitionistic sequent. Intuitionitsic Linear Logic is defined in the same way as Linear Logic. We do not include $\parr$ nor ? as a logical connective in Intuitionistic Linear Logic. The sequents are restricted to the intuitionistic sequents. Here, due to the restriction of the right hand-side of a sequent, exchange-right is not needed. And due to the lack of $\parr$ and ?, we do not need $\parr$-left, $\parr$-right, ?-left and ?-right in Intuitionistic Linear Logic.

The following logical rules (for intuitionistic sequents) are the inference rules of Intuitionistic Linear Logic. We list the right-introduction rules for each logical connective and, at the same time, we give a natural meaning of each logical connective induced from each right-rule (by the operational meaning of each rule), in a parenthesis. (We do not include the axioms and rules for logical constants for simplicity.)

Definition 2 (Inference rules for Intuitionistic Linear Logic)

- $\otimes$-right rule

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$$

(If A can be generated by using resource Γ and if B can be generated by using resource Δ, then $A \otimes B$ (A and B in parallel) can be generated by using resource Γ and Δ.)

- &-right rule

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B}$$

(If A can be generated by using resource Γ and if B can be generated by using resource Γ , then $A \,\&\, B$ (whichever A or B) can be generated by using resource Γ.)

- $\oplus$-right rule

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B}$$

(If A can be generated by resource Γ , then $A \oplus B$ (either A or B) can be generated by using resource Γ.)

- $\oplus$-right rule

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B}$$

(If B can be generated by resource Γ , then $A \oplus B$ (either A or B) can be generated by using resource Γ.)

- $\multimap$-right rule

$$\frac{A, \Gamma \vdash B}{\Gamma \vdash A \multimap B}$$

(If B can be generated by using resource A and Γ, then $A \multimap B$ (the state that B is generated by consuming A) can be generated by using resource Γ.)

Now, $A^{\perp}$ (the linear-negation of A) is defined as $A \multimap \perp$. Then the above $\multimap$-rule induces the following $^{\perp}$-rule.

- $^{\perp}$-right rule

$$\frac{A, \Gamma \vdash \perp}{\Gamma \vdash A^{\perp}}$$

(If $\perp$ (contradiction) can be generated by using resource A and Γ, then $A^{\perp}$ (the state that the contraction is generated by consuming A) can be generated by using resource Γ.)

Next, we list the left rules, (and we show how to justify each left rule in terms of the interpretation of the logical connectives introduced above by the right rules).

- $\otimes$-left rule

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

(Assume $A \otimes B$ and Γ. By the about interpretation of $A \otimes B$, A and B hold in parallel. Hence, A, B, and Γ hold in parallel. On the other hand, we assume by the upper sequent that C can be generated by using A, B and Γ. Therefore, C can be generated.)

- &-left rule

$$\frac{A, \Gamma \vdash C}{A \,\&\, B, \Gamma \vdash C} \qquad \frac{B, \Gamma \vdash C}{A \,\&\, B, \Gamma \vdash C}$$

(For the first rule, we assume by the upper sequent that C can be generated by A and Γ. Assume $A \,\&\, B$ and Γ hold. $A \,\&\, B$ means whichever between A and B can be generated. Hence, in particular, A can be generated. Therefore, A and Γ can be generated, Hence, with the help of the assumption for the upper sequent, C can be generated. For the second rule, the argument is similar.)

- $\oplus$-left rule

$$\frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{A \oplus B, \Gamma \vdash C}$$

(Assume by the upper sequents that C can be generated by A and Γ and that C can be generated by B and Γ. Now we assume $A \oplus B$ and Γ. $A \oplus B$ means either A or B can be generated. We argue by cases; assume that A can be generated. Then A and Γ hold. Hence, by the help of the assumption for the first upper sequent, C can be generated; assume that B can be generated. Then B and Γ hold. Hence, by the help of the assumption for the second upper sequent, C can be generated. Hence, for either case, C can be generated.)

- $\multimap$-left rule

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

(Assume that A can be generated by Γ, and that C can be generated by B and Δ. Assume that $A \multimap B$, Γ, and Δ are generated. There $A \multimap B$ means "B is generated by consuming A". Since Γ is assumed, with the help of the first assumption, A can be generated. Therefore, with $A \multimap B$, B can be generated. Hence, with the help of the second assumption, C can be generated.)

- $^\perp$-left rule

$$\frac{\Gamma \vdash A}{A^\perp, \Gamma \vdash}$$

(If A can be generated by using resource Γ, then $\perp$ (the contradiction) can be generated by using resource $A^\perp$ (not A) and Γ.)

Note that $\Delta \vdash$ is provable if and only if $\Delta \vdash \perp$. Hence, we identify the above lower sequent with $A^\perp, \Gamma \vdash \perp$.

The following rule is called "cut-rule".

- cut-rule

$$\frac{\Gamma \vdash A \quad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}$$

(If A can be generated by using resource Γ and if B can be generated by using resource A and Δ, then B can be generated by using resource Γ and Δ.)

Now we add the following modality rules for ! (bang). We first give the right-rule, then give the three left-rules.

- !-right rule

$$\frac{!A_1, \ldots, !A_n \vdash B}{!A_1, \ldots, !A_n \vdash !B}$$

(We interpret $!A$ as an infinite resource for A, which could be understood as a kind of copy machine to produce arbitrarily many A's including zero A's. Assume that $!A_1, \ldots, !A_n$ hold. Hence, $A_1, \ldots, A_n$ can be generated as one wants. By the assumption for the upper sequent, B can be generated from $!A_1, \ldots, !A_n$. Since those $!A_1, \ldots, !A_n$ can produce the necessary amount for generating B as many times as one wants, one can generate as many B's as one wants, which is the safe effect to generate $!B$.)

- Dereliction

$$\frac{A, \Gamma \vdash C}{!A, \Gamma \vdash C}$$

(Assume that C can be generated from A and Γ. Assume that $!A$ and Γ are generated. $!A$ can produce as many A's as one wants, hence, in particular, one A can be generated from $!A$. Therefore, A and Γ can be generated. Therefore, with the first assumption, C can be generated.)

- Weakening

$$\frac{\Gamma \vdash C}{!A, \Gamma \vdash C}$$

(Assume that C is produced from Γ. Assume that $!A$ and Γ are produced. There, $!A$ can produce as many A's as one wants (including zero A's). Hence, zero A's can be produced from $!A$. Therefore, Γ can be produced from $!A$ and Γ. By the first assumption, C can be produced.)

- Contraction

$$\frac{!A, !A, \Gamma \vdash C}{!A, \Gamma \vdash C}$$

(Assume that $!A$, $!A$ and Γ can produce C. Since any number of A's produced by the two $!A$'s can be produced by a single $!A$. Hence, C can be produced from $!A$ and Γ.)

- Axiom sequent

$$A \vdash A$$

(The justification of this is obvious.)

$A \parr B$ is interpreted as $(A^{\perp} \otimes B^{\perp})^{\perp}$ (*i.e.*, "$A^{\perp}$ and $B^{\perp}$ in parallel" implies the contradiction). $?A$ is interpreted as $(!(A^{\perp}))^{\perp}$. The sequent of the form $A_1, \ldots, A_n \vdash B_1, \ldots B_m$ is interpreted as $A_1 \otimes \cdots \otimes A_n \vdash B_1 \parr \cdots \parr B_m$. Then, the classical linear logic rules can also be justified with the above interpretations.

In terms of the above intuitive (operational) meaning of the linear logical connectives, one may specify a course menu of a restaurant as follows;

Lafont's Restaurant[2]
Today's course menu

((Vegetable Soup $\oplus$ Consommé Soup) & Salad)
$\otimes$ (Fish & Meat) $\otimes$!(Coffee) $\otimes$ (\$2 $\multimap$ (Cake & Ice Cream))

The above means: The customer gets either Soup or Salad as an entry (entrée) as the customer chooses, where, when Soup is chosen, either Vegetable Soup or Consommé Soup will be served depending on the day (which cannot be chosen by the customer). The customer also gets a main dish for which he/she can choose either a fish dish or a meat dish as he/she wishes. He/She also gets Coffee, which can be refilled as many times as he/she wishes (including 0-time, which means that he/she could skip Coffee). He/She also gets a dessert if he/she will pay extra 2 dollars, by which he/she can choose either Cake or Ice Cream.

The Classical Linear Logic is often represented by the one-sided sequent calculus formulation. The Right-Only sequent calculus of Classical Linear Logic is obtained from the above two-sided system by ignoring the left hand side formulas in a sequent and deleting all left-rules. With the absence of the left hand formulas of $^{\perp}$-right rule and of the $^{\perp}$-left rule, one cannot express any rule for $*^{\perp}$ in the one-sided sequent calculus. However, one can consider $P^{\perp}$ as an atomic formula if P is atomic, and $A^{\perp}$ as an abbreviation of its De Morgan dual if A is a complex formula; namely,

[2] This kind of example was first introduced by Y. Lafont in his lecture, as far as the author knows. Here, the example is modified by the author.

$(A \otimes B)^{\perp} \equiv_{def} A^{\perp} \parr B^{\perp}$, $(A \parr B)^{\perp} \equiv_{def} A^{\perp} \otimes B^{\perp}$, $(A \,\&\, B)^{\perp} \equiv_{def} A^{\perp} \oplus B^{\perp}$, $(A \oplus B)^{\perp} \equiv_{def} A^{\perp} \,\&\, B^{\perp}$, $(!A)^{\perp} \equiv_{def} ?A^{\perp}$, $(?A)^{\perp} \equiv_{def} !A^{\perp}$. Then one does not need any inference rules for $A^{\perp}$. Since $A \multimap B$ can be defined as $A^{\perp} \parr B$ in Classical Linear Logic (see Exercise 1 (13)), we do not need the rules for $\multimap$, either. The logical axiom sequent and the cut-rule is expressed as;

$$\vdash A, A^{\perp} \quad , \qquad \frac{\vdash \Gamma, A \quad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta} \quad .$$

It is easily seen that $\Gamma \vdash \Delta$ is provable in the two-sided Classical Linear Logic if and only if $\vdash \Gamma^{\perp}, \Delta$ is provable in the Right-Only Classical Linear Logic. See Appendix 2 for the full list of the Right-Only Sequent Calculus rules.

Similarly, we can also represent Classical Linear Logic as the Left-Only sequent calculus in the obvious manner.

We shall use the Right-Only formulation for our proof of the completeness theorem and the cut-elimination theorem in Section 3, and the Left-Only formulation for specifications of concurrent processes, in Section 4.

3 Phase Semantics, Completeness and Cut-Elimination of Linear Logic

3.1 Phase Semantics

In this subsection, we first introduce phase semantics, due to Girard[9], then in the next subsection we give two fundamental theorems, the completeness theorem and the cut-elimination theorem, at once by the use of phase semantics. The phase semantics is considered the linear logic version of the Tarski-style traditional logical semantics for the classical logic.

Let M be a commutative monoid, namely, M has the unit element 1 and a binary operator $* \cdot *$ such that (1) for any $a, b \in M$, $a \cdot b \in M$, (2) for any $a \in M$, $1 \cdot a = a \in M$, (3) for any $a, b, c \in M$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$, (4) for any $a, b \in M$, $a \cdot b = b \cdot a$. Let $\perp$ be a special subset of M, called *bottom*. We have the definition of the linear negation;

For any $\alpha \subseteq M$, let $\alpha^{\perp}$ be $\{b : \text{for all } a \in \alpha \; a \cdot b \in \perp\} = \{b : \alpha \cdot b \subseteq \perp\}$.

$\perp^{\perp}$ is denoted by $\mathbf{1}$. Let $I = \mathbf{1} \cap J$ where J is a submonoid of M which satisfies the weak idempotent property $\forall a \in J \;\; \{a\}^{\perp\perp} \subseteq \{a \cdot a\}^{\perp\perp}$. In particular, J can be $\{a \in M : aa = a\}$.

The following is easily proved.

Lemma 1 *For any $\alpha \subseteq M, \beta \subseteq M$,*

1. $\alpha \subseteq \alpha^{\perp\perp}$
2. $(\alpha^{\perp\perp})^{\perp\perp} \subseteq \alpha^{\perp\perp}$

3. $\alpha \subseteq \beta \Longrightarrow \alpha^{\perp\perp} \subseteq \beta^{\perp\perp}$

4. $\alpha^{\perp\perp} \cdot \beta^{\perp\perp} \subseteq (\alpha \cdot \beta)^{\perp\perp}$.

$\alpha \subseteq M$ is called a fact iff $\alpha^{\perp\perp} = \alpha$. The set of facts is denoted by D_M (or D if M is already fixed). Then, as easily seen, for any $\alpha \subseteq P$, $\alpha^{\perp\perp}$ is the smallest fact that includes α, and for any facts α and β, $\alpha \cap \beta$ is a fact. $\alpha^{\perp\perp\perp} = \alpha^{\perp}$, hence $\alpha^{\perp}$ is a fact for any $\alpha \subseteq P$. We define the following operators and constants;

$$
\begin{array}{lcl}
\mathbf{1} & = & \perp^{\perp}(= \{1\}^{\perp\perp}), \text{where 1 denotes the unit element of } M. \\
\mathbf{0} & = & \top^{\perp}, \text{ where} \top = M. \\
\alpha \& \beta & = & \alpha \cap \beta \\
\alpha \oplus \beta & = & (\alpha \cup \beta)^{\perp\perp} \\
\alpha \otimes \beta & = & (\alpha \cdot \beta)^{\perp\perp} \\
\alpha \parr \beta & = & (\alpha^{\perp} \cdot \beta^{\perp})^{\perp} \\
!\alpha & = & (I \cap \alpha)^{\perp\perp} \\
?\alpha & = & (I \cap \alpha^{\perp})^{\perp} (= (!\alpha^{\perp})^{\perp}).
\end{array}
$$

Note that the above operators are closed under the facts D_M. $(M, I, \perp, \mathbf{1}, \mathbf{0}, \top, \&, \oplus, \otimes, \parr, !, ?)$ is called a phase space. We also denote this as $(M, I, \perp)$.

On a phase space, as in Girard[9], the interpretation A^* of a formula A is defined to be a fact in the following way when an assignment (a valuation) φ of facts for (propositional) variables occurring in A is given. We call this value A^* the inner-value of A through this paper:

$$
\begin{array}{lcl}
R^* & = & \varphi(R) \text{ for assignment (valuation) } \varphi : A\text{-}Form \longrightarrow D, \\
 & & \quad \text{where } A\text{-}Form \text{ stands for the set of atomic formulas.} \\
(A^{\perp})^* & = & (R^*)^{\perp} \\
(A \& B)^* & = & A^* \& B^* \\
(A \oplus B)^* & = & A^* \oplus B^* \\
(A \otimes B)^* & = & A^* \otimes B^* \\
(A \parr B)^* & = & A^* \parr B^* \\
(?A)^* & = & ?(A^*) \\
(!A)^* & = & !(A^*).
\end{array}
$$

It is obvious that any inner value A^* is a fact. [3]

A formula A is said to be *true* if $1 \in A^*$, where 1 is the unit element of underlying monoid M. For a given phase space $(M, I, \perp)$ and a given assignment φ, $(M, I, \perp, \varphi)$ is called a phase model.

Since the Right-Only one-sided sequent calculus formulation is simpler than the two-sided formulation and usually used for the syntax of classical linear logic, we shall use the Right-Only formulation in the rest of this Section. Recall that $A^{\perp}$

[3] One can easily extend these interpretations to the case of the first order quantifiers (by interpreting these as additive operators), as usual. For the higher order extension, see Okada[25][26].

is defined as the De Morgan dual (except for the case where A is atomic), and $(A^{\perp})^* = (A^*)^{\perp}$ is derived from $(R^{\perp})^* = (R^*)^{\perp}$ with other definition of the inner value above. In the same way as in Girard[9], we have

Theorem 1 (Soundness Theorem (Girard[9])) *If a formula A is provable in (first order) linear logic, then A is true (namely, $1 \in A^*$) for any classical phase model. More generally, if a sequent $\vdash A_1, \ldots, A_n$ is provable, then $A_1 \parr \cdots \parr A_n$ is true, namely $1 \in A_1^* \parr \ldots \parr A_n^*$, or equivalently, $A_1^{*\perp} \cdot \cdots \cdot A_n^{*\perp} \subseteq \perp$ for any phase model.*

Proof. The proof is carried out by induction on the length (i.e., the number of the inference rules) of a proof of $\vdash A_1, \ldots, A_n$. (We use the properties in Lemma 1 repeatedly without explicit mentioning.) In the proof below, when Γ or Δ is empty, Γ^* or Δ^* is interpreted as $\perp$ (hence, $\Gamma^{*\perp}$ or $\Delta^{*\perp}$ is interpreted as $\perp^{\perp} = \mathbf{1}$.

(Base Case) When the length is 0, namely the proof is composed of one axiom sequent.

(Case 1) When it is of the form $\vdash A, A^{\perp}$. It suffices to show $A^{*\perp} \cdot A^{*\perp\perp} \subseteq \perp$, which is obvious by the definition of $\alpha^{\perp}$.

(Case 2) When it is of the form $\vdash \Delta, \top$. It suffices to show $\Delta^{*\perp} \cdot \top^{*\perp} \subseteq \perp$. since $\top^* = M$, $\Delta^{*\perp} \cdot \top^{*\perp} \subseteq M \cdot \top^{*\perp} = \top^* \cdot \top^{*\perp} \subseteq \perp$.

(Case 3) When it is of the form $\vdash 1$. It suffices to show $1^{\perp} \subseteq \perp$, which is obvious since $1^{\perp} = \perp$ by definition.

(Induction Steps) We prove by cases of the last inference rule of the proof.

(Case 1) When the last inference is $\otimes$-rule of the form;

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \ .$$

By the induction hypothesis, $\Gamma^{*\perp} \cdot A^{*\perp} \subseteq \perp$ and $\Delta^{*\perp} \cdot B^{*\perp} \subseteq \perp$. Hence, $\Gamma^{*\perp} \subseteq A^{*\perp\perp} = A^*$ and $\Delta^{*\perp} \subseteq B^{*\perp\perp} = B^*$. Therefore, $\Gamma^{*\perp} \cdot \Delta^{*\perp} \subseteq A^* \cdot B^*$. Hence, $\Gamma^{*\perp} \cdot \Delta^{*\perp} \cdot (A \otimes B)^{*\perp} = \Gamma^{*\perp} \cdot \Delta^{*\perp} \cdot (A^* \cdot B^*)^{\perp} \subseteq (A^* \cdot B^*)(A^* \cdot B^*)^{\perp} \subseteq \perp$.

(Case 2) When the last inference is $\parr$-rule of the form;

$$\frac{\vdash \Delta, A, B}{\vdash \Delta, A \parr B} \ .$$

By the induction hypothesis, $\Delta^{*\perp} \cdot A^{*\perp} \cdot B^{*\perp} \subseteq \perp$. Hence, $\Delta^{*\perp} \cdot (A \parr B)^{*\perp} = \Delta^{*\perp} \cdot (A^{*\perp} \cdot B^{*\perp}) \subseteq \perp$.

(Case 3) When the last inference is &-rule of the form;

$$\frac{\vdash \Delta, A \quad \vdash \Delta, B}{\vdash \Delta, A\&B} \ .$$

By the induction hypothesis, $\Delta^{*\perp} \cdot A^{*\perp} \subseteq \perp$ and $\Delta^{*\perp} \cdot B^{*\perp} \subseteq \perp$. Hence, $\Delta^{*\perp} \subseteq A^{*\perp\perp} = A^*$ and $\Delta^{*\perp} \subseteq B^{*\perp\perp} = B^*$. Therefore, $\Delta^{*\perp} \subseteq A^* \cap B^*$. Hence, $\Delta^{*\perp} \cdot (A\&B)^{*\perp} = \Delta^{*\perp} \cdot (A^* \cap B^*)^{\perp} \subseteq \perp$.

(Case 4) When the last inference is $\oplus$-rule of the form;

$$\frac{\vdash \Delta, A}{\vdash \Delta, A \oplus B} \ .$$

By the induction hypothesis, $\Delta^{*\perp} \cdot A^{*\perp} \subseteq \perp$. Hence, $\Delta^{*\perp} \subseteq A^{*\perp\perp} = A^*$. Hence, $\Delta^{*\perp} \subseteq A^* \subseteq A^* \cup B^* \subseteq (A^* \cup B^*)^{\perp\perp} = A^* \oplus B^*$. Therefore, $\Delta^{*\perp} \cdot (A^* \oplus B^*)^{\perp} \subseteq \perp$.

(Case 5) When the last inference is ?-rule of the form;

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A} \ .$$

By the induction hypothesis, $\Gamma^{*\perp} \cdot A^{*\perp} \subseteq \perp$. Hence, $\Gamma^{*\perp} \cdot (A^{*\perp} \cap I) \subseteq \perp$. Therefore, $\Gamma^{*\perp} \cdot (A^{*\perp} \cap I)^{\perp\perp} \subseteq \perp$.

(Case 6) When the last inference is ?-rule of the form;

$$\frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A} \ .$$

By the induction hypothesis, $\Gamma^{*\perp} \cdot (A^{*\perp} \cap I) \cdot (A^{*\perp} \cap I) \subseteq \perp$, hence, $\Gamma^{*\perp} \cdot \{aa \mid a \in A^{*\perp} \cap I\} \subseteq \perp$. On the other hand, for any $a \in J$ and for any $m \in M$, $maa \in \perp \Longrightarrow ma \in \perp$. Therefore, $\Gamma^{*\perp} \cdot (A^{*\perp} \cap I) \subseteq \perp$. Hence, $\Gamma^{*\perp} \cdot (?A)^{*\perp} = \Gamma^{*\perp} \cdot (A^{*\perp} \cap I)^{\perp\perp} \subseteq \perp$.

(Case 7) When the last inference is ?-rule of the form;

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A} \ .$$

By the induction hypothesis, $\Gamma^{*\perp} \subseteq \perp$. Hence, $\Gamma^{*\perp} \cdot (A^{*\perp} \cap I) \subseteq \Gamma^{*\perp} \cdot \mathbf{1} \subseteq \Gamma^{*\perp} \subseteq \perp$. Therefore $\Gamma^{*\perp} \cdot (A^{*\perp} \cap I)^{\perp\perp} \subseteq (\Gamma^{*\perp} \cdot (A^{*\perp} \cap I))^{\perp\perp} \subseteq \perp^{\perp\perp} = \perp$.

(Case 8) When the last inference is !-rule of the form;

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A} \ .$$

Let $?\Gamma$ be $?B_1, \ldots, ?B_m$. By the induction hypothesis, $(B_1^{*\perp} \cap I)^{\perp\perp} \cdot \cdots \cdot (B_m^{*\perp} \cap I)^{\perp\perp} \cdot A^{*\perp} \subseteq \perp$. Hence, $(B_1^{*\perp} \cap I) \cdot \cdots \cdot (B_m^{*\perp} \cap I) \subseteq A^{*\perp\perp}$. On the other hand, $(B_1^{*\perp} \cap I) \cdots (B_m^{*\perp} \cap I) \subseteq \mathbf{1} \cdots \mathbf{1} \subseteq \mathbf{1}$. $(B_1^{*\perp} \cap I) \cdots (B_m^{*\perp} \cap I) \subseteq J \cdots J \subseteq J$ (since

J is a submonoid). Therefore, $(B_1^{*\perp} \cap I) \cdot \cdots \cdot (B_m^{*\perp} \cap I) \subseteq A^* \cap \mathbf{1} \cap J = A^* \cap I$. Hence, $(B_1^{*\perp} \cap I) \cdot \cdots \cdot (B_m^{*\perp} \cap I) \cdot (A^* \cap I)^\perp \subseteq \perp$, therefore, $(?B_1)^{*\perp} \cdot \cdots \cdot (?B_m)^{*\perp} \cdot (!A)^{*\perp} \subseteq \perp$.

(Case 9) When the last inference is the cut-rule of the form;

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} \ .$$

By the induction hypothesis, $\Gamma^{*\perp} \cdot A^{*\perp} \subseteq \perp$ and $\Delta^{*\perp} \cdot A^* \subseteq \perp$. Hence, $\Gamma^{*\perp} \subseteq A^{*\perp\perp} = A^*$ and $\Delta^{*\perp} \subseteq A^{*\perp}$. Therefore, $\Gamma^{*\perp} \cdot \Delta^{*\perp} \subseteq A^* \cdot A^{*\perp} \subseteq \perp$.

The cases for exchange rules and the rule for constant $\perp$ are obvious and left to the reader. ■

For the two-sided sequent calculus of classical linear logic, the soundness proof can be carried out in the same way as above, by induction on the length of proof. Here, for a given proof of $B_1, \ldots, B_m \vdash A_1, \ldots, A_n$, we show $1 \in B_1^{*\perp} \parr \cdots \parr B_m^{*\perp} \parr A_1^* \parr \cdots \parr A_n^*$ or equivalently, $B_1^* \cdot \cdots \cdot B_m^* \cdot A_1^{*\perp} \cdot \cdots \cdot A_n^{*\perp} \subseteq \perp$.

3.2 Completeness Theorem and Cut-Elimination Theorem

In this subsection we give a strong form of the completeness theorem of linear logic with respect to the phase semantics. Our Completeness theorem also implies the Cut–elimination theorem.

Theorem 2 (Completeness Theorem (Girard[9])) *If a (first order) formula A is true for any phase model, A is provable in (first order) linear logic.*

On the other hand, we give the following slightly refined form of the above Completeness theorem;

Theorem 3 (Strong Completeness Theorem (Okada[25][26])) *If a formula A is true for any phase model, A is provable in (first order) linear logic without the cut-rule.*

We construct a canonical phase model $(M, I, \perp, \varphi)$ for which truth of a formula A implies its cut-free provability. We take the commutative free monoid generated by the set of formulas and denote it as M. An element of M is a multiset of formulas where multiple occurrences of a formula of the form $?A$ counts only once. For example, $\{A, A, A, ?B, ?B, C\}$ is identified with $\{A, A, A, ?B, C\}$. The construction of the phase space is the same as in Girard[9] in his completeness proof, except that we use *cut-free provability* instead of provability in the definition of $[\![A]\!]$ below. For any formula A (of linear logic), we define $[\![A]\!] = \{\Delta : \vdash_{cf} \Delta, A\}$, where $\vdash_{cf}$ means "cut-free provable". We call $[\![A]\!]$ the *outer value* of A in this paper. $\perp = [\![\perp]\!] = \{\Delta : \vdash_{cf} \Delta, \perp\} = \{\Delta : \vdash_{cf} \Delta\}$; the unit element 1 of M is ϕ (the empty sequence). I is defined as $\{?\Gamma : \Gamma$ is an arbitrary sequence of formulas$\}$, where $?\Gamma$ means $?A_1, \ldots, ?A_n$ if $\Gamma \equiv A_1, \ldots, A_n$. Finally, the assignment φ of the canonical model is defined as $\varphi(R) = [\![R]\!]$ for any atomic R.

Lemma 2 (Main Lemma (Okada[25][26])) *For any formula A, $A^* \subseteq [\![A]\!]$.*

It is easy to see that this *Main Lemma* directly implies the Strong Completeness; if formula A is true, then $\phi \in A^*$. On the other hand, $A^* \subseteq [\![A]\!]$, hence $\phi \in [\![A]\!]$, which means *"A is cut-free provable"*.

By combining this with the soundness theorem, we have

Theorem 4 (The Cut-Elimination Theorem (Girard[9], cf. Okada[25][26])) *If A is provable (with the cut-rule) then it is provable without cut.*

Remark: We can actually prove $A^* = [\![A]\!]$ if we interpret $[\![A]\!]$ as $\{\Gamma : \vdash \Gamma, A$ is provable with the cut-rule $\}$, which was the essential part of the original completeness proof by Girard[9]. On the other hand, we have taken a more restricted interpretation $[\![A]\!] = \{\Gamma : \vdash \Gamma, A$ is provable without the cut-rule $\} = \{\Gamma : \vdash_{cf} \Gamma, A\}$ and show a weaker version of the corresponding lemma (Main Lemma). (However, after having proved the cut-elimination theorem, it can be shown that $A^* = [\![A]\!]$.)

Lemma 3 *If $A^* \subseteq [\![A]\!]$, then $A \in A^{*\perp}$.*

Proof. Assume $A^* \subseteq [\![A]\!]$. Then $A \cdot A^* \subseteq A \cdot [\![A]\!] \subseteq [\![\perp]\!]$. Therefore $A \in A^{*\perp}$. ■

Lemma 4 $[\![A]\!]^{\perp\perp} = [\![A]\!]$ *for any formula A. (Therefore, any outer value is a fact.)*

Proof. Since $[\![A]\!] \subseteq [\![A]\!]^{\perp\perp}$ is trivial, we prove $[\![A]\!]^{\perp\perp} \subseteq [\![A]\!]$. Let $\Gamma \in [\![A]\!]^{\perp\perp}$. By definition, $\forall \Delta \in [\![A]\!]^{\perp} \vdash_{cf} \Gamma, \Delta$. On the other hand, $A \in [\![A]\!]^{\perp}$ is trivial. Hence $\vdash_{cf} \Gamma, A$, therefore $\Gamma \in [\![A]\!]$. ■

Now we prove the Main Lemma.

Proof of Main Lemma. The proof of this is carried by induction on the complexity of formula A.
(Case 1) When A is atomic. $A^* = [\![A]\!]$ by definition.

(Case 2) When A is the form $R^{\perp}$ for atomic R. If $\Gamma \in R^{*\perp}$, then by the definition of $R^{*\perp}$, $R^*\Gamma \subseteq [\![\perp]\!]$. On the other hand, $R^{\perp} \in [\![R]\!] = R^*$. Hence $R^{\perp}, \Gamma \in [\![\perp]\!]$, therefore $\Gamma \in [\![R^{\perp}]\!]$.

(Case 3) When A is of the form $B \otimes C$. By the induction hypothesis, $B^* \cdot C^* \subseteq [\![B]\!] \cdot [\![C]\!]$. On the other hand,

$$\frac{\vdash_{cf} B, \Gamma \quad \vdash_{cf} C, \Delta}{\vdash_{cf} B \otimes C, \Gamma, \Delta} .$$

Hence, $[\![B]\!] \cdot [\![C]\!] \subseteq [\![B \otimes C]\!]$. Therefore, $B^* \cdot C^* \subseteq [\![B \otimes C]\!]$. Since $[\![B \otimes C]\!]$ is a fact, $B^* \otimes C^* = (B^* \cdot C^*)^{\perp\perp} \subseteq [\![B \otimes C]\!]^{\perp\perp} = [\![B \otimes C]\!]$.

(Case 4) When A is of the form $B \parr C$. Assume that $\Gamma \in (B^{*\perp} \cdot C^{*\perp})^{\perp}$. Hence, $\Gamma \cdot (B^{*\perp} \cdot C^{*\perp}) \subseteq [\![\perp]\!]$. On the other hand, by the induction hypothesis, $B^* \subseteq [\![B]\!]$ and $C^* \subseteq [\![C]\!]$. Hence by Lemma 3, $B \in B^{*\perp}$ and $C \in C^{*\perp}$, hence $B, C \in B^{*\perp} \cdot C^{*\perp}$. Therefore, $\Gamma, B, C \in [\![\perp]\!]$. Since

$$\frac{\vdash_{cf} \Gamma, B, C}{\vdash_{cf} \Gamma, B \parr C},$$

it follows $\Gamma \in [\![B \parr C]\!]$.

(Case 5) When A is of the form $B \,\&\, C$. By the induction hypothesis, $B^* \subseteq [\![B]\!]$ and $C^* \subseteq [\![C]\!]$. Hence $B^* \& C^* = B^* \cap C^* \subseteq [\![B]\!] \cap [\![C]\!]$. On the other hand,

$$\frac{\vdash_{cf} B, \Gamma \quad \vdash_{cf} C, \Gamma}{\vdash_{cf} B \& C, \Gamma}.$$

Hence, $[\![B]\!] \cap [\![C]\!] \subseteq [\![B \& C]\!]$. Therefore, the claim holds.

(Case 6) When A is of the form $B \oplus C$. By the induction hypothesis, $B^* \cup C^* \subseteq [\![B]\!] \cup [\![C]\!]$. On the other hand,

$$\frac{\vdash_{cf} B, \Gamma}{\vdash_{cf} B \oplus C, \Gamma}, \qquad \frac{\vdash_{cf} C, \Gamma}{\vdash_{cf} B \oplus C, \Gamma}.$$

Hence $[\![B]\!] \cup [\![C]\!] \subseteq [\![B \oplus C]\!]$. Therefore, $B^* \cup C^* \subseteq [\![B \oplus C]\!]$. Since $[\![B \oplus C]\!]$ is a fact, $B^* \oplus C^* = (B^* \cup C^*)^{\perp\perp} \subseteq [\![B \oplus C]\!]^{\perp\perp} = [\![B \oplus C]\!]$.

(Case 7) When A is of the form $!B$. Assume $?\Gamma \in (I \cap B^*)$. By the induction hypothesis, $B^* \subseteq [\![B]\!]$. Hence $?\Gamma \in [\![B]\!]$. Since

$$\frac{\vdash_{cf} ?\Gamma, B}{\vdash_{cf} ?\Gamma, !B},$$

$?\Gamma \in [\![!B]\!]$. Therefore, $(I \cap B^*) \subseteq [\![!B]\!]$. Hence $!B^* = (I \cap B^*)^{\perp\perp} \subseteq [\![!B]\!]^{\perp\perp} = [\![!B]\!]$.

(Case 8) When A is of the form $?B$. Assume $\Gamma \in ?B^* = (I \cap B^{*\perp})^{\perp}$. Hence, $\Gamma \cdot (I \cap B^{*\perp}) \subseteq [\![\perp]\!]$. On the other hand, by the induction hypothesis, $B^* \subseteq [\![B]\!]$. Hence, by Lemma 3, $B \in B^{*\perp}$. Since

$$\frac{\vdash_{cf} B, \Delta}{\vdash_{cf} ?B, \Delta},$$

$?B \in B^{*\perp}$. Hence, $?B \in I \cap B^{*\perp}$. Therefore, $\Gamma, ?B \in [\![\perp]\!]$. Hence, $\Gamma \in [\![?B]\!]$.

The cases for constants are proved in the similar way and left to the reader. ■

The Main Lemma may be expressed in the following form.

Lemma 5 (Main Lemma, modified version) *For any formula A, $A^{\perp} \in A^* \subseteq [\![A]\!]$.*

Proof. The second half is the Main Lemma itself. $A^{\perp} \in A^{*}$ is proved by Lemma 3 with the help of the Main Lemma. ■

For the two-sided sequent calculus of classical linear logic, the canonical model can be constructed in the same way as above. Here, the outer value $[\![A]\!]$ is defined as $[\![A]\!] = \{\Gamma^{\perp}, \Delta : \Gamma \vdash \Delta, A \text{ is provable without cut-rule}\}$. Then , the modified form of the Main Lemma (Lemma 5 above) can be proved by induction on the complexity of formula A, in the way similar to the proof of the Main Lemma. Then, the completeness theorem and the cut-elimination theorem are derived from the Main Lemma in the same way as above.

4 Linear Logic as Concurrent Process Calculus by the Proof Search Paradigm

4.1 The Proofs-as-Processes Correspondences in the Proof-Search Paradigm

As explained in Introduction (§1) and §2, the linear logic inference rules capture the time-sensitivity[4] (e.g. parallelism) and resource-sensitivity. In this Section we shall show such an example from theory of concurrent processes, using some message-passing (or token-passing) models, by providing a correspondence between a certain class of linear logic proofs and a certain class of concurrent processes.

In this Section, we consider the following correspondence between the logical notions and the notions from the concurrency theory. We identify each logical connective symbols with an action name, and each logical inference rule (on a logical connective) with a state-transition (by the corresponding action). Under these identifications, each logical formula corresponds to a specification of a concurrent process.

For simplicity, we consider only the Left-Only one-sided sequent calculus of linear logic in this Section.

- $A \otimes B$ (Parallel-action) Process A and process B are started in parallel.
- $\alpha \otimes B$ (Sending-action) Token (or message) α is sent off, and process B is started (at the same time).
- $\alpha \multimap B$ (Receiving-action) Token α is received, and process B is started. More generally, $\alpha_1 \otimes \cdots \otimes \alpha_n \multimap B$ means; tokens $\alpha_1, \ldots, \alpha_n$ are received, and process B is started.
- $!A$ (Bang-action) A copy of A is produced as many as needed, and a copy A is started.

[4] More precisely, the original form of linear logic can capture concurrency and in order to capture time fully some temporal notion should be introduced in linear logic (see [17] for such an example).

- $A \oplus B$ (Choice-action) A or B is chosen, and start the chosen one.
- 0 (Abort-action) The whole process is forced to be terminated.

The process types (or, types) are the formulas constructed from a given set $\{\alpha_1, \alpha_2, \ldots\}$ of tokens (which corresponds to the set of atomic formulas) by the above logical connectives and constant 0. Here, $A \multimap B$ is a process type only if A is a conjunction of tokens of the form $\alpha_1 \otimes \cdots \otimes \alpha_n$ (including the case $n = 1$) and B is a process type.

A process schedule of type Γ_0 is a finite or infinite sequence $S \equiv \Gamma_0, \Gamma_1, \Gamma_2, \ldots$ where Γ_i is a finite multiset of process types (i.e., formulas) and Γ_{i+1} is the result of a transition from Γ_i. Here, the set of transition are expressed in terms of the logical inference rules of the Left-Only one-sided sequent calculus whose formulas are restricted to the process types. In this Section we consider Γ as a multiset of formulas for any (Left-Only) sequent $\Gamma \vdash$, namely, we identify $\Gamma \vdash$ and $\Gamma' \vdash$ for any $\Gamma' \vdash$ obtained from $\Gamma \vdash$ by finitely many uses of the exchange rule. Hence we do not assume the exchange rule in the syntax.

Each action corresponds to a logical inference, by reading each logical inference rule bottom-up (i.e., upwards).

- (Parallel-action)

$$\frac{\Gamma, P, Q, \Delta \vdash}{\Gamma, P \otimes Q, \Delta \vdash}$$

(Parallel-action $P \otimes Q$ invokes two processes P and Q in parallel. Here, Γ and Δ represent finite multisets of processes in the environment.)

A special form of this inference rule represents the Sending-action.

- (Sending-action)

$$\frac{\Gamma, \alpha, P, \Delta \vdash}{\Gamma, \alpha \otimes P, \Delta \vdash}$$

(Sending-action $\alpha \otimes P$ sends a token α and invokes the subprocess P.)

The following rule is a special case of the inference rule for $\multimap$-left.

- (Receiving-action)

$$\frac{\Gamma, Q, \Delta \vdash}{\Gamma, \alpha, \alpha \multimap Q, \Delta \vdash}$$

(Receiving-action $\alpha \multimap Q$ receives a token α (when α exists in the environment) and invokes Q.)

The following is a slightly generalized version of the receiving action.

$$\frac{\Gamma, Q, \Delta \vdash}{\Gamma, \alpha_1, \ldots, \alpha_n, (\alpha_1 \otimes \cdots \otimes \alpha_n) \multimap Q, \Delta \vdash}$$

(Receiving-action $(\alpha_1 \otimes \cdots \otimes \alpha_n) \multimap Q$ receives tokens $\alpha_1, \ldots, \alpha_n$ (when these are in the environment) and invoke Q.)

- (Choice-action)

$$\frac{P, \Delta \vdash}{P \oplus Q, \Delta \vdash} \qquad \frac{Q, \Delta \vdash}{P \oplus Q, \Delta \vdash}$$

(Choice-action $P\&Q$ can choose either P or Q, and invokes it.)

- (Bang-action)

$$\frac{\Gamma, P, !P, \Delta \vdash}{\Gamma, !P, \Delta \vdash}$$

($!P$ produces a copy P and invokes it.)

The following actions correspond to axiom sequents,

- Ligical Axioms (Abort-action)

$$\Gamma, 0, \Delta \vdash$$

(The process terminates as an invalid process when 0 appears.)

We may also consider non-logical axiom sequents, which correspond to the "Overflow-Buffer action". For example, the following expresses that the buffer to hold α's gets an overflow when the buffer receives more than or equal to n-many α's.

- Non-logical axioms (overflow abort-action)

$$\Gamma, \overbrace{\alpha, \ldots, \alpha}^{n-many}, \Delta \vdash$$

(The process terminates as an invalid process when n-many α's appears for a fixed n.)

Although the above inference rules are restricted comparing with the full-linear logic system, it is a conservative subsystem of linear logic with respect to the restricted language (i.e., the restricted form of formulas introduced above) (cf. e.g. Andreoli and R. Pareschi[3], Kobayashi-Yonezawa[18] and Okada[27]). In particular, the above inference rules generate only cut-free proofs (in the sense that we do not use the cut-inference rule).

Under the above interpretation of a proof as a process, it is natural to extend the notion of proof. For example, an infinite proof construction is traditionally considered as an unprovable proof or incomplete proof, but it is natural to include the infinite proof search/construction as a proof in order to treat an infinite (reactive) process.

$$\dfrac{\dfrac{\beta, \gamma^2 \vdash}{\beta^2, \beta \multimap \gamma^2 \vdash}}{\alpha, \alpha \multimap \beta^2, \beta \multimap \gamma^2 \vdash}$$

The above proof shows the following; process $\alpha \multimap \beta^2$ (which receives α and sends two tokens β), process $\beta \multimap \gamma^2$ (which receives β and sends γ^2) and token α are in the initial state (of the input tokens); then, two γ's and one β are produced and the process finishes.

On the other hand, the following infinite proof shows the following; A reactive process which receives α and sends β and another reactive process which sends (possibly infinite) α's as inputs. Then, as the outputs, β's are produced infinitely.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{!\alpha, !(\alpha \multimap \beta), \beta^n \vdash}}{!\alpha, !(\alpha \multimap \beta), \beta \vdash}}{!\alpha, \alpha, \alpha \multimap \beta, !(\alpha \multimap \beta) \vdash}}{!\alpha, \alpha, !(\alpha \multimap \beta) \vdash}}{!\alpha, !(\alpha \multimap \beta) \vdash}$$

The following shows the relationship between various notions on proof and various notions on process representations. We name P, Q for the two receiving actions.

$$\dfrac{\Gamma, \beta^2 \vdash}{\Gamma, \alpha, \alpha \multimap \beta^2 \vdash}\, P \qquad\qquad \dfrac{\Gamma, \gamma^2 \vdash}{\Gamma, \beta, \beta \multimap \gamma^2 \vdash}\, Q$$

Here, with the same end-sequent (i.e., a specification of the process) $\alpha^2, (\alpha \multimap \beta^2)^2, (\beta \multimap \gamma^2)^4$, we could construct the following three different proofs, which corresponds three different (sequential) schedulings of process.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\gamma^8 \vdash}{\qquad\qquad}\, Q}{\qquad\qquad}\, Q}{\qquad\qquad}\, Q}{\qquad\qquad}\, Q}{\qquad\qquad}\, P}{\alpha^2, (\alpha \multimap \beta^2)^2, (\beta \multimap \gamma^2)^4 \vdash}\, P$$

The above proof corresponds to the process schedule $PPQQQQ$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\gamma^8 \vdash}{\qquad\qquad}\, Q}{\qquad\qquad}\, Q}{\qquad\qquad}\, Q}{\qquad\qquad}\, P}{\qquad\qquad}\, Q}{\alpha^2, (\alpha \multimap \beta^2)^2, (\beta \multimap \gamma^2)^4 \vdash}\, P$$

The above proof corresponds to the process schedule $PQPQQQ$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\gamma^8 \vdash}{\qquad} Q}{\qquad} Q}{\qquad} P}{\qquad} Q}{\qquad} Q \qquad \alpha^2, (\alpha \multimap \beta^2)^2, (\beta \multimap \gamma^2)^4 \vdash P$$

The above proof corresponds to the process schedule $PQQPQQ$.

One could also consider a parallel proof figure, which allows parallel application of more than one logical inference rules at once.

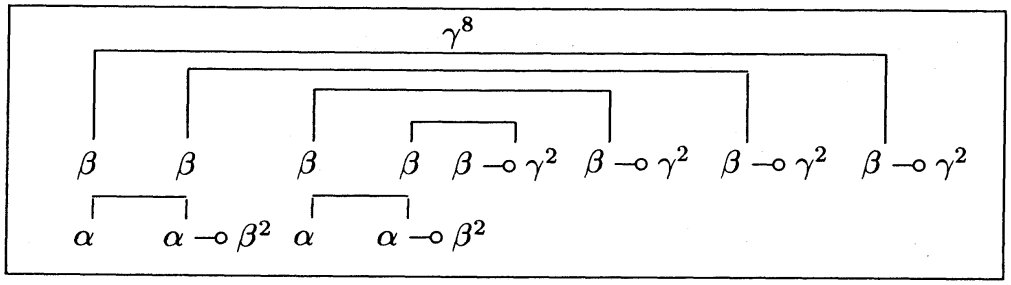
$$\dfrac{\alpha \vdash \alpha \qquad \dfrac{\beta \vdash \beta \qquad \dfrac{\dfrac{\gamma^8 \vdash}{\qquad} Q}{\qquad} Q}{\beta^3, (\beta \multimap \gamma^2)^3, \gamma^2 \vdash} Q}{\alpha, (\alpha \multimap \beta^2), \beta^2, (\beta \multimap \gamma^2)^4 \vdash} P, Q$$

$$\dfrac{\alpha \vdash \alpha \qquad \alpha, (\alpha \multimap \beta^2), \beta^2, (\beta \multimap \gamma^2)^4 \vdash}{\alpha^2, (\alpha \multimap \beta^2)^2, (\beta \multimap \gamma^2)^4 \vdash} P$$

This proof figure represents directly the concurrent process schedule $P(P//Q)QQQ$. (Here, $P//Q$ means that P and Q occur in parallel.)

The first two (sequential) schedules, $PPQQQQ$ and $PQPQQQ$, are viewed as the two possible sequentialized forms of this parallel schedule.

A very fascinating way of representing a process would be the way in which inessential differences due to different representations in the scheduling level of a process are ignored. One way of such a fascinating representation can be obtained by using the notion of proof net. A proof net is a graphic representation of linear logic proof where inessential repetitions of the occurrence of formulas are deleted. A short introduction of proof nets may be found in the Appendix 2 of Girard-Lafont-Taylor[11] and Girard[10]. With a proof net representation, the above all three schedules are identified as the same process. A proof net representation identifies the inessential differences appearing on the levels of the sequential and of the parallel schedulings.



Proof net

We call this process representation which corresponds to a proof net a *trace* of a process. It is actually closely related to the notion of *trace* in the concurrency model theory.

It is sometimes very natural that reaching to an axiom is interpreted as a failure of a process, or a dead-lock of a process, under the bottom-up proof-search (or proof-construction) interpretation.

The notion of *fairness* in the usual proof-search paradigm (or the automated theorem-proving theory) exactly corresponds to the notion of fairness (of a process schedule) in the concurrency theory, under our proofs-as-processes interpretation. A fairness (in a proof-search) means that each formula (which could be treated) in a given environment will be treated eventually.

For example, the following proof-search (proof-construction) strategy fails to complete the proof (without reaching any axiom sequent) when the strategy never treat α.

$$\dfrac{\dfrac{\dfrac{\vdots}{\alpha, \alpha \multimap 0, \ldots, \alpha \multimap 0, !(\alpha \multimap 0) \vdash}}{\alpha, \alpha \multimap 0, \alpha \multimap 0, !(\alpha \multimap 0) \vdash}}{\alpha, \alpha \multimap 0, !(\alpha \multimap 0) \vdash}\ .$$

Such a strategy is an example of unfair proof-search strategy. A fair strategy treats every possible formula to treat for applying inference rules, hence treats α eventually. Then

$$\dfrac{\dfrac{0, \alpha \multimap 0, \ldots, \alpha \multimap 0, !(\alpha \multimap 0) \vdash}{\alpha, \alpha \multimap 0, \alpha \multimap 0, \ldots, \alpha \multimap 0, !(\alpha \multimap 0) \vdash}}{\vdots}\ ,$$

and reaches an axiom sequent, in this case.

Under these correspondences, an incomplete and fair proof corresponds to a deadlock-free and fair process schedule, which is often called a *safe* process schedule.

Definition 3 A process schedule is *fair* if all possible actions are taken eventually if the process schedule is an infinite sequent. Any terminating (i.e., finite) process schedule is defined to be fair. A process schedule of type Γ is called *deadlock-free* if it is not aborted, i.e., if it does not reach any abort action. A process schedule of type Γ is called *safe* if it is fair and deadlock-free. (In particular, all fair infinite processes schedules and all deadlock-free finite process schedules are safe, according to this definition.)

Now assume that all axioms are logical axioms, namely that there is no axioms for overflow abort-action. Then the following is a statement corresponding to the Gödel's completeness theorem (of the traditional logics).

Theorem 5 (Completeness) *A specification Γ is consistent in Linear Logic if and only if there exists a safe process schedule of type Γ.* [5]

Here, "Γ is consistent" means that no contradiction is derived from Γ (i.e.,$\Gamma \vdash \bot$ is not provable.)

Proof. By the cut-elimination theorem for Linear Logic, $\Gamma \vdash \Delta$ is provable iff $\Gamma \vdash$ is provable without cut. Hence, for any process type Γ, $\Gamma \vdash \bot$ is not provable iff $\Gamma \vdash$ does not reach any axiom-sequent by the above-listed transition rules, for any proof search strategy (including the fair ones), which means there exists a safe process schedule of type Γ.

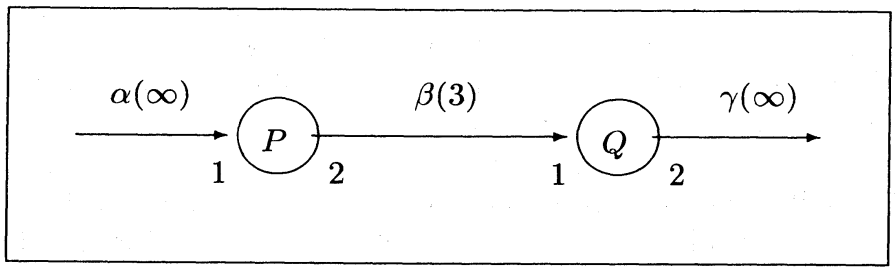
Note that the above transition rule for receiving-action is interpreted as the following combination of $\multimap$ left rule and the logical axiom sequent $\alpha \vdash \alpha$ in Linear Logic;

$$\frac{\alpha \vdash \alpha \quad \Gamma, Q, \Delta \vdash}{\Gamma, \alpha, \alpha \multimap Q, \Delta \vdash}, \qquad \frac{\alpha_1 \vdash \alpha_1 \quad \cdots \quad \alpha_n \vdash \alpha_n \quad \Gamma, Q, \Delta \vdash}{\Gamma, \alpha_1, \ldots, \alpha_n, (\alpha_1 \otimes \cdots \otimes \alpha_n) \multimap Q, \Delta \vdash}.$$

4.2 Some simple examples of concurrent process specification

We consider some examples of specifications with using Kahn's regular dataflow nets with bounded buffer-sizes, which give basic models of concurrent process computation (cf. Kahn[16]).

Example 1.



Dataflow specification

Here channel β has a bounded buffer of size 3, (which means that channel β can store up to three tokens and that when channel β gets more than three tokens, the whole process of this network stops by the overflow of channel β). Process (1)P(2) is a process to receive one token through channel α then to send two tokens through

[5] One could use &, instead of $\oplus$, for a choice action (with the same inference form as that of $\oplus$-choice action). Then the completeness is stated as "a specification Γ is consistent if and only if all process schedules of type Γ are deadlock-free".

channel β. Process (1)Q(2) is a process to receive one through channel β then to send two through channel γ.

Now we assume that two input tokens at (the buffer in) channel α and that other channels are empty at the beginning. The network is specified by formula (by regarding a token as an atom (atomic formula)),

$$(\alpha \multimap (\beta \otimes \beta)) \otimes (\beta \multimap (\gamma \otimes \gamma)) \ .$$

In this Section, we use δ^n to abbreviate $\underbrace{\delta \otimes \cdots \otimes \delta}_{n-many}$. Hence, the above formula may be written as $(\alpha \multimap \beta^2) \otimes (\beta \multimap \gamma^2)$.

The bounded buffer constraint (for the bounded buffer size β^3) is specified by an additional (non-logical) axiom

$$\Gamma, \beta, \beta, \beta, \beta \vdash \ .$$

Now consistency is checked by the following setting with the input.

$$\alpha, \alpha, (\alpha \multimap \beta^2) \otimes (\beta \multimap \gamma^2) \vdash \ .$$

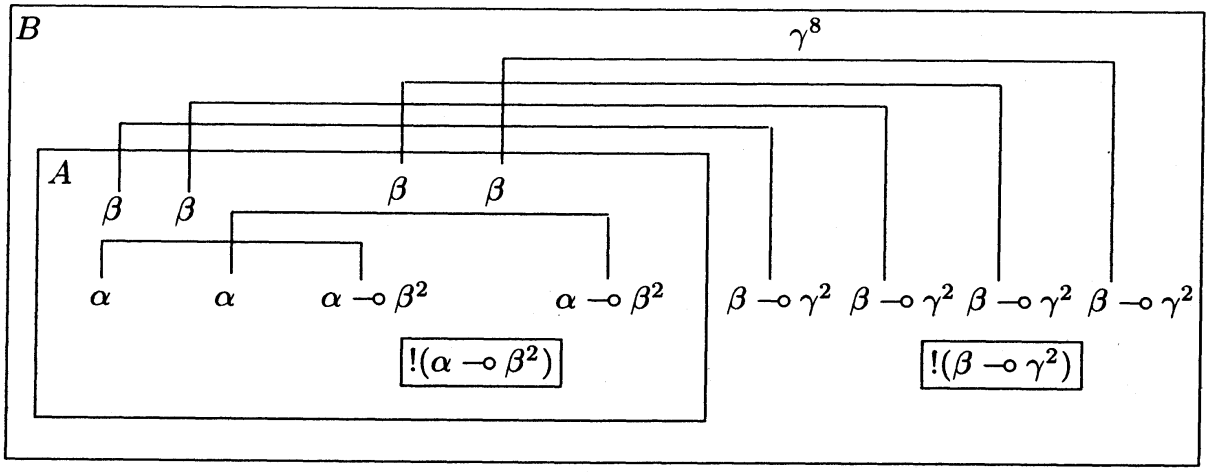
There are several schedules;

1. Two α are passed to P, and P produces four β's. Then the process stops by the buffer over-flow.

2. One α is received by P, and two β's are sent from P, then one β is taken by Q and two γ's are produced. Then, another one α is taken by P and two more β's are sent. Now three β's are in the buffer. Then each β is taken one by Q and six more γ's are produced.

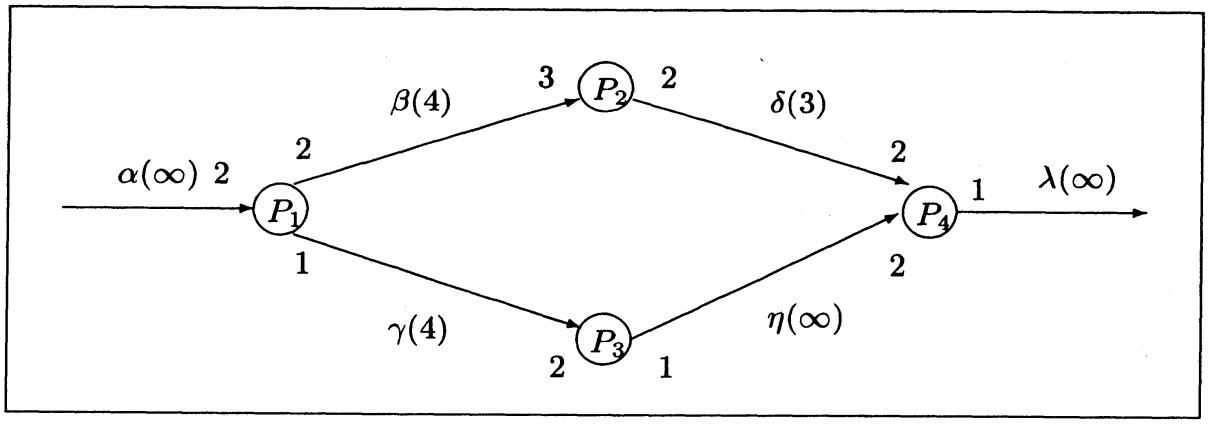
3. The first half is the same as 2. When the second α is taken by P, concurrently the one β left is taken by Q, and two more γ's are produced. Then the rest are treated in the similar way as 2.

Schedule 1 corresponds to a (complete) proof, which means an invalid process, i.e., the process by this schedule is aborted (by the OVER-FLOW-BUFFER action). Proof net A below corresponds to the trace of this process. Schedules 2 and 3 are different schedules but the same trace, which corresponds to the same proof net B, (however, different in the proof figures level). Note that, if one set up the size of the bounded buffer of channel β as 3, instead of 4, schedule 2 becomes an aborted process, which corresponds to a new complete proof net, which is different from the two. That means that in this case not only the schedules of process but also the traces of process become different.



Proof net

Example 2. Consider the dataflow diagram below;



Dataflow specification

We assume that channels β, γ, δ have bounds of buffer size, 4, 4, 3, respectively, while channels α, η, λ have no bound. Process P_1 receives two messages (tokens) from the channel α and produces two tokens to channel β and one token to channel γ concurrently. This is specified as $!(\alpha \otimes \alpha) \multimap (\beta \otimes \beta \otimes \gamma)$, which may also be written as $!(\alpha^2 \multimap \beta \otimes \gamma)$. The modality ! means this process can be repeated infinitely many times. P_2 is waiting for 3 tokens through channel β then is sending two tokens through channel δ, namely $!(\beta^3 \multimap \delta^2)$. P_3 is waiting for two tokens through channel γ then sending one token through channel η. P_3 is specified by $!(\gamma^2 \multimap \eta)$. When P_4 receives two tokens from channel δ and two from η concurrently, it produces one output token through channel λ. P_4 is written as $!(\delta^2 \otimes \eta^2 \multimap \lambda)$. Then the whole network is described as Γ, which is

$$!(\alpha^2 \multimap \beta^2 \otimes \gamma), !(\beta^3 \multimap \delta^2), !(\gamma^2 \multimap \eta), !(\delta^2 \otimes \eta^2 \multimap \lambda).$$

We consider a stream input (written as $!\alpha$) at channel α, which generates infinitely many tokens one by one through channel α. Now consider an input channel state

m, say $!\alpha$, β^2, η^3. This means that the network is started with channel state m, i.e., two tokens at channel β, three tokens at channel η and an (infinite) token generator (infinite stream input) at channel α. Then by the completeness theorem, the fact that m, Γ is consistent with the bounded buffer axioms for β^4, for γ^4 and for δ^3 in the system S is equivalent to the fact that there is a safe process of m, Γ. (Here, the commas between formulas mean concurrency.)

Example 3. If one wants to share the two buffers δ and γ, one can simply change $\delta(3)$ to $\gamma(7)$ and $\gamma(4)$ to $\gamma(7)$ on the dataflow graph above. Then the corresponding specification formula is obtained by replacing δ in Γ by γ. This type of shared channels makes a specification of Petri net possible. A place name of a Petri net corresponds to a channel name. (We shall give the definition and an example of Petri net in the next Section.)

Example 4. Next, when we specify process P_1 and process P_4 as an arbitrary choice and a merge, the specification Γ is changed to

$$!(\alpha^2 \multimap \beta^2 \& \gamma), !(\beta^3 \multimap \delta^2), !(\gamma^2 \multimap \eta), !(\delta^2 \oplus \eta^2 \multimap \lambda) \ .$$

Here, the choice process P_1 receives two tokens from channel α then send either two tokens to channel β or one token to channel γ. The merge process P_4 sends one token to channel λ when it receives either two tokens from channel δ or one token from channel η.

Example 5. An exclusive pair channels, namely a pair (α, β) of channels which pass only one token either through channel α or through channel β can be expressed expressed by the three axioms;

$$\Gamma, \alpha, \beta \vdash \quad \Gamma, \alpha, \alpha \vdash \quad \Gamma, \beta, \beta \vdash \ .$$

Using these constraint axioms, one can define a conditional choice process. For example, the specification of a deterministic choice process, which sends γ^3 if α is received, and which sends δ^2 if β is received, is written as; $(\alpha \multimap \otimes(\gamma^3)) \oplus (\beta \multimap \otimes(\delta^2))$.

5 Naive Phase Semantics and Completeness of the Petri Net Specification

We consider the following subsystem S_0 [6] of Intuitionistic Linear Logic. A formula of the form $a_1 \otimes \cdots \otimes a_n$ for atoms (atomic formulas) $a_1, \ldots, a_n$ is called a marking in this section. The formulas of S_0 is restricted in the way that for $A \multimap B$, A is a marking. The formulas of S_0 is defined as;

[6] By virtue of the cut-elimination theorem of the Intuitionistic Linear Logic S_0 is shown to be a conservative subsystem of Intuitionistic Linear Logic. Namely, for any sequent $\Gamma \vdash A$ of the language of S_0, the following are equivalent; (1) $\Gamma \vdash A$ is provable in S_0, (2) $\Gamma \vdash A$ is provable in Intuitionistic Linear Logic.

(1) if a is an atom, then a is a formula,

(2) if A and B are formulas, then so is $A \otimes B$, $A \,\&\, B$ and $A \oplus B$,

(3) if m is a marking and A is a formula then $m \multimap A$ is a formula,

We do not assume the linear negation as a basic logical connectives (since it is definable as $A^{\perp} \equiv A \multimap \perp$). We do not use modality !, either.

The inference rules of S_0 are those of Intuitionistic Linear Logic restricted to the above subclass of formulas, in particular, $\multimap$ rules are as follows, due to the restriction on $\multimap$ of (3) above;

($\multimap$-left) ($\multimap$-right)

$$\frac{\Gamma \vdash m \quad A, \Delta \vdash B}{m \multimap A, \Gamma, \Delta \vdash B}\ , \qquad \frac{m, \Gamma \vdash A}{\Gamma \vdash m \multimap A}\ .$$

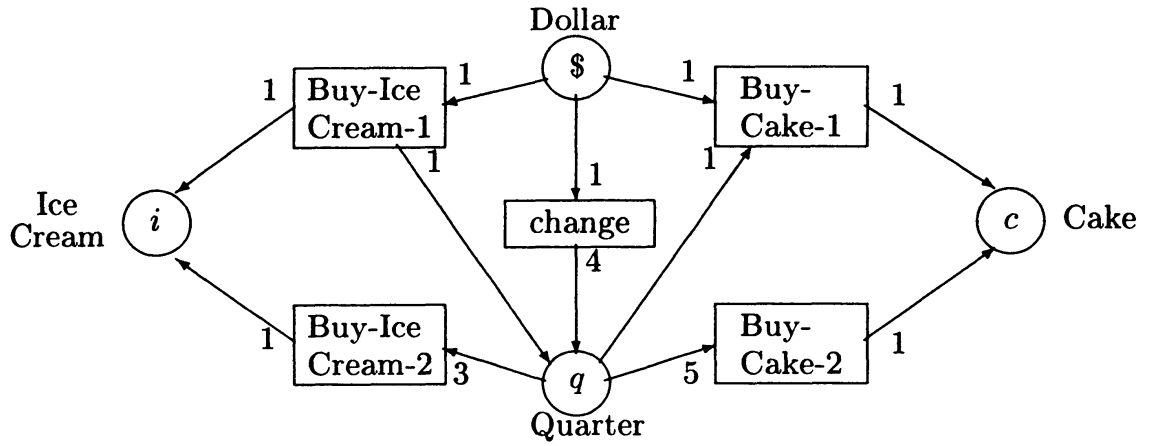
where m is a marking.

The purpose of this Section is to provide a natural and simple semantics for the provability of S_0. We also extend S_0 by adding non-logical axiom sequents essentially of the form $m \vdash n$ (for markings m and n), by which a Petri net specification can be naturally formulated, and we apply our simple semantics to this extended S_0. As well-known, a naive phase space semantics such as interpreting $\alpha \oplus \beta$ as $\alpha \cup \beta$ is sound with respect to intuitionistic linear logic but that it is not complete; for example, the distributive rule between & and $\oplus$ is true in this model, but cannot be provable by the linear logic. (The soundness of a naive phase semantics has been used in the literature of applications of linear logic to Petri net specifications (cf. Engberg and Winskel[8], Marti-Oliet and Messeguer[21]).) Hence, usually in order to obtain the completeness of linear logic, one needs a certain stronger closure conditions on the operator definitions of the phase semantics. (For example, the classical $\oplus$ is defined by $\alpha \oplus \beta = (\alpha \cup \beta)^{\perp\perp}$. The intuitionistic $\oplus$ is, for example, defined by $\alpha \oplus \beta =$ the smallest fast which contains $\alpha \cup \beta$, with some suitable closure conditions for the facts. Cf. Abrusci[2], Troelstra[32], Okada[26].) However, from the point of view of practical application of system S_0, including Petri net specifications, it is hard to understand any intuitive meaning with these closure conditions. A natural question is whether or not the naive and more intuitive phase semantics is complete with respect to this fragment of linear logic. We shall show in this Section that the more naive definitions of linear operators provide a meaningful Completeness theorem for system S_0 and its extended systems.

A Petri net is (P, N), where P is a set of places and N is a set of transitions. A multiset m over P is called a marking. M_P stands for the set of markings of P. A transition $t \in N$ is of the form $(m_1 : m_2)$ for markings m_1 and m_2. For markings m, m', and a transition $t = (m_1 : m_2) \in N$, $m[t > m'$ (m' is one step reachable from m) means that t fires from m to m', that is; $m[t > m'$ iff $\exists m'' \in M$

($m = m''m_1$ and $m' = m''m_2$). A reachability relation $m \to m'$ is the transitive and reflexive closure of the one step reachability. The downwards closure $\downarrow m$ of m means $\{m' \in M : m' \to m\}$.

The following is an example of a Petri net:



Vending Machine[7]

Here Petri net (P, N) is defined as; the set of places (i.e., tokens) $P = \{q, \$, i, c\}$, the set of transitions $N =$ { Buy-Ice Cream-1, Buy-Ice Cream-2, Buy-Cake-1, Buy-Cake-2, change } $= \{(\$: i, q), (q, q, q : i), (q, \$: c), (q, q, q, q, q : c), (\$: q, q, q, q)\}$, represented by the following linear logical formulas;

Buy-Ice Cream-1 :	$\$ \multimap i \otimes q$
Buy-Ice Cream-2 :	$q \otimes q \otimes q \multimap i$
Buy-Cake-1 :	$q \otimes \$ \multimap c$
Buy-Cake-2 :	$q \otimes q \otimes q \otimes q \otimes q \multimap c$
change :	$\$ \multimap q \otimes q \otimes q \otimes q$

We also often regard the transition set N itself as a Petri net.

As seen in the above, a transition of a Petri net can be represented by a linear logic formula of the form $m_1 \otimes \cdots \otimes m_k \multimap n_1 \otimes \cdots \otimes n_k$. We identify the set N of transition of a Petri net (P, N) and the corresponding set of the linear logic formulas, and denote the linear logical formulas as N. We extend system S_0 by adding non-logical axiom sequents $a_1, \ldots, a_n \vdash b_1 \otimes \ldots \otimes b_m$ for all $a_1 \otimes \cdots \otimes a_n \multimap b_1 \otimes \cdots \otimes b_m \in N$. We denote this system as $S_0(N)$. When all n of $m \multimap n \in N$ is a single atom, Petri net N is called a *strict* Petri net. Hence, for a strict Petri net N, the non-logical axiom sequent of $S(N)$ is of the form $a_1, \ldots, a_n \vdash b$ for atoms $a_1, \ldots, a_n$ and b.

For a cut inference J occurring in an $S_0(N)$-proof, if the left cut formula or the right cut formula of a cut inference J comes from an occurrence of a non-logical axiom sequence $a_1, \ldots, a_n \vdash b_1 \otimes \cdots \otimes b_m$, then the occurrence of cut inference J

[7] This Vending Machine example is taken from Martí-Oliet and Meseguer[21] with a slight modification.

is called a *non-free cut.* Otherwise, an occurrence of cut inference is called a *free cut.* There are two cases of non-free cut, as shown below.

- A non-free cut J (Case 1)

an N-axiom sequent
$a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n \vdash b_1 \otimes \cdots \otimes b_m$

$$\frac{\Delta \vdash a_i \quad a_i, \Gamma \vdash B}{\Delta, \Gamma \vdash B}\ J$$

- A non-free cut J (Case 2)

an N-axiom sequent
$a_1, \ldots, a_n \vdash b_1 \otimes \cdots \otimes b_m$

$$\frac{\Delta \vdash b_1 \otimes \cdots \otimes b_m \quad b_1 \otimes \cdots \otimes b_m, \Gamma \vdash B}{\Delta, \Gamma \vdash B}\ J$$

Then the following cut-elimination theorem holds. See the end of this subsection for a proof.

Fact 1 (Cut-Elimination Theorem)

1. *If* $\Gamma \vdash A$ *is provable in* S_0, *then* $\Gamma \vdash A$ *is provable without cut-rule in* S_0.
2. *If* $\Gamma \vdash A$ *is provable in* $S_0(N)$, *then* $\Gamma \vdash A$ *is provable without free-cut in* $S_0(N)$.

$(M_P, \otimes, \&, \oplus, \multimap)$ is a *naive phase space* on P where M is the commutative free monoid generated by a set P. Every subset of M_P is called a fact in a naive phase model on M. In this Section, an element of P is called a *token* (*place name*) and an element of M_P, i.e., a multiset over P, a *marking.*

On a naive phase space $(M_P, \otimes, \&, \oplus, \multimap)$, the interpretation of the linear operators & (with), $\oplus$ (and), $\otimes$ (tensor) and $\multimap$ (linear implication) are defined as follows; For any facts α and β,

L1. $\alpha \& \beta = \alpha \cap \beta$
L2. $\alpha \oplus \beta = \alpha \cup \beta$

L3. $\alpha \otimes \beta = \alpha\beta$
L4. $\alpha \multimap \beta = \{n : \forall m \in \alpha\ mn \in \beta\}$.

The interpretation A^* of a formula A is defined to be a fact in the following way, when an assignment of facts for atoms (atomic formulas) occurring in A is given. We call this value A^* the inner-value of A.

$$\begin{array}{lcl}
(A \multimap B) & = & A^* \multimap B^* \\
(A^{\perp})^* & = & (A^*)^{\perp} \\
(A\&B)^* & = & A^*\&B^* \\
(A \oplus B)^* & = & A^* \oplus B^* \\
(A \otimes B)^* & = & A^* \otimes B^* \ . \\
1^* & = & \mathbf{1} = \text{the smallest fact which contains } 1 \in M, \\
 & & \text{(which is } \{1\} \text{ in the case of a naive phase model.)}
\end{array}$$

Now we show the soundness and completeness of our fragment S_0 of linear logic with respect to the naive phase models.

First, we note that, as well-known, soundness holds for the naive semantics, not only for our fragment, but also for full-system of intuitionistic linear logic, as well known.

Theorem 6 (Soundness Theorem) *If $A_1, \ldots, A_n \vdash B$ is provable in S_0, then for any naive phase model, $A_1^* \cdot \cdots \cdot A_n^* \subseteq B^*$.*

Proof. The proof is carried by induction on the length of a given S_0-proof. For the case where the left hand-side of the sequent is empty (of the form $\vdash B$), we interpret this as $1 \vdash B$. ■

Now we apply this to Petri net specifications, as an example. Recall that a Petri net transition N is specified by a set of transition formulas of the form $m_i \multimap n_i$, where m_i and n_i are markings, i.e., $\otimes$-conjunctions of tokens. Here we identify a place name (of Petri net) with a token.

For a Petri net $N = \{m_i \multimap n_i\}_{i=1,\ldots,k}$, a free commutative monoid M_P, a set $\alpha \subseteq M_P$ satisfies the N-downward closure property if $m_j l \in \alpha$ then $n_i l \in \alpha$ for any $i = 1, \ldots, k$; in other words, $\downarrow m \subseteq \alpha$ for any $m \in \alpha$, in the Petri net terminology.

For a strict Petri net N, a strict N-phase space $(M_P, N, \otimes, \&, \oplus, \multimap)$ is defined in the same way as a naive phase space $(M_P, \otimes, \&, \oplus, \multimap)$, where a fact of a strict N-phase space is such a subset $\alpha \subseteq M_P$ which satisfies the N-downward closure property.

For a Petri net N, an N-phase space $(M_P, N, \otimes, \&, \oplus, \multimap)$ is defined in the same way as a strict N-phase space, where a fact of an N-phase space is a subset $\alpha \subseteq P$ which satisfies the N-downward closure property. For $\alpha \subseteq M_P$, the smallest fact which includes α is denoted by $Cl(\alpha)$. Then, the definition L3 of the operator $\otimes$ on a N-phase space needs to be modified as; for any facts α and β,

L3'. $\alpha \otimes \beta = Cl(\alpha\beta)$.

Note that the N-downward closure property is preserved for all operators $\alpha \& \beta = \alpha \cap \beta$, $\alpha \oplus \beta = \alpha \cup \beta$, $\alpha \otimes \beta = \alpha\beta$ and $\alpha \multimap \beta$ for a strict N-phase space, and is preserved for all operators with the above modification $\alpha \otimes \beta = Cl(\alpha\beta)$ for an N-phase space.

Since the N-downward closure property for the facts implies the soundness for the non-logical axiom sequents (i.e., $m^* \subseteq n^*$ for any $m \multimap n \in N$), we can easily see that the Soundness Theorem for S_0 is generalized to that for $S_0(N)$.

Theorem 7 (Soundness Theorem) *If $A_1, \ldots, A_n \vdash B$ is provable in $S_0(N)$ for a Petri net N (for a strict Petri net N, respectively), $A_1^* \cdot \cdots \cdot A_n^* \subseteq B^*$ in any N-phase model (in any strict N-phase model, respectively).*

A marking (i.e., a multiset of tokens) m is called a *precondition* of an specification formula A if A is satisfied for input m. The interpretation of each linear logic operator has a direct meaning for a specification formulas of processes given in Section 2. When a^* for atom (token) a is interpreted as a set of input tokens to fire a, A^* is a set of input tokens to realize specification A. The above soundness theorem means that a precondition of $A_1, \ldots, A_n$, namely a set of input tokens to realize $A_1, \ldots, A_n$, is also a precondition of B, namely a set of input tokens to realize A. (Then the notion of precondition above is understood as the precondition of the reachability in the sense of Petri nets.) Then the above soundness theorem implies that if $A \vdash B$ is provable in $S_0(N)$ then for any Petri net including the transition relation N any precondition of A is a precondition of B, in particular, if A is a marking, the condition A itself is a precondition of B.

For completeness proof we construct a canonical model on the domain of the set of marks (under the identification between a sequence of atoms and their $\otimes$-conjunctive form). Now we introduce the outer value $[\![A]\!]$ of a formula A as follows.

In the rest of this Section we identify a marking $a_1 \otimes \cdots \otimes a_n$ and a multiset $\{a_1, \ldots, a_n\}$.

Definition 4 *The* outer-value $[\![A]\!]$ *of A is of the form $\{m : m \vdash A\}$, where m is a mark, i.e., m is a sequence (i.e. a multiset) of atoms.*

The inner value A^* of a formula A is given in the way defined before by setting that for each atom (token) a, the inner value is defined to be $[\![a]\!]$.

Lemma 6 (Main Lemma) *For any formula A, $A^* = [\![A]\!]$.*

Then, the completeness theorem follows immediately from this.

Theorem 8 (Completeness (Okada)) *If formula A is true for any naive phase space, A is provable.*

If A is valid, then A is true in our canonical model, hence $\emptyset \in A^*$, therefore $\emptyset \in [\![A]\!]$ by the Main Lemma, which means $\vdash A$, i.e., A is provable.

Now we consider an application of our Completeness proof to a Petri net system $S_0(N)$. When a Petri net is specified by a set of transitions $N = \{m_i \multimap n_i\}_i$, we consider the following modification of the outer value;

Definition 5 *The* outer-value $[\![A]\!]$ *of* A *for Petri net* N *is of the form* $\{m : m \vdash A$ *is provable in* $S_0(N)\}$, *where* m *is a marking, i.e.,* m *is a sequence (i.e. a multiset) of atoms.*

Then the previous completeness proof works as it is. In particular, the canonical model satisfies the downwards closure condition. Hence, the following Completeness theorem holds;

Theorem 9 (Completeness for Petri net N (Okada)) *If* $A \multimap B$ *is valid for any* N*-phase model (for any strict* N*-phase model, respectively), then* $A \vdash B$ *is provable in* $S_0(N)$ *for a Petri net* N *(for a strict Petri net* N*, respectively).*

Proof of Main Lemma. The proof of this is carried by induction on the complexity of a formula A. We recall that we use marking expression m, n for expressing both a finite set of atoms and the $\otimes$-conjunction of that set, without distinguishing the two explicitly. For example, m may be $a_1, \ldots, a_n$ as well as $a_1 \otimes \cdots \otimes a_n$ depending on the context.

(Case 1) When A is atomic. $A^* = [\![A]\!]$ by definition.

(Case 2) A is the form $m \to C$, where m is a mark; we prove $m^* \multimap C^* = [\![B \multimap C]\!]$.

(2.1) $m^* \multimap C^* \subseteq [\![m \multimap C]\!]$.

Assume that $n \in m^* \multimap C^*$.
By the definition of $m^* \multimap C^*$, for any $l \in m^*$, $nl \in C^*$.
On the other hand, it is easily seen that $m \in m^*$, and by the induction hypothesis, $C^* \subseteq [\![C]\!]$. Hence, $n, m \in [\![C]\!]$. On the other hand,

$$\frac{n, m \vdash C}{n \vdash m \multimap C},$$

by $\otimes$-left rules and $\multimap$-right rule.

Therefore, $n \in [\![m \multimap C]\!]$.

(2.2) $[\![m \multimap C]\!] \subseteq m^* \multimap C^*$

Assume that $n \in [\![m \multimap C]\!]$. Hence $n, m \vdash C$. It suffices to show that for any $l \in m^*$, $l, n \in C^*$.
If $l \in m^*$, since $m^* = [\![m]\!]$ by the induction hypothesis, $l \vdash m$. Hence $n, l \vdash C$, i.e., $l, n \in [\![C]\!]$. On the other hand, $[\![C]\!] = C^*$ by the induction hypothesis.

(Case 3) $A^* \& B^* = [\![A \& B]\!]$.

(3.1) $A^* \& B^* \subseteq [\![A \& B]\!]$.
By the induction hypothesis, $A^* = [\![A]\!]$ and $B^* = [\![B]\!]$. Hence $A^* \& B^* = A^* \cap B^* = [\![A]\!] \cap [\![B]\!]$.
On the other hand,

$$\frac{m \vdash A \quad m \vdash B}{m \vdash A \& B}.$$

Hence, $[\![A]\!] \cap [\![B]\!] \subseteq [\![A \& B]\!]$.

(3.2) $[\![A \& B]\!] \subseteq A^* \& B^*$
Assume that $m \in [\![A \& B]\!]$. Then by analysis of cut-free proof of $m \vdash A \& B$, $m \vdash A$ and $m \vdash B$, since m is a mark. Then, by the induction hypothesis, $[\![A]\!] = A^*$ and $[\![B]\!] = B^*$. Hence, $m \in A^* \cap B^*$.

(Case 4) $A^* \oplus B^* = [\![A \oplus B]\!]$

(4.1) $A^* \oplus B^* \subseteq [\![A \oplus B]\!]$

Assume that $m \in A^* \oplus B^*$. By definition and the induction hypothesis, $m \in A^* \cup B^* = [\![A]\!] \cup [\![B]\!]$, which is included in $[\![A \oplus B]\!]$.
The last set inclusion is true because

$$\frac{m \vdash A}{m \vdash A \oplus B} \qquad \frac{m \vdash B}{m \vdash A \oplus B}.$$

(Case 5) $A^* \otimes B^* = [\![A \otimes B]\!]$

The set inclusion of the left hand side into the right hand side is obtained in the same way as above, namely by using the induction hypothesis and the right introduction rule of $\otimes$.

The reverse direction is obtained by analyzing the possible free cut-free proofs of $A \otimes B$, by using the fact that there is no additional axioms.

With the presence of non-logical axiom sequents N for a strict Petri net N, the above argument also holds, where we analyze the possible free-cut free proofs of $A \otimes B$ for the reverse case.

With the presence of non-logical axiom sequents N for a Petri net N, we modify the proof of (Case 5) as follows.

First we show $A^* \otimes B^* \subseteq [\![A \otimes B]\!]$. By the induction hypothesis $A^* \otimes B^* = Cl(A^* \cdot B^*) = Cl([\![A]\!] \cdot [\![B]\!])$. On the other hand, $[\![A]\!] \cdot [\![B]\!] \subseteq [\![A \otimes B]\!]$ and $[\![A \otimes B]\!]$ is N-downward closed. Hence, $Cl([\![A]\!] \cdot [\![B]\!]) \subseteq [\![A \otimes B]\!]$.

For $[\![A \otimes B]\!] \subseteq A^* \otimes B^*$, assume $n \in [\![A \otimes B]\!]$, then we show $n \in A^* \otimes B^*$. $n \vdash A \otimes B$ is provable. By analyzing the free-cut free proofs of $n \vdash A \otimes B$, we can see that if $A \otimes B$ is not a marking, there are markings n_1 and n_2 such that $n = n_1 n_2$ and $n_1 \vdash A$ and $n_2 \vdash B$. Then, $n = n_1 n_2 \in [\![A]\!][\![B]\!] = A^* \cdot B^*$ by the induction hypothesis, and the claim holds. If $A \otimes B$ is a marking, then $[\![A \otimes B]\!]$ is the smallest fact which includes $[\![A]\!][\![B]\!]$ $(= A^* \cdot B^*)$. Hence $[\![A \otimes B]\!] = A^* \otimes B^*$. ■

In the above proof, we analyzed the cut-elimination proofs (of S_0 and of $S_0(N)$), in order to prove the $[\![A]\!] \subseteq A^*$ direction. So we assumed the Fact 1 (the cut-elimination theorem). However, the Fact 1 (the cut-elimination theorem) can be proved in the same way as in Section 3. Here, we can follow all the above arguments (for $A^* \subseteq [\![A]\!]$) in the above proof of the Main Lemma for the $A^* \subseteq [\![A]\!]$ direction, for which we did not use the cut-elimination theorem. The rest of the argument to prove the Fact 1 from $A^* \subseteq [\![A]\!]$ with the help of the soundness theorem is exactly the same as that in Section 3.

For Further Study in Linear Logic

The first paper [9] of linear logic (which was introduced by J. Y. Girard in 1987) is self-contained and is now still useful and stimulating for the reader. The characterization theorem for the proof nets in [9] was simplified by Danos-Regnier[6]. Lafont[19] is a good reference to proof nets and their generalized form, interaction nets. A short introduction of proof nets may be also found in Appendix 2 of [11] as well as Girard[10]. Girard[10], Troelstra[32] and Scedrov[29] are good introductory surveys of linear logic. In particular, Girard[10] (as well as Girard[9]) contains a good introduction to coherent semantics, which is a basic denotational semantics for *proofs* of linear logic, while phase semantics is a basic Tarski-style semantics for *provability* of linear logic. A short but very informative explanation on the underlying idea of coherent semantics (in the traditional type-theoretical context) may be found in [11]. Scedrov[29] contains a good introduction on the subject of decidability and computational complexity problems for various subsystems of linear logic. (Takeuti[31] is also a good introductory textbook on this subject treating one of Kanovich's results, but is written only in Japanese.) Troelstra[32] contains, among others, an introduction to category-theoretical models of linear logic. Blass[5] and Abramsky-Jagadeesan[1] are good introductory references to game semantics. [23] is a collection of papers on the logic-programming paradigm (or, proof-search paradigm) of linear logic (although most of the papers are refined and published elsewhere later). [12] and [13] are collections of papers in various fields of linear logic, in which the reader can find further references to the recent work of linear logic, and to their connections to type theory, proof theory, and mathematics.

APPENDIX 1

Here we recall the (traditional) classical logic in the sequent calculus formulation for the comparison with linear logic. see Takeuti[30] for the detail.

We introduce the language of traditional logic as follows;

(1) logical connectives; $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implies),
(2) logical constants; $\top$ (true), $\bot$ (false),
(3) propositional variables; $P, Q, R \ldots, P_0, P_1, P_2, \ldots$

The formulas are constructed from logical constants and/or propositional variables by logical connectives, in the same way as in Section 2. Note that the traditional classical logic does not have the modal connectives ! nor ?.

We use meta-symbols $A, B, \ldots$ to express formulas. Finite multisets of formulas (possibly the empty multiset) are denoted by $\Gamma, \Delta, \ldots$. A Sequent is an expression of the form $\Gamma \vdash \Delta$, as in Section 2. The axiom sequent and the $\neg$-left, $\neg$-right, $\rightarrow$-left, $\rightarrow$-right, exchange-left, exchange-right, and cut-rules are the same as those of the classical linear logic rules in Section 2, where $A^{\perp}$ is now replaced by $\neg A$ and $A \multimap B$ is replaced by $A \rightarrow B$. Instead of the modality rules (the rules for ! and ?), we impose the following structural rules to all (non-modal) formulas.

Weakening - left

$$\frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}$$

Weakening - right

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}$$

Contraction - left

$$\frac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}$$

Contraction - right

$$\frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}$$

For the rules of $\wedge$ and of $\vee$, we can take either the additive rules (i.e. of & and of $\oplus$, respectively), or the multiplicative rules (i.e. of $\otimes$ and of $\parr$, respectively). With the presence of the above additional structural rules, it is easily verified that the additive rules and the multiplicative rules are equivalent (namely, one is a derived rule of the other). For example, the additive representation of the rules for $\wedge$ and $\vee$ are as follows.

$\wedge$ - left

$$\frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \qquad \frac{B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}$$

$\wedge$ - right

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}$$

$\vee$ - left

$$\frac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta}$$

$\vee$ - right

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, A \vee B} \qquad \frac{\Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \vee B}$$

APPENDIX 2 Syntax of Right-Only One-Sided Classical Linear Logic

Axiom and Inference Rules of Right-Only Classical Linear Logic are as follows.

- **Axioms**

  Logical axioms sequent

  $$\vdash A, A^{\perp}$$

  Logical constants

  $$\vdash 1 \qquad \vdash \Gamma, \top \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}$$

- **Structural Rules**

  Exchange-rule

  $$\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta}$$

  Cut-rule

  $$\frac{\vdash \Gamma, A \quad \vdash \Delta, A^{\perp}}{\vdash \Gamma, \Delta}$$

- **Multiplicative**

  $\otimes$-rule

  $$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}$$

  $\parr$-rule

  $$\frac{\vdash \Gamma, A \parr B}{\vdash \Gamma, A, B}$$

- **Additive**

  &-rule

  $$\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B}$$

  $\oplus$-rule

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

- **Modality**

  !-rule

  $$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}$$

  ?-rule

  $$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}$$

Acknowledgement
The author would like to express his sincere thanks to Mr. Koji Hasebe for his editorial assistance of this paper.

References

[1] S. Abramsky and R. Jagadeesan, Games and Full Completeness for Multiplicative Linear Logic, *Journal of Symbolic Logic 59*, pages 543-574, 1994.

[2] V. M. Abrusci, Sequent Calculus for Intuitionistic Linear Propositional Logic, in Mathematical Logic, Plenum Press 1990.

[3] J. M. Andreoli and R. Pareschi, Linear Objects: Logical processes with built-in inheritance, *New Generation Computing*, 1991.

[4] J.-M. Andreoli Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297-347, 1992.

[5] A. Blass, A Game Semantics for Linear Logic. *Annals of Pure and Applied Logic 56*, pages 183-220, 1992.

[6] V. Danos and L. Regnier, The Structure of Multiplicatives, *Arch. Math Logic 28*, pages 181-203, 1989.

[7] J.B. Dennis, Data Flow Computation, in *Control Flow and Data Flow: Concepts of Distributed Programming*, 345-398, Springer, 1984.

[8] U.Engberg and G. Winskel, Petri Nets as Models of Linear Logic, June 1991.

[9] Jean-Yves Girard, Linear Logic, *Theoretical Computer Science 50*, pages 1-102, 1987.

[10] J.-Y. Girard, Linear Logic: its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*. London Mathematical Society Lecture Note Series, Cambridge University Press, 1995. Available by anonymous ftp from lmd.univ-mrs.fr as /pub/girard/Synsem.ps.Z.

[11] J.-Y. Girard, Y. Lafont, P. Taylor, *Proofs and Types*, volume 7 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, 1989.

[12] J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, London Mathematical Society Lecture Note Series, Cambridge University Press, 1995.

[13] J.-Y. Girard, M. Okada and A. Scedrov (eds.), Linear Logic Special Issues, Part I and Part II, (two issues), *Theoretical Computer Science*, to appear.

[14] C. A. Gunter and V. Gehlot. Nets as tensor theories. In G. De Michelis, editor, *Proc. 10-th International Conference on Application and Theory of Petri Nets, Bonn*, pages 174-191, 1989.

[15] J.P. Jouannaud and M. Okada, A Computation Model for Higher Order Algebraic Specification Languages, *LICS 91*, Amsterdam, 1991, a revised version in Theoretical Computer Science (1997).

[16] G. Kahn, The Semantics of a Simple Language for Parallel Programming, 993-998, *Information Processing 74* (1977).

[17] M. Kanovitch, M. Okada, A. Scedrov, Specifying Real-Time Finite State Systems Based on Linear Logic, *Electronic Notes of Theoretical Computer Science 16 Part I*, Elsevier, 1998.

[18] N. Kobayashi and A. Yonezawa, Higher-order concurrent linear logic programming. In *Theory and Practice of Parallel Programming*, pages 137-166, Springer LNCS 907, 1995.

[19] Y. Lafont, From Proof-Nets to Interaction Nets *Advances in Linear Logic*, London Mathematical Society Lecture Note Series, Cambridge University Press, 1995.

[20] P. Lincoln and N. Shankar, Proof search in first order linear logic and other cut-free sequent calculi, In *Proc. 9-th Annual IEEE Symposium on Logic in Computer Science, Paris*, pages 282-291, July 1991.

[21] N. Martí-Oliet and J. Meseguer, From Petri nets to linear logic. *Mathematical Structures in Computer Science*, 1:66-101, 1991. Revised version of paper in Springer LNCS 389.

[22] McCarthy, John, and Patrick J. Hayes, Some Philosophycal Problems from the Standpoint of Artificial Intelligence, *Machine Intelligence* 4, pages 463-502, 1969.

[23] D. Miller (ed.), Proceedings of the Workshop on Linear Logic and Logic Programming, Washinton D. C., 1992.

[24] R. Milner, *Communication and Concurrency*, Printice Hall, 1989.

[25] M. Okada, Phase Semantic Cut-Elimination and Normalization Proofs of First and Higher Order Linear Logic, Theoretical Computer Science 227 (1999), 333-396.

[26] M. Okada, Phase Semantics for Higher Order Completeness, Cut-Elimination and Normalization Proof (Extended Abstract), Electronic Notes in Theoretical Computer Science 3 (1996).

[27] M. Okada, *Lecture Notes on Linear Logic*, Tutorial Lecture Notes of JSSS, the spring 1993. Japan Society for Software Science. An extended version available at the following ftp site under the title Mobile Linear Logic: `http://abelard.flet.keio.ac.jp/person/mitsu`.

[28] H. Ono, Semantics for Substructural Logics, *Substructural Logics*, K. Došen and P. Schroeder-Heister eds., Oxford University Press, pages 259-291, 1994.

[29] A. Scedrov, A Brief Guide to Linear Logic, In G. Rozenberg and A. Salomaa, editors, *Current Trends in Theoretical Computer Science*, Pages 377-394, World Scientific Publishing Co., 1993.

[30] G. Takeuti, *Proof Theory*, North Holland, 1985.

[31] G. Takeuti, *Senkei Ronri Nuumon*, Nihon Hyoron Shya, 1995.

[32] A. Troelstra, *Lectures on Linear Logic*, CSLI Lecture Notes 29, 1992.