Temporal Linear Logic and Its Applications

Hirai, Takaharu

(Degree)
博士（理学）
(Date of Degree)
2000-09-30
(Date of Publication)
2014-11-25
(Resource Type)
doctoral thesis
(Report Number)
甲2193
(URL)
https://hdl.handle.net/20.500.14094/D1002193

Doctoral Dissertation

Temporal Linear Logic and Its Applications

Takaharu Hirai

September 2000

The Graduate School of Science and Technology
Kobe University, Japan

博士論文

Temporal Linear Logic and Its Applications

(時相線形論理とその応用)

平成 12 年 9 月

神戸大学大学院自然科学研究科

平　井　崇　晴

Abstract

Linear logic, introduced by Girard in 1987, has been called a *resource conscious* logic. In order to express a dynamic change in process environment, it is useful to consider a concept of resource such as data consumption. The expressive power of linear logic is evidenced by some very natural encodings of computational models such as Petri nets, counter machines, Turing machines, and others. For example, in Petri nets, tokens are considered as resources that are consumed and transitions are considered as reusable resources. It is well known that the *reachability* problem for ordinary Petri nets is equivalent to the *provability* for the corresponding sequent of linear logic. Also, as a formal logical system, linear logic satisfies some basic theorems. In it the cut elimination theorem and the soundness and completeness theorems for *phase semantics* which is a standard semantics of linear logic hold true. In particular, the cut elimination theorem can be applied to logic programming, uniform proof and proof search, and so on. We think that linear logic has been given various applications in computer science through its resource consciousness and usefulness as a formal system.

However, since linear logic does not include a concept of *time* directly, it is not enough to treat a dynamic change in environments with the passage of time such as execution time and waiting time. A typical example is the encoding of *timed Petri nets*. Although ordinary Petri nets can be encoded into linear logic naturally as stated above, the encoding of timed Petri nets into the corresponding sequent is too complex for linear logic since the reachability problem for timed Petri nets includes a time concept.

Thus, it can be considered to extend linear logic with respect to the time concept. The aim of this thesis is to construct a *resource-conscious* and *time-dependent* logical system by means of extending linear logic and to provide an application to computer science. We think that such a logic can treat a dynamic change in environments with the passage of time. We call it *temporal linear logic*.

The basic idea is to introduce temporal operators. We assume discrete linear time in this thesis and introduce "$\bigcirc$", which means *next*, and "$\Box$", which means *anytime*. We make the interpretation of a formula include a time concept. For inference rules, we refer to modal logic **S4**. **S4** can thus be regarded as *temporal logic*, which also has a "$\Box$" operator meaning "always", which is similar to ours.

In this way, temporal linear logic includes linear logic as its subsystem. It can succeed to the resource-consciousness in linear logic. Also, **S4** can be embedded into temporal linear logic. We can say that the time concept works in our logical system. In addition, the cut elimination theorem and the subformula property hold in this logic as in the form of linear logic. The full propositional fragment of temporal linear logic has a complete semantics in terms of *temporal phase spaces*, which are an extension of phase spaces in the semantics of linear logic.

Using our temporal linear logic, several cases of a dynamic change in environments with the passage of time can be expressed. Timed Petri nets can be encoded naturally into it, that is, the reachability problem for timed Petri nets is equivalent to the provability for the corresponding sequent. This is connected to the decidability of temporal linear logic fragments. A logic programming language based on temporal linear logic is designed by using the idea of Miller's uniform proof. We also represent the description of a communication model, which is our own model, by temporal linear logic. It can distinguish a synchronous calculus from an asynchronous calculus.

Keywords

temporal linear logic, phase semantics, timed Petri nets, reachability, synchronous communication, decidability.

Contents

1 Introduction **4**

2 A Survey of Linear Logic **8**
2.1 Linear Logic 8
2.2 Linear Logic and Petri Nets 9
2.3 Phase Semantics for Linear Logic 10

3 Syntax of Temporal Linear Logic **11**
3.1 Interpretation of Formulas in Temporal Linear Logic 11
3.2 Propositional Temporal Linear Logic 12
3.3 Fundamental Theorems 13
3.4 Decidability of Temporal Linear Logic Fragments 15

4 Phase Semantics for Temporal Linear Logic **16**
4.1 Temporal Phase Spaces 16
4.2 Soundness 19
4.3 Completeness 20
4.3.1 Canonical Model 20
4.3.2 The Main Lemma and the Completeness Theorem 22

5 Some Applications to Computer Science **26**
5.1 Synchronous Communication and Temporal Linear Logic 26
5.2 Timed Petri Nets and Temporal Linear Logic 28
5.2.1 Timed Petri Nets 28
5.2.2 Reachability and Provability 29
5.3 Decidability of the Reachability Problem for Timed Petri Nets 32

6 Conclusions and Future Work **34**

A Syntax **39**
A.1 Classical Propositional Linear Logic 39
A.2 Propositional Temporal Linear Logic 40
A.2.1 Classical Temporal Linear Logic 40
A.2.2 Intuitionistic Temporal Linear Logic 40
A.3 Classical Logic 41
A.4 S4 41

B Properties of TLL Phase Semantics **42**

C Proofs of Theorem, Corollary and Lemma in Phase Semantics **43**
C.1 Soundness 43
C.2 Corollary of the Main Lemma 48
C.3 The Main Lemma 51

List of Tables

3.1 Modal Rules 12

5.1 **HTLL** 30

A.1 **LL** 39
A.2 **TLL** 40
A.3 **ITLL** 40
A.4 **LK** 41
A.5 **S4** 41

Chapter 1

Introduction

In this thesis, a logical system called *temporal linear logic* is constructed and its application studied. The basic idea is to extend linear logic by means of introducing temporal operators. Dynamic change in environments with the passage of time can be expressed by temporal linear logic.

In this chapter, the aim and the outline of this thesis is described, starting with the background and the motivation. Following that, we list the characteristics of temporal linear logic and applications to computer science. At the end of this chapter, the organization of the thesis is described.

Background

In order to express dynamic change in process environment, it is useful to consider a concept of *resource* such as data consumption. *Linear logic* (**LL**) [6] introduced by Girard in 1987 has been called a *resource conscious* logic. The expressive power is so rich that one can construct a counter machine within the propositional fragment of linear logic. Some computational models of concurrency are applications of **LL** [26, 2, 20]. In particular, the relation between linear logic and Petri nets [11, 36] has been well-studied [16, 35, 21, 4, 23]. According to [25], the algebraic point of view of Petri nets seems to be related to the algebraic semantics for linear logic [28, 34]. **LL** has a modal storage operator "!" which means an infinite resource. Using this operator, one can distinguish the treatment of a reusable resource from the treatment of a consumptive resource. For example, a token in Petri nets is treated as a consumptive resource and a transition is expressed by using the "!" operator.

Furthermore, linear logic is useful as a formal logical system. The cut elimination theorem holds in **LL**. The theorem plays an important part in logic programming, uniform proof and proof search, and so on. The full propositional fragment of **LL** has a complete semantics in terms of *phase spaces* [6]. We think that linear logic has been given various applications in computer science through its resource-consciousness and usefulness as a formal system.

The Motivation and Aim of this Study

Linear logic can represent a dynamically changing environment. However, it is not enough to treat dynamic change in process environments with the passage of *time* such as execution time and waiting time because **LL** does not explicitly include modal operators of time. Although [18] is a study on introducing the time concept into linear logic on the first order level, it is necessary for a wide application to develop comprehensive logical systems capable of handling both resource conscious and time dependent properties.

A typical example is the relation between **LL** and Petri nets. It is well known that the *reachability* problem for Petri nets is equivalent to the *provability* for the corresponding sequent of **LL** [16]. Petri nets can be extended to *timed Petri nets* [12, 3] with respect to the time concept. Since timed Petri nets can be replaced into ordinary Petri nets [29, 30], the reachability problem for timed Petri nets can be expressed by **LL** indeed. But the mass of places and transitions are supplemented for the replacement. It follows that the expression of timed Petri nets by **LL** is too complex. We cannot say that **LL** has enough expressive power for natural encoding of timed Petri nets.

Thus it can be considered to extend linear logic with respect to the time concept. The aim of this thesis is to construct a *resource-conscious* and *time-dependent* logical system that can treat dynamic change in environments with the passage of time within its propositional fragment by means of extending linear logic and to provide an application to computer science. We call the logical system *temporal linear logic* (**TLL**)[1].

The Requirements of Temporal Linear Logic

There are several logical systems called "temporal linear logic" to date [13, 33], yet these systems lack a modal storage operator which means an infinite resource. It follows that even ordinary Petri nets cannot be expressed naturally. The one in [33] expresses transitions by non-logical axioms. Thus the correspondence between the logical system and (timed) Petri nets is not sufficient. That is, though the soundness of the provability of the sequent in the logical system with respect to the reachability problem for (timed) Petri nets is shown, the completeness is not shown in [33]. In addition, the one in [13] has an inference rule that includes an infinite sequence of sequents in the inference rules. For this, the cut elimination theorem is proved using semantics, that is, one cannot obtain the cut free proof figure for a provable sequent constructively. It follows that one cannot use the idea of Miller's uniform proof when a logic programming language based on the logic is designed. Thus it is difficult for the systems in [13] and [33] to express naturally dynamic change in process environments with the passage of time.

We consider that it is the first step toward a natural expression of dynamic change in environments with the passage of time to fuse linear logic and *temporal logic* without destroying their characteristics. Our basic idea is to introduce some temporal operators into linear logic and to add the rules concerning them. The additional rules should be an extension of temporal logic rules. Also, usefulness as a formal logical system of linear logic should be satisfied, that is, the cut free proof figure should be given constructively and **TLL** should have the soundness and completeness theorems. **TLL** should be able to express timed Petri nets as well as ordinary Petri nets.

The Characteristics of Temporal Linear Logic

In this thesis, we restrict the time concept to discrete linear time. We think that it will not be difficult to extend this concept to continuous linear time (Chapter 6).

We introduce "$\bigcirc$", which means *next*, and "$\Box$", which means *anytime*. Furthermore, we make the interpretation of a formula include the time concept. In linear logic, a formula A can be considered as a resource meaning "A can be used *exactly once*". In **TLL**, we interpret it as "A can be used exactly once *just now*". Thus, $\bigcirc A$ and $\Box A$ can be interpreted as follows:

$$\begin{aligned} \bigcirc A &: \text{“}A\text{ can be used next time exactly once”}, \\ \Box A &: \text{“}A\text{ can be used anytime but exactly once”}. \end{aligned}$$

[1]Temporal linear logic, which is an extension of linear logic, is different from *linear temporal logic* (**LTL**), a kind of temporal logic that has no relation with linear logic. The meaning of formula A in **TLL** is not the same that of formula A in **LTL** because **LTL** has no concept of resource. That of in **TLL** means "A is usable only once now". The one in **LTL** means "A is usable (any number of times) now".

Also, in linear logic, since a formula $!A$ can be considered an infinite resource, it may be thought of as a printing press for A's, which can generate any number of A's. Thus $!A$ can be interpreted as "A can be reusable" in linear logic. In **TLL**, the interpretation of this modal storage operator "!" is also extended, which means *reusable at anytime*, that is;

$$!A \quad : \quad \text{“}A \text{ can be reusable at anytime”}.$$

Now, we characterize **TLL** by modal logic **S4** [1], since it can be regarded as *temporal logic*, which also has a "$\Box$" operator meaning "always", which is similar to ours. In terms of the operator, $\Box A \to A$ and $\Box A \to \Box\Box A$ are provable in **S4**. We referred to these characterizations in order to construct the **TLL** syntax. Indeed, $\Box$-rules in **TLL** are extensions of $\Box$-rules in **S4**. We show that **S4** is embedded into **TLL**. Note that temporal logic by itself has no concept of resource.

Phase semantics is extended by two kinds of homomorphisms $h, f : M \longrightarrow M'$, where M and M' are phase spaces. We obtain *temporal phase spaces*. $h(m)$ means "m at next time" for $m \in M$ and f corresponds to "anytime". $!X \subseteq M$ in phase spaces is extended by f. We show that the full propositional fragment **TLL** satisfies the soundness and completeness theorems in terms of temporal phase spaces.

The characteristics of **TLL** are:

- A natural extension of linear logic:
  - **TLL** includes **LL** as its subsystem.
  - The cut elimination theorem holds in **TLL** as in the form of **LL**. Also, the subformula property holds in **TLL**. It follows that **LL** can be embedded into **TLL**. The concept of resource is succeeded.
  - Phase spaces are extended to temporal phase spaces. The soundness and completeness theorems hold in terms of them.
- A natural extension of temporal logic (**S4**):
  - **S4** is embedded into **TLL**. We can say that time concept works in **TLL**.

Application to Computer Science

TLL can be constructed as a resource-conscious and time-dependent logical system. We can obtain application to computer science concerning resource and time. For example, place timed Petri nets can be represented naturally. Suppose p is a place with waiting time for one unit, that is, tokens appearing in p just now will become available the next time. Thereafter the tokens can be used anytime. Thus a token appearing in p just now can be expressed by $\bigcirc\Box p$. A transition can be expressed using the "!" operator, as in encoding in linear logic.

Also, although the correspondence between the concept of the concurrent processes and logical concept is rough, we can represent the description of a communication model, which is our own model, by **TLL**. In linear logic, $A \multimap (B \multimap C)$ is equivalent to $A \otimes B \multimap C$. It follows that we cannot specify the execution order of processes. Using "$\bigcirc$" and "$\Box$", in **TLL**, we can specify the order as $A \multimap \bigcirc\Box(B \multimap \bigcirc\Box C)$. Furthermore, we can distinguish a synchronous calculus from an asynchronous calculus [10].

TLL can be applied as follows:

- It is possible to design a logic programming language based on temporal linear logic [32].
  Since **TLL** satisfies the cut elimination theorem, we can use the idea of Miller's uniform proof to design a logic programming language.

- It is possible to express timed Petri nets naturally.
  Since **TLL** has concepts of both resource and time, transitions and tokens in timed Petri nets are expressed naturally. The reachability problem for timed Petri nets is equivalent to the provability for the corresponding **TLL** sequent.

- It is possible to consider a synchronous communication calculus model, which is our own model. In the model, we can specify the execution order of processes and distinguish a synchronous calculus from an asynchronous calculus [10].

The remainder of this thesis is organized as follows. In Chapter 2, we survey linear logic. Some basic interpretations, the relation between linear logic and Petri nets, and phase semantics are stated. Readers knowledgeable about these matters may wish to skip this chapter. In Chapter 3, we provide the sequent calculus for propositional temporal linear logic. The cut elimination theorem and the embedding are shown. At the end of this chapter, we consider the decidability of **TLL** fragments. Chapter 4 consists of the phase semantics for **TLL**, and the soundness and completeness theorems are shown. Chapter 5 consists of application to computer science. It is shown that the reachability problem for timed Petri nets is equivalent to the provability for the corresponding **TLL** sequent (Theorem 5.2.1). We consider our own model and can distinguish a synchronous calculus from an asynchronous calculus in the model. Chapter 6 contains some remarks, related work and future work.

Chapter 2

A Survey of Linear Logic

In this chapter, starting with the meaning of some logical connectives, we review basic theorems, the decidability of linear logic fragments and the relation to Petri nets. We also look over phase semantics for linear logic. The syntax for linear logic (**LL**) is given in Table A.1 in Appendix A.1.

2.1 Linear Logic

First, let us review several logical connectives of linear logic. Consider the propositions D, C and T, conceived of as resources:

$$\begin{aligned} D &\stackrel{\triangle}{=} \text{“We have one Dollar”} \\ C &\stackrel{\triangle}{=} \text{“We can obtain a cup of Coffee”} \\ T &\stackrel{\triangle}{=} \text{“We can obtain a cup of Tea”} \end{aligned}$$

Consider the axiomatization of vending machines:

$$D \text{ implies } C, \quad D \text{ implies } T.$$

In ordinary logic (classical logic **LK** [5, 31]), one can deduce D implies (C and T) from D implies C and D implies T. It may be read as “With one dollar, we may buy both a cup of coffee and a cup of tea”. Although the deduction is valid in classical logic, it is nonsense. This paradox arises out of the confusion in classical logic between two kinds of conjunction. Linear Logic has two kinds of conjunction “$\otimes$” which means “we have both” and “&” which means “we have a choice”. Although D does not imply $C \otimes T$, $D \otimes D$ implies $C \otimes T$ in linear logic. On the other hand, D implies $C \& T$ means “We can *choose* C or T but not both”. Moreover, the traditional implication is refined as *linear implication* “$\multimap$”. $D \multimap C$ means “D is consumed to produce C”. Thus, we can say again that $D \multimap C \otimes T$ is not deducible in linear logic. Also linear logic has “$\oplus$” which means “someone else's choice”. $D \multimap C \oplus T$ means “With one dollar, we can obtain either a cup of coffee or a cup of tea, but we don't know which”. In addition, there is a modal storage operator “!”. $!D$ may be thought of as a printing press for D's, which can generate any number of D's. For example, the government can be thought to have $!D$. $!D \multimap \underbrace{C \otimes \ldots \otimes C}_{n}$ holds for arbitrary n, which means “We can obtain as many C's as we like”.

In **LL**, $\mathbf{1}$ is the unit of “$\otimes$”, thus $A \multimap A \otimes \mathbf{1}$ and $A \otimes \mathbf{1} \multimap A$, for any formula A. $\top$ is the unit of “&”, $\perp$ is the unit of “$\wp$”, and $\mathbf{0}$ is the unit of “$\oplus$”. For any formula A, B, C, one can prove $A \multimap B = A^{\perp} \wp B$, $A \multimap (B \multimap C) = A \otimes B \multimap C$, $!A \multimap A$, $!A = !!A$, $(!A)^{\perp} = ?A^{\perp}$, and so on. Here, “$A = B$” means both $A \multimap B$ and $B \multimap A$ are provable in **LL**. The cut elimination theorem holds in **LL**, that is, if a sequent $\Gamma \rightarrow \Delta$ is provable in **LL** then it can be provable in **LL** without the (*cut*) rule.

One can obtain its cut free proof figure constructively. The subformula property follows from the cut elimination theorem, that is, for any sequent $\Pi \to \Lambda$ in $\mathcal{P}$, any formula in Π or Λ is the subformula in Γ or Δ, where $\mathcal{P}$ is a cut free **LL** proof of $\Gamma \to \Delta$.

The full fragment of **LL** is undecidable [22]. The fragment of **LL** without modals "!" nor "?" (**MALL**) is known to be PSPACE-complete [22].

2.2 Linear Logic and Petri Nets

Linear logic provides a natural encoding of Petri nets reachability.

A Petri net is a tuple (Pl, Tr, Ar), where Pl indicates a finite set of places, Tr a finite set of transitions (disjoint with Pl) and Ar (Weight of arcs) : $(Pl \times Tr) \cup (Tr \times Pl) \longrightarrow \mathbf{N}$. Here $\mathbf{N}$ means the set of natural numbers (including 0).

A multiset M of places is called a *marking*, which indicates *tokens*. We say that a transition τ *is enabled* at M if and only if $M^- \subseteq M$. Here, M^- is the multiset of input places to τ. If a transition τ is enabled and we *fire* it, the reached marking M' is defined by

$$M' = M - M^- \uplus M^+.$$

Here, M^+ indicates the multiset of output places p's from τ. $\uplus$ indicates a multiset union. The described firing is denoted by the notation $M[\tau\rangle M'$.

For a firing sequence $\sigma = \tau_1 \ldots \tau_n$ $(n \geq 0)$, we use the notation $M_0[\sigma\rangle M$ instead of $M_0[\tau_1\rangle M_1[\tau_2\rangle M_2 \ldots M_{n-1}[\tau_n\rangle M_n = M$. We say that M is *reachable* from M_0 iff there exists a firing sequence σ such that $M_0[\sigma\rangle M$.

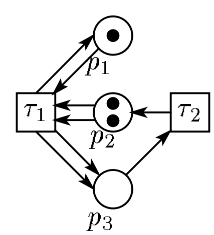

Fig.2.1: Petri net

We consider a Petri net in Fig. 2.1. The (initial) marking $M_0 = \{p_1, p_2, p_2\}$. The transition τ_1 is enabled at M_0. We fire it, then the reached marking $M_1 = \{p_1, p_3, p_3\}$. The transition τ_2 is enabled at M_1 and we fire it, then the reached marking $M = M_2 = \{p_1, p_2, p_3\}$. We can say that M is reachable from M_0 since $M_0[\tau_1\tau_2\rangle M$.

Using logical connectives of linear logic, one can represent Petri nets. The transition τ_1 in Fig. 2.1 is encoded to the formula $p_1 \otimes p_2 \otimes p_2 \multimap p_1 \otimes p_3 \otimes p_3$. Similary, the transition τ_2 is encoded to the formula $p_3 \multimap p_2$. Markings are represented as tensor product of atomic propositions. For example, the marking M_0 in Fig. 2.1 is encoded to the formula $p_1 \otimes p_2 \otimes p_2$. Similary, the marking M is encoded to the formula $p_1 \otimes p_2 \otimes p_3$.

It is well known that the reachability problem for Petri nets is decidable [24]. The reachability problem for Petri nets is equivalent to the provability problem for the !-Horn fragment of linear logic [16].

Theorem 2.2.1 (Kanovich [16]) *For a given Petri net (Pl, Tr, Ar), a marking M is reachable from a marking M_0 if and only if the following !-Horn sequent*

$$M_0^*, !Tr^* \to M^*$$

is provable in Linear Logic, where M_0^*, M^* are corresponding formulas and Tr^* is a corresponding sequence of formulas. ∎

Remark **2.2.1** The Horn fragment of linear logic is NP-complete [14, 15].

For a Petri net in Fig. 2.1, the reachability that M is reachable from M_0 is presented as a sequent

$$p_1 \otimes p_2 \otimes p_2, !(p_1 \otimes p_2 \otimes p_2 \multimap p_1 \otimes p_3 \otimes p_3), !(p_3 \multimap p_2) \to p_1 \otimes p_2 \otimes p_3.$$

Indeed, this sequent is provable in linear logic.

2.3 Phase Semantics for Linear Logic

In the previous section, we reviewed the correspondence between the provability of linear logic sequnet and the reachability problem for Petri nets. It can be considered that the resource-consciousness is one of the cause of the ability to express a dynamic change in a computational model of concurrency such as Petri nets.

We think that linear logic has another cause why it is applied to computer science widely. That is the usefulness as a formal logical system. The cut elimination theorem holds in **LL**. Furthermore, one can obtain the cut free proof figure constructively. This theorem plays an important part in logic programming, uniform proof, and proof search. The full propositional fragment **LL** has a complete semantics in terms of *phase spaces*[6]. We think that the soundness and completeness theorems can be useful to consider model checking.

In this section, we review the phase semantics for linear logic.

A *phase space* $(M, \perp)$ is a commutative monoid M with a distinguished subset $\perp \subseteq M$, called *bottom*. In a phase space $(M, \perp)$, we define

$$X^{\perp} := \{z \in M | z \cdot x \in \perp \text{ for any } x \in X\}$$

for any $X \subseteq M$. Immediately, $X \subseteq X^{\perp\perp}$ for any X, $Y^{\perp} \subseteq X^{\perp}$ whenever $X \subseteq Y$, and $X^{\perp\perp} \cdot Y^{\perp\perp} \subseteq (X \cdot Y)^{\perp\perp}$, where $X \cdot Y := \{x \cdot y | x \in X \text{ and } y \in Y\}$. A *fact* is an $X \subseteq M$ such that $X^{\perp\perp} = X$. It is straightforward to show that a phase space satisfies that X is a fact iff X is the form $Y^{\perp}$ for some $Y \subseteq M$. In particular $\perp = \{1\}^{\perp}$ is a fact, where 1 is the neutral element of M. Also, any intersection of facts is a fact. In particular, $X^{\perp\perp}$ is an intersection of all facts containing X. Note that all of the following are facts:

$$\begin{array}{cc} \mathbf{1} := \{1\}^{\perp\perp}, \quad \top := M, & \mathbf{0} := \emptyset^{\perp\perp}, \\ X \otimes Y := (X \cdot Y)^{\perp\perp}, & X \wp Y := (X^{\perp} \cdot Y^{\perp})^{\perp} \\ X \& Y := X \cap Y, & X \oplus Y := (X \cup Y)^{\perp\perp}, \\ \multicolumn{2}{c}{X \multimap Y := \{z \in M | x \cdot z \in Y \text{ for all } x \in X\},} \end{array}$$

for any facts X, Y.

It is easy to show that $J(M) := \{x \in \mathbf{1} | x \in \{x \cdot x\}^{\perp\perp}\}$ is a submonoid of M. Let K be a submonoid of $J(M)$. Note that K is not required to be a fact. For any fact $X \subseteq M$, we define following facts :

$$!X := (X \cap K)^{\perp\perp} \quad ?X := (X^{\perp} \cap K)^{\perp}.$$

One can deduce for any fact X:

$$!X \subseteq X, \quad !X \subseteq !X \otimes !X, \quad !X \subseteq \mathbf{1}.$$

A *phase model* is given by a phase space $(M, \perp)$ and a *valuation* which maps each (positive) atomic p of **LL** to a fact p^* of $(M, \perp)$. For each propositional formula A of **LL**, we can associate a fact A^* inductively, that is, $\perp^* := \perp$, $\mathbf{1}^* := \perp^{\perp} = \{1\}^{\perp\perp}$, $\top^* := M$, $\mathbf{0}^* := \emptyset^{\perp\perp}$, $(p^{\perp})^* := p^{*\perp}$, $(A \otimes B)^* := A^* \otimes B^*$, $(A \wp B)^* := A^* \wp B^*$, $(A \& B)^* := A^* \& B^*$, $(A \oplus B)^* := A^* \oplus B^*$, $(A \multimap B)^* := A^* \multimap B^*$, $(!A)^* := !A^*$, $(?A)^* := ?A^*$.

Let $C_1, \ldots, C_m \to D_1, \ldots, D_n$ be a sequent of **LL** and let $(\ )^*$ be a valuation. A valuation *satisfies* a sequent $C_1, \ldots, C_m \to D_1, \ldots, D_n$ iff $(C_1 \otimes \ldots \otimes C_m)^* \subseteq (D_1 \wp \ldots \wp D_n)^*$. A sequent of the form $\to D_1, \ldots, D_n$ is defined to be satisfied iff $\mathbf{1}^* \subseteq (D_1 \wp \ldots \wp D_n)^*$. A sequent of the form $C_1, \ldots, C_m \to$ is defined to be satisfied iff $(C_1 \otimes \ldots \otimes C_m)^* \subseteq \perp^*$. A sequent $C_1, \ldots, C_m \to D_1, \ldots, D_n$ is *valid* iff it is satisfied in any valuation in any phase model.

Theorem 2.3.1 (Girard [6]) *A sequent is provable in* **LL** *if and only if it is valid.* ■

Chapter 3

Syntax of Temporal Linear Logic

In this chapter, we define the syntax of propositional temporal linear logic (**TLL**). Note that the sequence of formulas $C_1, \ldots, C_m$ is regarded as a multiset, so that exchange is implicit for all logical systems in this thesis.

3.1 Interpretation of Formulas in Temporal Linear Logic

The syntax **TLL** is obtained from **LL** by introducing some temporal operators, and the rules concerning them. Here we assume discrete linear time and introduce the temporal operators "$\bigcirc$", which means *next* and "$\Box$", which means *anytime*. In order to introduce the time concept without destroying the fundamental characteristics of linear logic, we devise an interpretation of **TLL** formulas as follows:

A	:	"A can be used exactly once *just now*"
		(After use, it disappears);
$\bigcirc A$	:	"A can be used *next* time exactly once"
		(After use, it disappears);
$\Box A$	:	"A can be used *anytime* but exactly once"
		(After use, it disappears);
$!A$	:	"A can be *reused anytime*"
		(It never disappears).

Note that $!A$ does not mean "A can be reused *only now*"[1] in our system. If we use this interpretation, the meaning of $!\Box A$ is different from $\Box !A$. In our system, both of them are treated as having the same meaning. This is related to Remark 3.3.1. In a sense our interpretation does not damage the expressive power. Also it is suitable for the encoding of transitions in timed Petri nets.

Remember the axiomatization of vending machines in Section 2.2. We can consider various situations compared with the case in **LL**. The axiomatization of vending machines may be stated as follows:

$$\bigcirc^m D \text{ implies } \bigcirc^m C, \quad \bigcirc^n D \text{ implies } \bigcirc^n T,$$

for any m, n. Here, $\bigcirc^n C$ indicates $\underbrace{\bigcirc \ldots \bigcirc}_{n} C$. For example, we can express "With one dollar, we can buy a cup of coffee anytime"[2] by $\Box D \multimap \Box C$. $\Box D \otimes \Box D \multimap C \otimes C$ can be interpreted as "With two

[1] This is the early interpretation by Kanovich before [13]. In our system, this is expressed by $\mathbf{1} \& A \& (A \otimes A) \& \ldots \& \underbrace{(A \otimes \ldots \otimes A)}_{n}$ for any n.

[2] We consider here that one dollar, which you have now, can be used anytime but exactly once.

$$\frac{A, \Gamma \to \Delta}{\Box A, \Gamma \to \Delta} (\Box \to) \quad \frac{!\Gamma, \Box \Pi \to A, \Diamond \Lambda, ?\Sigma}{!\Gamma, \Box \Pi \to \Box A, \Diamond \Lambda, ?\Sigma} (\to \Box)$$

$$\frac{!\Gamma, \Box \Pi, A \to \Diamond \Lambda, ?\Sigma}{!\Gamma, \Box \Pi, \Diamond A \to \Diamond \Lambda, ?\Sigma} (\Diamond \to) \quad \frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, \Diamond A} (\to \Diamond)$$

$$\frac{!\Gamma, \Box \Pi, \Xi \to A, \Phi, \Diamond \Lambda, ?\Delta}{!\Gamma, \Box \Pi, \bigcirc \Xi \to \bigcirc A, \overline{\bigcirc} \Phi, \Diamond \Lambda, ?\Delta} (\bigcirc) \quad \frac{!\Gamma, \Box \Pi, \Xi, A \to \Phi, \Diamond \Lambda, ?\Delta}{!\Gamma, \Box \Pi, \bigcirc \Xi, \overline{\bigcirc} A \to \overline{\bigcirc} \Phi, \Diamond \Lambda, ?\Delta} (\overline{\bigcirc})$$

$$\frac{!\Gamma, \Box \Pi, \Xi \to \Phi, \Diamond \Lambda, ?\Delta}{!\Gamma, \Box \Pi, \bigcirc \Xi \to \overline{\bigcirc} \Phi, \Diamond \Lambda, ?\Delta} (\bigcirc \to \overline{\bigcirc})$$

Table 3.1: Modal Rules

dollars, we can buy two cups of coffee today", $\Box D \otimes \Box D \multimap C \otimes \bigcirc C$ as "With two dollars, we can buy a cup of coffee today and tomorrow", and $\Box D \multimap C \& \bigcirc C$ as "With one dollar, we have a choice of today or tomorrow (but not both) to buy a cup of coffee".

3.2 Propositional Temporal Linear Logic

Now, we shall define the syntax for **TLL**. We refer to modal logic **S4** [1], since it can be regarded as *temporal logic*, which also has a "$\Box$" operator meaning "always", which is similar to ours. $\Box$-rules in **TLL** are an extension of $\Box$-rules in **S4** which are listed in Table A.5 in Appendix A.4.

Roman capitals $A, B, \ldots$ stand for formulas. The connectives of **TLL** are:

- the *multiplicatives* $A \otimes B$, $A \wp B$, $A \multimap B$, $\bot$, $\mathbf{1}$;
- the *additives* $A \& B$, $A \oplus B$, $\top$, $\mathbf{0}$;
- the *exponentials* $!A$, $?A$;
- the *temporal modalities* $\Box A$, $\Diamond A$, $\bigcirc A$, $\overline{\bigcirc} A$.

The pairs $\otimes, \wp$; $\bot, \mathbf{1}$; $\&, \oplus$; $\top, \mathbf{0}$; $!, ?$; $\Box, \Diamond$; $\bigcirc, \overline{\bigcirc}$ are de Morgan duals. Greek capitals $\Gamma, \Pi, \ldots$ stand for sequences, which are multisets of formulas (including empty), so that exchange is implicit. $!\Gamma$ stands for the form $!C_1, \ldots, !C_m$. $?\Delta, \Box \Pi, \Diamond \Lambda, \bigcirc \Xi, \overline{\bigcirc} \Phi, \ldots$ stand for similar ones.

Definition 3.2.1 (TLL) The syntax for *propositional classical temporal linear logic* (**TLL**) is defined by adding *Modal Rules* to **LL**. The Modal Rules are listed in Table 3.1. ■

Obviously, **LL** is included in **TLL** as its subsystem.

Intuitionistic temporal linear logic (**ITLL**) is defined as a subsystem of **TLL** as follows:

Definition 3.2.2 (ITLL) The axioms and inference rules of *propositional intuitionistic temporal linear logic* (**ITLL**) are defined in Appendix Table A.3. ■

Note all right sides of each sequent of **ITLL**. There exists exactly one formula on each right side. It is different from standard intuitionistic logical systems such as **LJ** [5, 31]. This is from the phase semantics for **ITLL**.

The following are several syntactical remarks on **TLL** for any formulas of A, B.

- $\mathbf{1} \otimes A = A \otimes \mathbf{1} = A$,

- $!!A = !A$, $\Box\Box A = \Box A$,
- $!A \multimap \Box A$ is provable, but $\Box A \multimap !A$ is not provable,
- $\Box A \multimap \bigcirc^n A$ is provable, but $\bigcirc^n A \multimap \Box A$ is not provable ($n \geq 0$),
- Neither $\bigcirc A \multimap A$ nor $A \multimap \bigcirc A$ is provable.

"$A = B$" in the list means both $A \multimap B$ and $B \multimap A$ are provable in **TLL**.

3.3 Fundamental Theorems

The cut elimination theorem holds in both **TLL** and **ITLL**. Furthermore, one can obtain the cut free proof figure for a provable sequent constructively. This theorem plays an important part in logic programming [32], uniform proof and proof search.

Modal logic **S4** has a temporal operator which means "always". **S4** can be embedded into **TLL**. We can say that the time concept of this works in **TLL**.

Theorem 3.3.1 (Cut Elimination) *If a sequent $\Gamma \to \Delta$ is provable in* **TLL**, *then it is cut free provable in* **TLL**. ■

Proof. The proof is as in the linear logic case (see, for example, [34] for details).

Here we summarize the method.

First, we add $(!cut)$ and $(?cut)$ to **TLL**.

$$\frac{\Gamma \to \Delta, !D \quad (!D)^n, \Pi \to \Lambda}{\Gamma, \Pi \to \Delta, \Lambda}\ (!cut) \qquad \frac{\Gamma \to \Delta, (?D)^n \quad ?D, \Pi \to \Lambda}{\Gamma, \Pi \to \Delta, \Lambda}\ (?cut)$$

Here, $n > 0$, and $(D)^n$ indicates $\overbrace{D, \ldots, D}^{n}$. $(!cut)$ and $(?cut)$ can be reduced to ordinary (cut) rules. (cut),$(!cut)$ and $(?cut)$ are generically called simply "$(cuts)$". We define the *degree* of a $(cuts)$ as the number of logical symbols in a cut formula. Consider a proof figure that does not contain $(cuts)$ rules except for the last rule. The argument is by double induction on the numbers of inference rules in the proof figure, which is called *rank*, and the degree of the $(cuts)$. Below are several cases. Other cases are similar.

Case 1: The proof of which the last part is of the form

$$\frac{\dfrac{\begin{matrix}\vdots\\ !\Gamma_1 \to ?\Delta_1, D\end{matrix}}{!\Gamma_1 \to ?\Delta_1, !D}\ (\to !) \qquad \dfrac{\begin{matrix}\vdots\\ (!D)^n, !\Gamma, \Box\Pi \to A, \Diamond\Lambda, ?\Sigma\end{matrix}}{(!D)^n, !\Gamma, \Box\Pi \to \Box A, \Diamond\Lambda, ?\Sigma}\ (\to \Box)}{!\Gamma_1, !\Gamma, \Box\Pi \to \Box A, \Diamond\Lambda, ?\Delta_1, ?\Sigma}\ (!cut)$$

We replace this with

$$\frac{\dfrac{\dfrac{\begin{matrix}\vdots\\ !\Gamma_1 \to ?\Delta_1, D\end{matrix}}{!\Gamma_1 \to ?\Delta_1, !D}\ (\to !) \qquad \begin{matrix}\vdots\\ (!D)^n, !\Gamma, \Box\Pi \to A, \Diamond\Lambda, ?\Sigma\end{matrix}}{!\Gamma_1, !\Gamma, \Box\Pi \to A, \Diamond\Lambda, ?\Delta_1, ?\Sigma}\ (!cut)}{!\Gamma_1, !\Gamma, \Box\Pi \to \Box A, \Diamond\Lambda, ?\Delta_1, ?\Sigma}\ (\to \Box)$$

The rank of the $(!cut)$ has decreased.

Case 2: The proof of which the last part is of the form

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ !\Gamma_1, \Box\Pi_1, \Xi_1 \to D, \Phi_1, \Diamond\Lambda_1, ?\Delta_1\end{array}}{!\Gamma_1, \Box\Pi_1, \bigcirc\Xi_1 \to \bigcirc D, \overline{\bigcirc}\Phi_1, \Diamond\Lambda_1, ?\Delta_1}(\bigcirc) \quad \dfrac{\begin{array}{c}\vdots\\ !\Gamma_2, \Box\Pi_2, D, \Xi_2 \to A, \Phi_2, \Diamond\Lambda_2, ?\Delta_2\end{array}}{!\Gamma_2, \Box\Pi_2, \bigcirc D, \bigcirc\Xi_2 \to \bigcirc A, \overline{\bigcirc}\Phi_2, \Diamond\Lambda_2, ?\Delta_2}(\bigcirc)}{!\Gamma_1, !\Gamma_2, \Box\Pi_1, \Box\Pi_2, \bigcirc\Xi_1, \bigcirc\Xi_2 \to \bigcirc A, \overline{\bigcirc}\Phi_1, \overline{\bigcirc}\Phi_2, \Diamond\Lambda_1, \Diamond\Lambda_2, ?\Delta_1, ?\Delta_2}(cut)$$

We replace this with

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ !\Gamma_1, \Box\Pi_1, \Xi_1 \to D, \Phi_1, \Diamond\Lambda_1, ?\Delta_1\end{array} \quad \begin{array}{c}\vdots\\ !\Gamma_2, \Box\Pi_2, D, \Xi_2 \to A, \Phi_2, \Diamond\Lambda_2, ?\Delta_2\end{array}}{!\Gamma_1, !\Gamma_2, \Box\Pi_1, \Box\Pi_2, \Xi_1, \Xi_2 \to A, \Phi_1, \Phi_2, \Diamond\Lambda_1, \Diamond\Lambda_2, ?\Delta_1, ?\Delta_2}(cut)}{!\Gamma_1, !\Gamma_2, \Box\Pi_1, \Box\Pi_2, \bigcirc\Xi_1, \bigcirc\Xi_2 \to \bigcirc A, \overline{\bigcirc}\Phi_1, \overline{\bigcirc}\Phi_2, \Diamond\Lambda_1, \Diamond\Lambda_2, ?\Delta_1, ?\Delta_2}(\bigcirc)$$

The degree of the (cut) has decreased.

Case 3: The proof of which the last part is of the form

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ !\Gamma_1, \Box\Pi_1, \Xi_1 \to D, \Phi_1, \Diamond\Lambda_1, ?\Delta_1\end{array}}{!\Gamma_1, \Box\Pi_1, \bigcirc\Xi_1 \to \bigcirc D, \overline{\bigcirc}\Phi_1, \Diamond\Lambda_1, ?\Delta_1}(\bigcirc) \quad \dfrac{\begin{array}{c}\vdots\\ !\Gamma_2, \Box\Pi_2, D, \Xi_2 \to \Phi_2, \Diamond\Lambda_2, ?\Delta_2\end{array}}{!\Gamma_2, \Box\Pi_2, \bigcirc D, \bigcirc\Xi_2 \to \overline{\bigcirc}\Phi_2, \Diamond\Lambda_2, ?\Delta_2}(\bigcirc \to \overline{\bigcirc})}{!\Gamma_1, !\Gamma_2, \Box\Pi_1, \Box\Pi_2, \bigcirc\Xi_1, \bigcirc\Xi_2 \to \overline{\bigcirc}\Phi_1, \overline{\bigcirc}\Phi_2, \Diamond\Lambda_1, \Diamond\Lambda_2, ?\Delta_1, ?\Delta_2}(cut)$$

We replace this with

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ !\Gamma_1, \Box\Pi_1, \Xi_1 \to D, \Phi_1, \Diamond\Lambda_1, ?\Delta_1\end{array} \quad \begin{array}{c}\vdots\\ !\Gamma_2, \Box\Pi_2, D, \Xi_2 \to \Phi_2, \Diamond\Lambda_2, ?\Delta_2\end{array}}{!\Gamma_1, !\Gamma_2, \Box\Pi_1, \Box\Pi_2, \Xi_1, \Xi_2 \to \Phi_1, \Phi_2, \Diamond\Lambda_1, \Diamond\Lambda_2, ?\Delta_1, ?\Delta_2}(cut)}{!\Gamma_1, !\Gamma_2, \Box\Pi_1, \Box\Pi_2, \bigcirc\Xi_1, \bigcirc\Xi_2 \to \overline{\bigcirc}\Phi_1, \overline{\bigcirc}\Phi_2, \Diamond\Lambda_1, \Diamond\Lambda_2, ?\Delta_1, ?\Delta_2}(\bigcirc \to \overline{\bigcirc})$$

The degree of the (cut) has decreased.

(Q.E.D.)

Remark **3.3.1** Consider a logical system that has $(\to \Box)^*$ and $(\Diamond \to)^*$ rules instead of $(\to \Box)$ and $(\Diamond \to)$ in **TLL**.

$$\dfrac{!\Box\Gamma, \Box\Sigma \to A, \Diamond\Delta, ?\Diamond\Pi}{!\Box\Gamma, \Box\Sigma \to \Box A, \Diamond\Delta, ?\Diamond\Pi}(\to \Box)^* \quad \dfrac{!\Box\Gamma, \Box\Sigma, A \to \Diamond\Delta, ?\Diamond\Pi}{!\Box\Gamma, \Box\Sigma, \Diamond A \to \Diamond\Delta, ?\Diamond\Pi}(\Diamond \to)^*$$

In this logical system, $!\Box A$ is not the same as $\Box!A$. Also, the cut elimination fails. The sequent

$$!(p\&\Box q) \to \Box!\Box q$$

is provable, where p and q are propositional atoms. Indeed,

$$\dfrac{\dfrac{\dfrac{\dfrac{\Box q \to \Box q}{p\&\Box q \to \Box q}(\& \to)2}{!(p\&\Box q) \to \Box q}(! \to)}{!(p\&\Box q) \to !\Box q}(\to !) \quad \dfrac{!\Box q \to !\Box q}{!\Box q \to \Box!\Box q}(\to \Box)^*}{!(p\&\Box q) \to \Box!\Box q}(cut) \quad .$$

But the sequent has no cut-free proofs [8].

It is obvious that the cut elimination theorem holds in **ITLL** by Theorem 3.3.1. The subformula property follows from Theorem 3.3.1.

Corollary 3.3.1 (Subformula property) *Let $\mathcal{P}$ be a cut free* **TLL** *proof of $\Gamma \to \Delta$. For any sequent $\Pi \to \Lambda$ in $\mathcal{P}$, any formula in Π or Λ is the subformula in Γ or Δ.* ∎

It is obvious that the subformula property holds in **ITLL** by Corollary 3.3.1. The subformula property is used in Section 3.4.

S4 can be embedded into the subsystem of **TLL** [8].

Theorem 3.3.2 (embedding) *Suppose* **TLL′** *is a subsystem of* **TLL** *constructed by excluding the modal operators* ○ *and* $\overline{○}$, *and the inference rules concerning them. Then*

$$\mathbf{S4}\ \vdash \Gamma \to \Delta \ \overset{\text{iff}}{\Leftrightarrow}\ \mathbf{TLL'}\ \vdash \Gamma^- \to \Delta^+,$$

where

P^+	:=	$?P$	P^-	:=	$!P$	(for P atomic)
$(\neg B)^+$	:=	$?B^{-\perp}$	$(\neg B)^-$	:=	$!B^{+\perp}$	
$(B \supset C)^+$	:=	$?(B^- \multimap C^+)$	$(B \supset C)^-$	:=	$!(B^+ \multimap C^-)$	
$(B \wedge C)^+$	:=	$?(B^+ \& C^+)$	$(B \wedge C)^-$	:=	$!(B^- \& C^-)$	
$(B \vee C)^+$	:=	$?(B^+ \oplus C^+)$	$(B \vee C)^-$	:=	$!(B^- \oplus C^-)$	
$(\Box B)^+$	:=	$?\Box B^+$	$(\Box B)^-$	:=	$!\Box B^-$	
$(\Diamond B)^+$	:=	$?\Diamond B^+$	$(\Diamond B)^-$	:=	$!\Diamond B^-$	

∎

3.4 Decidability of Temporal Linear Logic Fragments

In this section, we state with respect to the decidability of **TLL** fragments.

The full fragment of propositional linear logic is known to be undecidable [22]. Obviously, the full fragment of **TLL** is also undecidable since it contains a full fragment of propositional linear logic.

MALL is the fragment of propositional linear logic that contains the multiplicative connectives, "$\otimes$" and "$\wp$", the additive connectives, "$\&$" and "$\oplus$", the constants $\mathbf{0}, \mathbf{1}, \top$ and $\perp$, but excludes the modal storage operators "!" and "?". **MALL** is known to be PSPACE-complete [22]. By Corollary 3.3.1, if a sequent that has no modal operators is provable in **TLL**, then it is also provable in **MALL**. Since **MALL** is known to be decidable, a **TLL** sequent without modal operators is also decidable.

The reachability problem for Petri nets is known to be decidable [24]. Kanovich obtained the decidability of the !-Horn fragment of linear logic by means of showing the equivalence between !-Horn sequent provability and the reachability problem for Petri nets [16]. The Horn like sequent of the form (5.2.3) can be rewritten as the !-Horn sequent in [16], which does not contain "○" nor "□" [9, 10]. (5.2.3) is in Subsection 5.2.1. This result implies that we can decide whether a sequent of the form (5.2.3) is provable in **ITLL**°, where **ITLL**° is a subsystem of **TLL** which appears in Subsection 5.2.2. The temporal linear logic system in [33] does not satisfy the completeness theorem for timed Petri nets because it includes non logical axioms. Therefore, decidability of the logical system in [33] has not been established yet.

[15] researched several kinds of Horn fragments of linear logic, for example, the $(\oplus, \&)$-Horn fragment, the $\&$-Horn fragment, the $\oplus$-Horn fragment, and so on. Using a similar method, we may obtain results concerning decidability of various kinds of Horn fragments of **TLL**. We plan to do research on this topic in future.

Chapter 4

Phase Semantics for Temporal Linear Logic

Phase semantics is a standard semantics for linear logic. The full propositional fragment **LL** has a complete semantics in terms of *phase spaces*[6, 7]. In this chapter, we extend phase spaces to *temporal phase spaces*. For phase spaces $(M, \perp)$ and $(M', \perp')$, we consider two kinds of homomorphisms $h, f : M \longrightarrow M'$. $h(m)$ means "m at next time" for $m \in M$ and f corresponds to "anytime". The definition of $!X \subseteq M$ in temporal phase spaces is obtained by adding f to the one in phase spaces. We show that the full propositional fragment **TLL** satisfies the soundness and completeness theorems in terms of temporal phase spaces. Also, we consider *temporal phase structures*, which are a generalization of temporal phase spaces. We show that **ITLL** satisfies the soundness and completeness theorems in terms of temporal phase structures.

In [13], the completeness theorem has been given in the strong form: if a sequent is valid then it is provable without the cut rule in their temporal linear logic. They have shown the cut elimination theorem by the strong completeness theorem. On the other hand, we have shown the cut elimination in Chapter 3 independently from the completeness theorem. In this chapter, we show the completeness theorem in the standard form.

4.1 Temporal Phase Spaces

A *phase space* $(M, \perp)$ is a commutative monoid M with a distinguished subset $\perp \subseteq M$, called *bottom*. In phase space $(M, \perp)$, we define

$$X^{\perp} := \{z \in M | z \cdot x \in \perp \text{ for any } x \in X\}$$

for any $X \subseteq M$. Immediately, $X^{\perp}$ has the following properties: $X \subseteq X^{\perp\perp}$ for any X, $Y^{\perp} \subseteq X^{\perp}$ whenever $X \subseteq Y$, and $X^{\perp\perp} \cdot Y^{\perp\perp} \subseteq (X \cdot Y)^{\perp\perp}$ where $X \cdot Y := \{x \cdot y | x \in X \text{ and } y \in Y\}$.

A *phase structure* (M, Cl) is a commutative monoid M with a *closure* operator Cl on M, that is, a mapping Cl from subsets of M to subsets of M satisfying the following properties for any $X, Y \subseteq M$:

1. $X \subseteq Cl(X)$,
2. $Cl(Cl(X)) \subseteq Cl(X)$,
3. if $X \subseteq Y$ then $Cl(X) \subseteq Cl(Y)$,
4. $Cl(X) \cdot Cl(Y) \subseteq Cl(X \cdot Y)$.

Clearly, a phase space is a special case of a phase structure with closure defined as $Cl(X) := X^{\perp\perp}$.

In a phase structure (M, Cl), a subset $X \subseteq M$ is said to be a *fact* iff $Cl(X) = X$. It is straightforward to show that a phase space satisfies that X is a fact iff X is the form $Y^{\perp}$ for some $Y \subseteq M$. In particular $\perp = \{1\}^{\perp}$ is a fact, where 1 is the neutral element of M. Also, any intersection of facts is a fact. In particular, $X^{\perp\perp}$ is an intersection of all facts containing X.

We consider several properties of a mapping from M to M', where M and M' are commutative monoids. $h : M \longrightarrow M'$ is called a *monoid homomorphism* iff for any $m, n \in M$, $h(m \cdot n) = h(m) \cdot' h(n)$, and $h(1) = 1'$, where $1 \in M$ and $1' \in M'$ are neutral elements respectively. In particular, a *phase homomorphism* is a monoid homomorphism $h : M \longrightarrow M'$ such that $h(\perp) \subseteq \perp'$, where $\perp \subseteq M$ and $\perp' \subseteq M'$.

For a given mapping $g : M \longrightarrow M$, let us consider its lower approximations (after [17]).

Definition 4.1.1 Let M be a commutative monoid. For a given mapping $g : M \longrightarrow M$, a mapping $f : M \longrightarrow M$ is *bounded by* g iff for every $n \in M$ there exists $m \in M$ such that $m \preceq n, f(n) \preceq g(m)$, where $x \preceq y$ iff $Cl(\{x\}) \subseteq Cl(\{y\})$. ∎

Now, we define a *temporal phase space* as follows:

Definition 4.1.2 Let $(M, \perp)$ be a phase space, $h : M \longrightarrow M$ be a phase homomorphism and $f : M \longrightarrow M$ a monoid homomorphism. A *temporal phase space*$((M, \perp), h, f)$ is a phase space $(M, \perp)$ with h, f such that

1. f is bounded by h,
2. $f(f(m)) = f(m)$ for all $m \in M$. ∎

A *temporal phase structure* $((M, Cl), h, f)$ is defined similarly, but h is only required to be a monoid homomorphism. Obviously, a temporal phase space is a special case of a temporal phase structure.

We consider $J(M) := \{x \in \mathbf{1} | x \in Cl(\{x \cdot x\})\}$, where $\mathbf{1} := Cl(\{1\})$. It is easy to see that $J(M)$ is a submonoid of M. Let K be a submonoid of $J(M)$. Note that K is not required to be a fact.

Definition 4.1.3 Given a temporal phase structure, we define:

$$
\begin{aligned}
\mathbf{1} := Cl(\{1\}), \quad \top := M, \quad & \mathbf{0} := Cl(\emptyset), \\
X \otimes Y &:= Cl(X \cdot Y), \\
X \& Y &:= X \cap Y, \\
X \oplus Y &:= Cl(X \cup Y), \\
X \multimap Y &:= \{z \in M | x \cdot z \in Y \text{ for all } x \in X\}, \\
\bigcirc X &:= Cl(h(X)), \\
\Box X &:= Cl(X \cap f(X)), \\
!X &:= Cl(X \cap f(X) \cap K),
\end{aligned}
$$

where $X, Y \subseteq M$ are arbitrary facts. ∎

In a temporal phase space we further define:

$$
\begin{aligned}
X \wp Y &:= (X^{\perp} \cdot Y^{\perp})^{\perp} = (X^{\perp} \otimes Y^{\perp})^{\perp}, \\
?X &:= (X^{\perp} \cap f(X^{\perp}) \cap K)^{\perp} = (!X^{\perp})^{\perp}, \\
\Diamond X &:= (X^{\perp} \cap f(X^{\perp}))^{\perp} = (\Box X^{\perp})^{\perp}, \\
\overline{\bigcirc} X &:= h(X^{\perp})^{\perp} = (\bigcirc X^{\perp})^{\perp},
\end{aligned}
$$

where $X, Y \subseteq M$ are arbitrary facts.

It is straightforward to show that all of $\mathbf{1}$, $\top$, $\mathbf{0}$, $X \otimes Y$, $X \& Y$, $X \oplus Y$, $X \multimap Y$, $\bigcirc X$, $\Box X$, $!X$, $X \wp Y$, $?X$, $\Diamond X$ and $\overline{\bigcirc} X$ are facts.

We have some properties similar to the case of linear logic. They are listed in Appendix B. The properties are helpful to prove the soundness theorem. We can focus on modal rules only. In addition, we have some properties concerning modal operators. Before showing those, we prepare a lemma on bounded mapping.

Lemma 4.1.1 *Let $A \subseteq M$ be a fact. If $f, h : M \longrightarrow M$ and f is bounded by h, then*

$$Cl(f(A)) \subseteq Cl(h(A)).$$

∎

Proof. Suppose $a' \in f(A)$. Then there exists $a \in A$ such that $a' = f(a)$. For the a, there exists $b \in M$ such that

$$b \preceq a \text{ and } f(a) \preceq h(b)$$

since f is bounded by h. We have

$$b \preceq a \subseteq Cl(A) = A.$$

Hence,

$$h(Cl(\{b\})) \subseteq h(A).$$

For the $b \in \{b\} \subseteq Cl(\{b\})$,

$$h(b) \in h(Cl(\{b\})) = \{h(c) | c \in Cl(\{b\})\},$$

that is, $\{h(b)\} \subseteq h(Cl(\{b\}))$. Hence,

$$Cl(\{h(b)\}) \subseteq Cl(h(Cl(\{b\}))) \subseteq Cl(h(A)).$$

Also we have

$$a' = f(a) \in f(a) \preceq h(b).$$

Therefore,

$$a' \in Cl(\{h(b)\}) \subseteq Cl(h(A)).$$

For the reasons stated above, we obtain that $f(A) \subseteq Cl(h(A))$. Hence,

$$Cl(f(A)) \subseteq Cl(h(A)).$$

(Q.E.D.)

Proposition 4.1.1 and Lemma 4.1.2, which are properties concerning modal operators, are straightforward.

Proposition 4.1.1 *Let A, B be facts. In any temporal phase space,*

$$(!A)^\perp = ?A^\perp, (?A)^\perp = !A^\perp, (\Box A)^\perp = \Diamond A^\perp, (\Diamond A)^\perp = \Box A^\perp, (\bigcirc A)^\perp = \overline{\bigcirc} A^\perp, (\overline{\bigcirc} A)^\perp = \bigcirc A^\perp.$$

∎

Lemma 4.1.2 *In any temporal phase structure,*

$$\begin{array}{lll} !A \subseteq \mathbf{1}, & !A \subseteq !A \otimes !A, & !A \subseteq A, \\ !A = !!A, & \text{if } !A \subseteq B \text{ then } !A \subseteq !B, & !A \otimes !B = !(A \& B), \\ \Box A \subseteq \Box\Box A, & !A \subseteq \Box A \subseteq \bigcirc^{(n)} A \quad (n \geq 0), & \bigcirc A \subseteq \overline{\bigcirc} A. \end{array}$$

∎

Also dual statements are satisfied, that is,

$$\begin{array}{lll} \bot \subseteq ?A, & ?A \wp ?A \subseteq ?A, & A \subseteq ?A, \\ ?A = ??A, & \text{if } A \subseteq ?B \text{ then } ?A \subseteq ?B, & ?A \wp ?B = ?(A \oplus B), \\ \Diamond\Diamond A \subseteq \Diamond A, & \overline{\bigcirc}^{(n)} A \subseteq \Diamond A \subseteq ?A \quad (n \geq 0), & \end{array}$$

in any temporal phase space.

A *temporal space model* is given by a temporal phase space $((M, \bot), h, f)$ and a *valuation* that maps each (positive) atomic p of **TLL** to a fact p^* of $((M, \bot), h, f)$. For each propositional formula A of **TLL**, we can associate a fact A^* inductively:

- $\bot^* := \bot$, $\mathbf{1}^* := \bot^\bot = \{1\}^{\bot\bot}$, $\top^* := M$, $\mathbf{0}^* := \emptyset^{\bot\bot}$, $(p^\bot)^* := p^{*\bot}$,
- $(A \otimes B)^* := A^* \otimes B^*$, $(A \wp B)^* := A^* \wp B^*$,
- $(A \& B)^* := A^* \& B^*$, $(A \oplus B)^* := A^* \oplus B^*$,
- $(A \multimap B)^* := A^* \multimap B^*$,
- $(\bigcirc A)^* := \bigcirc A^*$, $(\overline{\bigcirc} A)^* := \overline{\bigcirc} A^*$, $(\Box A)^* := \Box A^*$, $(\Diamond A)^* := \Diamond A^*$,
- $(!A)^* := !A^*$, $(?A)^* := ?A^*$.

A^* is called the *inner value* of A.

Similarly, we can consider a *temporal structure model* for a temporal phase structure $((M, Cl), h, f)$ and a valuation that maps each atomic p of **ITLL** to a fact p^* of $((M, Cl), h, f)$.

Proposition 4.1.2 *In any temporal space model,* $(A^\bot)^* = A^{*\bot}$. ∎

Proof. The argument is by induction on the structure of the formula A.

The cases where A is a constant are obvious. Other cases are obtained by $A \multimap B = A^\bot \wp B$ and Proposition 4.1.1. For example, $((B \otimes C)^\bot)^* = (B^\bot \wp C^\bot)^* = (B^\bot)^* \wp (C^\bot)^* = B^{*\bot} \wp C^{*\bot}$ since $(B^\bot)^* = B^{*\bot}$, $(C^\bot)^* = C^{*\bot}$ by the induction hypothesis. (Q.E.D.)

Now, we define the concept of *valid*.

Definition 4.1.4 (valid) Let $C_1, \ldots, C_m \to D_1, \ldots, D_n$ be a sequent of **TLL** and $(\ )^*$ be a valuation.

A valuation *satisfies* a sequent $C_1, \ldots, C_m \to D_1, \ldots, D_n$ iff $(C_1 \otimes \ldots \otimes C_m)^* \subseteq (D_1 \wp \ldots \wp D_n)^*$. A sequent of the form $\to D_1, \ldots, D_n$ is defined to be satisfied iff $\mathbf{1}^* \subseteq (D_1 \wp \ldots \wp D_n)^*$. A sequent of the form $C_1, \ldots, C_m \to$ is defined to be satisfied iff $(C_1 \otimes \ldots \otimes C_m)^* \subseteq \bot^*$.

A sequent $C_1, \ldots, C_m \to D_1, \ldots, D_n$ is *valid* iff it is satisfied in any valuation in any temporal space model. ∎

We can consider the similarities between a sequent of **ITLL** and a temporal structure model.

4.2 Soundness

Now, we are ready to discuss the soundness theorem.

Theorem 4.2.1 (Soundness) *If a sequent is provable in* **TLL**, *then it is valid.* ∎

Proof. The argument is by induction on the length of **TLL** proof. See Appendix C.1 for details. (Q.E.D.)

Remark **4.2.1** In the proof of Theorem 4.2.1 in Appendix C.1, we do not use the property $h(\bot) \subseteq \bot$ except for Case 7 concerning the rule $(\bigcirc \to \overline{\bigcirc})$.

By Remark 4.2.1, it follows that the soundness theorem also holds in **ITLL**.

Corollary 4.2.1 (Soundness) *If a sequent is provable in* **ITLL**, *then it is valid.* ∎

4.3 Completeness

In this section, we show the completeness theorem. The **ITLL** version follows from the **TLL** version.

4.3.1 Canonical Model

We start by preparing a canonical model. Let $\hat{M}$ be a set of sequences of formulas, that is,

$$\hat{M} := \{\Gamma, \Delta, \ldots\}.$$

We write $\vdash_{\mathcal{S}} \Gamma \to \Delta$ for "$\Gamma \to \Delta$ is provable in system $\mathcal{S}$". If there is no confusion, we omit $_{\mathcal{S}}$ for the sake of readability.

We can consider $\hat{M}$ as a commutative monoid of which the product of Γ by Δ is "Γ, Δ". The unit is the empty sequence ϵ.

For subset $X \subseteq \hat{M}$, $X^{\perp\perp}$ is a closure, as stated in Section 4.1. It is not difficult to say that

$$\Gamma \in X^{\perp\perp} \ \text{ iff } \ \forall\Delta(\forall\Pi \in X(\vdash_{\mathbf{TLL}} \Pi \to \Delta) \Rightarrow \vdash_{\mathbf{TLL}} \Gamma \to \Delta). \tag{4.3.1}$$

We use this paraphrasing depending on the situation.

Given the sequence of formulas Δ, we define the *outer value* $||\Delta||$ as

$$||\Delta|| := \{\Gamma | \vdash_{\mathbf{TLL}} \Gamma \to \Delta\}.$$

We take $\hat{\perp} \subseteq \hat{M}$ as the subset $||\epsilon||$, that is,

$$\hat{\perp} \ := \ ||\epsilon||.$$

After this, for $\Delta = D_1, \ldots, D_m$ $(m \geq 0)$, we use the notation $\Delta^\perp$ as $D_1^\perp, \ldots, D_m^\perp$.

Proposition 4.3.1 *For any sequence of formulas Δ, $||\Delta||$ is a fact.* ∎

Proof. We will show that

$$||\Delta||^{\perp\perp} \subseteq ||\Delta||.$$

At first, we show $\Delta^\perp \in ||\Delta||^\perp$.

Let $\Sigma \in ||\Delta||$, that is, $\vdash \Sigma \to \Delta$ by definition. Since we can deduce

$$\frac{\Sigma \to \Delta}{\Sigma, \Delta^\perp \to \ },$$

we obtain $\Sigma, \Delta^\perp \in ||\epsilon|| = \hat{\perp}$. Thus, $\Delta^\perp \in ||\Delta||^\perp$ since we can say that $\Sigma, \Delta^\perp \in \hat{\perp}$ for any $\Sigma \in ||\Delta||$.

Now, let $\Gamma \in ||\Delta||^{\perp\perp}$, that is, if $\Pi \in ||\Delta||^\perp$ then $\Pi, \Gamma \in \hat{\perp}$ for any Π. We can take $\Delta^\perp$ as Π. Hence, $\Gamma, \Delta^\perp \in \hat{\perp} = ||\epsilon||$, that is, $\vdash \Gamma \to \Delta$ from $\vdash \Gamma, \Delta^\perp \to$. Therefore, $\Gamma \in ||\Delta||$. (Q.E.D.)

We define the mapping $\hat{h} : \hat{M} \longrightarrow \hat{M}$ as

$$\begin{aligned} \hat{h}(\epsilon) &= \epsilon, \\ \hat{h}(A) &= \bigcirc A, \\ \hat{h}(C_1, \ldots, C_m) &= \hat{h}(C_1), \ldots, \hat{h}(C_m). \end{aligned}$$

Obviously, $\hat{h}$ is a monoid homomorphism.

Proposition 4.3.2 *$\hat{h}$ is a phase homomorphism.* ∎

Proof. We show that $\hat{h}(\hat{\perp}) \subseteq \hat{\perp}$ where $\hat{\perp} = ||\epsilon||$, that is, if $\vdash_{\mathbf{TLL}} \Delta \to$ then $\vdash_{\mathbf{TLL}} \hat{h}(\Delta) \to$.

Case 1: Δ is ϵ. Then $\hat{h}(\epsilon) = \epsilon$ and this case is a tautology.

Case 2: Δ is of the form $C_1, \ldots, C_m$ $(m \geq 1)$. If $\vdash_{\mathbf{TLL}} C_1, \ldots, C_m \rightarrow$, then $\vdash_{\mathbf{TLL}} \bigcirc C_1, \ldots, \bigcirc C_m \rightarrow$ by the $(\bigcirc \rightarrow \overline{\bigcirc})$ rule, that is $\vdash_{\mathbf{TLL}} \hat{h}(\Delta) \rightarrow$. (Q.E.D.)

We define the mapping $\hat{f} : \hat{M} \longrightarrow \hat{M}$ as

$$\begin{aligned} \hat{f}(\epsilon) &= \epsilon, \\ \hat{f}(A) &= \begin{cases} A & \text{if } A \text{ is of the form } !B \text{ or } \Box B \\ \Box A & \text{otherwise} \end{cases} \\ \hat{f}(C_1, \ldots, C_m) &= \hat{f}(C_1), \ldots, \hat{f}(C_m). \end{aligned}$$

Obviously, $\hat{f}$ is also a monoid homomorphism and $\hat{f}(\hat{f}(\Gamma)) = \hat{f}(\Gamma)$ for any $\Gamma \in \hat{M}$.

Proposition 4.3.3 (bounded) *$\hat{f}$ is bounded by $\hat{h}$.* ∎

Proof. We will show that for every $\Gamma \in \hat{M}$ there exists Γ itself as Σ such that $\Sigma \preceq \Gamma$ and $\hat{f}(\Gamma) \preceq \hat{h}(\Sigma)$. Since $\Gamma \preceq \Gamma$ is obvious, we show only $\hat{f}(\Gamma) \in \{\hat{h}(\Gamma)\}^{\perp\perp}$, which concludes $\hat{f}(\Gamma) \preceq \hat{h}(\Sigma)$.

Let $\Gamma = !\Gamma_1, \Box\Gamma_2, \Gamma_3$, where Γ_3 does not contain formulas of the form $!B$ nor $\Box B$. Then $\hat{f}(\Gamma) = !\Gamma_1, \Box\Gamma_2, \Box\Gamma_3$ and $\hat{h}(\Gamma) = \bigcirc!\Gamma_1, \bigcirc\Box\Gamma_2, \bigcirc\Gamma_3$.

Now, we suppose $\Delta \in \{\hat{h}(\Gamma)\}^{\perp}$, that is,

$$\vdash \bigcirc!\Gamma_1, \bigcirc\Box\Gamma_2, \bigcirc\Gamma_3, \Delta \rightarrow \quad .$$

Since

$$\frac{!C \rightarrow !C}{!C \rightarrow \bigcirc!C}(\bigcirc), \quad \frac{\Box C \rightarrow \Box C}{\Box C \rightarrow \bigcirc\Box C}(\bigcirc), \quad \cfrac{\cfrac{C \rightarrow C}{\Box C \rightarrow C}(\Box \rightarrow)}{\Box C \rightarrow \bigcirc C}(\bigcirc),$$

we can deduce $\vdash !\Gamma_1 \rightarrow \bigcirc!\Gamma_1$, $\vdash \Box\Gamma_2 \rightarrow \bigcirc\Box\Gamma_2$, $\vdash \Box\Gamma_3 \rightarrow \bigcirc\Gamma_3$. Using (cut) rules,

$$\cfrac{!\Gamma_1 \rightarrow \bigcirc!\Gamma_1 \quad \cfrac{\Box\Gamma_2 \rightarrow \bigcirc\Box\Gamma_2 \quad \cfrac{\Box\Gamma_3 \rightarrow \bigcirc\Gamma_3 \quad \bigcirc!\Gamma_1, \bigcirc\Box\Gamma_2, \bigcirc\Gamma_3, \Delta \rightarrow}{\bigcirc!\Gamma_1, \bigcirc\Box\Gamma_2, \Box\Gamma_3, \Delta \rightarrow}}{\bigcirc!\Gamma_1, \Box\Gamma_2, \Box\Gamma_3, \Delta \rightarrow}}{!\Gamma_1, \Box\Gamma_2, \Box\Gamma_3, \Delta \rightarrow},$$

that is, $\vdash \hat{f}(\Gamma), \Delta \rightarrow$. This means that $\hat{f}(\Gamma), \Delta \in \hat{\bot} = ||\epsilon||$. We can say that $\hat{f}(\Gamma), \Delta \in \hat{\bot}$ for any $\Delta \in \{\hat{h}(\Gamma)\}^{\perp}$. Therefore, $\hat{f}(\Gamma) \in \{\hat{h}(\Gamma)\}^{\perp\perp}$. (Q.E.D.)

We define

$$\hat{K} := \{!\Gamma | !\Gamma \in \hat{M}\}.$$

Then $\hat{K}$ is a monoid since $\epsilon \in \hat{K}$ and $!\Gamma_1, !\Gamma_2 = !\Gamma_2, !\Gamma_1$ and $(!\Gamma_1, !\Gamma_2), !\Gamma_3 = !\Gamma_1, (!\Gamma_2, !\Gamma_3)$.

Proposition 4.3.4 $\hat{K} \subseteq J(\hat{M})$. ∎

Proof.

1. Let $!\Gamma \in \hat{K}$. Suppose $\Delta \in ||\epsilon||$, that is, $\vdash \Delta \rightarrow$. We can deduce

$$\frac{\Delta \rightarrow}{\overline{!\Gamma, \Delta \rightarrow}}(!w),$$

that is, $!\Gamma, \Delta \in ||\epsilon|| = \hat{\bot}$. This means that $!\Gamma \in ||\epsilon||^{\perp} = \hat{\bot}^{\perp} = \mathbf{1}$. Therefore, $\hat{K} \subseteq \mathbf{1}$.

2. Let $\Delta \in \{(!\Gamma, !\Gamma)\}^{\perp}$, that is, $\vdash !\Gamma, !\Gamma, \Delta \rightarrow$. We can deduce

$$\frac{!\Gamma, !\Gamma, \Delta \rightarrow}{!\Gamma, \Delta \rightarrow} \ (!c)$$

that is, $!\Gamma, \Delta \in ||\epsilon|| = \hat{\perp}$. Therefore, $!\Gamma \in \{(!\Gamma, !\Gamma)\}^{\perp\perp}$. (Q.E.D.)

By the definition of $\hat{K}$ and $\hat{f}$, it is obvious that $\hat{K} = \hat{f}(\hat{K})$. Therefore, we can say that

$$\hat{K} \text{ is a submonoid of } J(\hat{M}) \text{ and } \hat{K} \subseteq \hat{f}(\hat{K}).$$

Note that

$$\begin{aligned} !A &= (A \cap \hat{f}(A) \cap \hat{K})^{\perp\perp} = (A \cap \hat{K})^{\perp\perp}, \\ ?A &= (A^{\perp} \cap \hat{f}(A^{\perp}) \cap \hat{K})^{\perp} = (A^{\perp} \cap \hat{K})^{\perp} \end{aligned}$$

since $\hat{f}(X) \cap \hat{K} = X \cap \hat{K}$ for any X.

Our canonical model is the temporal phase space $\{(M, \perp), h, f\}$, where $M = \hat{M}$, $\perp = \hat{\perp}$, $Cl = {}^{\perp\perp}$, $h = \hat{h}$, $f = \hat{f}$. Finally, we consider the valuation $p^* = ||p||$ for any atomic p. Note that for any formula A, A^* is a fact since $p^* = ||p||$ is a fact.

4.3.2 The Main Lemma and the Completeness Theorem

The completeness theorem is obtained by the Main Lemma. The Main Lemma is obtained by induction on the structure of the formula A. In the induction hypothesis of the proof, Corollary 4.3.1, which follows from the Main Lemma, is used.

Lemma 4.3.1 *For any formula B, if $B \in B^*$ then $!B \in B^*, \Box B \in B^*$.* ■

Proof. Let $\forall \Pi \in B^*(\vdash \Pi \rightarrow \Delta)$ for any Δ. Take $B \in B^*$ as Π, and we obtain $\vdash !B \rightarrow \Delta$ since

$$\frac{B \rightarrow \Delta}{!B \rightarrow \Delta} \ (! \rightarrow)$$

that is, if $\forall \Pi \in B^*(\vdash \Pi \rightarrow \Delta)$ then $\vdash !B \rightarrow \Delta$. This means that $!B \in B^{*\perp\perp} = B^*$.

Similarly, we also obtain that $\Box B \in B^*$. (Q.E.D.)

The completeness theorem follows from the Main Lemma.

Lemma 4.3.2 (Main Lemma) *For any formula A,*

$$A^* \subseteq ||A||.$$

■

The proof of the Main Lemma is in Appendix C.3.

Corollary 4.3.1 *For any formula A, $A \in A^*$.* ■

Proof. By induction on the structure of the formula A. See Appendix C.2 for details.

Lemma 4.3.3

$$||\epsilon|| = || \perp ||.$$

■

Proof. We show that $\vdash \Gamma \rightarrow$ iff $\vdash \Gamma \rightarrow \bot$. It is easy to show that if $\vdash \Gamma \rightarrow$ then $\vdash \Gamma \rightarrow \bot$ since

$$\frac{\Gamma \rightarrow}{\Gamma \rightarrow \bot} (\rightarrow \bot).$$

We show that if $\vdash \Gamma \rightarrow \bot$ then $\vdash \Gamma \rightarrow$ by induction on the length of the proof of $\vdash \Gamma \rightarrow \bot$. But it is standard. Below are several examples.

- When the proof is of the form $\Gamma', \mathbf{0} \rightarrow \bot$,

$$\frac{}{\Gamma', \mathbf{0} \rightarrow} (\mathbf{0} \rightarrow).$$

- When the last rule of the proof is $(\oplus \rightarrow)$, that is,

$$\frac{A, \Gamma' \rightarrow \bot \quad B, \Gamma' \rightarrow \bot}{A \oplus B, \Gamma' \rightarrow \bot} (\oplus \rightarrow).$$

  By the induction hypothesis, $\vdash A, \Gamma' \rightarrow$ and $\vdash B, \Gamma' \rightarrow$. Then,

$$\frac{A, \Gamma' \rightarrow \quad B, \Gamma' \rightarrow}{A \oplus B, \Gamma' \rightarrow} (\oplus \rightarrow).$$

Other cases are similar. (Q.E.D.)

Lemma 4.3.4 *For any formula A, if $A^* \subseteq ||A||$ then $A^\perp \in A^{*\perp}$.* ■

Proof. Let $\Delta \in A^*$, then we obtain $\vdash \Delta \rightarrow A$ by the assumption $A^* \subseteq ||A||$. We can deduce

$$\frac{\Delta \rightarrow A}{\Delta, A^\perp \rightarrow} ({}^\perp\!\rightarrow).$$

We can say that if $\Delta \in A^*$ then $\Delta, A^\perp \in \hat{\bot} = ||\epsilon||$. Therefore, $A^\perp \in A^*$. (Q.E.D.)

Theorem 4.3.1 (Completeness) *If $\Gamma \rightarrow \Delta$ is valid, then it is provable in* **TLL**. ■

Proof. Assume that $\Gamma \rightarrow \Delta$ is valid, and we have $\Gamma^{*\otimes} \subseteq \Delta^{*\wp}$ for any model, in particular for our canonical model.

By Main Lemma 4.3.2, and by Corollary 4.3.1, $\Gamma^\otimes \in \Gamma^{*\otimes}$. Also, by Main Lemma 4.3.2, $\Delta^{*\wp} \subseteq ||\Delta^\wp||$. Hence,

$$\Gamma^\otimes \in ||\Delta^\wp||,$$

that is, $\vdash \Gamma^\otimes \rightarrow \Delta^\wp$. From this, one can deduce

$$\frac{\Gamma^\otimes \rightarrow \Delta^\wp}{\Gamma \rightarrow \Delta},$$

that is, $\vdash \Gamma \rightarrow \Delta$. (Q.E.D.)

We also obtain the **ITLL** version of the completeness theorem, that is,

Corollary 4.3.2 (Completeness) *If $\Gamma \rightarrow D$ is valid, then it is provable in* **ITLL**. ■

In order to prove Corollary 4.3.2, we should change the closure operator in our canonical model. We consider $\mathbf{C}$ instead of ${}^{\perp\perp}$ as follows. For subset $X \subseteq \hat{M}$, we define

$$\mathbf{C}(X) := \{\Gamma \in \hat{M} | \text{ If } \forall \Delta \in X (\vdash_{\mathbf{ITLL}} \Delta \rightarrow D) \text{ then } \vdash_{\mathbf{ITLL}} \Gamma \rightarrow D, \text{ for any formula } D\}.$$

This is a special case of (4.3.1). Indeed, $\mathbf{C}(X)$ is a closure:

(1) $X \subseteq \mathbf{C}(X)$.

Let $\Pi \in X$. Suppose $\forall \Delta \in X(\vdash \Delta \rightarrow D)$. Take Π as Δ then $\vdash \Pi \rightarrow D$, that is, $\Pi \in \mathbf{C}(X)$.

(2) $\mathbf{C}(\mathbf{C}(X)) \subseteq \mathbf{C}(X)$.

Suppose $\forall \Delta \in X(\vdash \Delta \rightarrow D)$. We will show that if $\Pi \in \mathbf{C}(\mathbf{C}(X))$ then $\vdash \Pi \rightarrow D$.

At first, we show that $\forall \Delta' \in \mathbf{C}(X)(\vdash \Delta' \rightarrow D)$. Let $\Delta_0 \in \mathbf{C}(X)$. By the definition of $\mathbf{C}(X)$, we obtain that $\vdash \Delta_0 \rightarrow D$ using the assumption $\forall \Delta \in X(\vdash \Delta \rightarrow D)$. This concludes that if $\Delta_0 \in \mathbf{C}(X)$ then $\vdash \Delta_0 \rightarrow D$, that is,

$$\forall \Delta' \in \mathbf{C}(X)(\vdash \Delta' \rightarrow D). \tag{4.3.2}$$

Now, we suppose

$$\Pi \in \mathbf{C}(\mathbf{C}(X)) = \{\Gamma \in M | \text{ If } \forall \Delta' \in \mathbf{C}(X)(\vdash \Delta' \rightarrow D') \text{ then } \vdash \Gamma \rightarrow D'\}.$$

Using (4.3.2), we obtain

$$\vdash \Pi \rightarrow D.$$

(3) If $Z \subseteq X$ then $\mathbf{C}(Z) \subseteq \mathbf{C}(X)$.

Suppose $\forall \Delta \in X(\vdash \Delta \rightarrow D)$. We will show that if $\Pi \in \mathbf{C}(Z)$ then $\vdash \Pi \rightarrow D$.

For a $\Delta_0 \in Z$, one can say that $\Delta_0 \in X$ since $Z \subseteq X$. By the assumption $\forall \Delta \in X(\vdash \Delta \rightarrow D)$, we obtain $\vdash \Delta_0 \rightarrow D$, that is

$$\forall \Delta' \in Z(\vdash \Delta' \rightarrow D). \tag{4.3.3}$$

Now, we suppose

$$\Pi \in \mathbf{C}(Z) = \{\Gamma \in M | \text{ If } \forall \Delta' \in Z(\vdash \Delta' \rightarrow D] \text{ then } \vdash \Gamma \rightarrow D)\}.$$

Using (4.3.3), we obtain

$$\vdash \Pi \rightarrow D.$$

(4) $\mathbf{C}(Z) \cdot \mathbf{C}(X) \subseteq \mathbf{C}(Z \cdot X)$.

Suppose $\forall \Delta \in Z \cdot X(\vdash \Delta \rightarrow D)$. We will show that if $\Pi \in \mathbf{C}(Z) \cdot \mathbf{C}(X)$ then $\vdash \Pi \rightarrow D$.

Let $\Delta_1 \in Z, \Delta_2 \in X$ then $\Delta_1, \Delta_2 \in Z \cdot X$. By the assumption $\forall \Delta \in Z \cdot X(\vdash \Delta \rightarrow D)$,

$$\vdash \Delta_1, \Delta_2 \rightarrow D.$$

One can deduce

$$\Delta_1 \rightarrow \Delta_2^{\otimes} \multimap D.$$

Hence, one can say that

$$\forall \Delta_1' \in Z(\vdash \Delta_1' \rightarrow \Delta_2^{\otimes} \multimap D). \tag{4.3.4}$$

Now, suppose $\Pi \in \mathbf{C}(Z) \cdot \mathbf{C}(X)$, that is, $\Pi_1 \in \mathbf{C}(Z)$ and $\Pi_2 \in \mathbf{C}(X)$ for some Π_1, Π_2 such that $\Pi = \Pi_1, \Pi_2$.

From $\Pi_1 \in \mathbf{C}(Z)$, if $\forall \Delta'_1 \in Z(\vdash \Delta'_1 \to D_1)$ then $\vdash \Pi_1 \to D_1$ for any D_1. Put $\Delta_2^{\otimes} \multimap D$ as D_1 and we obtain that $\vdash \Pi_1 \to \Delta_2^{\otimes} \multimap D$ by (4.3.4). One can deduce

$$\dfrac{\dfrac{\dfrac{\Pi_1 \to \Delta_2^{\otimes} \multimap D \quad \dfrac{\Delta_2^{\otimes} \to \Delta_2^{\otimes} \quad D \to D}{\Delta_2^{\otimes} \multimap D, \Delta_2^{\otimes} \to D}}{\Pi_1, \Delta_2^{\otimes} \to D}(cut)}{\Pi_1, \Delta_2 \to D}}{\Delta_2 \to \Pi_1^{\otimes} \multimap D} .$$

Hence, one can say that

$$\forall \Delta'_2 \in X(\vdash \Delta'_2 \to \Pi_1^{\otimes} \multimap D). \tag{4.3.5}$$

Similarly from $\Pi_2 \in \mathbf{C}(X)$, we obtain that $\vdash \Pi_2 \to \Pi_1^{\otimes} \multimap D$ by (4.3.5). Therefore, one can deduce

$$\Pi_1, \Pi_2 \to D,$$

that is, $\Pi \to D$.

Lemma 4.3.1, Main Lemma 4.3.2 and Corollary 4.3.1 are also satisfied in the **ITLL** version. They are obtained by checking rules and connectives concerning **ITLL**.

Chapter 5

Some Applications to Computer Science

Temporal linear logic has been introduced by extending linear logic with respect to the time concept. In [32], as an application of **TLL**, we have shown that a logic programming language based on temporal linear logic has been designed by using the idea of Miller's uniform proof, and its efficient computation model has been given by using the idea of Hodas's IO model.

In this chapter, we continue to apply **TLL** to computer science. In the first section, we provide a correspondence between the concept of a parallel calculation and the logical concept roughly, and consider a communication model, which is our own model. We show that **TLL** can represent not only an asynchronous calculus but also a synchronous calculus [10].

The relation to timed Petri nets follows that. We show that the reachability problem for timed Petri nets is equivalent to the provability of the corresponding sequent of **TLL** [9, 10]. We also show that the reachability problem for timed Petri nets is decidable by a method different from [29, 30].

5.1 Synchronous Communication and Temporal Linear Logic

In linear logic, since $m \multimap (n \multimap P)$ is equivalent to $m \otimes n \multimap P$, the following cannot be distinguished:

- "A process which receives m, then receives n, and behaves like P",
- "A process which receives n, then receives m, and behaves like P",
- "A process which receives m and n simultaneously, and behaves like P".

It follows that we cannot specify the execution order of processes. Also, we cannot distinguish a synchronous calculus from an asynchronous calculus in linear logic.

Using "$\bigcirc$" and "$\Box$", in temporal linear logic, we can specify the order such as $m \multimap \bigcirc\Box(n \multimap \bigcirc\Box P)$. Furthermore, we can distinguish a synchronous calculus from an asynchronous calculus.

We compare descriptions of a parallel calculation by linear logic and by temporal linear logic. At first, we consider several descriptions by linear logic in the same manner as in Okada [27]. From here, we omit some inference rules applied in proof figures.

We consider the following correspondence:

- $m \otimes Q$;
  Send a message m, and then Q executes.
- $m \multimap Q$;
  Receive a message m, and then Q executes while consuming m.

- $!Q$;
  Copy any number of process Q, and then each of Q executes.

Also, we can express the transitions of states by means of reading the proof figure upwards from below as follows:

- sending;
  $$\frac{\Gamma, m, Q, \Delta \rightarrow}{\Gamma, m \otimes Q, \Delta \rightarrow}$$
  $m \otimes Q$ sends m and then executes process Q concurrently.

- receiving;
  $$\frac{(m \rightarrow m) \quad \Gamma, Q, \Delta \rightarrow}{\Gamma, m \multimap Q, m, \Delta \rightarrow}$$
  $m \multimap Q$ receives m and then executes process Q while consuming m. Here, Γ and Δ mean sequences of parallel processes which are executed concurrently.

- Milner's bang;
  $$\frac{\Gamma, Q, !Q, \Delta \rightarrow}{\Gamma, !Q, \Delta \rightarrow}$$
  $!Q$ generates a copy of Q and then executes it.

As an example, let us consider a process "sending message m repeatedly". This process can be represented by $!(p \multimap (m \otimes p))$. A process "receiving message m repeatedly" can be represented by $!(q \multimap (m \multimap q))$. Here, let us consider a situation including the parallel execution of these processes expressed by the sequent $p, q, !\Gamma \rightarrow$, where $!\Gamma = !(p \multimap (m \otimes p)), !(q \multimap (m \multimap q))$. See the following proof figure, which is read upwards from below:

$$\frac{(p \rightarrow p \quad q \rightarrow q) \qquad \dfrac{\dfrac{\dfrac{(p \rightarrow p) \quad \dfrac{\dfrac{\vdots}{m, m, p, m \multimap q, !\Gamma \rightarrow}}{m, m \otimes p, m \multimap q, !\Gamma \rightarrow}(\sharp 6)}{m, p, p \multimap (m \otimes p), m \multimap q, !\Gamma \rightarrow}(\sharp 5)}{m, p, m \multimap q, !\Gamma \rightarrow}(\sharp 4)}{m \otimes p, m \multimap q, !\Gamma \rightarrow}(\sharp 3)}{\dfrac{p, q, p \multimap (m \otimes p), q \multimap (m \multimap q), !\Gamma \rightarrow}{p, q, !\Gamma \rightarrow}(\sharp 1)}(\sharp 2)$$

($\sharp$1)–($\sharp$2) express the transition that $!(p \multimap (m \otimes p))$ copies $p \multimap (m \otimes p)$ for the behavior of sending a message m and $!(q \multimap (m \multimap q))$ copies $q \multimap (m \multimap q)$ for the behavior of receiving m. ($\sharp$3) expresses the sending of m. ($\sharp$4)–($\sharp$6) means that process $!(p \multimap (m \otimes p))$ can send the next message without waiting for m to be received by $m \multimap q$, that is, asynchronous behavior.

In temporal linear logic we can also represent synchronous behavior, while linear logic cannot. Processes stated above are expressed by $!(p \multimap (m \otimes \bigcirc \Box p))$ and $!(q \multimap (m \multimap \bigcirc \Box q))$ respectively. $m \otimes \bigcirc \Box p$ sends a message m synchronously, that is, until m is received, it is suspended. After m is received, $\Box p$ and $\Box q$ will be active the next time. The synchronous communication is represented as follows:

$$\frac{(p \rightarrow p \quad q \rightarrow q) \qquad \dfrac{\dfrac{(m \rightarrow m) \quad \dfrac{\dfrac{\vdots}{p, q, !\Gamma \rightarrow}}{\bigcirc \Box p, \bigcirc \Box q, !\Gamma \rightarrow}}{m, \bigcirc \Box p, m \multimap \bigcirc \Box q, !\Gamma \rightarrow}}{m \otimes \bigcirc \Box p, m \multimap \bigcirc \Box q, !\Gamma \rightarrow}}{\dfrac{p, q, p \multimap (m \otimes \bigcirc \Box p), q \multimap (m \multimap \bigcirc \Box q), !\Gamma \rightarrow}{p, q, !\Gamma \rightarrow}} .$$

Process $!(p \multimap (m \otimes \bigcirc \Box p))$ cannot send the next message in this case. As we considered above, temporal linear logic can distinguish a synchronous calculus from an asynchronous calculus.

5.2 Timed Petri Nets and Temporal Linear Logic

In this section, we consider timed Petri nets [3, 12] and the reachability problem as an application of **ITLL**.

5.2.1 Timed Petri Nets

We choose place timed Petri nets in this thesis.

Definition 5.2.1 (Timed Petri Net) A *(place) Timed Petri Net (TPN)* is a tuple (Pl, Tr, Ar, θ), where

Pl: Finite set of places

Tr: Finite set of transitions (disjoint with Pl)

Ar: $(Pl \times Tr) \cup (Tr \times Pl) \longrightarrow \mathbf{N}$ (Weight of arcs)

θ: $Pl \longrightarrow \mathbf{N}$

Here, $\mathbf{N}$ means the set of natural numbers (including 0). $\theta(p) \geq 0$ indicates the waiting time till the tokens which are usable in future become available in $p \in Pl$ ∎

A multiset of places (i.e. marking) is not sufficient to represent a state of TPN. We need not only the information of available tokens (i.e. *active tokens*), but also tokens to be usable in future (i.e. *pending tokens*). Thus, we consider a "state" of TPN, which contains "marking" with "time".

Definition 5.2.2 (State) A *state* of TPN is an infinite sequence of multisets of places $\langle M_0, M_1, \ldots \rangle$ where $M_m = M_{m+1} = \ldots = \emptyset$ for some $m \geq 0$. ∎

In a state S at some instant, M_0, which is called a *timed marking*, indicates active tokens and M_i $(i \geq 1)$ indicates pending tokens which will be active after i time units.

We will define reachability with respect to states. A reached state is derived by *firing derivation* or *time derivation*.

Definition 5.2.3 (Derivation) Let $S = \langle M_0, M_1, \ldots \rangle$ be a state at some instant t.

firing derivation : We say that a transition τ *is enabled* at S if and only if $M_0^- \subseteq M_0$. Here, M_0^- is a multiset of *input places* to τ. If a transition τ is enabled and we fire it at that instant, the reached state at the same instant t is the state S' defined by

$$S' = \langle M_0 - M_0^- \uplus M_0^+, M_1 \uplus M_1^+, M_2 \uplus M_2^+, \ldots \rangle.$$

Here, M_i^+ indicates a multiset of *output places* p's from τ with $\theta(p) = i \geq 0$. $\uplus$ indicates a multiset union. Note that a firing terminates at the instant. The described derivation is denoted by the notation $S[\tau\rangle S'$.

time derivation : The reached state at the instant $t+1$ from S is the state S' defined by

$$S' = \langle M_0 \uplus M_1, M_2, \ldots \rangle.$$

The described derivation is denoted by the notation $S[\delta\rangle S'$. ∎

We consider TPN in Fig.5.1. The numbers beside each place p_i indicates $\theta(p_i)$. The state $S = \langle\{p_1, p_2, p_2\}, \emptyset, \ldots\rangle$. The transition τ_1 is enabled at S. We fire it at $t = 0$, then the reached state $S_1 = \langle\{p_1\}, \emptyset, \{p_3, p_3\}, \emptyset, \ldots\rangle$ at $t = 0$. After 2 time units, the reached state $S_2 = \langle\{p_1, p_3, p_3\}, \emptyset, \ldots\rangle$ at $t = 2$. The transition τ_2 is enabled at S_2 and we fire it at $t = 2$, then the reached state $S' = \langle\{p_1, p_3\}, \{p_2\}, \emptyset, \ldots\rangle$ at $t = 2$.

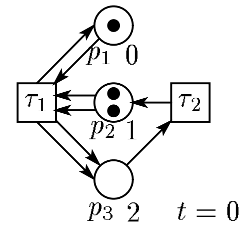

Fig.5.1: Timed Petri net

Now, we define the reachability for TPN with respect to states. For a derivation sequence $\sigma = \kappa_1 \ldots \kappa_n$ $(n \geq 0)$, we use the notation $S[\sigma\rangle S'$ instead of $S[\kappa_1\rangle S_1[\kappa_2\rangle S_2 \ldots S_{n-1}[\kappa_n\rangle S'$, where κ_i is either $\tau \in Tr$ or δ. Specially, if the number of δ in σ is t, we use the notation $S \overset{t}{[\sigma\rangle} S'$, which means that S' will be reached from S *after t time units*. For example, $S \overset{2}{[\tau_1\delta\delta\tau_2\rangle} S'$ for the TPN in Fig.5.1.

Definition 5.2.4 (Reachable) Let S and S' be states of a TPN. We say that S' is *reachable* from S, which will be denoted by $S' \in [S\rangle$, iff there exists a derivation sequence σ such that $S[\sigma\rangle S'$. ∎

Specially, we say that S' is *strictly reachable* from S at the instant t, which will be denoted by $S' \in [S\rangle_t$, iff there exists a derivation sequence σ such that $S \overset{t}{[\sigma\rangle} S'$.

5.2.2 Reachability and Provability

We can encode the reachability problem for TPN into the provability problem of the corresponding Horn sequent of a *Horn-like* system **HTPN** completely. **HTPN** is extended without destroying the equivalence to the reachability problem for TPN in order to associate with temporal linear logic. At the end of this section, we obtain Theorem 5.2.1, which claims that the reachability problem for TPN is equivalent to the provability problem for the Horn fragment of the subsystem of temporal linear logic.

At first, we define **HTLL** which include **HTPN** as a subsystem of it. We start from a constructive definition. For atomics $p, q, \ldots$, a *token formula* and a *simple product* are defined by

$$\alpha ::= \Box p \mid \bigcirc\alpha, \qquad M ::= \alpha \mid M \otimes M,$$

respectively. A token in $p \in Pl$ can be represented by a token formula, a state can be represented by a simple product. Let us consider the encoding for TPN in Fig.5.1 (See subsection 5.2.1). For a state S, we denote the corresponding simple product by S^*. In Fig.5.1, $S^* = \Box p_1 \otimes \Box p_2 \otimes \Box p_2$. The encoding of a transition τ is denoted by τ^*. $\tau_1^* = \Box p_1 \otimes \Box p_2 \otimes \Box p_2 \multimap \Box p_1 \otimes \bigcirc^2\Box p_3 \otimes \bigcirc^2\Box p_3$, $\tau_2^* = \Box p_3 \multimap \bigcirc\Box p_2$. In this thesis, simple products are denoted by $X, Y, Z, M, \ldots$. For $t \geq 0$, a *Horn sequent* is a sequent of the form

$$\Gamma; \Delta, \mathbf{1} \otimes M \to \bigcirc^t Z,$$

where Γ is a *set* of formulas of the form $X \multimap Y$ and Δ is a *multiset* of formulas of the form $X \multimap Y$. M will associate with the initial state, Z the goal state, Γ the whole transitions in TPN and Δ the used transitions for the derivation sequence.

By a Horn sequent, we can express the statement with respect to the reachability. For the TPN in Fig.5.1, the statement "S' is reachable from S after 2 time units" is represented by the following Horn sequent

$$\tau_1^*, \tau_2^*; \mathbf{1} \otimes \Box p_1 \otimes \Box p_2 \otimes \Box p_2 \to \bigcirc^2(\mathbf{1} \otimes \Box p_1 \otimes \Box p_3 \otimes \bigcirc\Box p_2). \tag{5.2.1}$$

$\mathbf{1}$ in a Horn sequent is a trick to be able to construct the corresponding state from a formula of the form $\bigcirc^n\Box p$. For example, although we can construct the state $\langle\{p_2\}, \emptyset, \ldots\rangle$ from $\bigcirc^3(\mathbf{1} \otimes \Box p_2)$, we cannot decide the corresponding state from the form $\bigcirc^n\Box p$ on the right side of the Horn sequent.

$$\frac{}{\Gamma; X \to X}\ (Ax_1)$$

$$\frac{\Gamma; \Delta, Y \otimes M \to \bigcirc^t Z}{\Gamma; \Delta, X \otimes M \to \bigcirc^t Z}\ (fire)$$
provided that $X \multimap Y \in \Gamma$.

$$\frac{\Gamma; \mathbf{1} \otimes M^{\Box} \otimes \alpha_1 \otimes \ldots \otimes \alpha_k \to \bigcirc^t Z}{\Gamma; \mathbf{1} \otimes M^{\Box} \otimes \bigcirc\alpha_1 \otimes \ldots \otimes \bigcirc\alpha_k \to \bigcirc^{t+1} Z}\ (next)$$
where $M^{\Box}$ is of the form $\Box p_1 \otimes \ldots \otimes \Box p_m$, each α_i is a token formula.

$$\frac{}{\Gamma; X \multimap Y, X \to Y}\ (Ax_2) \qquad \frac{}{\Gamma; \to \mathbf{1}}\ (\mathbf{1}) \qquad \frac{\Gamma; \Delta, X \to X}{\Gamma; \Delta, X \otimes Y \to X \otimes Y}\ (\otimes)$$

$$\frac{\Gamma; A, \Delta, M \to \bigcirc^t Z}{\Gamma; \Delta, M \to \bigcirc^t Z}\ (absorb)$$
provided that $A \in \Gamma$.

$$\frac{\Gamma'; \Delta_1, M \to X \quad \Gamma; \Delta_2, X \to \bigcirc^t Z}{\Gamma'; \Delta_1, \Delta_2, M \to \bigcirc^t Z}\ (Hcut)$$
where $\Gamma \subseteq \Gamma'$.

Table 5.1: Horn temporal linear logic

Now, we define **HTLL** as follows:

Definition 5.2.5 (HTLL) Let formulas be of the form $X, X \multimap Y, \bigcirc^n Z$, sequents be the form of Horn sequents. We define **HTLL** as a system constructed from Table 5.1. ∎

We call the subsystem which is constructed by (Ax_1), $(fire)$ and $(next)$ only as **HTPN**. It is not difficult to show the following lemma.

Lemma 5.2.1 *Let (Pl, Tr, Ar, θ) be a timed Petri net and S, S' states of it. Then $S\ [\overset{t}{\sigma}\rangle\ S'$ for some derivation sequence σ if and only if the following sequent*

$$Tr^*; \mathbf{1} \otimes S^* \to \bigcirc^t(\mathbf{1} \otimes S'^*)$$

is provable in **HTPN**, *where Tr^* is a sequence of τ^* such that $\tau \in Tr$.* ∎

Lemma 5.2.1 claims that we can encode the reachability problem for TPN into the provability problem of the corresponding Horn sequent of **HTPN** completely. For example, since $S\ [\overset{2}{\tau_1 \delta\delta \tau_2}\rangle\ S'$ in Fig.5.1, the Horn sequent (5.2.1) is provable in **HTPN** by Lemma 5.2.1. In fact, the following is the proof figure:

$$\dfrac{\dfrac{\dfrac{\dfrac{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \Box p_1 \otimes \Box p_3 \otimes \bigcirc\Box p_2 \to \mathbf{1} \otimes \Box p_1 \otimes \Box p_3 \otimes \bigcirc\Box p_2}{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \Box p_1 \otimes \Box p_3 \otimes \Box p_3 \to \mathbf{1} \otimes \Box p_1 \otimes \Box p_3 \otimes \bigcirc\Box p_2}\ (fire)}{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \Box p_1 \otimes \bigcirc\Box p_3 \otimes \bigcirc\Box p_3 \to \bigcirc(\mathbf{1} \otimes \Box p_1 \otimes \Box p_3 \otimes \bigcirc\Box p_2)}\ (next)}{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \Box p_1 \otimes \bigcirc^2\Box p_3 \otimes \bigcirc^2\Box p_3 \to \bigcirc^2(\mathbf{1} \otimes \Box p_1 \otimes \Box p_3 \otimes \bigcirc\Box p_2)}\ (next)}{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \Box p_1 \otimes \Box p_2 \otimes \Box p_2 \to \bigcirc^2(\mathbf{1} \otimes \Box p_1 \otimes \Box p_3 \otimes \bigcirc\Box p_2)}\ (fire)$$

Let us consider another Horn sequent with respect to Fig.5.1,

$$\tau_1^*, \tau_2^*; \mathbf{1} \otimes \bigcirc^2\Box p_3 \to \bigcirc^3(\mathbf{1} \otimes \Box p_2). \tag{5.2.2}$$

This is provable in **HTPN**:

$$\dfrac{\dfrac{\dfrac{\dfrac{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \Box p_2 \to \mathbf{1} \otimes \Box p_2}{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \bigcirc\Box p_2 \to \bigcirc(\mathbf{1} \otimes \Box p_2)}\ (next)}{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \Box p_3 \to \bigcirc(\mathbf{1} \otimes \Box p_2)}\ (fire)}{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \bigcirc\Box p_3 \to \bigcirc^2(\mathbf{1} \otimes \Box p_2)}\ (next)}{\tau_1^*, \tau_2^*; \mathbf{1} \otimes \bigcirc^2\Box p_3 \to \bigcirc^3(\mathbf{1} \otimes \Box p_2)}\ (next)$$

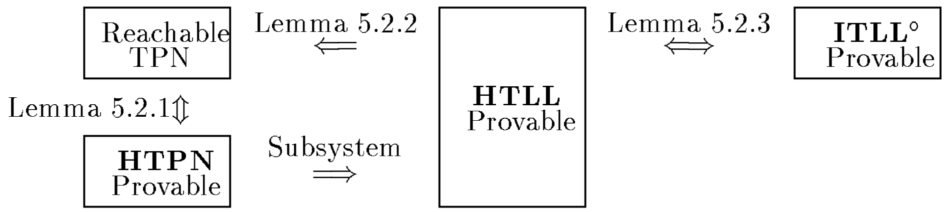

Fig.5.2: Illustration of the proof of Theorem 5.2.1

Suppose $S_1 = \langle \emptyset, \emptyset, \{p_3\}, \emptyset, \ldots \rangle$ and $S_2 = \langle \{p_2\}, \emptyset, \ldots \rangle$. By Lemma 5.2.1, $S_1 \overset{3}{[\sigma\rangle} S_2$ for some σ. Furthermore, we can construct $\sigma = \delta\delta\tau_2\delta$ from the proof figure.

Lemma 5.2.1 can be extended to the following lemma:

Lemma 5.2.2 *Let (Pl, Tr, Ar, θ) be TPN and S, S' states of it. Suppose Δ^* is a multiset structured from $\tau^* \in Tr^*$. If the Horn sequent*

$$Tr^*; \Delta^*, \mathbf{1} \otimes S^* \to \bigcirc^t(\mathbf{1} \otimes S'^*)$$

is provable in **HTLL** *then there is σ such that $S \overset{t}{[\sigma\rangle} S'$ and any $\tau \in \Delta$ has been really used in σ (i.e. For any $\tau \in \Delta$, $\tau \in \sigma$).* ∎

Proof. (sketch) The claim is shown by induction on the length of the proof of the Horn sequent. See [16]. (Q.E.D.)

Let **ITLL°** be a subsystem of **ITLL** by replacing $(\to \otimes)$ with $(\to \otimes)^\circ$ and provided that all atomics are of the form $\Box p$, where

$$\frac{\Gamma, A \to A \quad \Delta, B \to B}{\Gamma, \Delta, A, B \to A \otimes B} \ (\to \otimes)^\circ$$

We can associate **HTLL** with **ITLL°** by the following lemma.

Lemma 5.2.3 *Let $\Gamma; \Delta, M \to \bigcirc^t Z$ be a Horn sequent. Then $\Gamma; \Delta, M \to \bigcirc^t Z$ is provable in* **HTLL** *if and only if $!\Gamma, \Delta, M \to \bigcirc^t Z$ is provable in* **ITLL°**. ∎

Proof. It is not difficult to show that if $\Gamma; \Delta, M \to \bigcirc^t Z$ is provable in **HTLL** then $!\Gamma, \Delta, M \to \bigcirc^t Z$ is provable in **ITLL°**.

We sketch the proof of converse. Suppose $!\Gamma, \Delta, M \to \bigcirc^t Z$ is provable in **ITLL°** and $M = \alpha_1 \otimes \ldots \otimes \alpha_n$, where each α_i indicates a token formula. Then there exists some cut free proof of $!\Gamma, \Delta, \alpha_1, \ldots, \alpha_n \to \bigcirc^t Z$. One can prove the claim by induction on the length of the proof figure. (Q.E.D.)

Now, we obtain the completeness theorem for the reachability problem for timed Peri nets.

Theorem 5.2.1 (Completeness theorem) *Let (Pl, Tr, Ar, θ) be a timed Petri net and S, S' states of it. Then S' is reachable from S after t time units if and only if the sequent*

$$!Tr^*, \mathbf{1} \otimes S^* \to \bigcirc^t(\mathbf{1} \otimes S'^*) \quad (5.2.3)$$

is provable in **ITLL°**. ∎

Proof. (See Fig.5.2)

(Soundness) Suppose $!Tr^*, \mathbf{1} \otimes S^* \to \bigcirc^t(\mathbf{1} \otimes S'^*)$ is provable in **ITLL°**. By Lemma 5.2.3, $Tr^*; \mathbf{1} \otimes S^* \to \bigcirc^t(\mathbf{1} \otimes S'^*)$ is provable in **HTLL**. Then $S \overset{t}{[\sigma\rangle} S'$ for some derivation sequence σ by Lemma 5.2.2.

(Completeness) Suppose $S \overset{t}{[\sigma\rangle} S'$ for some σ. $Tr^*; \mathbf{1} \otimes S^* \to \bigcirc^t(\mathbf{1} \otimes S'^*)$ is provable in **HTPN** by Lemma 5.2.1. Therefore, it is provable in **HTLL**. By Lemma 5.2.3, $!Tr^*, \mathbf{1} \otimes S^* \to \bigcirc^t(\mathbf{1} \otimes S'^*)$ is provable in **ITLL°**. (Q.E.D.)

Unlike theorem 2.2.1, we have to restrict the tensor rule for the equivalence between the reachability of TPN and the provability of the corresponding sequent of temporal linear logic. This concludes that the following does not satisfy generally: if there exists some σ_1 such that $S_0[\sigma_1\rangle S$ and σ_2 such that $S_0'[\sigma_2\rangle S'$, then there exists σ such that $S_0 \uplus S_0'[\sigma\rangle S \uplus S'$. One can deduce $\Gamma_1, \Gamma_2, \bigcirc A, \bigcirc B \to \bigcirc(A \otimes B)$ from $\Gamma_1, \bigcirc A \to \bigcirc A$ and $\Gamma_2, \bigcirc B \to \bigcirc B$ in **ITLL°**. This concludes that if $!Tr^*, \mathbf{1} \otimes S_0^* \to \bigcirc^t(\mathbf{1} \otimes S^*)$ and $!Tr^*, \mathbf{1} \otimes S_0'^* \to \bigcirc^t(\mathbf{1} \otimes S'^*)$ are provable in **ITLL°** then $!Tr^*, \mathbf{1} \otimes S_0^* \otimes S_0'^* \to \bigcirc^t(\mathbf{1} \otimes S^* \otimes S'^*)$ is also provable in **ITLL°**. We can say that if we match between the passages of time then we can combine two derivation sequences.

5.3 Decidability of the Reachability Problem for Timed Petri Nets

By the previous section, the strict reachability for timed Petri nets is equivalent to the provability of the corresponding Horn sequent. In this section, we show the decidability of the strict reachability problem for timed Petri nets by rewriting a Horn sequent (Corollary 5.3.1).

Let $\mathcal{S}$ be the following Horn sequent of **HTPN**

$$\Gamma; \mathbf{1} \otimes M \to \bigcirc^t(\mathbf{1} \otimes Z),$$

where Γ is a set of formulas of the form $X \multimap Y$. We rewrite $\mathcal{S}$ to obtain the rewritten Horn sequent $\tilde{\mathcal{S}}$

$$\tilde{\Gamma}; clock^{(0)} \otimes \tilde{M} \to clock^{(t)} \otimes \tilde{Z},$$

which does not include temporal modalities. The rewriting steps are as follows:

Rewriting steps for a Horn sequent

- Rewriting for M and Z.
  1. Each $\mathbf{1}$ on both sides is removed. We put an atomic $clock^{(0)}$ in front of M and an atomic $clock^{(t)}$ instead of $\bigcirc^t$.
  2. Each token fomula of the form $\bigcirc^k \Box p$ in M and Z is rewritten into an atomic $p^{(k)}$ and $p^{(t+k)}$, respectively.
- Rewriting for Γ. The rewriting corresponds to the translation from TPN structure into PN structure.
  1. Each $X \multimap Y$ is rewritten into a series of linear implications of the forms

     $$clock^{(i)} \otimes X \multimap clock^{(i)} \otimes Y,$$

     where $0 \le i \le t$. Each token fomula of the form $\bigcirc^k \Box p$ in X and Y is rewritten into an atomic $p^{(i+k)}$.
  2. We add a series of auxiliary linear implications to Γ.
     (a) We add a series of auxiliary linear implications of the forms

     $$clock^{(j)} \multimap tmp^{(j)}, tmp^{(j)} \multimap clock^{(j+1)},$$

     where $0 \le j \le t-1$.

(b) For each p in X and Y, we add a series of auxiliary linear implications of the form

$$tmp^{(j)} \otimes p^{(j)} \multimap tmp^{(j)} \otimes p^{(j+1)}$$

where $0 \leq j \leq t-1$.

For example, we consider a Horn sequent (5.2.1) in subsection 5.2.2 as $\mathcal{S}$. We can rewrite it into the following Horn sequent as $\tilde{\mathcal{S}}$

$$\tilde{Tr}; clock^{(0)} \otimes p_1^{(0)} \otimes p_2^{(0)} \otimes p_2^{(0)} \rightarrow clock^{(2)} \otimes p_1^{(2)} \otimes p_3^{(2)} \otimes p_2^{(3)} \tag{5.3.1}$$

where $\tilde{Tr}$ is the following sequence of linear implications:

$$\begin{array}{l} clock^{(i)} \otimes p_1^{(i)} \otimes p_2^{(i)} \otimes p_2^{(i)} \multimap clock^{(i)} \otimes p_1^{(i)} \otimes p_3^{(i+2)} \otimes p_3^{(i+2)}, \\ clock^{(i)} \otimes p_3^{(i)} \multimap clock^{(i)} \otimes p_2^{(i+1)}, \\ \quad clock^{(j)} \multimap tmp^{(j)}, \quad tmp^{(j)} \multimap clock^{(j+1)}, \\ tmp^{(j)} \otimes p_1^{(j)} \multimap tmp^{(j)} \otimes p_1^{(j+1)}, \\ tmp^{(j)} \otimes p_2^{(j)} \multimap tmp^{(j)} \otimes p_2^{(j+1)}, \\ tmp^{(j)} \otimes p_3^{(j)} \multimap tmp^{(j)} \otimes p_3^{(j+1)} \qquad (0 \leq i \leq 3,\ 0 \leq j \leq 2). \end{array}$$

We can obtain the following lemma by replacing $(next)$ rules in the proof figure of $\mathcal{S}$ into $(fire)$ rules with respect to auxiliary linear implications in $\tilde{\Gamma}$:

Lemma 5.3.1 *For a given Horn sequent $\mathcal{S}$ of* **HTPN**, *suppose $\tilde{\mathcal{S}}$ is the rewritten Horn sequent. Then we can say that $\mathcal{S}$ is provable in* **HTPN** *if and only if $\tilde{\mathcal{S}}$ is provable in* **HTPN** *without $(next)$ rule.* ∎

For the proof figure of (5.2.1) on page 30, the corresponding proof figure of (5.3.1) without $(next)$ rule is as follows:

$$\cfrac{\cfrac{\cfrac{\cfrac{\tilde{Tr}; clock^{(2)} \otimes p_1^{(2)} \otimes p_3^{(2)} \otimes p_2^{(3)} \rightarrow clock^{(2)} \otimes p_1^{(2)} \otimes p_3^{(2)} \otimes p_2^{(3)}}{\tilde{Tr}; clock^{(2)} \otimes p_1^{(2)} \otimes p_3^{(2)} \otimes p_3^{(2)} \rightarrow clock^{(2)} \otimes p_1^{(2)} \otimes p_3^{(2)} \otimes p_2^{(3)}}\ 4}{\tilde{Tr}; clock^{(1)} \otimes p_1^{(1)} \otimes p_3^{(2)} \otimes p_3^{(2)} \rightarrow clock^{(2)} \otimes p_1^{(2)} \otimes p_3^{(2)} \otimes p_2^{(3)}}\ 3((\text{fire})3\text{ times})}{\tilde{Tr}; clock^{(0)} \otimes p_1^{(0)} \otimes p_3^{(2)} \otimes p_3^{(2)} \rightarrow clock^{(2)} \otimes p_1^{(2)} \otimes p_3^{(2)} \otimes p_2^{(3)}}\ 2((\text{fire})3\text{ times})}{\tilde{Tr}; clock^{(0)} \otimes p_1^{(0)} \otimes p_2^{(0)} \otimes p_2^{(0)} \rightarrow clock^{(2)} \otimes p_1^{(2)} \otimes p_3^{(2)} \otimes p_2^{(3)}}\ 1$$

Double lines mean that several inference rules are applied.

Each number 1 – 4 in the proof figure means one or several $(fire)$ rules which correspond to the following linear implications, respectively:

1. $clock^{(0)} \otimes p_1^{(0)} \otimes p_2^{(0)} \otimes p_2^{(0)} \multimap clock^{(0)} \otimes p_1^{(0)} \otimes p_3^{(2)} \otimes p_3^{(2)}$.
2. $clock^{(0)} \multimap tmp^{(0)}$, $tmp^{(0)} \otimes p_1^{(0)} \multimap tmp^{(0)} \otimes p_1^{(1)}$, $tmp^{(0)} \multimap clock^{(1)}$.
3. $clock^{(1)} \multimap tmp^{(1)}$, $tmp^{(1)} \otimes p_1^{(1)} \multimap tmp^{(1)} \otimes p_1^{(2)}$, $tmp^{(1)} \multimap clock^{(2)}$.
4. $clock^{(2)} \otimes p_3^{(2)} \multimap clock^{(2)} \otimes p_2^{(3)}$.

Lemma 5.3.1 concludes that the strict reachability problem for timed Petri nets can be translated into the reachability problem for Petri nets. Since the reachability problem for Petri nets is decidable [24], we can obtain the following corollary which claims that the strict reachability problem for timed Petri nets is decidable.

Corollary 5.3.1 (Decidability of the strict reachability problem)
Let S and S' be states of a timed Petri net, $t \in \mathbf{N}$. We can decide if $S' \in [S\rangle_t$. ∎

This result is similar to [29, 30]. We obtained another proof of it.

Chapter 6

Conclusions and Future Work

In this thesis, we developed a resource-conscious and time-dependent logic called temporal linear logic (**TLL**). It is a natural extension of both linear logic and temporal logic (**S4**). It has both modal storage operators and temporal operators. The temporal operators are "$\bigcirc$", which means "next", and "$\Box$", which means "anytime". The modal storage operator "!" means "reusable at anytime". A formula in **TLL** has an interpretation including concepts of both resource and time. It contains linear logic as its subsystem and **S4** can be embedded into it.

TLL is also useful as a formal logical system, in which the cut elimination theorem holds. One can obtain the cut free proof figure of a provable sequent constructively. This theorem plays an important part in logic programming, uniform proof and proof search. We designed a temporal linear logic programming language by using the idea of Miller's uniform proof [32]. Decidability and undecidability of **TLL** fragments were obtained from the results of linear logic using subformula property.

The phase semantics of linear logic was extended by a phase homomorphism. The full propositional fragment of temporal linear logic has a complete semantics in terms of temporal phase spaces. We think the soundness and completeness theorems are useful to consider model checking. We referred to the proof of the completeness theorem in [17]. In [17], the phase semantics has been extended to the second order case. It showed the strong completeness theorem, that is, if a formula is valid then it is cut free provable. It follows that the second order of the logic in [17] also satisfies the cut elimination. By a similar method, **TLL** will be able to extend to second order and be able to show the similar result.

We restricted the time concept to discrete and linear time in this thesis. We think that it will not be difficult to extend this concept to continuous linear time using the idea in [19] as follows: We introduce a new formula "$\Box^{t_1}_{t_2} A$" to mean "A can be used exactly once during time t_1 to t_2". $\bigcirc A$ can be considered a shorthand form for $\Box^1_1 A$, and $\Box A$ for $\Box^0_\infty A$.

Timed Petri nets are encoded naturally into **ITLL**$^\circ$, which is a subsystem of **TLL**. The reachability problem for timed Petri nets is equivalent to the provability of the corresponding **ITLL**$^\circ$ sequent. This result leads to the decidability of the reachability problem by a method different from [29, 30]. Using the & and the $\oplus$ fragments, it will be possible to give a detailed description of the behavior of timed Petri nets.

Although the correspondence between the concept of the concurrent processes and logical concept was rough, we considered our own communication model. In our model, using **TLL**, we can represent concurrent systems and distinguish synchronous models from asynchronous models. We will continue research on the connections with existing models such as π-calculus, CCS, CSP and so on. In particular, it is interesting to focus on statechart. The second order case of **TLL** may be useful for the argument about π-calculus.

The expressive power of our temporal linear logic could be sufficient to deal with dynamic change in process environments with the passage of time.

Acknowledgements

First of all, I would like to thank Professor Naoyuki Tamura for his help, advice and many discussions that greatly influenced my work. Through him I was first introduced to linear logic, and from him I learned a great deal about research methodology. He offered many hints that helped me achieve my work. Without his direction, this thesis would never have been completed. I would also like to thank Professor Yuzuru Kakuda. His suggestions were pointed, and his comments were invaluable. From him I learned much about research attitude and the strictness of studies.

I thank Joerg Brendle. He corrected my English in detail. His explanations about English sentences were very logical. Also I thank Miho Matsuyama, who made me realize that English in a thesis is different from daily English.

In addition I would like to thank Professor Makoto Takahashi. He lectured on Mayr's thesis [24] and gave me valuable advice. I would also like to thank Professor Susumu Hayashi, whose many suggestions helped clarify the direction of my future work.

I thank Professor Saburo Tamura who advised me to undertake the graduate course at Kobe University. My encounter with him was my first contact with logic. I learned many logical methods from him.

I also thank Makoto Kikuchi and members of CS32, my friends.

I thank Tetsuya Yokoyama and Osaka education publishing, and Horikoshi bookbinder for their help in binding my thesis.

I am indebted to everyone who generously helped me achieve my work.

Finally, I am thankful for the pure cheer of my son and I thank my wife for her support even though there were times when I tested her patience.

Thank you all.

Bibliography

[1] A.N.Prior. *Time and modality.* Oxford University Press, Oxford, 1968.

[2] Gianluigi Bellin and Philip J. Scott. On the π-calculus and linear logic. *Theoretical Computer Science*, 135(1, 5):11–65, December 1994. Proceedings of Mathematical Foundations of Programming Semantics (MFPS'92).

[3] I. I. Bestuzheva and V. V. Rudnev. Timed Petri nets: Classification and comparative analysis. *Automation and Remote Control*, 51(10):1308–1318, October 1990.

[4] F.Girault, B.Pradin-Chézalviel, L.A.Künzle, and R.Valette. Linear logic as a tool for reasoning on a Petri net model. In *1995 INRIA / IEEE Symposium on Emerging Technologies and Factory Automation*, October 1995.

[5] Gerhard Gentzen. Untersuchungen über das logische schließen. *Mathematische Zeitschrift*, 1935.

[6] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[7] Jean-Yves Girard. Linear logic: Its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 1–42. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.

[8] Takaharu Hirai. On cut-elimination of temporal linear logic and the equivalence to other systems. *Memoirs of the Graduate School of Science and Technology, Kobe Univ.*, 16-A:165 – 174, 1998.

[9] Takaharu Hirai. An application of a temporal linear logic to timed Petri nets. In *Petri Nets'99 Workshop on Applications of Petri nets to intelligent system development*, pages 2–13, June 1999.

[10] Takaharu Hirai. Propositional temporal linear logic and its application to concurrent systems. *IEICE Transactions (Special section on Concurrent Systems Technology)*, to appear.

[11] J.L.Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice Hall, 1981.

[12] J.Wang. *Timed Petri nets : Theory and application.* Kluwer Academic Publishers, Boston, 1998.

[13] Max Kanovich and Takayasu Ito. Temporal linear logic specifications for concurrent processes (extended abstract). In *Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 48–57, Warsaw, Poland, 29 June–2 July 1997. IEEE Computer Society Press.

[14] Max I. Kanovich. The Horn fragment of linear logic is NP-complete. ITLI Prepublication Series X-91-14, University of Amsterdam, 1991.

[15] Max I. Kanovich. Horn programming in linear logic is NP-complete. In *Seventh Annual Symposium on Logic in Computer Science*, pages 200–210, Santa Cruz, California, June 1992. IEEE Computer Society Press.

[16] Max I. Kanovich. Linear logic as a logic of computations. *Annals of Pure and Applied Logic*, 67(1-3):183–212, 1994. Also in *Logic at Tver '92*, Sokal, Russia, July 1992.

[17] Max I. Kanovich, Mitsuhiro Okada, and Andre Scedrov. Phase semantics for light linear logic. 23 pp., accepted for publication in Theoretical Computer Science. Extended abstract in: 13-th Annual Conference on the Mathematical Foundations of Programming Semantics, Pittsburgh, Pennsylvania, March, 1997, Electronic Notes in Theoretical Computer Science, Volume 6 (1997) 12 pp.

[18] Max I. Kanovich, Mitsuhiro Okada, and Andre Scedrov. Specifying real-time finite-state systems in linear logic. In 2-nd International Workshop on Constraint Programming for Time-Critical Applications and Multi-Agent Systems (COTIC), Nice, France, September, 1998, Electronic Notes in Theoretical Computer Science, Volume 16 Issue 1 (1998) 15 pp.

[19] Naoki Kobayashi. *Concurrent Linear Logic Programming.* PhD thesis, The University of Tokyo, April 1996.

[20] Naoki Kobayashi and Akinori Yonezawa. Higher-order concurrent linear logic programming. *Lecture Notes in Computer Science*, 907:137–166, 1995.

[21] J. Lilius. High-level nets and linear logic. In K. Jensen, editor, *Proceedings of the International Conference on Applications and Theory of Petri Nets*, pages 310–327, Sheffield, United Kingdom, June 1992. Springer-Verlag LNCS 616.

[22] Patrick Lincoln, John Mitchell, Andre Scedrov, and Natarajan Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56:239–311, April 1992. Also in the Proceedings of the 31th Annual Symposium on Foundations of Computer Science, St Louis, Missouri, October 1990, IEEE Computer Society Press. Also available as Technical Report SRI-CSL-90-08 from SRI International, Computer Science Laboratory.

[23] N. Martí-Oliet and J. Meseguer. From Petri nets to linear logic. In P. Dybjer, A. M. Pitts, D. H. Pitt, A. Poigné, and D. E. Rydeheard, editors, *Proceedings of the Conference on Category Theory and Computer Science*, Springer-Verlag LNCS 389, pages 313–340, Manchester, United Kingdom, September 1989.

[24] E. W. Mayr. An algorithm for the general Petri net reachability problem. In *Proc. of the 13th Annual ACM Symp. on Theory of Computing*, pages 238–246, 1981. Second Edition: SIAM J. Comput. Vol. 13, No. 3, Pages: 441-460, August 1984.

[25] J. Meseguer and U. Montanari. Petri nets are monoids, 1990. INFORMATION AND COMPUTATION, 88:105-155.

[26] D. Miller. The π-calculus as a theory in linear logic: Preliminary results. *Lecture Notes in Computer Science*, 660:242–265, 1993. Proc. 1992 Workshop on Extensions to Logic Programming.

[27] Mitsuhiro Okada. A concurrency model based on linear logic – towards the logical understanding of concurrency. *Journal of Information Processing Society of Japan*, 37(4):327–332, April 1996. in Japanese.

[28] Hiroakira Ono. Phase structures and quantales - a semantical study of logics without structural rules. Manuscript, 1990.

[29] V. Valero Ruíz, D. de Frutos Escrig, and F. Cuartero Gómez. Simulation of timed Petri nets by ordinary Petri nets and applications to decidability of the timed reachability problem and other related problems. In *Proceedings of the Fourth International Workshop on Petri Nets and Performance Models (PNPM91)*, pages 154–163, December 2-5 1991.

[30] V. Valero Ruíz, D. de Frutos Escrig, and F. Cuartero Gómez. Decidability of the strict reachability problem for TPN's with rational and real durations. In *5th International Workshop on Petri Nets and Performance Models*, pages 56–65,19.–22, Toulouse (F), October 1993.

[31] M. E. Szabo, editor. *The Collected Papers of Gerhard Gentzen.* North-Holland, Amsterdam, 1969.

[32] Naoyuki Tamura, Takaharu Hirai, Hideo Yoshikawa, Kyoung-Sun Kang, and Mutsunori Banbara. Logic programming in an intuitionistic temporal linear logic. *Information Processing Society of Japan Transactions on Programming*, to appear. in Japanese.

[33] Makoto Tanabe. Timed Petri nets and temporal linear logic. *Lecture Notes in Computer Science*, 1248:156–174, June 1997. 18th International Conference on Application and Theory of Petri Nets, Toulouse, France, June 1997.

[34] Anne S. Troelstra. *Lectures on Linear Logic.* CSLI Lecture Notes 29, Center for the Study of Language and Information, Stanford, California, 1992.

[35] U.Engberg and G. Winskel. Linear logic on Petri nets. *LNCS 803*, pages 176–229, Jun 1993. In A Decade of Concurrency, Reflections and Perspectives, REX School/Symposium.

[36] W.Resig. *Petri Nets.* Springer-Verlag, 1982.

Appendix A

Syntax

A.1 Classical Propositional Linear Logic

Identity and Cut rule:

$$\frac{}{D \to D}\,(I) \qquad \frac{\Gamma \to \Delta, D \quad D, \Pi \to \Lambda}{\Gamma, \Pi \to \Delta, \Lambda}\,(cut)$$

Propositional Rules :

$$\frac{\Gamma \to \Delta, D}{D^{\perp}, \Gamma \to \Delta}\,({}^{\perp}\!\to) \qquad \frac{D, \Gamma \to \Delta}{\Gamma \to \Delta, D^{\perp}}\,(\to^{\perp})$$

$$\frac{A, B, \Gamma \to \Delta}{A \otimes B, \Gamma \to \Delta}\,(\otimes \to) \qquad \frac{\Gamma \to \Delta, A \quad \Pi \to \Lambda, B}{\Gamma, \Pi \to \Delta, \Lambda, A \otimes B}\,(\to \otimes)$$

$$\frac{A, \Gamma \to \Delta \quad B, \Pi \to \Lambda}{A \wp B, \Gamma, \Pi \to \Delta, \Lambda}\,(\wp \to) \qquad \frac{\Gamma \to \Delta, A, B}{\Gamma \to \Delta, A \wp B}\,(\to \wp)$$

$$\frac{A, \Gamma \to \Delta}{A \& B, \Gamma \to \Delta}\,(\& \to)1 \qquad \frac{B, \Gamma \to \Delta}{A \& B, \Gamma \to \Delta}\,(\& \to)2 \qquad \frac{\Gamma \to \Delta, A \quad \Gamma \to \Delta, B}{\Gamma \to \Delta, A \& B}\,(\to \&)$$

$$\frac{A, \Gamma \to \Delta \quad B, \Gamma \to \Delta}{A \oplus B, \Gamma \to \Delta}\,(\oplus \to) \qquad \frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, A \oplus B}\,(\to \oplus)1 \qquad \frac{\Gamma \to \Delta, B}{\Gamma \to \Delta, A \oplus B}\,(\to \oplus)2$$

$$\frac{\Gamma \to \Delta, A \quad B, \Pi \to \Lambda}{A \multimap B, \Gamma, \Pi \to \Delta, \Lambda}\,(\multimap \to) \qquad \frac{A, \Gamma \to \Delta, B}{\Gamma \to \Delta, A \multimap B}\,(\to \multimap)$$

Constants :

$$\frac{\Gamma \to \Delta}{\mathbf{1}, \Gamma \to \Delta}\,(\mathbf{1} \to) \qquad \frac{}{\to \mathbf{1}}\,(\to \mathbf{1}) \qquad \frac{}{\Gamma \to \top, \Sigma}\,(\to \top)$$

$$\frac{}{\perp \to}\,(\perp \to) \qquad \frac{\Gamma \to \Delta}{\Gamma \to \Delta, \perp}\,(\to \perp) \qquad \frac{}{\Gamma, \mathbf{0} \to \Delta}\,(\mathbf{0} \to)$$

Exponential Rules :

$$\frac{A, \Gamma \to \Delta}{!A, \Gamma \to \Delta}\,(! \to) \qquad \frac{!\Gamma \to ?\Sigma, A}{!\Gamma \to ?\Sigma, !A}\,(\to !) \qquad \frac{\Gamma \to \Delta}{!A, \Gamma \to \Delta}\,(!w) \qquad \frac{!A, !A, \Gamma \to \Delta}{!A, \Gamma \to \Delta}\,(!c)$$

$$\frac{A, !\Gamma \to ?\Sigma}{?A, !\Gamma \to ?\Sigma}\,(? \to) \qquad \frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, ?A}\,(\to ?) \qquad \frac{\Gamma \to \Delta}{\Gamma \to \Delta, ?A}\,(?w) \qquad \frac{\Gamma \to \Delta, ?A, ?A}{\Gamma \to \Delta, ?A}\,(?c)$$

Table A.1: The sequent calculus for classical linear logic **LL**

A.2 Propositional Temporal Linear Logic

A.2.1 Classical Temporal Linear Logic

(Rules of **LL**)

Modal Rules :

$$\frac{A, \Gamma \rightarrow \Delta}{\Box A, \Gamma \rightarrow \Delta}(\Box \rightarrow) \quad \frac{!\Gamma, \Box\Pi \rightarrow A, \Diamond\Lambda, ?\Sigma}{!\Gamma, \Box\Pi \rightarrow \Box A, \Diamond\Lambda, ?\Sigma}(\rightarrow \Box)$$

$$\frac{!\Gamma, \Box\Pi, A \rightarrow \Diamond\Lambda, ?\Sigma}{!\Gamma, \Box\Pi, \Diamond A \rightarrow \Diamond\Lambda, ?\Sigma}(\Diamond \rightarrow) \quad \frac{\Gamma \rightarrow \Delta, A}{\Gamma \rightarrow \Delta, \Diamond A}(\rightarrow \Diamond)$$

$$\frac{!\Gamma, \Box\Pi, \Xi \rightarrow A, \Phi, \Diamond\Lambda, ?\Delta}{!\Gamma, \Box\Pi, \bigcirc\Xi \rightarrow \bigcirc A, \overline{\bigcirc}\Phi, \Diamond\Lambda, ?\Delta}(\bigcirc) \quad \frac{!\Gamma, \Box\Pi, \Xi, A \rightarrow \Phi, \Diamond\Lambda, ?\Delta}{!\Gamma, \Box\Pi, \bigcirc\Xi, \overline{\bigcirc}A \rightarrow \overline{\bigcirc}\Phi, \Diamond\Lambda, ?\Delta}(\overline{\bigcirc})$$

$$\frac{!\Gamma, \Box\Pi, \Xi \rightarrow \Phi, \Diamond\Lambda, ?\Delta}{!\Gamma, \Box\Pi, \bigcirc\Xi \rightarrow \overline{\bigcirc}\Phi, \Diamond\Lambda, ?\Delta}(\bigcirc \rightarrow \overline{\bigcirc})$$

Table A.2: The sequent calculus for classical temporal linear logic **TLL**

A.2.2 Intuitionistic Temporal Linear Logic

Identity and Cut rule:

$$\frac{}{D \rightarrow D}(I) \quad \frac{\Gamma \rightarrow D \quad D, \Pi \rightarrow C}{\Gamma, \Pi \rightarrow C}(cut)$$

Propositional Rules :

$$\frac{A, B, \Gamma \rightarrow C}{A \otimes B, \Gamma \rightarrow C}(\otimes \rightarrow) \quad \frac{\Gamma \rightarrow A \quad \Pi \rightarrow B}{\Gamma, \Pi \rightarrow A \otimes B}(\rightarrow \otimes)$$

$$\frac{A, \Gamma \rightarrow C}{A\&B, \Gamma \rightarrow C}(\& \rightarrow)1 \quad \frac{B, \Gamma \rightarrow C}{A\&B, \Gamma \rightarrow C}(\& \rightarrow)2 \quad \frac{\Gamma \rightarrow A \quad \Gamma \rightarrow B}{\Gamma \rightarrow A\&B}(\rightarrow \&)$$

$$\frac{A, \Gamma \rightarrow C \quad B, \Gamma \rightarrow C}{A \oplus B, \Gamma \rightarrow C}(\oplus \rightarrow) \quad \frac{\Gamma \rightarrow A}{\Gamma \rightarrow A \oplus B}(\rightarrow \oplus)1 \quad \frac{\Gamma \rightarrow B}{\Gamma \rightarrow A \oplus B}(\rightarrow \oplus)2$$

$$\frac{\Gamma \rightarrow A \quad B, \Pi \rightarrow C}{A \multimap B, \Gamma, \Pi \rightarrow C}(\multimap \rightarrow) \quad \frac{A, \Gamma \rightarrow B}{\Gamma \rightarrow A \multimap B}(\rightarrow \multimap)$$

Constants :

$$\frac{\Gamma \rightarrow C}{\mathbf{1}, \Gamma \rightarrow C}(\mathbf{1} \rightarrow) \quad \frac{}{\rightarrow \mathbf{1}}(\rightarrow \mathbf{1}) \quad \frac{}{\Gamma, \mathbf{0} \rightarrow C}(\mathbf{0} \rightarrow) \quad \frac{}{\Gamma \rightarrow \top}(\rightarrow \top)$$

Exponential Rules :

$$\frac{A, \Gamma \rightarrow C}{!A, \Gamma \rightarrow C}(! \rightarrow) \quad \frac{!\Gamma \rightarrow A}{!\Gamma \rightarrow !A}(\rightarrow !) \quad \frac{\Gamma \rightarrow C}{!A, \Gamma \rightarrow C}(!w) \quad \frac{!A, !A, \Gamma \rightarrow C}{!A, \Gamma \rightarrow C}(!c)$$

Modal Rules :

$$\frac{A, \Gamma \rightarrow C}{\Box A, \Gamma \rightarrow C}(\Box \rightarrow) \quad \frac{!\Gamma, \Box\Pi \rightarrow A}{!\Gamma, \Box\Pi \rightarrow \Box A}(\rightarrow \Box) \quad \frac{!\Gamma, \Box\Pi, \Xi \rightarrow A}{!\Gamma, \Box\Pi, \bigcirc\Xi \rightarrow \bigcirc A}(\bigcirc)$$

Table A.3: The sequent calculus for intuitionistic temporal linear logic **ITLL**

A.3 Classical Logic

Identity:

$$\frac{}{D \to D}\ (I)$$

Structural Rules :

$$\frac{\Gamma \to \Delta}{A, \Gamma \to \Delta}\ (w \to) \quad \frac{\Gamma \to \Delta}{\Gamma \to \Delta, A}\ (\to w) \quad \frac{A, A, \Gamma \to \Delta}{A, \Gamma \to \Delta}\ (c \to) \quad \frac{\Gamma \to \Delta, A, A}{\Gamma \to \Delta, A}\ (\to c)$$

$$\frac{\Gamma \to \Delta, D \quad D, \Pi \to \Lambda}{\Gamma, \Pi \to \Delta, \Lambda}\ (cut)$$

Propositional Rules :

$$\frac{\Gamma \to \Delta, D}{\neg D, \Gamma \to \Delta}\ (\neg \to) \qquad \frac{D, \Gamma \to \Delta}{\Gamma \to \Delta, \neg D}\ (\to \neg)$$

$$\frac{A, \Gamma \to \Delta}{A \wedge B, \Gamma \to \Delta}\ (\wedge \to)1 \qquad \frac{\Gamma \to \Delta, A \quad \Gamma \to \Delta, B}{\Gamma \to \Delta, A \wedge B}\ (\to \wedge)$$

$$\frac{B, \Gamma \to \Delta}{A \wedge B, \Gamma \to \Delta}\ (\wedge \to)2$$

$$\frac{A, \Gamma \to \Delta \quad B, \Gamma \to \Delta}{A \vee B, \Gamma \to \Delta}\ (\vee \to) \qquad \frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, A \vee B}\ (\to \vee)1$$

$$\frac{\Gamma \to \Delta, B}{\Gamma \to \Delta, A \vee B}\ (\to \vee)2$$

$$\frac{\Gamma \to \Delta, A \quad B, \Pi \to \Lambda}{A \supset B, \Gamma, \Pi \to \Delta, \Lambda}\ (\supset\to) \qquad \frac{A, \Gamma \to \Delta, B}{\Gamma \to \Delta, A \supset B}\ (\to\supset)$$

Table A.4: The sequent calculus for classical logic **LK**

A.4 S4

(Rules of **LK**)

Modal Rules :

$$\frac{A, \Gamma \to \Delta}{\Box A, \Gamma \to \Delta}\ (\Box \to) \quad \frac{\Box\Gamma \to A, \Diamond\Lambda}{\Box\Gamma \to \Box A, \Diamond\Lambda}\ (\to \Box)$$

$$\frac{\Box\Gamma, A \to \Diamond\Lambda}{\Box\Gamma, \Diamond A \to \Diamond\Lambda}\ (\Diamond \to) \quad \frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, \Diamond A}\ (\to \Diamond)$$

Table A.5: The sequent calculus for **S4**

Note that for all of systems, exchange is implicit.

Appendix B

Properties of TLL Phase Semantics

We list here some properties of **TLL** phase semantics similar to the case of linear logic:

- If $A \subseteq B$ then $A \otimes C \subseteq B \otimes C$ in any temporal phase structure,
- $\mathbf{1} \otimes A = A$ in any temporal phase structure,
- $A \otimes (B \oplus C) = (A \otimes B) \oplus (A \otimes C)$ in any temporal phase structure,
- $A \multimap B = A^{\perp} \wp B$ in any temporal phase space,
- $\mathbf{1}^{\perp} = \perp$, $\perp^{\perp} = \mathbf{1}$ in any temporal phase space,
- $\top^{\perp} = \mathbf{0}$, $\mathbf{0}^{\perp} = \top$ in any temporal phase space,
- $(A \otimes B)^{\perp} = A^{\perp} \wp B^{\perp}$, $(A \wp B)^{\perp} = A^{\perp} \otimes B^{\perp}$ in any temporal phase space,
- $(A \& B)^{\perp} = A^{\perp} \oplus B^{\perp}$, $(A \oplus B)^{\perp} = A^{\perp} \& B^{\perp}$ in any temporal phase space,
- if $A \subseteq B$ then $A \wp C \subseteq B \wp C$ in any temporal phase space,
- $A = A \wp \perp$ in any temporal phase space,
- $A \wp (B \& C) = (A \wp B) \& (A \wp C)$ in any temporal phase space,
- $(A \wp B) \otimes C \subseteq (A \otimes C) \wp B$ in any temporal phase space,

where A, B, C are facts.

Appendix C

Proofs of Theorem, Corollary and Lemma in Phase Semantics

C.1 Soundness

Theorem 4.2.1 (Soundness) *If a sequent is provable in* **TLL**, *then it is valid.* ∎

Proof. The argument is by induction on the length of propositional **TLL** proof. For $\Gamma = C_1, \ldots, C_m$ $(m \geq 0)$, we use a notation $\Gamma^{*\otimes}$ as $C_1^* \otimes \ldots \otimes C_m^*$. Similarly, we use $\Gamma^{*\wp}$, and so on. We consider only the modal rules, since all other cases are standard [6].

Case 1: The last rule of the proof is $(\Box \rightarrow)$ rule of the form:

$$\frac{A, \Gamma \rightarrow \Delta}{\Box A, \Gamma \rightarrow \Delta} \ (\Box \rightarrow)$$

By the induction hypothesis,

$$A^* \otimes \Gamma^{*\otimes} \subseteq \Delta^{*\wp}.$$

By Lemma 4.1.2 we have $\Box A^* \subseteq A^*$. Thus,

$$\Box A^* \otimes \Gamma^{*\otimes} \subseteq A^* \otimes \Gamma^{*\otimes}.$$

Hence,

$$\Box A^* \otimes \Gamma^{*\otimes} \subseteq \Delta^{*\wp}.$$

Case 2: The last rule of the proof is $(\rightarrow \Box)$ rule of the form:

$$\frac{!\Gamma, \Box \Pi \rightarrow A, \Diamond \Lambda, ?\Sigma}{!\Gamma, \Box \Pi \rightarrow \Box A, \Diamond \Lambda, ?\Sigma} \ (\rightarrow \Box)$$

Let $\Gamma = B_1, \ldots, B_m$, $\Sigma = C_1, \ldots, C_n$, $\Pi = D_1, \ldots, D_l$, $\Lambda = E_1, \ldots, E_u$ $(m, n, l, u \geq 0)$. Note that each B_i^*, C_j^*, D_k^*, E_r^* and A^* is a fact.

2.1 $\Gamma = \Pi = \Lambda = \Sigma = \epsilon$, where ϵ denotes an empty sequence.
By the induction hypothesis, $\mathbf{1}^* \subseteq A^*$, that is, $1 \in A^*$. We have

$$f(1) = 1 \in f(A^*)$$

since f is a monoid homomorphism. Hence,

$$1 \in f(A^*) \cap A^*.$$

Therefore,

$$\mathbf{1}^* \subseteq f(A^*) \cap A^* \subseteq \Box A^*.$$

2.2 Other cases.

By the induction hypothesis,

$$!B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \subseteq A^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*.$$

From this, it is straightforward to obtain

$$!B_1^* \otimes \ldots \otimes !B_m^* \otimes !C_1^{*\perp} \otimes \ldots \otimes !C_n^{*\perp} \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \Box E_1^{*\perp} \otimes \ldots \otimes \Box E_u^{*\perp} \subseteq A^*.$$

Hence,

$$(f(B_1^*) \cap B_1^* \cap K) \cdots (f(B_m^*) \cap B_1^* \cap K) \cdot (f(C_1^{*\perp}) \cap C_1^{*\perp} \cap K) \cdots (f(C_n^{*\perp}) \cap C_n^{*\perp} \cap K) \cdot$$
$$(f(D_1^*) \cap D_1^*) \cdots (f(D_l^*) \cap D_l^*) \cdot (f(E_1^{*\perp}) \cap E_1^{*\perp}) \cdots (f(E_1^{*\perp}) \cap E_u^{*\perp}) \subseteq A^*,$$

Here, let

$$\begin{array}{lllllll} \tilde{B}_i & = & f(B_i^*) \cap B_i^* \quad (0 \leq i \leq m), & \tilde{C}_j & = & f(C_j^{*\perp}) \cap C_j^{*\perp} \quad (0 \leq j \leq n), \\ \tilde{D}_k & = & f(D_k^*) \cap D_k^* \quad (0 \leq k \leq l), & \tilde{E}_s & = & f(E_s^{*\perp}) \cap E_s^{*\perp} \quad (0 \leq s \leq u). \end{array}$$

Then,

$$(\tilde{B}_1 \cap K) \cdots (\tilde{B}_m \cap K) \cdot (\tilde{C}_1 \cap K) \cdots (\tilde{C}_n \cap K) \cdot \tilde{D}_1 \cdots \tilde{D}_l \cdot \tilde{E}_1 \cdots \tilde{E}_u \subseteq \quad A^*.$$

By definition, $f(f(X)) = f(X)$ for any $X \subseteq M$. Hence,

$$f(X) \cap X \subseteq f(X) = f(f(X)) \cap f(X) = f(f(X) \cap X)$$

for any $X \subseteq M$.

Similarly,

$$f(X) \cap X \cap K \subseteq f(f(X) \cap X) \cap K \subseteq f(f(X) \cap X) \cap f(K) = f(f(X) \cap X \cap K)$$

since $K \subseteq f(K)$ by definition. Hence,

$$\begin{array}{ll} & (\tilde{B}_1 \cap K) \cdots (\tilde{B}_m \cap K) \cdot (\tilde{C}_1 \cap K) \cdots (\tilde{C}_n \cap K) \cdot \tilde{D}_1 \cdots \tilde{D}_l \cdot \tilde{E}_1 \cdots \tilde{E}_u \\ \subseteq & f(\tilde{B}_1 \cap K) \cdots f(\tilde{B}_m \cap K) \cdot f(\tilde{C}_1 \cap K) \cdots f(\tilde{C}_n \cap K) \cdot f(\tilde{D}_1) \cdots f(\tilde{D}_l) \cdot f(\tilde{E}_1) \cdots f(\tilde{E}_u) \\ \subseteq & f((\tilde{B}_1 \cap K) \cdots (\tilde{B}_m \cap K) \cdot (\tilde{C}_1 \cap K) \cdots (\tilde{C}_n \cap K) \cdot \tilde{D}_1 \cdots \tilde{D}_l \cdot \tilde{E}_1 \cdots \tilde{E}_u) \\ \subseteq & f(A^*). \end{array}$$

since f is a monoid homomorphism. Hence, we have obtained

$$(\tilde{B}_1 \cap K) \cdots (\tilde{B}_m \cap K) \cdot (\tilde{C}_1 \cap K) \cdots (\tilde{C}_n \cap K) \cdot \tilde{D}_1 \cdots \tilde{D}_l \cdot \tilde{E}_1 \cdots \tilde{E}_u \quad \subseteq \quad A^* \cap f(A^*).$$

Therefore,

$$!B_1^* \otimes \ldots \otimes !B_m^* \otimes !C_1^{*\perp} \otimes \ldots \otimes !C_n^{*\perp} \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \Box E_1^{*\perp} \otimes \ldots \otimes \Box E_u^{*\perp} \subseteq \Box A^*.$$

From this, it is straightforward to obtain

$$!B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \subseteq \Box A^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*.$$

Case 3: The last rule of the proof is $(\diamond \to)$ rule of the form:

$$\frac{!\Gamma, \Box\Pi, A \to \diamond\Lambda, ?\Sigma}{!\Gamma, \Box\Pi, \diamond A \to \diamond\Lambda, ?\Sigma} \ (\diamond \to)$$

Let $\Gamma = B_1, \ldots, B_m$, $\Sigma = C_1, \ldots, C_n$, $\Pi = D_1, \ldots, D_l$, $\Lambda = E_1, \ldots, E_u$ $(m, n, l, u \geq 0)$. Note that each B_i^*, C_j^*, D_k^*, E_r^* and A^* is a fact.

By the induction hypothesis,

$$!B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes A^* \subseteq \diamond E_1^* \wp \ldots \wp \diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*.$$

From this, it is straightforward to obtain

$$!B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \subseteq A^{*\perp} \wp \diamond E_1^* \wp \ldots \wp \diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*.$$

Since we can reduce the Case 2, we can obtain

$$!B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \subseteq \Box A^{*\perp} \wp \diamond E_1^* \wp \ldots \wp \diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*,$$

that is,

$$!B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \diamond A^* \subseteq \diamond E_1^* \wp \ldots \wp \diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*.$$

Case 4: The last rule of the proof is $(\to \diamond)$ rule of the form:

$$\frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, \diamond A} \ (\to \diamond)$$

Let $\Gamma = B_1, \ldots, B_m$, $\Delta = C_1, \ldots, C_n$,$(m, n \geq 0)$. Note that each A^* is a fact.

By the induction hypothesis,

$$B_1^* \otimes \ldots \otimes B_m^* \subseteq C_1^* \wp \ldots \wp C_n^* \wp A^*.$$

From this, it is straightforward to obtain

$$B_1^* \otimes \ldots \otimes B_m^* \otimes A^{*\perp} \subseteq C_1^* \wp \ldots \wp C_n^*.$$

Since we can reduce the Case 1, we can obtain

$$B_1^* \otimes \ldots \otimes B_m^* \otimes \Box A^{*\perp} \subseteq C_1^* \wp \ldots \wp C_n^*,$$

that is,

$$B_1^* \otimes \ldots \otimes B_m^* \subseteq C_1^* \wp \ldots \wp C_n^* \wp \diamond A^*.$$

Case 5: The last rule of the proof is $(\bigcirc)$ rule of the form:

$$\frac{!\Gamma, \Box\Pi, \Xi \to A, \Phi, \diamond\Lambda, ?\Sigma}{!\Gamma, \Box\Pi, \bigcirc\Xi \to \bigcirc A, \overline{\bigcirc}\Phi, \diamond\Lambda, ?\Sigma} \ (\bigcirc)$$

Let $\Gamma = B_1, \ldots, B_m$, $\Sigma = C_1, \ldots, C_n$, $\Pi = D_1, \ldots, D_l$, $\Lambda = E_1, \ldots, E_u$, $\Xi = F_1, \ldots, F_v$, $\Phi = G_1, \ldots, G_w$ $(m, n, l, u, v, w \geq 0)$. Note that each B_i^*, C_j^*, D_k^*, E_r^*, F_s^*, G_t^* and A^* is a fact.

5.1 $\Gamma = \Pi = \Xi = \Phi = \Lambda = \Sigma = \epsilon$, where ϵ denotes an empty sequence.
By the induction hypothesis, $\mathbf{1}^* \subseteq A^*$, that is, $1 \in A^*$. We have

$$h(1) = 1 \in h(A^*)$$

since h is a monoid homomorphism. Therefore,

$$\mathbf{1}^* \subseteq \bigcirc A^*.$$

5.2 Other cases.
By the induction hypothesis,

$$\begin{array}{rl} & !B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes F_1^* \otimes \ldots \otimes F_v^* \\ \subseteq & A^* \wp G_1^* \wp \ldots \wp G_w^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*. \end{array}$$

From this, it is straightforward to obtain

$$\begin{array}{l} !B_1^* \otimes \ldots \otimes !B_m^* \otimes !C_1^{*\perp} \otimes \ldots \otimes !C_n^{*\perp} \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \Box E_1^{*\perp} \otimes \ldots \otimes \Box E_u^{*\perp} \\ \otimes F_1^* \otimes \ldots \otimes F_v^* \otimes G_1^{*\perp} \otimes \ldots \otimes G_w^{*\perp} \quad \subseteq \quad A^*. \end{array}$$

Hence,

$$\begin{array}{l} !B_1^* \cdot \ldots \cdot !B_m^* \cdot !C_1^{*\perp} \cdot \ldots \cdot !C_n^{*\perp} \cdot \Box D_1^* \cdot \ldots \cdot \Box D_l^* \cdot \Box E_1^{*\perp} \cdot \ldots \cdot \Box E_u^{*\perp} \\ \cdot F_1^* \cdot \ldots \cdot F_v^* \cdot G_1^{*\perp} \cdot \ldots \cdot G_w^{*\perp} \quad \subseteq \quad A^*. \end{array}$$

Since h is a monoid homomorphism,

$$\begin{array}{l} h(!B_1^*) \cdot \ldots \cdot h(!B_m^*) \cdot h(!C_1^{*\perp}) \cdot \ldots \cdot h(!C_n^{*\perp}) \\ \cdot h(\Box D_1^*) \cdot \ldots \cdot h(\Box D_l^*) \cdot h(\Box E_1^{*\perp}) \cdot \ldots \cdot h(\Box E_u^{*\perp}) \\ \cdot h(F_1^*) \cdot \ldots \cdot h(F_v^*) \cdot h(G_1^{*\perp}) \cdot \ldots \cdot h(G_w^{*\perp}) \quad \subseteq \quad h(A^*). \end{array}$$

Hence,

$$\begin{array}{l} \bigcirc !B_1^* \cdot \ldots \cdot \bigcirc !B_m^* \cdot \bigcirc !C_1^{*\perp} \cdot \ldots \cdot \bigcirc !C_n^{*\perp} \cdot \bigcirc \Box D_1^* \cdot \ldots \cdot \bigcirc \Box D_l^* \cdot \bigcirc \Box E_1^{*\perp} \cdot \ldots \cdot \bigcirc \Box E_u^{*\perp} \\ \cdot \bigcirc F_1^* \cdot \ldots \cdot \bigcirc F_v^* \cdot \bigcirc G_1^{*\perp} \cdot \ldots \cdot \bigcirc G_w^{*\perp} \quad \subseteq \quad \bigcirc A^*. \end{array}$$

For any $X \subseteq M$, we can say that

$$!X \subseteq !!X \subseteq \bigcirc !X$$

by Lemma 4.1.2. Also,

$$\Box X \subseteq \Box\Box X \subseteq \bigcirc \Box X$$

by the same Lemma. Hence,

$$\begin{array}{l} !B_1^* \cdot \ldots \cdot !B_m^* \cdot !C_1^{*\perp} \cdot \ldots \cdot !C_n^{*\perp} \cdot \Box D_1^* \cdot \ldots \cdot \Box D_l^* \cdot \Box E_1^{*\perp} \cdot \ldots \cdot \Box E_u^{*\perp} \\ \cdot \bigcirc F_1^* \cdot \ldots \cdot \bigcirc F_v^* \cdot \bigcirc G_1^{*\perp} \cdot \ldots \cdot \bigcirc G_w^{*\perp} \quad \subseteq \quad \bigcirc A^*. \end{array}$$

Therefore,

$$\begin{array}{l} !B_1^* \otimes \ldots \otimes !B_m^* \otimes !C_1^{*\perp} \otimes \ldots \otimes !C_n^{*\perp} \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \Box E_1^{*\perp} \otimes \ldots \otimes \Box E_u^{*\perp} \\ \otimes \bigcirc F_1^* \otimes \ldots \otimes \bigcirc F_v^* \otimes \bigcirc G_1^{*\perp} \otimes \ldots \otimes \bigcirc G_w^{*\perp} \quad \subseteq \quad \bigcirc A^*. \end{array}$$

From this, it is straightforward to obtain

$$\begin{array}{rl} & !B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \bigcirc F_1^* \otimes \ldots \otimes \bigcirc F_v^* \\ \subseteq & \bigcirc A^* \wp \overline{\bigcirc} G_1^* \wp \ldots \wp \overline{\bigcirc} G_w^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*. \end{array}$$

Case 6: The last rule of the proof is $(\overline{\bigcirc})$ rule of the form:

$$\frac{!\Gamma, \Box\Pi, \Xi, A \to \Phi, \Diamond\Lambda, ?\Sigma}{!\Gamma, \Box\Pi, \bigcirc\Xi, \overline{\bigcirc}A \to \overline{\bigcirc}\Phi, \Diamond\Lambda, ?\Sigma} \ (\overline{\bigcirc})$$

Let $\Gamma = B_1, \ldots, B_m$, $\Sigma = C_1, \ldots, C_n$, $\Pi = D_1, \ldots, D_l$, $\Lambda = E_1, \ldots, E_u$, $\Xi = F_1, \ldots, F_v$, $\Phi = G_1, \ldots, G_w$ $(m, n, l, u, v, w \geq 0)$. Note that each B_i^*, C_j^*, D_k^*, E_r^*, F_s^*, G_t^* and A^* is a fact.

By the induction hypothesis,

$$\begin{array}{ll} & !B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes F_1^* \otimes \ldots \otimes F_v^* \otimes A^* \\ \subseteq & G_1^* \wp \ldots \wp G_w^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*. \end{array}$$

From this, it is straightforward to obtain

$$\begin{array}{ll} & !B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes F_1^* \otimes \ldots \otimes F_v^* \\ \subseteq & A^{*\bot} \wp G_1^* \wp \ldots \wp G_w^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*. \end{array}$$

Since we can reduce the Case 5, we can obtain

$$\begin{array}{ll} & !B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \bigcirc F_1^* \otimes \ldots \otimes \bigcirc F_v^* \\ \subseteq & \bigcirc A^{*\bot} \wp \overline{\bigcirc} G_1^* \wp \ldots \wp \overline{\bigcirc} G_w^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*, \end{array}$$

that is,

$$\begin{array}{ll} & !B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \bigcirc F_1^* \otimes \ldots \otimes \bigcirc F_v^* \otimes \overline{\bigcirc} A^* \\ \subseteq & \overline{\bigcirc} G_1^* \wp \ldots \wp \overline{\bigcirc} G_w^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*. \end{array}$$

Case 7: The last rule of the proof is $(\bigcirc \to \overline{\bigcirc})$ rule of the form:

$$\frac{!\Gamma, \Box\Pi, \Xi \to \Phi, \Diamond\Lambda, ?\Sigma}{!\Gamma, \Box\Pi, \bigcirc\Xi \to \overline{\bigcirc}\Phi, \Diamond\Lambda, ?\Sigma} \ (\bigcirc \to \overline{\bigcirc})$$

Let $\Gamma = B_1, \ldots, B_m$, $\Sigma = C_1, \ldots, C_n$, $\Pi = D_1, \ldots, D_l$, $\Lambda = E_1, \ldots, E_u$, $\Xi = F_1, \ldots, F_v$, $\Phi = G_1, \ldots, G_w$ $(m, n, l, u, v, w \geq 0)$. Note that each B_i^*, C_j^*, D_k^*, E_r^*, F_s^*, G_t^* is a fact.

As in Case 5, we can obtain

$$\begin{array}{l} h(!B_1^*) \cdot \ldots \cdot h(!B_m^*) \cdot h(!C_1^{*\bot}) \cdot \ldots \cdot h(!C_n^{*\bot}) \\ \cdot h(\Box D_1^*) \cdot \ldots \cdot h(\Box D_l^*) \cdot h(\Box E_1^{*\bot}) \cdot \ldots \cdot h(\Box E_u^{*\bot}) \\ \cdot h(F_1^*) \cdot \ldots \cdot h(F_v^*) \cdot h(G_1^{*\bot}) \cdot \ldots \cdot h(G_w^{*\bot}) \quad \subseteq \quad h(\bot^*). \end{array}$$

Since h is a phase homomorphism, we have $h(\bot^*) \subseteq \bot^*$. Hence,

$$\begin{array}{l} h(!B_1^*) \cdot \ldots \cdot h(!B_m^*) \cdot h(!C_1^{*\bot}) \cdot \ldots \cdot h(!C_n^{*\bot}) \\ \cdot h(\Box D_1^*) \cdot \ldots \cdot h(\Box D_l^*) \cdot h(\Box E_1^{*\bot}) \cdot \ldots \cdot h(\Box E_u^{*\bot}) \\ \cdot h(F_1^*) \cdot \ldots \cdot h(F_v^*) \cdot h(G_1^{*\bot}) \cdot \ldots \cdot h(G_w^{*\bot}) \quad \subseteq \quad \bot^* . \end{array}$$

Then, as in Case 5,

$$\begin{array}{ll} & !B_1^* \otimes \ldots \otimes !B_m^* \otimes \Box D_1^* \otimes \ldots \otimes \Box D_l^* \otimes \bigcirc F_1^* \otimes \ldots \otimes \bigcirc F_v^* \\ \subseteq & \overline{\bigcirc} G_1^* \wp \ldots \wp \overline{\bigcirc} G_w^* \wp \Diamond E_1^* \wp \ldots \wp \Diamond E_u^* \wp ?C_1^* \wp \ldots \wp ?C_n^*. \end{array}$$

(Q.E.D.)

C.2 Corollary of the Main Lemma

Corollary 4.3.1 (Corollary of the Main Lemma) *For any formula A, $A \in A^*$.* ■

Proof. By induction on the structure of the formula A. We consider $A^* = A^{*\perp\perp}$.

Case 1: $A = \perp$

$\vdash \perp \rightarrow$ by ($\perp\rightarrow$). In other words, $\perp \in ||\epsilon|| = \perp^*$.

Case 2: $A = \mathbf{1}$

Let $\vdash \ \rightarrow \Delta$ for any formula Δ. Then

$$\frac{\rightarrow \Delta}{\mathbf{1} \rightarrow \Delta}\ (\mathbf{1} \rightarrow)$$

that is, $\mathbf{1} \in \mathbf{1}^*$.

Case 3: $A = \top$

Obviously, $\top \in M = \top^*$.

Case 4: $A = \mathbf{0}$

Obviously, $\mathbf{0} \in \mathbf{0}^*$ since $\vdash \mathbf{0} \rightarrow \Delta$ for any Δ by ($\mathbf{0} \rightarrow$).

Case 5: $A = p$ (atomic)

$p \in ||p|| = p^*$ since $\vdash p \rightarrow p$ by (I).

Case 6: $A = B^\perp$

Let $\Gamma \in B$. By the Main Lemma, $\vdash \Gamma \rightarrow B$. We can deduce

$$\frac{\Gamma \rightarrow B}{\Gamma, B^\perp \rightarrow}\ ({}^\perp\!\rightarrow)$$

that is, $B^\perp \in (B^*)^\perp$.

Case 7: $A = B \otimes C$. Suppose $\forall \Pi \in B^*, C^* (\vdash \Pi \rightarrow \Delta)$ for any Δ. By the Main Lemma and the induction hypothesis,

$$B \in B^* \text{ and } C \in C^*.$$

Then $B, C \in B^*, C^*$. Take B, C as Π, and we obtain $\vdash B, C \rightarrow \Delta$. We can deduce

$$\frac{B, C \rightarrow \Delta}{B \otimes C \rightarrow \Delta}\ (\otimes \rightarrow)$$

Hence $\vdash B \otimes C \rightarrow \Delta$, that is, $B \otimes C \in B^* \otimes C^* = (B^*, C^*)^{\perp\perp}$.

Case 8: $A = B \wp C$.

Let $\Delta_1 \in B^{*\perp}$ and $\Delta_2 \in C^{*\perp}$. By $\Delta_1 \in B^{*\perp}$, $\forall \Sigma \in B^* (\vdash \Delta_1, \Sigma \rightarrow\ )$. By the Main Lemma and the induction hypothesis, we can take $B \in B^*$ as Σ. Hence, $\vdash \Delta_1, B \rightarrow$. Similarly, $\vdash \Delta_2, C \rightarrow$. Then, we can deduce

$$\frac{B, \Delta_1 \rightarrow \qquad C, \Delta_2 \rightarrow}{B \wp C, \Delta_1, \Delta_2 \rightarrow}\ (\wp \rightarrow)$$

In other words, $B \wp C \in (B^{*\perp}, C^{*\perp})^\perp = B^* \wp C^*$.

Case 9: $A = B\&C$.

Suppose $\forall \Pi \in B^*(\vdash \Pi \to \Delta)$ for any Δ. By the Main Lemma and the induction hypothesis, we can take $B \in B^*$ as Π, that is, $\vdash B \to \Delta$. We can deduce

$$\dfrac{B \to \Delta}{B\&C \to \Delta}\;(\& \to)$$

Hence, $\vdash B\&C \to \Delta$. Thus, $B\&C \in B^{*\perp\perp} = B^*$ since B^* is a fact.

Similarly, $B\&C \in C^*$. Therefore, we obtain $B\&C \in B^* \cap C^* = B^*\&C^*$.

Case 10: $A = B \oplus C$.

Suppose $\forall \Pi \in B^* \cup C^*(\vdash \Pi \to \Delta)$ for any Δ. By the Main Lemma and the induction hypothesis, we can take $B \in B^* \subseteq B^* \cup C^*$ as Π, that is, $\vdash B \to \Delta$.

Similarly, $\vdash C \to \Delta$. We can deduce

$$\dfrac{B \to \Delta \quad C \to \Delta}{B \oplus C \to \Delta}\;(\oplus \to)$$

that is, $B \oplus C \in B^* \oplus C^* = (B^* \cup C^*)^{\perp\perp}$.

Case 11: $A = B \multimap C$.

Suppose $\Lambda \in B^*$. By the Main Lemma, $B^* \subseteq ||B||$. Hence, $\vdash \Lambda \to B$.

Suppose $\forall \Pi \in C^*(\vdash \Pi \to \Delta)$ for any Δ. By the Main Lemma and the induction hypothesis, we can take $C \in C^*$ as Π, that is, $\vdash C \to \Delta$. We can deduce

$$\dfrac{\Lambda \to B \quad C \to \Delta}{B \multimap C, \Lambda \to \Delta}\;(\multimap \to)$$

Hence, $\vdash B \multimap C, \Lambda \to \Delta$. We can say that $\forall \Pi \in C^*(\vdash \Pi \to \Delta)$ implies $\vdash B \multimap C, \Lambda \to \Delta$. In other words, $B \multimap C, \Lambda \in C^*$.

Moreover, we can say that $\Lambda \in B^*$ implies $B \multimap C, \Lambda \in C^*$ for any Λ. Threfore, $B \multimap C \in B^* \multimap C^*$.

Case 12: $A = \bigcirc B$.

By the Main Lemma and the induction hypothesis, $B \in B^*$. Therefore,

$$\bigcirc B = h(B) \in h(B^*) \subseteq h(B^*)^{\perp\perp} = \bigcirc B^*.$$

Case 13: $A = \overline{\bigcirc} B$.

Suppose $\Pi \in h(B^{*\perp})$, that is, $\Pi = \bigcirc \Pi'$ and $\Pi' \in B^{*\perp}$. By $\Pi' \in B^{*\perp}$, $\forall \Delta \in B^*(\vdash \Pi', \Delta \to \ )$. By the Main Lemma and the induction hypothesis, we can take $B \in B^*$ as Δ, that is, $\vdash \Pi', B \to$. We can deduce

$$\dfrac{\dfrac{\dfrac{\dfrac{B \to B}{B^\perp, B \to}\;(^\perp\!\!\to)}{\bigcirc B^\perp, \overline{\bigcirc} B \to}\;(\overline{\bigcirc})}{\overline{\bigcirc} B \to (\bigcirc B^\perp)^\perp}\;(\to^\perp) \qquad \dfrac{\dfrac{\dfrac{\Pi', B \to}{\Pi' \to B^\perp}\;(\to^\perp)}{\bigcirc \Pi' \to \bigcirc B^\perp}\;(\bigcirc)}{(\bigcirc B^\perp)^\perp, \bigcirc \Pi' \to}\;(^\perp\!\!\to)}{\overline{\bigcirc} B, \Pi \to}\;(cut)$$

We can say that $\Pi \in h(B^{*\perp})$ implies $\vdash \overline{\bigcirc} B, \Pi \to$. In other words, $\overline{\bigcirc} B \in h(B^{*\perp})^\perp = \overline{\bigcirc} B^*$.

Case 14: $A = \Box B$.

By the Main Lemma and the induction hypothesis, $B \in B^*$. By Lemma 4.3.1, $\Box B \in B^*$. By the definition of f, $\Box B = f(\Box B) \in f(B^*)$. Hence,

$$\Box B \in B^* \cap f(B^*) \subseteq (B^* \cap f(B^*))^{\perp\perp} = \Box B^*.$$

Case 15: $A = \Diamond B$.

Suppose $\Pi \in B^{*\perp} \cap f(B^{*\perp})$, that is, $\Pi = \Box \Pi'$ and $\forall \Delta \in B^{*}(\vdash \Pi', \Delta \rightarrow\ )$ since $\Pi' \in B^{*\perp}$. By the Main Lemma and the induction hypothesis, we can take $B \in B^{*}$ as Δ, that is, $\vdash \Pi', B \rightarrow\ $. We can deduce

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \rightarrow B}{B^{\perp}, B \rightarrow}\,({}^{\perp}\!\rightarrow)}{\Box B^{\perp}, B \rightarrow}\,(\Box \rightarrow)}{\Box B^{\perp}, \Diamond B \rightarrow}\,(\Diamond \rightarrow)}{\Diamond B \rightarrow (\Box B^{\perp})^{\perp}}\,(\rightarrow^{\perp}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\Pi', B \rightarrow}{\Pi' \rightarrow B^{\perp}}\,(\rightarrow^{\perp})}{\Box \Pi' \rightarrow B^{\perp}}\,(\Box \rightarrow)}{\Box \Pi' \rightarrow \Box B^{\perp}}\,(\rightarrow \Box)}{(\Box B^{\perp})^{\perp}, \Box \Pi' \rightarrow}\,({}^{\perp}\!\rightarrow)}{\Diamond B, \Pi \rightarrow}\,(cut)$$

We can say that $\Pi \in B^{*\perp} \cap f(B^{*\perp})$ implies $\vdash \Diamond B, \Pi \rightarrow\ $, that is, $\Diamond B \in (B^{*\perp} \cap f(B^{*\perp})^{\perp} = \Diamond B^{*}$.

Case 16: $A = !B$.

Note that $!B^{*} = (B^{*} \cap K)^{\perp\perp}$ in our canonical model. By the Main Lemma and the induction hypothesis, $B \in B^{*}$. By Lemma 4.3.1, $!B \in B^{*}$. Also, $!B \in K$ by the definition of K. Hence,

$$!B \in B^{*} \cap K \subseteq (B^{*} \cap K)^{\perp\perp} = !B^{*}.$$

Case 17: $A = ?B$.

Note that $?B^{*} = (B^{*\perp} \cap K)^{\perp}$ in our canonical model.

Suppose $\Pi \in B^{*\perp} \cap K$, that is, $\Pi = !\Pi'$ and $\forall \Delta \in B^{*}(\vdash !\Pi', \Delta \rightarrow\ )$ since $!\Pi' \in B^{*\perp}$. By the Main Lemma and the induction hypothesis, we can take $B \in B^{*}$ as Δ, that is, $\vdash !\Pi', B \rightarrow\ $. We can deduce

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \rightarrow B}{B^{\perp}, B \rightarrow}\,({}^{\perp}\!\rightarrow)}{!B^{\perp}, B \rightarrow}\,(! \rightarrow)}{!B^{\perp}, ?B \rightarrow}\,(? \rightarrow)}{?B \rightarrow (!B^{\perp})^{\perp}}\,(\rightarrow^{\perp}) \qquad \dfrac{\dfrac{\dfrac{!\Pi', B \rightarrow}{!\Pi' \rightarrow B^{\perp}}\,(\rightarrow^{\perp})}{!\Pi' \rightarrow !B^{\perp}}\,(\rightarrow !)}{(!B^{\perp})^{\perp}, !\Pi' \rightarrow}\,({}^{\perp}\!\rightarrow)}{?B, \Pi \rightarrow}\,(cut)$$

We can say that $\Pi \in B^{*\perp} \cap K$ implies $\vdash ?B, \Pi \rightarrow\ $, that is, $?B \in (B^{*\perp} \cap K)^{\perp} = ?B^{*}$. (Q.E.D.)

C.3 The Main Lemma

Lemma 4.3.2 (Main Lemma) *For any formula A,*

$$A^* \subseteq ||A||.$$

■

Proof. By induction on the structure of the formula A.

Case 1: $A = \perp$
$\perp^* = ||\epsilon||$ by definition. By Lemma 4.3.3, $||\epsilon|| = || \perp ||$.

Case 2: $A = \mathbf{1}$
Let $\Gamma \in \mathbf{1}^* = \{\epsilon\}^{\perp\perp}$, that is, if $\vdash \ \rightarrow \Delta$ then $\vdash \Gamma \rightarrow \Delta$ for any Δ. We can take $\mathbf{1}$ as Δ since $\vdash \ \rightarrow \mathbf{1}$. Hence, $\Gamma \in ||\mathbf{1}||$.

Case 3: $A = \top$
Obviously, $\vdash \Gamma \rightarrow \top$ for any $\Gamma \in M$.

Case 4: $A = \mathbf{0}$
Let $\Gamma \in \mathbf{0}^* = \emptyset^{\perp\perp}$. Then $\vdash \Gamma \rightarrow \Delta$ for any Δ. Take $\mathbf{0}$ as Δ, and we obtain $\vdash \Gamma \rightarrow \mathbf{0}$, that is, $\Gamma \in ||\mathbf{0}||$.

Case 5: $A = p$ (atomic).
Obviously, $p^* = ||p||$ by definition.

Case 6: $A = B^\perp$
Let $\Gamma \in (B^\perp)^* = B^{*\perp}$, that is, if $\vdash \Delta \rightarrow$ then $\vdash \Gamma, \Delta \rightarrow$ for any $\Delta \in B^*$. By the induction hypothesis $B^* \subseteq ||B||$, and hence by Corollary 4.3.1, $B \in B^*$. Thus, we can take B as Δ. We can deduce

$$\frac{\Gamma, B \rightarrow}{\Gamma \rightarrow B^\perp} \ (\rightarrow^\perp)$$

Therefore, $\Gamma \in ||B^\perp||$.

Case 7: $A = B \otimes C$.
By the induction hypothesis, $B^* \subseteq ||B||$. Suppose $\Delta_1 \in B^*$, and we obtain $\Delta_1 \in ||B||$, that is, $\vdash \Delta_1 \rightarrow B$. Similarly, if $\Delta_2 \in C^*$ then $\vdash \Delta_2 \rightarrow C$, by the induction hypothesis $C^* \subseteq ||C||$. Hence for $\Delta_1, \Delta_2 \in B^*, C^*$, $\vdash \Delta_1, \Delta_2 \rightarrow B \otimes C$ since

$$\frac{\Delta_1 \rightarrow B \quad \Delta_2 \rightarrow C}{\Delta_1, \Delta_2 \rightarrow B \otimes C} \ (\rightarrow \otimes)$$

In other words, $\Delta_1, \Delta_2 \in ||B \otimes C||$. Hence, we obtain $B^*, C^* \subseteq ||B \otimes C||$.

Therefore, $B^* \otimes C^* = (B^*, C^*)^{\perp\perp} \subseteq ||B \otimes C||^{\perp\perp} = ||B \otimes C||$ since $||B \otimes C||$ is a fact.

Case 8: $A = B \wp C$.
Let $\Gamma \in B^* \wp C^* = (B^{*\perp}, C^{*\perp})^\perp$, that is, if $\Delta^\perp \in B^{*\perp}$,$\Sigma^\perp \in C^{*\perp}$ then $\vdash \Gamma, \Delta^\perp, \Sigma^\perp \rightarrow$. By the induction hypothesis and Lemma 4.3.4, $B^\perp \in B^{*\perp}$ and $C^\perp \in C^{*\perp}$. Take $B^\perp$ as $\Delta^\perp$ and $C^\perp$ as $\Sigma^\perp$ above. We obtain $\vdash \Gamma, B^\perp, C^\perp \rightarrow$. We can deduce

$$\cfrac{\cfrac{\Gamma, B^\perp, C^\perp \rightarrow}{\Gamma \rightarrow B, C}}{\Gamma \rightarrow B \wp C} \ (\rightarrow \wp)$$

Threfore, $\Gamma \in ||B \wp C||$.

Case 9: $A = B\&C$.

Let $\Gamma \in B^*\&C^* = B^* \cap C^*$. By the induction hypothesis $B^* \subseteq ||B||$, $\Gamma \in ||B||$, that is, $\vdash \Gamma \to B$. Similarly, $\vdash \Gamma \to C$. Hence,

$$\frac{\Gamma \to B \quad \Gamma \to C}{\Gamma \to B\&C} \ (\to \&)$$

that is, $\Gamma \in ||B\&C||$.

Case 10: $A = B \oplus C$.

By the induction hypothesis, $B^* \subseteq ||B||$ and $C^* \subseteq ||C||$. Hence $B^* \cup C^* \subseteq ||B|| \cup ||C||$.

Now, let $\Gamma \in ||B||$. We can deduce

$$\frac{\Gamma \to B}{\Gamma \to B \oplus C} \ (\to \oplus)$$

that is, $\Gamma \in ||B \oplus C||$. Hence, $||B|| \subseteq ||B \oplus C||$. Similarly, we obtain $||C|| \subseteq ||B \oplus C||$. Hence, $||B|| \cup ||C|| \subseteq ||B \oplus C||$, that is, $B^* \cup C^* \subseteq ||B \oplus C||$ Therefore, $B^* \oplus C^* \subseteq ||B \oplus C||$ since $||B \oplus C||$ is a fact.

Case 11: $A = B \multimap C$.

Let $\Gamma \in B^* \multimap C^*$, that is, $\forall \Lambda \in B^*(\Gamma, \Lambda \in C^*)$. By the induction hypothesis, and hence by Corollary 4.3.1, we can take $B \in B^*$ as Λ. Hence $B, \Gamma \in C^*$. By the induction hypothesis, $C^* \subseteq ||C||$. Hence, $B, \Gamma \in ||C||$, that is, $\vdash B, \Gamma \to C$. One can deduce

$$\frac{B, \Gamma \to C}{\Gamma \to B \multimap C} \ (\to \multimap)$$

Therefore, $\Gamma \in ||B \multimap C||$.

Case 12: $A = \bigcirc B$.

Let $\Gamma \in h(||B||)$, that is, $\Gamma = \bigcirc\Gamma'$ and $\Gamma' \in ||B||$. Since $\vdash \Gamma' \to B$, one can deduce

$$\frac{\Gamma' \to B}{\bigcirc\Gamma' \to \bigcirc B} \ (\bigcirc)$$

that is, $\Gamma = \bigcirc\Gamma' \in ||\bigcirc B||$. Hence, $h(||B||) \subseteq ||\bigcirc B||$.

Now, by the induction hypothesis $B^* \subseteq ||B||$,

$$h(B^*) \subseteq h(||B||) \subseteq ||\bigcirc B||.$$

Therefore,

$$\bigcirc B^* = h(B^*)^{\perp\perp} \subseteq ||\bigcirc B||,$$

since $||\bigcirc B||$ is a fact.

Case 13: $A = \overline{\bigcirc} B$.

Let $\Gamma \in h(B^{*\perp})^{\perp}$, that is, for any $\Sigma \in h(B^{*\perp})$, $\vdash \Gamma, \Sigma \to \ $.

By the induction hypothesis and Lemma 4.3.4, $B^{\perp} \in B^{*\perp}$. Thus, $\bigcirc B^{\perp} = h(B^{\perp}) \in h(B^{*\perp})$. We can take $\bigcirc B$ as Σ above. Then, we can deduce

$$\frac{\dfrac{\Gamma, \bigcirc B^{\perp} \to}{\Gamma \to (\bigcirc B^{\perp})^{\perp}} \ (\to^{\perp}) \qquad \dfrac{\dfrac{\dfrac{B \to B}{\to B^{\perp}, B} \ (\to^{\perp})}{\to \bigcirc B^{\perp}, \overline{\bigcirc} B} \ (\bigcirc)}{(\bigcirc B^{\perp})^{\perp} \to \overline{\bigcirc} B} \ ({}^{\perp}\!\to)}{\Gamma \to \overline{\bigcirc} B} \ (cut)$$

Threfore, $\vdash \Gamma \to \overline{\bigcirc} B$, that is, $\Gamma \in ||\overline{\bigcirc} B||$.

Case 14: $A = \Box B$.

Let $\Gamma \in B^* \cap f(B^*)$. By the induction hypothesis, $B^* \subseteq ||B||$. Hence, $\Gamma \in ||B|| \cap f(||B||)$, that is, $\Gamma = !\Gamma_1, \Box\Gamma_2$ and $\vdash !\Gamma_1, \Box\Gamma_2, \rightarrow B$. Then, we can deduce

$$\frac{!\Gamma_1, \Box\Gamma_2, \rightarrow B}{!\Gamma_1, \Box\Gamma_2, \rightarrow \Box B} \; (\rightarrow \Box)$$

that is, $\Gamma = !\Gamma_1, \Box\Gamma_2 \in ||\Box B||$. Therefore, $\Box B^* \subseteq ||\Box B||$ since $||\Box B||$ is a fact.

Case 15: $A = \Diamond B$.

By the induction hypothesis and Lemma 4.3.4, $B^\perp \in B^{*\perp}$. Hence, $\Box B^\perp \in B^{*\perp}$ by Lemma 4.3.1. Since $\Box B^\perp = f(B^\perp) \in f(B^{*\perp})$,

$$\Box B^\perp \in B^{*\perp} \cap f(B^{*\perp}).$$

Now, let $\Gamma \in (B^{*\perp} \cap f(B^{*\perp}))^\perp$, that is, for any $\Sigma \in B^{*\perp} \cap f(B^{*\perp})$, $\vdash \Gamma, \Sigma \rightarrow$. We can take $\Box B^\perp$ as Σ. We can deduce

$$\dfrac{\dfrac{\Gamma, \Box B^\perp \rightarrow}{\Gamma, \rightarrow (\Box B^\perp)^\perp}\;(\rightarrow^\perp) \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \rightarrow B}{\rightarrow B^\perp, B}\;(\rightarrow^\perp)}{\rightarrow B^\perp, \Diamond B}\;(\rightarrow \Diamond)}{\rightarrow \Box B^\perp, \Diamond B}\;(\rightarrow \Box)}{(\Box B^\perp)^\perp \rightarrow \Diamond B}\;({}^\perp\!\!\rightarrow)}{\Gamma \rightarrow \Diamond B}\;(cut)$$

We obtain $\vdash \Gamma \rightarrow \Diamond B$, that is $\Gamma \in ||\Diamond B||$.

Case 16: $A = !B$.

Note that $!B^* = (B^* \cap K)^{\perp\perp}$ in our canonical model. Let $\Gamma \in B^* \cap K$. By the induction hypothesis, $B^* \subseteq ||B||$. Hence, $\Gamma \in ||B|| \cap K$, that is, $\Gamma = !\Gamma'$ and $\vdash !\Gamma' \rightarrow B$. Then

$$\frac{!\Gamma' \rightarrow B}{!\Gamma' \rightarrow !B} \; (\rightarrow !)$$

that is, $\Gamma = !\Gamma' \in ||!B||$. Therefore, $!B^* \subseteq ||!B||$ since $||!B||$ is a fact.

Case 17: $A = ?B$.

Note that $?B^* = (B^{*\perp} \cap K)^\perp$ in our canonical model.

By the induction hypothesis and Lemma 4.3.4, $B^\perp \in B^{*\perp}$. Hence, $!B^\perp \in B^{*\perp}$ by Lemma 4.3.1, that is,

$$!B^\perp \in B^{*\perp} \cap K.$$

Now, let $\Gamma \in (B^{*\perp} \cap K)^\perp$, that is, for any $\Sigma \in B^{*\perp} \cap K$, $\vdash \Gamma, \Sigma \rightarrow$. We can take $!B^\perp$ as Σ. We can deduce

$$\dfrac{\dfrac{\Gamma, !B^\perp \rightarrow}{\Gamma, \rightarrow (!B^\perp)^\perp}\;(\rightarrow^\perp) \qquad \dfrac{\dfrac{\dfrac{B \rightarrow B}{\rightarrow B^\perp, B}\;(\rightarrow^\perp)}{\rightarrow ?B^\perp, B}\;(\rightarrow ?)}{(!B^\perp)^\perp \rightarrow ?B}\;({}^\perp\!\!\rightarrow)}{\Gamma \rightarrow ?B}\;(cut)$$

We obtain $\vdash \Gamma \rightarrow ?B$, that is $\Gamma \in ||?B||$. (Q.E.D.)