Relating processes with real-time property.

Takashi Kitamura, Atsushi Togashi

Abstract

In this research, we consider some (bi)simulation-based relations of processes in timed process algebra(TPA), which reflect real-time property; we attempt to reflect real-time property into the discussion of relating processes in TPA. To be more precise, we relate functionally identical processes with respect to real-time property. We discuss the issue on rTeCCS, which is one of timed process algebras. As a result, we propose six kinds of speed, hence six relations of processes that reflect real-time property in TPA. We also present the some properties of these relations, including congruence property of relations. We show that these relations are not congruent for rTeCCS, but congruent for all the operators of kTeCCS, whose semantics is slightly diverse from that of rTeCCS for the sake of congruence. We also show an example for those relations.

Contents

Chapter 1

Introduction

Process algebra is widely studied as a mathematical model to reasoning about the behaviour of concurrent systems. Its tools are algebraic languages and it is useful in both specification and verification of concurrent systems. Most traditional and main stream of such approaches are Milner's CCS(Calculus of Communicating Systems)[Mil89][Mil99], CSP(Communicating Sequential Processes)[Hoa85][Hoa78] and ACP(Algebra of Communicating Processes)[BK85][Ber86]. In such traditional works, time properties are ignored for the sake of abstractions. And this abstraction view turns out to be a major contributing factor to the success of process algebra. In other words, this abstraction brings an elegant theory to a calculus of communicating systems.

On the other hand, it is a fact that time often plays an important role in describing many concurrent systems. There are a lot of scenes to be formalized or verified with time notion in real systems. Several researchers noticed the importance of time notion and over the years conducted researches to develop process algebras with time constraints, [MT90] [Wan90] [NS90] [Sch91] [HR95] etc, extending CCS, CSP, ACP. They are well-founded and gave fruitful results to timed systems.

But among those works, study on the discussion of relating processes under the time notion is missing. In most of the works, functional behaviour and temporal behaviour are treated thoroughly in the same way; two processes are related only if they are observed equivalent in both their functional and temporal behaviour by external observers. This may be one approach to relate processes in the stream of Milner's bisimulation relation. But apparently it is not an enough good strategy to inherit Milner's bisimulation notion, which is developed for the functional behaviour, and apply it to the temporal behaviour as it is. This is because functional and temporal properties are completely different in their character. Therefore we need to deliberate time properties well and design a new framework to reflect them into the discussion of relating processes.

Moller et al. first invented a new and practical idea of relating process in a TPA(Timed Process algebra) in [MT91]. They paid attention to the fact that the real-time property was the one of the most important properties in practical timed systems from engineering viewpoint. Therefore they attempted to reflect real-time property into relating processes in TPA and consequently they invented a new binary relation over processes, called "faster-than" relation, based on Milner's simulation relation. The faster-than relation of processes relates two processes with respect to speed; it provides an order relation of two processes, which relates two processes whose functional behaviours are identical but operate at different speed. The faster-than relation of processes is an order relation so that processes can be compared with respect to their speeds. This idea of relating way of processes reflects real-time property, because by relating processes, whose functional behaviours are equivalent, with speeds we can guarantee time constraint of action occurrence of processes. Hence we can guarantee a process to be certainly completed within a given time. It is an important result that the new notion of real-time property has been introduced into the discussion of relating processes in TPA. But Moller's faster-than relation reflect only one aspect of real-time property. With Moller's faster-than relation, we cannot acquire two processes to be in the relation which express the other aspects considered as a guarantee of real-time property. That is, this Moller's relation does not reflect a real-nature of real-time property. Also Moller's faster-than relation confine the processes to be related too strictly so that we often cannot relate two processes that have very similar temporal behaviour as well as completely identical functional behaviours. So we may well want to construct more general relation which is more general than the Moller's faster-than relation.

Recently, Luttegen et al. addressed to reflect real-time property into the relating processes on TPA in another way[LV01]. They developed a new timed language, called TACS(Timed Asynchronous Communicating Systems), with new treatment of time notion. In contrast to other TPAs, TACS treats time as maximum delays of action. This treatment of time is useful in some aspects in computer systems, because sometimes in practice we want to design a system with maximum delay. He proposed some new relations that reflects real-time property, relating processes with respect to speed. He set up start states and end states into LTS(Labelled Transition System) in a TPA as in automata, and compared the speed of two processes in a TPA by means of the time for action sequences from start state to end state of process to be performed. This is interesting in a respect that we are able to compare the speed of the action sequence from the start state to the end state of two processes and to know which of two processes has a faster action sequence. But still it is not based on the principle of process algebra to establish start states and end states in LTSs in process algebra. Hence we can say that this faster-than relation of processes is not basic to the principle idea of process algebra. What is more, such relations Luttgen proposed are useless in the presence of infinite process in process algebra and hence have to be confine into the finite process in process algebra.

In this paper, we consider more general and more basic relations between processes in TPA which take real-time property into consideration. That is, we introduce some relating systems of processes of TPA, whose functional behaviour is equivalent but operate at different speeds, and accordingly propose relations of processes which reflect real-time property. Moreover, the relations to be proposed are more general than Moller's relation and more basic than Luttgen's relation. We work on this matter on the calculus of lTeCCS[MT90], which is one of TPAs, proposed by Moller and Tofts. lTeCCS is a timed extension of CCS and the time notion in lTeCCS treated as a time delay. In considering inventing the relating systems of processes in lTeCCS, we first have to examine the LTS, which is a description of process, in lTeCCS. However how to construct a LTS in lTeCCS, whose transition is composed of the functional action transition and temporal clock transition, is various therefore rather complicated. But we regard the functional actions as a central role of the behaviours of processes and temporal behaviour as a temporal description for action execution. That is, we regard the time of lTeCCS as time delay which is imposed on action executions. Based on the idea, we acquire the some typical or normal form for LTSs of lTeCCS. Also, we carry out the work of relating processes on the basis of such LTSs. The relations to be proposed will be based on the traditional framework of relating processes in process algebra; the simulation relation. Real-time property designed to be reflected on such a LTS within the framework of simulation relation by manipulating newly-introduced time notion, namely temporal clock transitions. In this way we reflect real-time property into the relating processes within the framework of the notion of simulation relation so that relations to be proposed is unique to the LTS in lTeCCS and the notion of simulation relation. As a result, we will propose 6 kinds of relations of processes in lTeCCS, that reflect real-time property. All the relations are considered under the real-time property, but each expresses different aspects of real-time properties. Among these 6 relations, each relation is related with the other by descriptive. We also show that one of 6 relations is more general one than the equivalence relation, which is commonly introduced in the former researches on TPA, and Moller's faster-than relation. In other words, one relation of the 6 relations we propose in this paper is a general one of the common equivalence relation in most of the TPAs, meaning this relation include the equivalence relation of most of the TPAs, and another relation of them is a general one of Moller's faster-than relation, meaning this relation include the Moller's faster-than relation. After discussing these relations, we formalize these 6 relations in the manner of process algebra, modifying the basic simulation relation. We also provide properties of each of these relation, including congruence property, and the relation of these 6 relation.

Chapter 2

Preliminary

In this chapter, to work on the relating systems and relations on timed process algebra(TPA)s,we review ℓTeCCS(loose Temporal Calculus of Communicating System) and relations of TPAs suggested in preceding works, as a preliminary. Many works on timed process algebras are conducted, hence quite a few timed languages have been suggested already. ℓTeCCS is one of those timed languages proposed by Moller and Tofts [MT90][MT91]. It is an extension of CCS with time constraints, treating the time as a time delay. We work on the simple calculus of ℓTeCCS. Hence we first introduce the calculus ℓTeCCS of its syntax and semantics. Then to review the history of relations in TPAs, we will see a time-indepent and a time dependent relation in TPA already proposed in former researches. We also show some problems about these two relations, already proposed in the former works, are not well designed with the notion of time or real-time property.

2.1 The calculus: ℓTeCCS

In this section, we review the calculus ℓTeCCS by presenting the syntax and operational semantics of the language. A description is follows, however, readers are strongly recommended to read original papers [MT90] [MT91] to refer to the full treatment of the calculus. ℓTeCCS is a natural extension of CCS with notion of discrete time and has a conceptual similarity with a Timed CCS proposed by [Wan90] in its syntax and semantics.

2.1.1 Syntax

We first presuppose an infinite set Λ of actions and $\bar{\Lambda}$ of complementary actions. Then we take the set of atomic action symbols *Act* to be $\Lambda \cup \bar{\Lambda} \cup \{\tau\}$. An action communicates with its complement $\bar{a}$ to produce the internal action τ, as in CCS. And we assume $T = \{1, 2, 3, ..., \omega\}$ to denote division in discrete and continuous time. The syntax of the language, which is a subset of TeCCS(Temporal Calculus of Communicating Sysytems), is given by following BNF expressions.

Definition 1 *(Abstract Syntax)*

$$
\begin{array}{lcll}
P & ::= & 0 & \textit{(Nil)} \\
 & | & X & \textit{(Variable)} \\
 & | & \alpha\ .P & \textit{(Action prefix)} \\
 & | & (t)\ .P & \textit{(Clock prefix)} \\
 & | & P + Q & \textit{(Summation)} \\
 & | & P \mid Q & \textit{(Parallel composition)} \\
 & | & P \setminus L & \textit{(Restriction)} \\
 & | & P[S] & \textit{(Relabeling)} \\
 & | & \mu_i\ \tilde{X}.\tilde{P} & \textit{(Recursion)}
\end{array}
$$

where t denotes an element of T and X is a variable belonging to a countable infinite set Var of variables.

□

Here, we give intuition of these operators. First, 0 represents the nil process, which idles and cannot perform any action. X means the process bound to the variable X. $\alpha.P$ represents the process which tries to execute the action α immediately and evolves into the process P by doing so. This transition itself takes no time, i.e. action is atomic. And also we give a property $\alpha.P$ will wait forever until its environment becomes ready. The summation combinatory $P + Q$ is used to describe non-deterministic choices. The process will behave as the process P or the process Q, with the choice being made at the time of the first action. Thus, for instance an initial passage of time must be allowed by both P and Q. $P|Q$ represents the parallel composition of P and Q. This can be understood in the same way as the handshake and interleaving of CCS. But differences are on a treatment of time action, representing the passage of time. Passage of time is fair to all the processes composed of $|$. That is, time action is a broadcast event over $|$. $P \setminus L$, $P[S]$ and $\mu\tilde{X}_i.\tilde{P}$ respectively means restriction, relabeling and, recursive operator inherited from CCS.

2.1.2 Semantics

This section shows the operational semantics of *l*TeCCS. The operational semantics is given as transition based, structural operational rules in two parts: one for action transitions $\longrightarrow \subseteq P \times \mathit{Act} \times P$ and the other for clock transitions $\rightsquigarrow \subseteq P \times T \times P$, satisfying the rules in Table 2.1 and Table 2.2 respectively.

Table 2.1: Operational semantics for *l*TeCCS(action transition)

$$\frac{}{a.P \xrightarrow{a} P} \qquad \frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'}$$

$$\frac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'} \qquad \frac{P \xrightarrow{a} P'}{P|Q \xrightarrow{a} P'|Q}$$

$$\frac{Q \xrightarrow{a} Q'}{P|Q \xrightarrow{a} P|Q'} \qquad \frac{P \xrightarrow{a} P',\ Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

$$\frac{P \xrightarrow{a} P'}{P \setminus L \xrightarrow{a} P' \setminus L}\ (a, \bar{a} \notin L) \qquad \frac{P \xrightarrow{a} P'}{P[S] \xrightarrow{S(a)} P'[S]}$$

$$\frac{P_i\{\mu\,\tilde{X}.\tilde{P}/\tilde{X}\} \xrightarrow{a} P'}{\mu_i\tilde{X}.\tilde{P} \xrightarrow{a} P'}$$

Here we introduce useful sets $Sort_t(P)$, which indicates the set of all the actions enabled in the first step within a given time interval t that P can execute, as shown in Table 2.3 [Wan90]. This is useful to express the "maximum progress" property [NS91], which ensures that whenever process can perform τ action, it blocks the progress of time and enforce

Table 2.2: Operational semantics for lTeCCS(clock transition)

$$\frac{}{0 \overset{t}{\rightsquigarrow} 0} \qquad \frac{}{a.P \overset{t}{\rightsquigarrow} a.P}$$

$$\frac{}{(s+t).P \overset{s}{\rightsquigarrow} (t).P} \qquad \frac{}{(t).P \overset{t}{\rightsquigarrow} P}$$

$$\frac{P \overset{s}{\rightsquigarrow} P'}{(t).P \overset{s+t}{\rightsquigarrow} P'} \qquad \frac{P \overset{t}{\rightsquigarrow} P',\ Q \overset{t}{\rightsquigarrow} Q'}{P+Q \overset{t}{\rightsquigarrow} P'+Q'}$$

$$\frac{P \overset{t}{\rightsquigarrow} P',\ Q \overset{t}{\rightsquigarrow} Q'}{P|Q \overset{t}{\rightsquigarrow} P'|Q'} \quad Sort_t(P) \cap \overline{Sort_t(Q)} = \emptyset$$

$$\frac{P \overset{t}{\rightsquigarrow} P'}{P \setminus L \overset{t}{\rightsquigarrow} P' \setminus L} \qquad \frac{P \overset{t}{\rightsquigarrow} P'}{P[S] \overset{t}{\rightsquigarrow} P'[S]}$$

$$\frac{P_i\{\mu\, \tilde{X}.\tilde{P}/\tilde{X}\} \overset{t}{\rightsquigarrow} P'}{\mu_i \tilde{X}.\tilde{P} \overset{t}{\rightsquigarrow} P'}$$

the execution of an action immediately before some delay; a property that two processes communicate with each other as soon as they are ready to do so. Formally,

$$\forall P, P', t, Q : P \xrightarrow{\tau} P' \Rightarrow P \overset{t}{\not\rightsquigarrow} Q$$

So the side condition $Sort_t(P) \cap \overline{Sort_t(Q)} = \emptyset$ in Table2.1 guarantees that there is no communication within t time units between processes P and Q, thus expression at all means both processes consent to pass time interval t.

Another important property that lTeCCS possesses is time-determinancy. This property ensures that when a process idles for some duration t, then resulting behaviour is completely determined from P and t. Namely, the progress of time should be deterministic, expressed by

Table 2.3: Timed sort for processes

$Sort_t(Nil)=\emptyset$	$Sort_t((t+u).P)=\emptyset$
$Sort_t(a.P)=\{a\}$	$Sort_{t+u}((t).P)=Sort_u(P)$
$Sort_t(\tau.P)=\emptyset$	$Sort_t((t).P)=\emptyset$
$Sort_t(P+Q)=Sort_t(P)\cup Sort_t(Q)$	
$Sort_t(X)=Sort_t(P)\ whenX \overset{def}{=} P$	

$$\forall P,P',P'',t:\ P \overset{t}{\rightsquigarrow} P' \wedge P \overset{t}{\rightsquigarrow} P'' \ \Rightarrow P'\equiv P''$$

where $\equiv$ is the syntactic equality. The treatment of time passage action of both + and | operator is a device that ensures this property; it ensures this time-determinacy property that "+" is not decided by the time passage action and time passage action is broadcasted to all the processes over "|".

2.2 LTS(Labelled Transition System) and relations of processes in TPA

In this section, we review the history of relating of processes in TPAs. Although many works on timed process algebra are conducted, most of them are proposing a new calculus with various time constraints to express various aspects of time and a few works conducted relating processes.

First, before we review the relations of TPA, we see the bisimulation relation as an established relation in process algebra. We also see LTS(Labelled Transition System), which defines process behaviour, therefore is bais of defining simulation relation. We see what is LTS of process algebra of lTeCCS like to be. After that, for considering how we can reflect real-time property into the relating process in timed process algebra, we see how is LTS of timed process algebra like to be. Then we see the two relations of TPA which have already proposed in former works. One of them, is a relation that inherit Milner's bisimulation relation and apply it to that of TPA as it is. And the other is a relation which reflect real-time property, therefore relates processes whose behaviour are identical, but operate at different speed. His idea is good enough to relate processes with respect to speeds, but we state that his idea is not an essence of real-time property and demonstrate that it is only an aspect of real-time property.

2.2.1 LTS and relations of process algebra

In this section, we review the LTS and the bisimulation relation of the most basic process algebra: CCS.

In process algebra, the discussion of relating processes is one of the main issues in a calculus. Processes are related equivalent with the notion of 'simulation relation', therefore equivalence notion defined with the notion of simulation relation are called "bisimulation equivalence", which is invented by Park and Milner. The substance of each process in process algebra is a LTS, which has no start state and also no accepting state. But each process is a black box to an external observer. Its behaviour is known by interacting with it in some sequences; We identify each process with its behaviour by interacting with the process. Therefore two processes regarded as equivalent if they cannot be distinguished

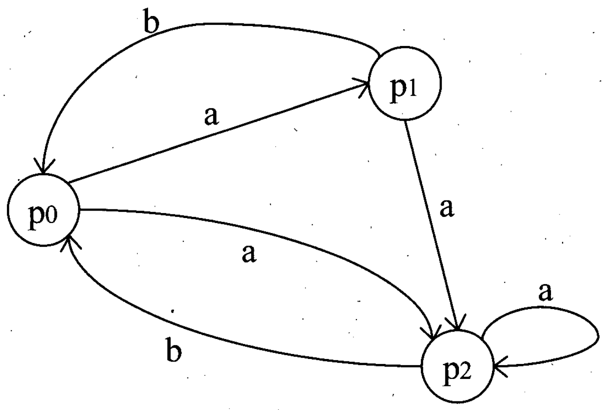

Figure 2.1: Labelled transition system

by external observers interacting with them. In other words, if the behaviour of two processes are completely identical from external observers, they are regarded as equivalent. And this equivalence notion of processes is defined in terms of the notion of simulation. So, a bisimulation relation is defined if one of two processes completely simulates the behaviour of the other process. More precisely, it defines behaviour of one process term are all simulated by the other process and vice-versa, then two processes are regarded to be bisimulation equivalence. Therefore, the resulting relation is called "bisimulation relation". The following is the definition of bisimulation relation.

Definition 2 (Strong bisimulation, strong equivalence) *A binary relation $\mathcal{S} \subseteq P \times P$ over processes is a bisimulation if, for all $\langle P, Q\rangle \in \mathcal{S}$ and for all $a \in Act$,*
(1) $if\ P \xrightarrow{a} P'\ then\ \exists Q' : Q \xrightarrow{a} Q'\ and\ \langle P', Q'\rangle \in \mathcal{S}$;
(2) $if\ Q \xrightarrow{a} Q'\ then\ \exists P' : P \xrightarrow{a} P'\ and\ \langle P', Q'\rangle \in \mathcal{S}$;
We say P and Q are strongly equivalent, written $P \sim Q$, if there exits a strong bisimulation $\mathcal{S}$ such that $\langle P, Q\rangle \in \mathcal{S}$. □

In the definition of bisimulation relation, first relation $\mathcal{S}$ which is a binary relation over states of process is defined. Then to verify this S is a simulation relation, for every pair $(P, Q) \in S$, we have to consider each transition $P \xrightarrow{\alpha} Q'$ of the second member, and show that is properly matched by some transition $P \xrightarrow{\alpha} P'$ of the first member Q. And if S and its converse $\mathcal{S}^{-\infty}$ are simulations, $\mathcal{S}$ is said to be a strong bisimulation.

Here we take an example demonstrated in Milner's book[Mil99]. Consider the following LTS in figure.2.2.

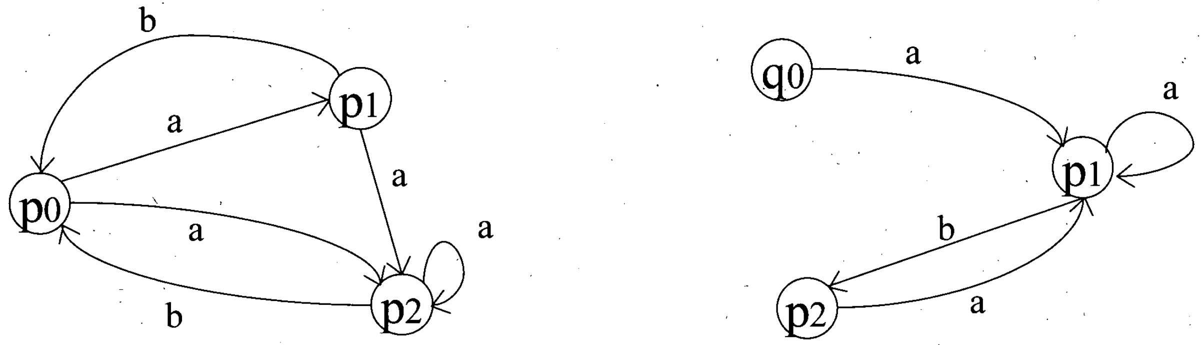

Figure 2.2:

Then $P_0 \sim Q_0$. To prove this, we define

$$\mathcal{S} = \{(P_0, Q_0), (P_0, Q_2), (P_1, Q_1), (P_2, Q_1)\};$$

then we show that $\mathcal{S}$ is a bisimulation, and this is enough because $P_0 \mathcal{S} Q_0$.

It often helps to show a bisimulation graphically, by linking the related states on a transition graph. Take above figure2.2 for example,it looks like figure.2.3

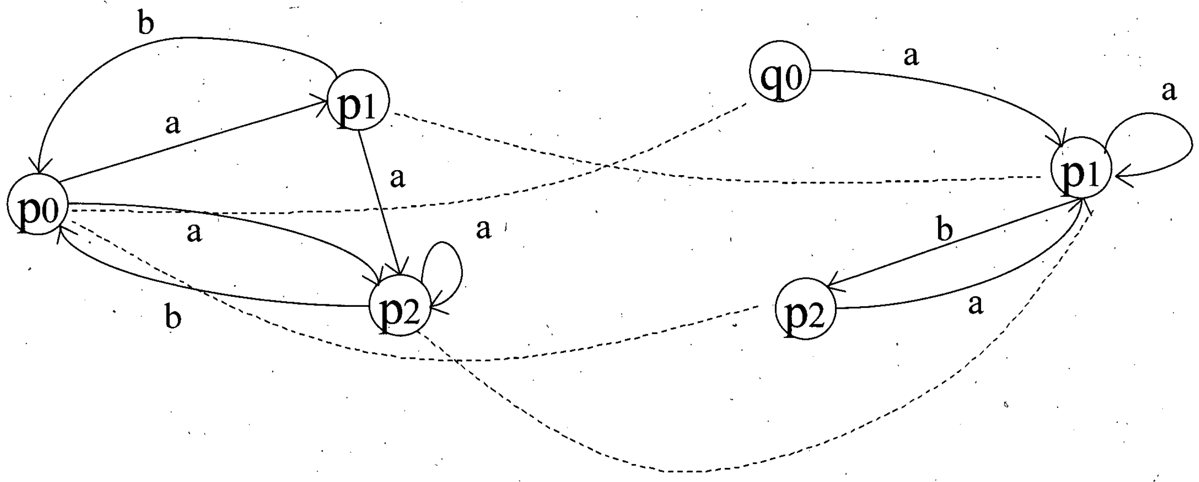

Figure 2.3: bisimulation graphically

It is obvious from the definition that the bisimulation equivalence, $\sim$, is an equivalence relation.

Proposition 1 *1. The binary relation $\sim$ is an equivalence relation over the CCS processes, i.e. followings hold.*

$P \sim P$ (reflexivity)

$P_1 \sim P_2$ then $P_2 \sim P_1$ (symmetry)

$P_1 \sim P_2$ and $P_2 \sim P_3$ imply $P_1 \sim P_3$ (transitivity)

2. $\sim$ is itself a strong bisimulation.

□

Bisimulation relation is a very fine equivalent notion, reflecting the fact that in concurrent systems the description of process behaviour is more important than what traces or words processes accept as in Automata.

2.2.2 Relations of processes in TPA

In this part, we will review the two relations of timed processes, that have been proposed in preceding researches on TPAs. One of them, which is proposed in the most TPAs, is a normal bisimulation equivalence relation. It inherited Milner's bisimulation relation and applied it to the relating system of process of TPA as it is. And the other is an Mollers order relation which relates processes with respect to speed. His attempt is new and interesting in a respect that he reflect real-time property into the relating processe by relating processes whose functional behaviour is identical but operate at different speed. Therefore his relation with a notion of speed are called as "faster-than relation" But we also show the problem of about both equivalence relation in TPAs and Moller's faster-than relation.

Strong bisimulation and strong equivalence in lTeCCS

At first, we introduce the most common relation proposed in the most TPAs. In most TPAs, the discussions of relating processes are straightforward. The relation in the works is a what is called "bisimulation equivalence relation" in lTeCCS. It inherit Milner's bisimulation relation and apply it to the relating system of process of TPA as it is. That is, functional behaviour and temporal behaviour are treated thoroughly in the same way; two processes are related if only if they are observed equivalent in both their functional and temporal behaviour by external observers. Thus the derived relation is a simple bisimulation equivalence. This is expressed in the following definitions:

Definition 3 (Strong bisimulation and strong equivalence in lTeCCS)
A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong exact-time bisimulation if, for all $\langle P, Q\rangle \in \mathcal{S}$ and for all $a \in Act$ and for all $t \in T$,

(1) *if* $P \xrightarrow{a} P'$ *then* $\exists Q' : Q \xrightarrow{a} Q'$ *and* $\langle P', Q'\rangle \in \mathcal{S}$;
(2) *if* $Q \xrightarrow{a} Q'$ *then* $\exists P' : P \xrightarrow{a} P'$ *and* $\langle P', Q'\rangle \in \mathcal{S}$;
(3) *if* $P \overset{t}{\rightsquigarrow} P'$ *then* $\exists Q' : Q \overset{t}{\rightsquigarrow} Q'$ *and* $\langle P', Q'\rangle \in \mathcal{S}$;
(4) *if* $Q \overset{t}{\rightsquigarrow} Q'$ *then* $\exists P' : P \overset{t}{\rightsquigarrow} P'$ *and* $\langle P', Q'\rangle \in \mathcal{S}$. □

Definition 4 *We say that P and Q are strongly equivalent, written $P \sim Q$, if $\langle P, Q\rangle \in \mathcal{S}$ for some strong bisimulation $\mathcal{S}$.* □

Here are some properties of strong equivalence $\sim$ on lTeCCS$\sim$.

Proposition 2

1. *The binary relation $\sim$ is an equivalence relation over the lTeCCS processes, i.e. followings hold.*
   (1) $P \sim P$ (reflexivity)
   (2) $P_1 \sim P_2$ then $P_2 \sim P_1$ (symmetry)
   (3) $P_1 \sim P_2$ and $P_2 \sim P_3$ imply $P_1 \sim P_3$ (transitivity)

2. $\sim \; = \; \bigcup\{ \mathcal{S} \mid \mathcal{S}$ *is a strong bisimulation on lTeCCS* $\}$

□

It is important to prove that the relation is a congruence. This is because it means that if $P \sim Q$ then, in any system with our constructions, we can substitute the process P by the other process Q without altering the behaviour of the system. This property does indeed hold. Here we state that strong equivalence is a congruence.

Theorem 1 (Strong equivalence is a congruence)
$\sim$ is a process congruence with respect to the operators of lTeCCS; that is, if $P \sim Q$ then

1. $a.P \sim a.Q$
2. $P + M \sim Q + M$
3. $P \mid Q \sim Q \mid M$
4. $P[S] \sim Q[S]$
5. $(t).P \sim (t).Q$
6. $P\backslash L \sim Q\backslash L$

□

The idea of bisimulation equivalence relation in lTeCCS is straightforward, moreover poor. Bisimulation equivalence relation on lTeCCS inherits Milner's bisimulation relation, which is developed for the functional behaviour, and applies it to the temporal behaviour as it is. Namely it treats the functional and temporal behaviour of processes thoroughly in the same way. This may be one approach in the stream of Milner's bisimulation relation. But apparently it is not a good enough strategy to treat temporal behaviour in the same way as functional behaviour, because functional and temporal properties are completely different in their character. It is natural that the time notion is newly introduced into the system so that we should consider the characteristic of the newly-introduced time notion and reflect it to the relating systems of processes.

Moller's faster-than relation

Then, what is time property in timed systems? What is the characteristic property of time notion? In considering practical timed systems, one of the most important and characteristic property of time is real-time property. Real-time property is a property that guarantees for some action or procedure to be surely completed within a given time interval. This property plays an critical role in practical timed systems. Therefore it is an interesting idea to introduce this real-time property into the discussion of relating processes in TPA.

Moller et. al. first introduce real-time property into relating processes [MT91]. They paid attention to the real-time property in timed systems and attempted to reflect real-time property into relating processes in TPA. Consequently they invented a new binary relation over processes, called "faster-than" relation, which relates processes with respect to speed, based on Milner's simulation relation. It provides an order relation of two processes, which relates two processes whose functional behaviour are identical but operate at different speed. In more detail, with faster-than relation, processes are related if their functional behaviour are equivalent and one process can execute its function actions faster than the other process. The faster-than relation of processes is an ordered relation so that processes can be compared with respect to their speeds.

This is defined as follows:

Definition 5 *(Strong Moller's faster-than bisimulation, Strong Moller's faster-than relation)*
A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong faster-than bisimulation if, for all $\langle P, Q\rangle \in \mathcal{S}$ and for all $a \in Act$ and for all $t \in T$,

(1) *if* $P \xrightarrow{a} P'$ *then* $\exists m, \exists Q', \exists Q'', \exists P'' : Q \overset{s}{\rightsquigarrow} Q' \xrightarrow{a} Q''$ *and* $P' \overset{s}{\rightsquigarrow} P''$ *with* $\langle P'', Q''\rangle \in \mathcal{S}$;

(2) *if* $Q \xrightarrow{a} Q'$ *then* $\exists P' : P \xrightarrow{a} P'$ *and* $\langle P', Q'\rangle \in \mathcal{S}$;

(3) *if* $P \overset{t}{\rightsquigarrow} P'$ *then* $\exists Q' : Q \overset{t}{\rightsquigarrow} Q'$ *and* $\langle P', Q'\rangle \in \mathcal{S}$;

(4) *if* $Q \overset{t}{\rightsquigarrow} Q'$ *then* $\exists P' : P \overset{t}{\rightsquigarrow} P'$ *and* $\langle P', Q'\rangle \in \mathcal{S}$. □

Definition 6 *We say that P and Q are Moller's faster-than relation, written* $P \overset{M}{\lesssim} Q$, *if* $\langle P, Q\rangle \in \mathcal{S}$ *for some strong faster-than bisimulation* $\mathcal{S}$. □

The characteristic part which expresses "faster-than" property of this definition appears in the first clause. If the first (faster) process term can perform a particular action, then the second (slower) process term can either perform that action right away and evolve into a new process state or else it can idle for some time s and reach a state in which it can perform the action and thus in doing so evolve into a new process state. In the first case that slower process term evolve into new process state with no delay, the new process state, which it evolved by performing a particular action immediately with no delay is also slower than that into which the first process evolved. In the second case that slower process term evolve into the new state by performing a particular action with some time interval s, then while it is necessarily not slower than that into which the first process evolved, but slower than that state after waiting time which faster process missed out.

As an example, we would clearly want

$$a \mid (1)b \ \lesssim \ (1)a \mid (1)b$$

Now in the faster term, the action transition

$$a \mid (1)b \xrightarrow{a} 0 \mid (1)b$$

is matched in the slower term by the sequence of transitions

$$(1)a \mid (1)b \overset{1}{\rightsquigarrow} a \mid b \xrightarrow{a} 0 \mid b$$

and while $0 \mid (1)b \not\lesssim^{M} 0 \mid b$, we only require in the definition after the clock transition $0 \mid (1)b \overset{1}{\rightsquigarrow} 0 \mid b$, and $0 \mid b \lesssim^{M} 0 \mid b$.

In this way, Moller invented a relation which relate processes with respect to speed. We can see this relation of processes exactly reflects real-time property. We can guarantee between two processes in the relation of Moller's faster-than relation the faster process are certainly ensured to perform all actions faster than the slower process does. That is, we can guarantee the behaviour a process is faster than a criterion process. This faster-than relation offers more important significance if it has a congruence or substitutive property. This is shown in the following proposition:

This idea of relating way of processes reflects real-time property, because by relating processes, whose functional behaviour are equivalent, with speeds we can guarantee time constraint of action occurrence of processes. Hence we can guarantee a process to be certainly completed within a given time. It is an important result that the new notion of real-time property has been introduced into the discussion of relating processes in TPA.

Furthermore this relation is shown to be a precongruence over lTeCCS terms which do not have any parallel operators within the scope of a recursion. Proof are given in [MT91].

Proposition 3 $\overset{<M}{\sim}$ *is a congruent with respect to the operators of lTeCCS except for parallel operators within the scope of a recursion.*
□

Problems from Moller's faster-than relation

But Moller's faster-than relation reflects only one aspect of real-time property. That is, although we can admit that Moller's faster-than relation reflect real-time property well, we can see that this relation expresses nothing more than one aspect of real-time property. In the definitions, each action of faster process occurs faster than that of slower process, however later on faster process are obliged to wait until corresponding action of slower process finishes. This means that the faster process are obliged to wait the same time as it preempted the other process by the next action occurrence. For example, consider the following process term;

$$P_1 = a.(2).P_1', \qquad P_1' = b.0$$

$$Q_1 = (2).a.Q_1', \qquad Q_1' = b.0$$

And we define $\mathcal{S}_1$ as follows:

$$\mathcal{S}_\infty = \{(P_1, Q_1), (P_1', Q_1')\}$$

Obviously, we can get process $\mathcal{S}_1$ is a "faster-than" relation above, $P_1 \overset{<M}{\sim} Q_2$, because P_1 executes the first action a faster than Q_1 by two units of time and wait the same clock it preempted by. Here we can find that this guarantees more than the fact process P_1 is faster than Q_1. That is, it guarantees not only process P is faster than Q but also the fact that their necessary total time for action occurrence. But it is not always necessary to wait slower process from real-time property viewpoint. Furthermore, in reflecting the real-time property, it is much more natural and convincing not to wait slower processes. It is because many systems have a property of "the faster, the better". For instance, consider the processes;

$$P_2 = a.P_2', \qquad P_2' = b.0$$

$$Q_2 = (1).a.Q'_2, \qquad Q'_2 = b.0$$

And we define $\mathcal{S}_2$ as follows:

$$\mathcal{S}_2 = \{(P_1, Q_1), (P'_1, Q'_1)\}$$

Intuitively, process P_2 is faster than the other process Q_2. However we unfortunately cannot consider such two processes P_2 and Q_2 are related from Moller's faster-than relation, because process P_2 does not wait the slower process Q_2 after an action.

Also we can reflect real-time property into relating processes in TPAs in another way. We can realize to reflect real-time property, by guaranteeing two processes which operate at the same speeds. As mentioned already, real-time property is a property that guarantees for actions or procedures to be certainly completed within a given time interval. And the notion of same speed can guarantee that actions or procedures are certainly completed within a given time interval and accordingly can ensure real-time property. Moreover, in some real-time system we need not to guarantee the "the faster" notion, but the "the same speed" or "synchronous" notion, this means occasionally critical in timed systems.In order to reflect real-time property, we do not always have to attribute it to a problem of faster or not; relations which reflect real-time property do not necessarily have to be an order relation of speed. Thus, If we can relate two processes whose functional behaviour are identical and operate at "the same speed", then this can be also considered as a relation with real-time property. For example, consider the two processes and relation $\mathcal{S}_3$:

$$P_3 = a.(1).P'_3, \quad P'_3 = (2).b.P''_3, \quad P''_3 = c.0$$

$$Q_3 = (1).a.Q'_3, \quad Q'_3 = b.(2).Q''_3, \quad Q''_3 = c.0$$

$$\mathcal{S}_3 = \{(P_3, Q_3), (P'_3, Q'_3), (P''_3, Q''_3)\}$$

This is an example of two processes that operate at the same speed. P_3 performs an action a faster than Q_3, but in the second action, Q'_3 operates faster than P'_3. And for every action of two processes, the faster process waits the slower process after executing an action. Thus we can say these two processes operate at the same speed so that we can guarantee between two processes a process are certainly ensured to perform each action at the same speed as the other. As a result, although we do not tell which is the faster among these two processes, however, we can see these processes operate at the same speed and hence in this way time is guaranteed. We can see this is a reflection of real-time property.

Another problem from Moller's relation is about the generalization of relation. Moller's faster-than relation confines the processes to be related too strictly. That is, Moller's faster-than relation lacks the generalization of relation. For example, consider the two processes,

$$P_4 = (2).a.(1).P_4', \quad P_4' = (2).b.(2).Q_4'', \quad P_4'' = (2)c.0$$

$$Q_4 = (3).a.Q_4', \quad Q_4' = (4).b.P_4'', \quad Q_4'' = (2).c.0$$

To check $P_4 \overset{<M}{\sim} Q_4$, we define $\mathcal{S}_4$ as follows;

$$\mathcal{S}_4 = \{(P_4, P_4), (P_4', P_4'), (P_4'', P_4'')\}$$

This relation $\mathcal{S}_4$ intuitively expresses the same speed, because time to be taken for every action is same. But according to the definition of Moller's faster-than relation, we cannot acquire the relation $\mathcal{S}_4$ is Moller's faster-than relation. So we may well want to construct more general relation which is more general than the Moller's faster-than relation.

2.3 Summary

We have briefly reviewed lTeCCS, which is one of TPAs, and the two relations on TPAs proposed in the former researches. One of two relations is the equivalence relation, which inherited Milner's bisimulation relation and applied it to that of TPA as it is. The other is Moller's faster-than relation, which reflect real-time property and relates processes with respect to speeds. And consequently proposed order relation of speed "faster-than relation". Moller's faster-than relation is interesting in a respect that he expressed real-time property, which plays an important role in timed systems. But we found some problems from both equivalence relation and Moller's relation. Equivalence relation is not practical relation in a respect that time property is ignored. Moller's faster-than relation is interesting because it expresses real-time property. But Moller's relation does not fully express the real-time property. That is, this Moller's relation does not reflect a real-nature of real-time property. Also we saw the problem about generalization of Moller's relation; the relation is too strict so that two processes who have analogous temporal behaviour can not be regarded as in the relation. In next chapter, we will try to solve the probems concerning to these relations; we pursue the real nature of real-time property and introduce it into the relating processes in TPA and propose general relations which fully reflect real-time property.

Chapter 3

Design Choices for Relations of Processes with Real-time Property

In this chapter, we attempt to reflect real-time property, which plays an important role in timed systems, into the discussion of relating processes in TPA. Here we reflect real-time property into relating processes in TPA as relating processes, which are identical in functional behaviour but operate at different speed, as Moller did. But we prefer to state something more general. As we saw in the second chapter, Moller's faster-than relation is one aspect of real-time property. We can propose some more relations with various speeds. To begin with, to achieve to reflect the real-time property into the scene of relating processes in TPAs, arranging various speeds which could be proposed in the frame work of bisimulation relation. That is, we arrange and classify various speeds with peculiar property to simulation relation. In such a way, we design real-time property in the framework of bisimulation relation of process algebra. In this way, we show that there are six relations which respect to real-time poperty in the scene of relating processes in TPA. We can see that each of exact bisimulation and Moller's faster-than relation, which are shown in former chapter, is one aspect of the result of reflecting real-time property into the scene of relating processes. Also we see the characteristics of each of relations, including congruence, and also relation of those real-time-reflected relation. We formalize some relation of processes which reflects a real-time property and discuss about the properties or character of these relations.

3.1 LTS(labelled transition system) and relations of TPA(Timed Process Algebra)

In this section, we introduce the LTS in lTeCCS. Bisimulation relation is defined on the basis of LTS so that it is important to see the LTS, which is the description of a process in lTeCCS.

However when we consider constructing a LTS of lTeCCS, we have to deliberate how to construct it. This is because the transitions of lTeCCS are composed of the functional action transition and temporal clock transition. This makes LTS of lTeCCS complicated. But when we consider the LTS of lTeCCS, which is one of process algebra, we have to pay attention to the fact that process algebra is a mathematical model to reasoning about the behaviour of concurrent systems. We insist that the functional actions of a process are main behaviour and temporal transitions are supplement description to describe temporal behaviour of processes. And we define the LTS of lTeCCS.

Definition 7 (Labelled transition system(LTS) of TPA) *A labelled transition system(LTS) over Act and Time is pair $(Q, Trans)$ consisting of*

- *a set of Q of states;*
- *a ternary relation $Trans \subseteq (Q \times Act \times Q \ \bigcup \ Q \times Time \times Act \times Time \times Q \ \bigcup \ Q \times Act \times Time \times Q \ \bigcup \ Q \times Time \times Act \times Q \ \bigcup \ Q \times Time \times Q\ )$, known as a transition relation.*

□

Definition 8 (Labelled transition system(LTS) of TPA) *A labelled transition system(LTS) over Act and Time is pair $(Q, Trans)$ consisting of*

- *a set of Q of states;*
- *a ternary relation $Trans \subseteq (Q \times Time \times Act \times Time \times Q \ \bigcup \ Q \times Time \times Q\ )$, known as a transition relation.* □

An LTS of lTeCCS can be thought of as an LTS of untimed CCS with timed transition around an action transition. That is, how to transit from a state to another, there are five kinds of transition as shown in the figure.3.1. As we see in the figure, the main behaviour of transition is a functional action transition and it accompanies temporal clock transition before and after the action transition. These clock transition can be thought as a time delay imposed on the action execution; we consider the time transition before action transition as a time delay for preparation of the action execution, and the time transition after the action transition as time delay for post disposition or a garbage collection of the action execution. Although we set an action transition as a main behaviour of process in transitions, we also take a transition which is composed of temporal clock transition only. Also as usual in process algebra this LTS do not have a start state or accepting sates.

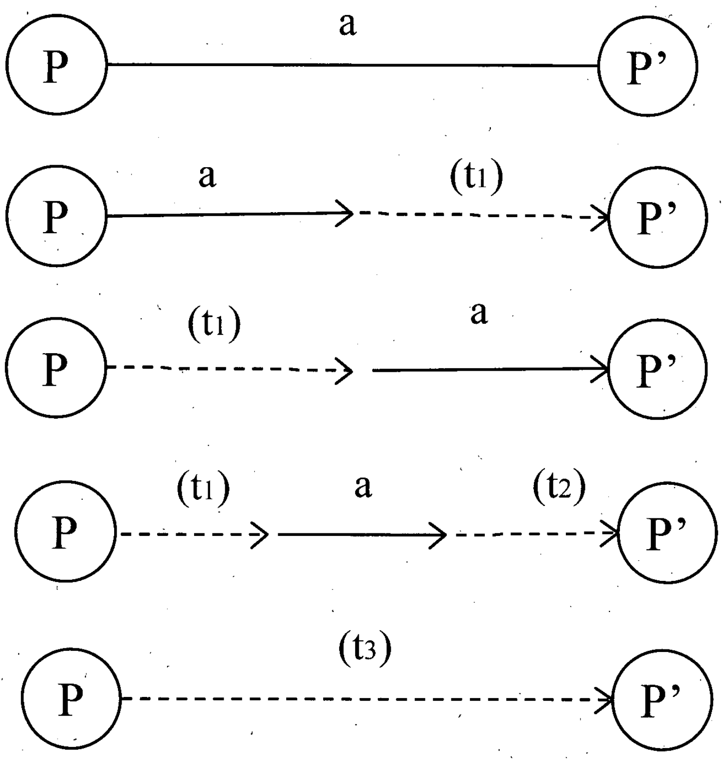

Figure 3.1: One transition in timed process algebra

An image of an LTS of *l*TeCCS is like figure3.2;

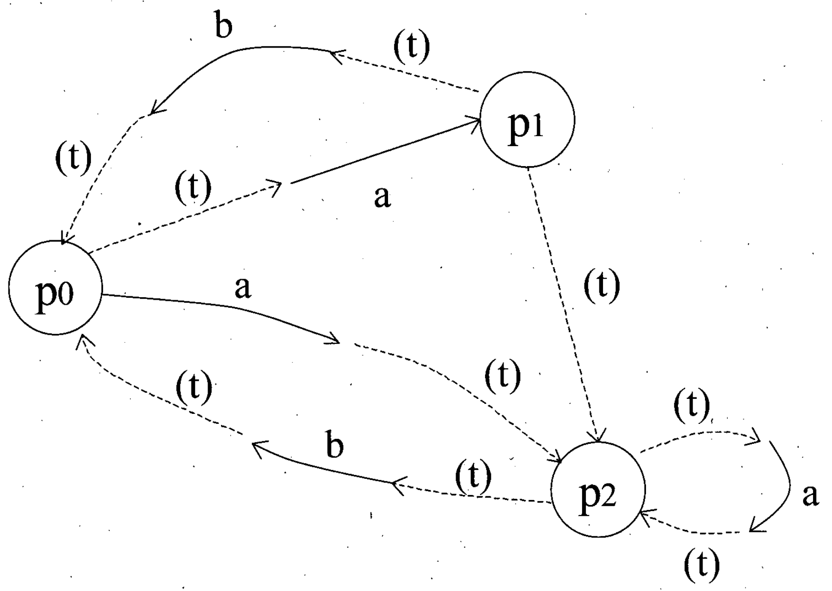

Figure 3.2: Timed Labelled transition system

3.2 How to Design Real-time Property into relation of *l*TeCCS

In this section, we examine various speeds that a process operate at in the framework of LTS and the notion of simulation relation in TPA. And we propose six relations of speed relation which reflect real-time property. To begin with, we review the simulation relation of TPA, thus LTS of *l*TeCCS. We consider how we can reflect time property in relating systems and how we can design speeds which is peculiar for LTS of *l*TeCCS and simulation relation on the basis of simulation relation of process algebra,. In such a way, We design real-time property within the framework of simulation relation of process algebra and propose six relations which all respect real-time property in the scene of relating processes in TPA. We will see that each of bisimulation equivalence and Moller's faster-than relation shown in the former chapter, is one aspect of the result of reflecting real-time property into the scene of relating processes.

In process algebra, processes are related as equivalent with the notion of 'simulation relation', therefore equivalence notion defined with the notion of simulation relation are called "bisimulation equivalence", which is invented by Park and Milner. In the definition of bisimulation relation, at first relation $\mathcal{S}$, which is a binary relation over states of process, is defined. Then to verify this $\mathcal{S}$ is a simulation relation, for every pair $(p, q) \in \mathcal{S}$ we have to consider each transition $q \xrightarrow{\alpha} q'$ of the second member, and show that is properly matched by some transition $p \xrightarrow{\alpha} p'$ of the first member p. And if S and its converse are simulations, $\mathcal{S}$ is said to be a strong bisimulation. That is, after define relation S over state of process, we check if a each transition from a state of pair is matched by the other in every pair in $\mathcal{S}$. The bisimulation relation is a very fine equivalent notion, reflecting the fact that in concurrent systems the description of process behaviour or reaction is regarded as important.

In considering relating processes in lTeCCS, which is one of process algebra, therefore it is natural that relations to be proposed should be based on the simulation relation. That is, we inherit the Milner's simulation relation, which is fine notion for concurrent processes, and construct the relating systems of processes within the framework of simulation relation. So by manipulating the definition of simulation relation, we can fully reflect real-time property into the relating processes. That is, the real-time property are reflected in the definition of simulation of processes. In LTS of lTeCCS, each transition of lTeCCS processes is one of five forms in the figure.3.1. Thus it is important to consider how each transition of one process simulates each transition of the other process. The point in the definition is how to treat newly-introduced time transitions. In other words, we have to deliberate how to treat clock transition(temporal behaviour) to reflect the real-time property within the simulation relation.

In such a transition like this, we found two places which can express the speed from the view point of both the engineering and the character of LTS viewpoint. Hence we can classify the notion of speed into two kinds of speeds. One kind of speeds is one based on the timing of the occurrence of action execution. This is the time required for the action to occur or fire, indicated by the time t_1 in the figure of fig.3.1. If the time required for an action to occur or fire (t_1 in fig.3.1) is shorter(faster) than that of the other process, then we can regard that it is faster in a respect of the timing of an action occurrence. The other kind of speeds is one based on the total time required for action execution to be completed, indicated by the time $t_1 + t_2$ in the fig.3.1. If the time required for an action execution to be completed is shorter(faster) than that of the other process, then we can regard that it is faster in a respect of the total time of action execution. Generally in LTS, which is not necessarily that of process algebra, it is important to guarantee the total time required for one transition when we consider the timed LTS. It is because what interact with a LTS is a something sequencial. By guaranteeing the total time required for each transitions, we can guarantee the total time required for a sequence of transitions. Process algebra is not an exception of those LTS either so that it is important to compare the total time required for a transition in LTS of TPA.

In this way, we measure the speeds of processes with two kinds of speeds:the time required for an action to occur(t_1) and the total time required for an action execution to be completed($t_1 + t_2$). Therefore we classify processes with the following combinations of these two kinds of speeds.

1. The time required for action execution to occur (t_1 in fig.3.1) is shorter or equal.

2. The time required for whole action execution ($t_1 + t_2$ in fig.3.1) is shorter or equal.

Here we strictly distinguish the notion of the "faster" transition expressed by required time is shorter, and the "synchronous" transition expressed by required time is equal. In real-time systems, we more often than not come across the situations where the notion of "faster" and the notion of "synchronous" is critical therefore have to be distinguished.

With the combination of two kinds of speeds (faster or synchronous), and two places to reflect them (the time required for an action to occur (t_1 in the fig.3.1) or the time required for an action execution to be completed (t_1+t_2) in the fig.3.1)), we can arrange six kinds of speeds of process or LTS in *l*TeCCS. These six relations all guarantee the speed therefore reflect real-time property. We explain these six variations of speed or relation in turn.

The variation of relation of two processes with speeds

1. The total time required for an action execution of the faster process to be completed ($t_1 + t_2$) is equal to that of the other(slower) one ($t'_1 + t'_2$). Also the timing of action occurrence of the faster process (t_1) is equal to that of the other(slower) one (t'_1).

   $t_1 + t_2 = t'_1 + t'_2, \quad t_1 = t'_1$

2. The total time required for an action execution of the faster process to be completed ($t_1 + t_2$) is equal to that of the other(slower) one ($t'_1 + t'_2$). Also also the timing of action occurrence of the faster process (t_1) is faster than that of the other(slower) one (t'_1).

   $t_1 + t_2 = t'_1 + t'_2, \quad t_1 \leq t'_1$

3. The total time required for an action execution to be completed (t_1+t_2) of the faster process is equal to that of the other(slower) one $(t'_1+t'_2)$.

   $t_1+t_2=t'_1+t'_2$

4. The total time required for an action execution of the faster action to be completed (t_1+t_2) is shorter than that of the other(slower) one. Also the timing of action occurrence of the faster process (t_1) is equal to that of the other one (t'_1).

   $t_1+t_2 \leq t'_1+t'_2, \quad t_1=t'_1$,

5. The total time required for an action execution of the faster process to be completed (t_1+t_2) is shorter than that of the other(slower) one. Also the timing of action occurrence of the faster process (t_1) is faster than that of the other one(t'_1).

   $t_1+t_2 \leq t'_1+t'_2, \quad t_1 \leq t'_1$

6. The total time required for an action execution of the faster process to be completed (t_1+t_2) is shorter than that of the other(slower) one.

   $t_1+t_2 \leq t'_1+t'_2$

□

Here we can find that the "bisimulation relation" introduced in the former chapter is expressed by the first case listed above. Also Moller's faster-than relation is expressed by the second case. These speed relations are summarized in the table.4.1.4 The symbol of each relation is also shown in this table.

No.	relations	symbol	$t_1?t'_1$	$t_2\ ?\ t'_2$	$t_1+t_2\ ?\ t'_1+t'_2$
1.	exact-time equivalent	$\overset{e}{\sim}$	$=$	$(=)$	$=$
2.	action-occurrence-faster-than	$\overset{ac}{\lesssim}$	$\leq$	$(\geq)$	$=$
3.	total-time equivalent	$\overset{t}{\sim}$			$=$
4.	after-faster-than	$\overset{af}{\lesssim}$	$=$	$(\leq)$	$\leq$
5.	fairly-faster-than	$\underset{\sim}{\lll}$	$\leq$		$\leq$
6.	total-time-faster-than	$\overset{t}{\lesssim}$			$\leq$

Here in the table the condition surrounded by parenthesis means the condition is inferred from the other conditions.

All these relation reflects different aspects of real-time property. That is to say, processes are related with different speed relation. Bisimulation relation and the Moller's faster-than relation are respectively included by the exact-time bisimulation and action-occurrence-faster-than relation. Namely, exact-time bisimulation is a more general relation than bisimulation relation and action-occurrence-faster-than relation is a more general relation of Moller's faster-than relation. Although these reflect different aspects of real-time property, these relations themselves are related with each other by descriptive capability.

We explain relation of these relations by descriptive power in order. At first, the first relation of exact-time equivalence is a special case of the second one of action-occurrence-faster-than relation and the fourth one of after-faster-than relation; we can acquire $\overset{ac}{\lesssim} \supseteq \overset{e}{\sim}$ and $\overset{=}{\lesssim} \supseteq \overset{e}{\sim}$. The second relation of action-occurrence-faster-than relation is a special case of third relation of total-time equivalence and fifth case of fairly-faster-than relation; ; $\overset{t}{\sim} \supseteq \overset{ac}{\lesssim}$ and $\lessapprox \supseteq \overset{ac}{\lesssim}$. The fourth relation of after-faster-than relation is a special case of third relation of total-time equivalence; $\overset{=}{\lesssim} \supseteq \overset{t}{\sim}$. Finally the fifth relation of fairly-faster-than relation is a special case of the last one of total-time-faster-than equivalence; $\overset{=}{\lesssim} \supseteq \overset{t}{\lesssim}$.

The relation of these relations are exhibited in the figure below.

In this section, we considered how we can reflect the real-time property into relating process algebra in lTeCCS. Relating processes of lTeCCS had to be based on the the notion of Milner's simulation relation. That is, we had to attempt to reflect real-time property within the framework of simulation relation. We examined an LTS of lTeCCS and found there are two kinds of speed of processes. Consequently we proposed six relations which reflect the real-time property with in the framework of simulation relation. In next section, we formalize these six relations. Also we consider some properties of each relations, including a congruence.

3.3 Formalization of relations with real-time property

In this section, we conduct formalization of each relation proposed in the former section. Properties of each relation, including congruence, are also given with the proofs.

3.3.1 Exact-time bisimulation, exact-time equivalence

At first, we formalize the relation of the first case in the table 4.1.4, where the total time required for an action execution of the faster process to be completed $(t_1 + t_2)$ is equal

$$\begin{tikzcd}
 & \sim \arrow[dl] \arrow[d] & \\
\lesssim^{M} \arrow[d] & {}^{1}\overset{e}{\sim} \arrow[dl] \arrow[dr] & \\
{}^{2}\lesssim^{ac} \arrow[d] \arrow[drr] & & {}^{4}\lesssim^{af} \arrow[d] \\
{}^{3}\overset{t}{\sim} \arrow[drr] & & {}^{5}\overset{\ll}{\sim} \arrow[d] \\
 & & {}^{6}\lesssim^{t}
\end{tikzcd}$$

Figure 3.3: The inheritance elation of the relations

to that of the other(slower) one $(t'_1 + t'_2)$. Also the timing of action occurrence of the faster process (t_1) is equal to that of the other(slower) one (t'_1). Namely, the case of $t_1 = t'_1$ and $t_1 + t_2 = t'_1 + t'_2$.

No.	relations	symbol	$t_1 ? t'_1$	$t_2 ? t'_2$	$t_1 + t_2 ? t'_1 + t'_2$
1.	exact-time equivalent	$\overset{e}{\sim}$	$=$	$(=)$	$=$

Definition 9 (Strong exact-time bisimulation, strong exact-time equivalence) *A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong exact-time bisimulation if,*

for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s_2, s'_1, s'_2, t \in T$

(1) if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ then $\exists s'_1$, $\exists s'_2$, $\exists Q'$: $Q \overset{s'_1}{\leadsto}\xrightarrow{a}\overset{s'_2}{\leadsto} Q'$ such that $s_1 = s'_1$, $s_1+s_2 = s'_1+s'_2$ with $(P', Q') \in \mathcal{S}$;

(2) if $Q \overset{s'_1}{\leadsto}\xrightarrow{a}\overset{s'_2}{\leadsto} Q'$ then $\exists s_1$, $\exists s_2$, $\exists P'$: $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ such that $s_1 = s'_1$, $s_1+s_2 = s'_1+s'_2$ with $(P', Q') \in \mathcal{S}$;

(3) *if $P \overset{t}{\leadsto} P'$ then $\exists Q' : Q \overset{t}{\leadsto} Q'$ and $(P', Q') \in \mathcal{S}$;*

(4) *if $Q \overset{t}{\leadsto} Q'$ then $\exists P' : P \overset{t}{\leadsto} P'$ and $(P', Q') \in \mathcal{S}$.* □

Definition 10 *We say that P and Q are strongly exact-time equivalent, written $P \overset{e}{\sim} Q$, if $(P, Q) \in \mathcal{S}$ for some strong exact-time bisimulation $\mathcal{S}$.* □

Here are some properties of $\overset{e}{\sim}$.

Proposition 4

1. *$\overset{e}{\sim}$ is an equivalence relation over the lTeCCS processes, i.e. followings hold.*
   (1) $P \overset{e}{\sim} P$ (reflexivity)
   (2) $P_1 \overset{e}{\sim} P_2$ then $P_2 \overset{e}{\sim} P_1$ (symmetry)
   (3) $P_1 \overset{e}{\sim} P_2$ and $P_2 \overset{e}{\sim} P_3$ imply $P_1 \overset{e}{\sim} P_3$ (transitivity)

2. *$\overset{e}{\sim} \; = \; \bigcup\{\mathcal{S} \mid \mathcal{S}$ is a exact-time bisimulation$\}$*

proof:

1. For reflexivity, it is enough to show that the identity relation over lTeCCS, that is the relation $\{Id_{lTeCCS}\} = \{(P, P)|P \in lTeCCS\}$, is a exact-time bisimulation. This is obvious.

   For symmetry, we have to show that if $\mathcal{S}$ is a exact-time bisimulation then so is its converse $\mathcal{S}^{-1}$. But this is obvious from Definition.

   For transitivity, we must show that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are exact-time bisimulations, then so is their relational composition

$$\mathcal{S}_1\mathcal{S}_2 \;=\; \{(P, R) \mid \exists Q, P\mathcal{S}_1Q \text{ and } Q\mathcal{S}_2R\}.$$

   It is enough to show that this is a exact-time simulation. Let $(P, R) \in \mathcal{S}_1\mathcal{S}_2$, and $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$. Since there exists Q such that $P\mathcal{S}_1Q$ and $Q\mathcal{S}_2R$, there exist also

Q' such that $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ such that $s_1 = s_1'$, $s_1 + s_2 = s_1' + s_2'$ and $P'\mathcal{S}_1Q'$, and hence R' such that $R \overset{s_1''}{\rightsquigarrow}\xrightarrow{a}\overset{s_2''}{\rightsquigarrow} R'$ such that $s_1' = s_1''$, $s_1' + s_2' = s_1'' + s_2''$ and $Q'\mathcal{S}_1R'$. So $(P', R') \in \mathcal{S}_1\mathcal{S}_2$, and we have established the exact-time simulation condition for $\mathcal{S}_1\mathcal{S}_2$.

2. Let $P \overset{e}{\sim} Q$. Then by definition $P\mathcal{S}Q$ for some exact-time bisimulation $\mathcal{S}$. Therefore if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$, there exists Q' for which $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ such that $s_1 = s_1'$ and $s_1 + s_2 = s_1' + s_2'$ and $P'\mathcal{S}Q'$ - hence also $P' \overset{e}{\sim} Q'$. Thus $\overset{e}{\sim}$ satisfies the exact-time simulation condition and by symmetry so does its converse.

□

Proposition 5 (Strong equivalence and strong exact-time equivalence)

$$P \overset{e}{\sim} Q \textit{ implies } P \sim Q$$

□

Proposition 6 (Moller's faster-than relation and strong exact-time equivalence)

$$P \overset{e}{\sim} Q \textit{ implies } P \overset{<M}{\sim} Q$$

□

The definition and proposition are as follows;
We wish to find Q' which completes the following diagram:

Theorem 2 (Congruence)
$\overset{e}{\sim}$ is a process congruence with respect to the operators of lTeCCS;that is, if $P \overset{e}{\sim} Q$ then

1. $a.P \overset{e}{\sim} a.Q$
2. $P + M \overset{e}{\sim} Q + M$
3. $P \mid Q \overset{e}{\sim} Q \mid M$
4. $P[S] \overset{e}{\sim} Q[S]$
5. $(t).P \overset{e}{\sim} (t).Q$
6. $P\backslash L \overset{e}{\sim} Q\backslash L$

proof: We have to prove each of six cases above.

1. At fist, We prove that $\mathcal{S} = \{(a.P,\ a.Q) \mid P \overset{e}{\sim} Q\}$ is a exact-time bisimulation.

   (1) Assume that $a.P \overset{s_1}{\rightsquigarrow}\xrightarrow{a'}\overset{s_2}{\rightsquigarrow} P'$. By the assumption, this transition is described as $a.P \xrightarrow{a} P$. $a.P \xrightarrow{a} P$ then $a.Q \xrightarrow{a} Q$ and $P \overset{e}{\sim} Q$.

   (2) In the same way as (1).

   (3) Assume that $a.P \overset{t}{\rightsquigarrow} P$. From the assumption $a.P$, we acquire $P' = a.P$. $a.Q \overset{t}{\rightsquigarrow} Q$ and $(a.P,\ a.Q) \in \mathcal{S}$.

   (4) In the same way as (3).

2. Secondly, we prove that $\mathcal{S} = \{(P{+}M,\ Q{+}M) \mid P \overset{e}{\sim} Q\}$ is a exact-time bisimulation.

   (1) For any $(P+M, Q+M) \in \mathcal{S}$, we assume that $P+M \overset{s_1}{\rightsquigarrow}\xrightarrow{a'}\overset{s_2}{\rightsquigarrow} P'$. Then from the assumption these transitions can be described as $P{+}M \overset{s_1}{\rightsquigarrow} \hat{P}{+}M' \xrightarrow{a'}\overset{s_2}{\rightsquigarrow} P'$ for some M' such that $M \overset{s_1}{\rightsquigarrow} M'$. There are two possibilities for $\hat{P}+M' \xrightarrow{a'} P'$; in each case we find P' and for some Q', $Q+M \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ with $(P', Q') \in \mathcal{S}$.

      i. Suppose $\hat{P} \xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$, i.e. $\hat{P}$ has the form $a.(s_2).P'$, and $\hat{P}+M' \xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$. Since $P \overset{e}{\sim} Q$, we have $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ for some Q' such that $s_1 = s_1'$, $s_1{+}s_1' = s_2 + s_2'$ and $(P', Q') \in \mathcal{S}$.

      ii. Suppose $M' \xrightarrow{a''} M''$, i.e. M' has the form $a''.(s_2).M''$, and $\hat{P}+M' \xrightarrow{a''} M'' \overset{s_2}{\rightsquigarrow} P'$ for some M''. Then $Q+M \overset{s_1}{\rightsquigarrow} \hat{Q}+M' \xrightarrow{a''} M'' \overset{s_2}{\rightsquigarrow} P'$, such that $s_1 = s_1$, and $s_1+s_2 = s_1+s_2$ and $(P', P') \in \mathcal{S}$

   (2) Similar to (1).

   (3) We assume that $P+M \overset{t}{\rightsquigarrow} P'$. By the assumption, $P+M \overset{t}{\rightsquigarrow} \hat{P}+M'$ for some $P \overset{t}{\rightsquigarrow} \hat{P}$ and for some $M \overset{t}{\rightsquigarrow} M'$. Since $P \overset{e}{\sim} Q$, thus $Q+M \overset{t}{\rightsquigarrow} \hat{Q}+M'$ for some $Q \overset{t}{\rightsquigarrow} \hat{Q}$ and $(\hat{P}, \hat{Q}) \in \mathcal{S}$. We have $(\hat{P}+M', \hat{Q}+M') \in \mathcal{S}$.

   (4) Similar to (3).

3. Thirdly, we prove that $\mathcal{S} = \{(P \mid M,\ Q \mid M)\ \mid P \overset{e}{\sim} Q\}$ is a exact-time bisimulation.

   (1) For any $(P \mid M, Q \mid M) \in \mathcal{S}$, we assume that $P \mid M \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ for some P'. Then from the assumption these transitions can be described as $P \mid M \overset{s_1}{\rightsquigarrow} P_1 + M_1 \xrightarrow{a} P_2 \mid M_2 \overset{s_2}{\rightsquigarrow} P_3\ (= P')$ for some P_1, P_2, P_3, M_1, M_2, M_3 such that

$P \overset{s_1}{\rightsquigarrow} P_1$, $P_2 \overset{s_2}{\rightsquigarrow} P_3$ and $M \overset{s_1}{\rightsquigarrow} M_1$, $M_2 \overset{s_2}{\rightsquigarrow} M_3$. But here are three possibilities for $P_1 \mid M_1 \xrightarrow{a} P_2 \mid M_2$; in each case we find the form $P_2 \mid M_2$ such that $P_2 \mid M_2 \overset{s_2}{\rightsquigarrow} P'$and that for some Q', $Q \mid M \overset{s'_1}{\rightsquigarrow}\xrightarrow{a}\overset{s'_2}{\rightsquigarrow} Q'$ with $(P', Q') \in \mathcal{S}$.

i. $P_1 \xrightarrow{a} P_2$, i.e. P_1 has the form $a.P_2$, and $M_1 = M_2$. Thus $P \mid M \overset{s_1}{\rightsquigarrow} P_1 \mid M_1 \xrightarrow{a} P_2 \mid M_1 \overset{s_2}{\rightsquigarrow} P_3 \mid M_3 (= P')$. Since $P \overset{e}{\sim} Q$, we have $Q|M \overset{s'_1}{\rightsquigarrow} Q_1 \mid M_1 \xrightarrow{a} Q_2 \mid M_1 \overset{s'_2}{\rightsquigarrow} Q_3|M_3(= Q')$ for some Q' such that $s_1 = s'_1, s_1 + s_2 = s'_1 + s'_2$ and $(P', Q') \in \mathcal{S}$, as required.

ii. $P_1 = P_2$ and $M_1 \xrightarrow{a} M_2$. Since $P \overset{e}{\sim} Q$, if $P \overset{s_1}{\rightsquigarrow} P_1$, then we have $Q \overset{s_1}{\rightsquigarrow} Q_1$ for some Q_1 with $P_1 \overset{e}{\sim} Q_1$. Furthermore, $P_1(= P_2) \overset{s_2}{\rightsquigarrow} P_3$ implies $Q_1 \overset{s_2}{\rightsquigarrow} Q_3$ for some Q_3 with $P_3 \overset{e}{\sim} Q_3$. Thus we have $Q|M \overset{s'_1}{\rightsquigarrow} Q_1 \mid M_1 \xrightarrow{a} Q_1 \mid M_1 \overset{s'_2}{\rightsquigarrow} Q_3|M_3(= Q')$ for some Q' such that $s_1 = s'_1, s_1 + s_2 = s'_1 + s'_2$ and $(P', Q') \in \mathcal{S}$, as required.

iii. $P_1 \xrightarrow{b} P_2$ and $M_1 \xrightarrow{\bar{b}} M_2$, therefore a is a synchronization event τ, namely $a = \tau$. Since $P \overset{e}{\sim} Q$ and $P \mid M \overset{s_1}{\rightsquigarrow} P_1 \mid M_1 \xrightarrow{b} P_2 \mid M_2 \overset{s_2}{\rightsquigarrow} P_3 \mid M_3(= P')$, we have $Q \overset{s'_1}{\rightsquigarrow} Q_1 \xrightarrow{b} Q_2 \overset{s'_2}{\rightsquigarrow} Q_3$ for some Q_1, Q_2, Q_3 such that $s_1 = s'_1$, $s_1 + s_2 = s'_1 + s'_2$ and $P_3 \overset{e}{\sim} Q_3$. Thus, $Q|M \overset{s'_1}{\rightsquigarrow} Q_1 \mid M_1 \xrightarrow{a} Q_1 \mid M_2 \overset{s'_2}{\rightsquigarrow} Q_3|M_3(= Q')$ for some Q' such that $s_1 = s'_1, s_1 + s_2 = s'_1 + s'_2$ and $(P', Q') \in \mathcal{S}$, as required.

(2) Similar to (1).

(3) We assume that $P \mid M \overset{t}{\rightsquigarrow} P'$. By the assumption, this transition can be described as $P \mid M \overset{t}{\rightsquigarrow} P_1 \mid M_1$ for some $M \overset{t}{\rightsquigarrow} M_1$. Since $P \overset{e}{\sim} Q$, $Q \mid M \overset{t}{\rightsquigarrow} Q_1 \mid M_1$ for some $Q \overset{t}{\rightsquigarrow} Q_1$ and $(P_1, Q_1) \in \mathcal{S}$. We have $(P_1 \mid M_1, Q_1 \mid M_1) \in \mathcal{S}$.

(4) Similar to (3).

4. Next, we prove that $\mathcal{S} = \{(P[S], Q[S]) \mid P \overset{e}{\sim} Q\}$ is a exact-time bisimulation.

(1) Assume that $P[S] \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ for some P'. By the assumption, $P' = P'[S]$. $P \overset{e}{\sim} Q$ and $P[S] \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'[S]$, then $Q[S] \overset{s'_1}{\rightsquigarrow}\xrightarrow{a}\overset{s'_2}{\rightsquigarrow} Q'[S]$ for some $Q'[S]$ such that $s_1 = s'_1, s_1 + s'_1 = s_2 + s'_2$ and $(P'[S], Q'[S]) \in \mathcal{S}$

(2) Similar to (1).

(3) Assume that $P[S] \overset{t}{\rightsquigarrow} P'$ for some P'. By the assumption, $P' = P'[S]$. $P \overset{e}{\sim} Q$ and $P[S] \overset{t}{\rightsquigarrow} P'[S]$, then $Q[S] \overset{t}{\rightsquigarrow} Q'[S]$ for some $Q'[S]$ and $(P'[S], Q'[S]) \in \mathcal{S}$

(4) Similar to (3)

5. Here, We prove that $\mathcal{S} = \{((t).P,\ (t).Q) \mid P \overset{e}{\sim} Q\}$ is a exact-time bisimulation.

   (1) We do not have to consider this case.

   (2) Similar to (1).

   (3) Assume that $(t).P \overset{t}{\leadsto} P'$. From the assumption, we acquire $P' = P$. $(t).Q \overset{t}{\leadsto} Q$ and $(P,\ Q) \in \mathcal{S}$.

   (4) In the same way as (3).

6. Finally, we prove that $\mathcal{S} = \{(P\backslash L, Q\backslash L) \mid P \overset{e}{\sim} Q\}$ is a exact-time bisimulation.

   (1) Assume that $P\backslash L \overset{s_1}{\leadsto}\overset{a}{\longrightarrow}\overset{s_2}{\leadsto} P'$ for some P'. By the assumption, $P' = P'\backslash L$. $P \overset{e}{\sim} Q$ and $P\backslash L \overset{s_1}{\leadsto}\overset{a}{\longrightarrow}\overset{s_2}{\leadsto} P'\backslash L$, then $Q\backslash L \overset{s'_1}{\leadsto}\overset{a}{\longrightarrow}\overset{s'_2}{\leadsto} Q'\backslash L$ for some $Q'\backslash L$ such that $s_1 = s'_1, s_1 + s'_1 = s_2 + s'_2$ and $(P'\backslash L, Q'\backslash L) \in \mathcal{S}$

   (2) Similar to (1).

   (3) Assume that $P\backslash L \overset{t}{\leadsto} P'$ for some P'. By the assumption, $P' = P'\backslash L$. $P \overset{e}{\sim} Q$ and $P\backslash L \overset{t}{\leadsto} P'\backslash L$, then $Q\backslash L \overset{t}{\leadsto} Q'\backslash L$ for some $Q'\backslash L$ and $(P'\backslash L, Q'\backslash L) \in \mathcal{S}$

   (4) Similar to (3)

□

3.3.2 Action-occurrence-faster-than relation

Here we formalize the relation of the second case in the table 4.1.4, where the total time required for an action execution of the faster process to be completed $(t_1 + t_2)$ is equal to that of the other(slower) one $(t'_1 + t'_2)$. Also the timing of action occurrence of the faster process (t_1) is faster than that of the other(slower) one (t'_1). Namely, this is the case of $t_1 \le t'_1$ and $t_1 + t_2 = t'_1 + t'_2$.

No.	relations	symbol	$t_1 ? t'_1$	$t_2 ? t'_2$	$t_1 + t_2 ? t'_1 + t'_2$
2.	action-occurrence-faster-than	$\overset{ac}{\lesssim}$	$\le$	$(\ge)$	$=$

Definition 11 ($\overset{<^{ac}}{\sim}$-bisimulation, action-occurrence-faster-than relation) *A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong $\overset{<^{ac}}{\sim}$-bisimulation if, for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s_1', s_2's_2, t \in T$,*

(1) if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ then $\exists s_1'$, $\exists s_2'$, $\exists Q'$, : $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ such that $s_1 \leq s_1'$, $s_1 + s_2 = s_1' + s_2'$ and $(P', Q') \in \mathcal{S}$;

(2) if $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ then $\exists s_1'$, $\exists s_1'$, $\exists P'$: $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ such that $s_1 \leq s_1'$, $s_1 + s_2 = s_1' + s_2'$ and $(P', Q') \in \mathcal{S}$;

(3) if $P \overset{t}{\rightsquigarrow} P'$ then $\exists Q' : Q \overset{t}{\rightsquigarrow} Q'$ and $(P', Q') \in \mathcal{S}$;

(4) if $Q \overset{t}{\rightsquigarrow} Q'$ then $\exists P' : P \overset{t}{\rightsquigarrow} P'$ and $(P', Q') \in \mathcal{S}$. □

Definition 12 *We say that P and Q are strongly action-occurrence-faster-than relation, written $P \overset{<^{ac}}{\sim} Q$, if $(P, Q) \in \mathcal{S}$ for some strong $\overset{<^{ac}}{\sim}$-bisimulation $\mathcal{S}$.* □

Here are some properties of $\overset{<^{ac}}{\sim}$.

Proposition 7

1. $\overset{<^{ac}}{\sim}$ is an partial order relation over the lTeCCS processes, i.e. followings hold.
   (1) $P \overset{<^{ac}}{\sim} P$ (reflexivity)
   (2) $P_1 \overset{<^{ac}}{\sim} P_2$ and $P_2 \overset{<^{ac}}{\sim} P_1$ then $P_1 = P_2$ (asymmetry)
   (3) $P_1 \overset{<^{ac}}{\sim} P_2$ and $P_2 \overset{<^{ac}}{\sim} P_3$ imply $P_1 \overset{<^{ac}}{\sim} P_3$ (transitivity)

2. $\overset{<^{ac}}{\sim} \; = \; \bigcup\{\mathcal{S} \mid \mathcal{S}$ is a $\overset{<^{ac}}{\sim}$- bisimulation$\}$

proof:

1. For reflexivity, it is enough to show that the identity relation over lTeCCS, that is the relation $\{Id_{lTeCCS}\} = \{(P, P) | P \in lTeCCS\}$, is a $\overset{<^{ac}}{\sim}$-bisimulation. This is obvious.

   For asymmetry, we have to show that if $P_1 \overset{<^{ac}}{\sim} P_2$ and $P_2 \overset{<^{ac}}{\sim} P_1$ then $P_1 = P_2$. From the definition, if $P_1 \overset{<^{ac}}{\sim} P_2$ then $s_1 \leq s_1'$, also if $P_2 \overset{<^{ac}}{\sim} P_1$ then $s_1' \leq s_1$. Therefore $s_1 = s_1'$. From the definition $s_1 + s_2 = s_1' + s_2'$, so $s_2 = s_2'$. We can get $P_1 = P_2$.

   For transitivity, we must show that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are $\overset{<^{ac}}{\sim}$-bisimulations, then so is their relational composition

$$\mathcal{S}_1\mathcal{S}_2 = \{(P, R) \mid \exists Q, P\mathcal{S}_1 Q \textit{ and } Q\mathcal{S}_2 R\}.$$

It is enough to show that this is a $\overset{ac}{\lesssim}$- simulation. Let $(P, R) \in \mathcal{S}_1\mathcal{S}_2$, and $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$. Since there exists Q such that $P\mathcal{S}_1Q$ and $Q\mathcal{S}_2R$, there exist also Q' such that $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ such that $s_1 = s_1'$, $s_2 = s_2'$ and $P'\mathcal{S}_1Q'$, and hence R' such that $R \overset{s_1''}{\leadsto}\xrightarrow{a}\overset{s_2''}{\leadsto} R'$ such that $s_1' = s_1''$, $s_2' = s_2''$ and $Q'\mathcal{S}_1R'$. So $(P', R') \in \mathcal{S}_1\mathcal{S}_2$, and we have established the $\overset{ac}{\lesssim}$- simulation condition for $\mathcal{S}_1\mathcal{S}_2$.

2. Let $P \overset{ac}{\lesssim} Q$. Then by definition $P\mathcal{S}Q$ for some $\overset{ac}{\lesssim}$-bisimulation $\mathcal{S}$. Therefore if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$, there exists Q' for which $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ such that $s_1 = s_2$, $s_1' = s_2'$ and $P'\mathcal{S}Q'$ - hence also $P' \overset{ac}{\lesssim} Q'$. Thus $\overset{ac}{\lesssim}$ satisfies the $\overset{ac}{\lesssim}$- simulation condition and by symmetry so does its converse.

□

It is clear from the similarities in the definitions of $\overset{e}{\sim}$ and $\overset{ac}{\lesssim}$ that for P and Q being two terms of lTeCCS, if $P \overset{e}{\sim} Q$ then we can acquire $P \overset{ac}{\lesssim} Q$. However the reverse implication does not hold; that is, $P \overset{ac}{\lesssim} Q$ does not always imply that $P \overset{e}{\sim} Q$. This is shown in the following Proposition.

Proposition 8 (Moller's faster-than relation, strong action-occurrence-faster-than rel

$$P \overset{M}{\lesssim} Q \textit{ implies } P \overset{ac}{\lesssim} Q$$

□

We here state the notion of 'simulation up to'. In [Mil99], this notion of 'simulation up to' is introduced and shown why it is useful. A completely analogous result holds for $\overset{ac}{\lesssim}$-bisimulation up to $\overset{M}{\lesssim}$. The definition and proposition are as follows;

Definition 13 (Strong $\overset{ac}{\lesssim}$-bisimulation up to $\overset{M}{\lesssim}$)
A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong $\overset{ac}{\lesssim}$-bisimulation up to $\overset{M}{\lesssim}$ if, for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s_1', s_2, s_2', t \in T$, such that $s_1 \leq s_1'$ and $s_1 + s_2 = s_1' + s_2'$

(1) if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ then $\exists\ s_1', s_2'\ \exists Q'$: $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ and $P' \overset{M}{\lesssim} \mathcal{S} \overset{M}{\lesssim} Q'$;
(2) if $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ then $\exists\ s_1, s_2\ \exists P'$: $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ and $P' \overset{M}{\lesssim} \mathcal{S} \overset{M}{\lesssim} Q'$;
(3) if $P \overset{t}{\leadsto} P'$ then $\exists Q' : Q \overset{t}{\leadsto} Q'$ and $(P', Q') \in \mathcal{S}$;
(4) if $Q \overset{t}{\leadsto} Q'$ then $\exists P' : P \overset{t}{\leadsto} P'$ and $(P', Q') \in \mathcal{S}$.

□

Thus for $\mathcal{S}$ to be a strong simulation up to $\lesssim^{M}$ we must be able to complete the following diagram, given the top row and the left transition:

$$\begin{tikzcd}
 & P \arrow[dl, "a"'] & \mathcal{S} & Q \arrow[dr, "a"] & \\
P' & \lesssim^{M} & \mathcal{S} & \lesssim^{M} & Q'
\end{tikzcd}$$

Figure 3.4:

Proposition 9 *If $\mathcal{S}$ is a strong $\lesssim^{ac}$-bisimulation up to $\lesssim^{M}$ and $P\ \mathcal{S}\ Q$, then $P \lesssim^{ac} Q$.*

proof: Clearly $P\ \mathcal{S}\ Q$ implies $P \lesssim^{M} \mathcal{S} \lesssim^{M} Q$. So it will be enough to show that $P \lesssim^{M} \mathcal{S} \lesssim^{M} Q$ is a strong $\lesssim^{ac}$ bisimulation, for then $P \lesssim^{M} \mathcal{S} \lesssim^{M} Q$ implies $P \lesssim^{M} Q$ and we are done. Let $P \lesssim^{M} \mathcal{S} \lesssim^{M} Q$ implies $P \lesssim^{M} Q$ and $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$. We wish to find Q' which completes the following diagram:

$$\begin{tikzcd}
P \arrow[d, "a"] & \lesssim^{M} & \mathcal{S} & \lesssim^{M} & Q \arrow[d, "a"] \\
P' & \lesssim^{M} & \mathcal{S} & \lesssim^{M} & Q'
\end{tikzcd}$$

Figure 3.5:

To do this, first note that for some P_1 and Q_1 we have $P \lesssim^{M} P_1'$, $P_1\mathcal{S}Q_1$ and $Q_1 \lesssim^{M} Q$. Thus with the help Proposition 6 and knowing that $\mathcal{S}$ is a $\lesssim^{ac}$-bisimulation up to $\lesssim^{M}$, we can fill in the following three diagrams in sequence form left to right: Composing these, using the transitivity of $\lesssim^{M}$, we obtain the required diagram. □

Unfortunately, $\lesssim^{ac}$ is not a congruence for all operators of lTeCCS. This can be shown by a counter exmpale. As an exmple, we take the following processes

$$P_1 = a.(3).b.0$$

$$\begin{tikzcd}
P \arrow[r, phantom, "\lesssim^{M}"] \arrow[d, "a"'] & P_1 \arrow[d, "a"] & & P_1 \arrow[dl, "a"'] & \mathcal{S} & Q_1 \arrow[dr, "a"] & & Q_1 \arrow[r, phantom, "\lesssim^{M}"] \arrow[d, "a"'] & Q \arrow[d, "a"] \\
P' \arrow[r, phantom, "\lesssim^{M}"] & P_1' & P' \arrow[r, phantom, "\lesssim^{M}"] & & \mathcal{S} \arrow[rr, phantom, "\lesssim^{M}"] & & Q' & Q_1' \arrow[r, phantom, "\lesssim^{M}"] & Q'
\end{tikzcd}$$

Figure 3.6:

and

$$P_2 = (1).a.(2).b.0$$

Clearly we can establish that

$$P_1 = a.(3).b.0 \lesssim^{ac} (1).a.(2).b.0 = P_2$$

But when we take $M = (2).a.(2).b.0$ and compose process P_1 and P_2 with the process M, then we acquire $P_1 \mid M \not\lesssim^{ac} P_2 \mid M$ Obviously, the left hand side

$$P_1 \mid M \;=\; a.(3).b.0 \mid (2).\bar{a}.(2).\bar{b}.0 \overset{(2)}{\rightsquigarrow}\xrightarrow{\tau}\overset{(3)}{\rightsquigarrow} P_1' \mid M' \xrightarrow{\tau} 0$$

On the other hand, right hand side is

$$P_2 \mid M \;=\; (1).a.(2).b.0 \mid (2).\bar{a}.(2).\bar{b}.0 \overset{(2)}{\rightsquigarrow}\xrightarrow{\tau}\overset{(2)}{\rightsquigarrow} P_2' \mid M' \xrightarrow{\tau} 0$$

And we cannot acquire $P_1|M \lesssim^{ac} P_2|M$, hence $P_1|M \not\lesssim^{ac} P_2|M$. This example shows that $\lesssim^{ac}$ is not preserved by parallel operator " |". However we have the followings.

Theorem 3 (Congruence for lTeCCS)
$\lesssim^{ac}$ is a process congruence with respect to the operators of lTeCCS except for parallel operater " |";

that is, if $P \lesssim^{ac} Q$ then

1. $a.P \lesssim^{ac} a.Q$
2. $P + M \lesssim^{ac} Q + M$
4. $P[S] \lesssim^{ac} Q[S]$
5. $(t).P \lesssim^{ac} (t).Q$
6. $P\backslash L \lesssim^{ac} Q\backslash L$

□

The problem arises from the semantics of "$\mid$". So we here slightly change the semantics of lTeCCS as follows.

$$\frac{P \overset{s}{\leadsto} P', \quad Q \overset{t}{\leadsto} Q'}{P \mid Q \overset{s+t}{\leadsto} P \mid Q'}$$

And we call kTeCCS, which have the semantics above.

With the language kTeCCS, we can have the following.

Theorem 4 (Congruence for kTeCCS)
$\overset{ac}{\lesssim}$ *is a process congruence with respect to all the operators of kTeCCS; that is, if* $P \overset{ac}{\lesssim} Q$ *then*

1. $a.P \overset{ac}{\lesssim} a.Q$
2. $P + M \overset{ac}{\lesssim} Q + M$
3. $P \mid Q \overset{ac}{\lesssim} Q \mid M$
4. $P[S] \overset{ac}{\lesssim} Q[S]$
5. $(t).P \overset{ac}{\lesssim} (t).Q$
6. $P\backslash L \overset{ac}{\lesssim} Q\backslash L$

proof: We only demonstrate here the different part of the proof of subsitutivity in kTeCCS from lTeCCS, that respect to the parallel operator . The others are similar to those of $\overset{e}{\sim}$.

3. we prove that $\mathcal{S} = \{(P \mid M,\ Q \mid M) \ \mid P \overset{ac}{\lesssim} Q\}$ is a $\overset{ac}{\lesssim}$-bisimulation.

(1) For any $(P \mid M, Q \mid M) \in \mathcal{S}$, we assume that $P \mid M \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ for some P'. Then from the assumption these transitions can be described as $P \mid M \overset{s_1}{\leadsto} P_1 \mid M_1 \xrightarrow{a} P_2 \mid M_2 \overset{s_2}{\leadsto} P_3\ (= P')$ for some P_1, P_2, P_3, M_1, M_2, M_3: $P \overset{\hat{s}_1}{\leadsto} P_1$, $P_2 \overset{\hat{s}_2}{\leadsto} P_3$ and $M \overset{\tilde{s}_1}{\leadsto} M_1$ $M_2 \overset{\tilde{s}_2}{\leadsto} M_3$, such that $s_1 = \hat{s}_1 + \tilde{s}_1$ and $s_2 = \hat{s}_2 + \tilde{s}_2$. But here are three possibilities for $P_1 \mid M_1 \xrightarrow{a} P_2 \mid M_2$; in each case we find the form $P_2 \mid M_2$ such that $P_2 \mid M_2 \overset{s_2}{\leadsto} P'$ and that for some Q', s_1', s_2'; $Q \mid M \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ such that $s_1 \leq s_1'$, $s_1 + s_2 = s_1 + s_2'$ with $(P', Q') \in \mathcal{S}$.

i $P_1 \xrightarrow{a} P_2$, i.e. P_1 has the form $a.P_2$, and $M_1 = M_2$. Thus $P \mid M \overset{s_1}{\leadsto} P_1 \mid M_1 \xrightarrow{a} P_2 \mid M_1 \overset{s_2}{\leadsto} P_3 \mid M_3 (= P')$. Since $P \overset{ac}{\lesssim} Q$, we have $Q|M \overset{s_1'}{\leadsto}$

$Q_1 \mid M_1 \xrightarrow{a} Q_2 \mid M_1 \overset{s_2'}{\rightsquigarrow} Q_3|M_3(=Q')$ for some Q' such that $s_1 \leq s_1'$, $s_1 + s_2 = s_1' + s_2'$ and $(P', Q') \in \mathcal{S}$, as required.

ii $P_1 = P_2$ and $M_1 \xrightarrow{a} M_2$. Since $P \overset{<^{ac}}{\sim} Q$, if $P \overset{s_1}{\rightsquigarrow} P_1$, then we have $Q \overset{s_1}{\rightsquigarrow} Q_1$ for some Q_1 with $P_1 \overset{<^{ac}}{\sim} Q_1$. Furthermore, $P_1(=P_2) \overset{s_2}{\rightsquigarrow} P_3$ implies $Q_1 \overset{s_2}{\rightsquigarrow} Q_3$ for some Q_3 with $P_3 \overset{<^{ac}}{\sim} Q_3$. Thus we have $Q|M \overset{s_1'}{\rightsquigarrow} Q_1 \mid M_1 \xrightarrow{a} Q_1 \mid M_1 \overset{s_2'}{\rightsquigarrow} Q_3|M_3(=Q')$ for some Q' such that $s_1 \leq s_1'$, $s_1 + s_2 = s_1' + s_2'$ and $(P', Q') \in \mathcal{S}$, as required.

iii $P_1 \xrightarrow{b} P_2$ and $M_1 \xrightarrow{\bar{b}} M_2$, therefore a is a synchronization event τ, namely $a = \tau$. Since $P \mid M \overset{s_1}{\rightsquigarrow} P_1 \mid M_1 \xrightarrow{b} P_2 \mid M_2 \overset{s_2}{\rightsquigarrow} P_3 \mid M_3(=P')$, $Q|M \overset{s_1'}{\rightsquigarrow} Q_1 \mid M_1 \xrightarrow{a} Q_1 \mid M_2 \overset{s_2'}{\rightsquigarrow} Q_3|M_3(=Q')$ for some Q'. Here we consider $Q \overset{\hat{s}_1'}{\rightsquigarrow} Q_1$, $Q_2 \overset{\hat{s}_2'}{\rightsquigarrow} Q_3$, such that $s_1 = \hat{s}_1' + \tilde{s}_1$ and $s_2' = \hat{s}_2' + \tilde{s}_2$. Since $P \overset{<^{ac}}{\sim} Q$, accordingly $s_1 \leq s_1'$, $s_1 + s_2 = s_1' + s_2'$ and $(P', Q') \in \mathcal{S}$, as required.

(2) Similar to (1).

(3) We assume that $P \mid M \overset{t}{\rightsquigarrow} P'$. By the assumption, this transition can be described as $P \mid M \overset{t}{\rightsquigarrow} P_1 \mid M_1$ for some $M \overset{t}{\rightsquigarrow} M_1$. Since $P \overset{<^{ac}}{\sim} Q$, $Q \mid M \overset{t}{\rightsquigarrow} Q_1 \mid M_1$ for some $Q \overset{t}{\rightsquigarrow} Q_1$ and $(P_1, Q_1) \in \mathcal{S}$. We have $(P_1 \mid M_1, Q_1 \mid M_1) \in \mathcal{S}$.

(4) Similar to (3).

□

3.3.3 Strong total-time equivalence

In this section we formalize the relation of the third case in the table 4.1.4, where the total time required for an action execution of the faster process to be completed $(t_1 + t_2)$ is equal to that of the other(slower) one $(t_1' + t_2')$. Namely, this is the case of $t_1 + t_2 = t_1' + t_2'$

No.	relations	symbol	$t_1 ? t_1'$	$t_2 ? t_2'$	$t_1 + t_2 ? t_1' + t_2'$
3.	total-time equivalent	$\overset{t}{\sim}$			$=$

Namely, we formalize the binary relation of processes whose functional behaviour is identical but operate at the speed in the 3rd case of speed comparison in the table4.1.4. We deem two functionally equivalent processes, where two processes operate at the same speed in their total time to be related even if they are different from each other in timing of action occurrence. This is formally defined as follows,

Definition 14 (Strong total-time bisimulation, strong total-time equivalence) *A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong total-time bisimulation if, for all $(P,Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s, t \in T$,*

(1) if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ then $\exists s_1'$, $\exists s_2'$, $\exists Q'$: $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ such that $s_1 + s_2 = s_1' + s_2'$ and $(P', Q') \in \mathcal{S}$;

(2) if $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ then $\exists s_1$, $\exists s_2$, $\exists P'$: $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ such that $s_1 + s_2 = s_1' + s_2'$ and $(P', Q') \in \mathcal{S}$;

(3) *if $P \overset{t}{\rightsquigarrow} P'$ then $\exists Q' : Q \overset{t}{\rightsquigarrow} Q'$ and $(P', Q') \in \mathcal{S}$;*

(4) *if $Q \overset{t}{\rightsquigarrow} Q'$ then $\exists P' : P \overset{t}{\rightsquigarrow} P'$ and $(P', Q') \in \mathcal{S}$.* □

Definition 15 *We say that P and Q are strongly total-time equivalent, written $P \overset{t}{\sim} Q$, if $(P,Q) \in \mathcal{S}$ for some strong total-time bisimulation $\mathcal{S}$.* □

Here are some properties of $\overset{t}{\sim}$.

Proposition 10

1. *$\overset{t}{\sim}$ is an equivalence relation over the lTeCCS processes, i.e. followings hold.*
   (1) $P \overset{t}{\sim} P$ (reflexivity)
   (2) $P_1 \overset{t}{\sim} P_2$ then $P_2 \overset{t}{\sim} P_1$ (symmetry)
   (3) $P_1 \overset{t}{\sim} P_2$ and $P_2 \overset{t}{\sim} P_3$ imply $P_1 \overset{t}{\sim} P_3$ (transitivity)
2. *$\overset{t}{\sim} \; = \; \bigcup\{\mathcal{S} \mid \mathcal{S}$ is a total-time bisimulation$\}$*

proof:

1. For reflexivity, it is enough to show that the identity relation over lTeCCS, that is the relation $\{Id_{lTeCCS}\} = \{(P,P) | P \in lTeCCS\}$, is a bisimulation. This is obvious.

   For symmetry, we have to show that if $\mathcal{S}$ is a bisimulation then so is its converse $\mathcal{S}^{-1}$. Let $(P,Q) \in \mathcal{S}$. Then from definition, it is obvious that Q $\overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ and $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ *with* $(Q', P') \in \mathcal{S}^{-1}$;

   For transitivity, we must show that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are bisimulations, then so is their relational composition

$$\mathcal{S}_1\mathcal{S}_2 \;=\; \{(P,R) \mid \exists Q, P\mathcal{S}_1Q \text{ and } Q\mathcal{S}_2R\}.$$

It is enough to show that this is a simulation. Let $(P, R) \in \mathcal{S}_1\mathcal{S}_2$, and $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$. Since there exists Q such that $P\mathcal{S}_1Q$ and $Q\mathcal{S}_2R$, there exist also Q' such that $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$, where $s_1+s_2 = s_1'+s_2'$, and $P'\mathcal{S}_1Q'$, and hence R' such that $R \overset{s_1''}{\rightsquigarrow}\xrightarrow{a}\overset{s_2''}{\rightsquigarrow} R'$, where $s_1' + s_2' = s_1'' + s_2''$, and $Q'\mathcal{S}_1R'$. So $(P', R') \in \mathcal{S}_1\mathcal{S}_2$, and we have established the simulation condition for $\mathcal{S}_1\mathcal{S}_2$.

2. Let $P \overset{t}{\sim} Q$. Then by definition $P\mathcal{S}Q$ for some total-time bisimulation $\mathcal{S}$. Therefore if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$, there exists Q' for which $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$, where $s_1 + s_2 = s_1' + s_2'$, and $P'\mathcal{S}Q'$ - hence also $P' \overset{t}{\sim} Q'$. Thus $\sim$ satisfies the simulation condition and by symmetry so does its converse.

□

Proposition 11 (Strong exact-time equivalence and Strong exact-time equivalence)

$$P \overset{e}{\sim} Q \textit{ implies } P \overset{t}{\sim} Q$$

□

Definition 16 (Strong total-time bisimulation up to $\overset{e}{\sim}$)
A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong total-time bisimulation up to $\overset{e}{\sim}$ if, for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s, t \in T$, $s_1 + s_2 = s_1' + s_2'$
(1) if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ then $\exists$ s_1, s_2 $\exists Q'$:$Q \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} Q'$ and $P' \overset{e}{\sim} \mathcal{S} \overset{e}{\sim} Q'$;
(2) if $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ then $\exists$ s_1, s_2 $\exists P'$:$P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ and $P' \overset{e}{\sim} \mathcal{S} \overset{e}{\sim} Q'$;
(3) if $P \overset{t}{\rightsquigarrow} P'$ then $\exists Q' : Q \overset{t}{\rightsquigarrow} Q'$ and $(P', Q') \in \mathcal{S}$;
(4) if $Q \overset{t}{\rightsquigarrow} Q'$ then $\exists P' : P \overset{t}{\rightsquigarrow} P'$ and $(P', Q') \in \mathcal{S}$.

□

Thus for $\mathcal{S}$ to be a strong simulation up to $\overset{e}{\sim}$ we must be able to complete the following diagram, given the top row and the left transition:

Proposition 12 *If $\mathcal{S}$ is a strong bisimulation up to $\overset{e}{\sim}$ and $P\mathcal{S}Q$, then $P \sim Q$.*

proof: Clearly $P\mathcal{S}Q$ implies $P \overset{e}{\sim} \mathcal{S} \overset{e}{\sim} Q$. So it will be enough to show that $P \overset{e}{\sim} \mathcal{S} \overset{e}{\sim} Q$ is a strong total-time bisimulation, for then $P \overset{e}{\sim} \mathcal{S} \overset{e}{\sim} Q$ implies $P \sim Q$ and we are done.

Let $P \overset{e}{\sim} \mathcal{S} \overset{e}{\sim} Q$ implies $P \sim Q$ and $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$. We with to find Q' which completes the following diagram:

To do this, first note that for some P_1 and Q_1 we have $P \overset{e}{\sim} P_1$, $P_1 \mathcal{S} \overset{e}{\sim} Q_1$ and $Q_1 \overset{e}{\sim} Q$. Thus with the help Proposition 6 and knowing that $\mathcal{S}$ is a strong bisimulation up to $\overset{e}{\sim}$, we can fill in the following three diagrams in sequence form left to right:

Composing these, using the transitivity of $\overset{e}{\sim}$, we obtain the required diagram □

Theorem 5 (Congruence for *l*TeCCS)
$\overset{t}{\sim}$ *is a process congruence with respect to the operators of lTeCCS except for parallel operater " |"* □

Theorem 6 (Congruence for *k*TeCCS)
$\overset{t}{\sim}$ *is a process congruence with respect to all the operators of kTeCCS.*

proof: The proof is similar to that of $\overset{e}{\sim}$. □

3.3.4 Strong after-faster-than relation

Here we formalize the relation of the fourth case in the table 4.1.4, where the total time required for an action execution of the faster process to be completed $(t_1 + t_2)$ is shorter than that of the other(slower) one. Also the timing of action occurrence of the faster process (t_1) is equal to that of the other one (t_1').

No.	relations	symbol	$t_1?t_1'$	$t_2 \ ? \ t_2'$	$t_1 + t_2 \ ? \ t_1' + t_2'$
4.	after-faster-than	$\overset{<af}{\sim}$	$=$	$(\leq)$	$\leq$

Definition 17 ($\overset{<af}{\sim}$-bisimulation, after-faster-than relation) *A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong $\overset{<af}{\sim}$-bisimulation if, for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s_1', s_2, s_2', t \in T$,*

(1) if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ then $\exists s_1'$, $\exists s_2'$, $\exists Q'$,: $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ such that $s_1 = s_1'$, $s_1 + s_2 \leq s_1' + s_2'$ with $(P', Q') \in \mathcal{S}$;

(2) if $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ then $\exists s_1'$, $\exists s_1'$, $\exists P'$,: $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ such that $s_1 = s_1'$, $s_1 + s_2 \leq s_1' + s_2'$ with $(P', Q') \in \mathcal{S}$;

(3) *if $P \overset{t}{\leadsto} P'$ then $\exists Q' : Q \overset{t}{\leadsto} Q'$ and $(P', Q') \in \mathcal{S}$;*

(4) *if $Q \overset{t}{\leadsto} Q'$ then $\exists P' : P \overset{t}{\leadsto} P'$ and $(P', Q') \in \mathcal{S}$.* □

Definition 18 *We say that P and Q are strongly after-faster-than relation, written* $P \overset{af}{\lesssim} Q$, *if* $(P, Q) \in \mathcal{S}$ *for some strong* $\overset{af}{\lesssim}$*-bisimulation* $\mathcal{S}$. □

Here are some properties of $\overset{af}{\lesssim}$.

Proposition 13 .

1. $\overset{af}{\lesssim}$ *is an partial order relation over the lTeCCS processes, i.e. followings hold.*
   (1) $P \overset{af}{\lesssim} P$ *(reflexivity)*
   (2) $P_1 \overset{af}{\lesssim} P_2$ *and* $P_2 \overset{af}{\lesssim} P_1$ *then* $P_1 = P_2$ *(asymmetry)*
   (3) $P_1 \overset{af}{\lesssim} P_2$ *and* $P_2 \overset{af}{\lesssim} P_3$ *imply* $P_1 \overset{af}{\lesssim} P_3$ *(transitivity)*
2. $\overset{af}{\lesssim} \;=\; \bigcup\{\mathcal{S} \mid \mathcal{S}$ *is a* $\overset{af}{\lesssim}$*- bisimulation*$\}$

proof:

1. For reflexivity, it is enough to show that the identity relation over lTeCCS, that is the relation $\{Id_{lTeCCS}\} = \{(P, P) | P \in lTeCCS\}$, is a $\overset{af}{\lesssim}$-bisimulation. This is obvious.

   For asymmetry, we have to show that if $P_1 \overset{af}{\lesssim} P_2$ and $P_2 \overset{af}{\lesssim} P_1$ then $P_1 = P_2$. From the definition, if $P_1 \overset{af}{\lesssim} P_2$ then $s_1 + s_2 \leq s'_1 + s'_2$, also if $P_2 \overset{af}{\lesssim} P_1$ then $s'_1 + s'_2 \leq s_1 + s_2$. Therefore $s_1 + s_2 = s'_1 + s'_2$. From the definition $s_1 = s'_1$, so $s_2 = s'_2$. We can get $P_1 = P_2$.

   For transitivity, we must show that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are $\overset{af}{\lesssim}$-bisimulations, then so is their relational composition

   $$\mathcal{S}_1\mathcal{S}_2 \;=\; \{(P, R) \mid \exists Q, P\mathcal{S}_1 Q \; and \; Q\mathcal{S}_2 R\}.$$

   It is enough to show that this is a $\overset{af}{\lesssim}$- simulation. Let $(P, R) \in \mathcal{S}_1\mathcal{S}_2$, and $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$. Since there exists Q such that $P\mathcal{S}_1Q$ and $Q\mathcal{S}_2R$, there exist also Q' such that $Q \overset{s'_1}{\leadsto}\xrightarrow{a}\overset{s'_2}{\leadsto} Q'$ such that $s_1 = s'_1$, $s_1 + s_2 \leq s'_1 + s'_2$ and $P'\mathcal{S}_1Q'$, and hence R' such that $R \overset{s''_1}{\leadsto}\xrightarrow{a}\overset{s''_2}{\leadsto} R'$ such that $s'_1 = s''_1$, $s'_1 + s'_2 \leq s''_1 + s''_2$ and $Q'\mathcal{S}_1R'$. So $(P', R') \in \mathcal{S}_1\mathcal{S}_2$, and we have established the $\overset{af}{\lesssim}$- simulation condition for $\mathcal{S}_1\mathcal{S}_2$.

2. Let $P \overset{af}{\lesssim} Q$. Then by definition $P\mathcal{S}Q$ for some $\overset{af}{\lesssim}$-bisimulation $\mathcal{S}$. Therefore if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$, there exists Q' for which $Q \overset{s'_1}{\leadsto}\xrightarrow{a}\overset{s'_2}{\leadsto} Q'$ such that $s_1 = s_2$, $s_1 + s_2 \leq s'_1 + s'_2$ and $P'\mathcal{S}Q'$ - hence also $P' \overset{af}{\lesssim} Q'$. Thus $\overset{af}{\lesssim}$ satisfies the $\overset{af}{\lesssim}$- simulation condition and by symmetry so does its converse.

□

Proposition 14 (Exact-time faster-than relation, strong after-faster-than relation)

$$P \overset{e}{\sim} Q \textit{ implies } P \overset{<af}{\sim} Q$$

□

Definition 19 (Strong $\overset{<af}{\sim}$-bisimulation up to $\overset{e}{\sim}$)
A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong $\overset{<af}{\sim}$-bisimulation up to $\overset{e}{\sim}$ if, for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s_1', s_2, s_2', t \in T$, such that $s_1 = s_1'$ and $s_1 + s_2 \leq s_1' + s_2'$

(1) if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ then $\exists$ s_1', s_2' $\exists Q'$: $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ and $P' \sim \mathcal{S} \sim Q'$;
(2) if $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ then $\exists$ s_1, s_2 $\exists P'$: $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ and $P' \sim \mathcal{S} \sim Q'$;
(3) if $P \overset{t}{\leadsto} P'$ then $\exists Q' : Q \overset{t}{\leadsto} Q'$ and $(P', Q') \in \mathcal{S}$;
(4) if $Q \overset{t}{\leadsto} Q'$ then $\exists P' : P \overset{t}{\leadsto} P'$ and $(P', Q') \in \mathcal{S}$.

□

Proposition 15 *If $\mathcal{S}$ is a strong $\overset{<af}{\sim}$-bisimulation up to $\overset{e}{\sim}$ and $P\mathcal{S}Q$, then $P \overset{<af}{\sim} Q$.*

proof: Similar to Theorem 3.

□

Theorem 7 (Congruence for lTeCCS)
$\overset{<af}{\sim}$ is a process congruence with respect to the operators of lTeCCS except for parallel operater " |". □

Theorem 8 (Congruence for kTeCCS)
$\overset{<af}{\sim}$ is a process congruence with respect to all the operators of kTeCCS.

proof: The proof is similar to that of $\overset{e}{\sim}$. □

3.3.5 Strong fairly-faster-than relation

Here we formalize the relation of the second case in the table 4.1.4, where In this section, where total time required for an action execution of the faster process to be completed (t_1+t_2) is shorter than that of the other(slower) one. Also the timing of action occurrence of the faster process (t_1) is faster than that of the other one(t'_1). Namely, this is the case of $t_1 \leq t'_1$, $t_1+t_2 \leq t'_1+t'_2$.

No.	relations	symbol	$t_1 ? t'_1$	$t_2\ ?\ t'_2$	$t_1+t_2\ ?\ t'_1+t'_2$
5.	fairly-faster-than	$\overset{\ll}{\sim}$	$\leq$		$\leq$

Definition 20 (Strong $\overset{\ll}{\sim}$-bisimulation, strong fairly-faster-than relation) *A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong $\overset{\ll}{\sim}$-bisimulation if, for all $(P,Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s'_1, s'_2, s_2, t \in T$,*

(1) if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ then $\exists s'_1$, $\exists s'_2$, $\exists Q'$,: $Q \overset{s'_1}{\rightsquigarrow}\xrightarrow{a}\overset{s'_2}{\rightsquigarrow} Q'$, such that $s_1 \leq s'_1$, $s_1+s_2 \leq s'_1+s'_2$, with $(P',Q') \in \mathcal{S}$;

(2) if $Q \overset{s'_1}{\rightsquigarrow}\xrightarrow{a}\overset{s'_2}{\rightsquigarrow} Q'$ then $\exists s'_1$, $\exists s'_1$, $\exists P'$,: $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$, such that $s_1 \leq s'_1$, $s_1+s_2 \leq s'_1+s'_2$, and $(P',Q') \in \mathcal{S}$;

(3) *if* $P \overset{t}{\rightsquigarrow} P'$ *then* $\exists Q' : Q \overset{t}{\rightsquigarrow} Q'$ *and* $(P',Q') \in \mathcal{S}$;

(4) *if* $Q \overset{t}{\rightsquigarrow} Q'$ *then* $\exists P' : P \overset{t}{\rightsquigarrow} P'$ *and* $(P',Q') \in \mathcal{S}$. □

Definition 21 *We say that P and Q are strongly fairly-faster-than relation, written $P \overset{<^{ac}}{\sim} Q$, if $(P,Q) \in \mathcal{S}$ for some strong $\overset{\ll}{\sim}$-bisimulation $\mathcal{S}$.* □

Proposition 16 *The binary relation $\overset{\ll}{\sim}$ is a partial order over lTeCCS processes, i.e. the followings hold:*

(1) $P \overset{\ll}{\sim} P$ (reflexivity)

(2) $P_1 \overset{\ll}{\sim} P_2$ and $P_2 \overset{\ll}{\sim} P_3$ imply $P_1 \overset{\ll}{\sim} P_3$ (transitivity)

(3) $P_1 \overset{\ll}{\sim} P_2$ and $P_2 \overset{\ll}{\sim} P_1$ imply $P_1 = P_2$ (asymmetry)

proof:

1. For reflexivity, it is enough to show that the identity relation over lTeCCS, that is the relation $\{Id_{lTeCCS}\} = \{(P, P) | P \in lTeCCS\}$, is a $\overset{\ll}{\sim}$-bisimulation. This is obvious.

   For asymmetry, we have to show that if $P_1 \overset{\ll}{\sim} P_2$ and $P_2 \overset{\ll}{\sim} P_1$ then $P_1 = P_2$. From the definition, if $P_1 \overset{\ll}{\sim} P_2$ then $s_1 \leq s_1'$, $s_1 + s_2 \leq s_1' + s_2'$, also if $P_2 \overset{\ll}{\sim} P_1$ then $s_1' \leq s_1$, $s_1' + s_2' \leq s_1 + s_2$. Therefore $s_1 = s_1'$, $s_1 + s_2 = s_1' + s_2'$ and $s_2 = s_2'$. Consequently, we can get $P_1 = P_2$.

   For transitivity, we must show that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are $\overset{\ll}{\sim}$-bisimulations, then so is their relational composition

   $$\mathcal{S}_1\mathcal{S}_2 \;=\; \{(P, R) \mid \exists Q, P\mathcal{S}_1 Q \ and \ Q\mathcal{S}_2 R\}.$$

   It is enough to show that this is a $\overset{\ll}{\sim}$- simulation. Let $(P, R) \in \mathcal{S}_1\mathcal{S}_2$, and $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$. Since there exists Q such that $P\mathcal{S}_1 Q$ and $Q\mathcal{S}_2 R$, there exist also Q' such that $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ such that $s_1 \leq s_1'$, $s_1 + s_2 \leq s_1' + s_2'$ and $P'\mathcal{S}_1 Q'$, and hence R' such that $R \overset{s_1''}{\rightsquigarrow}\xrightarrow{a}\overset{s_2''}{\rightsquigarrow} R'$ such that $s_1' \leq s_1''$, $s_1' + s_2' \leq s_1'' + s_2''$ and $Q'\mathcal{S}_1 R'$. So $(P', R') \in \mathcal{S}_1\mathcal{S}_2$, and we have established the $\overset{\ll}{\sim}$- simulation condition for $\mathcal{S}_1\mathcal{S}_2$.

2. Let $P \overset{\ll}{\sim} Q$. Then by definition $P\mathcal{S}Q$ for some $\overset{\ll}{\sim}$-bisimulation $\mathcal{S}$. Therefore if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$, there exists Q' for which $Q \overset{s_1'}{\rightsquigarrow}\xrightarrow{a}\overset{s_2'}{\rightsquigarrow} Q'$ such that $s_1 = s_2$, $s_1 + s_2 \leq s_1' + s_2'$ and $P'\mathcal{S}Q'$ - hence also $P' \overset{\ll}{\sim} Q'$. Thus $\overset{\ll}{\sim}$ satisfies the $\overset{\ll}{\sim}$- simulation condition and by symmetry so does its converse.

□

Proposition 17 (Strong fairly-faster-than relation)

$$P \overset{<^{ac}}{\sim} Q \ implies \ P \overset{\ll}{\sim} Q$$

$$P \overset{<^{af}}{\sim} Q \ implies \ P \overset{\ll}{\sim} Q$$

□

Definition 22 (Strong $\overset{\ll}{\sim}$-bisimulation up to $\overset{<^{ac}}{\sim}$)
A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong $\overset{\ll}{\sim}$-bisimulation up to $\overset{<^{ac}}{\sim}$ if, for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s_1', s_2, s_2', t \in T$, such that $s_1 \leq s_1'$ and $s_1 + s_2 \leq s_1' + s_2'$

(1) if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ *then* $\exists\ s_1',\ s_2'\ \exists Q'$*:* $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ *and* $P' \sim \mathcal{S} \sim Q'$*;*
(2) if $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ *then* $\exists\ s_1,\ s_2\ \exists P'$*:* $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ *and* $P' \sim \mathcal{S} \sim Q'$*;*
(3) if $P \overset{t}{\leadsto} P'$ *then* $\exists Q' : Q \overset{t}{\leadsto} Q'$ *and* $(P', Q') \in \mathcal{S}$*;*
(4) if $Q \overset{t}{\leadsto} Q'$ *then* $\exists P' : P \overset{t}{\leadsto} P'$ *and* $(P', Q') \in \mathcal{S}$*.*

□

Proposition 18 *If* $\mathcal{S}$ *is a strong* $\overset{\lessdot}{\sim}$*-bisimulation up to* $\overset{<^{ac}}{\sim}$ *and* $P\mathcal{S}Q$*, then* $P \overset{\lessdot}{\sim} Q$*.*

□

Definition 23 (Strong $\overset{\lessdot}{\sim}$-bisimulation up to $\overset{<^{af}}{\sim}$)
A binary relation $\mathcal{S} \subseteq P \times P$ *over lTeCCS processes is a strong* $\overset{\lessdot}{\sim}$*-bisimulation up to* $\overset{<^{af}}{\sim}$ *if, for all* $(P, Q) \in \mathcal{S}$ *and for all* $a \in Act$ *and for all* $s_1, s_1', s_2, s_2', t \in T$*, such that* $s_1 \leq s_1'$ *and* $s_1 + s_2 \leq s_1' + s_2'$
(1) if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ *then* $\exists\ s_1',\ s_2'\ \exists Q'$*:* $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ *and* $P' \sim \mathcal{S} \sim Q'$*;*
(2) if $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ *then* $\exists\ s_1,\ s_2\ \exists P'$*:* $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ *and* $P' \sim \mathcal{S} \sim Q'$*;*
(3) if $P \overset{t}{\leadsto} P'$ *then* $\exists Q' : Q \overset{t}{\leadsto} Q'$ *and* $(P', Q') \in \mathcal{S}$*;*
(4) if $Q \overset{t}{\leadsto} Q'$ *then* $\exists P' : P \overset{t}{\leadsto} P'$ *and* $(P', Q') \in \mathcal{S}$*.*

□

Proposition 19 *If* $\mathcal{S}$ *is a strong* $\overset{\lessdot}{\sim}$*-bisimulation up to* $\overset{<^{af}}{\sim}$ *and* $P\mathcal{S}Q$*, then* $P \overset{\lessdot}{\sim} Q$*.*

□

We show here that $\overset{\lessdot}{\sim}$ is congruence (substitute) over *l*TeCCS terms except for parallel operater " |".

Theorem 9 (Congruence for *l*TeCCS) $\overset{\lessdot}{\sim}$ *is a process congruence with respect to the operators of lTeCCS except for parallel operater " |".* □

Theorem 10 (Congruence for *k*TeCCS)
$\overset{\lessdot}{\sim}$ *is a process congruence with respect to all the operators of kTeCCS.*

proof: The proof is similar to that of $\overset{e}{\sim}$. □

3.3.6 Strong total-time-faster-than relation

Finally, we formalize the relation of the sixth case in the table 4.1.4, where the total time required for an action of the faster process execution to be completed ($t_1 + t_2$) is shorter than that of the other(slower) one. This is the case of $t_1 + t_2 \leq t'_1 + t'_2$.

No.	relations	symbol	$t_1 ? t'_1$	$t_2 ? t'_2$	$t_1 + t_2 ? t'_1 + t'_2$
6.	total-time-faster-than	$\overset{<t}{\sim}$			$\leq$

Definition 24 ($\overset{<t}{\sim}$-bisimulation, total-time-faster-than relation) *A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong $\overset{<t}{\sim}$-bisimulation if, for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s'_1, s_2, s'_2, t \in T$,*

(1) if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$ then $\exists s'_1$, $\exists s'_2$, $\exists Q'$: $Q \overset{s'_1}{\rightsquigarrow}\xrightarrow{a}\overset{s'_2}{\rightsquigarrow} Q'$, such that $s_1 + s_2 = s'_1 + s'_2$, and $(P', Q') \in \mathcal{S}$;

(2) if $Q \overset{s'_1}{\rightsquigarrow}\xrightarrow{a}\overset{s'_2}{\rightsquigarrow} Q'$ then $\exists s'_1$, $\exists s'_1$, $\exists P'$,: $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$, such that $s_1 + s_2 = s'_1 + s'_2$, and $(P', Q') \in \mathcal{S}$;

(3) *if* $P \overset{t}{\rightsquigarrow} P'$ *then* $\exists Q' : Q \overset{t}{\rightsquigarrow} Q'$ *and* $(P', Q') \in \mathcal{S}$;

(4) *if* $Q \overset{t}{\rightsquigarrow} Q'$ *then* $\exists P' : P \overset{t}{\rightsquigarrow} P'$ *and* $(P', Q') \in \mathcal{S}$. □

Definition 25 *We say that P and Q are strongly total-time-faster-than relation, written $P \overset{<t}{\sim} Q$, if $(P, Q) \in \mathcal{S}$ for some strong $\overset{<t}{\sim}$-bisimulation $\mathcal{S}$.* □

Here are some properties of $\overset{<t}{\sim}$.

Proposition 20

1. $\overset{<t}{\sim}$ is an partial order relation over the lTeCCS processes, i.e. followings hold.

(1) $P \overset{<t}{\sim} P$ (reflexivity)

(2) $P_1 \overset{<t}{\sim} P_2$ and $P_2 \overset{<t}{\sim} P_1$ then $P_1 = P_2$ (asymmetry)

(3) $P_1 \overset{<t}{\sim} P_2$ and $P_2 \overset{<t}{\sim} P_3$ imply $P_1 \overset{<t}{\sim} P_3$ (transitivity)

2. $\overset{<t}{\sim} \;=\; \bigcup\{\mathcal{S} \mid \mathcal{S}$ is a $\overset{<t}{\sim}$- bisimulation$\}$

proof:

1. For reflexivity, it is enough to show that the identity relation over lTeCCS, that is the relation $\{Id_{lTeCCS}\} = \{(P, P) | P \in lTeCCS\}$, is a $\overset{<^t}{\sim}$-bisimulation. This is obvious.

   For asymmetry, we have to show that if $P_1 \overset{<^t}{\sim} P_2$ and $P_2 \overset{<^t}{\sim} P_1$ then $P_1 = P_2$. From the definition, if $P_1 \overset{<^t}{\sim} P_2$ then $s_1 + s_2 \leq s'_1 + s_{2'}$, also if $P_2 \overset{<^t}{\sim} P_1$ then $s'_1 + s'_2 \leq s_1 + s_2$. Therefore $s_1 + s_2 = s'_1 + s'_2$. From the definition $s_1 + s_2 = s'_1 + s'_2$, so $s_2 = s'_2$. We can get $P_1 = P_2$.

   For transitivity, we must show that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are $\overset{<^t}{\sim}$-bisimulations, then so is their relational composition

   $$\mathcal{S}_1\mathcal{S}_2 \;=\; \{(P, R) \mid \exists Q, P\mathcal{S}_1Q \; and \; Q\mathcal{S}_2R\}.$$

   It is enough to show that this is a $\overset{<^t}{\sim}$- simulation. Let $(P, R) \in \mathcal{S}_1\mathcal{S}_2$, and $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$. Since there exists Q such that $P\mathcal{S}_1Q$ and $Q\mathcal{S}_2R$, there exist also Q' such that $Q \overset{s'_1}{\rightsquigarrow}\xrightarrow{a}\overset{s'_2}{\rightsquigarrow} Q'$ such that $s_1 = s'_1$, $s_2 = s'_2$ and $P'\mathcal{S}_1Q'$, and hence R' such that $R \overset{s''_1}{\rightsquigarrow}\xrightarrow{a}\overset{s''_2}{\rightsquigarrow} R'$ such that $s'_1 = s''_1$, $s'_2 = s''_2$ and $Q'\mathcal{S}_1R'$. So $(P', R') \in \mathcal{S}_1\mathcal{S}_2$, and we have established the $\overset{<^t}{\sim}$- simulation condition for $\mathcal{S}_1\mathcal{S}_2$.

2. Let $P \overset{<^t}{\sim} Q$. Then by definition $P\mathcal{S}Q$ for some $\overset{<^t}{\sim}$-bisimulation $\mathcal{S}$. Therefore if $P \overset{s_1}{\rightsquigarrow}\xrightarrow{a}\overset{s_2}{\rightsquigarrow} P'$, there exists Q' for which $Q \overset{s'_1}{\rightsquigarrow}\xrightarrow{a}\overset{s'_2}{\rightsquigarrow} Q'$ such that $s_1 = s_2$, $s'_1 = s'_2$ and $P'\mathcal{S}Q'$ - hence also $P' \overset{<^t}{\sim} Q'$. Thus $\overset{<^t}{\sim}$ satisfies the $\overset{<^t}{\sim}$- simulation condition and by symmetry so does its converse.

□

It is clear from the similarities in the definitions of $\overset{e}{\sim}$ and $\overset{<^t}{\sim}$ that for P and Q being two terms of lTeCCS, if $P \overset{e}{\sim} Q$ then we can acquire $P \overset{<^t}{\sim} Q$. However the reverse implication does not hold; that is, $P \overset{<^t}{\sim} Q$ does not always imply that $P \overset{e}{\sim} Q$. This is shown in the following Proposition.

Proposition 21 (Strong fairy-faster-than relation, strong total-time-faster-than relati

$$P \overset{<^M}{\sim} Q \; implies \; P \overset{<^t}{\sim} Q$$

□

Definition 26 (Strong $\overset{t}{\lesssim}$-bisimulation up to $\lessapprox$)
A binary relation $\mathcal{S} \subseteq P \times P$ over lTeCCS processes is a strong $\overset{t}{\lesssim}$-bisimulation up to $\lessapprox$ if, for all $(P, Q) \in \mathcal{S}$ and for all $a \in Act$ and for all $s_1, s_1', s_2, s_2', t \in T$, such that $s_1 \leq s_1'$ and $s_1 + s_2 = s_1' + s_2'$

(1) if $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ then $\exists$ s_1', s_2' $\exists Q'$: $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ and $P' \sim \mathcal{S} \sim Q'$;
(2) if $Q \overset{s_1'}{\leadsto}\xrightarrow{a}\overset{s_2'}{\leadsto} Q'$ then $\exists$ s_1, s_2 $\exists P'$: $P \overset{s_1}{\leadsto}\xrightarrow{a}\overset{s_2}{\leadsto} P'$ and $P' \sim \mathcal{S} \sim Q'$;
(3) if $P \overset{t}{\leadsto} P'$ then $\exists Q' : Q \overset{t}{\leadsto} Q'$ and $(P', Q') \in \mathcal{S}$;
(4) if $Q \overset{t}{\leadsto} Q'$ then $\exists P' : P \overset{t}{\leadsto} P'$ and $(P', Q') \in \mathcal{S}$.

□

Proposition 22 *If $\mathcal{S}$ is a strong exact-time bisimulation up to $\lessapprox$ and $P\mathcal{S}Q$, then $P \overset{t}{\lesssim} Q$.*

Theorem 11 (Congruence for *l*TeCCS)
$\overset{t}{\lesssim}$ is a process congruence with respect to the operators of lTeCCS except for parallel operater " $|$". □

Theorem 12 (Congruence for *k*TeCCS)
$\overset{t}{\lesssim}$ is a process congruence with respect to all the operators of kTeCCS.

proof: The proof is similar to that of $\overset{e}{\lesssim}$. □

Chapter 4

Examples

4.1 Timed queue system

4.1.1 asic system of queue system

We would like to build a timed queue system *Queue-system*, which is composed of the three components, *Queue*, *Enque*, *Deque*. The component *Queue* is a body part of *Queue-system*. It has box to accept balls, which is considered to be a capacity of " ", and it stocks balls pushed into. Traditionally, balls are pushed into the *Queue* component with push operation and popped out from it with pop operation. If balls are tried to pushed into the component *Queue* which is full in its capacity, then the component *Queue* throws an overflow exception. Similarly, If balls are tried to pop out from the component *Queue* which is empty in its stock, then the component *Queue* throws an underflow exception. The component *Enque* is a component which pushes 'balls' into the component *Queue*. And the component *Deque* is a component which pops 'balls' out from the component *Queue*. The image of the *Queue-system* composed of the thee components is shown in figure.4.1.

4.1.2 Constructing queue system with untimed CCS

First we construct this system with untimed CCS. The first process to be defined is the *Queue* component. An -ary ueue $Queue^{(n)}(push, pop, overflow, underflow)$ is a process used to express the *Queue* component with a capacity . And the defining equations for -ary ueue is as follows:

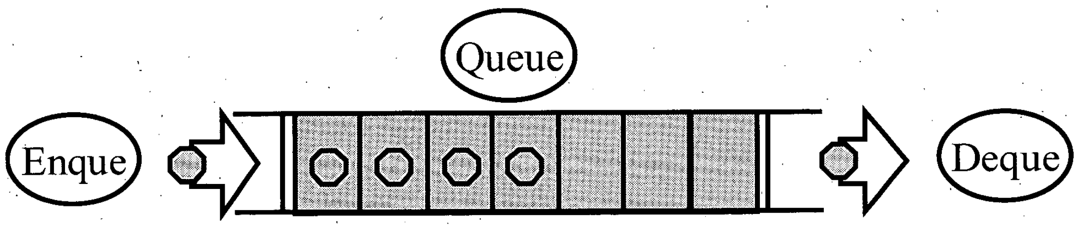

Figure 4.1: Queue system

$$Queue = Queue^{(0)}$$

$$\begin{aligned}
Queue^{(0)} &= \overline{push}.Queue^{(1)} + \overline{pop}.underflow.Error \\
Queue^{(1)} &= \overline{push}.Queue^{(2)} + \overline{pop}.Queue^{(0)} \\
Queue^{(2)} &= \overline{push}.Queue^{(3)} + \overline{pop}.Queue^{(1)} \\
&\cdots\cdots \\
Queue^{(n-1)} &= \overline{push}.Queue^{(n)} + \overline{pop}.Queue^{(n-2)} \\
Queue^{(n)} &= \overline{push}.overflow.Error + \overline{pop}.Queue^{(n-1)}
\end{aligned}$$

A ball is pushed into the component *Queue* by occurrence of the action $\overline{push}$ of $Queue^{(x)}$ process. And a ball in the Queue component is popped out by occurrence of the action $\overline{pop}$ of $Queue^{(x)}$ process. As stated before, this *Queue* component has a capacity of n box to accept balls. When action $\overline{pop}$ occurred in the situation where *Queue* process already is full in its capacity, then it will throw an overflow exception, executing $overflow$ action (see $Queue^{(n)}$ above). In the similar way, when action $\overline{pop}$ occurred in the situation where Queue process is empty in its stock, then it will throw an underflow exception, executing $underflow$ action (see $Queue^{(0)}$ above). These exceptions are ones from process *Queue*, but are also considered as exceptions from whole system of *Queue-system*.

Next we define the *Enque* component which pushes balls into the Queue and the *Deque* component which pops balls out from Queue. This is described as following:

$$Enque = push.Enque$$

$$Deque = pop.Deque$$

Now the whole system *Queue-system*, composed of *Queue*, *Enque* and *Deque*, is following.

$$Queue\text{-}system \;=\; (Queue \mid Enque \mid Deque)\backslash\{push,\ pop\}$$

Here as stated in the chapter of preliminary, *push* and *pop* action are restricted so that we can not access this port or action from the outside. On the other hand, the action $overflow$ and $underflow$ is not restricted so that we can observe the actions from the outside. Therefore we can consider the two action $overflow$ and $underflow$ are the exception from the system *Queue-system*

Here is the flowgraph of *Queue-system*:

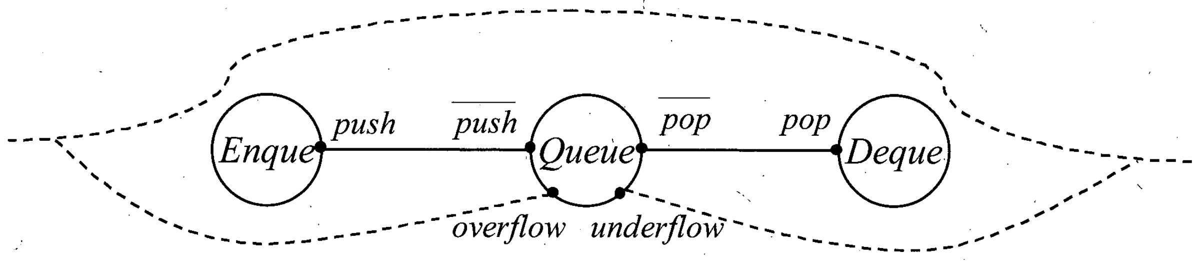

Figure 4.2: Queue flowgraph

An important arena for the use of CCS process descriptions is modeling protocols. This example models a simple protocol for Queue system.

However with untimed CCS, we cannot give a time description directly so that we cannot capture a long term features of systems. In other words, we cannot know the property before and after program execution, which is the property of system. Take the Queue system for example, we cannot tell that this queue system will eventually overflow or underflow and the maximum or minimum number of the balls contained in the Queue. The followings are the formal description of these properties, where the function $\sharp Num(Queue\text{-}system)$ means the number of contents in *Queue-system*

$$Queue\text{-}system \overset{?}{\models} overflow,\ non\text{-}overflow$$

$$\textit{Queue-system} \overset{?}{\models} \textit{underflownon-underflow}$$

$$\sharp \textit{Num}(\textit{Queue-system}) = ?$$

4.1.3 Constructing queue system with timed CCS

In this section, we consider to describe the system with time notion. At first, we consider the process $\textit{Enque}_0$, which is a timed extension with time. For example, we define $\textit{Enque}_0$ as following:

$$\textit{Enque}_0 = (3).\textit{push}.(3).\textit{Enque}_0$$

$\textit{Enque}_0$ process is a process which starts with waiting for passing three units of time, executes the action *push* and waits another three units of time, and transits into $\textit{Enque}_0$ process again. Also we define the timed process $\textit{Deque}_0$, which is also a process *Enque* extended with time notion.

$$\textit{Deque}_0 = (3).\textit{pop}.(3).\textit{Deque}_0$$

$\textit{Deque}_0$ process behaves like $\textit{Enque}_0$ except for their functional behaviour. That is, $\textit{Deque}_0$ process is identical with $\textit{Enque}_0$ process in its timing of actions, but different in its actions; $\textit{Deque}_0$ process is a process which starts with waiting for passing three units of time, executes the action *pop* and waits another three units of time, and transits into $\textit{Deque}_0$ process again. We substitute two process *Enque* and *Deque* in *Queue-system* for two processes of $\textit{Enque}_0$ and $\textit{Deque}_0$ respectively. And we name the new system acquired from the substitution as $\textit{Queue-system}_0$. This is formally written as follows;

$$\textit{Queue-system}_0 = (\textit{Queue} \mid \textit{Enque}_0 \mid \textit{Deque}_0)\backslash\{\textit{push},\ \textit{pop}\}$$

Accordingly this new system $\textit{Queue-system}_0$ is a time extension version of *Queue-system*. Constructing the system like this, we can see some of the property of systems concerned with time notion, which we cannot tell in untimed system. The followings are the properties acquired here.

$$\textit{Queue-system}_0 \models \textit{non-overflow},\ \textit{non-underflow}$$

$$\sharp \textit{Num}(\textit{Queue-system}_0) = n \qquad (\textit{n is a a constant})$$

Namely, we can tell that this timed system $\textit{Queue-system}_0$ will neither overflow nor underflow throughout the program run. And we can guarantee the number of balls in the Queue system is a constant. This is because the timing of the action *enque* and *deque* is completely identical.

Here we will see some systems, where one of components of the system is substituted. That is to say, we will consider systems whose *Enque* is substituted for a process related with each of the relations proposed in the former chapter. And we also see properties of the systems, which are acquired by substituting *Enque* process for another process, which is in a relation proposed in the former chapter.

Exact-time equivalence

At first we consider to substitute *Enque* process for the process *Enque* process which is related with exact-time equivalence. Then we name the new queue system acquired by the substituting *Enque* process for *Enque* process as $Queue - system_1$. This is formally written as following.

$$Enque_1 \overset{e}{\sim} Enque_0$$

$$Queue\text{-}system_1 \; = \; (Queue \mid Enque_1 \mid Deque_0)\backslash\{push,\ pop\}$$

As shown in theorem previously in second chapter, exact-time equivalence is preserved between the system $Queue - system_1$ and the system $Queue_system_0$.

$$Queue\text{-}system_1 \; \overset{e}{\sim} \; Queue\text{-}system_0$$

Also we can capture some property of this system $Queue - system_1$ as followings .

$$Queue\text{-}system_1 \models non\text{-}overflow,\ non\text{-}underflow$$

$$\sharp Num(Queue\text{-}system_1) = \sharp Num(Queue\text{-}system_0)$$

The first line of expressions above means that this system $Queue - system_1$ will neither overflow nor underflow in long term systems run. In other words, $Queue - system_1$ will throw neither overflow exception nor underflow exception. This is because between the processes which is in exact-time equivalence, they are designed to be equivalent in their total-time for a transition; the time required for every transition is same between two processes in exact-time equivalence. This is a guarantee of safety property, which states that nothing bad ever happens. A process has a safety property just in case no run from it contains the bad feature. Here bad feature means throwing the overflow and underflow exceptions. The second line of the expressions above denotes that the number of balls stocked in $Queue - system_1$ is equivalent to that of $\mathit{Queue\text{-}system}_0$. It is because between two processes in exact-time equivalence, the timing of actions in every transition is all same; the actions of two processes in the relation of exact-time equivalence occurs at the same time. Due to this, we can tell that the number of balls stocked in $Queue - system_1$ is equivalent to that of $\mathit{Queue\text{-}system}_0$.

Action-occurrence-faster-than relation

Secondly we consider to substitute $Enque_0$ process in the system $Queue - system_0$ for $Enque_2$ process which is in action-occurrence-faster-than relation and faster than $Enque_0$ process. We call the new system acquired by the substitution as $\mathit{Queue\text{-}system}_2$. We will see the formalization of these as follows;

$$Enque_2 \overset{ac}{\lesssim} Enque_0$$

$$\mathit{Queue\text{-}system}_2 = Queue \mid Enque_2 \mid Deque_0$$

As shown in theorem in second chapter, action-occurrence-faster-than relation is preserved between the system $Queue - system_2$ and the system $Queue - system_0$.

$$\mathit{Queue\text{-}system}_2 \overset{ac}{\lesssim} \mathit{Queue\text{-}system}_0$$

In this case, we can capture some properties below.

$$\mathit{Queue\text{-}system}_2 \models \mathit{non\text{-}overflow, non\text{-}underflow}$$

$$\sharp Num(\mathit{Queue\text{-}system}_0) \leq \sharp Num(\mathit{Queue\text{-}system}_2) \leq \sharp Num(\mathit{Queue\text{-}system}_0) + 1$$

In this case as a long term feature of systems, $\mathit{Queue\text{-}system}_2$ will throw neither overflow exception nor underflow exception, either. This is because between two processes in action-occurrence-faster-than relation the time required for every transition is the same, so that we can tell that the frequency of *push* action and that of *pop* action is same. However, timing of action occurrence is different. That is, the action occurrence of the faster process preempt that of slower process. So here because *push* actions always have a possibility of preempting *pop* actions, although frequency of occurrence of the two actions, the number of balls in *Queue-system*$_0$ can be larger than that of *Queue-system*$_2$ from time to time.

Total-time equivalence

As a third case, we consider to substitute $Enque_0$ process in the system $Queue{-}system_0$ for $Enque_3$ process which is in total-time equivalent relation. We call the new system acquired by the substitution of $Enque_0$ for $Enque_3$ as *Queue-system*$_3$. These are formally written like these;

$$Enque_3 \overset{t}{\sim} Enque_0$$

$$Queue\text{-}system_3 \;=\; Queue \mid Enque_3 \mid Deque_0$$

As shown in theorem in second chapter, total-time equivalence is preserved in between the system $Queue - system_3$ and the system $Queue - system_0$. This is important.

$$Queue\text{-}system_3 \overset{t}{\sim} Queue\text{-}system_0$$

In this case, we can acquire properties below as a property of the system.

$$Queue\text{-}system_3 \models \textit{non-underflow, non-overflow}$$

$$\sharp Num(Queue\text{-}system_0) - 1 \leq \sharp Num(Queue\text{-}system_3) \leq \sharp Num(Queue\text{-}system_0) + 1$$

In this case as a long term feature of systems, *Queue-system*$_2$ will throw neither overflow exception nor underflow exception, either. The reason is same as that of exact-time equivalence, action-occurrence-faster-than relation, which are already shown. However, total-time equivalence does not provide condition about the timing of action in transitions of processes so that we cannot tell the property of timing of actions of processes. That is, the timing of occurrence of *push* action and *pop* action so that sometimes *push* action preempts *pop* action and sometimes vice-versa. When *push* action preempts *pop* action, the number of balls in the *Queue-system*$_2$ might be temporally equal to that of $(Queue\text{-}system_0 + 1)$. When *pop* action preempts *push* action, the number of balls in the *Queue-system*$_2$ might be temporally equal to that of $(Queue\text{-}system_0 - 1)$.

After-faster-than relation

Next we consider to substitute $Enque_0$ process in the system $Queue\text{-}system_0$ for $Enque_4$ process which is faster than $Enque_0$ with respect to after-faster-than relation. We call the new system acquired by the substitution of $Enque_0$ for $Enque_4$ as $Queue\text{-}system_4$. The formalization are the following;

$$Enque_4 \overset{af}{\lesssim} Enque_0$$

$$Queue\text{-}system_4 = Queue \mid Enque_4 \mid Deque_0.$$

From theorem in second chapter we can see after-faster-than relation is preserved in between the system $Queue - system_4$ and the system $Queue - system_0$.

$$Queue\text{-}system_4 \overset{af}{\lesssim} Queue\text{-}system_0$$

In this case, we can acquire the followings as a property of the system.

$$Queue\text{-}system_4 \not\models non\text{-}overflow$$

$$Queue\text{-}system_4 \models non\text{-}underflow$$

$$\sharp Num(Queue\text{-}system_0) \leq \sharp Num(Queue\text{-}system_4) \leq overflow$$

Here we cannot tell this system $Queue\text{-}system_4$ will not overflow as a long term feature of the system. More directly, this system $Queue\text{-}system_4$ might throw an overflow exception. This is because between two processes in the after-faster-than relation the required timed for a transition are not always equivalent. This relation defines that the time required for each transition of the faster process is shorter than that of the slower process. This means here that the frequency of *push* action occurrence of $Enque_4$ might be higher than that of $Enque_0$. Accordingly, the frequency of *push* action occurrence of $Enque_4$ might be higher than *push* action occurrence of $Deque_0$. So $Queue\text{-}system_4$ might overflow.

Fairly-faster-than relation

Here, we consider to substitute $Enque_0$ process in the system $Queue\text{-}system_0$ for $Enque_5$

process which is faster than $Enque_0$ with respect to fairly-faster-than relation. We call the new system acquired by the substitution of $Enque_0$ for $Enque_5$ as $Queue\text{-}system_5$. The formalization are like these;

$$Enque_5 \overset{\lll}{\sim} Enque_0$$

$$Queue\text{-}system_5 \;=\; Queue \mid Enque_5 \mid Deque_0$$

From theorem in second chapter we can see fairly-faster-than relation is preserved in between the system $Queue\text{-}system_5$ and the system $Queue\text{-}system_0$.

$$Queue\text{-}system_5 \;\overset{\lll}{\sim}\; Queue\text{-}system_0$$

In this case, we can acquire the followings as a property of the system $Queue\text{-}system_5$.

$$Queue\text{-}system_5 \;\not\models\; non\text{-}overflow$$

$$Queue\text{-}system_5 \;\models\; non\text{-}underflow$$

$$\sharp Num(Queue\text{-}system_0) \leq \sharp Num(Queue\text{-}system_5) \leq overflow$$

This system $Queue\text{-}system_5$ might throw an overflow exception. The reason is same as that in after-faster-than relation, which we saw just before. Also there are no situation where the number of balls in $Queue\text{-}system_5$ is smaller than that of $Queue\text{-}system_0$. This is because the fairly-faster-than relation ensures that the faster process is faster in both timing of every action occurrence and time required for every transition. So from the first action the occurrence of *enque* action might preempt and at least synchronize with the occurrence of *deque* action. Needless to say the maximum number of the balls to be stocked in $Queue\text{-}system_5$ is the maximum balls which $Queue\text{-}system_5$ can stock.

Total-time-faster-than relation

Finally, we consider to substitute $Enque_0$ process in the system $Queue\text{-}system_0$ for $Enque_6$ process which is faster than $Enque_0$ with respect to total-time-faster-than relation. We call the new system acquired by the substitution of $Enque_0$ for $Enque_6$ as $Queue\text{-}system_6$. The formalization are like these;

$$Enque_6 \overset{t}{\lesssim} Enque_0$$

$$\mathit{Queue\text{-}system}_6 \doteq \mathit{Queue} \mid \mathit{Enque}_6 \mid \mathit{Deque}_0$$

From theorem in second chapter we can see total-time-faster-than relation is preserved in between the system *Queue-system*$_6$ and the system *Queue-system*$_0$.

$$\mathit{Queue\text{-}system}_6 \overset{t}{\lesssim} \mathit{Queue\text{-}system}_0$$

In this case, we can acquire the followings as a property of the system *Queue-system*$_6$.

$$\mathit{Queue\text{-}system}_6 \not\models \mathit{non\text{-}overflow}$$

$$\mathit{Queue\text{-}system}_6 \models \mathit{non\text{-}underflow}$$

$$\sharp \mathit{Num}(\mathit{Queue\text{-}system}_0) - 1 \leq \sharp \mathit{Num}(\mathit{Queue\text{-}system}_6) \leq \mathit{overflow}$$

This system *Queue-system*$_5$ might throw an overflow exception. The reason is same as that in after-faster-than relation and in fairly-faster-than relation. However, total-time-faster-than relation does tell nothing about the condition about the timing of action in transitions of processes, which is similar to total-time equivalence in this respect so that we cannot tell the property of timing of actions of processes. That is, the timing of occurrence of *push* action and *pop* action so that sometimes *push* action preempts *pop* action and sometimes vice-versa. When *pop* action preempts *push* action, especially in the earlier stage, there is a possibility that the number of balls in the *Queue-system*$_2$ might be temporally equal to that of (*Queue-system*$_0$ − 1).

4.1.4 Summary

The properties of the systems are summarized in the following table.

No.	relations	symbol	non-UF	non-OF	$\sharp Num$
1.	exact-time equivalent	$\overset{e}{\sim}$	$\surd$	$\surd$	$n = \sharp Num$
2.	action-occurrence-faster-than	$\overset{ac}{\lesssim}$	$\surd$	$\surd$	$n \leq \sharp Num \leq n+1$
3.	total-time equivalent	$\overset{t}{\sim}$	$\surd$	$\surd$	$n-1 \leq \sharp Num \leq n+1$
4.	after-faster-than	$\overset{af}{\lesssim}$	$\surd$		$n\mathit{overflow}$
5.	fairly-faster-than	$\lessapprox$	$\surd$		$n\mathit{overflow}$
6.	total-time-faster-than	$\overset{t}{\lesssim}$	$\surd$		$n-1\mathit{overflow}$

					$\sharp Num$	
No.	relations	symbol	non-UF	non-OF	min	max
1.	Et equivalent	$\overset{e}{\sim}$	$\surd$	$\surd$	n	n
2.	Ac-faster than	$\lesssim^{ac}$	$\surd$	$\surd$	n	$n+1$
3.	Tt equivalent	$\overset{t}{\sim}$	$\surd$	$\surd$	$n-1$	$n+1$
4.	Af-faster than	$\lesssim^{af}$	$\surd$		n	*overflow*
5.	F-faster than	$\overset{\ll}{\sim}$	$\surd$		n	*overflow*
6.	Tt-faster than	$\lesssim^{t}$	$\surd$		$n-1$	*overflow*

Chapter 5

Conclusions and Future Work

In this research, we considered some (bi)simulation-based relations between processes rTeCCS[MT91] which is one of timed process algebra(TPA)s, taking a real-time property into consideration. We attempted to reflect real-time property into the discussion of relating processes in rTeCCS[MT91]

We decided that we attempt to reflect real-time property into the discussion of relating processes in rTeCCS. Although some calculi of process algebra with time notion proposed, we noticed that a missing work among them is study on the discussion of relating processes under the time notion. The relating systems of processes in these works are straightforward; that is, time property is ignored in the discussion of relating processes. We claimed that we need to take time property more into consideration in timed system. We noticed real-time property played an important role in practical timed systems from an engineering point of view. Consequently we decided to attempt to reflect real-time property into the discussion of relating processes in TPA; we relate processes which are identical in functional behaviour but operate at different speed. That is, we related processes with respect to speed.

We discussed the issue on the calculus of rTeCCS, which is one of TPAs. In considering relation, we construct the relation within simulation relation which is invented by Milner. The simulation relation is a fine notion to identify concurrent processes so that we inherit it and construct the relating systems of processes within the framework of simulation relation. By manipulating the definition of simulation relation, we have to reflect real-time property into the relating processes. Also we have to examined processes of rTeCCS and consider how we can reflect real-time property into the relation of rTeCCS on the processes of rTeCCS. Consequently we found two places which can express the speed in processes of lTeCCS; One kind of speeds is one based on the timing of the occurrence of action execution, and the other is one based on the time required for whole action execution. Also we strictly distinguish the notion of the "faster" transition expressed by required time is shorter, and the "synchronous" transition expressed by required time is equal. As a result with a combination of two places and two kinds of speeds, we propose six kinds of speed, hence six relations of processes that reflect real-time property in TPA. These six relations to be proposed here all express the different aspects of real-time property. And We explain relation of these relations by descriptive capability in order. We show that bisimulation relation and the Moller's faster-than relation, that are proposed in preceding researches, are respectively included by one of relations of these.

We conduct formalization of each relation proposed in the former section. We also show Properties of each relation, including congruence, are also given with the proofs. Concerning the congruence property, we show that these relations are not congruent for lTeCCS in the presence of parallel operator "|" So we proposed a new timed calculus k TeCCS, whose semantics is slightly diverse from that of lTeCCS, and show that theses relation are congruent for all the operators of it.

In the fourth chapter, we showed a timed queue system as an example of the six relations. Through this example, we saw the difference of the six relations in the concrete.

It is a substantial result to consider to reflect real-time property, which is one of main roles in timed systems, into the discussion of relating processes in TPA. We examined and captured real-time property, and reflected it within the framework of TPA; labelled transition systems of lTeCCS and the simulation relations of process algebra. Ac a result we found and propose six speeds in the framework, therefore six relation , which reflect real-time property. Each of six relations shows different aspect of real-time property within lTeCCS. By the classification of speed, we can arrange the processes which is in the relation of speed. Also to show the congruence property, we guaranteed substituibity between process in systems. And this enable us to modular-analysis of complex systems in foundation level. The theory for the modular analysis will offer a foundation for the technology of component oriented software development. Furthermore, we brought rich theory to timed process algebras in general; this theory will apply for another timed calculus. The theory in this research is effective to the specification with process-algebra-like LTS as well as TPA itself. Also we can directly use this theory to programming languages based on process algebra such as Pict.

As future work, we would like to consider a congruence property of calculus. In this work we showed that a congruence property is not guaranteed by the parallel composition operator "|"of lTeCCS. Thereupon we manipulate the semantics the parallel composition operator "|"of lTeCCS and proposed kTeCCS. We show that we can acquire the congruence property in the calculus of kTeCCS. However the calculus of kTeCCS does not purely realize the concurrency, meaning this is more or less artificial. As a future work, we want to solve this problem; we want to built a pure theory for TPAs. Secondly, we intend to extend our theory to weak relation of processes. Also, we want to enrich the theory of these relations with algebraic theory. Thirdly, we want to attempt to apply this theory to a network theory. The treatment of time notion in lTeCCS is similar to that of transmission delay in the network. We view we can develop verification system for a problem of which is the faster between RPC(Remote Procedure Call)s and agent migrations. Furthermore we will develop verification and implementation tool for this theory.

References

[AP94] R.Amadio and S.Prasad. Localities and Failures.In Proc.of FST & TCS,volume 880 of LNCS,p.205-216,1994.

[BK85] J.A. Bergstra J.W. Klop, Algebra of Communicating Processes with Abstraction", Theoretical Computer Science 37(1):77-121 1985

[Ber86] J.A Bergstra,J.W. Klop. Algebra of communicating processes. Proceedings of CWI Symposium on Mathematics and CS, pp.89-138, Oct.6-7 1986.

[CG98] L. Cardelli and A.D.Gordon. Mobile ambients. In Proc.of FoSSaCS, volume 1378 of LNCS,p. 140-155,1998.

[CG00] L.Cardelli and A.D.Gordon. Anytime,anywhere:Modal logics for mobile ambients. In Proc. of POPL,2000.

[Fou98] C.Fournet. Le join-calcul:un calcul pour la programmation repartie et mobile.PhD thesis,Ecole Polytechnique,1998(in english).

[HT91] Kohei Honda and Mario Tokoro.An Object Calculus for Asynchronous Communication. Proc.ECOOP'91,LNCS 512,pp.133–147,Springer-Verlag,1991. Available at ftp://ftp.dcs.ed.ac.uk/export/kohei/objcal.ps.gz.

[Hon] K.Honda.Selected bibliography on mobile processes. http://www.cs.auc.dk/mobility/bib/honda.html.

[HR95] M.Hennessy and T.Regan. A process algebra for timed systems. Information and Computation 117. 1995

[Hoa78] C.A.R.Hoare. Communicating Sequential Processes. Communications of the ACM, Vol.21, No.8, pp.666-667, August 1978.

[Hoa85] C.A.R.Hoare. Communicating Sequential Processes. Prentice-Hall International series in computing science,1985.

[HR98] M.Hennessy and J.Riely. Resource access control in systems of mobile agents. In Proc. of HLCL,volume 16.3 of Electronic Notes in Theoretical Computer Science,1998

[JD93] Milner,R.,Parrow,J.,and Walker,D.Modal Logics for Mobile Processes.Theoretical Computer Science,Vol.114,pp.149-171,1993.

[LV01] G.Luttgen and W.Vogler. A faster-than relation for asynchronous processes. In CONCUR '01, LNCS 2154, Springer-Verlag. 2001.

[MT90] F.Moller and C.Tofts. A temporal calculus of communicating systems. In CONCUR '90, LNCS 458, pp401-415. Springer-Verlag. 1990.

[MT91] F.Moller and C.Tofts. Relating processes with respect to speed. In CONCUR '91, LNCS 527, pp424-438. Springer-Verlag. 1991.

[Mil89] R.Milner. Communication and Concurrency. Prentice Hall. 1989.

[Mil99] R.Milner. Communicating and Mobile Systems: the π-calculus. Cambridge University Press. 1999.

[Mil83] R.Milner. Calculi for synchrony and asynchrony. Theoretical Computer Science. 25 1983.

[Mil93] Robin Milner.Elements of Interaction. Communication of ACM,January 1993.

[MPD89] R.Milner, J.Parrow, D.Walker. A Calculus of Mobile Process , Part 1/2. Available from http://lampwww.epfl.ch/mobility/ 1989.

[MS92] Robin Milner and Davide Sangiorgi. Barbed Bisimulation.Proc.of 19th international Colloquium on Automata,languages and Programming (ICALP'92),Lecture Notes in Computer Science,Vol.623,pp.685-695,Springer-Verlag,1992. Available at http://www.inria.fr/meije/personnel/Davide.Sangiorgi/mypapers.html.

[Mil93] R.Milner.Elements of interaction.Communications of the ACM,36(1):78-89,1993.

[Mil] Milner,Communication and Concurrency,Prentice-Hall.

[NV98] U. Nestmann and B. Victor. Calculi for mobile processes:Bibliography and web pages. Bulletin of the EATCS,64:139-144,1998.

[NS90] X. Nicolin and J. Sifakis. The algebra for timed processes ATP: theory and application. In proceedings of REX Workshop "Real-Time: Theory in Practice". 1990.

[NS91] X. Nicollin and J. Sifakis. An overview and synthesis on timed process algebras. LNCS 575, Springer-Verlag. 1991.

[PS93] Benjamin Pierce and Davide Sangiorgi,Typing and Subtyping for Mobile Processes.LICS'93,July 1993. Available at http://www.cs.indiana.edu/hyplan/pierce/pierce/ftp/index.html.

[PT97] Benjamin C.Pierce and David N.Turner.Pict:A Programming Language Based on the Pi-Calculus.1997,Computer Science Department,Indiana University,To appear in Milner Festschrift,MIT Press,1997. Available at http://www.cs.indiana.edu/hyplan/pierce/pierce/ftp/index.html.

[Pie] Benjamin C.Pierce.Programming in Pi-Calculus:A Tutorial Introduction to Pice. Available at http://www.cs.indiana.edu/hyplan/pierce/ftp/pict/pict-4.0/Doc/tutorial.ps.gz.

[PS96] B.C.Pierce and D.Sangiorgi.Typing and subtyping for mobile processes.Mathematical Structures in Computer Science,6(5):409-453,1996

[Pie96] B.C.Pierce. Foundational calculi for programming languages. In A.B.Tucker,editor,Handbook of Computer Science and Engineering,chapter 139.CRC Press,1996.

[Par] Joachim Parrow. An Introduction to the Π-Calculus.

[PW93] R.Milner,J,Parrow,and D.Walker. Modal logics for mobile processes.Theoretical Computer Science,114(1):149-171,1993.

[Sch91] S.Schneider. An operational semantics for timed CSP. Programming research group, Oxford University, UK, 1991

[Sti96] C.Stirling. Modal and Temporal Logics for Processes, 1996

[Wan90] Wang Yi. Real-time behaviour of asynchronous agents. In CONCUR '90, LNCS 458. , pp502-520, Springer-Verlag. 1990.