UNIVERSITE D'ORLEANS
*Faculté des Sciences*
**LIFO**

Laboratoire d'Informatique Fondamentale d'Orléans
4, rue Léonard de Vinci, BP 6759
F-45067 Orléans Cedex 2
FRANCE

Rapport de Recherche

www : http://www.univ-orleans.fr/SCIENCES/LIFO/

# A Proof System for Weak Congruence in Timed $\pi$ Calculus

Jing CHEN, LIFO, Orléans (Fr.)

Rapport N$^{o}$ **2004-13**

# A Proof System for Weak Congruence in Timed $\pi$ Calculus

Jing Chen*
Laboratoire d'Informatique Fondamentale d'Orléans
CNRS - Université d'Orléans, France
E-mail: chen@lifo.univ-orleans.fr

**Abstract**

A complete axiomatization for weak congruence in the recursion-free fragment of timed $\pi$ calculus is presented, and its soundness and completeness are shown. The system consists of some basic algebraic axioms together with several inference rules. The system is first designed for *late* bisimulation, and can be modified for *early* bisimulation via simply replacing the rule concerning input action. The proof of the completeness result relies on the theory of *symbolic bisimulation*.

**Keywords: Process Algebra, Mobile, Real-time, Bisimulation, Axiomatization, Symbolic Bisimulation**

## 1 Introduction

This paper aims to provide a complete proof system for weak congruence in the Timed $\pi$ calculus, which, developed in [3] and [4], is a calculus for real time mobile systems. The calculus is a timed extension of the standard $\pi$ [12].

Nowadays, real time is an important feature of many systems, but the real time calculi such as Timed CCS [18], Timed CSP [15] and etc. can not be used for mobile systems, and the current timed extensions for $\pi$ as in [5] and [2] are based on discrete time domain and have no new theory of bisimulations with respect to time.

A main area in process algebra is to study equivalence relations between processes, where proof systems and algorithms are developed to analyze the relations. Bisimulation is a commonly used equivalence relation in Concurrency field. The basic theory of bisimulations of $\pi$ calculus was presented in [12] and [13], and Barbed Bisimulation, which was developed for high order processes, has been presented and studied in [14]. Open Bisimulation has been studied in [16], and other important results can be seen in [17]. In this paper, we are going to axiomatize timed bisimulation, i.e., bisimulation with respect to time.

There is a usual distinction between *Strong* and *Weak* bisimulations depending on whether the internal action $\tau$ is given the same weight as any other actions. In weak versions of bisimulations, $\tau$ is abstracted away, and is always treated invisible, therefore it doesn't contribute in the matching step of weak bisimulations. As in standard $\pi$[12] and CCS[11], weak bisimulation is not preserved by *Choice* operator '+', hence we have to look for a suitable observation congruence based on the bisimulation relation. In $\pi$ calculus, another distinction between *Late* and *Early* equivalences [12, 13] can be made, according to when input names are instantiated. And in this paper, we will focus on *Late* equivalences and we will show how *Early* equivalences can be handled. Another point of dealing with *Late Weak* equivalences is that, we have to be careful in defining *Weak transitions* which contain arbitrarily many $\tau$ actions: It was shown in [17], due to Milner, input moves should not be allowed to absorb the $\tau-$moves *after* them, for otherwise the resulted relation would not be transitive.

In [3], a semantics theory of Timed $\pi$ calculus with respect to timed bisimulations has been studied, with a complete axiomatization of strong bisimulation over a recursion-free fragment

*Supported by National Science Foundation of China, and by **le_STUDIUM**, `http://lestudium.cnrs-orleans.fr/`, Orléans

of the calculus. In this paper, based on the *Symbolic Bisimulation* theory [6, 7, 8], a complete axiomatization for weak congruence will be presented. The judgments of the inference system are as those in [8] and [10], which are of the form $C \triangleright E = F$, where $C$ is a set of equality or inequality tests among names, and this conditional equation asserts that $E$ and $F$ are equivalent with respect to condition $C$.

It has been shown that, in [11] and [10], weak congruences can be axiomatized by adding three $\tau-$Laws into the axiom systems for strong bisimulations in CCS and $\pi$. And in this timed case, we also have three $\tau-$Laws where two of them coincide with CCS $\tau-$laws, yet one CCS $\tau-$Law needs to be strengthened as:

$$\alpha.\sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i = \alpha.[\sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i + \epsilon(e).\tau.\sum_{i\in I} M_i\epsilon(e_i \dot{-} e).\alpha_i.E_i]$$

where $M_i$ is the standard *Match* condition in $\pi$, and $\epsilon(d)$ is the time delay prefix. When all these $M_i$ and $\epsilon(d)$ are dropped, this law will become $\alpha.E = \alpha.(E + \tau.E)$, which, together with another CCS $\tau-$law $\tau.E + E = \tau.E$, will imply $\alpha.E = \alpha.\tau.E$. Therefore all of those *standard* $\tau-$laws can be implied by our timed $\tau-$laws.

The paper is organized as follows: The basic definitions of timed $\pi$ and bisimulations are restated in the next section. In section 3, after the basic notations of symbolic technique are introduced, the axioms and inference rules will be presented, with soundness and completeness proved. The paper is concluded in section 4.

# 2 Timed $\pi$ Calculus and Bisimulations

## 2.1 The Language and Operational Semantics

Let $\mathcal{N}$ be a countably infinite set of *names*, ranged over by $x, y, u, v \ldots$. For any $A \subseteq \mathcal{N}$, $\overline{A} = \{\overline{x} | x \in A\}$, and $\overline{\overline{x}} = x$ for all $x \in \mathcal{N} \cup \overline{\mathcal{N}}$. $\mathcal{R}^{\geq 0}$ is the set of non-negative real numbers while $\mathcal{R}^{>0} = \mathcal{R}^{\geq 0} \setminus \{0\}$. Let $\mathcal{C}$ be the set of clock variables, and for any $e, f \in \mathcal{R}^{\geq 0}$,

$$e \dot{-} f = \begin{cases} 0 & e \leq f \\ e - f & \texttt{otherwise} \end{cases}$$

$\delta_{\mathcal{R}^{\geq 0}} = \{\epsilon(p) | p \in \mathcal{R}^{\geq 0}\}$ is the set of time delay actions where $p$ is the length of $\epsilon(p)$. The language is defined by the following BNF grammar:

$$\begin{array}{lll} E & ::= & \mathbf{0} \mid \alpha.E \mid \epsilon(e).E \mid \nu x E \mid [x = y]E \mid E|E \mid E + E \mid A(\tilde{v}) \\ \alpha & ::= & \tau \mid \overline{x}y \mid x(y) \end{array}$$

Most of these operators are from $\pi$ calculus[12]: $\mathbf{0}$ is an empty process, $\alpha$ stands for actions such as $\bar{x}y$ (output), $x(y)$ (input) and internal action $\tau$. Let $\mathcal{L}$ be the set of all actions ranged over by $\eta$, while $\Lambda = \mathcal{L} \setminus \delta_{\mathcal{R}^{\geq 0}}$ is ranged over by $\alpha$, $\beta \ldots$. For an action in $\Lambda \setminus \{\tau\}$, the *subj*ect and *obj*ect can be defined by: $subj(\overline{x}y) = \overline{x}, subj(x(y)) = x, obj(\overline{x}y) = obj(x(y)) = y$. Bound names are introduced by input prefix $a(x).E$ and restriction $\nu x E$. The set of bound names and free names of $E$ are written $bn(E)$ and $fn(E)$ respectively. For actions, we have: $fn(\overline{x}y) = \{x, y\}$, $fn(x(y)) = \{x\}$, $bn(x(y)) = \{y\}$, $bn(\overline{x}y) = \emptyset$, and for $\eta \in \{\tau\} \cup \delta_{\mathcal{R}^{\geq 0}}$, $fn(\eta) = bn(\eta) = \emptyset$. Let name set $n(E) = fn(E) \cup bn(E)$ and $n(\eta) = fn(\eta) \cup bn(\eta)$. Each *identifier* $A$ has a defining equation $A(\tilde{x}) \stackrel{def}{=} E$ associated. Recursion could be introduced by those defining equations, but in this paper, for axiomatization, we only consider recursion-free equations. *Subst* is the set of substitutions over names. For $\sigma \in Subst$, let $supp(\sigma) = \{x | x\sigma \neq x\}$, $cosupp(\sigma) = \{x\sigma | x \in supp(\sigma)\}$, and $n(\sigma) = supp(\sigma) \cup cosupp(\sigma)$ be the name set of $\sigma$. Naturally, for any $x \in \mathcal{N}$, we have $\overline{x}\sigma = \overline{x\sigma}$, and $\eta\sigma = \eta$ for any $\eta \in \{\tau\} \cup \delta_{\mathcal{R}^{\geq 0}}$. Processes equivalent up to $\alpha-$conversion are deemed to be the same and we will write $\equiv$ for both syntax equivalence and $\equiv_\alpha$.

The operational semantics of the language is reported mainly in Fig. 1, where a transition is of the form $P \xrightarrow{\eta} P'$ with $\eta \in \mathcal{L}$.

$$Pre:\ \frac{}{\alpha.E\xrightarrow{\alpha}E} \qquad\qquad Idle:\ \frac{}{\alpha.E\xrightarrow{\epsilon(d)}\alpha.E}\ \alpha\neq\tau$$

$$Delay_0:\ \frac{}{\epsilon(e+d).E\xrightarrow{\epsilon(d)}\epsilon(e).E} \qquad\qquad Delay_1:\ \frac{E\xrightarrow{\epsilon(d)}E'}{\epsilon(e).E\xrightarrow{\epsilon(e+d)}E'}$$

$$Delay_2:\ \frac{E\xrightarrow{\eta}E'}{\epsilon(0).E\xrightarrow{\eta}E'} \qquad\qquad Idle_0:\ \frac{}{\mathbf{0}\xrightarrow{\epsilon(d)}\mathbf{0}}$$

$$Sum:\ \frac{E\xrightarrow{\alpha}E'}{E+F\xrightarrow{\alpha}E'} \qquad\qquad Sum-Delay:\ \frac{E\xrightarrow{\epsilon(d)}E'\quad F\xrightarrow{\epsilon(d)}F'}{E+F\xrightarrow{\epsilon(d)}E'+F'}$$

$$Par:\ \frac{E\xrightarrow{\alpha}E'}{E|F\xrightarrow{\alpha}E'|F}\ bn(\alpha)\cap fn(F)=\emptyset \qquad\qquad Com:\ \frac{E\xrightarrow{\overline{x}y}E'\quad F\xrightarrow{x(z)}F'}{E|F\xrightarrow{\tau}E'|F'\{y/z\}}$$

$$Cond_1:\ \frac{E\xrightarrow{\eta}E'}{[x=x]E\xrightarrow{\eta}E'} \qquad\qquad Cond_2:\ \frac{}{[x=y]E\xrightarrow{\epsilon(d)}[x=y]E}\ x\neq y$$

$$Open:\ \frac{E\xrightarrow{\overline{x}y}E'}{\nu yE\xrightarrow{\overline{x}(w)}E'\{w/y\}}\ y\neq x,\ w\notin fn(\nu yE') \qquad\qquad Close:\ \frac{E\xrightarrow{\overline{x}(w)}E'\quad F\xrightarrow{x(w)}F'}{E|F\xrightarrow{\tau}\nu w(E'|F')}$$

$$Res:\ \frac{E\xrightarrow{\eta}E'}{\nu xE\xrightarrow{\eta}\nu xE'}\ x\notin n(\eta) \qquad\qquad Abs:\ \frac{E\{\tilde{v}/\tilde{x}\}\xrightarrow{\eta}E'}{A(\tilde{v})\xrightarrow{\eta}E'}\ A(\tilde{x})\stackrel{def}{=}E$$

Fig. 1: Operational Semantics of Timed $\pi$

where the symmetrical rules for '+' and '|' (which are commutative) are omitted, and bound output action $\bar{x}(w)$ is introduced only by *Open*, with $bn(\bar{x}(w)) = \{w\}$, $fn(\bar{x}(w)) = \{x\}$, $subj(\bar{x}(w)) = \bar{x}$ and $obj(\bar{x}(w)) = w$. As for the semantics about time delay behavior of a parallel composition, to ensure the *Maximal Progress* property [3, 4], we need a side condition:

$$\boxed{Par-Delay:\ \frac{E\xrightarrow{\epsilon(d)}E'\quad F\xrightarrow{\epsilon(d)}F'}{E|F\xrightarrow{\epsilon(d)}E'|F'}\quad [Sort_d(E)\cap\overline{Sort_d(F)}=\emptyset]}$$

where $Sort_d(E)$ is the subject set of the actions that $E$ can perform within $d$ units of time. Intuitively the side condition guarantees that, within $d$ units of time, no communication between $E$ and $F$ can happen (*i.e.* no $\tau$ action can be generated).

Let $Sort_0(E) = \emptyset$. Then $Sort_d(E)$ is the minimal set satisfying (without loss of generality, we assume $bn(E) \cap fn(E) = \emptyset$):

$$\begin{array}{rcl}
Sort_d(\mathbf{0}) &=& \emptyset\\
Sort_d(\alpha.E) &=& \{subj(\alpha)\}\\
Sort_d(\tau.E) &=& \emptyset\\
Sort_d(\epsilon(e).E) &=& Sort_{d\,\dot{-}\,e}(E)\\
Sort_d(E+F) &=& Sort_d(E)\cup Sort_d(F)\\
Sort_d(E|F) &=& Sort_d(E)\cup Sort_d(F)\\
Sort_d([x=x]E) &=& Sort_d(E)\\
Sort_d([x=y]E) &=& \emptyset\quad if\ x\neq y\\
Sort_d(\nu xE) &=& Sort_d(E)-\{x,\overline{x}\}\\
Sort_d(A(\tilde{v})) &=& Sort_d(E\{\tilde{v}/\tilde{x}\}),\quad A(\tilde{x})\stackrel{def}{=}E
\end{array}$$

And we restate the following result from [3]: For the relationship between name sets, we have:

**Lemma 1** *If $E\xrightarrow{\eta}E'$ then $fn(\eta)\subseteq fn(E)$ and $fn(E')\subseteq fn(E)\cup bn(\eta)$.* □

**Proof.** By transition induction on $E\xrightarrow{\eta}E'$.

1. $E \equiv \mathbf{0}$.

   There is only time delay $E\xrightarrow{\epsilon(d)}E$, the proposition is trivially true.

2. $E \equiv \epsilon(e).F$.
Then $\eta = \epsilon(d)$. When $d \leq e$, $E' = \epsilon(e-d).F$, $n(\epsilon(d)) = \emptyset$, thus $fn(\epsilon(d)) \subseteq fn(E)$ and $fn(\epsilon(e-d).F) = fn(\epsilon(e).F) \cup bn(\epsilon(d))$, as required. When $d > e$, by a shorter inference, $F \xrightarrow{\epsilon(d-e)} E'$, and $E \xrightarrow{\epsilon(e)} F$. We still have $fn(\epsilon(d)) = \emptyset \subseteq fn(E)$. By $n(\epsilon(e)) = \emptyset$, we have $fn(F) = fn(E)$. By induction, $fn(E') \subseteq fn(F) \cup bn(\epsilon(d-e))$, and by $n(\epsilon(d)) = n(\epsilon(d-e)) = \emptyset$ we have $fn(E') \subseteq fn(F) = fn(E) = fn(E) \cup bn(\epsilon(d))$.

3. $E \equiv \alpha.F$.
We need case analysis on $\alpha = \overline{x}y$, $\tau$ or $x(y)$. For $\alpha = \overline{x}y$ or $x(y)$, There exists time delay transition $E \xrightarrow{\epsilon(d)} \alpha.F$, i.e. $E' = \alpha.F$, and it is trivial. We only show the case $\alpha = x(y)$ and $\eta$ is of the form $x(z)$: By the semantics rule Pre and $\alpha-$equivalence, $x(y).F \xrightarrow{x(w)} F\{w/y\}$, where $w \notin fn(\nu y F)$. Then by $fn(x(w)) = \{x\}$ and $bn(x(w)) = \{w\}$, we have $fn(x(w)) \subseteq fn(x(y).F)$ and $fn(F\{w/y\}) \subseteq fn(x(y).F) \cup \{w\} = fn(x(y).F) \cup bn(x(w))$.

4. $E \equiv F + G$.
Without loss of generality, there must exist a short inference, $F \xrightarrow{\eta} E'$. By induction, $fn(\eta) \subseteq fn(F)$ and $fn(E') \subseteq fn(F) \cup bn(\eta)$. Then by $fn(F) \subseteq fn(F+G)$, we have $fn(\eta) \subseteq fn(F+G)$ and $fn(E') \subseteq fn(F+G) \cup bn(\eta)$.

5. $E \equiv F|G$. Similar to the third case.

6. $E \equiv [x=y]F$, $\nu x F$ or $A(\tilde{v})$. By induction. Note that when $E \equiv [x=y]F$ and $x$ is not equal to $y$, $E$ has only time delay transition $E \xrightarrow{\epsilon(d)} E$, and the result follows from $n(\epsilon(d)) = \emptyset$.

$\square$

**Definition 1** *In the following we will use the phrase:*

$$\textit{If } P \xrightarrow{\eta} P' \textit{ then equally } Q \xrightarrow{\eta'} Q'$$

*means that if $P \xrightarrow{\eta} P'$ may be inferred from the transition rules then so, by an inference of no greater depth, may be $Q \xrightarrow{\eta'} Q'$.* $\square$

**Lemma 2** *Suppose $E \xrightarrow{a(y)} E'$ where $a = x$ or $a = \overline{x}$ and that $z \notin n(E)$. Then equally for some $E'' \equiv_\alpha E'\{z/y\}$, $E \xrightarrow{a(z)} E''$.* $\square$

**Proof.** By transition induction on $E \xrightarrow{a(y)} E'$.

1. $E \equiv x(w).Q$.
   (a) $w = y$. Then $E' = Q$. By $z \notin n(E)$, $z \notin n(Q)$. We choose $E'' = Q\{z/y\}$ and then $E \xrightarrow{x(z)} E''$.
   (b) $w \neq y$. Then $E' = Q\{y/w\}$, $y \notin fn(\nu w Q)$. But by $w \neq y$, $y \notin fn(Q)$. Then $z \notin n(Q)$ follows directly from $z \notin n(E)$, and
   $$Q\{y/w\}\{z/y\} \equiv_\alpha Q\{z/w\}$$
   Take $E'' = Q\{z/w\}$, as required.

2. $E \equiv \nu w \overline{x} w.Q$ $(x \neq w)$.
   (a) $w = y$. Then $E' = Q$. By $z \notin n(E)$ we have $z \notin n(Q)$. Choose $E'' = Q\{z/y\}$, and we have $E \xrightarrow{\overline{x}(z)} E''$.
   (b) $w \neq y$. Then $E' = Q\{y/w\}$, $y \notin fn(\nu w Q)$. By $w \neq y$, $y \notin fn(Q)$. And by $z \notin n(E)$, we have $z \notin n(Q)$, therefore
   $$Q\{y/w\}\{z/y\} \equiv_\alpha Q\{z/w\}$$
   Choose $E'' = Q\{z/w\}$ as required.

3. Other cases are similar. (Note that when $E \equiv \epsilon(d).Q$, it must be $d = 0$).

$\square$

**Theorem 1 (Time Sort)** *Whenever $E \xrightarrow{\epsilon(p)} \xrightarrow{\alpha} E''$ for $p < d$, we have $subj(\alpha) \in Sort_d(E)$.*

**Proof.** It is given in the Appendix. $\square$

**Lemma 3 (Real-time Properties)**

1. **Maximal Progress**: If $P \xrightarrow{\tau} P'$ for some $P'$, then $P \xrightarrow{\epsilon(d)} P''$ for no $d > 0$ and $P''$.
2. **Time Determinacy**: If $P \xrightarrow{\epsilon(d)} P'$ and $P \xrightarrow{\epsilon(d)} P''$, then $P'$ and $P''$ are identical.
3. **Time Continuity**: For all $c$ and $d$, $P \xrightarrow{\epsilon(c+d)} P''$ **iff** $P \xrightarrow{\epsilon(c)} P'$ and $P' \xrightarrow{\epsilon(d)} P''$ for some $P'$.
4. **Persistency**: If $P \xrightarrow{\epsilon(d)} P'$ and $P \xrightarrow{\alpha} Q$ for some Q, then $P' \xrightarrow{\alpha} Q'$ for some $Q'$.

**Proof.** It is given in the Appendix. $\square$

## 2.2 Weak Bisimulation Equivalences

Strong bisimulation for the timed $\pi$ calculus and its proof system have been studied in [3], and here we only consider weak equivalences. As mentioned before, there is a distinction between *late* and *early* bisimulations [12, 13], according to when input names are instantiated. In this paper we will concentrate on *late* bisimulations, while *early* bisimulations can be handled with a minor modification.

To introduce weak bisimulations, double arrow needs to be defined:

1. $P \stackrel{\epsilon}{\Longrightarrow} P$
2. $P \xrightarrow{\alpha} P'$ implies $P \stackrel{\alpha}{\Longrightarrow} P'$
3. $P \xrightarrow{\tau}\stackrel{\alpha}{\Longrightarrow} P'$ implies $P \stackrel{\alpha}{\Longrightarrow} P'$
4. $P \stackrel{\alpha}{\Longrightarrow}\xrightarrow{\tau} P'$ implies $P \stackrel{\alpha}{\Longrightarrow} P'$, where $\alpha$ is not of the form $a(x)$ (input action)
5. $P(\xrightarrow{\tau})^* \xrightarrow{\delta_1} (\xrightarrow{\tau})^* \ldots (\xrightarrow{\tau})^* \xrightarrow{\delta_k} (\xrightarrow{\tau})^* P'$ implies $P \stackrel{\delta}{\Longrightarrow} P'$, where $\delta_i \in \mathcal{R}^{\geq 0}$ and $\Sigma_{i\leq k}\delta_i = \delta$

Let $\hat{\tau}$ be $\epsilon$, and $\hat{\alpha}$ be $\alpha$ for $\alpha \not\equiv \tau$. We will also write $\Rightarrow$ for $\stackrel{\epsilon}{\Longrightarrow}$. The *Weak* (*late* ) Ground Bisimulation can be obtained via modifying the matching step (replacing $\xrightarrow{\eta}$ by $\stackrel{\hat{\eta}}{\Longrightarrow}$) of Strong Ground Bisimulation. But due to the same reason as in standard $\pi$, late bisimulation insists that an input move from one process be matched by a *single* input move from the other, therefore in defining late weak transitions, input moves should *not* be allowed to absorb $\tau$ moves *after* them (for otherwise the resulted bisimulation relation would not be transitive [17]).

**Definition 2 (Weak (Late) Ground Bisimulation)** *A symmetric relation $S$ between processes is a weak ground bisimulation if $(P, Q) \in S$ implies*

- whenever $P \xrightarrow{x(y)} P'$ and $y \notin n(P, Q)$, then there exists some $Q''$ such that $Q \stackrel{x(y)}{\Longrightarrow} Q''$ and for any $w$ there is $Q''\{w/y\} \Rightarrow Q'$, $(P'\{w/y\}, Q') \in S$.
- whenever $P \xrightarrow{\overline{x}(y)} P'$ and $y \notin n(P, Q)$, then there exists some $Q'$ such that $Q \stackrel{\overline{x}(y)}{\Longrightarrow} Q'$ and $(P', Q') \in S$.
- for any other actions $\eta \in \mathcal{L}$, whenever $P \xrightarrow{\eta} P'$, then there exists some $Q'$ such that $Q \stackrel{\hat{\eta}}{\Longrightarrow} Q'$ and $(P', Q') \in S$.

$\square$

If there exists a weak ground bisimulation relation $S$ such that $(P, Q) \in S$, we say $P$ and $Q$ are weakly ground-bisimilar, denoted by $P \dot{\approx} Q$.

$\dot{\approx}$ is the largest weak ground bisimulation relation, i.e., $\dot{\approx} = \bigcup\{S | S$ is a weak ground bisimulation$\}$. Using the same notation as in [3], we write $\sim$, $\dot{\sim}$ for Strong Bisimulation and Strong Ground Bisimulation respectively.

Now we show some important properties of $\dot{\approx}$ which will be used to construct a congruence relation.

**Lemma 4** *$P \dot{\approx} Q$ implies that if $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$, then $Sort_d(P) = Sort_d(Q)$.* $\square$

**Proof.** It is given in the Appendix.

Note that $P \dot{\approx} Q$ doesn't mean $Sort_d(P) = Sort_d(Q)$. e.g., $\tau.E \dot{\approx} E$, but there can be $E \xrightarrow{\epsilon(d)} E'$ for some $d > 0$, while $\tau.E$ absolutely cannot have a $\epsilon(d)$ transition for any $d > 0$.

The relation $\dot{\approx}$ is not a congruence, in that it is first not preserved by summation (e.g., $b.\mathbf{0} \dot{\approx} \tau.b.\mathbf{0}$ but $a.\mathbf{0} + b.\mathbf{0} \not\dot{\approx} a.\mathbf{0} + \tau.b.\mathbf{0}$, as pointed out in CCS [11]). And like Strong Ground Bisimulation [3, 12], it is also not preserved by substitution (which in turn leads to the result that $\dot{\approx}$ is not preserved by input prefix). These two defects can be remedied either separately or together, and here we derive equivalences as follows, where '$\approx^c$' is expected to be a congruence which is preserved by all the operators:

**Definition 3** $P \dot{\approx}^c Q$ **if** *for all process* $R$, $P + R \dot{\approx} Q + R$.
$P \approx^c Q$ **if** *for all substitution* $\sigma \in Subst$, $P\sigma \dot{\approx}^c Q\sigma$. □

Clearly, $\approx^c$ is included in $\dot{\approx}$.

**Theorem 2** $P \approx^c Q$ *implies* $P \dot{\approx}^c Q$, *and* $P \dot{\approx} Q$.

**Proof.** By taking identity substitution ($P\sigma = P$ and $Q\sigma = Q$), $P \approx^c Q$ implies $P \dot{\approx}^c Q$. Furthermore by $P + \mathbf{0} \dot{\approx} Q + \mathbf{0}$, we have $P \dot{\approx} Q$. □

And, it is larger than all the congruences included in $\dot{\approx}$.

**Theorem 3** *If* $\approx'$ *is a congruence, then* $\approx' \subseteq \dot{\approx}$ *implies* $\approx' \subseteq \approx^c$.

**Proof.** Assume $P \approx' Q$. For any process $R$, $P + R \approx' Q + R$ implies $P + R \dot{\approx} Q + R$ (by $\approx' \subseteq \dot{\approx}$). Therefore $P \dot{\approx}^c Q$. Given $P \approx' Q$, to imply $x(y).P \approx' x(y).Q$, we must show for any $v$, $P\{v/y\} \approx' Q\{v/y\}$. Since $y$ and $v$ are arbitrary, we have for any $\sigma \in Subst$, $P \approx' Q$ implies $P\sigma \approx' Q\sigma$, i.e. $P\sigma \dot{\approx}^c Q\sigma$, and so $P \approx^c Q$. Therefore $\approx' \subseteq \approx^c$. □

With this result, if we can show that $\approx^c$ is a congruence, then it is the largest congruence that we are looking for. Due to the technical difficulties, we have to show this via an alternative characterization of $\approx^c$.

First we need to define $\xrightarrow{\eta}_p$ ($\eta \in \Lambda \cup \{\epsilon\}$) as follows.

**Definition 4**

1. $P \xrightarrow{\epsilon(p)} P'$ implies $P \xrightarrow{\epsilon}_p P'$
2. $P \xrightarrow{\epsilon(p)}\xrightarrow{\alpha} P'$ implies $P \xrightarrow{\alpha}_p P'$

□

And its weak version is defined as follows.

**Definition 5**

1. $P \xrightarrow{\epsilon}_p P'$ implies $P \stackrel{\epsilon}{\Longrightarrow}_p P'$
2. $P \xrightarrow{\epsilon(p)}\stackrel{\alpha}{\Longrightarrow} P'$ implies $P \stackrel{\alpha}{\Longrightarrow}_p P'$

□

Note that $\stackrel{\tau}{\Longrightarrow}_p$ should contain at least one $\tau$ transition, therefore $\epsilon(p).\mathbf{0}$ cannot have $\stackrel{\tau}{\Longrightarrow}_p$ while $\epsilon(p).\tau.\mathbf{0}$ can.

**Definition 6** $\dot{\simeq}$ *is a symmetric relation between processes such that, if* $P \dot{\simeq} Q$ *implies*

- whenever $P \xrightarrow{x(y)}_p P'$ and $y \notin n(P,Q)$, then there exists some $Q''$ such that $Q \stackrel{x(y)}{\Longrightarrow}_p Q''$ and for any $w$ there is $Q''\{w/y\} \Rightarrow Q'$, $P'\{w/y\} \dot{\approx} Q'$.
- whenever $P \xrightarrow{\overline{x}(y)}_p P'$ and $y \notin n(P,Q)$, then there exists some $Q'$ such that $Q \stackrel{\overline{x}(y)}{\Longrightarrow}_p Q'$ and $P' \dot{\approx} Q'$.
- for any other actions $\eta \in \Lambda \cup \{\epsilon\}$, whenever $P \xrightarrow{\eta}_p P'$, then there exists some $Q'$ such that $Q \stackrel{\eta}{\Longrightarrow}_p Q'$ and $P' \dot{\approx} Q'$.

□

**Definition 7** $P \simeq Q$ *if for all* $\sigma \in Subst$, $P\sigma \dot{\simeq} Q\sigma$.

For example, we have $\tau.\alpha + \alpha \simeq \tau.\alpha$ but $\tau.\alpha \not\simeq \alpha$.

**Theorem 4**

1. $\simeq$ is an equivalence relation.
2. $\simeq \subset \dot{\approx}$.

$\square$

**Proof.** Easy to see that $\mathbf{Id} = \{(P,P)\} \subseteq \approx^c$, $\simeq \circ \simeq = \simeq$ and $(\simeq)^{-1} = \simeq$, and $\simeq \cup \dot{\approx}$ is a weak ground bisimulation. $\square$

We expect that $\simeq$ coincides with $\approx^c$ when a similar hypothesis is imposed on processes as suggested by Milner [11]: the action space $\Lambda$ is arbitrarily large so that no process can exhaust all the actions in $\Lambda$. This assumption is naturally and always used in the $\pi$ calculus, for example, when defining bisimulations, a fresh name should always be available for matching input or bound output actions.

Before going further, we establish a lemma first.

**Lemma 5**

1. $P \dot{\approx} Q$ implies that if $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$, then $P' \dot{\approx} Q'$.
2. $P \dot{\simeq} Q$ implies that if $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$, then $P' \dot{\simeq} Q'$.
3. $P \dot{\approx}^c Q$ implies that if $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$, then $P' \dot{\approx}^c Q'$.

**Proof.** It is given in the Appendix. $\square$

The above lemma actually states *Time Determinacy* in a general way. Recall the *Time Determinacy* lemma, i.e., if $P \xrightarrow{\epsilon(d)} P'$ and $P \xrightarrow{\epsilon(d)} P''$, Then $P' = P''$. In other words, $P = Q$ implies that if $P \xrightarrow{\epsilon(d)} P'$, $Q \xrightarrow{\epsilon(d)} Q'$, then $P' = Q'$. Clearly, the result also holds for strong equivalence. Then the *Time Determinacy* lemma can be generalized as follows.

**Lemma 6** *Let* $\bowtie \in \{=, \dot{\sim}, \dot{\approx}, \dot{\simeq}, \dot{\approx}^c\}$*, Then* $P \bowtie Q$ *implies that if* $P \xrightarrow{\epsilon(d)} P'$ *and* $Q \xrightarrow{\epsilon(d)} Q'$*, then* $P' \bowtie Q'$*.*

Now we show the main theorem concerning $\simeq$ and $\approx^c$.

**Theorem 5** *For any* $P$*, if there exists* $\alpha \in \Lambda$ *such that* $P \xrightarrow{\alpha} Q$ *for no* $Q$*, then* $\approx^c = \simeq$*.* $\square$

**Proof.** By definition, Only need to show $\dot{\approx}^c$ coincides with $\dot{\simeq}$.

We first show $P \dot{\simeq} Q$ implies $P \dot{\approx}^c Q$. Then we only need to show $R$ is a weak ground bisimulation:

$$R \;=\; \{(P+G, Q+G) \mid P \dot{\simeq} Q\} \cup \dot{\approx}$$

1. Assume $P + G \xrightarrow{\epsilon(d)} P' + G'$. We must have $P \xrightarrow{\epsilon(d)} P'$ and $G \xrightarrow{\epsilon(d)} G'$. By definition, $P \xrightarrow{\epsilon}_d P'$ and $G \xrightarrow{\epsilon}_d G'$. Because $P \dot{\simeq} Q$, $Q \overset{\epsilon}{\Longrightarrow}_d Q'$, i.e. $Q \xrightarrow{\epsilon(d)} Q'$. Hence, $Q + G \xrightarrow{\epsilon(d)} Q' + G'$. Then $Q + G \overset{\epsilon(d)}{\Longrightarrow} Q' + G'$. By lemma 6 we have $P' \dot{\simeq} Q'$, Then $(P' + G', Q' + G') \in R$, as required.
2. Assume $P + G \xrightarrow{\alpha} P'$. Here we only consider $bn(\alpha) = \emptyset$, other cases are similar. Assume $P + G \xrightarrow{\alpha} P'$ is inferred by $P \xrightarrow{\alpha} P'$. By definition, $P \xrightarrow{\alpha}_0 P'$. By $P \dot{\simeq} Q$, $Q \overset{\alpha}{\Longrightarrow}_0 Q'$. Then $Q \overset{\alpha}{\Longrightarrow} Q'$, i.e. $Q + G \overset{\alpha}{\Longrightarrow} Q'$ and $P' \dot{\approx} Q'$, $(P', Q') \in R$. The case when $P + G \xrightarrow{\alpha} G'$ is inferred by $G \xrightarrow{\alpha} G'$ is similar.

Now we show $P \dot{\approx}^c Q$ implies $P \dot{\simeq} Q$. Only need to show: if $P \xrightarrow{\phi}_p P'$, then there exists $Q'$, $Q \overset{\phi}{\Longrightarrow}_p Q'$ and when $\phi$ is not an input action, $P' \dot{\approx} Q'$; otherwise if $\phi = a(z)$ $(z \notin n(P,Q))$, for any $y$ there is a $Q'\{y/z\} \Rightarrow Q''$ such that $P'\{y/z\} \dot{\approx} Q''$.

Assume $P \xrightarrow{\phi}_p P'$, There are two possibilities:

1. $\phi = \epsilon$ and $p > 0$ (the case $p = 0$ is trivial, $P = P'$, $Q = Q'$). We must show that $Q \overset{\epsilon}{\Longrightarrow}_p Q'$, i.e. $Q \xrightarrow{\epsilon(p)} Q'$ and $P' \dot{\approx} Q'$. First we show $Q \xrightarrow{\epsilon(p)}$. By $\dot{\approx}^c \subset \dot{\approx}$, $Q \overset{\epsilon(p)}{\Longrightarrow}$. Now we show there exists no $d < p$ such that $Q \xrightarrow{\epsilon(d)}\xrightarrow{\tau}$. Assume that $Q \xrightarrow{\epsilon(d)} Q' \xrightarrow{\tau} Q''$, then choose $\beta \in \Lambda$ such that $Q'' \overset{\beta}{\not\Longrightarrow}$. Then we have $Q + \beta \xrightarrow{\epsilon(d)}\xrightarrow{\tau} Q''$. By $d < p$ and $P \xrightarrow{\epsilon(p)} P'$, By *Maximal Progress* lemma, $P + \beta \overset{\epsilon(d)}{\Longrightarrow} P'' + \beta$ is the only $\overset{\epsilon(d)}{\Longrightarrow}\overset{\hat{\tau}}{\Longrightarrow}$ transition

for $P+\beta$, where $P \stackrel{\epsilon(d)}{\longrightarrow} P''$. Clearly $Q'' \not\approx P''+\beta$, because $Q''$ refuses to do $\beta$. Hence $P+\beta \not\approx Q+\beta$, which contradicts $P \approx^c Q$.

Thus there exists $Q'$, $Q \stackrel{\epsilon(p)}{\longrightarrow} Q'$. By lemma 6, $P' \dot{\approx}^c Q'$, therefore $P' \dot{\approx} Q'$.

2. $\phi = \alpha$. Here we only consider $bn(\alpha) = \emptyset$, other cases are similar. the case $p = 0$ is trivial, consider $p > 0$. Then we need to show $Q \stackrel{\alpha}{\Longrightarrow}_p Q'$, i.e. there exist $Q'$ and $Q''$, $Q \stackrel{\epsilon(p)}{\longrightarrow} Q'' \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \dot{\approx} Q'$. By $P \stackrel{\alpha}{\longrightarrow}_p P'$, there exists $P''$, $P \stackrel{\epsilon(p)}{\longrightarrow} P'' \stackrel{\alpha}{\longrightarrow} P'$. From the case $\phi = \epsilon$, there exists $Q''$, $Q \stackrel{\epsilon(p)}{\longrightarrow} Q''$. By lemma 6, $P'' \dot{\approx}^c Q''$. Thus it is enough to show: if $P'' \stackrel{\alpha}{\longrightarrow} P'$, then there exists $Q'$, $Q'' \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \dot{\approx} Q'$.
If $\alpha \neq \tau$, then the result follows directly from the fact that $P'' \dot{\approx}^c Q''$ implies $P'' \dot{\approx} Q''$. Now consider $\alpha = \tau$:
Assume that $P'' \stackrel{\tau}{\longrightarrow} P'$, but $Q'' \stackrel{\tau}{\not\longrightarrow}$. Then choose $\beta \in \Lambda$ such that $P' \stackrel{\beta}{\not\Longrightarrow}$. $P''+\beta \stackrel{\tau}{\longrightarrow} P'$; but $Q''+\beta \Rightarrow Q''+\beta$ is the only $\stackrel{\hat{\tau}}{\Longrightarrow}$ transition of $Q''+\beta$ because $Q'' \stackrel{\tau}{\not\longrightarrow}$. Since $P'$ refuses to do $\beta$, $P' \not\approx Q''+\beta$, hence $P''+\beta \not\approx Q''+\beta$, which contradicts $P'' \dot{\approx}^c Q''$. Therefore there must be $Q'' \stackrel{\tau}{\longrightarrow}$. Furthermore by $P'' \dot{\approx}^c Q''$, $P''+\beta \dot{\approx} Q''+\beta$. Then by $P''+\beta \stackrel{\tau}{\longrightarrow} P'$, there exists $R$, $Q''+\beta \stackrel{\tau}{\Longrightarrow} R$, $P' \dot{\approx} R$. Clearly $R \neq Q''+\beta$, because $P'$ refuses $\beta$, then $R$ must be some $Q'$, $Q'' \stackrel{\tau}{\longrightarrow} (\stackrel{\tau}{\longrightarrow})^* Q'$, i.e. $Q'' \stackrel{\tau}{\Longrightarrow} Q'$ and $P' \dot{\approx} Q'$.

□

We can show $\simeq$ is preserved by all the operators (therefore it is a congruence).

**Theorem 6** *$P \simeq Q$ implies*

$$\begin{array}{rcl}
\alpha.P & \simeq & \alpha.Q \\
\epsilon(d).P & \simeq & \epsilon(d).Q \\
[x=y]P & \simeq & [x=y]Q \\
\nu x P & \simeq & \nu x Q \\
P \mid R & \simeq & Q \mid R \\
P+R & \simeq & Q+R
\end{array}$$

□

**Proof.** We prove the fifth. The others are straightforward. Assume $P \simeq Q$, we need to show $P \mid R \simeq Q \mid R$ for any arbitrary $R$. Given $\sigma \in Subst$, $P_\sigma = P\sigma$, $Q_\sigma = Q\sigma$ and $R_\sigma = R\sigma$. Then $P_\sigma \dot{\simeq} Q_\sigma$.

1. Assume $P_\sigma \mid R_\sigma \stackrel{\eta}{\longrightarrow}_p P' \mid R'$ ($\eta \in \Lambda$). There are three possibilities.

   (a) $P_\sigma \stackrel{\eta}{\longrightarrow}_p P'$, $R_\sigma \stackrel{\epsilon(p)}{\longrightarrow} R'$ and $Sort_p(P_\sigma) \cap \overline{Sort_p(R_\sigma)} = \emptyset$. We only consider the case $\eta = a(z)$ ($z \notin n(P_\sigma, Q_\sigma, R_\sigma)$), other cases are its simplified versions. Then there exists $Q''$ such that $Q_\sigma \stackrel{a(z)}{\Longrightarrow}_p Q''$ and for any $y$ there is a $Q''\{y/z\} \Rightarrow Q'$ such that $P'\{y/z\} \dot{\approx} Q'$. Obviously we have $Q_\sigma \mid R_\sigma \stackrel{a(z)}{\Longrightarrow}_p Q'' \mid R'$, and $(Q'' \mid R')\{y/z\} = Q''\{y/z\} \mid R' \Rightarrow Q' \mid R'$. Since $\dot{\approx}$ is preserved by parallel composition '|' (see [3]) and $P'\{y/z\} \dot{\approx} Q'$, we have $(P' \mid R')\{y/z\} = P'\{y/z\} \mid R' \dot{\approx} Q' \mid R'$ as required.

   (b) $R_\sigma \stackrel{\eta}{\longrightarrow}_p R'$, $P_\sigma \stackrel{\epsilon(p)}{\longrightarrow} P'$ and $Sort_p(P_\sigma) \cap \overline{Sort_p(R_\sigma)} = \emptyset$. Similar to the case above.

   (c) $P_\sigma \stackrel{\alpha}{\longrightarrow}_p P'$, $R_\sigma \stackrel{\beta}{\longrightarrow}_p R'$ where $subj(\alpha) = \overline{subj(\beta)}$ and $Sort_p(P_\sigma) \cap \overline{Sort_p(R_\sigma)} = \emptyset$. Similar to the first case. (When either $\alpha$ or $\beta$ is a bound output, a version of "*Up to*" technique has to be used similarly as in [3] and [12].)

2. Assume $P_\sigma \mid R_\sigma \stackrel{\epsilon}{\longrightarrow}_p P' \mid R'$. Then we must have $P_\sigma \stackrel{\epsilon(p)}{\longrightarrow} P'$, $R_\sigma \stackrel{\epsilon(p)}{\longrightarrow} R'$ and $Sort_p(P_\sigma) \cap \overline{Sort_p(R_\sigma)} = \emptyset$. Since $P_\sigma \dot{\simeq} Q_\sigma$, there exists $Q'$ such that $Q \stackrel{\epsilon}{\Longrightarrow}_p Q'$, i.e. $Q \stackrel{\epsilon(p)}{\longrightarrow} Q'$ and $P' \dot{\approx} Q'$. Further, from $P_\sigma \dot{\simeq} Q_\sigma$, we have $P_\sigma \dot{\approx} Q_\sigma$. By Lemma 4, $Sort_p(P_\sigma) = Sort_p(Q_\sigma)$ and then $Sort_p(Q_\sigma) \cap \overline{Sort_p(R_\sigma)} = \emptyset$. Thus $Q_\sigma \mid R_\sigma \stackrel{\epsilon(p)}{\longrightarrow} Q' \mid R'$, i.e. $Q_\sigma \mid R_\sigma \stackrel{\epsilon}{\Longrightarrow}_p Q' \mid R'$. And, $P' \mid R' \dot{\approx} Q' \mid R'$ because $P' \dot{\approx} Q'$.

□

Easy to see, $\dot{\simeq}$ is preserved by all the operators except *input* prefix, due to the fact that, similar to $\dot{\approx}$, $\dot{\simeq}$ is not preserved by name substitutions.

# 3 Axiomatization

## 3.1 Notations for Symbolic Technique

In this section we'll consider complete axiomatization of weak congruence between processes (recursion-free) defined in the last section. First we restate the *Expansion Law* (presented in [3]) for Strong Bisimulation:

**Theorem 7 (Expansion Law)** *Suppose* $P \equiv \sum_{i\in I} M_i\epsilon(e_i).\alpha_i.P_i$ *and* $Q \equiv \sum_{j\in J} N_j\epsilon(f_j).\beta_j.Q_j$*, where* $M_i$ *and* $N_j$ *are conjunctions of equality tests of the form* $[x = y]$*.* $\forall i \in I.bn(\alpha_i) \cap fn(Q) = \emptyset$ *and* $\forall j \in J.bn(\beta_j) \cap fn(P) = \emptyset$*, then*

$$P|Q \;=\; \sum_{i\in I} M_i\epsilon(e_i).\alpha_i.(P_i|Q') + \sum_{j\in J} N_j\epsilon(f_j).\beta_j.(P'|Q_j) + \sum_{\alpha_i\ opp\ \beta_j} M_iN_j[x_i = y_j]\epsilon(max(e_i, f_j)).\tau.R_{ij}$$

*where*

- $Q' \equiv \sum_{j\in J} N_j\epsilon(f_j \dot{-} e_i).\beta_j.Q_j$
- $P' \equiv \sum_{i\in I} M_i\epsilon(e_i \dot{-} f_j).\alpha_i.P_i$

and $\alpha_i$ *opp* $\beta_j$ is defined by:

1. $\alpha_i = \bar{x_i}u$ and $\beta_j = y_j(v)$; then $R_{ij} = P_i|Q_j\{u/v\}$.
2. $\alpha_i = \bar{x_i}(u)$ and $\beta_j = y_j(v)$; then $R_{ij} = \nu w(P_i\{w/u\}|Q_j\{w/v\})$, where $w$ is not free in $\nu uP_i$ *or* $\nu vQ_j$.
3. $\alpha_i = x_i(v)$ and $\beta_j = \bar{y_j}u$; then $R_{ij} = P_i\{u/v\}|Q_j$.
4. $\alpha_i = x_i(v)$ and $\beta_j = \bar{y_j}(u)$; then $R_{ij} = \nu w(P_i\{w/v\}|Q_j\{w/u\})$, where $w$ is not free in $\nu vP_i$ *or* $\nu uQ_j$.

$\square$

Since strong bisimulation '$\sim$' is a congruence contained in $\dot{\approx}$, so $\sim\subseteq\simeq$, hence, by this equation, any process expression containing '|' can be reduced to a *equivalent* (up to $\simeq$) process containing no '|'. Therefore in the following we will focus on laws and inference rules concerning no '|'.

We would not use any axioms about Match, e.g. like $[x = x]E = E$ in [12], and would appeal to **Symbolic Bisimulation** technique to obtain the complete axiomatization.

Following the notations in [8] and [10], let's first introduce **Conditions**, ranged over by $C, D$, which are finite sets of equality or inequality tests on names. We will write $n(C)$ for the set of names appearing in $C$. We say $C$ is a condition on a name set $V$ if $n(C) \subseteq V$. The empty condition will sometimes be denoted by $true$.

A substitution $\sigma$ satisfies $C$, written $\sigma \models C$, if $x\sigma = y\sigma$ for any $x = y \in C$, and $x\sigma \neq y\sigma$ for any $x \neq y \in C$. We write $C \Rightarrow D$ to mean that $\sigma \models C$ implies $\sigma \models D$ for any substitution $\sigma$.

Two substitutions $\sigma, \sigma'$ are equal on $V$, written $\sigma =^V \sigma'$, if $x\sigma = x\sigma'$ for all $x \in V$. $\sigma, \sigma'$ are *elementarily equivalent* on $V$, if for any $x, y \in V$, $\sigma \models x = y$ if and only if $\sigma' \models x = y$.

**Lemma 7** *Suppose* $\sigma, \sigma'$ *are substitutions on* $V$*. If they are elementarily equivalent on* $V$*, then there exists an injective substitution* $\varsigma$ *such that* $\sigma =^V \sigma'\varsigma$*.*

**Proof.** This lemma is the lemma 2.5 in [8]. Let $\varsigma$ be the substitution such that $\varsigma(\sigma'(x)) = \sigma(x)$ and $\varsigma(\sigma(x)) = \sigma'(x)$ for all $x \in V$, and is identity otherwise. $\varsigma$ is well defined because $\sigma, \sigma'$ are elementarily equivalent on $V$. Furthermore $\varsigma$ is injective and satisfies $\sigma =^V \sigma'\varsigma$. $\square$

A condition $C$ is *consistent* if there are no $x, y \in \mathcal{N}$ such that $C \Rightarrow x = y$ and $C \Rightarrow x \neq y$. $C$ is *maximally consistent* on $V \subset \mathcal{N}$ if for any $x, y \in V$, either $C \Rightarrow x = y$ or $C \Rightarrow x \neq y$. $C'$ is a *maximally consistent extension* of $C$ on $V$, written $C' \in MCE_V(C)$, if $C \subseteq C'$ and $C'$ is *maximally consistent* on $V$. The set of maximally consistent extensions of a given condition on a finite set of names $V$ is finite. We will abbreviate $MCE_V(true)$ as $MC_V$.

Given a condition $C$, define $E_V(C) = \{x = y | x, y \in V, C \not\Rightarrow x = y \text{ and } C \not\Rightarrow x \neq y\}$, and for any $I \subseteq E_V(C)$ define $I^* = \{x = y | x, y \in V, I \Rightarrow x = y\}$ and $\overline{I} = \{x \neq y | x = y \in E_V(C) - I^*\}$. Clearly $C$ is maximally consistent on $V$ if and only if $E_V(C) = \emptyset$.

**Lemma 8** $D \in MCE_V(C)$ *if and only if* $D = C \cup I \cup \overline{I}$ *for some* $I \subseteq E_V(C)$*.*

**Proof.** This lemma is the lemma 2.6 in [8]. For any $I \subseteq E_V(C)$, by construction $C \cup I \cup \overline{I}$ is a maximally consistent extension of $C$ on $V$. On the other hand, for any $D \in MCE_V(C)$, let $I = \{x = y | D \Rightarrow x = y \text{ and } C \not\Rightarrow x = y, C \not\Rightarrow x \neq y\}$, then clearly $I \subseteq E_V(C)$. Since $D$ is maximally consistent, then $\overline{I} = \{x \neq y | D \Rightarrow x \neq y \text{ and } C \not\Rightarrow x = y, C \not\Rightarrow x \neq y\}$, and clearly $D = C \cup I \cup \overline{I}$. □

**Lemma 9** $\bigvee MCE_V(C) = C$

**Proof.** This lemma is the Corollary 2.7 in [8].

$$\begin{array}{rcl}\bigvee MCE_V(C) & = & \bigvee\{C \cup I \cup \overline{I} | I \subseteq E_V(C)\} \\ & = & C \wedge \bigvee\{I \cup \overline{I} | I \subseteq E_V(C)\} \\ & = & C \wedge true \\ & = & C\end{array}$$

□

Lemma 9 shows that the set of all consistent extensions of a condition (on a given name set) constitutes a particular partition, or decomposition of the condition.

**Theorem 8** *$C$ is maximally consistent on $V$ if and only if all substitutions satisfying $C$ are elementarily equivalent on $V$.*

**Proof.** This lemma is the Corollary 2.8 in [8]. The "only if" part is trivial. For the "if" part, suppose $C$ is not maximally consistent on $V$. Then there exist $x, y \in V$ such that neither $C \Rightarrow x = y$ nor $C \Rightarrow x \neq y$. So there exists a substitution $\sigma$ such that $\sigma \models C$ and $\sigma \models x \neq y$. Let $\sigma'$ be the substitution such that

$$v\sigma' = \begin{cases} y\sigma, & v = x \\ v\sigma, & \texttt{otherwise} \end{cases}$$

Then we still have $\sigma' \models C$. But $\sigma'$ is not elementarily equivalent to $\sigma$ on $V$ because $\sigma' \models x = y$. A contradiction. □

**Lemma 10** *Suppose $C$ is maximally consistent on $V$. If $\sigma$ and $\sigma'$ both satisfy $C$, then $\sigma =^V \sigma' \varsigma$ for some injective substitution $\varsigma$.*

**Proof.** Immediately from Lemma 7 and Theorem 8. □

**Lemma 11** *Suppose $D$ is maximally consistent on $V$, $C$ is a condition with $n(C) \subseteq V$. If $\sigma \models D$ and $\sigma \models C$ for some $\sigma$ (in other words, $D \cup C$ is consistent), then $D \Rightarrow C$.*

**Proof.** By induction on the size of $C$. If $C$ is empty then the result holds trivially. Otherwise let $C = C' \cup \{e\}$ with $e \notin C'$. By induction $D \Rightarrow C'$. Since $\sigma \models D$ and $\sigma \models e$, $D \cup \{e\}$ is consistent. Since $D$ is maximally consistent on $V$ and $n(e) \subseteq V$, $D \Rightarrow e$. Therefore $D \Rightarrow C$. □

Since terms (processes) which are $\alpha$-equivalent have the same behavior, we will not distinguish these terms. Let $M, N$ be conditions on names, we will abbreviate the union of $M$ and $N$ as $MN$. And we will not distinguish $\{x = y\}, [x = y]$ and $x = y$ when the context is understood.

The transitions defined previously are *concrete* in the sense that they will always be fired regardless the context in which terms are placed. To use symbolic technique, we turn to a more abstract form of transitions - *symbolic transition*, which is defined in Fig. 2.

$$PRE \; \frac{}{\alpha.E \xrightarrow{true,\alpha} E} \qquad IDLE \; \frac{}{\alpha.E \xrightarrow{true,\epsilon(d)} \alpha.E} \; \alpha \neq \tau$$

$$IDLE0 \; \frac{}{\mathbf{0} \xrightarrow{true,\epsilon(d)} \mathbf{0}} \qquad DELAY_0 \; \frac{}{\epsilon(e+d).E \xrightarrow{true,\epsilon(d)} \epsilon(e).E}$$

$$DELAY_1 \; \frac{E \xrightarrow{M,\epsilon(d)} E'}{\epsilon(e).E \xrightarrow{M,\epsilon(e+d)} E'} \qquad DELAY_2 \; \frac{E \xrightarrow{M,\eta} E'}{\epsilon(0).E \xrightarrow{M,\eta} E'}$$

$$MATCH0 \; \frac{}{[x=y]E \xrightarrow{x\neq y,\epsilon(d)} [x=y]E} \qquad MATCH \; \frac{E \xrightarrow{M,\eta} E'}{[x=y]E \xrightarrow{ML,\eta} E'} \quad L = \begin{cases} x=y & if\ x\neq y \\ \emptyset & \texttt{otherwise} \end{cases}$$

$$SUM \; \frac{E \xrightarrow{M,\alpha} E'}{E+F \xrightarrow{M,\alpha} E'} \qquad SUM\text{-}DELAY \; \frac{E \xrightarrow{M,\epsilon(d)} E' \quad F \xrightarrow{N,\epsilon(d)} F'}{E+F \xrightarrow{MN,\epsilon(d)} E'+F'}$$

$$RES \; \frac{E \xrightarrow{M,\alpha} E'}{\nu x E \xrightarrow{M,\alpha} \nu x E'} \; x \notin n(M,\alpha) \qquad RES0 \; \frac{E \xrightarrow{M,\epsilon(d)} E'}{\nu x E \xrightarrow{M',\epsilon(d)} \nu x E'} \; \begin{array}{l} \nexists\ x=y \in M, \\ M' = M \backslash \{x\neq y | x\neq y \in M\} \end{array}$$

$$OPEN \; \frac{E \xrightarrow{M,\overline{a}x} E'}{\nu x E \xrightarrow{M,\overline{a}(x)} E'} \; x \notin n(M,a) \qquad ID \; \frac{E\{\tilde{v}/\tilde{x}\} \xrightarrow{M,\eta} P}{A(\tilde{v}) \xrightarrow{M,\eta} P} \; A(\tilde{x}) \stackrel{def}{=} E$$

Fig. 2: Symbolic Transition Semantics of Timed $\pi$

**Lemma 12**

1. If $E \xrightarrow{M,\eta} E'$, then $n(M) \cup fn(\eta) \subseteq fn(E)$.
2. If $E \xrightarrow{M,c(x)} E'$ where $c$ is $a$ or $\bar{a}$, then for any $z \notin fn(E)$, $E \xrightarrow{M,c(z)} E''$ with $E'' \equiv E'\{z/x\}$.

**Proof.** By transition induction. $\square$

Next we will show the relations between *Symbolic Transitions* and *Concrete Transitions*:

**Lemma 13** *If $E\sigma \xrightarrow{\eta} P$, $bn(\eta) \cap (fn(E) \cup n(\sigma)) = \emptyset$, then there exist $M, \eta', E'$ such that $\sigma \models M$, $\eta = \eta'\sigma$, $P \equiv E'\sigma$ and $E \xrightarrow{M,\eta'} E'$.*

**Proof.** By transition induction on $E\sigma \xrightarrow{\eta} P$:

1. $E \equiv \mathbf{0}$. It can only have $\mathbf{0} \equiv \mathbf{0}\sigma \equiv E\sigma \xrightarrow{\epsilon(d)} E\sigma$ and $E \xrightarrow{true,\epsilon(d)} E$, trivial.
2. $E \equiv \alpha.F$.
   (a) $\eta = \alpha\sigma$. Let $\eta' = \alpha$, then $E \xrightarrow{true,\alpha} F = E'$.
   (b) $\eta = \epsilon(d)$. Then $\eta' = \eta$ and $E \xrightarrow{true,\epsilon(d)} E = E'$.
3. $E \equiv [x = y]F$. Then $E\sigma \xrightarrow{\eta} P$ can only be inferred finally by either $\text{Cond}_1$ or $\text{Cond}_2$:
   (a) When $\text{Cond}_1$ is finally applied. There must be $x\sigma = y\sigma$, i.e., $\sigma \models x = y$, and $F\sigma \xrightarrow{\eta} P$. Then by induction assumption, there must exist $F \xrightarrow{M',\eta''} F'$, $P \equiv F'\sigma$, $\eta = \eta''\sigma$, $\sigma \models M'$. Let $M = M' \cup \{x = y\}$, then $\sigma \models M$, and just choose $\eta' = \eta''$, $E' = F'$, as required.
   (b) When $\text{Cond}_2$ is finally applied. There must be $\eta = \epsilon(d)$, and $x\sigma \neq y\sigma$, i.e., $\sigma \models x \neq y$. Then by MATCH0, $E \xrightarrow{x\neq y,\epsilon(d)} E = E'$, as required.
4. $E \equiv \nu xF$. When $\eta$ is a non-delay action, it follows from induction assumption. We only need to prove the case $\eta = \epsilon(d)$:
   Without loss of generality, we only need to consider the case $x \notin n(\sigma)$. Then $E\sigma \xrightarrow{\eta} P$ can only be inferred finally by *Res*, therefore there must be $F\sigma \xrightarrow{\epsilon(d)} P'$, $P = \nu xP'$. By induction assumption, there exist $F \xrightarrow{M',\epsilon(d)} F'$, $P' \equiv F'\sigma$, $\sigma \models M'$.
   When $\nexists\ x \neq y \in M'$, then take $M = M'$, $E' = \nu xF'$.
   When $\exists\ x \neq y \in M'$, then take $M = M' \setminus \{x \neq y | x \neq y \in M'\}$. Then $\sigma \models M$, $E \xrightarrow{M,\epsilon(d)} E' = \nu xF'$, thus $P = \nu xP' \equiv \nu x(F'\sigma) \equiv (\nu xF')\sigma = E'\sigma$ as required.
5. Other cases of $E$ and transitions can be proved similarly.

$\square$

**Lemma 14** *If $E \xrightarrow{M,\eta} E'$, then for any $\sigma \models M$ and $bn(\eta) \cap (fn(E) \cup n(\sigma)) = \emptyset$, $E\sigma \xrightarrow{\eta\sigma} P \equiv E'\sigma$.*

**Proof.** By transition induction on $E \xrightarrow{M,\eta} E'$:

1. $E \equiv \mathbf{0}$. It only has $E \xrightarrow{true,\epsilon(d)} E$ and $E \xrightarrow{\epsilon(d)} E$, trivial.
2. $E \equiv \alpha.F$.
   (a) $\eta = \alpha$. Then $E' = F$, and we have $E\sigma \xrightarrow{\alpha\sigma} F\sigma$ as required.
   (b) $\eta = \epsilon(d)$. Then $\eta' = \eta$, $E' = E$, and we have $E\sigma \xrightarrow{\epsilon(d)} E\sigma$ as required.
3. $E \equiv [x = y]F$. Then $E \xrightarrow{M,\eta} E'$ can finally be inferred by either MATCH0 or MATCH.
   (a) $E \xrightarrow{M,\eta} E'$ is finally inferred by MATCH0: then $\eta = \epsilon(d)$, $E' = E$ and M must be $x \neq y$, thus for any $\sigma \models x \neq y$, we have $x\sigma \neq y\sigma$. Therefore $E\sigma \equiv [x\sigma = y\sigma]F\sigma$, and $E\sigma \xrightarrow{\epsilon(d)} E\sigma$ by Cond$_2$, as required.
   (b) $E \xrightarrow{M,\eta} E'$ is inferred finally by MATCH: When $x = y$, it follows from induction assumption. We only need to consider the case $x \neq y$: Then there must be $F \xrightarrow{M',\eta} E'$, where $M = M' \cup \{x = y\}$. By induction assumption, for any $\sigma \models M'$ and $bn(\eta) \cap (fn(F) \cup n(\sigma)) = \emptyset$, we have $F\sigma \xrightarrow{\eta\sigma} E'\sigma$. Clearly, $\sigma \models M$ implies $\sigma \models M'$, and $bn(\eta) \cap (fn(E) \cup n(\sigma)) = \emptyset$ implies $bn(\eta) \cap (fn(F) \cup n(\sigma)) = \emptyset$, therefore $E\sigma \equiv [x\sigma = y\sigma]F\sigma \xrightarrow{\eta\sigma} E'\sigma$ by Cond$_1$, as required.
4. $E \equiv \nu x F$. $E \xrightarrow{M,\eta} E'$ can only be inferred finally by either OPEN, RES or RES0. We only show the case for RES0, others can be proved similarly.

   When $E \xrightarrow{M,\eta} E'$ is inferred finally by RES0, then $\eta = \epsilon(d)$, and $F \xrightarrow{M',\epsilon(d)} F'$ such that $\nu x F' = E'$, $M = M' \setminus \{x \neq y | x \neq y \in M'\}$. By induction assumption, for any $\sigma \models M'$ and $bn(\eta) \cap (fn(F) \cup n(\sigma)) = \emptyset$, $F\sigma \xrightarrow{\epsilon(d)} F'\sigma$. Without loss of generality, we only have to consider $x \notin n(\sigma)$, where $\sigma \models M$ and $\sigma \models M'$. By *Res*, $E\sigma \equiv (\nu x F)\sigma = \nu x F\sigma \xrightarrow{\epsilon(d)} \nu x F'\sigma \equiv (\nu x F')\sigma = E'\sigma$ as required.
5. Other cases can be similarly proved. $\square$

Now we need a *weak* version of Lemma 2.

**Lemma 15** *Suppose $E \overset{a(y)}{\Longrightarrow} E'$ where $a = x$ or $a = \overline{x}$ and that $z \notin n(E)$. Then equally for some $E'' \equiv_\alpha E'\{z/y\}$, $E \overset{a(z)}{\Longrightarrow} E''$.*

**Proof.** By induction on why $E \overset{a(y)}{\Longrightarrow} E'$, using Lemma 2. $\square$

**Definition 8** *The symbolic weak transitions are defined as the least relations satisfying the following rules:*

- $E \overset{true,\epsilon}{\Longrightarrow} E$
- $E \xrightarrow{M,\eta} E'$ implies $E \overset{M,\eta}{\Longrightarrow} E'$
- $E \xrightarrow{M,\tau}\overset{N,\eta}{\Longrightarrow} E'$ implies $E \overset{MN,\eta}{\Longrightarrow} E'$
- If $\eta$ is not of the form $a(x)$ (input action), then $E \overset{M,\eta}{\Longrightarrow}\xrightarrow{N,\tau} E'$ implies $E \overset{MN,\eta}{\Longrightarrow} E'$

Let $\overset{M}{\Longrightarrow}$ be $\overset{M,\epsilon}{\Longrightarrow}$, and $\overset{M,\hat{\eta}}{\Longrightarrow}$ be $\overset{M,\eta}{\Longrightarrow}$ for $\eta \not\equiv \tau$.

**Lemma 16**

1. If $E\sigma \overset{\eta}{\Longrightarrow} P$, $bn(\eta) \cap (fn(E) \cup n(\sigma)) = \emptyset$, then there exist $M, \eta', E'$ such that $\sigma \models M$, $\eta = \eta'\sigma$, $P \equiv E'\sigma$ and $E \overset{M,\eta'}{\Longrightarrow} E'$.
2. If $E \overset{M,\eta}{\Longrightarrow} E'$, then for any $\sigma \models M$ with $bn(\eta) \cap (fn(E) \cup n(\sigma)) = \emptyset$, $E\sigma \overset{\eta\sigma}{\Longrightarrow} P \equiv E'\sigma$.

**Proof.**

1. Apply induction on why $E\sigma \stackrel{\eta}{\Longrightarrow} P$.
   - $E\sigma \stackrel{\epsilon}{\Longrightarrow} E\sigma$. Then $E \stackrel{true,\epsilon}{\Longrightarrow} E$.
   - $E\sigma \stackrel{\eta}{\longrightarrow} P$. Then by Lemma 13, there exist $M, \eta', E'$ such that $\sigma \models M$, $\eta = \eta'\sigma$, $P \equiv E'\sigma$ and $E \stackrel{M,\eta'}{\longrightarrow} E'$. Hence $E \stackrel{M,\eta'}{\Longrightarrow} E'$.
   - $E\sigma \stackrel{\eta}{\Longrightarrow} P$ because $E\sigma \stackrel{\tau}{\longrightarrow} Q \stackrel{\eta}{\Longrightarrow} P$. By Lemma 13, there exist $M, E''$ such that $\sigma \models M$, $Q \equiv E''\sigma$ and $E \stackrel{M,\tau}{\longrightarrow} E''$. Now $E''\sigma \stackrel{\eta}{\Longrightarrow} P$. So by induction hypothesis there exist $N, \eta', E'$ such that $\sigma \models N$, $\eta = \eta'\sigma$, $P \equiv E'\sigma$ and $E'' \stackrel{N,\eta'}{\Longrightarrow} E'$. Hence $E \stackrel{MN,\eta'}{\Longrightarrow} E'$ and $\sigma \models MN$.
   - $\eta$ is not an input action and $E\sigma \stackrel{\eta}{\Longrightarrow} P$ because $E\sigma \stackrel{\eta}{\Longrightarrow} Q \stackrel{\tau}{\longrightarrow} P$. By Lemma 13 and induction, similar to the above case.
2. Apply induction on why $E \stackrel{M,\eta}{\Longrightarrow} E'$. Details are similar to the above case.

□

We need to define symbolic equivalence between actions:

**Definition 9** $\eta =^C \eta'$ *if*

$$\begin{array}{ll} \text{If } \eta \equiv \tau \text{ or } \epsilon(d), & \text{then } \eta = \eta' \\ \text{If } \eta \equiv \overline{a}x, & \text{then } \eta' \equiv \overline{b}x \text{ and } C \Rightarrow a = b \wedge x = y \\ \text{If } \eta \equiv \overline{a}(x), & \text{then } \eta' \equiv \overline{b}(x) \text{ and } C \Rightarrow a = b \\ \text{If } \eta \equiv a(x), & \text{then } \eta' \equiv b(x) \text{ and } C \Rightarrow a = b \end{array}$$

**Definition 10 (Weak Symbolic Bisimulation)** *A condition indexed family of symmetric relations $\mathcal{S} = \{S^C\}$ is a weak symbolic bisimulation, if $(E, F) \in S^C$ implies:*

Whenever $E \stackrel{M,\eta}{\longrightarrow} E'$ with $bn(\eta) \cap fn(E, F, C) = \emptyset$, then for each $C' \in MCE_{fn(E,F)}(C \cup M)$ there is a $F \stackrel{N,\hat{\eta}'}{\Longrightarrow} F'$ such that $C' \Rightarrow N, \eta =^{C'} \eta'$, and

- If $\eta \equiv a(x)$, then for each $D \in MCE_{fn(E,F)\cup\{x\}}(C')$, there is $F' \stackrel{N'}{\Longrightarrow} F''$ such that $D \Rightarrow N'$ and $(E', F'') \in S^D$.
- If $\eta \equiv \bar{a}(x)$ then $(E', F') \in S^{C' \cup \{x \neq y \mid y \in fn(\eta.E', \eta'.F')\}}$.
- Otherwise $(E', F') \in S^{C'}$.

$E \approx^C F$ if and only if there exists a weak symbolic bisimulation $\mathcal{S} = \{S^C\}$ such that $(E, F) \in S^C$. □

**Lemma 17** $E \approx^C F$ *if and only if* $E \approx^D F$ *for any* $D \in MCE_{fn(E,F)}(C)$.

**Proof.** This lemma is the Proposition 2.14 in [10].

($\Rightarrow$) Suppose $E \approx^C F$ and $D \in MCE_{fn(E,F)}(C)$. So there is a late weak symbolic bisimulation $\mathcal{S}$ such that $(E, F) \in S^C \in \mathcal{S}$. Set $S' = \{(E, F)\}$ and $\mathcal{S}' = \mathcal{S} \cup \{S'\}$.

To show $\mathcal{S}'$ is also a symbolic bisimulation we only need to check the pair $(E, F) \in S'$. Let $E \stackrel{M,\eta}{\longrightarrow} E'$ with $bn(\eta) \cap fn(E, F, D) = \emptyset$. Because $D \Rightarrow C$, for each $D' \in MCE_{fn(E,F)}(D \cup M)$ we also have $D' \in MCE_{fn(E,F)}(C \cup M)$. Using the fact that $(E, F) \in S^C$ and $\mathcal{S}$ is a symbolic bisimulation the rest of the proof is immediate.

($\Leftarrow$) Suppose for every $D \in MCE_{fn(E,F)}(C)$ we have $E \approx^D F$, i.e. there is a weak symbolic bisimulation $\mathcal{S}_D$ such that $(E, F) \in S^D$ for some $S^D \in \mathcal{S}_D$. Set $S'' = \{(E, F)\}$ and $\mathcal{S} = (\bigcup\{\mathcal{S}_D \mid D \in MCE_{fn(E,F)}(C)\}) \cup \{S''\}$.

To show $\mathcal{S}$ is a symbolic bisimulation we only need to check the pair $(E, F) \in S''$. Let $E \stackrel{M,\eta}{\longrightarrow} E'$ with $bn(\eta) \cap fn(E, F, C) = \emptyset$. For any $D \in MCE_{fn(E,F)}(C \cup M)$ we also have $D \in MCE_{fn(E,F)}(D \cup M)$. The rest of the proof is straightforward using the fact that $(E, F) \in S^D \in \mathcal{S}_D$ and $\mathcal{S}_D$ is a symbolic bisimulation. □

**Lemma 18** *Suppose* $C \in MC_{fn(E,F)}$. *Then* $E \approx^C F$ **iff** *for any* $\sigma \models C$, $E\sigma \dot{\approx} F\sigma$.

**Proof.** This Lemma is the Proposition 2.15 in [10].

($\Rightarrow$) Define

$$R = \{(E\sigma, F\sigma) \mid E \approx^C F \textit{ for some } C \in MC_{fn(E,F)}, \textit{ and } \sigma \models C\}$$

We show $R$ is a weak ground bisimulation.

Suppose $(E\sigma, F\sigma) \in R$, i.e. there is a $C \in MC_{fn(E,F)}$ such that $\sigma \models C$ and $E \approx^C F$. Let $E\sigma \xrightarrow{\eta} P$ with $bn(\eta) \cap fn(E\sigma, F\sigma) = \emptyset$. The proof proceeds by case analysis on the type of $\eta$.

- $\eta \equiv a(x)$. Choose a fresh name $w$ s.t. $w \notin fn(E,F) \cup n(\sigma) \cup n(C)$ (hence $w \notin fn(E\sigma, F\sigma)$). By Lemma 2, $E\sigma \xrightarrow{a(w)} P_1$ with $P \equiv P_1\{x/w\}$. Note that $x \notin fn(P_1)$. By Lemma 13 there exist $M, a', E'$ such that $\sigma \models M$, $a \equiv a'\sigma$, $P_1 \equiv E'\sigma$ and $E \xrightarrow{M,a'(w)} E'$. By Lemma 12, $n(M) \subseteq fn(E,F)$. Since $\sigma \models C$, $\sigma \models M$ and $C \in MC_{fn(E,F)}$, by Lemma 11 $C \Rightarrow M$. So $C \cup M = C$. Hence up to logical equivalence, $C$ is the only element in $MCE_{fn(E,F)}(C \cup M)$. By definition, there is $F \overset{N,b'(w)}{\Longrightarrow} F'$ with $C \Rightarrow N$, $C \Rightarrow a' = b'$ and the other properties guaranteed by the definition. From $\sigma \models C$ and $C \Rightarrow N$ follows $\sigma \models N$, hence by Lemma 16
$$F\sigma \overset{b(w)}{\Longrightarrow} Q_1 \equiv F'\sigma \textit{ with } b \equiv b'\sigma \qquad ...(1)$$
Since $C \Rightarrow a' = b'$ and $\sigma \models C$, $\sigma \models a' = b'$. Hence $a = a'\sigma = b'\sigma = b$. Apply Lemma 15 we have
$$F\sigma \overset{b(x)}{\Longrightarrow} Q \text{ with } Q\{w/x\} \equiv Q_1 \qquad ...(2)$$
Now we show for any $y$ there is a $Q'$ such that $Q\{y/x\} \Rightarrow Q'$ and $(P\{y/x\}, Q') \in R$. From $\sigma \models C$ and $w \notin n(C) \cup n(\sigma)$ we know $\sigma\{y/w\} \models C$. Let $D = C \cup \{e \mid e \textit{ has the form } w = z \textit{ or } w \neq z, z \in fn(E,F), \sigma\{y/w\} \models e\}$. Then, since $C \in MC_{fn(E,F)}$,
$$D \in MCE_{fn(E,F)\cup\{w\}}(C) \text{ and } \sigma\{y/w\} \models D \qquad ...(3)$$
Since $E \approx^C F$, there is $F' \overset{N'}{\Longrightarrow} F''$ such that $D \Rightarrow N'$ (hence $\sigma\{y/w\} \models N'$) and $E' \approx^D F''$. By Lemma 16
$$F'\sigma\{y/w\} \Rightarrow Q' \text{ for some } Q' \equiv F''\sigma\{y/w\} \qquad ...(4)$$
From (1) and (2), $F'\sigma \equiv Q\{w/x\}$, so $F'\sigma\{y/w\} \equiv Q\{w/x\}\{y/w\}$. If $w \equiv x$ then $Q\{w/x\}\{y/w\} \equiv Q\{y/x\}$ holds trivially. If $w \not\equiv x$ then from $fn(Q) \subseteq fn(F\sigma) \cup \{x\}$ we know $w \notin fn(Q)$, so $Q\{w/x\}\{y/w\} \equiv Q\{y/x\}$ also. Therefore $F'\sigma\{y/w\} \equiv Q\{y/x\}$. From (4) we have $Q\{y/x\} \Rightarrow Q' \equiv F''\sigma\{y/w\}$. Now from (3) and $fn(E', F'') \subseteq fn(E,F) \cup \{w\}$, we have $D \in MC_{fn(E',F'')}$. Hence by the construction of $R$, $(E'\sigma\{y/w\}, F''\sigma\{y/w\}) \in R$. Since $x \notin fn(P_1)$, $P\{y/x\} \equiv P_1\{x/w\}\{y/x\} \equiv P_1\{y/w\} \equiv E'\sigma\{y/w\}$. Moreover, $Q' \equiv F''\sigma\{y/w\}$. Therefore $(P\{y/x\}, Q') inR$.

- $\eta \equiv \bar{a}(x)$. We proceed in the same way as in the previous case until (2), with $a, a', b', b$ replaced by $\bar{a}, \overline{a'}, \overline{b'}, \bar{b}$, respectively, and $E' \approx^D F'$ where $D = C \cup \{w \neq y \mid y \in fn(\overline{a'}(w).E', \overline{b'}(w).F')\}$. Now we have $D \in MC_{fn(E',F')}$. From $\sigma \models C$ and $w \notin n(C) \cup n(\sigma)$ follows $\sigma\{x/w\} \models C$. Furthermore, since $x \notin fn(E\sigma, F\sigma)$ and $fn(\overline{a'}(w).E', \overline{b'}(w).F') \subseteq fn(E,F)$, $\sigma\{x/w\} \models w \neq y$ for every $y \in fn(\overline{a'}(w).E', \overline{b'}(w).F')$. Hence $\sigma\{x/w\} \models D$. Therefore, by the construction of $R$, $(E'\sigma\{x/w\}, F'\sigma\{x/w\}) \in R$. Since $w \notin fn(Q)$, $Q \equiv Q\{w/x\}\{x/w\} \equiv Q_1\{x/w\} \equiv F'\sigma\{x/w\}$. Moreover $P \equiv P_1\{x/w\} \equiv E'\sigma\{x/w\}$. Therefore $(E,F) \in R$.

Other cases are similar and easier.

($\Leftarrow$) Define

$$S^C = \{(E,F) \mid C \in MC_{fn(E,F)},\ E\sigma \dot{\approx} F\sigma \textit{ for some } \sigma \models C\}$$

and $\mathcal{S} = \{S^C\}$. We show $\mathcal{S}$ is a weak symbolic bisimulation.

Suppose $(E,F) \in S^C$, i.e. $C \in MC_{fn(E,F)}$, $E\sigma \dot{\approx} F\sigma$ for some $\sigma \models C$. Let $E \xrightarrow{M,\eta} E'$ with $bn(\eta) \cap fn(E,F,C) = \emptyset$. We may assume $bn(\eta) \cap n(\sigma) = \emptyset$. By Lemma 12, $n(M) \subseteq fn(E,F)$. For each condition in $MCE_{fn(E,F)}(C \cup M)$ we need to find a matching symbolic transition from $F$. If $M$ is inconsistent with $C$ then $MCE_{fn(E,F)}(C \cup M)$ is empty and we are done; otherwise, since $C \in MC_{fn(E,F)}$ and $n(M) \subseteq fn(E,F)$, we have $C \Rightarrow M$ by Lemma 11. Hence $MCE_{fn(E,F)}(C \cup M) = \{C\}$ and $\sigma \models M$. By Lemma 14 $E\sigma \stackrel{\eta\sigma}{\longrightarrow} P \equiv E'\sigma$. Again there are cases to consider according to the type of $\eta$.

- $\eta \equiv a(x)$. Since $E\sigma \dot{\approx} F\sigma$, there is $F\sigma \stackrel{\eta\sigma}{\Longrightarrow} Q'$ such that for all $y$ there exists $Q'\{y/x\} \Rightarrow Q$ and $P\{y/x\} \dot{\approx} Q$. Apply Lemma 16 we get $N, \eta', F'$ such that $\sigma \models N, \eta'\sigma = \eta\sigma, F'\sigma \equiv Q'$ and $F \stackrel{N,\eta'}{\Longrightarrow} F'$. Since $fn(\eta, \eta') \subseteq fn(E,F)$ and $C$ is maximally consistent on $fn(E,F)$, by Lemma 11 $\eta =^C \eta'$.

  Let $D \in MCE_{fn(E,F)\cup\{x\}}(C)$ we must show there is $F' \stackrel{N'}{\Longrightarrow} F''$ such that $D \Rightarrow N'$ and $(E', F'') \in S^D$. Let

  $$y = \begin{cases} z\sigma & \textit{If } D \Rightarrow x = z \textit{ for some } z \in fn(E,F) \\ x & \textit{otherwise} \end{cases}$$

  The well-definedness of $y$ can be argued as follows: if $D \Rightarrow x = z'$ and $D \Rightarrow x = z''$ for $z', z'' \in fn(E,F)$ then $D \Rightarrow z' = z''$, i.e. $z' = z''$ is consistent with $C$. Since $z', z'' \in fn(E,F)$ and $C$ is maximally consistent on $fn(E,F)$, $C \Rightarrow z' = z''$. Therefore $z'\sigma = z''\sigma$.

  We claim $\sigma\{y/x\} \models D$: for any $e \in D$, if $e \in C$ then $\sigma \models e$, so $\sigma\{y/x\} \models e$ because $x \notin n(e)$. Otherwise $e$ must be of the form $x = z$ or $x \neq z$ with $z \in fn(E,F)$. In the former case $x(\sigma\{y/x\}) = z\sigma$ hence $\sigma\{y/x\} \models x = z$; in the latter case $x(\sigma\{y/x\}) = x$ hence $\sigma\{y/x\} \models x \neq z$. Therefore $\sigma\{y/x\} \models e$ holds in any case.

  Now $F'\sigma\{y/x\} \equiv Q'\{y/x\} \Rightarrow Q$, so by Lemma 16 there exist $N', F''$ such that $\sigma\{y/x\} \models N'$, $F''\sigma\{y/x\} \equiv Q$ and $F' \stackrel{N'}{\Longrightarrow} F''$. Since $D \in MC_{fn(E,F)}\cup\{x\}$ and $n(N') \subseteq fn(E,F) \cup \{x\}$, $D \Rightarrow N'$. Since $E'\sigma\{y/x\} \equiv P\{y/x\}$ and $P\{y/x\} \dot{\approx} Q$, $E'\sigma\{y/x\} \dot{\approx} F''\sigma\{y/x\}$ as well. Furthermore $D \in MC_{fn(E',F'')}$ since $fn(E', F'') \subseteq fn(E,F) \cup \{x\}$. Therefore $(E', F'') \in S^D$.

- $\eta \equiv \bar{a}(x)$. Since $E\sigma \dot{\approx} F\sigma$, there is $F\sigma \stackrel{\eta\sigma}{\Longrightarrow} Q$ such that $P \dot{\approx} Q$. Apply Lemma 16 we get $N, b, F'$ such that $\sigma \models N$, $\sigma \models a = b$, $F'\sigma \equiv Q$ and $F \stackrel{N,\bar{b}(x)}{\Longrightarrow} F'$. Since $n(N) \cup \{a, b\} \subseteq fn(E,F)$, $\sigma \models C$ and $C$ is maximally consistent on $fn(E,F)$, we have $C \Rightarrow N$ and $\bar{a}(x) =^C \bar{b}(x)$. Moreover, since $x \notin fn(E,F) \cup n(\sigma)$ and $fn(\bar{a}(x).E', \bar{b}(x).F') \subseteq fn(E,F) \cup \{x\}$, we know $C \cup \{x \neq y \mid y \in fn(\bar{a}(x).E', \bar{b}(x).F')\} \in MC_{fn(\bar{a}(x).E', \bar{b}(x).F')}$. Finally from $x \notin fn(E,F) \cup n(\sigma)$ it follows that $\sigma \models \{x \neq y \mid y \in fn(\bar{a}(x).E', \bar{b}(x).F')\}$, hence $\sigma \models C \cup \{x \neq y \mid y \in fn(\bar{a}(x).E', \bar{b}(x).F')\}$. Therefore $(E', F') \in S^{C\cup\{x\neq y \mid y \in fn(\bar{a}(x).E', \bar{b}(x).F')\}}$.

Other cases are similar and easier. □

Combining the above two results gives the following conclusion.

**Theorem 9** *$E \approx^C F$ if and only if $E\sigma \dot{\approx} F\sigma$ for every $\sigma \models C$.*

**Proof.** By Lemma 17 $E \approx^C F$ if and only if $E \approx^D F$ for every $D \in MCE_{fn(E,F)}(C)$. By Lemma 9, $\sigma \models C$ if and only if $\sigma \models D$ for some $D \in MCE_{fn(E,F)}(C)$. By Lemma 18, for any $D \in MCE_{fn(E,F)}(C)$, $E \approx^D F$ if and only if $E\sigma \dot{\approx} F\sigma$ for any $\sigma \models D$. Hence the conclusion. □

Since $\dot{\approx}$ is not preserved by '+' (neither is $\approx$), we have defined $\dot{\simeq}$ (and furthermore $\simeq$) based on $\dot{\approx}$. So in the following, for the *Symbolic* version, we will define $\simeq^C$ based on $\approx^C$. First, we need to define the *Symbolic* version of $\stackrel{\eta}{\longrightarrow}_p$:

**Definition 11** $\stackrel{M,\eta}{\longrightarrow}_p$ *($\eta \in \Lambda \cup \{\epsilon\}$) is defined as the least relations satisfying the following rules:*

- $E \stackrel{M,\epsilon(p)}{\longrightarrow} E'$ implies $E \stackrel{M,\epsilon}{\longrightarrow}_p E$
- $E \stackrel{M,\epsilon(p)}{\longrightarrow}\stackrel{N,\alpha}{\longrightarrow} E'$ implies $E \stackrel{MN,\alpha}{\longrightarrow}_p E'$

□

**Definition 12** *The weak version of* $\stackrel{M,\eta}{\longrightarrow}_p$ *($\eta \in \Lambda \cup \{\epsilon\}$) is defined as:*

- $E \stackrel{M,\eta}{\longrightarrow}_p E'$ implies $E \stackrel{M,\eta}{\Longrightarrow}_p E$
- $E \stackrel{M,\epsilon(p)}{\longrightarrow}\stackrel{N,\alpha}{\Longrightarrow} E'$ implies $E \stackrel{MN,\alpha}{\Longrightarrow}_p E'$

$\square$

**Definition 13 (Weak Symbolic Congruence)** *E and F are* weakly symbolically congruent *with respect to C, written $E \simeq^C F$, if whenever $E \stackrel{M,\eta}{\longrightarrow}_p E'$ with $bn(\eta) \cap fn(E,F,C) = \emptyset$, then for each $C' \in MCE_{fn(E,F)}(C \cup M)$ there is a $F \stackrel{N,\eta'}{\Longrightarrow}_p F'$ such that $C' \Rightarrow N, \eta =^{C'} \eta'$, and*

- If $\eta \equiv a(x)$, then for each $D \in MCE_{fn(E,F)\cup\{x\}}(C')$, there is $F' \stackrel{N'}{\Longrightarrow} F''$ such that $D \Rightarrow N'$ and $E' \approx^D F''$.
- If $\eta \equiv \bar{a}(x)$ then $E' \approx^{C' \cup \{x \neq y | y \in fn(\eta.E', \eta'.F')\}} F'$.
- Otherwise $E' \approx^{C'} F'$.

And similarly for $F$. $\square$

**Theorem 10 (Symbolic vs. Concrete)** $E \simeq^C F$ **iff** *for any $\sigma \models C$, $E\sigma \doteq F\sigma$.*

**Proof.** Similar to that of Theorem 9. $\square$

According to the above result, Symbolic Bisimulation $\simeq^{true}$ coincides with the Weak Congruence $\simeq$ defined previously. In the following we will use symbolic bisimulation technique to axiomatize $\simeq$.

## 3.2 Axioms and Inference Rules

In this subsection we will present the axiom system $\mathcal{SW}$ for Weak Congruence. We use the basic axioms of equivalence, the rule "if $E \equiv F$ then $E = F$" and "if $A(\tilde{x}) \stackrel{def}{=} E$ then $A(\tilde{v}) = E\{\tilde{v}/\tilde{x}\}$" as the fundamental axioms in $\mathcal{SW}$. The axiom system also contains the *Expansion Law* (Theorem 7), together with axioms for summation ($\mathbf{S}_0$-$\mathbf{S}_3$), axioms for restriction ($\mathbf{R}_0$-$\mathbf{R}_4$) and Real-time properties namely **Maximal Progress (MP)**, **Time Determinacy (TD)**, **Time Continuity (TC)** and **Time Persistency (TP)** in Fig. 3 (which has been proved to be true for Strong Bisimulation in [3], therefore also hold for $\simeq$).

| | | | |
|---|---|---|---|
| $\mathbf{S}_0$ | $X + 0 = X$ | $\mathbf{R}_0$ | $\nu y \mathbf{0} = \mathbf{0}$ |
| $\mathbf{S}_1$ | $X + X = X$ | $\mathbf{R}_1$ | $\nu y \nu z X = \nu z \nu y X$ |
| $\mathbf{S}_2$ | $X + Y = Y + X$ | $\mathbf{R}_2$ | $\nu y(X + Y) = \nu y X + \nu y Y$ |
| $\mathbf{S}_3$ | $(X + Y) + Z = X + (Y + Z)$ | $\mathbf{R}_3$ | $\nu y \eta.X = \eta.\nu y X$ *if* $y \notin n(\eta)$ |
| | | $\mathbf{R}_4$ | $\nu y \eta.X = \mathbf{0}$ *if* $y = subj(\eta)$ |
| **MP** | $\tau.X + \epsilon(d).Y = \tau.X,\ d > 0$ | **TD** | $\epsilon(e).(X + Y) = \epsilon(e).X + \epsilon(e).Y$ |
| $\mathbf{TC}_1$ | $\epsilon(e + f).X = \epsilon(e).\epsilon(f).X$ | $\mathbf{TC}_2$ | $\epsilon(0).X = X$ |
| $\mathbf{TC}_3$ | $\epsilon(e).\mathbf{0} = \mathbf{0}$ | **TP** | $\alpha.X + \epsilon(e).\alpha.X = \alpha.X$ |

Fig. 3: Axioms for Choice, Restriction and Real-time properties

Besides, we also need three *timed* $\tau-$laws as in the case of CCS.

**Theorem 11** *The following equations hold when $=$ is interpreted to be $\simeq$.*

T1 $\tau.E + E = \tau.E$

T2 $\alpha.(E + \tau.F) = \alpha.(E + \tau.F) + \alpha.F$

T3 $\alpha.\sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i = \alpha.[\sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i + \epsilon(e).\tau.\sum_{i\in I} M_i\epsilon(e_i \dot{-} e).\alpha_i.E_i]$

**Proof.** The first two are straightforward. We show the last.

To show $\alpha.E \simeq \alpha.F$, i.e. $\alpha.E \simeq^{true} \alpha.F$, by definition 13, we must show $\forall C \in MC_{fn(\alpha.E,\alpha.F)}$, $\alpha.E \simeq^C \alpha.F$. Therefore, it is sufficient to show $\forall D \in MC_{fn(E,F)}$, $E \approx^D F$. So we only need to prove $\forall D \in MC_{fn(E,F)}$,

$$\sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i \approx^D \sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i + \epsilon(e).\tau.\sum_{i\in I} M_i\epsilon(e_i \dot{-} e).\alpha_i.E_i$$

And further by Lemma 18, to prove $E \approx^D F$ we only need to show $\forall\sigma \models D$, $E\sigma \dot{\approx} F\sigma$.

Given a $\sigma \models D$, If $\sigma \not\models M_i$ then $(M_i E_i)\sigma \dot{\approx} \mathbf{0}$. Suppose $(\alpha_i.E_i)\sigma \equiv \alpha_i'.E_i'$, by $(\epsilon(d).E)\sigma = \epsilon(d).E\sigma$ and $X + \mathbf{0} \dot{\approx} X$, we have

$$(\textstyle\sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i)\sigma \dot{\approx} \sum_{\sigma\models M_i} \epsilon(e_i).\alpha_i'.E_i'$$

and

$$(\textstyle\sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i + \epsilon(e).\tau.\sum_{i\in I} M_i\epsilon(e_i \dot{-} e).\alpha_i.E_i)\sigma \dot{\approx} \sum_{\sigma\models M_i} \epsilon(e_i).\alpha_i'.E_i' + \epsilon(e).\tau.\sum_{\sigma\models M_i} \epsilon(e_i \dot{-} e).\alpha_i'.E_i'$$

So we only need to show

$$\sum_{\sigma\models M_i} \epsilon(e_i).\alpha_i'.E_i' \dot{\approx} \sum_{\sigma\models M_i} \epsilon(e_i).\alpha_i'.E_i' + \epsilon(e).\tau.\sum_{\sigma\models M_i} \epsilon(e_i \dot{-} e).\alpha_i'.E_i'$$

Let

$$P_t \equiv \sum_{i\in I} \epsilon(e_i \dot{-} t).\alpha_i'.E_i'$$

Then, it can be rewritten as follows:

$$P_0 \dot{\approx} P_0 + \epsilon(e).\tau.P_e$$

And by $(\epsilon(e).E)\sigma \equiv \epsilon(e).E\sigma$, it is enough to show

$$P_0 \dot{\approx} P_0 + \epsilon(e).\tau.P_e$$

We show the general case,

$$P_f \dot{\approx} P_f + \epsilon(e).\tau.P_{e+f}$$

Let

$$R = \{(P_f, P_f + \epsilon(e).\tau.P_{e+f}) \mid e, f \in \mathcal{R}^{\geq 0}\} \cup \mathbf{Id}$$

We check $R$ is a weak ground bisimulation (where $\mathbf{Id} = \{(P,P)\}$).

1. Assume $P_f \xrightarrow{\epsilon(d)} P_{f+d}$. If $d \leq e$, we have $P_f + \epsilon(e).\tau.P_{e+f} \xrightarrow{\epsilon(d)} P_{f+d} + \epsilon(e \dot{-} d).\tau.P_{e+f} = P_{f+d} + \epsilon(e \dot{-} d).\tau.P_{(e \dot{-} d)+(f+d)}$ and $(P_{f+d}, P_{f+d} + \epsilon(e \dot{-} d).\tau.P_{(e \dot{-} d)+(f+d)}) \in R$.

   If $d > e$, we have $P_f + \epsilon(e).\tau.P_{e+f} \xrightarrow{\epsilon(e)}\xrightarrow{\tau} P_{e+f} \xrightarrow{\epsilon(d-e)} P_{e+f+(d-e)} = P_{f+d}$ and $(P_{f+d}, P_{f+d}) \in \mathbf{Id} \subseteq R$, as required.

2. Assume $P_f \xrightarrow{\alpha_i'} E_i'$. We must have $e_i - f = 0$ and $P_f + \epsilon(e).\tau.P_{e+f} \xrightarrow{\alpha_i'} E_i'$ and $(E_i', E_i') \in R$ as required.

Similarly we can show whenever $P_f + \epsilon(e).\tau.P_{e+f} \xrightarrow{\eta} Q$ then there exists $P$ such that $P_f \xRightarrow{\hat{\eta}} P$ and $(P,Q) \in R$. □

**Remarks**: As mentioned earlier in the introduction, when there is neither time nor name matching, i.e. all $\epsilon(d)$ and $M_i$ are dropped (or equally, replaced by $\epsilon(0)$ and $true$, respectively), T3 will become the simplified $\alpha.\tau.E = \alpha.(E + \tau.E)$, therefore T3+T1 implies the standard '$\alpha.\tau.E = \alpha.E$', and those CCS $\tau$-laws are obviously the special cases of T1,T2 and T3 in this paper.

The remained part of $\mathcal{SW}$ consists of the inference rules shown in Fig. 4.

We write $\mathcal{SW} \vdash C \triangleright E = F$ or simply $\vdash C \triangleright E = F$ if $C \triangleright E = F$ can be inferred by $\mathcal{SW}$.

**Theorem 12 (Soundness)** *For any processes $E, F$, if $C \triangleright E = F$ then $E \simeq^C F$.*

**Proof.** Since $\simeq^{true}$ is exactly $\simeq$, so the axioms in $\mathcal{SW}$ are valid. The remained can be done by Theorem 10, and by induction on the depth of the proof tree of $C \triangleright E = F$ :

$$AXIOM\ \frac{}{true \rhd E=F}\quad E=F \text{ is an axiom instance}$$

$$CHOICE\ \frac{C \rhd E_i=F_i}{C \rhd E_1+E_2=F_1+F_2}\quad i\in\{1,2\}$$

$$DELAY\ \frac{C \rhd E=F}{C \rhd \epsilon(d).E=\epsilon(d).F}$$

$$TAU\ \frac{C \rhd E=F}{C \rhd \tau.E=\tau.F}$$

$$INPUT\ \frac{C \rhd E=F}{C \rhd a(x).E=b(x).F}\quad C\Rightarrow a=b,\ x\notin n(C)$$

$$OUTPUT\ \frac{C \rhd E=F}{C \rhd \overline{a}x.E=\overline{b}y.F}\quad C\Rightarrow a=b\wedge x=y$$

$$MATCH\ \frac{C\cup\{x=y\} \rhd E=F \qquad C\cup\{x\neq y\} \rhd \mathbf{0}=F}{C \rhd [x=y]E=F}$$

$$IR\_RES\ \frac{C\cup\{x\neq y \mid y\in fn(\nu xE,\nu xF)\} \rhd E=F}{C \rhd \nu xE=\nu xF}\quad x\notin n(C)$$

$$PARTITION\ \frac{C\cup\{x=y\} \rhd E=F \qquad C\cup\{x\neq y\} \rhd E=F}{C \rhd E=F}$$

$$CONSEQ\ \frac{C \rhd E=F}{C' \rhd E=F}\quad C'\Rightarrow C$$

$$ABSURD\ \frac{}{false \rhd E=F}$$

Fig. 4: Inference Rules of $\mathcal{SW}$

1. If $C \rhd E=F$ is proved by finally applying rule $CHOICE$, then without loss of generality, $E\equiv E_1+E_2$ and $F\equiv F_1+F_2$. By $C \rhd E_i=F_i$, $E_i\sigma \doteq F_i\sigma$ for any $\sigma\models C$. And clearly $E_1\sigma+E_2\sigma \doteq F_1\sigma+F_2\sigma$, i.e., $(E_1+E_2)\sigma \doteq (F_1+F_2)\sigma$. Therefore by the arbitrary of $\sigma$, $C \rhd E_1+E_2=F_1+F_2$.
2. If $C \rhd E=F$ is proved by finally applying rule $DELAY$, then $E\equiv \epsilon(d).E'$ and $F\equiv \epsilon(d).F'$. By $C \rhd E'=F'$, $E'\sigma \doteq F'\sigma$ for any $\sigma\models C$. Easy to see, $\epsilon(d).E'\sigma \doteq \epsilon(d).F'\sigma$. Thus $(\epsilon(d).E')\sigma=\epsilon(d).E'\sigma \doteq \epsilon(d).F'\sigma=(\epsilon(d).F')\sigma$, and by the arbitrary of $\sigma$, $C \rhd \epsilon(d).E'=\epsilon(d).F'$.
3. If $C \rhd E=F$ is proved by finally applying rule $INPUT$, then $E\equiv a(x).E'$ and $F\equiv b(x).F'$. By $C \rhd E=F$, $E'\sigma \doteq F'\sigma$ for any $\sigma\models C$ and $a\sigma=b\sigma$. Take $y\notin fn(E',F')\cup n(\sigma)\cup fn(C)$ and let $\sigma'=\{y/x\}$, then $(a(x).E')\sigma\equiv a\sigma(y).E'\sigma'\sigma$, $(b(x).F')\sigma\equiv b\sigma(y).F'\sigma'\sigma$. Since $x,y\notin fn(C)$, $\sigma'\sigma\models C$ and furthermore $E'\sigma'\sigma \doteq F'\sigma'\sigma$, therefore $E'\sigma'\sigma \stackrel{\cdot}{\approx} F'\sigma'\sigma$. Then it is easy to check that $(a(x).E')\sigma \doteq (b(x).F')\sigma$. By the arbitrary of $\sigma$, we have $C \rhd a(x).E'=b(x).F'$.
4. Other cases can be proved similarly. □

**Theorem 13**

1. If $C\Rightarrow M$ and $\vdash C \rhd E=F$, then $\vdash C \rhd ME=F$.
2. If $C\cup M\Rightarrow false$ then $\vdash C \rhd ME=\mathbf{0}$.

**Proof.** It comes from Proposition 3.1 in [10].

1. Only need to prove: If $C\Rightarrow x=y$ and $\vdash C \rhd E=F$ then $\vdash C \rhd [x=y]E=F$.
   Since $C\Rightarrow x=y$, $C\cup\{x\neq y\}\Rightarrow false$ and $C\cup\{x=y\}\Leftrightarrow C$. By $ABSURD$,

$$false \rhd \mathbf{0}=F$$

   Then by $CONSEQ$,

$$C\cup\{x\neq y\} \rhd \mathbf{0}=F$$

On the other hand, by $\vdash C \rhd E = F$ and $CONSEQ$,

$$C \cup \{x = y\} \rhd E = F$$

and finally applying $MATCH$, $\vdash C \rhd ME = F$.

2. Only need to prove: If $C \cup \{x = y\} \Rightarrow false$ then $\vdash C \rhd [x = y]E = \mathbf{0}$.
   By $ABSURD$, $false \rhd E = \mathbf{0}$, and then by $CONSEQ$,

$$C \cup \{x = y\} \rhd E = \mathbf{0}$$

   Clearly $C \cup \{x \neq y\} \rhd \mathbf{0} = \mathbf{0}$, and by applying $MATCH$, $\vdash C \rhd [x = y]E = \mathbf{0}$. □

**Theorem 14** $\vdash E = E + ME$

**Proof.** It comes from Proposition 3.1 in [10].

Apply induction on the length of $M$. When $M$ is empty the result defaults to $\mathbf{S}_1$. For the induction step let $M \equiv [x = y]N$. Using $PARTITION$ we need to show

$$\vdash x = y \rhd E = E + [x = y]NE \quad and \quad \vdash x \neq y \rhd E = E + [x = y]NE$$

By Theorem 13 we have $\vdash x = y \rhd NE = [x = y]NE$, so using the induction hypothesis we derive

$$\begin{aligned} \vdash x = y \rhd E &= E + NE \\ &= E + [x = y]NE \end{aligned}$$

For the second subgoal we have $\vdash x \neq y \rhd [x = y]NE = \mathbf{0}$ by Theorem 13, hence $\vdash x \neq y \rhd E = E + [x = y]NE$ by $\mathbf{S}_0$. □

**Theorem 15** *If $\vdash D \rhd E = F$ for each $D \in MCE_V(C)$, where $V = fn(E, F)$, then $\vdash C \rhd E = F$.*

**Proof.** This theorem is the proposition 3.2 in [8].

Suppose $\vdash D \rhd E = F$ for each $D \in MCE_V(C)$. By Lemma 8, $D \in MCE_V(C)$ If and only if $D = C \cup I \cup \overline{I}$ for some $I \subseteq E_V(C)$. Apply induction on the cardinality of $E_V(C)$.

If $E_V(C) = \emptyset$, then $C$ is the only maximally consistent extension of itself, and the result is trivial.

Otherwise assume $E_V(C)$ has $n + 1$ elements and let $x = y \in E_V(C)$. We have

$$MCE_V(C) = MCE_V(C \cup \{x = y\}) \cup MCE_V(C \cup \{x \neq y\})$$

and

$$\begin{aligned} E_V(C \cup \{x = y\}) &\subseteq E_V(C) - \{x = y\} \\ E_V(C \cup \{x \neq y\}) &\subseteq E_V(C) - \{x = y\} \end{aligned}$$

So by induction $\vdash C \cup \{x = y\} \rhd E = F$, $\vdash C \cup \{x \neq y\} \rhd E = F$. By $PARTITION$, $\vdash C \rhd E = F$. □

**Theorem 16**

1. $\vdash \nu x[x = x]E = \nu x E$.
2. $\vdash \nu x[x = y]E = \mathbf{0}$.
3. $\vdash \nu x[y = z]E = [y = z]\nu x E$.

**Proof.** This Theorem is the Proposition 3.3 in [10].

1. By $AXIOM$ $true \rhd E = E$, and by $ABSURD$ $false \rhd \mathbf{0} = E$. Clearly $x = x \Rightarrow true$, $x \neq x \Rightarrow false$, and then by $CONSEQ$

$$x = x \rhd E = E \qquad x \neq x \rhd \mathbf{0} = E$$

   thus by $MATCH$, $\vdash [x = x]E = E$. By $IR\_RES$, $\vdash \nu x[x = x]E = \nu x E$ as required.
2. By theorem 13, $x \neq y \rhd [x = y]E = \mathbf{0}$, and by $CONSEQ$ together with $IR\_RES$, $\vdash \nu x[x = y]E = \nu x \mathbf{0}$, and then by axiom $\mathbf{R}_0$, $\vdash \nu x[x = y]E = \mathbf{0}$.
3. By $MATCH$

$$y = z \triangleright [y = z]E = E$$

and then by $IR\_RES$

$$y = z \triangleright \nu x[y = z]E = \nu xE$$

i.e.

$$y = z \triangleright \nu xE = \nu x[y = z]E \qquad \ldots (1)$$

Similarly by theorem 13, $y \neq z \triangleright [y = z]E = \mathbf{0}$, and then by $IR\_RES$ and axiom $\mathbf{R}_0$

$$y \neq z \triangleright \mathbf{0} = \nu x[y = z]E \qquad \ldots (2)$$

Apply $MATCH$ on (1),(2)

$$\vdash [y = z]\nu xE = \nu x[y = z]E$$

□

**Theorem 17** $\vdash C \triangleright E = F \Rightarrow \vdash C \triangleright ME = MF$.

**Proof.** Only need to prove $\vdash C \triangleright E = F \Rightarrow \vdash C \triangleright [x = y]E = [x = y]F$. By $MATCH$ $x = y \triangleright [x = y]F = F$, and then by $CONSEQ$

$$C \cup \{x = y\} \triangleright [x = y]F = F$$

By $C \triangleright E = F$ and $CONSEQ$

$$C \cup \{x = y\} \triangleright E = F$$

therefore

$$C \cup \{x = y\} \triangleright E = [x = y]F$$

By $ABSURD$ and $CONSEQ$ $C \cup \{x \neq y\} \triangleright \mathbf{0} = [x = y]F$, thus by $MATCH$ $\vdash C \triangleright [x = y]E = [x = y]F$. □

**Lemma 19** $\vdash M\epsilon(d).NE = MN\epsilon(d).E$

**Proof.** Only need to prove $\vdash M\epsilon(d).[x = y]E = [x = y]M\epsilon(d).E$.

By $MATCH$

$$x = y \triangleright [x = y]M\epsilon(d).E = M\epsilon(d).E \qquad \ldots (a)$$

Similarly by $MATCH$ $x = y \triangleright [x = y]E = E$, then by $DELAY$ and theorem 17

$$x = y \triangleright M\epsilon(d).[x = y]E = M\epsilon(d).E \qquad \ldots (b)$$

Then by $(a), (b)$

$$x = y \triangleright M\epsilon(d).[x = y]E = [x = y]M\epsilon(d).E \qquad \ldots (1)$$

By theorem 13 $x \neq y \triangleright [x = y]E = \mathbf{0}$, then by $DELAY$ and theorem 17

$$x \neq y \triangleright M\epsilon(d).[x = y]E = M\epsilon(d).\mathbf{0}$$

By $MATCH$, $M\mathbf{0} = \mathbf{0}$, and by $\mathbf{TC}$

$$x \neq y \triangleright M\epsilon(d).[x = y]E = \mathbf{0}$$

and by theorem 13 $x \neq y \triangleright [x = y]M\epsilon(d).E = \mathbf{0}$, therefore

$$x \neq y \triangleright M\epsilon(d).[x = y]E = [x = y]M\epsilon(d).E \qquad \ldots (2)$$

Apply $PARTITION$ on $(1), (2)$ $\vdash M\epsilon(d).[x = y]E = [x = y]M\epsilon(d).E$. □

**Theorem 18** $\vdash M(E + F) = ME + MF$.

**Proof.** It comes from Proposition 3.1 in [10].

Only need to show $\vdash [x = y](E + F) = [x = y]E + [x = y]F$.

By $MATCH$ we have $x = y \triangleright E = [x = y]E$ and $x = y \triangleright F = [x = y]F$, and then by $CHOICE$

$$x = y \triangleright E + F = [x = y]E + [x = y]F$$

Similarly by $MATCH$ and axiom $\mathbf{S}_1$ $\{x \neq y\} \triangleright \mathbf{0} = [x = y]E + [x = y]F$, therefore by $MATCH$, $\vdash [x = y](E + F) = [x = y]E + [x = y]F$. □

We also have the following derived inference rule for bound output:

**Theorem 19** *Suppose* $C \Rightarrow a = b,\ x \notin n(C)$. *If* $\vdash C \cup \{x \neq y | y \in fn(\bar{a}(x).E, \bar{b}(x).F)\} \triangleright E = F$ *then* $\vdash C \triangleright \bar{a}(x).E = \bar{b}(x).F$

**Proof.** Since $\vdash C \cup \{x \neq y | y \in fn(\bar{a}(x).E, \bar{b}(x).F)\} \triangleright E = F$ and $C \Rightarrow a = b$, by *OUTPUT* $\vdash C \cup \{x \neq y | y \in fn(\bar{a}(x).E, \bar{b}(x).F)\} \triangleright \bar{a}x.E = \bar{b}x.F$. By $x \notin n(C)$ and *IR_RES* we have $\vdash C \triangleright \bar{a}(x).E = \bar{b}(x).F$. □

**Remarks**: Intuitively the condition $\{x \neq y | y \in fn(\bar{a}(x).E, \bar{b}(x).F)\}$ just insists on that $x$ should be distinguished from $a$ and $b$, because , $\bar{x}(x).E = \nu x \bar{x}x.E$ has no non-delay actions, therefore $\vdash C \triangleright \bar{a}(x).E = \bar{b}(x).F$ trivially holds. Since $x \notin n(C)$, we have $C \cup \{x \neq y | y \in fn(\bar{a}(x).E, \bar{b}(x).F)\} \Rightarrow C$, and by the inference rule *CONSEQ*, this theorem can be relaxed to be: If $\vdash C \triangleright E = F$ then $\vdash C \triangleright \bar{a}(x).E = \bar{b}(x).F$, where $C \Rightarrow a = b,\ x \notin n(C)$.

**Lemma 20** *For any process expression* $E$*, there exists a* $E'$ *containing no '|' such that* $\vdash E = E'$

**Proof.** By *Expansion Law* (Theorem 7). □

To apply induction, we need to define the depth of terms without '|':

$$\begin{array}{rcl} |\mathbf{0}| &=& 0 \\ |\eta.E| &=& |E| + 1 \\ |[x = y]E| &=& |E| \\ |E + F| &=& max(|E|, |F|) \end{array}$$

And to accomplish our objective, we need to define some kind of *Normal Form* so that processes can be compared in a uniform and syntactic way.

**Definition 14 (Normal form)** *E is a* Normal Form*, if*

1. $E \equiv \mathbf{0}$ or
2. $E \equiv \sum_{i \in I} M_i \epsilon(e_i).\alpha_i.E_i$
   where $M_i$ is a condition, $E_i$ is a normal form. □

**Lemma 21** *For any process expression* $E$ *containing no '|', there must be a normal form* $E'$, $\vdash E = E'$ *and* $|E'| \leq |E|$.

**Proof.** First, by theorem 16 and axioms $\mathbf{R}_0$-$\mathbf{R}_4$, the restriction operator can be pushed inside the expressions until it either disappears or gives rise to a bound output action. Therefore there is a so-obtained $E''$ such that $\vdash E = E''$ and $E''$ doesn't contain restriction operators. This process does not increase the depth of $E$. Thus we only need to apply structural induction on $E''$: If $E''$ is $\mathbf{0}$, then $E''$ is already a normal form. If $E''$ is $M\epsilon(d).\alpha.F$, then by induction, there is a normal form $H'$ satisfying $\vdash F = H'$ and $|H'| \leq |F|$, thus $H = M\epsilon(d).\alpha.H'$ is the normal form s.t. $\vdash E'' = H$, and $|H| \leq |E''|$ as required. If $E'' \equiv E_1 + E_2$, then by induction, there are normal forms $H_1$ and $H_2$ such that $\vdash E_1 = H_1$ and $\vdash E_2 = H_2$, where $|H_1| \leq |E_1|$ and $|H_2| \leq |E_2|$. So $\vdash E'' = H$ where clearly $H \equiv H_1 + H_2$ is a normal form, and $|H| = max(|H_1|, |H_2|) \leq max(|E_1|, |E_2|) = |E''|$. Otherwise, if $E'' \equiv MF$, then the condition $M$ can be distributed inside $F$ by Theorem 18; if $E'' = \epsilon(d).F$ then by induction, assume there is a normal form $H \equiv \sum M_i \epsilon(e_i).\alpha_i.H_i$ such that $\vdash F = H$ and $|H| \leq |F|$. Then by axiom **TD**, $\vdash \epsilon(d).H = \sum \epsilon(d).M_i\epsilon(e_i).\alpha_i.H_i$. And by Lemma 19 $\vdash \epsilon(d).M_i\epsilon(e_i).\alpha_i.H_i = M_i\epsilon(d).\epsilon(e_i).\alpha_i.H_i$, and further by **TC** we have $\vdash M_i\epsilon(d).\epsilon(e_i).\alpha_i.H_i = M_i\epsilon(d + e_i).\alpha_i.H_i$. Thus let $H' \equiv \sum M_i\epsilon(d + e_i).\alpha_i.H_i$ which is a normal form, we have $\vdash E = H'$, and $|H'| = |H| \leq |F| = |E''| - 1$, i.e. $|H'| < |E''|$. □

We will need a generalized form of T2:

**Lemma 22** *If* $n(M) \cap bn(\alpha) = \emptyset$ *then* $\vdash \alpha.(E + M\tau.F) = \alpha.(E + M\tau.F) + M\alpha.F$.

**Proof.** First by Theorem 13 we have $M \triangleright ME = E$. Then by inference rules *TAU, INPUT, OUTPUT* and Theorem 19 we have $M \triangleright \alpha.ME = \alpha.E$, and by $M \triangleright ME = E$ we have $M \triangleright \alpha.E = M(\alpha.E)$, therefore $M \triangleright \alpha.ME = M\alpha.E$, together with $\neg M \triangleright \mathbf{0} = M\alpha.E$, using inference rule *MATCH*, we have

$$\vdash M\alpha.ME = M\alpha.E \qquad ...(1)$$

By Theorem 14, we have $\vdash \alpha.(E + M\tau.F) = \alpha.(E + M\tau.F) + M\alpha.(E + M\tau.F)$. Moreover

$$
\begin{array}{rcll}
\vdash \ M\alpha.(E+M\tau.F) & = & M\alpha.M(E+M\tau.F) & \textit{by (1)} \\
& = & M\alpha.M(E+\tau.F) & \textit{by Theorem } 18 \\
& = & M\alpha.(E+\tau.F) & \textit{by (1)} \\
& = & M(\alpha.(E+\tau.F)+\alpha.F) & \textit{by T2} \\
& = & M\alpha.(E+\tau.F)+M\alpha.F & \textit{by Theorem } 18 \\
& = & M\alpha.(E+M\tau.F)+M\alpha.F &
\end{array}
$$

Hence $\vdash \alpha.(E+M\tau.F) = \alpha.(E+M\tau.F)+M\alpha.F$. □

**Definition 15 (Full Normal form)** *A* normal form $E \equiv \sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i$ *is a* full normal form, *if*

1. $E \overset{M,\alpha}{\Longrightarrow}_p E'$, where $bn(\alpha)\cap n(M)=\emptyset$, implies $E \overset{M,\alpha}{\longrightarrow}_p E'$
2. Each $E_i$ is a full normal form.

□

**Lemma 23** *If* $E \overset{M,\alpha}{\Longrightarrow} E'$ *with* $bn(\alpha)\cap n(M)=\emptyset$ *then* $\vdash E = E+M\alpha.E'$

**Proof.** By induction on why $E \overset{M,\alpha}{\Longrightarrow} E'$.

- $E \overset{M,\alpha}{\longrightarrow} E'$. Apply induction on why $E \overset{M,\alpha}{\longrightarrow} E'$.
  - $E \equiv \alpha.E' \overset{true,\alpha}{\longrightarrow} E'$. Then $\vdash E = E+\alpha.E'$ by $\mathbf{S}_1$.
  - $E \equiv \epsilon(0).E'' \overset{M,\alpha}{\longrightarrow} E'$ because $E'' \overset{M,\alpha}{\longrightarrow} E'$. By induction $\vdash E'' = E''+M\alpha.E'$, and by $\mathbf{TC}_2$ we have $\vdash E = E+M\alpha.E'$.
  - $E \equiv [x=y]E'' \overset{M,\alpha}{\longrightarrow} E'$ because $E'' \overset{N,\alpha}{\longrightarrow} E'$. Therefore $M \equiv [x=y]N$. By induction $\vdash E'' = E''+N\alpha.E'$. Hence

$$
\begin{array}{rcll}
\vdash \ [x=y]E'' & = & [x=y](E''+N\alpha.E') & \\
& = & [x=y]E''+[x=y]N\alpha.E' & \textit{by Theorem } 18
\end{array}
$$

  - other cases are similar.
- $E \overset{M_1,\tau}{\longrightarrow} E'' \overset{M_2,\alpha}{\Longrightarrow} E'$ with $M \equiv M_1M_2$. By induction $\vdash E = E+M_1\tau.E''$ and $\vdash E'' = E''+M_2\alpha.E'$. So

$$
\begin{array}{rcll}
\vdash \ E & = & E+M_1\tau.(E''+M_2\alpha.E') & \\
& = & E+M_1(\tau.(E''+M_2\alpha.E')+E''+M_2\alpha.E') & \textit{by T1} \\
& = & E+M_1(\tau.(E''+M_2\alpha.E')+E''+M_2\alpha.E'+M_2\alpha.E') & \textit{by } \mathbf{S}_1 \\
& = & E+M_1(\tau.(E''+M_2\alpha.E')+M_2\alpha.E') & \textit{by T1} \\
& = & E+M_1\tau.(E''+M_2\alpha.E')+M_1M_2\alpha.E' & \textit{by theorem } 18 \\
& = & E+M_1M_2\alpha.E' &
\end{array}
$$

- $E \overset{M_1,\alpha}{\Longrightarrow} E'' \overset{M_2,\tau}{\longrightarrow} E'$ with $M \equiv M_1M_2$. By induction $\vdash E = E+M_1\alpha.E''$ and $\vdash E'' = E''+M_2\tau.E'$. So

$$
\begin{array}{rcll}
\vdash \ E & = & E+M_1\alpha.(E''+M_2\tau.E') & \\
& = & E+M_1(\alpha.(E''+M_2\tau.E')+M_2\alpha.E') & \textit{by Lemma } 22 \\
& = & E+M_1\alpha.(E''+M_2\tau.E')+M_1M_2\alpha.E' & \textit{by theorem } 18 \\
& = & E+M_1M_2\alpha.E' &
\end{array}
$$

□

**Lemma 24 (Absorption)** *If* $E \overset{M,\alpha}{\Longrightarrow}_p E'$ *with* $bn(\alpha)\cap n(M)=\emptyset$ *then* $\vdash E = E+M\epsilon(p).\alpha.E'$

**Proof.** By definition, $E \overset{M,\alpha}{\Longrightarrow}_p E'$ can only be implied by $E \overset{M_1,\epsilon(p)}{\longrightarrow} E'' \overset{M_2,\alpha}{\Longrightarrow} E'$ where $M \equiv M_1M_2$. By Lemma 23, we have $\vdash E'' = E''+M_2\alpha.E'$. And we only to apply induction on why $E \overset{M_1,\epsilon(p)}{\longrightarrow} E''$.

- $E \equiv \epsilon(p).E'' \stackrel{true,\epsilon(p)}{\longrightarrow} \epsilon(0).E'' = E''$. Then $M \equiv M_2$. So,

$$\begin{array}{rcll} \vdash E & = & \epsilon(p).(E'' + M_2\alpha.E') & \\ & = & \epsilon(p).E'' + \epsilon(p).M_2\alpha.E' & \textit{by } \mathbf{TD} \\ & = & \epsilon(p).E'' + M_2\epsilon(p).\alpha.E' & \textit{by Lemma } 19 \\ & = & E + M_2\epsilon(p).\alpha.E' & \end{array}$$

- $E \equiv \epsilon(d).P \stackrel{M_1,\epsilon(p)}{\longrightarrow} E''$ because $P \stackrel{M_1,\epsilon(p-d)}{\longrightarrow} E''$ and $d < p$. Therefore $P \stackrel{M,\alpha}{\longrightarrow}_{p-d} E'$. By induction $\vdash P = P + M\epsilon(p-d).\alpha.E'$, and so

$$\begin{array}{rcll} \vdash E & = & \epsilon(d).(P + M\epsilon(p-d).\alpha.E') & \\ & = & \epsilon(d).P + \epsilon(d).M\epsilon(p-d).\alpha.E' & \textit{by } \mathbf{TD} \\ & = & \epsilon(d).P + M\epsilon(d).\epsilon(p-d).\alpha.E' & \textit{by Lemma } 19 \\ & = & \epsilon(d).P + M\epsilon(p).\alpha.E' & \textit{by } \mathbf{TC}_1 \\ & = & E + M\epsilon(p).\alpha.E' & \end{array}$$

- Other cases can be similarly proved.

$\Box$

**Lemma 25** *For any process expression $E$ containing no '$|$', there is a full normal form $E'$ such that $\vdash E = E'$ and $|E'| \leq |E|$.*

**Proof.** By Lemma 21 we assume $E$ is a *normal form*. Apply induction on the depth of $E$. For the non-trivial case when $E$ is not $\mathbf{0}$, let $E \equiv \sum_{i\in I}\{M_i\epsilon(e_i).\alpha_i.E_i\}$ and consider the term $E' \equiv \sum\{M\epsilon(p).\alpha.F \mid E \stackrel{M,\alpha}{\Longrightarrow}_p F\}$. $E'$ has no depth greater than $E$, and by Lemma 24 $\vdash E = E'$. $\Box$

To prove the completeness of $\mathcal{SW}$, we need a timed version of the so-called Hennessy's theorem [11].

**Lemma 26** *Suppose $D \in MC_{fn(E,F)}$. $E \equiv \sum_{i\in I} M_i\epsilon(e_i).\alpha_i.E_i$ and $F \equiv \sum_{j\in J} N_j\epsilon(f_j).\alpha_j.F_j$. Then $E \approx^D F$ iff there exists $e$ such that*

1. $E \simeq^D F + \epsilon(e).\tau.F_e$ or
2. $F \simeq^D E + \epsilon(e).\tau.E_e$ or
3. $E \simeq^D F$

where

- $E_e = \sum_{i\in I} M_i\epsilon(e_i \dot{-} e).\alpha_i.E_i$
- $F_e = \sum_{j\in J} N_j\epsilon(f_j \dot{-} e).\alpha_j.F_j$

**Proof.** ($\Leftarrow$) is simpler and we just show the proof of ($\Rightarrow$):

Assume $E \approx^D F$. It is easy to see:

1. If $E \stackrel{M,\tau}{\longrightarrow}_e E'$ and $F \stackrel{N,\epsilon(e)}{\longrightarrow} F'$, such that $D \Rightarrow M$, $D \Rightarrow N$ and $F' \approx^D E'$, then $E \simeq^D F + \epsilon(e).\tau.F'$. Now we only need to show $F' \simeq^D F_e$.

   Since $D$ is maximally consistent, therefore for each $N_j$, we have either $D \Rightarrow N_j$ or $D \Rightarrow \neg N_j$. So $N = \{N_j \mid D \Rightarrow N_j\} \cup \{\neg N_j \mid D \Rightarrow \neg N_j\}$, and $F' \equiv \sum_{D\Rightarrow N_j} \epsilon(f_j \dot{-} e).\alpha_j.F_j + \sum_{D\Rightarrow\neg N_j} N_j\epsilon(f_j).\alpha_j.F_j$.

   For those $N_j$ such that $D \Rightarrow N_j$:

   $$N_j\epsilon(f_j \dot{-} e).\alpha_j.F_j \simeq^D \epsilon(f_j \dot{-} e).\alpha_j.F_j$$

   while for those $N_j$ such that $D \Rightarrow \neg N_j$ (i.e. $D \cup N_j \Rightarrow false$):

   $$N_j\epsilon(f_j \dot{-} e).\alpha_j.F_j \simeq^D N_j\epsilon(f_j).\alpha_j.F_j \simeq^D \mathbf{0}$$

   Therefore by Theorem 6,

   $$F_e = \sum_{j\in J} N_j\epsilon(f_j \dot{-} e).\alpha_j.F_j \simeq^D \sum_{D\Rightarrow N_j} \epsilon(f_j \dot{-} e).\alpha_j.F_j + \sum_{D\Rightarrow\neg N_j} N_j\epsilon(f_j).\alpha_j.F_j \equiv F'$$

2. If $F \xrightarrow{M,\tau}_e F'$ and $E \xrightarrow{N,\epsilon(e)} E'$, such that $D \Rightarrow M$, $D \Rightarrow N$ and $E' \approx^D F'$, then $F \simeq^D E + \epsilon(e).\tau.E'$.
3. If neither of the above two cases holds, then $E \simeq^D F$. (Which means both $E$ and $F$ have no transitions of the form $\xrightarrow{M,\tau}_p$ such that $D \Rightarrow M$.)

$\square$

The following lemma will also be useful in proving the completeness result:

**Lemma 27** *If $E \stackrel{M,\hat{\tau}}{\Longrightarrow} E'$, $C \Rightarrow M$ and $\vdash C \triangleright \tau.E' = \tau.F$, then $\vdash C \triangleright \tau.E = \tau.E + \tau.F$.*

**Proof.** If $E \equiv E'$ then $\vdash C \triangleright \tau.E = \tau.F$, hence $\vdash C \triangleright \tau.E = \tau.E + \tau.F$ by $\mathbf{S}_1$.
Otherwise, by Lemma 24 $\vdash E = E + M\tau.E'$. Hence

$$\begin{array}{rcll} \vdash\ C \triangleright \tau.E & = & \tau.(E + M\tau.E') & \\ & = & \tau.(E + M\tau.E') + M\tau.E' & \textit{by Lemma } 22 \\ & = & \tau.E + M\tau.E' & \\ & = & \tau.E + M\tau.F & \\ & = & \tau.E + \tau.F & \textit{by Theorem } 13 \end{array}$$

$\square$

**Theorem 20 (Completeness)** *For all processes $E, F$, if $E \simeq^C F$ then $\vdash C \triangleright E = F$.*

**Proof.** By Lemma 20 and Lemma 25, assume $E, F$ are full normal forms such that $E \equiv \sum_i M_i\epsilon(e_i).\alpha_i.E_i$, $F \equiv \sum_i N_j\epsilon(f_j).\beta_j.F_j$. By $\alpha$ conversion, we may assume $\alpha_i, \beta_j$ contain the same bound name $z \notin fn(E, F, C)$.

We apply induction on $|E| + |F|$.

By theorem 15, we only need to prove $\vdash D \triangleright E = F$ for each $D \in MCE_{fn(E,F,C)}(C)$, and it is sufficient to show: For all $j$, $\vdash D \triangleright E = E + N_j\epsilon(f_j).\beta_j.F_j$.

If $D \not\Rightarrow N_j$, then since $D$ is maximally consistent, then $D \cup N_j \Rightarrow false$. Thus by theorem 13, $\vdash D \triangleright N_j\epsilon(f_j).\beta_j.F_j = \mathbf{0}$, and by axiom $\mathbf{S}_0$ we have $\vdash D \triangleright E = E + N_j\epsilon(f_j).\beta_j.F_j$.

Suppose $D \Rightarrow N_j$ (therefore $D \in MCE_{fn(E,F)}(C \cup N_j)$), apply case analysis on $\beta_j$:

1. $\beta_j \equiv \tau$. Then $F \xrightarrow{N_j,\tau}_{f_j} F_j$. Since $E \simeq^C F$, $E \stackrel{M_i,\tau}{\Longrightarrow}_{f_j} E_i$ for some $i$ s.t. $D \Rightarrow M_i$ and $E_i \approx^D F_j$. Since $E$ is a full normal form, we have
$$\vdash\ E\ =\ E + M_i\epsilon(f_j).\tau.E_i \qquad \text{(i)}$$
We can not invoke the induction hypothesis here because we only have $E_i \approx^D F_j$. But suppose $E_i \equiv \sum_{k\in K} M_k\epsilon(e_k).\alpha_k.E_k$ and $F_j \equiv \sum_{l\in L} M_l\epsilon(f_l).\alpha_k.F_l$, by Lemma 26 we have either $E_i \simeq^D F_j$, or $F_j \simeq^D E_i + \epsilon(e).\tau.E'$, or $E_i \simeq^D F_j + \epsilon(e).\tau.F'$ for some $e$, where
   - $E' \equiv \sum_{k\in K} M_k\epsilon(e_k \dot{-} e).\alpha_k.E_k$
   - $F' \equiv \sum_{l\in L} M_l\epsilon(f_l \dot{-} e).\alpha_k.F_l$

   In the first case ($E_i \simeq^D F_j$), we can apply induction to get $\vdash D \triangleright E_i = F_j$, and by $TAU$, $\vdash D \triangleright \tau.E_i = \tau.F_j$. In the second case, by Lemma 25 there is a full normal form $E''$ with depth no greater than $E$ s.t. $\vdash E_i + \epsilon(e).\tau.E' = E''$. By the soundness of $\vdash$, $E_i + \epsilon(e).\tau.E' \simeq E''$, hence $E'' \simeq^D F_j$. By induction $\vdash D \triangleright E'' = F_j$, so $\vdash D \triangleright E_i + \epsilon(e).\tau.E' = F_j$. Furthermore by $TAU$, $\vdash D \triangleright \tau.(E_i + \epsilon(e).\tau.E') = \tau.F_j$. By T3, we have $\vdash \tau.(E_i + \epsilon(e).\tau.E') = \tau.E_i$, and so $\vdash D \triangleright \tau.E_i = \tau.F_j$. In the third case we can similarly get $\vdash D \triangleright \tau.E_i = \tau.F_j$. Thus, by $DELAY$, $\vdash D \triangleright \epsilon(f_j).\tau.E_i = \epsilon(f_j).\tau.F_j$. Since $D \Rightarrow M_i$, $D \Rightarrow N_j$, by Theorem 13, $\vdash D \triangleright M_i\epsilon(f_j).\tau.E_i = N_j\epsilon(f_j).\tau.F_j$. By equation (i), $\vdash D \triangleright E = E + N_j\epsilon(f_j).\tau.F_j$.

2. $\beta_j \equiv b(z)$, $z \notin fn(E, F, C)$. Then $F \xrightarrow{N_j,b(z)}_{f_j} F_j$. Since $E \simeq^C F$, $E \stackrel{M_i,a_i(z)}{\Longrightarrow}_{f_j} E_i$ for some $i$ s.t. $D \Rightarrow M_i$, $D \Rightarrow a_i = b$, and for each $D' \in MCE_{fn(E,F)\cup\{z\}}(D)$ there is $E_i \stackrel{M',\hat{\tau}}{\Longrightarrow} E'$ s.t. $D' \Rightarrow M'$ and $E' \approx^{D'} F_j$. Since $E$ is a full normal form, we have
$$\vdash\ E\ =\ E + M_i\epsilon(f_j).a_i(z).E_i \qquad \text{(ii)}$$
Using Lemma 26, by the same argument as in the $\tau$ case above, we can show $\vdash D' \triangleright \tau.E' = \tau.F_j$. Note that here $E'$ depends on $D'$. But by Lemma 27, $\vdash D' \triangleright \tau.E_i = \tau.E_i + \tau.F_j$. Now we can use Theorem 15 to obtain $\vdash D \triangleright \tau.E_i = \tau.E_i + \tau.F_j$. By $INPUT$, $\vdash D \triangleright a_i(z).\tau.E_i = a_i(z).(\tau.E_i + \tau.F_j)$. So

$$
\begin{array}{rcll}
\vdash D \triangleright a_i(z).E_i & = & a_i(z).\tau.E_i & \textit{by T3 and T1} \\
& = & a_i(z).(\tau.E_i + \tau.F_j) & \\
& = & a_i(z).(\tau.E_i + \tau.F_j) + a_i(z).F_j & \textit{by T2} \\
& = & a_i(z).E_i + a_i(z).F_j & \\
& = & a_i(z).E_i + b(z).F_j &
\end{array}
$$

By rule *DELAY* and axiom **TD**, $\vdash D \triangleright \epsilon(f_j).a_i(z).E_i = \epsilon(f_j).a_i(z).E_i + \epsilon(f_j).b(z).F_j$. Since $D \Rightarrow M_i$ and $D \Rightarrow N_j$, by Theorem 13, $\vdash D \triangleright M_i\epsilon(f_j).a_i(z).E_i = M_i\epsilon(f_j).a_i(z).E_i + N_j\epsilon(f_j).b(z).F_j$. Hence $\vdash D \triangleright E = E + N_j\epsilon(f_j).b(z).F_j$, by equation (ii).

3. $\beta_j \equiv \bar{b}(z)$. $F \xrightarrow{N_j,\bar{b}(z)}_{f_j} F_j$, so $E \overset{M_i,\bar{a}_i(z)}{\Longrightarrow}_{f_j} E_i \overset{M',\hat{\tau}}{\Longrightarrow} E'$ for some $i$ s.t. $D \Rightarrow M_iM'$, $D \Rightarrow a_i = b$ and $E' \approx^{D'} F_j$, where $D' \equiv D \cup \{z \neq y \mid y \in fn(\bar{a}_i(z).E_i, \bar{b}(z).F_j)\}$. Note that $D' \in MC_{fn(E_i,F_j)}$, so by the same argument as in the $\tau$ case, $\vdash D' \triangleright \tau.E' = \tau.F_j$. By Lemma 27, $\vdash D' \triangleright \tau.E_i = \tau.E_i + \tau.F_j$. By Theorem 19, $\vdash D \triangleright \bar{a}_i(z).\tau.E_i = \bar{a}_i(z).(\tau.E_i + \tau.F_j)$. The rest is the same as the previous case.

4. $\beta_j \equiv \bar{b}y$. Then $F \xrightarrow{N_j,\bar{b}y}_{f_j} F_j$, so $E \overset{M_i,\bar{a}_i x}{\Longrightarrow}_{f_j} E_i$ for some $i$ s.t. $D \Rightarrow M_i$, $D \Rightarrow \bar{a}_i = \bar{b}$, $D \Rightarrow x = y$ and $E_i \approx^D F_j$. By the same argument as in the $\tau$ case, $\vdash D \triangleright \tau.E_i = \tau.F_j$. By *OUTPUT* and T3, T1, $\vdash D \triangleright \bar{a}_i x.E_i = \bar{b}y.F_j$, further by *DELAY*, $\vdash D \triangleright \epsilon(f_j).\bar{a}_i x.E_i = \epsilon(f_j).\bar{b}y.F_j$. By Theorem 13, $\vdash D \triangleright M_i\epsilon(f_j).\bar{a}_i x.E_i = N_j\epsilon(f_j).\bar{b}y.F_j$. Hence by $E$ is a full normal form, we have $\vdash D \triangleright E = E + M_i\epsilon(f_j).\bar{a}_i x.E_i = E + N_j\epsilon(f_j).\bar{b}y.F_j$.

In sum, for all $j$, $\vdash D \triangleright E = E + N_j\epsilon(f_j).\beta_j.F_j$, and by $\mathbf{S}_1$ we have $\vdash D \triangleright E = E + F$. By symmetry we also have $\vdash D \triangleright F = F + E$, and by $\mathbf{S}_2$ and basic axioms for equivalence we have $\vdash D \triangleright E = F$. Therefore $\vdash C \triangleright E = F$ by theorem 15. □

# 4 Conclusion

Based on the basic theory in [3], this paper presents an axiom system of Timed $\pi$ calculus. According to when input names are instantiated, a distinction for bisimulations can be made, namely *Early* and *Late* bisimulations [13]. Thus the bisimulations defined previously in this paper are actually *Late* bisimulations. Following the way in [8] and [10], proof system for early bisimulation can be obtained by simply replacing the inference rule *INPUT* by the following:

$$E-INPUT \ \frac{C \triangleright \sum_{i\in I} \tau.E_i = \sum_{j \in J} \tau.F_j}{C \triangleright \sum_{i\in I} a_i(x).E_i = \sum_{j\in J} b_j(x).F_j} \quad C \Rightarrow a_i = b_j,\ i \in I,\ j \in J,\ x \notin n(C)$$

For the infinite processes containing recursion, the technique used in [9] can also be applied here.

*Symbolic Bisimulation* is very useful in dealing with boolean conditions, and it works well for this Timed $\pi$. The related results of axiomatization on *mobile* systems and on *timed* systems can be found in [12], [18], [17] and [8] *et al*, but the work combining both mobility and real-time within the same framework has never been done so far. The main challenge for our calculus now is how to handle "@$t$" [3, 4] (like in Timed CCS [18], which is used to record the waiting time of the process, for example, $\alpha.P(t) \xrightarrow{\epsilon(d)}\xrightarrow{\alpha} P(d)$) in axiom systems, and we wish to overcome this difficulty using e.g. *region* technique [1] or others.

# 5 Acknowledgement

The author would like to sincerely thank Prof. Huimin Lin for many important suggestions and help, and many thanks are also due to Xia for her careful reading.

# References

[1] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*. 126:183-235, 1994 (preliminary versions appeared in Proc. 17th ICALP, LNCS 443, 1990, and Real Time: Theory in Practice, LNCS 600, 1991).

[2] M.Berger and K.Honda. The two-phase commit protocol in an extended $\pi$-calculus. *Electronic Notes in Theoretical Computer Science*, 39:No.1, Elsevier Science, 2000.

[3] J.Chen. Study of Real-time Value-passing and Real-time mobile systems. PhD Thesis in Institute of Software. Chinese Academy of Sciences, 2003. Extended abstract can be found in: `http://www.univ-orleans.fr/lifo/Members/chen/paper/Ex_PhD.ps`

[4] J.Chen. Some Results on Timed Process Algebra, In: *Proc.International Workshop on Security Analysis of Systems (SASYFT2004)*, Orléans, France. 21-22 June, 2004.

[5] P.Degano, J.Loddo and C.Priami. Mobile Processes with Local Clocks. In: *Proceedings of Workshop on Analysis and Verification of Multiple-Agent Languages*, Stockholm 1996, LNCS 1192:296-319, (M. Dam Ed.), Springer-Verlag.

[6] M.Hennessy and H.Lin. Symbolic Bisimulations. *Theoretical Computer Science,*138:353-389,1995.

[7] M.Hennessy and H.Lin. Proof systems for message-passing process algebra. *Formal Aspects of Computing,* vol.8:379-407.

[8] H.Lin. Symbolic bisimulation and proof systems for the $\pi$-calculus. Technical Report 7/94, University of Sussex, 1994.

[9] H. Lin. Unique fixpoint induction for the $\pi$-calculus. In *CONCUR'95*, LNCS 962. Springer-Verlag, 1995.

[10] H. Lin. A Complete Inference Systems for Weak Bisimulation Equivalences in the $\pi$-Calculus. *Information and Computation.* 180(1), 1-29 (2003). Academic Press. An extended abstract of this paper appeared in *TAPSOFT'95*, Aarhus, Denmark, May 1995. LNCS 915:187-201.

[11] R.Milner. *Communication and Concurrency.* Prentice-Hall, 1989.

[12] R.Milner,J.Parrow and D.Walker. A calculus of mobile processes, part I,II. *Information and Computation*, 100:1-77,1992.

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

[14] R.Milner and D.Sangiorgi. Barbed bisimulation. in: *ICALP'92: Automata, Languages and Programming,* LNCS 623, Springer-Verlag, 1992.

[15] G.M.Reed and A.W.Roscoe, *A Timed Model for Communicating Sequential Processes*, LNCS 226,pp 314-323.Springer-Verlag,1986

[16] D.Sangiorgi. A theory of bisimulation for the $\pi$-calculus. *Acta Informatica*, 33:69-97, 1996.

[17] D.Sangiorgi and D.Walker.*The pi-calculus: a Theory of Mobile Processes.* Cambridge University Press,2001.

[18] Y.Wang. A Calculus of Real Time Systems. PhD thesis, Chalmers University of Technology, Göteborg, Sweden,1991.


# Appendix

## Proof of Theorem 1

The proof is by *transition induction* on $E \xrightarrow{\epsilon(p)} E'$ and $E' \xrightarrow{\alpha} E''$.

Assume $E \xrightarrow{\epsilon(p)} E'$ and $E' \xrightarrow{\alpha} E''$.

1. The transitions are inferred by Idle + Pre: then $E$ must be of the form $\alpha.F (\alpha \neq \tau)$. The result follows from that $Sort_d(\alpha.F) = \{subj(\alpha)\}$ for all $d$.
2. They are inferred by Delay$_1$ + Pre: then $E \equiv \epsilon(e).F$ and $p \geq e$.

   It must be the case that $F \xrightarrow{\epsilon(p \dot{-} e)} E' \xrightarrow{\alpha} E''$. Since $p < d$, we have $p \dot{-} e < d \dot{-} e$. By induction, $subj(\alpha) \in Sort_{d \dot{-} e}(E)$. By definition, $Sort_d(\epsilon(e).F) = Sort_{d \dot{-} e}(F)$. Therefore, $subj(\alpha) \in Sort_d(\epsilon(e).F)$, as required.
3. They are inferred by Sum-Delay and Sum: then $E \equiv F + G$.

   Thus by a shorter inference, $F \xrightarrow{\epsilon(p)} F'$, $G \xrightarrow{\epsilon(p)} G'$. Assume $F' \xrightarrow{\alpha} E''$, then by induction, $subj(\alpha) \in Sort_d(F)$. By definition, $Sort_d(F+G) = Sort_d(F) \cup Sort_d(G)$, thus $subj(\alpha) \in Sort_d(F + G)$. The case that $G' \xrightarrow{\alpha} E''$ is symmetrical.

4. They are inferred by Par-Delay and Par: then $E \equiv F|G$. Similar to the above case.
5. Other cases can be similarly proved by induction.

$\square$

## Proof of Lemma 3

We only give the proof of the 1st property, others are similar.

Assume $E \xrightarrow{\tau} E'$, we show $E \xrightarrow{\epsilon(d)} E''$ for no $d > 0$ and $E''$, by transition induction on $E \xrightarrow{\tau} E'$.

1. $E \xrightarrow{\tau} E'$ is inferred by Pre or Delay$_2$: then $E \equiv \tau.F$ or $E \equiv \epsilon(0).F$.
   By the semantics, $\tau.F$ has no $\epsilon(d)(d > 0)$ transition, the property is true. While by Delay$_2$, $\epsilon(0).F$ has the same behavior as $F$, thus the proof can be obtained by induction.
2. It is inferred by Sum: then $E \equiv F + G$.
   Without loss of generality, suppose $F + G \xrightarrow{\tau} E' = F'$ is inferred from $F \xrightarrow{\tau} F'$. By induction, there are no $d > 0$ and $F''$, $F \xrightarrow{\epsilon(d)} F''$. But by Sum-Delay, to infer $F + G \xrightarrow{\epsilon(d)} E''$, such a $F \xrightarrow{\epsilon(d)} F''$ is indispensable. Thus, there are no $d > 0$ and $E''$ such that $F + G \xrightarrow{\epsilon(d)} E''$. The case that $F + G \xrightarrow{\tau} F'$ is inferred from $G \xrightarrow{\tau} G'$ is symmetrical.
3. It is inferred by Com or Close: then $E \equiv F|G$.
   Assume $F|G \xrightarrow{\tau} E'$. Then by a shorter inference, the following cases are possible.
   i. $F \xrightarrow{\tau} F'$ or $G \xrightarrow{\tau} G'$. Similar to the case of summation (case 2).
   ii.$F \xrightarrow{\eta} F'$ and $G \xrightarrow{\eta_c} G'$, where $subj(\eta) = \overline{subj(\eta_c)}$. By Theorem 1, $\underline{subj(\eta)} \in Sort_d(F)$ and $subj(\eta_c) \in Sort_d(G)$ for all $d > 0$. Then $subj(\eta) \in Sort_d(F) \cap \overline{Sort_d(G)} \neq \emptyset$. Thus $F|G \xrightarrow{\epsilon(d)} F''|G''$ for no $d > 0$, $F''$ and $G''$.
4. Other cases can be proved similarly.

$\square$

## Proof of Lemma 4

Assume $P \dot{\approx} Q$. $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$. If $Sort_d(P) \neq Sort_d(Q)$, then there must exist a name $a$ belonging to only one of $Sort_d(P)$ and $Sort_d(Q)$. Without loss of generality, assume $a \in Sort_d(P)$ and $a \notin Sort_d(Q)$. By $a \in Sort_d(P)$, there must be a $p < d$ and $P''$, $P \xrightarrow{\epsilon(p)}\xrightarrow{\alpha} P''$, $subj(\alpha) = a$. And since $a \notin Sort_d(Q)$, by Theorem 1, there is no $p' < d$ and $R$ such that $Q \xrightarrow{\epsilon(p')}\xrightarrow{\alpha} R$. By $P \dot{\approx} Q$, there must exist $Q''$ such that $Q \xRightarrow{\epsilon(p)}\xRightarrow{\alpha} Q''$, $P'' \dot{\approx} Q''$. Therefore by the definition of double arrows, there must exist a sequence $Q \xrightarrow{\epsilon(p_1)} (\xrightarrow{\tau})^* \xrightarrow{\epsilon(p_2)} (\xrightarrow{\tau})^* \cdots \xrightarrow{\epsilon(p_n)} (\xrightarrow{\tau})^* \xrightarrow{\alpha} Q'''$ such that $p = \sum_{1 \leq i \leq n} p_i$ among which there is at least one $\tau$ transition (otherwise $Q \xrightarrow{\epsilon(p)}\xrightarrow{\alpha} Q'''$). Thus there is a $k \leq n$ such that $Q \xrightarrow{\epsilon(q_k)} R' \xrightarrow{\tau} R''$, where $q_k = \sum_{i \leq k} p_i$. Obviously $q_k < d$ (by $q_k \leq p$ and $p < d$), and $R'$ can't delay anymore by *Maximal Progress* property. While since $q_k < d$, by *Time Continuity* and *Time Determinacy* (see [3, 4]), there is a conflict with the assumption $Q \xrightarrow{\epsilon(d)} Q'$. $\square$

## Proof of Lemma 5

1. Suppose that $P \dot{\approx} Q$, $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$. We show $P' \dot{\approx} Q'$.
   (a) Assume $P' \xrightarrow{\epsilon(c)} P''$. Then $P \xrightarrow{\epsilon(c+d)} P''$. By $P \dot{\approx} Q$, there exists $Q''$, $Q \xRightarrow{\epsilon(c+d)} Q''$ and $P'' \dot{\approx} Q''$. Since $Q \xrightarrow{\epsilon(d)} Q'$, by *Maximal progress*, there can't be any $p < d$ such that $Q \xrightarrow{\epsilon(p)}\xrightarrow{\tau}$. By *Time Continuity* and *Time Determinacy* Lemma, $Q \xrightarrow{\epsilon(d)} Q'$ and $Q' \xRightarrow{\epsilon(c)} Q''$.

(b) Assume $P' \xrightarrow{\tau} P''$. Then By $P \dot{\approx} Q$, $Q \stackrel{\epsilon(d)}{\Longrightarrow}\stackrel{\tau}{\Longrightarrow} Q''$ and $P'' \dot{\approx} Q''$. By $Q \xrightarrow{\epsilon(d)} Q'$, Then by *Maximal Progress* there can't be any $p < d$ such that $Q \xrightarrow{\epsilon(p)}\xrightarrow{\tau}$. Therefore by *Time Determinacy* Lemma and $Q \xrightarrow{\epsilon(d)} Q'$, $Q' \stackrel{\tau}{\Longrightarrow} Q''$ as required.

(c) Assume $P' \xrightarrow{\alpha} P''$, $\alpha \neq \tau$. By $P \xrightarrow{\epsilon(d)} P'$ and $P \dot{\approx} Q$, there is a $Q'''$ such that $Q \stackrel{\epsilon(d)}{\Longrightarrow} Q'''$ and $P' \dot{\approx} Q'''$. Then there exists $Q''$ such that $Q''' \stackrel{\alpha}{\Longrightarrow} Q''$ and $P'' \dot{\approx} Q''$. By *Time Determinacy* Lemma and $Q \xrightarrow{\epsilon(d)} Q'$, $Q \stackrel{\epsilon(d)}{\Longrightarrow} Q'''$ must be inferred by $Q \xrightarrow{\epsilon(d)} Q'$ and $Q'(\xrightarrow{\tau})^* Q'''$. Therefore $Q' \stackrel{\alpha}{\Longrightarrow} Q''$ and $P'' \dot{\approx} Q''$. When $\alpha = x(z)$, $Q'(\xrightarrow{\tau})^* \xrightarrow{x(z)} Q'''$ and for any $y$, $Q'''\{y/z\} \Rightarrow Q''$, $P''\{v/y\} \dot{\approx} Q''$.

By the symmetry of the relation, we have $P' \dot{\approx} Q'$.

2. Suppose that $P \dot{\simeq} Q$ and $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$. we show $P' \dot{\simeq} Q'$.

Assume $P' \xrightarrow{\phi}_p P''$.

(a) $\phi = \epsilon$. By $P \xrightarrow{\epsilon(d)} P'$ and $P' \xrightarrow{\epsilon}_p P''$, by *Time Determinacy* lemma, $P \xrightarrow{\epsilon(p+d)} P''$, i.e. $P \xrightarrow{\epsilon}_{p+d} P''$. By $P \dot{\simeq} Q$, there exists $Q \stackrel{\epsilon}{\Longrightarrow}_{p+d} Q''$ i.e. $Q \xrightarrow{\epsilon(p+d)} Q''$, and $P'' \dot{\approx} Q''$. Since $Q \xrightarrow{\epsilon(d)} Q'$, by Time Continuity and *Time Determinacy* lemma , we have $Q' \xrightarrow{\epsilon(p)} Q''$, i.e. $Q' \stackrel{\epsilon}{\Longrightarrow}_p Q''$.

(b) $\phi = \alpha$. Only need to consider either $bn(\alpha) = \emptyset$, or $bn(\alpha) \nsubseteq n(P', Q')$. First consider $bn(\alpha) = \emptyset$ and $\alpha = \overline{x}(z)$: By $P \xrightarrow{\epsilon(d)} P'$ and $P' \xrightarrow{\alpha}_p P''$, we have $P' \xrightarrow{\alpha}_{p+d} P''$. Then By $P \dot{\simeq} Q$, there exists $Q''$ such that $Q \stackrel{\alpha}{\Longrightarrow}_{p+d} Q''$ i.e. $Q \xrightarrow{\epsilon(p+d)} (\xrightarrow{\tau})^* \xrightarrow{\alpha} (\xrightarrow{\tau})^* Q''$, and $P'' \dot{\approx} Q''$. By *Time Continuity* and *Time Determinacy* lemma , after $d$ units of time, $Q$ will become $Q'$, which is the only possibility. Thus $Q' \xrightarrow{\epsilon(p)} (\xrightarrow{\tau})^* \xrightarrow{\alpha} (\xrightarrow{\tau})^* Q''$, and $P'' \dot{\approx} Q''$. When $\alpha = x(z)$, $Q' \xrightarrow{\epsilon(p)} (\xrightarrow{\tau})^* \xrightarrow{x(z)} Q'''$ and for any $y$, $Q'''\{y/z\} \Rightarrow Q''$, $P''\{v/y\} \dot{\approx} Q''$.

By the symmetry of the relation, we have $P' \dot{\simeq} Q'$.

3. Suppose that $P \dot{\approx}^c Q$ and $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$. We show $P' \dot{\approx}^c Q'$:

Let $R$ be an arbitrary process. By $P \dot{\approx}^c Q$, $P + \epsilon(d).R \dot{\approx} Q + \epsilon(d).R$. By $P \xrightarrow{\epsilon(d)} P'$ and $Q \xrightarrow{\epsilon(d)} Q'$, we have $P + \epsilon(d).R \xrightarrow{\epsilon(d)} P' + R$ and $Q + \epsilon(d).R \xrightarrow{\epsilon(d)} Q' + R$. By $\dot{\approx}^c \subset \dot{\approx}$, The equivalence $P' + R \dot{\approx} Q' + R$ holds directly from the case for $\dot{\approx}$, and we can conclude $P' \dot{\approx}^c Q'$ because $R$ is arbitrary.

$\square$