An Interleaving Model for Real-Time Systems

Liang Chen[1]

Laboratory for the Foundations of Computer Science
Department of Computer Science
University of Edinburgh
Edinburgh EH9 3JZ, UK

September 15, 1991

Abstract

In this paper, we present a general interleaving model for real-time systems which is an extension of Milner's CCS with time. We make no assumption about the underlying nature of time, allowing time to be discrete (such as the integers) or dense (such as the rationals or the reals). We also make no assumption of the Maximal Progress Principle. The time variables allow us to represent the notion of time dependency and thus the language is more expressive than those without time variables. We extend bisimulations to timed processes and study the theories of strong equivalence, weak equivalence and observational congruence. Also we show that strong equivalence is decidable for finite processes, i.e. the processes without recursion. The decidability is independent of the choice of time domain.

1 Introduction

Process algebras, such as CCS [Mil80, Mil89], CSP [Hoa85] and ACP [BK84, BK85], are structured description languages for concurrent systems and have a variety of well developed semantic theories and verification methods. They are considered to be very successful specification languages. However, none of them consider the time aspects of concurrent systems. Instead they usually deal with the quantitative aspects of time of a concurrent system in a qualitative way and ignore explicit time information. There are many systems and applications, such as real-time systems in which interactions with the environment must satisfy some time constraints, for which purely qualitative specification and analysis are inadequate.

[1]The author is supported by TC Scholarship. e-mail: cl@dcs.ed.ac.uk

Recently there have been a number of attempts to introduce time in process algebras. The works include extensions of Milner's CCS with time [CAM90, HR90, MT90, Wan90], extensions of Hoare's CSP with time [RR88] and extensions of Bergstra and Klop's ACP with time [BB91, Gro90, Sif90]. Among these works, there are many features in common. Works such as [HR90, Gro90, Sif90] only deal with discrete time and introduce a special action to represent one unit of time. In general, these languages cannot express the notion of time dependency. For example, consider a system M which can perform action a followed by action b. Action a may happen at any time t. If a happens at time t, then action b must happen before a further t time. Clearly, we cannot define the system M in these languages. Some works, such as [CAM90, Wan90, RR88, HR90], are based on an assumption of the Maximal Progress Principle and therefore are not general.

In this paper, we explore the possibility of incorporating time in a well developed process algebra. We present a more general interleaving model for real-time systems. The work is based on Milner's CCS. We make no assumption about the underlying nature of time, allowing time to be discrete (such as the integers) or dense (such as the rationals or the reals). The time variables allow us to represent the notion of time dependency in the language. Also we make no assumption of the Maximal Progress Principle, but the whole theory is consistent with the principle.

The atomic actions take no time, i.e. they are instantaneous. This assumption is essential if we are to use an interleaving model for real-time systems. We say two actions happen at the same time if one action follows the other instantaneously. To be more realistic, we may insist that for any two actions which are causally related, there should be some non-zero constant delay between them.

In section 2, we present the formal syntax and semantics of the language. In section 3, we define some derived operators and give some examples to demonstrate the utility of the language serving for the specification of real-time systems. In section 4, we define strong bisimulation over timed processes and study the properties of strong equivalence. We show that weakly guarded equations have an unique solution up to strong equivalence. Also, we show that the strong equivalence is decidable for finite processes, i.e. those processes without recursion. The decidability is independent of the choice of time domain. In section 5, we discuss behaviour abstraction for timed processes. We define weak bisimulation and C-bisimulation for timed processes and study the properties of weak equivalence and observation congruence. We show that sequential and guarded equations have an unique solution up to observational congruence.

2 Syntax and Semantics

2.1 The Language of Timed CCS

To define Timed CCS, we presuppose a set Λ, ranged over by a, b, of atomic actions not containing τ and ϵ. Let $\mathrm{Act} = \Lambda \bigcup \{\tau\}$, ranged over by α, β. As in CCS, Λ can be partitioned into Γ, the set of names, and $\overline{\Gamma} = \{\bar{a} \mid a \in \Gamma\}$, the set of co-names, with the provision that $\bar{\bar{a}} = a$. a and $\bar{a}$ are called complementary actions which form the basis of communications in our language, analogous to CCS. We also presuppose an infinite set $Tvar$ of time variables, ranged over by t, s, r. Let the time domain be $(\mathcal{T} \bigcup \{\infty\}, \leq)$, where $\mathcal{T}$ contains an element 0 to represent the starting time and $\leq$ is a linear order over $\mathcal{T}$. Note that we make no assumption about the underlying nature of time, allowing $\mathcal{T}$ to be $\mathbf{N}$ (the set of the integers), $\mathbf{Q}$ (the set of the rationals) or $\mathbf{R}$ (the set of the reals). We introduce ∞ to represent infinite time. Time expressions, ranged over by e, f, g, are defined as follows:

Definition 2.1

1 *For any $u \in \mathcal{T}$ and $t \in Tvar$, u and t are time expressions.*

2 *If e and f are time expressions, then $e+f$, $e-f$, $e\dot{-}f$, $max(e, f)$ and $min(e, f)$ are all time expressions, where $e\dot{-}f$ represents the conditional expression if $e < f$ then 0 else $(e - f)$.*

By convention, for any time expression e, we have $e \leq \infty$, $\infty \circ e = \infty$, $max(e, \infty) = \infty$ and $\min(e, \infty) = e$, where $\circ$ is $+$, $-$, or $\dot{-}$.

We also presuppose an infinite set $Pvar$ of process variables, ranged over by X, Y. The process expressions of Timed CCS, ranged over by E, F, are defined by the following BNF expressions, where $S : \Lambda \longrightarrow \Lambda$ is a relabelling function which satisfies $\overline{S(a)} = S(\bar{a})$. Also, by convention, $S(\tau) = \tau$.

$$E ::= X \mid nil \mid \alpha(t)_e^{e'}.E \mid E + F \mid E \mid F \mid E \backslash a \mid E[S] \mid \mu X.E$$

where e' is a time expression or $e' \equiv \infty$ ($\equiv$ is the syntactical equality).

The informal interpretation of the language is as follows:

1. X represents the process bound to it.

2. nil represents the process which does no action, but idles any time.

3. $\alpha(t)_e^{e'}.E$ represents the process which performs action α between time e and e', after α happens, it evolves to process $E\{u/t\}$, where u is the time at which α happens and $E\{u/t\}$ is the result of substituting all free occurrences of t in E.

4. $E + F$ represents deterministic and nondeterministic choice between processes E and F. The choice is made at the time of the first action of E or F, or at times when one process can idle but another cannot. So the choice is deterministic with time proceeding.

5. $E|F$ represents the parallel composition of processes E and F. Each of them may perform actions independently, or they synchronise on complementary actions which represent communications between them. Parallel composition must be synchronous with time proceeding.

6. $E\backslash a$ represents the process which behaves as E, but action a and its complementary action $\bar{a}$ are forbidden.

7. $E[S]$ represents the process resulting from E by relabelling its actions using S.

8. $\mu X.E$ represents the process defined by equation $X = E$.

There are two kinds of variables: time variables and process variables. The operator $\alpha(t)_e^{e'}.$ in $\alpha(t)_e^{e'}.E$ binds all free occurrences of time variable t in E. The operator μX in $\mu X.E$ binds all free occurrences of process variable X in E. These give, in the usual sense, the notions of free and bound time variables (process variables). We use $ftv(E)$ ($fpv(E)$) to represent the set of all free time (process) variables occurring in E. We identify those process expressions which are the same up to change of bound time and process variables. A process expression E is closed with respect to time variables (process variables) if there is no free occurrences of time variables (process variables) in E. An agent is a process expression which is closed with respect to both time and process variables. Let $\mathcal{P}$ represent the set of agents, ranged over by P, Q, R.

2.2 The Operational Semantics of Timed CCS

In order to define a semantics for Timed CCS, we use the labelled transition system

$$(\mathcal{P}, \{\xrightarrow{\alpha}_u \bigcup \longrightarrow_u \mid \alpha \in \mathrm{Act} \bigwedge u \in \mathcal{T}\})$$

The understanding of transition $P\xrightarrow{\alpha}_u P'$ is that agent P performs action α at time u and then evolves to P'. The transition $P\longrightarrow_u P'$ means that agent P idles up to time u without any action and then evolves to P'. $P\{u/t\}$ represents the result of substituting all free occurrences of t in P by u.

To define the transition rules, we need to formalise a notion of maximum delay times of processes before any actions.

Definition 2.2 *For any process E, its maximum delay time $|E|_\mathcal{T}$ is defined inductively as follows*

$|X|_{\mathcal{T}} = 0$

$|nil|_{\mathcal{T}} = \infty$

$|\alpha(t)_e^{e'}.E|_{\mathcal{T}} = max(0, e')$

$|E + F|_{\mathcal{T}} = max(|E|_{\mathcal{T}}, |F|_{\mathcal{T}})$

$|E \mid F|_{\mathcal{T}} = min(|E|_{\mathcal{T}}, |F|_{\mathcal{T}})$

$|E\backslash a|_{\mathcal{T}} = |E|_{\mathcal{T}}$

$|E[S]|_{\mathcal{T}} = |E|_{\mathcal{T}}$

$|\mu X.E|_{\mathcal{T}} = |E|_{\mathcal{T}}$

For example, $|\mu X.\alpha(t)_1^8.X + X|_{\mathcal{T}} = 8$ and $|\mu X.\alpha(t)_1^8.X \mid X|_{\mathcal{T}} = 0$

Table 1 presents transition rules of the language. The rules are presented in natural deduction style which are read as follows: if the transition or transitions above the inference line can be inferred, then we can infer the transition below the line. The operational semantics of the language is then given by the least transition relations $\xrightarrow{\alpha}_u$ and $\longrightarrow_u$, where $\alpha \in$ Act and $u \in \mathcal{T}$, defined in Table 1.

Notation $\delta \stackrel{\text{def}}{=} (a(t)_0^0\, nil)\backslash a$

Informally, δ represents a dead process which neither performs any action, nor idles.

Proposition 2.1 (1) *For any agent P, $P \longrightarrow_0 P$.*

(2) *For any agent P and $u \in \mathcal{T}$, $P \longrightarrow_u P'$ for some P' if and only if $u \leq |P|_{\mathcal{T}}$.*

By induction on P, the result follows from the definition of transition rules.

3 Derived Operators and Examples

3.1 Derived Operators

We may define some useful operators in the language. The first one is time prefix $(e)E$ for any time expression e and weakly guarded process E. We say a process E is weakly guarded if every occurrence of process variables is guarded by some action. For example, if X is weakly guarded in E, then X can only occur in some subterm of form $\alpha(t)_e^{e'}.F$.

Definition 3.1 *For any time expression e and weakly guarded process E, we define $(e)E$ inductively on E as follows:*

$$\frac{}{P \longrightarrow_0 P}$$

$$\frac{}{\alpha(t)^{e'}_{e}.\, P \xrightarrow{\alpha}_u P\{u/t\}} \qquad max(0,e) \leq u \leq e'$$

$$\frac{}{\alpha(t)^{e'}_{e}.\, P \longrightarrow_u \alpha(t)^{e'-u}_{e-u}.\, (P\{u+t/t\})} \qquad 0 \leq u \leq e'$$

$$\frac{}{nil \longrightarrow_u nil} \qquad u \geq 0 \qquad\qquad \frac{P \longrightarrow_u P' \qquad Q \longrightarrow_u Q'}{P + Q \longrightarrow_u P' + Q'}$$

$$\frac{P \longrightarrow_u P'}{P + Q \longrightarrow_u P'} \qquad |Q|_{\mathcal{T}} < u \qquad\qquad \frac{Q \longrightarrow_u Q'}{P + Q \longrightarrow_u Q'} \qquad |P|_{\mathcal{T}} < u$$

$$\frac{P \xrightarrow{\alpha}_u P'}{P + Q \xrightarrow{\alpha}_u P'} \qquad\qquad \frac{Q \xrightarrow{\alpha}_u Q'}{P + Q \xrightarrow{\alpha}_u Q'}$$

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

$$\frac{P \xrightarrow{\alpha}_u P' \qquad Q \longrightarrow_u Q'}{P \,|\, Q \xrightarrow{\alpha}_u P' \,|\, Q'} \qquad\qquad \frac{P \longrightarrow_u P' \qquad Q \xrightarrow{\alpha}_u Q'}{P \,|\, Q \xrightarrow{\alpha}_u P' \,|\, Q'}$$

$$\frac{P \longrightarrow_u P'}{P \backslash a \longrightarrow_u P' \backslash a} \qquad\qquad \frac{P \xrightarrow{\alpha}_u P'}{P \backslash a \xrightarrow{\alpha}_u P' \backslash a} \qquad \alpha \neq a \bigwedge \alpha \neq \bar{a}$$

$$\frac{P \longrightarrow_u P'}{P[S] \longrightarrow_u P'[S]} \qquad\qquad \frac{P \xrightarrow{\alpha}_u P'}{P[S] \xrightarrow{S(\alpha)}_u P'[S]}$$

$$\frac{E\{\mu X.E/X\} \longrightarrow_u P'}{\mu X.E \longrightarrow_u P} \qquad\qquad \frac{E\{\mu X.E/X\} \xrightarrow{\alpha}_u P'}{\mu X.E \xrightarrow{\alpha}_u P}$$

Table 1: Operational Semantics

$(e)nil = nil$

$(e)(a(t)_f^{f'}. E) = a(t)_{f+e}^{f'+e}. (E\{t - e/t\}) \qquad t \notin fv(e)$

$(e)(E + F) = (e)E + (e)F$

$(e)(E \mid F) = (e)E \mid (e)F$

$(e)(E\backslash a) = ((e)E)\backslash a$

$(e)(E[S]) = ((e)E)[S]$

$(e)(\mu X.E) = (e)(E\{\mu X.E/X\})$

Informally, $(e)P$ represents a process which will idle time e and then behave as process P.

Note that ∞ is not a time expression, for convenience, we use $(\infty)\delta$ in place of $(a(t)_0^{\infty}. nil)\backslash a$ in the sequel.

We can also define the time shift $e \gg E$ of [BB91], where e is a time expression and E is a weakly guarded process.

Definition 3.2 *For any time expression e and weakly guarded process E, the time shift $e \gg E$ is defined inductively over E as follows:*

$e \gg nil = nil$

$e \gg (a(t)_f^{f'}. E) = a(t)_{f-e}^{f'-e}. (P\{t + e/t\}) \qquad t \notin fv(e)$

$e \gg (E + F) = e \gg E + e \gg F$

$e \gg (E \mid F) = e \gg E \mid e \gg F$

$e \gg (E\backslash a) = (e \gg E)\backslash a$

$e \gg (E[S]) = (e \gg E)[S]$

$e \gg (\mu X.E) = e \gg (E\{\mu X.E/X\})$

Informally, for any process P, $e \gg P$ represents the result of P after idling time e without any action. So we have following property:

Proposition 3.1 *For any agent P, $P \longrightarrow_u u \gg P$, where $0 \leq u \leq |P|_{\mathcal{T}}$.*

In Timed CCS, we represent explicitly the notion of timeout. For example, in $a(t)_2^{10}. nil + (15)\delta$, the timeout of action a is 10 seconds (if we measure time in seconds) and after 10 seconds action a is blocked. Also, we can define another important notion, duration control, of real-time systems in Timed CCS. Consider a process E which may behave as process E' before time e and from time e it must behave as process E''. We use $E' \,\mathcal{U}_e E''$ to represent such a process. To simplify the definition of $E'\mathcal{U}_e E''$, we assume that for every relabelling function S, there is a relabelling function S^{-1}, the inverse function of S, such that $SS^{-1} = S^{-1}S = id$, where id is the identity function.

Definition 3.3 $nil\mathcal{U}_e F = (e)F$

$$(\alpha(t)_f^{f'}.\, E)\mathcal{U}_e F = \alpha(t)_f^{min(e,f')}.\,(E\,\mathcal{U}_{e\dot{-}t} F) + (e)F$$

$$(E + E')\mathcal{U}_e F = (E\,\mathcal{U}_e F) + (E'\,\mathcal{U}_e F)$$

$$(E \mid E')\mathcal{U}_e F = (E\,\mathcal{U}_e F) \mid (E'\,\mathcal{U}_e\, nil)$$

$$(E\backslash a)\mathcal{U}_e F = ((E\,\mathcal{U}_e(F[\lambda/a]))\backslash a)[a/\lambda] \qquad \lambda \textit{ is a fresh action for } E,\ F.$$

$$(E[S])\mathcal{U}_e F = (E\,\mathcal{U}_e(F[S^{-1}]))[S]$$

$$(\mu X.E)\mathcal{U}_e F = (E\{\mu x.E/X\})\mathcal{U}_e F$$

3.2 Examples

In this section, we give some examples to show the utility of the language serving for the specification of real-time systems.

Example 1 (Timeout) Consider a timed implementation of the Alternating Bit Protocol. The protocol consists of four parts: *Sender*, *Receiver* and two unreliable transition lines, *Ack* and *Trans*. The unreliable transition lines may lose (but not corrupt) the messages. Messages and acknowledgements are sent tagged with bits 0 and 1 alternately. In the sequel, we assume $\hat{b} = 1 - b$, where $b = 0, 1$.

Sender, after accepting a message from the environment, sends it with an attached bit b, where $b = 0$ or 1, along the line *Trans*. Then it waits for an appropriate acknowledgement within time D, the time *Sender* will wait after sending a message before assuming the message had been lost and retransmitting it. If it receives a correct acknowledgement within time D, *Sender* becomes ready to accept the next message from the environment and then transmits it with bit $\hat{b}$. If it get a wrong acknowledgement or does not receive any acknowledgement within time D, *Sender* retransmits the message at time $D + \Delta$.

$$Sender = Sender_0$$

$$Sender_b = accept(t)_0^{\infty}.\,(\Delta)\,\overline{send}_b(s)_{0'}^{0}\,(\Delta)\,Sending_b$$

$$Sending_b = ack_{\hat{b}}(t)_0^{\mathrm{D}}.\,(\Delta)\,\overline{send}_b(s)_{0'}^{0}\,(\Delta)\,Sending_b\, +$$
$$ack_b(t)_0^{\mathrm{D}}.\,(\Delta)\,Sender_{\hat{b}} + (\mathrm{D} + \Delta)\,\overline{send}_b(s)_{0'}^{0}\,(\Delta)\,Sending_b$$

Receiver, which we assume starts in a state waiting for a message tagged with 0, is defined as:

$$Receiver = tran_0(t)_0^{\infty}.\,(\Delta)\,Deliver_0$$

$Deliver_b = \overline{deliver}(t)^0_{0'}\,(\Delta)\,\overline{reply_b}(s)^0_{0'}\,(\Delta)\,Reply_b$

$Reply_b = tran_b(t)^{\mathrm{D}}_{0}.\,(\Delta)\,\overline{reply_b}(s)^0_{0'}\,(\Delta)\,Reply_b +$

$$tran_{\hat{b}}(t)^{\mathrm{D}}_{0}.\,(\Delta)\,Deliver_{\hat{b}} + (D+\Delta)\,\overline{reply_b}(s)^0_{0'}\,(\Delta)\,Reply_b$$

Two unreliable transition lines *Trans* and *Ack* behave as buffers of capacity one, but they may lose the messages. We suppose the transition times for both transition lines are D'.

$Trans = send_0(t)^{\infty}_{0}.\,(\Delta)\,(\tau(s)^{\mathrm{D}'}_{0}.\,(\Delta)\,Trans + (\mathrm{D}')\,\overline{tran_0}(s)^0_{0'}\,(\Delta)\,Trans)$

$$+\, send_1(t)^{\infty}_{0}.\,(\Delta)\,(\tau(s)^{\mathrm{D}'}_{0}.\,(\Delta)\,Trans + (\mathrm{D}')\,\overline{tran_1}(s)^0_{0'}\,(\Delta)\,Trans)$$

$Ack = reply_0(t)^{\infty}_{0}.\,(\Delta)\,(\tau(s)^{\mathrm{D}'}_{0}.\,(\Delta)\,Ack + (\mathrm{D}')\,\overline{ack_0}(s)^0_{0'}\,(\Delta)\,Ack)$

$$+\, reply_1(t)^{\infty}_{0}.\,(\Delta)\,(\tau(s)^{\mathrm{D}'}_{0}.\,(\Delta)\,Ack + (\mathrm{D}')\,\overline{ack_1}(s)^0_{0'}\,(\Delta)\,Ack)$$

Now we may define the protocol as

$$Protocal = (Sender \mid Receiver \mid Trans \mid Ack)\backslash L$$

where $L = \{send_0,\ send_1,\ tran_0,\ tran_1,\ ack_0,\ ack_1,\ reply_0,\ reply_1\}$.

Since the least time for *Sender* to receive a correct acknowledgement after sending a message to the transition line is $2\mathrm{D}' + 4\Delta$, where $\mathrm{D}'(>>\ \Delta)$ is the transition delay of transition lines, we should take the timeout $\mathrm{D} > 2\mathrm{D}' + 4\Delta$.

Example 2 (Duration Control) Consider a login procedure. After a terminal is connected to the system, the system waits for the user's valid login response. If the user enters a valid response r within 2 minutes, then the login procedure pass the control to a process Q which formalizes the user's task. If the user's login response is invalid, the system goes back and still waits for user's valid response. After waiting for 2 minutes without any valid response, the system rejects the current login request and the connection between the terminal and the system is broken.

$LP = connect(t)^{\infty}_{0}.\,(\Delta)\,(P \mid (b(t)^{\infty}_{0}.\,Q + a(t)^{\infty}_{0}.\,nil)\backslash\{a,b\}$

$P = L\ \mathcal{U}_2\,\bar{a}(s)^0_{0'}\,LP$

$L = r(t)^{\infty}_{0}.\,(\Delta)\,\bar{b}(t)^0_{0'}\,nil{+}ir(t)^{\infty}_{0}.\,(\Delta)L$

4 Strong Equivalence and its Properties

4.1 Strong Bisimulation

We do not wish to distinguish agents which, in some sense, have the same behaviour. The notion of bisimulation between agents captures the idea of having the same behaviour. We say two agents are not equivalent if a distinction can be detected by an experimenter who interacts with each of them.

Definition 4.1 *A binary relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a strong bisimulation if $(P, Q) \in \mathcal{R}$ implies that for any $\alpha \in \mathrm{Act}$ and $u \in \mathcal{T}$*

(1) *if $P \xrightarrow{\alpha}_u P'$, then there is some Q' such that $Q \xrightarrow{\alpha}_u Q'$ and $(P', Q') \in \mathcal{R}$.*

(2) *if $P \longrightarrow_u P'$, then there is some Q' such that $Q \longrightarrow_u Q'$ and $(P', Q') \in \mathcal{R}$.*

(3) *if $Q \xrightarrow{\alpha}_u Q'$, then there is some P' such that $P \xrightarrow{\alpha}_u P'$ and $(P', Q') \in \mathcal{R}$.*

(4) *if $Q \longrightarrow_u Q'$, then there is some P' such that $P \longrightarrow_u P'$ and $(P', Q') \in \mathcal{R}$.*

We say two agents P and Q are strongly bisimilar, denoted by $P \sim Q$, if there is a strong bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

Definition 4.2 $\sim = \bigcup \{\mathcal{R} \mid \mathcal{R}$ is a strong bisimulation$\}$.

Proposition 4.1 *$\sim$ is the largest strong bisimulation. Moreover, $\sim$ is an equivalence relation.*

The proof follows from the facts that the union, composition and converse of strong bisimulations are still strong bisimulations and also the identity over agents is a strong bisimulation.

Lemma 4.2 *For any agents P and Q, $P \sim Q$ implies $|P|_{\mathcal{T}} = |Q|_{\mathcal{T}}$.*

Definition 4.3 *A binary relation $\mathcal{R}$ over $\mathcal{P}$ is a strong $\mathcal{T}$-bisimulation if $(P, Q) \in \mathcal{R}$ implies that for any $\alpha \in \mathrm{Act}$ and $u \in \mathcal{T}$*

(1) *if $P \xrightarrow{a}_u P'$, then there is a Q' such that $Q \xrightarrow{a}_u Q'$ and $(P', Q') \in \mathcal{R}$.*

(2) *if $Q \xrightarrow{a}_u Q'$, then there is a P' such that $P \xrightarrow{a}_u P'$ and $(P', Q') \in \mathcal{R}$.*

(3) $|P|_{\mathcal{T}} = |Q|_{\mathcal{T}}$.

Similar to that of strong bisimulation, we say two agents P and Q are strong $\mathcal{T}$-bisimilar, denoted by $P \sim_{\mathcal{T}} Q$, if there is a strong $\mathcal{T}$-bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

Definition 4.4 $\sim_{\mathcal{T}} = \bigcup \{\mathcal{R} \mid \mathcal{R}$ is a strong $\mathcal{T}-$bisimulation$\}$

Proposition 4.3 *$\sim_{\mathcal{T}}$ is the largest strong $\mathcal{T}$-bisimulation. Moreover, $\sim_{\mathcal{T}}$ is an equivalence relation.*

Lemma 4.4 (1) *if $P \longrightarrow_u P'$ and $P \longrightarrow_u P''$, then $P' \equiv P''$*

(2) *if $P \longrightarrow_u P'$ and $P' \longrightarrow_v P''$, then $P \longrightarrow_{u+v} P''$*

(3) *if $P \longrightarrow_u P'$ and $P' \xrightarrow{\alpha}_v P''$, then $P \xrightarrow{\alpha}_{u+v} P''$*

(4) *If* $P \xrightarrow{\alpha}_u P'$, *then* $P \longrightarrow_u P''$ *and* $P'' \xrightarrow{\alpha}_0 P'$ *for some* P''.

The proof follows from the definition of transition rules. (1) shows the property of time determinacy. (2) and (3) show the property of time continuity and (4) says that atomic actions are instantaneous.

Proposition 4.5 *For any agents* P *and* Q, $P \sim Q$ *if and only if* $P \sim_{\mathcal{T}} Q$.

Proof ($\Rightarrow$) It is easy to show that $\sim$ is a strong $\mathcal{T}$-bisimulation.
($\Leftarrow$) We prove $P \not\sim Q$ implies $P \not\sim_{\mathcal{T}} Q$ by induction on the maximum depth of P and Q, where the depth of P is defined to be the maximum number of nested actions in P. Suppose $P \not\sim Q$, then it must be one of the following cases:

Case 1 $P \xrightarrow{\alpha}_u P'$ and for any Q', if $Q \xrightarrow{\alpha}_u Q'$ then $P' \not\sim Q'$. By induction, if $Q \xrightarrow{\alpha}_u Q'$ then $P' \not\sim_{\mathcal{T}} Q'$. So $P \not\sim_{\mathcal{T}} Q$.

Case 2 $Q \xrightarrow{\alpha}_u Q'$ and for any P', if $P \xrightarrow{\alpha}_u P'$ then $P' \not\sim Q'$. Similar to case (1).

Case 3 $|P|_{\mathcal{T}} \neq |Q|_{\mathcal{T}}$, clearly, $P \not\sim_{\mathcal{T}} Q$.

Case 4 All above cases do not hold for P and Q, this is impossible. Otherwise, let $\mathcal{R} = \{(P', Q') \,|\, (P' = P \bigwedge Q' = Q) \bigvee \exists u \leq |P|_{\mathcal{T}}.(P \longrightarrow_u P' \bigwedge Q \longrightarrow_u Q')\} \bigcup \sim$. It is easy to show that $\mathcal{R}$ is a strong bisimulation. So $P \sim Q$, contradiction. $\Box$

The reason that strong bisimulation and strong $\mathcal{T}$-bisimulation coincide is because time is deterministic, i.e. the transition $\longrightarrow_u$ is deterministic. Since the maximal idle time of an agent is syntacticly defined, it will be easier to show a relation of strong $\mathcal{T}$-bisimulation than that of strong bisimulation.

4.2 Some Properties of Strong Equivalence

In this section, we show some equational laws for strong equivalence. By using these laws (note that every law can be regarded as two direction rewriting rule), every process can be rewritten into a special form which we call the normal form.

Proposition 4.6 (Laws for Prefix)

1 $a(t)_e^{e'}.P \ \sim\ a(s)_e^{e'}.(P\{s/t\})$ $\qquad$ *s is free for t in P*

2 $a(t)_e^{e'}.P \ \sim\ (max(0, e'))\delta$ $\qquad$ $e > e'$

3 $a(t)_e^{e'}.P \ \sim\ a(t)_e^{e'}.P + (max(0, e'))\delta$

4 $a(t)_e^{f}.P \ + a(t)_f^{e'}.P \ \sim\ a(t)_e^{e'}.P$ $\qquad$ $e \leq f \leq e'$

All these laws may be proved by showing appropriate strong bisimulations.

(1) shows that α-conversion preserves strong equivalence. From (2), we see that timing may also block actions. For example, agent $a(t)_{10}^{5}\,nil$ cannot perform action a, but idles time 5 before it dies.

Proposition 4.7 (Laws for Summation)

1 $P + \delta \ \sim \ P$

2 $P + P \ \sim \ P$

3 $P + Q \ \sim \ Q + P$

4 $P + (Q + R) \ \sim \ (P + Q) + R$

Similarly, all these laws may be proved by exhibiting appropriate strong $\mathcal{T}$-bisimulations. These laws together say that $(\mathcal{P}, +, \delta)$ is a monoid.

Proposition 4.8 (Laws for Restriction)

1 $(\infty)\delta \ \sim \ nil$

2 $nil\backslash a \ \sim \ nil$

3 $(\alpha(t)_e^{e'}.P)\backslash a \ \sim \ \alpha(t)_e^{e'}.(P\backslash a) \qquad \alpha \neq a \bigwedge \alpha \neq \bar{a}$

4 $(\alpha(t)_e^{e'}.P)\backslash a \ \sim \ (max(0, e'))\delta \qquad \alpha = a \bigvee \alpha = \bar{a}$

5 $(P + Q)\backslash a \ \sim P\backslash a + Q\backslash a$

Since the restriction operator is distributive over choice, for any agent P which does not contain the parallel operator, we can get an agent P' by using these laws (regarding every law as a two direction rewriting rule) such that $P \ \sim \ P'$, where P' does not contain the restriction operator (except δ).

Proposition 4.9 (Laws for Relabelling)

1 $((e)\delta)[S] \ \sim \ (e)\delta$

2 $(\alpha(t)_e^{e'}.P)[S] \ \sim \ (S(\alpha))(t)_e^{e'}.(P[S])$

3 $(P + Q)[S] \ \sim \ P[S] + Q[S]$

Similarly, for any agent P which does not contain the parallel operator, we can find an agent P' by using these laws such that $P \ \sim \ P'$, where P' does not contain the relabelling operator.

Proposition 4.10 (Laws for Recursion)

1 $P \sim E \ \mathit{if} \ P \stackrel{\text{def}}{=} E$.

2 $\mu X.E \sim E\{\mu X.E/X\}$

Up to now, we have only defined strong equivalence over agents. However, we may also extend strong equivalence to any processes in the standard way.

Definition 4.5 *For any processes E and F which are closed with respect to process variables and contain at most free time variables $\vec{t} = (t_1, \ldots, t_n)$, we say $E \sim F$ if for any $\vec{u} \in \mathcal{T}^n$ we have $E\{\vec{u}/\vec{t}\} \sim F\{\vec{u}/\vec{t}\}$.*

Definition 4.6 *For any processes E and F which contain at most free process variables $\vec{X} = (X_1, \ldots, X_n)$, we say $E \sim F$ if for any agents $\vec{P} = (P_1, \ldots, P_n)$, we have $E\{\vec{P}/\vec{X}\} \sim F\{\vec{P}/\vec{X}\}$.*

Clearly, all laws discussed before still hold for processes. Moreover, we have the following expansion law. To simplify the presentation, we need a notion of normal form.

Definition 4.7 *A process $P \equiv \sum_{i \in I} a_i(t_i)_{e_i}^{e_i'}.\, P_i + (e)\delta$ is in normal form if for any $i \in I$, $e_i' \leq e$ and P_i is also in normal form.*

Proposition 4.11 (Expansion Law)

Given

$$E \equiv \sum_{i \in I} \alpha_i(t_i)_{e_i}^{e_i'}.\, E_i + (e)\delta$$

and

$$F \equiv \sum_{j \in J} \beta_j(s_j)_{f_j}^{f_j'}.\, F_j + (f)\delta$$

in normal forms, then

$$
\begin{aligned}
E \,|\, F \;\sim\; & \\
& \sum_{i \in I} \alpha_i(r_i)_{e_i}^{min(e_i, f)}.\, (E_i\{r_i/t_i\} \,|\, r_i \gg F) \\
+ & \\
& \sum_{j \in J} \beta_j(r_j')_{f_j}^{min(f_j', e)}.\, (r_j' \gg E \,|\, F_j\{r_j'/s_j\}) \\
+ & \\
& \sum_{\substack{\alpha_i = \bar{\beta}_j \\ i \in I \\ j \in J}} \tau(r_{ij})_{max(e_i, f_j)}^{min(e_i', f_j')}.\, (E_i\{r_{ij}/t_i\} \,|\, F_j\{r_{ij}/s_j\}) \\
+ & \\
& (min(e, f))\delta
\end{aligned}
$$

where for any $i \in I$, $j \in J$, r_i, r_j' and r_{ij} are some fresh time variables.

Proof We only need to consider the case in which E and F are agents and show that $\mathcal{R}$, which is the least relation satisfying

(1) $\sim_{\mathcal{T}} \subseteq \mathcal{R}$.

(2) For any normal forms $P \equiv \sum_{i\in I} \alpha_i(t_i)^{e'_i}_{e_i}.\, P_i + (u)\delta$ and $Q \equiv \sum_{j\in J} \beta_j(s_j)^{f'_j}_{f_j}.\, Q_j + (v)\delta$, let G be the right hand side of expansion law for P and Q, then $(P \,|\, Q,\ G) \in \mathcal{R}$

is a strong $\mathcal{T}$-bisimulation.

Case 1 Suppose $P \,|\, Q \xrightarrow{\alpha}_u R$

Case 1.1 $P \xrightarrow{\alpha}_u P'$, $Q \longrightarrow_u Q'$ and $R \equiv P' \,|\, Q'$. Then for some $i \in I$ and $j \in J$, $\alpha = \alpha_i$, $max(0, e_i) \leq u \leq e'_i$, $P' \equiv P_i\{u/t_i\}$ and $Q' \equiv u \gg Q$. Clearly, $G \xrightarrow{\alpha}_u (P_i\{r_i/t_i\} \,|\, r_i \gg Q)\{u/r_i\} \sim_{\mathcal{T}} R$.

Case 1.2 $P \longrightarrow_u P'$, $Q \xrightarrow{\alpha}_u Q'$ and $R \equiv P' \,|\, Q'$. Similar to case 1.1.

Case 1.3 $\alpha = \tau$, $P \xrightarrow{a}_u P'$, $Q \xrightarrow{\bar{a}}_u Q'$ and $R \equiv P' \,|\, Q'$ for some a. Then $\alpha_i = a$, $\beta_j = \bar{a}$, $max(0, e_i, f_j) \leq u \leq min(e'_i, f'_j)$, $P' \equiv P_i\{u/t_i\}$ and $Q' \equiv Q_j\{u/s_j\}$ for some $i \in I$, $j \in J$. Clearly, $G \xrightarrow{\tau}_u (P_i\{r_{ij}/t_i\} \,|\, Q_j\{r_{ij}/s_j\})\{u/r_{ij}\} \sim_{\mathcal{T}} R$.

Case 2 Suppose $G \xrightarrow{\alpha}_u R$. Similar to case 1.

Case 3 $|P \,|\, Q|_{\mathcal{T}} = min(u, v) = |G|_{\mathcal{T}}$. □

Strong equivalence is a congruence relation, asshown by the following two propositions.

Proposition 4.12 *For any processes E and F, $E \sim F$ implies*

1 $\alpha(t)^{e'}_{e}.\, E \sim \alpha(t)^{e'}_{e}.\, F$

2 $E + G \sim F + G$

3 $E \,|\, G \sim F \,|\, G$

4 $E \backslash a \sim F \backslash a$

5 $E[S] \sim F[S]$

All of these properties can be shown by exhibiting appropriate strong bisimulations. In the sequel, $\vec{E} \sim \vec{F}$ represents componentwise strong equivalences between $\vec{E}$ and $\vec{F}$.

Proposition 4.13 *For any processes $\vec{E}$ and $\vec{F}$ which contain at most free process variables $\vec{X}$, $\vec{E} \sim \vec{F}$ implies $\mu\vec{X}.\vec{E} \sim \mu\vec{X}.\vec{F}$.*

Proof We only deal with a single recursive equation. Let E and F be closed with respect to time variables and contain at most free process variable X. Suppose $E \sim F$. To show $\mu X.E \sim \mu X.F$, it is enough to show (by taking $G \equiv X$) that

$$\{(G\{\mu X.E/X\},\ G\{\mu X.F/X\}) \,|\, \mathit{fpv}(G) \subseteq \{X\} \text{ and } \mathit{ftv}(G) = \emptyset\}$$

is a strong bisimulation up to $\sim$. To do that, it is enough to prove (by the symmetric arguments) that

1 If $G\{\mu X.E/X\} \longrightarrow_u P'$, then for some Q' and Q'', $G\{\mu X.F/X\} \longrightarrow_u Q''$ with $Q'' \sim Q'$ and $(P', Q') \in \mathcal{S}$.

2 If $G\{\mu X.E/X\} \xrightarrow{\alpha}_u P'$, then for some Q' and Q'', $G\{\mu X.F/X\} \xrightarrow{\alpha}_u Q''$ with $Q'' \sim Q'$ and $(P', Q') \in \mathcal{S}$.

We prove the claim by transition induction on the depth of inferences.

1. Suppose $G\{\mu X.E/X\} \longrightarrow_u P'$. Consider the cases for G.

Case 1 $G \equiv X$.
Then $\mu X.E \longrightarrow_u P'$ and thus $E\{\mu X.E/X\} \longrightarrow_u P'$. By induction, for some Q' and Q'', $E\{\mu X.F/X\} \longrightarrow_u Q''$ with $Q' \sim Q''$ and $(P', Q') \in \mathcal{S}$. Since $E \sim F$, we have $F\{\mu X.F/X\} \longrightarrow_u Q'''$ and $Q'' \sim Q'''$. Thus $\mu X.F \longrightarrow_u Q'''$ with $Q' \sim Q'''$ and $(P', Q') \in \mathcal{S}$.

Case 2 $G \equiv \alpha(t)_v^{v'}.\, G'$.
Then

$$G\{\mu X.E/X\} \equiv \alpha(t)_v^{v'}.\,(G'\{\mu X.E/X\})$$

and

$$P' \equiv \alpha(t)_{v \dot{-} u}^{v' \dot{-} u}.\,(G'\{\mu X.E/X\})\{t + u/t\}$$

where $0 \leq u \leq v'$, Also

$$G\{\mu X.F/X\} \equiv \alpha(t)_v^{v'}.\,(G'\{\mu X.F/X\}) \longrightarrow_u \alpha(t)_{v \dot{-} u}^{v' \dot{-} u}.\,(G'\{\mu X.F/X\})\{t{+}u/t\}$$

Since $\mu X.E$ and $\mu X.F$ are closed with respect to time variables, we have

$$\alpha(t)_{v \dot{-} u}^{v' \dot{-} u}.\,(G'\{\mu X.E/X\})\{t + u/t\} \equiv (\alpha(t)_{v \dot{-} u}^{v' \dot{-} u}.\, G'\{t + u/t\})\{\mu X.E/X\}$$

and

$$\alpha(t)_{v \dot{-} u}^{v' \dot{-} u}.\,(G'\{\mu X.F/X\})\{t + u/t\} \equiv (\alpha(t)_{v \dot{-} u}^{v' \dot{-} u}.\, G'\{t + u/t\})\{\mu X.F/X\}$$

Case 3 $G \equiv G_1 \,|\, G_2$.
Then $G_1\{\mu X.E/X\} \longrightarrow_u P_1$, $G_2\{\mu X.E/X\} \longrightarrow_u P_2$ and $P' \equiv P_1 \,|\, P_2$.
By induction

$$G_1\{\mu X.F/X\} \longrightarrow_u Q_1' \text{ with } Q_1' \sim Q_1 \text{ and } (P_1, Q_1) \in \mathcal{S}$$

and

$$G_2\{\mu X.F/X\} \longrightarrow_u Q_2' \text{ with } Q_2' \sim Q_2 \text{ and } (P_2, Q_2) \in \mathcal{S}$$

Let $Q' \equiv Q_1 \,|\, Q_2$, then $G\{\mu X.F/X\} \longrightarrow_u (Q_1' \,|\, Q_2') \sim Q'$. It remains to show that $(P', Q') \in \mathcal{S}$. Since $(P_i, Q_i) \in \mathcal{S}$ $(i = 1, 2)$, there are some H_i such that $P_i \equiv H_i\{\mu X.E/X\}$ and $Q_i \equiv H_i\{\mu X.F/X\}$. So $(P', Q') \in \mathcal{S}$.

Case 4 $G \equiv G_1 + G_2$, $G \equiv G'\backslash a$ or $G \equiv G'[S]$.
All these cases are similar to case 3.

Case 5 $G \equiv \mu Y.G'$.

(1) $Y \equiv X$.
Then X is not a free process variable of G and both $G\{\mu X.E/X\}$ and $G\{\mu X.F/X\}$ are just $\mu Y.G'$. Clearly $(P', P') \in \mathcal{S}$.

(2) $Y \not\equiv X$.
Then $G\{\mu X.E/X\} \equiv \mu Y(G'\{\mu X.E/X\})$ and $\mu Y(G'\{\mu X.E/X\}) \longrightarrow_u P'$.
Thus

$$G'\{\mu Y.G'/Y\}\{\mu X.E/X\} \longrightarrow_u P'$$

where $G'\{\mu Y.G'/Y\}\{\mu X.E/X\} \equiv G'\{\mu X.E/X\}\{\mu Y.(G'\{\mu X.E/X\})/Y\}$.
By induction, $G'\{\mu Y.G'/Y\}\{\mu X.F/X\} \longrightarrow_u Q''$ with $Q' \sim Q''$ and $(P', Q') \in \mathcal{S}$. Hence, $\mu Y.G'\{\mu X.F/X\} \longrightarrow_u Q''$ with $Q' \sim Q''$ and $(P', Q') \in \mathcal{S}$.

2. Suppose $G\{\mu X.E/X\} \xrightarrow{\alpha}_u P'$. Similar to 1. $\square$

4.3 Unique Solution of Equations

In this section, we focus on the solutions of the equations $\vec{X} = \vec{E}$. We will show that under certain conditions, the equations $\vec{X} = \vec{E}$ have an unique solution up to strong equivalence.

Definition 4.8 *A process variable X is weakly guarded in E if every occurrence of X is within some subterm $\alpha(t)_e^{e'}.F$ of E.*

Definition 4.9 *We say the equations $\vec{X} = \vec{E}$ are weakly guarded if all $\vec{X}$ are weakly guarded in $\vec{E}$.*

The most important property of a weakly guarded process is that its first possible actions are completely independent of whatever we may substitute for its process variables.

Lemma 4.14 *For any process E, which is closed with respect to time variables, and agents $\vec{P}$, if process variables $\vec{X}$ are weakly guarded in E and $E\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P'$ (or $E\{\vec{P}/\vec{X}\}\longrightarrow_u P'$) , then P' has form $E'\{\vec{P}/\vec{X}\}$ for some process E'. Furthermore, for any agents $\vec{Q}$, $E\{\vec{Q}/\vec{X}\}\xrightarrow{\alpha}_u E'\{\vec{Q}/\vec{X}\}$ (or $E\{\vec{Q}/\vec{X}\}\xrightarrow{\alpha}_u E'\{\vec{Q}/\vec{X}\}$).*

Proof We proceed by transition induction on the depth of the inference of $E\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P'$ and $E\{\vec{P}/\vec{X}\}\longrightarrow_u P'$.

1. Suppose $E\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P'$. Consider the cases for E.

Case 1 $E \equiv Y$.

Since process variables $\vec{X}$ are weakly guarded in E, Y is not in $\vec{X}$ and $Y\{\vec{P}/\vec{X}\} \equiv Y$ has no derivatives. This case is impossible.

Case 2 $E \equiv \alpha(t)_v^{v'}.F$

Then $v \leq u \leq v'$ and $P' \equiv F\{\vec{P}/\vec{X}\}\{u/t\}$. Since $\vec{P}$ are agents, we have

$$F\{\vec{P}/\vec{X}\}\{u/t\} \equiv F\{u/t\}\{\vec{P}/\vec{X}\}$$

Also, for any agents $\vec{Q}$,

$$E\{\vec{Q}/\vec{X}\} \equiv \alpha(t)_v^{v'}.(F\{\vec{Q}/\vec{X}\})\xrightarrow{\alpha}_u F\{\vec{Q}/\vec{X}\}\{u/t\}$$

and

$$F\{\vec{Q}/\vec{X}\}\{u/t\} \equiv F\{u/t\}\{\vec{Q}/\vec{X}\}$$

Case 3 $E \equiv E_1 + E_2$

Then $E_1\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P'$ or $E_2\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P'$. Suppose $E_1\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P'$. By induction, P' has the form $E'\{\vec{P}/\vec{X}\}$ and for any agents $\vec{Q}$,

$$E_1\{\vec{Q}/\vec{X}\}\xrightarrow{\alpha}_u E'\{\vec{Q}/\vec{X}\}$$

So, $E\{\vec{Q}/\vec{X}\}\xrightarrow{\alpha}_u E'\{\vec{Q}/\vec{X}\}$. The other case is similar.

Case 4 $E \equiv E_1 \mid E_2$

Subcase 4.1 $E_1\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P_1$, $E_2\{\vec{P}/\vec{X}\}\longrightarrow_u P_2$ and $P' \equiv P_1 \mid P_2$. By induction, $P_1 \equiv E_1'\{\vec{P}/\vec{X}\}$, $P_2 \equiv E_2'\{\vec{P}/\vec{X}\}$ for some E_1', E_2', and for any agents $\vec{Q}$, $E_1\{\vec{Q}/\vec{X}\}\xrightarrow{\alpha}_u E_1'\{\vec{Q}/\vec{X}\}$, $E_2\{\vec{Q}/\vec{X}\}\longrightarrow_u E_2'\{\vec{Q}/\vec{X}\}$. Hence, $P' \equiv (E_1' \mid E_2')\{\vec{P}/\vec{X}\}$ and $E\{\vec{Q}/\vec{X}\}\xrightarrow{\alpha}_u (E_1' \mid E_2')\{\vec{Q}/\vec{X}\}$.

Subcase 4.2 $E_1\{\vec{P}/\vec{X}\}\longrightarrow_u P_1$, $E_2\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P_2$ and $P' \equiv P_1 \mid P_2$. Samilar to case 4.1.

Subcase 4.3 $\alpha = \tau$, $E_1\{\vec{P}/\vec{X}\}\xrightarrow{a}_u P_1$, $E_2\{\vec{P}/\vec{X}\}\xrightarrow{\bar{a}}_u P_2$ and $P' \equiv P_1 \mid P_2$. Similar to case 4.1.

Case 5 $E \equiv F\backslash a$ or $E \equiv F[S]$. Similar to case 4.

2. Suppose $E\{\vec{P}/\vec{X}\} \longrightarrow_u P'$. Similar to 1. $\square$

Lemma 4.15 *For any process E which contain at most free process variables $\vec{X}$ and every $X_i \in \vec{X}$ is weakly guarded in E, then for any agents $\vec{P}$ and $\vec{Q}$, $|E\{\vec{P}/\vec{X}\}|_{\mathcal{T}} = |E\{\vec{Q}/\vec{X}\}|_{\mathcal{T}}$.*

The result follows directly from the definition of maximal delay time of processes.

For any weakly guarded equations $\vec{X} = \vec{E}$, if they have solutions, then the solutions are unique up to strong equivalence.

Proposition 4.16 *For any processes E_i $(i \in I)$ which are closed with respect to time variables and contain at most free process variables X_i $(i \in I)$, let all X_i $(i \in I)$ be weakly guarded in each E_i $(i \in I)$. Then*

$$\vec{P} \sim \vec{E}\{\vec{P}/\vec{X}\} \text{ and } \vec{Q} \sim \vec{E}\{\vec{Q}/\vec{X}\} \text{ implies } \vec{P} \sim \vec{Q}$$

Proof Suppose $\vec{P} \sim \vec{E}\{\vec{P}/\vec{X}\}$ and $\vec{Q} \sim \vec{E}\{\vec{Q}/\vec{X}\}$. We want to prove $P_i \sim Q_i$ $(i \in I)$, and this follows by taking $E \equiv X_i$, if we can show that

$$\mathcal{S} = \{(E\{\vec{P}/\vec{X}\},\, E\{\vec{Q}/\vec{X}\}) \,|\, fpv(E) \subseteq \vec{X} \text{ and } ftv(E) = \emptyset\}$$

is a strong $\mathcal{T}$-bisimulation up to $\sim$.

1. Suppose $E\{\vec{P}/\vec{X}\} \xrightarrow{\alpha}_u P'$. Consider the cases for E.

Case 1 $E = X_i$

Then $E\{\vec{P}/\vec{X}\} \equiv P_i \xrightarrow{\alpha}_u P'$ and $E_i\{\vec{P}/\vec{X}\} \xrightarrow{\alpha}_u P'' \sim P'$ $(P_i \sim E_i\{\vec{P}/\vec{X}\})$. By the lemma, $P'' \equiv E'\{\vec{P}/\vec{X}\}$ and $E_i\{\vec{Q}/\vec{X}\} \xrightarrow{\alpha}_u E'\{\vec{Q}/\vec{X}\}$. But $E\{\vec{Q}/\vec{X}\} \equiv Q_i \sim E_i\{\vec{Q}/\vec{X}\}$, so $E\{\vec{Q}/\vec{X}\} \xrightarrow{\alpha}_u Q' \sim E'\{\vec{Q}/\vec{X}\}$. Hence $P' \sim \mathcal{S} \sim Q'$. and $P' \sim Q'$.

Case 2 $E \equiv \alpha(t)_v^{v'}.\, F$.

Then $v \leq u \leq v'$ and $P' \equiv F\{\vec{P}/\vec{X}\}\{u/t\}$. Since all $\vec{P}$ are agents, we have $P' \equiv F\{u/t\}\{\vec{P}/\vec{X}\}$. Also $E\{\vec{Q}/\vec{X}\} \equiv \alpha(t)_v^{v'}.\, F\{\vec{Q}/\vec{X}\} \xrightarrow{\alpha}_u Q'$, where $Q' \equiv F\{\vec{Q}/\vec{X}\}\{u/t\} \equiv F\{u/t\}\{\vec{Q}/\vec{X}\}$. Hence $(P', Q') \in \mathcal{S}$.

Case 3 $E \equiv E_1 + E_2$, $E \equiv E_1 \,|\, E_2$, $E \equiv F \backslash a$ and $E \equiv F[S]$. All these cases are similar.

2. Suppose $E\{\vec{P}/\vec{X}\} \xrightarrow{\alpha}_u P'$. Similar to 1.

3. $|E\{\vec{P}/\vec{X}\}|_{\mathcal{T}} = |E\{\vec{Q}/\vec{X}\}|_{\mathcal{T}}$ (by the lemma). $\square$

Now we are in the position of showing that for any guarded equations $\vec{X} = \vec{E}$, they have a solution, i.e. there are agents $\vec{P}$ such that $\vec{P} \sim \vec{E}\{\vec{P}/\vec{X}\}$.

Proposition 4.17 *For any processes E_i $(i \in I)$ which are closed with respect to time variables and contain at most free process variables X_i $(i \in I)$, let all $\vec{X}$ be weakly guarded in every E_i, then there are some agents $\vec{P}$ such that*

$$\vec{P} \sim \vec{E}\{\vec{P}/\vec{X}\}$$

Proof By induction on the number m of equations in $\vec{X} = \vec{E}$.

Case 1 $m = 1$. Then there is a single equation $X = E$, where E contains at most free process variable X and is closed with respect to time variables. Clearly, agent $\mu X.E$ is a solution of $X = E$.

Case 2 For $m + 1$ equations $\vec{X} = \vec{E}, x_{n+1} = E_{n+1}$, we have

$$\vec{X} = \vec{E}\{\mu X_{n+1}.E_{n+1}/X_{n+1}\}$$

By induction, there are m agents $\vec{P}$ such that

$$\vec{P} \sim \vec{E}\{\mu X_{n+1}.E_{n+1}/X_{n+1}\}\{\vec{P}/\vec{X}\}$$

Also, equation $X_{m+1} = E_{m+1}\{\vec{P}/\vec{X}\}$ has a solution

$$P_{n+1} = \mu X_{m+1}.E_{m+1}\{\vec{P}/\vec{X}\}$$

Clearly, $\vec{P}$, P_{m+1} are agents which we are looking for. □

Therefore, any weakly guarded equations $\vec{X} = \vec{E}$, where $\vec{E}$ are closed with respect to time variables and contain at most free process variables $\vec{X}$, have an unique solution up to strong equivalence.

Corollary 4.18 (Unique Solution) *For any weakly guarded equations $\vec{X} = \vec{E}$, where $\vec{E}$ are closed with respect to time variables and contain at most free process variables $\vec{X}$, there is an unique solution $\vec{P}$ (up to strong equivalence) such that*

$$\vec{P} \sim \vec{E}\{\vec{P}/\vec{X}\}$$

4.4 Decidability

In this section, we only consider finite processes, i.e. those without recursion and constants.

For any process P, there is a normal form P' (by using the above laws) such that $P \sim P''$. In the sequel of this section, we suppose every process is in normal form.

Definition 4.10 *For any processes P and Q which contain at most free time variables $\vec{t} = (t_1, \ldots, t_n)$, and a formula A which contains at most variables $(t_1, \ldots, t_n)$, we say A is a condition for P to be bisimilar to Q, written as $A \models P \sim Q$, if for any $\vec{u} \in \mathcal{T}^n$, $A\{\vec{u}/\vec{t}\}$ implies $P\{\vec{u}/\vec{t}\} \sim Q\{\vec{u}/\vec{t}\}$.*

Definition 4.11 *A formula A is said to be the weakest condition for P to be bisimilar to Q if $A \models P \sim Q$ and $B \models P \sim Q$ implies $B \rightarrow A$.*

Proposition 4.19 *If A is the weakest condition for P to be bisimilar to Q, then*

$$B \models P \sim Q \quad \textit{if and only if} \quad B \rightarrow A$$

For any processes P and Q, we construct $WC(P,Q)$, the weakest condition for P to be bisimilar to Q.

Definition 4.12 *For any processes P and Q, where*

$$P \equiv \sum_{i \in I} a_i(t_i)_{e_i}^{e_i'} . P_i + (e)\delta$$

and

$$Q \equiv \sum_{j \in J} b_j(s_j)_{f_j}^{f_j'} . Q_j + (f)\delta$$

let $WC(P,Q) \quad = \quad (e = f) \bigwedge$

$$\bigwedge_{i \in I} (\forall t(max(0, e_i) \leq t \bigwedge t \leq e_i' \rightarrow \bigvee_{\substack{a_i = b_j \\ j \in J}} (f_j \leq t \bigwedge t \leq f_j' \bigwedge WC(P_i\{t/t_i\}, Q_j\{t/s_j\})))))$$

$$\bigwedge_{j \in J} (\forall s(max(0, f_j) \leq s \bigwedge s \leq f_j' \rightarrow \bigvee_{\substack{a_i = b_j \\ i \in I}} (e_i \leq s \bigwedge s \leq e_i' \bigwedge WC(P_i\{s/t_i\}, Q_j\{s/s_j\}))))$$

where t, s are some fresh variables.

We show that the formula $WC(P,Q)$ is the weakest condition for P to be bisimilar to Q.

Proposition 4.20 *For any processes P and Q, $WC(P,Q)$ is the weakest condition for P to be bisimilar to Q.*

The proof can be found in [Che91b].

Corollary 4.21 *For any processes P and Q, $WC(P,Q)$ if and only if $P \sim Q$.*

For any time expression e over $(\mathcal{T} \bigcup \{\infty\}, \leq)$, we have $e \leq \infty$. For any processes P and Q, we can replace terms of the form $e \leq \infty$ by $0 = 0$ throughout $WC(P,Q)$.

Definition 4.13 *So for any processes P and Q, let $WC^*(P,Q)$ be the result of substituting all terms of form $e \leq \infty$ by $0=0$ in $WC(P,Q)$.*

Lemma 4.22 *For any processes P and Q, $WC(P,Q)$ if and only if $WC^*(P,Q)$.*

Lemma 4.23 *For any processes P and Q which contain at most time variables $(t_1, \ldots, t_n)$,*

$$P \sim Q \text{ if and only if } \forall t_1 \ldots \forall t_n WC^*(P,Q)$$

If the time domain is $(\mathbf{N} \bigcup \{\infty\}, \leq)$, then for any processes P and Q which contain at most free time variables $(t_1, \cdots, t_n)$, $\forall t_1 \cdots \forall t_n . WC^*(P,Q)$ is a sentence of the first order theory $(\mathbf{N}, +)$.

Theorem 4.24 $\sim$ *is decidable over time domain $(\mathbf{N} \bigcup \{\infty\}, \leq)$.*

The result follows from the decidability of the first-order theory of the integers with $+$. Since the first-order theory of the integers with $+$ and $\times$, i.e. Th$(\mathbf{N}, +, \times)$, is undecidable, this justifies our decisions in the choice of time expressions.

If the time domain is $(\mathbf{R} \bigcup \{\infty\}, \leq)$, then for any processes P and Q which contain at most free time variables $(t_1, \cdots, t_n)$, $\forall t_1 \cdots \forall t_n . WC^*(P,Q)$ is a sentence of the first order theory $(\mathbf{R}, +, \times)$.

Theorem 4.25 $\sim$ *is decidable over dense time domain $(\mathbf{R} \bigcup \{\infty\}, \leq)$.*

The result follows from the decidability of the first-order theory of the reals with $+$ and $\times$ (Tarski, 1948).

5 Behavoiur Abstraction in Timed Processes

5.1 Weak Bisimulation

The strong bisimulation does not consider the problem of abstraction. It deals with the internal action τ in the same way as all other actions and requires that every τ action in one agent must be matched by a τ action of the other. In this section, we define a weak equivalence which allows a τ action to be matched by *zero* or *more* τ actions.

Definition 5.1 *We say $P \stackrel{\epsilon}{\Longrightarrow}_u P'$ if $P \stackrel{\tau}{\longrightarrow}_{u_1} \cdots \stackrel{\tau}{\longrightarrow}_{u_n} \cdot \longrightarrow_v P'$ for some $u_1, \ldots, u_n, v \in \mathcal{T}$ such that $u = v + u_1 + \cdots + u_n$.*

Since the internal action τ is not observable, an agent which performs some τ actions during time u can not be distinguished from an agent which just idles time u without any action, including τ actions. Thus $P \stackrel{\epsilon}{\Longrightarrow}_u P'$ means that process P idles time u without any observable actions.

Definition 5.2 *We say* $P \overset{a}{\Longrightarrow}_u P'$ *if* $P \overset{\epsilon}{\Longrightarrow}_v \cdot \overset{a}{\longrightarrow}_w \cdot (\overset{\tau}{\longrightarrow}_0)^* P'$ *for some* $v, w \in \mathcal{T}$ *such that* $u = v + w$.

Transition $P \overset{a}{\Longrightarrow}_u P'$ means that an action a is observed at time u. However, before action a, there may be many (but finite) unobservable actions, i.e. τ actions. Also there may be many (but finite) internal actions happening at the same time as action a.

Notation $\hat{\alpha} \overset{\text{def}}{=} \text{if } \alpha = \tau \text{ then } \epsilon \text{ else } \alpha$

Definition 5.3 *A binary relation* $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ *is a weak bisimulation if* $(P, Q) \in \mathcal{R}$ *implies that for any* $\alpha \in \mathrm{Act}$ *and* $u \in \mathcal{T}$

1 *if* $P \overset{\alpha}{\longrightarrow}_u P'$, *then there is a* Q' *such that* $Q \overset{\hat{\alpha}}{\Longrightarrow}_u Q'$ *and* $(P', Q') \in \mathcal{R}$.

2 *if* $P \longrightarrow_u P'$, *then there is a* Q' *such that* $Q \overset{\epsilon}{\Longrightarrow}_u Q'$ *and* $(P', Q') \in \mathcal{R}$.

3 *if* $Q \overset{\alpha}{\longrightarrow}_u Q'$, *then there is a* P' *such that* $P \overset{\hat{\alpha}}{\Longrightarrow}_u P'$ *and* $(P', Q') \in \mathcal{R}$.

4 *if* $Q \longrightarrow_u Q'$, *then there is a* P' *such that* $P \overset{\epsilon}{\Longrightarrow}_u P'$ *and* $(P', Q') \in \mathcal{R}$.

We say two agents P and Q are weak bisimilar if there is a weak bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

Definition 5.4 $\approx = \bigcup\{\mathcal{R} \mid \mathcal{R} \text{ is a weak bisimulation}\}$

Proposition 5.1 $\approx$ *is the largest weak bisimulation. Moreover* $\approx$ *is an equivalence relation.*

The proof is similar to that for strong bisimulation.

Proposition 5.2 $P \sim Q$ *implies* $P \approx Q$

The proof follows directly from the definitions of strong and weak bisimulations. All laws for strong equivalence still hold for weak equivalence.

Proposition 5.3

1 $\tau(t)_e^{e'}.(t \gg (e)P) + (e)P \approx \tau(t)_e^{e'}.(t \gg (e)P)$, *where* $e \leq e'$ *and* $t \notin ftv(P)$.

2 $Q + \tau(t)_e^{e'}.(t \gg (e)P) + (e)P \approx Q + \tau(t)_e^{e'}.(t \gg (e)P)$, *where* $|Q|_{\mathcal{T}} \leq e \leq e'$ *and* $t \notin ftv(P)$.

3 $\tau(t)_e^{e'}.(t \gg ((e)P + (e')\delta)) \approx (e)P + (e')\delta$, *where* $e \leq e'$ *and* $t \notin ftv((e)P + (e')\delta)$.

4 $Q + \tau(t)_e^{e'}.(t \gg ((e)P + (e')\delta) \approx Q + (e)P + (e')\delta$, *where* $|Q|_{\mathcal{T}} < e \leq e'$ *and* $t \notin ftv((e)P + (e')\delta)$.

Proof All properties may be proved by exhibiting appropriate weak bisimulations. For (3), it only needs to show that

$$\{(\tau(t)_e^{e'}.(t \gg ((e)P + (e')\delta)), (e)P + (e')\delta) \mid t \notin ftv((e)P + (e')\delta)\}$$

is a weak bisimulation. $\square$

Similar to CCS, weak equivalence is not a congruence relation. In general, it does not preserve summation. For example, $\tau(t)_2^8.nil \approx nil$, but $\tau(t)_2^8.nil + a(s)_8^\infty.nil \not\approx nil + a(s)_8^\infty.nil$. The right hand side may always perform action a, but the left hand side may reject action a. However, weak equivalence is preserved by the other operations.

Proposition 5.4 *For any agents P and Q, $P \approx Q$ implies*

1 $\alpha(t)_e^{e'}.P \approx \alpha(t)_e^{e'}.Q$

2 $P \mid R \approx Q \mid R$

3 $P \backslash a \approx Q \backslash a$

4 $P[S] \approx Q[S]$

All properties may be proved by exhibiting appropriate weak bisimulations. As with strong bisimulation, we may also generalize weak bisimulation to processes.

5.2 Observational Congruence

Weak equivalence is not fully substitutive. In general, it is not preserved by nondeterministic choice. In this section, we define an equality which will be fully substitutive and also the largest congruence relation included by $\approx$.

Definition 5.5 *We say $E =^+ F$ if for any agent P, we have $E + P \approx F + P$.*

Clearly $=^+$ is included in $\approx$. In fact $=^+$ lies between the strong and the weak equivalences.

Proposition 5.5 *$E \sim F$ implies $E =^+ F$ and $E =^+ F$ implies $E \approx F$.*

Proof The first part follows from the facts that strong equivalence is preserved by summation and $E \sim F$ implies $E \approx F$. The second part is direct (by taking $P \equiv \delta$ in the definition of $=^+$).

The converse of the above does not hold. For example, $\tau(t)_0^0..nil \approx nil$, but $\tau(t)_0^0.nil \neq^+ nil$. Also $a(t)_2^{10}.\tau(s)_0^0.nil =^+ a(t)_2^{10}.nil$, but $a(t)_2^{10}.\tau(s)_0^0.nil \not\sim a(t)_2^{10}.nil$.

We say an agent P is stable if P cannot perform an internal action τ as its first action.

Definition 5.6 *An agent P is stable if for any $u \in \mathcal{T}$ and any $P' \in \mathcal{P}$, $P \stackrel{\tau}{\not\longrightarrow}_u P'$.*

Proposition 5.6 *For any stable agents P and Q, $P \approx Q$ implies $P =^+ Q$*

Proof We only need to show that

$$\{(P + R,\, Q + R) \,|\, P \approx Q \text{ and } P, Q \text{ stable}\} \bigcup \approx$$

is a weak bisimulation.

Proposition 5.7 *$P \approx Q$ implies $\alpha(t)_e^{e'}.\, P =^+ \alpha(t)_e^{e'}.\, Q$.*

Proof We only need to show that

$$\{(\alpha(t)_e^{e'}.\, P + R,\, \alpha(t)_e^{e'}.\, Q + R) \,|\, P \approx Q\} \bigcup \approx$$

is a weak bisimulation.

Proposition 5.8 (τ-Laws)

1 $\tau(t)_0^{e'}.\, (\alpha(s)_0^0.\, P + Q) =^+ \alpha(s)_0^0.\, (P\{0/t\}) + \tau(t)_0^{e'}.\, (\alpha(s)_0^0.\, P + Q) \qquad e' \geq 0$

2 $\alpha(t)_e^{e'}.\, (\tau(s)_0^0.\, P + Q) =^+ \alpha(t)_e^{e'}.\, (P\{0/s\}) + \alpha(t)_e^{e'}.\, (\tau(s)_0^0.\, P + Q)$

3 $\alpha(t)_f^{f'}.\, (Q + \tau(s)_e^{e'}.\, (t \gg (e)P) + (e)P) =^+ \alpha(t)_f^{f'}.\, (Q + \tau(s)_e^{e'}.\, (t \gg (e)P))$, *where* $|Q|_{\mathcal{T}} \leq e \leq e'$ *and* $t \notin ftv(P)$.

4 $\alpha(t)_f^{f'}.\, (Q + \tau(s)_e^{e'}.\, (t \gg ((e)P + (e')\delta))) =^+ \alpha(t)_f^{f'}.\, (Q + (e)P + (e')\delta)$, *where* $|Q|_{\mathcal{T}} < e \leq e'$ *and* $s \notin ftv((e)P + (e')\delta)$.

(1) and (2) may be proved be showing appropriate weak bisimulations. (3) and (4) follow from the properties of weak bisimulation and the above property.

We can also give an alternative definition of $=^+$ in the style of bisimulation. To do so, we first define an equivalence $=^c$ and then show that $=^+$ and $=^c$ coincide.

Definition 5.7 *A binary relation $\mathcal{R}$ over agents is a C-bisimulation if $(P, Q) \in \mathcal{R}$ implies that for any $\alpha \in Act$ and $u \in \mathcal{T}$*

(1) *If $P \stackrel{\alpha}{\longrightarrow}_0 P'$, then $Q \stackrel{\alpha}{\Longrightarrow}_0 Q'$ and $P' \approx Q'$ for some Q'.*

(2) *If $P \longrightarrow_u P'$, then $Q \longrightarrow_u Q'$ and $(P', Q') \in \mathcal{R}$.*

(3) *symmetric to* (1) *and* (2).

Definition 5.8 $=^c \stackrel{\text{def}}{=} \bigcup\{\mathcal{R} \,|\, \mathcal{R}$ *is a* $C\!-\!bisimulation\}$.

Proposition 5.9 *For any agents P and Q, $P =^c Q$ if and only if $P =^+ Q$.*

Proof ($\Rightarrow$) It is enough to show that

$$\{(P + R,\, Q + R) \mid P =^c Q\}$$

is a weak bisimulation.
($\Leftarrow$) Suppose $P \neq^c Q$ and for any $a \in Act$, $u \in \mathcal{T}$

Case 1 $P \longrightarrow_u \cdot \stackrel{\alpha}{\longrightarrow}_0$, but whenever $Q \longrightarrow_u \cdot \stackrel{\alpha}{\Longrightarrow}_0 Q'$ then $P' \not\approx Q'$.
We choose $R \equiv \ell(t)^u_u.\, nil$, where ℓ is a fresh action which does not occur in P and Q. Since $P + R \longrightarrow_u \cdot \stackrel{\alpha}{\longrightarrow}_0 P'$, we should show that whenever $Q + R \Longrightarrow_u \stackrel{\alpha}{\Longrightarrow}_0 Q'$ then $P' \not\approx Q'$. Clearly, $\Longrightarrow_u \equiv \longrightarrow_u$, otherwise $Q + R \Longrightarrow_u Q''$, where $Q \Longrightarrow_u Q''$, and $u \gg (P + R) \not\approx Q''$. If $\alpha = \tau$ and $Q' \equiv u \gg (Q + R)$, then $P' \not\approx Q'$ (Q' may perform action ℓ). In all other cases, we have $Q + R \longrightarrow_u \cdot \stackrel{\alpha}{\Longrightarrow}_0 Q'$, where $Q \longrightarrow_u \cdot \stackrel{\alpha}{\Longrightarrow}_0 Q'$, and $P' \not\approx Q'$.

Case 2 $P \longrightarrow_u P'$, but $|Q|_{\mathcal{T}} < u$.
We choose a v, where $|Q|_{\mathcal{T}} < v \leq u$, and let $R \equiv \ell(t)^v_v.\, nil$, where ℓ is a fresh action which does not occur in P and Q. Then $P + R \longrightarrow_v v \gg (P + R)$ is not matched by $Q + R$.

All other cases are similar. $\square$

The equivalence $=^+$ is fully substitutive, as shown by the following propositions.

Proposition 5.10 *For any processes E and F, $E =^+ F$ implies*

1 $\alpha(t)^{e'}_e.\, E =^+ \alpha(t)^{e'}_e.\, F$

2 $E + G =^+ F + G$

3 $E \mid G =^+ F \mid G$

4 $E \backslash a =^+ F \backslash a$

5 $E[S] =^+ F[S]$

All of these properties may be proved by showing appropriate C-bisimulations.

Proposition 5.11 *For any processes $\vec{E}$ and $\vec{F}$ which contain at most process variables $\vec{X}$, $\vec{E} =^+ \vec{F}$ implies $\mu\vec{X}.\vec{E} =^+ \mu\vec{X}.\vec{F}$.*

Proof For simplicity, we only consider one equation. Let E and F be closed with respect to time variables and contain at most free process variable X. Suppose $E =^+ F$. By showing that

$$\mathcal{S} \stackrel{\text{def}}{=} \{(G\{\mu X.E/X\},\, G\{\mu X.F/X\}) \,|\, ftv(G) = \text{ and } fpv(G) \subseteq \{X\}\}$$

is a weak bisimulation up to $\approx$, we can establish $\mu X.E \approx \mu X.F$ (by taking $G \equiv X$). To achieve $\mu X.E =^+ \mu X.F$, we need to prove that $\mathcal{S}$ has the following properties:

(1) If $G\{\mu X.E/X\} \xrightarrow{\alpha}_0 P'$, then $G\{\mu X.F/X\} \stackrel{\alpha}{\Longrightarrow}_0 Q'$ and $P' \approx \cdot\mathcal{S}\cdot \approx Q'$ for some Q'.

(2) If $G\{\mu X.E/X\} \longrightarrow_u P'$, then $G\{\mu X.F/X\} \longrightarrow_u Q'$ and $P' \sim \cdot\mathcal{S}\cdot \sim Q'$ for some Q'.

The above two properties may be proved by induction on the inferences. Clearly, these two properties make $\mathcal{S}$ to be a weak bisimulation up to $\approx$. Hence $P' \approx \cdot\mathcal{S}\cdot \approx Q'$ implies $P' \approx Q'$. By choosing $G \equiv X$, we have $\mu X.E =^+ \mu X.F$. Note that if $\mathcal{S}$ is a C-bisimulation, then $\sim \cdot\mathcal{S}\cdot \sim$ is also a C-bisimulation. □

5.3 Unique Solution of Equations

In this section, we show that under certain conditions the solutions of equations $\vec{X} = \vec{E}$ are still unique up to observational congruence.

Definition 5.9 *X is sequential in E if every subterm of E which contains X, apart from X itself, is of the form $\alpha(t)_e^{e'}.F$ or $F_1 + F_2$.*

Definition 5.10 *X is guarded in E if every occurrence of X is within some subterm $a(t)_e^{e'}.F$ of E, where $a \neq \tau$.*

We say that equations $\vec{X} = \vec{E}$ are guarded (sequential) if all $\vec{X}$ are guarded (sequential) in $\vec{E}$. Similar to weak guardedness, the guardedness ensures that process variables cannot affect the first non-τ action.

Lemma 5.12 *Let E be guarded, sequential and closed with respect to time variables, containing at most free process variable $\vec{X}$. If $E\{\vec{P}/\vec{X}\} \xrightarrow{\alpha}_u P'$, then there is a process H such that $P' \equiv H\{\vec{P}/\vec{X}\}$ and for any agents $\vec{Q}$, $E\{\vec{Q}/\vec{X}\} \xrightarrow{\alpha}_u H\{\vec{P}/\vec{X}\}$. Moreover H is sequential and contains at most free process variables $\vec{X}$. Also if $\alpha = \tau$, then H is still guarded.*

Proof Suppose $E\{\vec{P}/\vec{X}\}\xrightarrow{\alpha}_u P'$. We proceed by induction on the structure of E. If E is a Constant, a Composition, a Restriction or a Relabelling, then it contains no free process variables and the result is immediate. If E is a Summation, the result follows from induction. If $E \equiv \alpha(t)_v^{v'}.F$, then $P' \equiv F\{\vec{P}/\vec{X}\}\{u/t\}$, where $v \leq u \leq v'$. Since $\vec{P}$ are agents, $F\{\vec{P}/\vec{X}\}\{u/t\} \equiv F\{u/t\}\{\vec{P}/\vec{X}\}$. Also, for any agents $\vec{Q}$, $E\{\vec{Q}/\vec{X}\}\xrightarrow{\alpha}_u F\{\vec{Q}/\vec{X}\}\{u/t\}$ and $F\{\vec{Q}/\vec{X}\}\{u/t\} \equiv F\{u/t\}\{\vec{Q}/\vec{X}\}$. The result follows by taking $H \equiv F\{u/t\}$. Note that H must also be guarded if $\alpha = \tau$. $\square$

Now we can show that for any guarded and sequential equations $\vec{X} = \vec{E}$, where processes $\vec{E}$ contain at most free process variables $\vec{X}$, there is a unique solution up to observational congruence.

Proposition 5.13 *Let $\vec{E}$ be guarded and sequential processes which are closed with respect to time variable and contain at most free process variables $\vec{X}$,*

$$\vec{P} =^+ \vec{E}\{\vec{P}/\vec{X}\} \text{ and } \vec{Q} =^+ \vec{E}\{\vec{Q}/\vec{X}\} \text{ implies } \vec{P} =^+ \vec{Q}$$

Proof For simplicity, we only consider one equation $X = E$. Suppose $P =^+ E\{P/X\}$ and $Q =^+ E\{Q/X\}$. Let

$$\mathcal{S} \stackrel{\text{def}}{=} \{(G\{P/X\},\, G\{Q/X\}) \,|\, \mathit{ftv}(G) = \emptyset \text{ and } \mathit{fpv}(G) \subseteq \{X\} \text{ and } G \text{ sequential}\}$$

If we can show that $\mathcal{S}$ has the properties:

(1) If $G\{P/X\}\xrightarrow{\alpha}_0 P'$, then $G\{Q/X\}\stackrel{\alpha}{\Longrightarrow}_0 Q'$ and $P' \approx \cdot\mathcal{S}\cdot \approx Q'$ for some Q'.

(2) If $G\{P/X\}\longrightarrow_u P'$, then $G\{Q/X\}\longrightarrow_u Q'$ and $P' \sim \cdot\mathcal{S}\cdot \sim Q'$ for some Q'.

then, by symmetric arguments and by taking $G \equiv X$, we have the result $P =^+ Q$.

To prove (1), we first show that if $G\{P/X\}\stackrel{\epsilon}{\Longrightarrow}_0 P'$, then $G\{Q/X\}\stackrel{\epsilon}{\Longrightarrow}_0 Q'$ and $P' \approx \cdot\mathcal{S}\cdot \approx Q'$ for some Q'. Suppose $G\{P/X\}\stackrel{\epsilon}{\Longrightarrow}_0 P'$. Since $G\{P/X\} =^+ G\{E\{P/X\}/X\}$ (substitutivity of equality), we have

$$G\{E\{P/X\}/X\}\stackrel{\epsilon}{\Longrightarrow}_0 P'' \approx P' \text{ for some } P''$$

But $G\{E/X\}$ is guarded and sequential, by the above lemma there is a sequential G' such that

$$P'' \equiv G'\{P/X\} \text{ and } (G\{E/X\})\{P/X\}\stackrel{\epsilon}{\Longrightarrow}_0 G'\{Q/X\}$$

Also $G\{Q/X\} =^+ G\{E\{P/X\}/X\} \equiv (G\{E/X\})\{P/X\}$, we have

$$G\{Q/X\}\stackrel{\epsilon}{\Longrightarrow}_0 Q' \approx G'\{Q/X\} \text{ for some } Q'$$

Hence, $P' \approx G'\{P/X\}\mathcal{S}G'\{Q/X\} \approx Q'$.
Now suppose $G\{P/X\}\xrightarrow{\alpha}_0 P'$. Then

$$(G\{E/X\})\{P/X\}\overset{\epsilon}{\Longrightarrow}_0 \cdot \xrightarrow{\alpha}_0 \overset{\epsilon}{\Longrightarrow}_0 P'' \approx P'$$

By applying the above lemma repeatedly, we have a sequential G' such that

$$(G\{E/X\})\{P/X\}\overset{\epsilon}{\Longrightarrow}_0 \cdot \xrightarrow{\alpha}_0 G'\{P/X\}\overset{\epsilon}{\Longrightarrow}_0 P'' \approx P'$$

and

$$(G\{E/X\})\{Q/X\}\overset{\epsilon}{\Longrightarrow}_0 \cdot \xrightarrow{\alpha}_0 G'\{Q/X\}$$

Hence

$$G'\{Q/X\}\overset{\epsilon}{\Longrightarrow}_0 Q'' \text{ with } P'' \approx \cdot\mathcal{S}\cdot \approx Q''$$

and

$$G\{Q/X\}\overset{\alpha}{\Longrightarrow}_0 Q' \approx Q''$$

So

$$P' \approx \cdot\mathcal{S}\cdot \approx Q'$$

The proof of (2) is similar to that for strong equivalence. □

Proposition 5.14 *For any processes E_i $(i \in I)$ which are closed with respect to time variables and contain at most free process variables X_i $(i \in I)$, let all $\vec{X}$ are guarded and sequential in every E_i $(i \in I)$. Then there are some agents $\vec{P}$ such that*

$$\vec{P} =^+ \vec{E}\{\vec{P}/\vec{X}\}$$

The proof is similar to that for strong equivalence.

Corollary 5.15 (Unique Solution) *For any guarded and sequential equations $\vec{X} = \vec{E}$, where $\vec{E}$ are closed with respect to time variables and contain at most free process variables $\vec{X}$, there is an unique solution $\vec{P}$ (up to observational congruence) such that*

$$\vec{P} =^+ \vec{E}\{\vec{P}/\vec{X}\}$$

6 Conclusion

In this paper, we propose a general interleaving model for real-time systems. We make no assumption about the underlying nature of time allowing time to be discrete (such as the integers) or dense (such as the rationals and the reals). We also make no assumption of the Maximal Progress Principle. The time variables of the language allow us to represent the notion of time dependency and the language is therefore more expressive than those without time variables. We extend the notion of bisimulation to timed processes. Strong equivalence over finite timed processes, those processes without recursion, is still decidable and the decidability is independent of the choice of time domain. In [Che91b], we also present a sound and complete proof system for strong equivalence over finite processes. The proof system is finite in the sense that there are no infinite proof rules. Therefore the proof system is of practical significance.

The works such as [HR90, Gro90, Sif90] are only for discrete time and therefore cannot express the notion of time dependency. The works in [BB91, CAM90, MT90, Wan90] all make no assumption of the underlying nature of time. However, the works in [CAM90, Wan90] are based on an assumption of the maximal progress principle. There are not time variables in Moller and Tofts' *TCCS* [MT90] and it cannot express the notion of time dependency. There is no expansion law over dense time in *TCCS*. Maybe this is why some people misunderstand that *TCCS* is only for the discrete time. Baeten and Bergstra's ACPρI is on a different process algebra *ACP* and has a notion of integration which can be seen as an alternative composition over a continuum of alternatives. It is more general than our *Timed CCS*. However, to get the completeness result, Klusener [Klu91] had to introduce some machinery, such as *basic* terms and *condition* terms, and restrict to a sublanguage which only allows "prefixed integration". In fact, this restricted sublanguage just corresponds to our *Timed CCS*. Klusener's proof system contains some infinite proof rules and is for absolute time, not for relative time. The approach of [Che91b] may also be used to the restricted language of ACPρI.

As future work, we may consider proof systems for strong equivalence over regular processes, i.e. the processes which may contain recursion. Also, we may consider proof systems for weak equivalence.

Acknowledgements: I would like to thank S. Anderson for suggesting me to study timed calculi and also many helpful discussions and comments, to F. Moller for many helpful discussions about timed calculi and also useful comments, to J. Power for many helpful discussions and suggestions on the models of timed processes, to M. Jerrum for interesting discussions on the decidability of strong equivalence.

References

[BB91] J.C.M. Baeten and J.A. Bergstra(1991), Real Time Process Algebra, Formal Aspects of Computing, 3(2), pp142-188

[BKR86] M. Ben-Or, D. Kozen and J. Reif(1986), The Complexity of Elementary Algebra and Geometry, Journal of Computer and System Sciences, 32, pp 251-264

[BK84] J.A. Bergstra and J.W. Klop(1984), Process Algebra for Synchronous Communication, Information and Control 60(1-3), pp. 109-137

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

[Che91a] L. Chen(1991), Specification and Verification of Real-Time Systems, Draft Note.

[Che91b] L. Chen(1991), Decidability and Completeness in Real-Time Processes, Technical Report ECS-LFCS-91-185, Edinburgh University

[Che91c] L. Chen(1991), Timed Synchronization Trees, Draft, LFCS, Edinburgh University

[CAM90] L. Chen, S. Anderson and F. Moller(1990), A Timed Calculus of Communicating System, Technical Report ECS-LFCS-90-127, University of Edinburgh

[Col75] G. E. Collins(1975), Quantifer Elimination for Real Closed Fields by Cylindric Algebraic Decomposition, in Proc. 2nd GI Conference on Automata Theory and Formal Language, pp 134-183.

[End72] H. B. Enderton(1972), A Mathematical Introduction to Logic, Academic Press, Inc.

[Gro90] J.F. Groote(1990), Specification and Verification of Real Time Systems in ACP, Report CS-R9015, CWI, Amsterdam

[HR90] M. Hennessy and T. Regan(1990), A Temporal Process Algebra, Technical Report 2/90 University of Sussex

[Hoa85] C.A.R. Hoare(1985), Communicating Sequential Processes, Prentice-Hall international

[Klu91] A.S. Klusener(1991), Completeness in Real Time Process Algebra, Proceedings of CONCUR'91, Lecture Notes in Computer Science 527, pp96-110

[Mil80] R. Milner(1980), A Calculus of Communicating systems, Lecture Notes in Computer Science 92, Springer-verlag

[Mil83] R. Milner(1983), Calculi for Synchrony and Asynchrony, Theoretical Computer Science, 25, pp 267-310

[Mil89] R. Milner(1989), Communication and Concurrency, Prentice-Hall international

[MT90] F. Moller and C. Tofts(1990), A Temporal Calculus of Communicating System, Lecture Notes in Computer Science 458, pp 401-415

[MT91] F. Moller and C. Tofts(1991), Relating Processes With Respect to Speed, Technical Report ECS-LFCS-91-143, University of Edinburgh

[Sif90] X. Nicollin, J. Richier, J. Sifakis and J. Voiron(1990), ATP : an Algebra for Timed Processes, Laboratoire de Genie Informatique, IMAG-Campus, 38041 Grenoble Cedex, France

[Par81] D.M.R. Park(1981), Concurrency and Automata on Infinite Sequences, Lecture Notes in Computer Science 104, Springer-Verlag

[Plo81] G.D. Plotkin(1981), A Structured Approach to Operational Semantics, DAIMI FN-19, Computer Science Department, Arhus University, Denmark

[RR88] R. Reed and A. W. Roscoe(1988), A Timed Model for Communicating Sequential Processes, Theoretical Computer Science, 58, pp 249-261

[Wan90] Y. Wang(1990), Real-time Behaviour of Asynchronous Agents, Lecture Notes in Computer Science 458, pp 502-520