Approximative Analysis by Process Algebra with Graded Spatial Actions

Yoshinao ISOBE, Yutaka SATO, Kazuhito OHMAKI

Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki 305, Japan
E-mail:{ isobe|ysato|ohmaki }@etl.go.jp

Abstract. In this paper we propose a process algebra, *CCSG*, in which we can *approximately* analyze processes by neglecting unimportant distant actions. Although many kinds of process algebra have already been proposed, there is a common problem that the number of feasible action sequences explosively increases with the number of concurrent processes. Therefore, an approximative approach is useful for large systems.
We assume that each action has a *grade* which represents the importance. In CCSG, processes can be distributed in a space, and grades of observed actions decrease with distance. Hence observations of a system depend on the positions of observers. In this paper we give *shift*-(s) *equivalence* to relate observations at different positions, and give *level*-$\langle r\rangle$ *equivalence* to relate an approximative observation and the complete observation.

1 Introduction

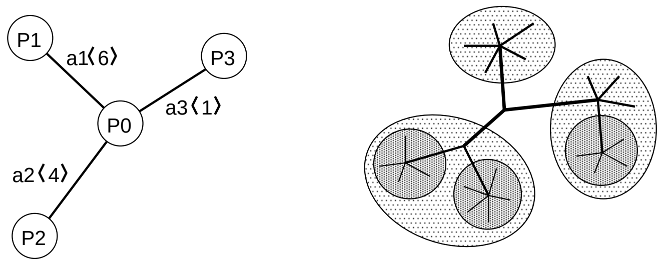
Concurrent processes are more complex than sequential processes, because actions of concurrent processes can be independently performed and sometimes synchronize with each other. Process algebra is a mathematical tool to analyze concurrent processes. Actions of processes are described as *(process) expressions* in a process algebra, then equality between the actions of two processes can be checked by rewriting their expressions according to algebraic laws of the process algebra. A real problem of analysis of concurrent processes is that the number of feasible action sequences explosively increases by interleaving of actions [1].

We propose a process algebra *CCSG* (a Calculus of Communicating Systems with Graded spatial actions) to *approximately* analyze processes. In CCSG each action has a *grade* which represents the importance and a *position* where the action occurs, thus unimportant distant actions can be neglected by observers.

Distributed processes are connected through *routers* with *loss* of grades. Each router consists of a name a and a loss r, then has the form $a\langle r\rangle$. Grades of actions observed through a router $a\langle r\rangle$ decrease by the loss r. Routers are connected in a star structure as shown in Fig.1(a). Branches represent routers and nodes represent processes. Routers can be hierarchically connected as shown in Fig.1(b).

CCSG is an extension of CCS [2]. CCS is a well known fundamental process algebra. A new combinator @ called *Route combinator* is introduced in CCSG as compared with CCS. For example, the system of Fig.1(a) is described as S_0

$$S_0 \equiv P_0|(P_1@a_1\langle 6\rangle)|(P_2@a_2\langle 4\rangle)|(P_3@a_3\langle 1\rangle)$$



(a) Star Structure (b) Hierarchical star structure

Fig. 1. Connection examples of routers

by an observer standing at the position of P_0. $\equiv$ represents syntactic identity and $|$ is a composition combinator. Notice that observations depend on positions of observers. For example, the system of Fig.1(a) is also described as S_2 and S_3

$$S_2 \equiv P_2|((P_0|(P_1@a_1\langle 6\rangle))|(P_3@a_3\langle 1\rangle))@a_2\langle 4\rangle)$$
$$S_3 \equiv P_3|((P_0|(P_1@a_1\langle 6\rangle))|(P_2@a_2\langle 4\rangle))@a_3\langle 1\rangle)$$

by observers standing at the positions of P_2 and P_3, respectively.

We give *shift*-(s) *equivalence* $\stackrel{\prec}{\sim}_{(s)}$ to relate processes observed at different positions, where s is a parameter which represents the difference between the positions. Namely s is the *route* from the position of the left-side observer to the position of the right-side observer. The route between two points is a sequence of routers between them. For example, S_2 and S_3 are shift-$(a_2\langle 4\rangle a_3\langle 1\rangle)$ equivalent, $S_2 \stackrel{\prec}{\sim}_{(a_2\langle 4\rangle a_3\langle 1\rangle)} S_3$, because the route from P_2 to P_3 is $(a_2\langle 4\rangle a_3\langle 1\rangle)$. Particularly $\stackrel{\prec}{\sim}_{(\varepsilon)}$ is identical with strong equivalence in [2], where ε is the empty sequence.

An action of CCSG consists of a label α, a grade r, and a route s, then has the form $\alpha\langle r\rangle@s$. This @ is not the combinator over processes previously introduced. We use the same symbol @ for actions and processes, because their roles are the same and they can be distinguished by grammar. $\alpha\langle r\rangle@s$ represents that an action named α with the grade r occurs at the position pointed to by the route s. The grade r is a real number. Positive grades are assigned to important actions and negative grades are assigned to unimportant actions. s is the route from the position where the action occurs to the position of the observer. The empty route is sometimes omitted, and thus $\alpha\langle r\rangle$ is used for representing $\alpha\langle r\rangle@\varepsilon$. For example, in Fig.1(a), an action α with a grade 7 which occurs at P_1 is observed as the action $\alpha\langle 7\rangle@(a_1\langle 6\rangle a_2\langle 4\rangle)$ by an observer at P_2.

The total sum of losses of routers between two points is called the *loss distance* between them. For example, the loss distance between the positions of P_1 and P_2 is $(6+4=)$ 10. It is important that the grade of $\alpha\langle 7\rangle@(a_1\langle 6\rangle a_2\langle 4\rangle)$ is actually observed as $(7-(6+4)=)\ -3$. This decreased grade -3 by the loss distance is called the *actual grade* of $\alpha\langle 7\rangle@(a_1\langle 6\rangle a_2\langle 4\rangle)$.

In CCSG the following condition of synchronization is very important.

Two actions can synchronize only if the sum of their grades is not less than the loss distance between them.	**(Condition 1)**

For example, a graded action $\alpha\langle 9\rangle$ which occurs at P_1 can synchronize with a graded action $\overline{\alpha}\langle 3\rangle$ which occurs at P_2, because the sum $(9+3=)$ 12 of their grades is greater than the loss distance $(6+4=)$ 10 between P_1 and P_2.

It is possible to approximately analyze processes by neglecting low actual graded actions in CCSG. We give a relation called *level-*$\langle r\rangle$ *equivalence* $=_{\langle r\rangle}$ for such approximative analysis. r is a parameter which represents a level of similarity. Level-$\langle r\rangle$ equivalence bases on the following *level-*$\langle r\rangle$ *observation.*

Actions which have lower actual grades (at the position of the observer) than $-r$, can not be observed.	**(Assumption 1)**

Intuitively, r represents the radius of the observable area in level-$\langle r\rangle$ observation. Particularly level-$\langle\infty\rangle$ equivalence $=_{\langle\infty\rangle}$ corresponds to observation congruence $=$ defined in [2]. For example, the following relations hold.

$$(\alpha\langle 1\rangle@a\langle 4\rangle).P =_{\langle 2\rangle} \tau.P, \quad (\alpha\langle 1\rangle@a\langle 2\rangle).P \neq_{\langle 2\rangle} \tau.P,$$

$(\alpha\langle 1\rangle@a\langle 4\rangle).P$ is a process which can perform the action $(\alpha\langle 1\rangle@a\langle 4\rangle)$, and thereafter behaves like the process P. τ is an internal action which can not be observed. The action $(\alpha\langle 1\rangle@a\langle 4\rangle)$ need not be observed in level-$\langle 2\rangle$ observation, because its actual grade $(1-4=)$ -3 is less than the minus level, thus $-3<-2$.

An important property is that level-$\langle r\rangle$ equivalence is preserved by Composition combinator $|$. Therefore we can check level-$\langle r\rangle$ equivalence part by part.

The outline of this paper is as follows: In Section 2, we define the syntax and the semantics of CCSG. In Section 3, shift-(s) equivalence and level-$\langle r\rangle$ equivalence are defined. Then, we give a sound and complete axiom system for level-$\langle r\rangle$ equivalence of finite sequential processes. In Section 4, an example of approximative analysis in CCSG is shown. In Section 5, we discuss space process algebra already proposed. In Section 6, we conclude this paper.

2 Definition of CCSG

In Subsection 2.1, various sets used in CCSG are given. In Subsection 2.2, three operators over routes are defined, and their properties are shown. In Subsection 2.3 and 2.4, the syntax and the semantics of CCSG are defined, respectively.

2.1 Actions of CCSG

We assume that an infinite set of *names* $\mathcal{N}$ is given. The set of routers Ω, ranged over by ω, is given as the Cartesian product $\{a\langle r\rangle : a\in\mathcal{N}, r\in\mathcal{R}^+\}$ of the set of names $\mathcal{N}$ and the set of non-negative real numbers $\mathcal{R}^+$. Two routers $a_1\langle r_1\rangle$ and $a_2\langle r_2\rangle$ are not distinct if $a_1=a_2$ and $r_1=r_2$. We assume that:

All routers connected to a node are distinct from each other.	**(Assumption 2)**

As shown in Fig.2, if P_2 and P_4 are connected to P_0 through two indistinct routers $a_2\langle 4\rangle$, then it is interpreted that P_2 and P_4 are positioned in the same place. Namely, the route between P_2 and P_4 in Fig.2 is not $(a_2\langle 4\rangle a_2\langle 4\rangle)$ but ε. Thus any route between two points is expressed with no adjacent indistinct

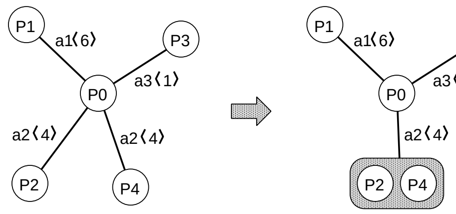


Fig. 2. The interpretation of indistinct connections

routers. Therefore the set of routes Ψ can be defined as $\{s \in \Omega^* | \textit{No-adjacent}(s)\}$, where Ω^* is the set of finite router sequences, and *No-adjacent*(s) expresses that s does not have two adjacent routers. For example, $(a_2\langle 4\rangle a_2\langle 4\rangle) \notin \Psi$.

The set of *co-names* $\overline{\mathcal{N}}$ is given as $\{\overline{a} : a \in \mathcal{N}\}$, where the overbar represents a bijection such that $\overline{\overline{a}} = a$ for all $a \in \mathcal{N}$. Then the union of $\mathcal{N}$ and $\overline{\mathcal{N}}$ is called the set of *label* $\mathcal{A}(= \mathcal{N} \cup \overline{\mathcal{N}})$ ranged over by α. The set of *observable actions* Act is the Cartesian product $\{\alpha\langle r\rangle @s : \alpha \in \mathcal{A}, r \in \mathcal{R}, s \in \Psi\}$ of $\mathcal{A}$, the set of real numbers $\mathcal{R}$, and Ψ. Finally the set of *actions* Act_τ, ranged over by μ, is given as $\mathit{Act} \cup \{\tau\}$, where τ is called an internal action.

The sets given in this subsection are summarized in Table 1.

Table 1. The sets used in CCSG

Sets	Elements	Variables	Sets	Elements	Variables
$\mathcal{N}$	name	$a, a', a_1, \cdots$	Ω	router	$\omega, \omega', \omega_1, \cdots$
$\overline{\mathcal{N}}$	co-name	$\overline{a}, \overline{a'}, \overline{a_1}, \cdots$	Ψ	route	$s, s', s_1, \cdots$
$\mathcal{R}$	real number	$r, r', r_1, \cdots$	Act	observable action	$\nu, \nu', \nu_1, \cdots$
$\mathcal{A}$	label	$\alpha, \alpha', \alpha_1, \cdots$	Act_τ	action	$\mu, \mu', \mu_1, \cdots$

2.2 Operators over Routes

In Fig.1(a), the route s_{12} from P_1 to P_2 is $(a_1\langle 6\rangle a_2\langle 4\rangle)$ and the route s_{23} from P_2 to P_3 is $(a_2\langle 4\rangle a_3\langle 1\rangle)$. In this case, the *sum* of s_{12} and s_{23} is expected to be the route $(a_1\langle 6\rangle a_3\langle 1\rangle)$ from P_1 to P_3 by considering **Assumption 2**. The sum of routes is not a simple concatenation of two routes such as $(a_1\langle 6\rangle a_2\langle 4\rangle a_2\langle 4\rangle a_3\langle 1\rangle)$.

In this subsection, three operators over routes are defined, then their properties are shown. One of them is *Sum operator* $\circ$. The sum of two routes s_1 and s_2 is a route produced by connecting the terminal point of s_1 to the initial point of s_2, and it is denoted by $(s_1 \circ s_2)$. The initial point of $(s_1 \circ s_2)$ is the initial point of s_1 and the terminal point of $(s_1 \circ s_2)$ is the terminal point of s_2.

Definition 2.1 *Sum operator* $\circ : \Psi \times \Psi \to \Psi$ *is inductively defined by*

$$s_1 \circ s_2 = \begin{cases} s_1' \circ s_2' & (s_1 = s_1'\omega, s_2 = \omega s_2') \\ s_1 s_2 & (\textit{otherwise}) \end{cases}$$

□

We explain how to calculate the sum of routes by using Fig.3. Each ω_i is a router and each s_i is a route such as s_1=$\omega_1\omega_2\omega_3\omega_4$, s_2=$\omega_4\omega_3\omega_5$, and s_3=$\omega_1\omega_2\omega_5$.

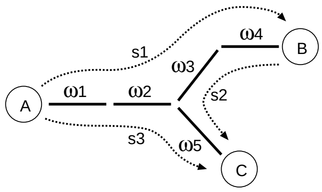


Fig. 3. An example of calculation of routes

In this case s_3 is the sum route of s_1 and s_2 as follows:

$$s_1 \circ s_2 = \omega_1\omega_2\omega_3\omega_4 \circ \omega_4\omega_3\omega_5 = \omega_1\omega_2\omega_3 \circ \omega_3\omega_5 = \omega_1\omega_2 \circ \omega_5 = \omega_1\omega_2\omega_5 = s_3$$

The next operator is *Reverse operator* `rev` to reverse the direction of a route.

Definition 2.2 *Reverse operator* $\texttt{rev} : \Psi \to \Psi$ *is inductively defined by*

$$\texttt{rev}(s) = \begin{cases} \texttt{rev}(s_2)\texttt{rev}(s_1) & (s = s_1 s_2) \\ \omega & (s = \omega) \\ \varepsilon & (s = \varepsilon) \end{cases} \qquad \square$$

The last operator is *Difference operator* $\lhd$. The difference of two routes s_1 and s_2 is a route produced by connecting the terminal point of s_1 to the terminal point of s_2, and it is denoted by $(s_1 \lhd s_2)$. The initial point of $(s_1 \lhd s_2)$ is the initial point of s_1 and the terminal point of $(s_1 \lhd s_2)$ is the initial point of s_2. Difference operator is defined by using Sum and Reverse operators.

Definition 2.3 *Difference operator* $\lhd : \Psi \times \Psi \to \Psi$ *is defined by*

$$s_1 \lhd s_2 = s_1 \circ \texttt{rev}(s_2) \qquad \square$$

For example, s_1 is the difference route of s_3 and s_2 in Fig.3 as follows:

$$s_3 \lhd s_2 = s_3 \circ \texttt{rev}(s_2) = \omega_1\omega_2\omega_5 \circ \omega_5\omega_3\omega_4 = \omega_1\omega_2 \circ \omega_3\omega_4 = \omega_1\omega_2\omega_3\omega_4 = s_1$$

It is needed to evaluate the loss distance of a route to check whether **Condition 1** is satisfied or not. The function π is given to evaluate the loss distance.

Definition 2.4 *The function* $\pi : \Psi \to \mathcal{R}^+$ *is defined by*

$$\pi(s) = \begin{cases} \pi(s_1) + \pi(s_2) & (s = s_1 s_2) \\ r & (s = a\langle r \rangle) \\ 0 & (s = \varepsilon) \end{cases} \qquad \square$$

It is often needed to evaluate the loss distance between the two terminal points of two routes whose initial points are the same. For example, the loss distance between A and B is evaluated by $\pi(s_3 \lhd s_2)$ in Fig.3.

Calculus of routes is similar to calculus of vectors. Some equations of routes are shown in Proposition 2.1. The proofs are omitted because of lack of space.

Proposition 2.1 *For any* $s, s_i \in \Psi$, *the following equations hold.*

(1) $s \circ \varepsilon = \varepsilon \circ s = s$	(6) $(s_1 \circ s_2) \lhd s_3 = s_1 \lhd (s_3 \lhd s_2)$
(2) $s \lhd s = \varepsilon$	(7) $s_1 \lhd s_2 = (s_1 \circ s) \lhd (s_2 \circ s)$
(3) $(s_1 \circ s_2) \circ s_3 = s_1 \circ (s_2 \circ s_3)$	(8) $s_1 \circ s_2 = s_3$ *iff* $s_1 = s_3 \lhd s_2$
(4) $(s_1 \lhd s_2) \lhd s_3 = s_1 \lhd (s_3 \circ s_2)$	(9) $\pi(s_1 \circ s_2) \geq \lvert\pi(s_1) - \pi(s_2)\rvert$
(5) $(s_1 \circ s_2) \lhd s_3 = s_1 \circ (s_2 \lhd s_3)$	(10) $\pi(s_1 \circ s_2) \leq \pi(s_1) + \pi(s_2)$ $\square$

2.3 Syntax of CCSG

In process algebra, actions of processes are described as *process expressions*. We introduce a set of *Variables* $\mathcal{X}$ ranged over by X and a set of *Constants* $\mathcal{K}$ ranged over by A. We define the set of process expressions $\mathcal{E}$ ranged over by $E, F, \cdots$.

Definition 2.5 *The set of process expressions $\mathcal{E}$ is the smallest set including the following expressions:*

X : *Variable* ($X \in \mathcal{X}$)	$E\|F$: *Composition*
A : *Constant* ($A \in \mathcal{K}$)	$E[f]$: *Relabelling* (f : *a relabelling function*)
$\mathbf{0}$: *Inaction*	$E\backslash^{L}_{\langle r\rangle(s)}$: *Restriction* ($r \in \mathcal{R}, s \in \Psi, L \subseteq \mathcal{A}$)
$\mu.E$: *Prefix* ($\mu \in Act_\tau$)	$E@s$: *Route* ($s \in \Psi$)
$E + F$: *Choice*	

where E and F are already in $\mathcal{E}$. □

The relabelling function f is a function from $\mathcal{A}$ to $\mathcal{A}$ such that $f(\overline{\alpha}) = \overline{f(\alpha)}$. We practically extend f over Act_τ by decreeing that $f(\alpha\langle r\rangle@s) = f(\alpha)\langle r\rangle@s$ and $f(\tau) = \tau$. Notice that names of routers can not be changed by Relabelling.

A *process* is a process expression with no Variables. The set of processes is denoted by $\mathcal{P}$ and is ranged over by $P, Q, R, \cdots$. A *Constant* is a process whose meaning is given by a defining equation. In fact, we assume that for every Constant $A \in \mathcal{K}$, there is a defining equation of the form $A \stackrel{\text{def}}{=} P$, where $P \in \mathcal{P}$.

We informally explain roles of each combinator and relations of positions of an expression and subexpressions as follows:

- $\mu.E$ can perform the action μ, and thereafter behaves like E. $\mu.E$ and E are positioned at the same place. If $\mu = \alpha\langle r\rangle@s$, then the graded action $\alpha\langle r\rangle$ occurs the route s away from E.
- $E+F$ represents a choice between E and F. The choice is made by an action of E or F. $E + F$, E, and F are positioned at the same place.
- $E|F$ represents a concurrent composition of E and F. $E|F$, E, and F are positioned at the same place.
- $E[f]$ behaves like E except that actions of E are relabelled by f. $E[f]$ and E are positioned at the same place.
- $E\backslash^{L}_{\langle r\rangle(s)}$ locally restricts actions in the *restriction area* decided by r and s. s is the route from the center of the restriction area to E. r is the *restriction power* at the center, and the restriction power decreases with loss distance. Thus, the *actual restriction power* for an action which occurs the loss distance r' away from the center, is $(r - r')$. If the absolute value of the grade of the action is less than the actual restriction power and the label of the action is included in L, then the action can not occur. We will explain this local restriction at the end of Subsection 2.4 by using an example. $E\backslash^{L}_{\langle r\rangle(s)}$ and E are positioned at the same place.
- $E@s$ behaves like E, but $E@s$ is the route s away from E.

To avoid too many parentheses, combinators have binding power in the following order: {Restriction, Relabelling, Route} >Prefix>Composition>Choice.

$$\textbf{Act}\ \frac{}{\mu.E \xrightarrow{\mu} E} \qquad \textbf{Con}\ \frac{P \xrightarrow{\mu} P'}{A \xrightarrow{\mu} P'}\ (A \stackrel{\text{def}}{=} P)$$

$$\textbf{Choice}_\mathbf{1}\ \frac{E \xrightarrow{\mu} E'}{E + F \xrightarrow{\mu} E'} \qquad \textbf{Choice}_\mathbf{2}\ \frac{F \xrightarrow{\mu} F'}{E + F \xrightarrow{\mu} F'}$$

$$\textbf{Com}_\mathbf{1}\ \frac{E \xrightarrow{\mu} E'}{E|F \xrightarrow{\mu} E'|F} \qquad \textbf{Com}_\mathbf{2}\ \frac{F \xrightarrow{\mu} F'}{E|F \xrightarrow{\mu} E|F'}$$

$$\textbf{Com}_\mathbf{3}\ \frac{E \xrightarrow{\alpha\langle r_1\rangle @ s_1} E' \quad F \xrightarrow{\overline{\alpha}\langle r_2\rangle @ s_2} F'}{E|F \xrightarrow{\tau} E'|F'}\ (r_1 + r_2 \geq \pi(s_1 \triangleleft s_2))$$

$$\textbf{Res}_\mathbf{1}\ \frac{E \xrightarrow{\alpha\langle r_1\rangle @ s_1} E'}{E\backslash^{L}_{\langle r_2\rangle(s_2)} \xrightarrow{\alpha\langle r_1\rangle @ s_1} E'\backslash^{L}_{\langle r_2\rangle(s_2)}}\ \begin{pmatrix} \alpha, \overline{\alpha} \notin L \text{ or} \\ |r_1| > r_2 - \pi(s_1 \triangleleft s_2) \end{pmatrix}$$

$$\textbf{Res}_\mathbf{2}\ \frac{E \xrightarrow{\tau} E'}{E\backslash^{L}_{\langle r\rangle(s)} \xrightarrow{\tau} E'\backslash^{L}_{\langle r\rangle(s)}} \qquad \textbf{Rel}\ \frac{E \xrightarrow{\mu} E'}{E[f] \xrightarrow{f(\mu)} E'[f]}$$

$$\textbf{Rou}_\mathbf{1}\ \frac{E \xrightarrow{\alpha\langle r\rangle @ s_1} E'}{E@s_2 \xrightarrow{\alpha\langle r\rangle @ (s_1 \circ s_2)} E'@s_2} \qquad \textbf{Rou}_\mathbf{2}\ \frac{E \xrightarrow{\tau} E'}{E@s \xrightarrow{\tau} E'@s}$$

Fig. 4. The operational semantics

2.4 Semantics of CCSG

The semantics is given by the labelled transition system $(\mathcal{E}, Act_\tau, \{\xrightarrow{\mu} : \mu \in Act_\tau\})$. $E \xrightarrow{\mu} E'$ indicates that E may perform μ and thereafter behaves like E'.

Definition 2.6 *The transition relation $\xrightarrow{\mu}$ over process expressions is the smallest relation satisfying the inference rules in Fig.4. Each rule is read as follows: if the transition relation(s) above the line are inferred and the side condition(s) are satisfied, then the transition relation below the line can be also inferred.* □

The five rules, $\textbf{Rou}_{\mathbf{1,2}}$, $\textbf{Com}_\mathbf{3}$, $\textbf{Res}_{\mathbf{1,2}}$, are different from the rules for CCS. For $\textbf{Rou}_\mathbf{1}$, the action $\alpha\langle r\rangle$ occurs the route s_1 away from E. Therefore the action occurs the route $(s_1 \circ s_2)$ away from $E@s_2$.

$\textbf{Com}_\mathbf{3}$ infers a synchronization of two actions with complementary labels. The side condition represents **Condition 1**. Namely, the sum $(r_1 + r_2)$ of their grades is not less than the loss distance $\pi(s_1 \triangleleft s_2)$ between their positions.

The side condition $(|r_1| > r_2 - \pi(s_1 \triangleleft s_2))$ of $\textbf{Res}_\mathbf{1}$ means that the action $(\alpha\langle r_1\rangle @ s_1)$ with the grade r_1, whose absolute value $|r_1|$ is greater than the actual restriction power $(r_2 - \pi(s_1 \triangleleft s_2))$ for the action, is not restricted even though the label α of the action is included in L. $\pi(s_1 \triangleleft s_2)$ is the loss distance between the center of the restriction area and the position where the action occurs. We explain this local restriction by using the following example.

$$SYS \equiv (P@s_1)\backslash^{\{\alpha\}}_{\langle 7\rangle(s_2)}, \quad P \equiv (\alpha\langle 5\rangle@\varepsilon).\mathbf{0}$$

where $s_1 = a_2\langle 1\rangle a_1\langle 4\rangle$ and $s_2 = a_3\langle 2\rangle a_1\langle 4\rangle$. The process P is the route s_1 away from SYS, and SYS locally restricts α. The route from the center of the restriction area to SYS is s_2, and the restriction power at the center is 7. Then the loss distance between the action and the center is evaluated as follows:

$$\pi(s_1 \triangleleft s_2) = \pi(s_1 \circ \texttt{rev}(s_2)) = \pi((a_2\langle 1\rangle a_1\langle 4\rangle) \circ (a_1\langle 4\rangle a_3\langle 2\rangle)) = \pi(a_2\langle 1\rangle a_3\langle 2\rangle) = 3$$

Thus the action $\alpha\langle 5\rangle$ is not restricted, because the actual restriction power for the action is $(7 - 3 =)$ 4.

3 Equality in CCSG

In Subsection 3.1, we define *shift-(s) equivalence* $\precsim_{(s)}$, introduced in Section 1. In Subsection 3.2, *weak level-$\langle r\rangle$ equivalence* $\approx_{\langle r\rangle}$ is defined before *level-$\langle r\rangle$ equivalence* $=_{\langle r\rangle}$, because level-$\langle r\rangle$ equivalence is defined based on weak level-$\langle r\rangle$ equivalence. In Subsection 3.3, we define level-$\langle r\rangle$ equivalence, which is the largest equivalence relation preserved by Choice + and included in $\approx_{\langle r\rangle}$. In Subsection 3.4, we give a sound and complete axiom system for level-$\langle r\rangle$ equivalence of finite sequential processes. In Subsection 3.5, we discuss a strong version of weak level-$\langle r\rangle$ equivalence, where the number of transitions by τ must be matched.

3.1 Shift-(s) Equivalence

We define shift-(s) equivalence by using *shift-(s) bisimulations*, in order to cancel the difference s between positions of two observers.

Definition 3.1 *Let $s \in \Psi$. A binary relation $\mathcal{S} \subseteq \mathcal{P} \times \mathcal{P}$ over processes is a* shift-(s) bisimulation *if $(P, Q) \in \mathcal{S}$ implies, for all $\alpha\langle r\rangle@s' \in Act$, that*

(i) whenever $P \xrightarrow{\alpha\langle r\rangle@s'} P'$ then, for some $Q', Q \xrightarrow{\alpha\langle r\rangle@(s'\circ s)} Q'$ and $(P', Q') \in \mathcal{S}$,
(ii) whenever $P \xrightarrow{\tau} P'$ then, for some $Q', Q \xrightarrow{\tau} Q'$ and $(P', Q') \in \mathcal{S}$,
(iii) whenever $Q \xrightarrow{\alpha\langle r\rangle@s'} Q'$ then, for some $P', P \xrightarrow{\alpha\langle r\rangle@(s'\triangleleft s)} P'$ and $(P', Q') \in \mathcal{S}$,
(iv) whenever $Q \xrightarrow{\tau} Q'$ then, for some $P', P \xrightarrow{\tau} P'$ and $(P', Q') \in \mathcal{S}$. □

Definition 3.2 *P and Q are* shift-(s) equivalent*, written $P \precsim_{(s)} Q$, if $(P, Q) \in \mathcal{S}$ for some shift-(s) bisimulation $\mathcal{S}$.* □

Although shift-(s) equivalence is not an equivalence relation, *parameterized* reflexive, symmetric, and transitive laws hold as shown in Proposition 3.1.

Proposition 3.1 (1) $P \precsim_{(\varepsilon)} P$
(2) *If $P \precsim_{(s)} Q$, then $Q \precsim_{(\texttt{rev}(s))} P$*
(3) *If $P \precsim_{(s_1)} Q$ and $Q \precsim_{(s_2)} R$, then $P \precsim_{(s_1\circ s_2)} R$* □

Therefore the total union of $\precsim_{(s)}$ over $s \in \Psi$ is an equivalence relation.

The differences of shift-(s) equivalence $\precsim_{(s)}$ from strong equivalence $\sim$ are only $(s' \circ s)$ and $(s' \triangleleft s)$ in (i) and (iii) of Definition 3.1. Thus shift-(ε) equivalence $\precsim_{(\varepsilon)}$ is strong equivalence, because $(s' \circ \varepsilon = s' \triangleleft \varepsilon = s')$. We also conventionally use the symbol $\sim$ for $\precsim_{(\varepsilon)}$. The following equations for Route @ hold.

Proposition 3.2 (1) $((\alpha\langle r\rangle @s).P)@s' \sim (\alpha\langle r\rangle @(s \circ s')).(P@s')$
(2) $(P_1|P_2)@s \sim (P_1@s)|(P_2@s)$
(3) $(P@s_1)@s_2 \sim P@(s_1 \circ s_2)$ $\square$

For (1), the action $\alpha\langle r\rangle$ occurs the route s away from P. Thus $\alpha\langle r\rangle$ is the route $(s \circ s')$ away from $P@s'$, because $P@s'$ is the route s' away from P. For (2), the route between P_1 and P_2 of the left-side is clearly ε. And the route between P_1 and P_2 of the right-side is also ε by **Assumption 2**.

The following proposition shows properties of shift-(s) equivalence very well.

Proposition 3.3 (1) *If* $P \precsim_{(s)} Q$, *then* $P \precsim_{(s \circ s')} Q@s'$
(2) *If* $P \precsim_{(s)} Q$, *then* $P@\mathtt{rev}(s') \precsim_{(s' \circ s)} Q$ $\square$

Shift-(s) equivalence is preserved by Composition combinator | as follows.

Proposition 3.4 *If* $P_i \precsim_{(s)} Q_i$ $(i \in \{1,2\})$, *then* $P_1|P_2 \precsim_{(s)} Q_1|Q_2$. $\square$

3.2 Weak Level-$\langle r\rangle$ Equivalence

In this subsection, *weak level-*$\langle r\rangle$ *equivalence* is defined based on **Assumption 1**.

First we give the sequential transition relations. The set Act^*_τ, ranged over by $t, t', \cdots$, is the set of action sequences including the empty sequence ε, and if $E \xrightarrow{\mu_1} \cdots \xrightarrow{\mu_n} E'$ for some $t = \mu_1 \cdots \mu_n \in Act^*_\tau$, then we write $E \xrightarrow{t} E'$.

Secondly, we define a (single) threshold function to neglect unobservable actions which have lower actual grades than $-r$, considering **Assumption 1**.

Definition 3.3 *The single threshold function* $\phi : Act^*_\tau \times \mathcal{R} \rightarrow Act^*$ *is defined by*

$$\phi(t,r) = \begin{cases} \phi(t_1,r)\phi(t_2,r) & (t = t_1t_2, t_1 \neq \varepsilon, t_2 \neq \varepsilon) \\ a\langle r'\rangle @s & (t = a\langle r'\rangle @s, r' - \pi(s) \geq -r) \\ \varepsilon & (otherwise) \end{cases}$$ $\square$

It is important to notice that too high graded actions are *ambiguous*, because they can synchronize with unobservable actions. More exactly, in level-$\langle r\rangle$ observation, if an action occurs the route s away from the observer and has higher grades than $(r - \pi(s))$, then it is ambiguous, because observable level decreases with loss distance. Thus we define a *double* threshold function as follows:

Definition 3.4 *The double threshold function* $\theta : Act^*_\tau \times \mathcal{R} \rightarrow Act^*$ *is defined by*

$$\theta(t,r) = \begin{cases} \theta(t_1,r)\theta(t_2,r) & (t = t_1t_2, t_1 \neq \varepsilon, t_2 \neq \varepsilon) \\ a\langle r'\rangle @s & (t = a\langle r'\rangle @s, |r'| \leq r - \pi(s)) \\ \varepsilon & (otherwise) \end{cases}$$ $\square$

For example, the following applications show properties of ϕ and θ very well.

$$\phi(\alpha\langle -2\rangle @\varepsilon, 1) = \varepsilon, \quad \phi(\alpha\langle 0\rangle @\varepsilon, 1) = \alpha\langle 0\rangle @\varepsilon, \quad \phi(\alpha\langle 2\rangle @\varepsilon, 1) = \alpha\langle 2\rangle @\varepsilon$$
$$\theta(\alpha\langle -2\rangle @\varepsilon, 1) = \varepsilon, \quad \theta(\alpha\langle 0\rangle @\varepsilon, 1) = \alpha\langle 0\rangle @\varepsilon, \quad \theta(\alpha\langle 2\rangle @\varepsilon, 1) = \varepsilon$$

In level-$\langle 1\rangle$ observation, $(\alpha\langle -2\rangle @\varepsilon)$ is unobservable since $(-2 < -1)$, and $(\alpha\langle 2\rangle @\varepsilon)$ is ambiguous since $(2 > 1)$.

We define the new labelled transition system $(\mathcal{E}, Act^*_\tau, \{\overset{t}{\Longrightarrow}_{\langle r\rangle} : t \in Act^*_\tau\})$ for any $r \in \mathcal{R}$, in which the transition relations $\overset{t}{\Longrightarrow}_{\langle r\rangle}$ implicitly includes transitions through unobservable actions and ambiguous actions in level-$\langle r\rangle$ observation.

Definition 3.5 *Let* $r \in \mathcal{R}$*. If* $\theta(t,r) = \varepsilon$ *and* $E \xrightarrow{t} E'$ *for some* $t \in Act_\tau^*$*, then* $E \overset{\varepsilon}{\Longrightarrow}_{\langle r \rangle} E'$ *(also written* $E \Longrightarrow_{\langle r \rangle} E'$*). If* $E \Longrightarrow_{\langle r \rangle} \xrightarrow{\mu_1} \Longrightarrow_{\langle r \rangle} \cdots \Longrightarrow_{\langle r \rangle} \xrightarrow{\mu_n} \Longrightarrow_{\langle r \rangle}$ E' *for some* $t = \mu_1 \cdots \mu_n \in Act_\tau^*$*, then* $E \overset{t}{\Longrightarrow}_{\langle r \rangle} E'$. □

We define weak level-$\langle r \rangle$ equivalence by using *level-*$\langle r \rangle$ *bisimulations.*

Definition 3.6 *Let* $r \in \mathcal{R}$*. A binary relation* $\mathcal{S} \subseteq \mathcal{P} \times \mathcal{P}$ *over processes is a* level-$\langle r \rangle$ bisimulation *if* $(P,Q) \in \mathcal{S}$ *implies, for all* $\mu \in Act_\tau$*, that*

(i) whenever $P \xrightarrow{\mu} P'$ *then, for some* Q'*,* $Q \overset{\phi(\mu,r)}{\Longrightarrow}_{\langle r \rangle} Q'$*, and* $(P',Q') \in \mathcal{S}$*,*

(ii) whenever $Q \xrightarrow{\mu} Q'$ *then, for some* P'*,* $P \overset{\phi(\mu,r)}{\Longrightarrow}_{\langle r \rangle} P'$*, and* $(P',Q') \in \mathcal{S}$*.* □

Definition 3.7 P *and* Q *are* weakly level-$\langle r \rangle$ equivalent*, written* $P \approx_{\langle r \rangle} Q$*, if* $(P,Q) \in \mathcal{S}$ *for some level-*$\langle r \rangle$ *bisimulation* $\mathcal{S}$*.* □

Notice that ϕ is used on $\Longrightarrow_{\langle r \rangle}$ in Definition 3.6, because ambiguous actions can be observed. Proposition 3.5 shows the basic properties of $\approx_{\langle r \rangle}$.

Proposition 3.5 (1) $\approx_{\langle r \rangle}$ *is an equivalence relation.*
(2) *If* $r \geq r'$*, then* $\approx_{\langle r \rangle} \subseteq \approx_{\langle r' \rangle}$*.* □

If the level is high enough that no action is neglected, then unobservable actions are only τ. Particularly $\approx_{\langle \infty \rangle}$ corresponds to weak equivalence $\approx$ defined in [2].

$\approx_{\langle r \rangle}$ is preserved by Composition combinator $|$, and conditionally preserved by Restriction combinator $\backslash$ as shown in Proposition 3.6.

Proposition 3.6 *Let* $P_1 \approx_{\langle r \rangle} P_2$*. Then*

(1) $P_1|Q \approx_{\langle r \rangle} P_2|Q$
(2) $P_1\backslash^L_{\langle r' \rangle(s')} \approx_{\langle r \rangle} P_2\backslash^L_{\langle r' \rangle(s')}$ *if* $r' + \pi(s') \leq r$
(3) $P_1@s' \approx_{\langle r' \rangle} P_2@s'$ *if* $r' + \pi(s') \leq r$ □

Intuitively the condition of (2) shows that the restriction area, inside the circle whose center is $\pi(s')$ away from the observer and whose radius is r', must not be overlapped on the unobservable area, outside the circle whose radius is r.

3.3 Level-$\langle r \rangle$ Equivalence

Weak level-$\langle r \rangle$ equivalence $\approx_{\langle r \rangle}$ is not preserved by Choice combinator $+$ like weak equivalence $\approx$. In this subsection, we define a relation called *level-*$\langle r \rangle$ *equivalence* preserved by $+$. First, a binary relation over actions is defined.

Definition 3.8 *Let* $r \in \mathcal{R}$*. Level-*$\langle r \rangle$ *substitution* $\trianglerighteq_{\langle r \rangle} (\subset Act_\tau \times Act_\tau)$ *is a binary relation over actions defined by*

$$\trianglerighteq_{\langle r \rangle} = \{(\mu,\mu) : \mu \in Act_\tau\} \cup \{(\mu,\mu') : \mu,\mu' \in Act_\tau, \phi(\mu,r) = \theta(\mu',r) = \varepsilon\}$$ □

$(\mu \trianglerighteq_{\langle r \rangle} \mu')$ implies that μ' can be substituted for μ in level-$\langle r \rangle$ observation. For example the following relations show properties of $\trianglerighteq_{\langle r \rangle}$.

$$\tau \trianglerighteq_{\langle 1 \rangle} (\alpha\langle -2 \rangle@\varepsilon), \quad (\alpha\langle -2 \rangle@\varepsilon) \trianglerighteq_{\langle 1 \rangle} \tau, \quad \tau \trianglerighteq_{\langle 1 \rangle} (\alpha\langle 2 \rangle@\varepsilon), \quad (\alpha\langle 2 \rangle@\varepsilon) \not\trianglerighteq_{\langle 1 \rangle} \tau$$

In level-$\langle 1\rangle$ observation, $(\alpha\langle -2\rangle@\varepsilon)$ is unobservable and $(\alpha\langle 2\rangle@\varepsilon)$ is ambiguous. Unobservable actions correspond to internal actions τ, while ambiguous actions do not correspond to τ. Ambiguous actions can be substituted for internal actions, but internal actions can not be substituted for ambiguous actions.

Then we define level-$\langle r\rangle$ equivalence.

Definition 3.9 *Let $r \in \mathcal{R}$. P and Q are* level-$\langle r\rangle$ equivalent, *written $P =_{\langle r\rangle} Q$, if for all $\mu \in Act_\tau$, that*

(i) whenever $P \xrightarrow{\mu} P'$ then, for some (Q',μ'), $Q \overset{\mu'}{\Longrightarrow}_{\langle r\rangle} Q'$,$P' \approx_{\langle r\rangle} Q'$,$\mu \trianglerighteq_{\langle r\rangle} \mu'$,

(ii) whenever $Q \xrightarrow{\mu} Q'$ then, for some (P',μ'), $P \overset{\mu'}{\Longrightarrow}_{\langle r\rangle} P'$,$P' \approx_{\langle r\rangle} Q'$,$\mu \trianglerighteq_{\langle r\rangle} \mu'$ □

For $=_{\langle r\rangle}$, each *initial* action must be matched by a substitutive action unlike $\approx_{\langle r\rangle}$. Particularly $=_{\langle\infty\rangle}$ corresponds to observation congruence. Proposition 3.7 show that $=_{\langle r\rangle}$ is the largest relation preserved by $+$ and included in $\approx_{\langle r\rangle}$.

Proposition 3.7 *(A characterization of $=_{\langle r\rangle}$)*

1. *If $P_1 =_{\langle r\rangle} P_2$, then $P_1 + R =_{\langle r\rangle} P_2 + R$, for any R.*
2. *Let $\mathcal{Q} \subseteq \approx_{\langle r\rangle}$ such that $(P_1 + R, P_2 + R) \in \mathcal{Q}$ for any R, if $(P_1, P_2) \in \mathcal{Q}$. Then if $(P_1, P_2) \in \mathcal{Q}$ and $\mathcal{L}(P_1) \cup \mathcal{L}(P_2) \neq \mathcal{A}$ [1], then $P_1 =_{\langle r\rangle} P_2$.*

Proof *We show only a proof of 2. We choose that R is $A \overset{\text{def}}{=} (a_0\langle r_0\rangle@\varepsilon).A$ such as $a_0 \notin \mathcal{L}(P_1)\cup\mathcal{L}(P_2)$ and $r_0 \geq -r$. Let $P_1 \xrightarrow{\mu} P'$. By* **Choice$_1$**, *$P_1 + A \xrightarrow{\mu} P'$. Since $P_1 + A \approx_{\langle r\rangle} P_2 + A$, for some Q', $P_2 + A \overset{\phi(\mu,r)}{\Longrightarrow}_{\langle r\rangle} Q'$ and $P' \approx_{\langle r\rangle} Q'$.*

If μ is $(a_1\langle r_1\rangle@s_1)$ such as $(r_1 \geq \pi(s_1)-r)$, then $\phi(a_1\langle r_1\rangle@s_1, r) = a_1\langle r_1\rangle@s_1$. In this case, we easily obtain that $P_2 \overset{\mu}{\Longrightarrow}_{\langle r\rangle} Q'$, $P' \approx_{\langle r\rangle} Q'$, and $\mu \trianglerighteq_{\langle r\rangle} \mu$.

Otherwise, $\phi(\mu, r) = \varepsilon$. Therefore, $P_2 + A \Longrightarrow_{\langle r\rangle} Q'$. Now we show that $Q' \not\equiv P_2 + A$ by inconsistency. Suppose that $Q' \equiv P_2 + A$. In this case Q' has $(a_0\langle r_0\rangle@\varepsilon)$-derivations, because $\phi(a_0\langle r_0\rangle@\varepsilon, r) = a_0\langle r_0\rangle@\varepsilon$ since $r_0 \geq -r$. Thus P' must also have $(a_0\langle r_0\rangle@\varepsilon)$-derivations, since $P' \approx_{\langle r\rangle} Q'$, but it is impossible, because $a_0 \notin \mathcal{L}(P') \subseteq \mathcal{L}(P)$. Hence $Q' \not\equiv P_2 + A$, namely, $P_2 + A \overset{\mu'}{\Longrightarrow}_{\langle r\rangle} Q'$ for some μ' such as $\theta(\mu', r) = \varepsilon$. This transition must be caused by P_2, because P' has no $(a_0\langle r_0\rangle@\varepsilon)$-derivation. Hence $P_2 \overset{\mu'}{\Longrightarrow}_{\langle r\rangle} Q'$ and $\mu \trianglerighteq_{\langle r\rangle} \mu'$, since $\phi(\mu, r) = \varepsilon = \theta(\mu', r)$. □

$\approx_{\langle r\rangle}$ is not a congruence relation, because it is not always preserved by Restriction and Route combinators. Proposition 3.6 for $\approx_{\langle r\rangle}$ also holds for $=_{\langle r\rangle}$.

3.4 Axiom System $\mathcal{A}\langle r\rangle$

In order to compare level-$\langle r\rangle$ equivalence $=_{\langle r\rangle}$ and observation congruence $=$, we give an axiom system $\mathcal{A}\langle r\rangle$ for finite sequential processes which consist only of Inaction '**0**', Prefix '.', and Choice '+'. The set of finite sequential processes is denoted by $\mathcal{P}_{seq}(\subset \mathcal{P})$, and is ranged over by $P, Q, \cdots$.

$=_{\langle r\rangle}$ is a congruence relation for $\mathcal{P}_{seq}$, because it is preserved by Prefix and Choice combinators. A sound and complete axiom system $\mathcal{A}_\infty$ for observation congruence of $\mathcal{P}_{seq}$ has already been given in [2] as follows.

[1] $\mathcal{L}(P)$ is the set of labels of all actions which P can perform in the future.

Definition 3.10 *We write* $\mathcal{A}_\infty \vdash P = Q$ *if the equality of two processes* P *and* Q *can be proven by equational reasoning from the axiom system* $\mathcal{A}_\infty$*, which consists of the following equations:*

M1 $P_1 + P_2 = P_2 + P_1$	**T1** $\mu.\tau.P = \mu.P$
M2 $(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$	**T2** $P + \tau.P = \tau.P$
M3 $P = P + P$	**T3** $\mu.(P + \tau.Q) + \mu.Q = \mu.(P + \tau.Q)$
M4 $P = P + \mathbf{0}$	

□

Theorem 3.8 *Let* $P, Q \in \mathcal{P}_{seq}$*. Then* $P = Q$ **iff** $\mathcal{A}_\infty \vdash P = Q$*.* □

We define an axiom system $\mathcal{A}\langle r\rangle$ for any $r \in \mathcal{R}$ as follows.

Definition 3.11 *Let* $r \in \mathcal{R}$*. We write* $\mathcal{A}\langle r\rangle \vdash P = Q$ *if the equality of two processes* P *and* Q *can be proven by equational reasoning from the axiom system* $\mathcal{A}\langle r\rangle$*, which consists of the equations in* $\mathcal{A}_\infty$ *and the following equations:*

$$\mathbf{A1}\langle r\rangle\ (\alpha\langle r'\rangle@s).P = \tau.P \qquad \text{if } r' < -(r - \pi(s))$$
$$\mathbf{A2}\langle r\rangle\ (\alpha\langle r'\rangle@s).P = (\alpha\langle r'\rangle@s).P + \tau.P \qquad \text{if } r' > r - \pi(s)$$
□

A1$\langle r\rangle$ and **A2**$\langle r\rangle$ are equations for unobservable actions and ambiguous actions, respectively. We define a standard form to prove completeness of $\mathcal{A}\langle r\rangle$.

Definition 3.12 P *is a* level-$\langle r\rangle$ standard form*, or is in level-*$\langle r\rangle$ *standard form, if* [2]
(*i*) $P \equiv \sum_{i=1}^{m} \mu_i.P_i$ *where each* P_i *is also in level-*$\langle r\rangle$ *standard form,*
(*ii*) $P \overset{a\langle r'\rangle@s}{\not\longrightarrow}$ *such as* $r' < -(r - \pi(s))$*,*
(*iii*) *whenever* $P \xrightarrow{a\langle r'\rangle@s} P'$ *such as* $r' > r - \pi(s)$*, then* $P \xrightarrow{\tau} P'$*.* □

(*ii*) means that all unobservable actions except τ can not occur. (*iii*) means that all ambiguous actions must be bypassed through τ.

Proposition 3.9 strengthens relations between processes.

Proposition 3.9 *Let* P *and* Q *be in level-*$\langle r\rangle$ *standard form. Then,*

$$P \approx_{\langle r\rangle} Q \text{ implies } P \approx Q, \text{ and } P =_{\langle r\rangle} Q \text{ implies } P = Q.$$

Proof *(Key points) Let* P *be in level-*$\langle r\rangle$ *standard form. It is important to prove that if* $P \xrightarrow{\mu} P'$ *and* $\theta(\mu, r) = \varepsilon$ *then* $P \xrightarrow{\tau} P'$*.* $\theta(\mu, r) = \varepsilon$ *implies* $\mu = \tau$ *or* $\mu = (\alpha\langle r'\rangle@s')$ *such as* $|r'| > \pi(s') - r$*. Then,* $P \xrightarrow{\tau} P'$ *is easily obtained by the conditions* (*ii*) *and* (*iii*) *of level-*$\langle r\rangle$ *standard form.* □

Lemma 3.10 is used for the proof of completeness of $\mathcal{A}\langle r\rangle$ for $=_{\langle r\rangle}$ of $\mathcal{P}_{seq}$.

Lemma 3.10 *For any* $P \in \mathcal{P}_{seq}$*, there is a level-*$\langle r\rangle$ *standard form* P' *of equal depth, such that* $\mathcal{A}\langle r\rangle \vdash P = P'$*.*
Proof (*ii*) *and* (*iii*) *are satisfied by* **A1**$\langle r\rangle$ *and* **A2**$\langle r\rangle$*, respectively.* □

Finally we give Theorem 3.11 which shows that $\mathcal{A}\langle r\rangle$ is sound and complete for level-$\langle r\rangle$ equivalence of finite sequential processes.

Theorem 3.11 *Let* $P, Q \in \mathcal{P}_{seq}$*. Then* $P =_{\langle r\rangle} Q$ **iff** $\mathcal{A}\langle r\rangle \vdash P = Q$*.*
Proof ($\Leftarrow$) *A level-*$\langle r\rangle$ *bisimulation for each equation can be found.*
($\Rightarrow$) *By Lemma 3.10, Proposition 3.9, and Theorem 3.8.* □

[2] If $m \geq 1$, then $\sum_{i=1}^{m} P_i$ is the short notation of $P_1 + P_2 + \cdots + P_m$, otherwise it is $\mathbf{0}$.

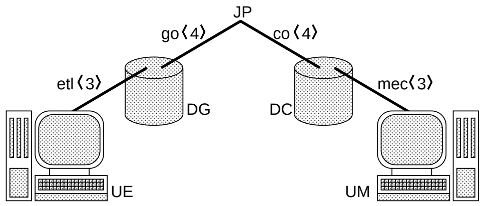


Fig. 5. A communicating system with deadlocks

3.5 Strong Level-$\langle r\rangle$ Equivalence

In [2] *strong* equivalence $\sim$ is considered before *weak* equivalence $\approx$, because $\sim$ is simpler than $\approx$. In this subsection we discuss a *strong version* $\sim_{\langle r\rangle}$ of $\approx_{\langle r\rangle}$.

Strong level-$\langle r\rangle$ bisimulation may be defined by using the condition that

(i) whenever $P \xrightarrow{\mu} P'$ then, for some (Q', μ'), $Q \xrightarrow{\mu'} Q'$, $(P', Q') \in \mathcal{S}$, $\mu \trianglerighteq_{\langle r\rangle} \mu'$,

instead of (i) in Definition 3.6, and (ii) is symmetric. $P \sim_{\langle r\rangle} Q$ implies $P \approx_{\langle r\rangle} Q$, and a sound and complete axiom system for $\sim_{\langle r\rangle}$ of $\mathcal{P}_{seq}$ is given from **M1**-**M4**, **A1**$\langle r\rangle$, **A2**$\langle r\rangle$. Unfortunately, $\sim_{\langle r\rangle}$ is not preserved by Composition $|$. For example, consider the following three processes: $P_1 \equiv \alpha\langle -2\rangle.\mathbf{0}$, $P_2 \equiv \tau.\mathbf{0}$, $P_3 \equiv \alpha\langle 2\rangle.\mathbf{0}$. P_1 and P_2 are strongly level-$\langle 1\rangle$ equivalent, because $\alpha\langle -2\rangle$ is unobservable, but $P_1|P_3$ and $P_2|P_3$ are *not* strongly level-$\langle 1\rangle$ equivalent, because $P_1|P_3$ can reach a stop process through an internal action τ by $\mathbf{Com_3}$, while $P_2|P_3$ can not do so.

4 An Example of Approximative Analysis

We show an example of approximative analysis in CCSG by using the system in Fig.5. DG and DC are databases of government and corporations, respectively. UE and UM are interfaces of the national laboratory ETL and the corporation MEC, respectively. They are connected through the four routers as shown in Fig.5. This system is described by an observer at JP as follows:

$$SYS \stackrel{\text{def}}{=} ((GO@go\langle 4\rangle)|(CO@co\langle 4\rangle))\backslash^{L}_{\langle 18\rangle(\varepsilon)} \qquad GO \stackrel{\text{def}}{=} DG|(UE@etl\langle 3\rangle)$$
$$L = \{lk_1, lk_2, ul_1, ul_2\} \qquad CO \stackrel{\text{def}}{=} DC|(UM@mec\langle 3\rangle)$$

We assume that locks are needed for access to databases, and each interface trys to lock the near database at first and another one after that, when it accepts the action ac_i. Each component process is described as follows:

$$UE \stackrel{\text{def}}{=} ac_1.\overline{lk_1}\langle 3\rangle.\overline{lk_2}\langle 11\rangle.su_1.\overline{ul_2}\langle 11\rangle.\overline{ul_1}\langle 3\rangle.UE \qquad DG \stackrel{\text{def}}{=} lk_1.ul_1.DG$$
$$UM \stackrel{\text{def}}{=} ac_2.\overline{lk_2}\langle 3\rangle.\overline{lk_1}\langle 11\rangle.su_2.\overline{ul_1}\langle 11\rangle.\overline{ul_2}\langle 3\rangle.UM \qquad DC \stackrel{\text{def}}{=} lk_2.ul_2.DC$$

where the empty route ε and the zero grade $\langle 0\rangle$ are omitted. su_i is used to inform success of locking. The grades of ac and su are set to 0, because they are local at their interfaces. lk and ul are used for locking and unlocking, respectively. lk and ul of interfaces have grades high enough to synchronize with databases. For example, the grade 11 of $\overline{lk_2}\langle 11\rangle$ is the loss distance between UE and DC. These lk and ul are restricted from environment. The restriction power 18 of SYS is the minimal to restrict $\overline{lk_i}\langle 11\rangle$ and $\overline{ul_i}\langle 11\rangle$ the loss distance 7 away from JP.

In order to understand the behavior of SYS, we give the sequential process SP which is observation congruent to SYS, written $SYS = SP$, as follows:

$$SP \stackrel{\text{def}}{=} ac_1@s_1.R_1 + ac_2@s_2.R_2, \qquad s_1 = etl\langle 3\rangle go\langle 4\rangle, \quad s_2 = mec\langle 3\rangle co\langle 4\rangle$$
$$R_1 \stackrel{\text{def}}{=} \tau.(\tau.(su_1@s_1.SP + ac_2@s_2.su_1@s_1.R_2) + ac_2@s_2.O_{12}) + ac_2@s_2.O$$
$$R_2 \stackrel{\text{def}}{=} \tau.(\tau.(su_2@s_2.SP + ac_1@s_1.su_2@s_2.R_1) + ac_1@s_1.O_{21}) + ac_1@s_1.O$$
$$O \stackrel{\text{def}}{=} \tau.O_{12} + \tau.O_{21}, \quad O_{12} \stackrel{\text{def}}{=} \tau.su_1@s_1.R_2 + \tau.\mathbf{0}, \quad O_{21} \stackrel{\text{def}}{=} \tau.su_2@s_2.R_1 + \tau.\mathbf{0}$$

SP *explicitly* shows that SYS has deadlocks. Although SP has useful information, but it is somewhat complex even for the simple example SYS.

We often stay in ETL and are interested only in the situation near ETL. Thus actions of UM are unobservable. First, the position of the observer is shifted from JP to ETL by $SYS \precsim_{(s)} SYS@s$, where s is the route $(go\langle 4\rangle etl\langle 3\rangle)$ from JP to ETL. Let $SP_{ETL} \stackrel{\text{def}}{=} ac_1.(\tau.su_1.SP_{ETL} + \tau.\mathbf{0})$, then the equation

$$SYS \precsim_{(s)} SYS@s =_{\langle r\rangle} \tau.SP_{ETL} \approx_{\langle r\rangle} SP_{ETL}$$

holds, where r is less than 14 which is the loss distance between ETL and MEC. The τ of $\tau.SP_{ETL}$ is needed for matching unobservable actions in MEC. SP_{ETL} shows that SYS may fall to a deadlock after ac_1 while it never falls just after su_1.

5 Related Work

Several process algebras considering space have already been proposed, for example [3, 4, 5] as extensions of CCS and [7] as an extension of ACP [6].

An advantage of [7] is that time and space are integrated. For example, the possibility of communication between distributed processes can be checked by considering the velocity of communication. The main purpose of CCSG is approximative analysis, while [7] is not interested in such analysis. Although CCSG has no notion of time yet, we are interested in introducing the notion of time [8] to CCSG. The velocity of communication may be expressed by routers with *delay*.

In [3, 4, 5], equality of processes is checked by considering locations of actions. For example, $P_1 \equiv (\alpha_1.\mathbf{0}|\alpha_2.\mathbf{0})$ and $P_2 \equiv (\alpha_1.\alpha_2.\mathbf{0} + \alpha_2.\alpha_1.\mathbf{0})$ are not *location equivalent* [3], because the locations of α_1 and α_2 are independent of each other in P_1, while they are dependent in P_2, as shown in the *location transitions*:

$$P_1 \xrightarrow[u_1]{\alpha_1} (u_1 :: \mathbf{0})|(\alpha_2.\mathbf{0}) \xrightarrow[u_2]{\alpha_2} (u_1 :: \mathbf{0})|(u_2 :: \mathbf{0}),$$
$$P_2 \xrightarrow[u_1]{\alpha_1} u_1 :: \alpha_2.\mathbf{0} \xrightarrow[u_1 u_2]{\alpha_2} u_1 u_2 :: \mathbf{0}.$$

where u of $\xrightarrow[u]{\alpha}$ represents the location of α. In the transitions of P_2, the location u_1u_2 of α_2 depends on u_1 of α_1. Location transitions *automatically* assign locations. Thus, concurrent processes are distinct from sequential processes.

In CCSG positions of actions are *explicitly* described, and concurrent processes are not always distinct from sequential processes. For example, the following processes P'_1 and P'_2 are level-$\langle\infty\rangle$ equivalent (i.e. observation congruent).

$$P'_1 \equiv (\alpha_1.\mathbf{0})@s_1|(\alpha_2.\mathbf{0})@s_2, \qquad P'_2 \equiv (\alpha_1@s_1).(\alpha_2@s_2).\mathbf{0} + (\alpha_2@s_2).(\alpha_1@s_1).\mathbf{0}$$

It is not recommended to apply location transitions into CCSG, because locations automatically assigned to actions by location transitions may be inconsistent with the explicitly described positions. The purpose of location equivalence is different from ours. We introduce the positions for estimating the loss distance.

6 Conclusion

We have proposed CCSG by introducing *grades* and *routes* to CCS. The grades represent the importance of actions and the routes point to positions where actions occur. An advantage of CCSG is to approximately analyze systems under assumption that unimportant distant actions can not be observed. We have given an approximative equivalence relation called level-$\langle r \rangle$ equivalence. The difference of level-$\langle r \rangle$ equivalence from observation congruence is shown by $\mathbf{A1}\langle r \rangle$ and $\mathbf{A2}\langle r \rangle$ in the axiom system $\mathcal{A}\langle r \rangle$.

The most interesting and urgent future work is to modify the connection of routers to *graph structure* from the hierarchical star structure.

MBone [11] is known as a communication style where a value assigned to each message restricts the receivable area of the message. In MBone each router has a threshold, and messages with values less than the threshold can not pass the router. Although CCSG hsa a similar communication style to MBone, CCSG has point-to-point communication, while MBone has broadcast communication. CB-S [9] and CCB [10] have already been proposed as process algebra with broadcast communication. We want to extend CCSG with broadcast communication.

Acknowledgement

The authors wish to express our gratitude to Dr. Kimihiro Ohta, Director of Computer Science Division, ETL. They also thank Dr. Yoshiki Kinoshita in Computer Language Section and all our colleagues in Information Base Section for their helpful discussions.

References

1. M.Tsujigao, T.Hikita, and J.Ginbayashi, "Process Composition and Interleave Reduction in Parallel Process Specification", Proc. of JCSE'93, pp.218-225, 1993.
2. R.Milner, "*Communication and Concurrency*", Prentice-Hall, 1989.
3. G.Boudol, I.Castellani, M.Hennessy, and A.Kiehn, "Observing localities", *Theoretical Computer Science*, Vol.114, pp.31-61, 1993.
4. P.Krishnan,"Distributed CCS", CONCUR'91, LNCS 527, Springer-Verlag, pp.393-407, 1991
5. Ugo Montanari and Daniel Yankelevich, "A Parametric Approach to Localities", ICALP'92, LNCS 623, Springer-Verlag, pp.617 - 628, 1992
6. J.C.M.Baeten and W.P.Weijland, "*Process Algebra*", Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990.
7. J.C.M.Baeten and J.A.Bergstra, "Real Space Process Algebra", CONCUR'91, LNCS 527, Springer-Verlag, pp.96 - 110, 1991
8. F.Moller and C.Tofts,"An overview of TCCS", Proc. of EUROMICRO'92, 1992.
9. K.V.S.Prasad, "A Calculus of Broadcasting Systems", TAPSOFT'91, Vol.1:CAAP, LNCS 493, Springer-Verlag, pp.338 - 358, 1991
10. Y.Isobe, Y.Sato, and K.Ohmaki, "A Calculus of Countable Broadcasting Systems", AMAST'95, LNCS 936, Springer-Verlag, pp.489-503, 1995.
11. "The MBONE Information Web", `http://www.eit.com/techinfo/mbone/mbone.html`.

This article was processed using the LaTeX macro package with LLNCS style