π計算に対する時間拡張と代数的意味論

An Algebraic Theory for a Timed Extension of the π-Calculus

桑原 寛明*　結縁 祥治†　阿草 清滋‡

Summary. In this paper, we propose an extension of the π-calculus with time called *timed π-calculus*. We define the syntax, semantics and bisimilarities of timed π-calculus. These bisimilarities are equivalence relations and non-input congruence. Our bisimilarities mean each process has same input/output sequences and waits for an equal time length simultaneously. Additionally, we propose preorders that represent which process in first comes to be able to execute actions after some time passage actions. These relations can be used to analyze and verify the behavioral properties of real-time systems. We present the expressiveness of timed π-calculus and a method for verification using an example.

1　はじめに

実時間システムは動作に時間制約を持つシステムであり，与えられた時間制約を満たす必要がある．システムの動作の正しさは，計算結果の正しさだけではなく結果が得られるまでに経過した時間の長さにも依存する．動作にかかる時間はメッセージ通信や並行性など動的な要因に影響される．そのため，システムを実際に動作させてテストするだけで常に正しく動作すると確認することは難しい．形式的な検証手法を用いてシステムの動作の正しさを示すことが必要である．

この問題に対し，我々は実時間システムを構成するソフトウェアの抽象的な形式モデルを得るための手法として，リアルタイムオブジェクト指向言語からπ計算による記述への変換規則を提案した [9]．時間に関する性質を表現するために時間概念を導入してπ計算を拡張した．そこでは，π計算の構文と動作意味を時間拡張し，時間付き双模倣関係を定義している．しかし，時間付き双模倣関係が持つ代数的な性質についてはほとんど示されていない．

本稿では，時間付き双模倣関係が入力プレフィックス以外のコンテキストに対して合同性を満たすことを示す．すべての代入に対して双模倣性を保つ full bisimilarity を定義し，合同であることを示す．時間付き双模倣関係は時間待ちが発生するタイミングとその長さが等しいことを要求するため，実装が時間に関する仕様を満たすか否かの検証に利用するには条件が厳しい．そこで，遅延時間順関係と呼ばれる時間待ちの長さの違いを反映する順序関係を与える．この順序関係が限られたコンテキストに対して合同性を満たすことを示す．時間付き双模倣関係にある 2 つのプロセスは，互いに交換してもそれらのプロセスを部分要素として構成されるシステムの振舞いに関する性質は変化しない．2 つのプロセス間に時間付き双模倣関係や遅延時間順関係があるか調べる場合，これらの関係は合同性を満たすため 2 つのプロ

*Hiroaki Kuwabara, 名古屋大学大学院情報科学研究科

†Shoji Yuen, 名古屋大学大学院情報科学研究科/科学技術振興機構さきがけ研究 21

‡Kiyoshi Agusa, 名古屋大学大学院情報科学研究科

セスに共通するコンテキストを除外して調べることができる．このことは振舞いの解析や検証に有用である．

本稿の構成は以下の通りである．2 節で時間付き π 計算の構文と動作意味を定義する．3 節で双模倣関係を定義し，制限されたコンテキストに対する合同性を示す．4 節で順序関係を定義し，5 節で時間付き π 計算による記述と順序関係の例を挙げる．6 節で関連研究に触れ，最後にまとめと今後の課題を述べる．

2 時間付き π 計算

π 計算 [6] [7] [8] は名前通信に基づいて計算を表す並行計算モデルである．プロセス間通信に利用するリンク自身をメッセージとする通信が表現可能であり，高い表現能力を持っている．時間に関する性質を表現するため，離散時間の経過を表すプリミティブを導入し，選択演算子を用いてタイムアウトをモデル化する．

2.1 構文

π 計算に自然数によって添字付けされたプレフィックス t を導入する．m 単位時間でタイムアウトする時間待ちを $t[\underline{m}]$ と記述し，時間経過アクションと呼ぶ．t は名前ではないとする．

$\mathcal{N}ame$ を名前の集合，$\mathcal{I}$ を自然数を表す名前の集合とする．$\mathcal{I} = \{\underline{0}, \underline{1}, \ldots\} \subset \mathcal{N}ame$ とする．ここで $\underline{i}$ は自然数 i を表す名前である．さらに $\mathcal{N} = \mathcal{N}ame - \mathcal{I}$，$\overline{\mathcal{N}} = \{\overline{x} | x \in \mathcal{N}\}$，$\mathcal{L} = \mathcal{N} \cup \overline{\mathcal{N}}$ とする．$\widetilde{x}$ は名前のリストを表し，i 番目の要素を x_i と書く．π 計算のプロセス全体の集合を $\mathcal{P}$，アクションの集合を $Act = \mathcal{L} \cup \{\tau\}$ と書く．以下では，$x \in \mathcal{N}$，$m \in \mathcal{I}$，$n \in \mathcal{N}ame$，k を自然数とし，リスト $\widetilde{y}$ はすべての i について $y_i \in \mathcal{N}$ を満たし，$\widetilde{z}$ はすべての i について $z_i \in \mathcal{N}ame$ を満たすとする．

定義 1 時間付き π 計算のプロセス式 P は以下の構文によって定義される．

$$\begin{array}{rcl} \pi & ::= & x(\widetilde{y}) \mid \overline{x}\langle\widetilde{z}\rangle \mid \tau \mid t[n] \\ P & ::= & M \mid P_1|P_2 \mid \boldsymbol{\nu} x\, P \mid !\,P \\ M & ::= & \mathbf{0} \mid \pi.P \mid M_1 + M_2 \end{array}$$

ただし P 中に出現する入出力アクションのサブジェクトの集合と，時間経過アクションの時間長を表す名前の集合は互いに素であるとする．また $!P$ における P は以下で定義されるアクションガードつきプロセスである．

$\pi ::= x(\widetilde{y}) \mid \overline{x}\langle\widetilde{z}\rangle \mid \tau$ とする時

- 任意のプロセス P に対し $\pi.P$ はアクションガードつきプロセスである
- P_1, P_2 がアクションガードつきプロセスである時

$$P_1|P_2,\ P_1 + P_2,\ \boldsymbol{\nu} a\, P_1,\ !\,P_1$$

はアクションガードつきプロセスである □

定義 2 $P = t[x].P'$ に対し，$x \in \mathcal{N}$ かつ x が P を内部に含むプロセスにおいて入力アクションによって束縛されていない場合，P は動作不能であるといい $P{\uparrow}$ と書く．また $P{\uparrow}$ の時，

$$\begin{array}{c} (x(\widetilde{y}).P){\uparrow}, (\overline{x}\langle\widetilde{z}\rangle.P){\uparrow}, (\tau.P){\uparrow}, (t[n].P){\uparrow}, (P \mid Q){\uparrow}, \\ (Q \mid P){\uparrow}, (P + Q){\uparrow}, (Q + P){\uparrow}, (\boldsymbol{\nu} z\, P){\uparrow}, (!\,P){\uparrow} \end{array}$$

とする．動作不能でない場合，動作可能であるといい $P\,\not\!\downarrow$ と書く． □

定義 3 プロセス $x(z).P$ 及び $\boldsymbol{\nu} z\, P$ において名前 z のスコープは P に制限される．この時，名前 z は束縛されるという．束縛されない名前を自由であるという．プロセス P に含まれる自由な名前の集合を $\mathsf{fn}(P)$，束縛される名前の集合を $\mathsf{bn}(P)$ と書く．アクション α の自由な名前の集合，束縛される名前の集合は以下のように定義される．

α	$x(y)$	$\overline{x}\langle z\rangle$	τ	$t[n]$
$\mathsf{fn}(\alpha)$	$\{x,y\}$	$\{x\}$	$\emptyset$	$\{n\}$
$\mathsf{bn}(\alpha)$	$\emptyset$	$\{z\}$	$\emptyset$	$\emptyset$

□

定義 4 代入は $\mathcal{N}$ から $\mathcal{N}ame$ への関数である．プロセス P に代入 σ を適用したプロセス $P\sigma$ を以下のように定義する．ここで $x\sigma$ は名前 x に σ を適用して得られる名前である．

$$
\begin{array}{rcl@{\qquad}rcl}
\mathbf{0}\sigma & \stackrel{def}{=} & \mathbf{0} & (P \mid Q)\sigma & \stackrel{def}{=} & P\sigma \mid Q\sigma \\
(x(\widetilde{y}).P)\sigma & \stackrel{def}{=} & x\sigma(\widetilde{y}).P\sigma & (P+Q)\sigma & \stackrel{def}{=} & P\sigma + Q\sigma \\
(\overline{x}\langle\widetilde{z}\rangle.P)\sigma & \stackrel{def}{=} & \overline{x\sigma}\langle\widetilde{z\sigma}\rangle.P\sigma & (\boldsymbol{\nu} x\, P)\sigma & \stackrel{def}{=} & \boldsymbol{\nu} x\, P\sigma \\
(\tau.P)\sigma & \stackrel{def}{=} & \tau.P\sigma & (!\, P)\sigma & \stackrel{def}{=} & !\, P\sigma \\
(t[n].P)\sigma & \stackrel{def}{=} & t[n\sigma].P\sigma & & &
\end{array}
$$

ただし $x\sigma \in \mathcal{N}$ でなければならないとする．また，代入の適用によって束縛されている名前が出現する場合は重複しないよう適切に名前替えが行われるとする．代入 σ を $\{y_1,\ldots,y_n/x_1,\ldots,x_n\}$ とも書く．これは x_i に対する y_i の代入を表す． □

時間経過アクション $t[i]$ は i 単位時間待機することを表す．例えばプロセス $t[\underline{10}].P$ は 10 単位時間待機した後にプロセス P として振舞う．一定時間以内にイベントが発生したか否かでその後の動作が異なる典型的なタイムアウトは，$a.P + t[\underline{5}].\tau.Q$ のように時間経過アクション，非決定選択，τ 動作を用いて記述できる．このプロセスはアクション a の発生を最大で 5 単位時間待機する．4 単位時間以内に a を実行すれば P へ遷移する．5 単位時間経過した時に a が実行可能であれば P か Q へ非決定的に，不可能であればタイムアウトし Q へ遷移する．

他のプレフィックスと演算子の直観的な意味を以下に示す．

- （入力）$x(\widetilde{y})$：x を通して $\widetilde{y}$ を受信
- （出力）$\overline{x}\langle\widetilde{z}\rangle$：$x$ を通して $\widetilde{z}$ を送信
- （τ 動作） τ：外部からは不可視な内部アクション
- （選択）$P+Q$：プロセス P,Q のうち実行可能なプロセスを選択し実行する．いずれのプロセスも実行可能であれば一方を非決定的に選択する
- （並行）$P|Q$：プロセス P,Q が並行に動作する
- （制限）$\boldsymbol{\nu} x\, P$：プロセス P 中の自由な変数 x を束縛する
- （複製）$!P$：無限個のプロセス P が $\mid$ 演算子によって結合しているとみなす．

2.2 時間動作意味

$\mathcal{P}$ 上の遷移関係 $\{\stackrel{\alpha}{\longrightarrow} \mid \alpha \in Act\} \cup \rightsquigarrow$ を図 1 の遷移規則及び図 2 の時間経過規則によって定義する．ABORT 規則以外のすべての規則において $P\!\not\downarrow, Q\!\not\downarrow$ とする．また図 1 においてリスト $\widetilde{w}$ の要素はすべて $\mathcal{N}ame$ に含まれるとする．

$$\text{OUT}: \frac{}{\overline{x}\langle\tilde{z}\rangle.P \xrightarrow{\overline{x}\langle\tilde{z}\rangle} P} \qquad \text{INPUT}: \frac{}{x(\tilde{y}).P \xrightarrow{x(\tilde{w})} P\{\tilde{w}/\tilde{y}\}} \qquad \text{TAU}: \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\text{SUM-L}: \frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'} \qquad \text{SUM-R}: \frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$$

$$\text{PAR-L}: \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}\ \mathsf{bn}(\alpha) \cap \mathsf{fn}(Q) = \emptyset$$

$$\text{PAR-R}: \frac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'}\ \mathsf{bn}(\alpha) \cap \mathsf{fn}(P) = \emptyset$$

$$\text{COMM-L}: \frac{P \xrightarrow{\overline{x}\langle\tilde{y}\rangle} P' \quad Q \xrightarrow{x(\tilde{y})} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \qquad \text{COMM-R}: \frac{P \xrightarrow{x(\tilde{y})} P' \quad Q \xrightarrow{\overline{x}\langle\tilde{y}\rangle} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\text{RES}: \frac{P \xrightarrow{\alpha} P'}{\boldsymbol{\nu} x\, P \xrightarrow{\alpha} \boldsymbol{\nu} x\, P'}\ \text{if } \alpha \notin \{x, \overline{x}\}$$

$$\text{REP-ACT}: \frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P' \mid !P} \qquad \text{REP-COMM}: \frac{P \xrightarrow{\overline{x}\langle\tilde{y}\rangle} P' \quad P \xrightarrow{x(\tilde{y})} P''}{!P \xrightarrow{\tau} (P' \mid P'') \mid !P}$$

$$\text{TIMEOUT}: \frac{P \xrightarrow{\alpha} P'}{t[0].P \xrightarrow{\alpha} P'} \qquad \text{ABORT}: \frac{}{P \xrightarrow{abort}}\ \text{if}\ \ P\!\uparrow$$

図 1　遷移規則

$$\text{PASS}_\text{T}: \frac{}{t[\underline{n}].P \rightsquigarrow t[\underline{n-1}].P}\ \text{if } n > 0 \qquad \text{INACT}_\text{T}: \frac{}{\mathbf{0} \rightsquigarrow \mathbf{0}}$$

$$\text{OUT}_\text{T}: \frac{}{\overline{x}\langle\tilde{z}\rangle.P \rightsquigarrow \overline{x}\langle\tilde{z}\rangle.P} \qquad \text{IN}_\text{T}: \frac{}{x(\tilde{y}).P \rightsquigarrow x(\tilde{y}).P}$$

$$\text{SUM}_\text{T}: \frac{P \rightsquigarrow P' \quad Q \rightsquigarrow Q'}{P+Q \rightsquigarrow P'+Q'} \qquad \text{PAR}_\text{T}: \frac{P \rightsquigarrow P' \quad Q \rightsquigarrow Q'}{P \mid Q \rightsquigarrow P' \mid Q'}\ \text{if } P \mid Q \overset{\tau}{\not\rightarrow}$$

$$\text{RES}_\text{T}: \frac{P \rightsquigarrow P'}{\boldsymbol{\nu} x\, P \rightsquigarrow \boldsymbol{\nu} x\, P'} \qquad \text{REP}_\text{T}: \frac{P \rightsquigarrow P'}{!P \rightsquigarrow !P'}\ \text{if } P \mid P \overset{\tau}{\not\rightarrow}$$

$$\text{STRUCT}_\text{T}: \frac{P \rightsquigarrow P'}{Q \rightsquigarrow Q'}\ \text{if } P \equiv Q,\ P' \equiv Q'$$

$$\text{TIMEOUT}_\text{T}: \frac{P \rightsquigarrow P'}{t[\underline{0}].P \rightsquigarrow P'}$$

図 2　時間経過規則

$P \xrightarrow{\alpha} P'$ は入力，出力，内部アクションのいずれかのアクション α により P から P' に遷移することを表し，$P \rightsquigarrow P'$ は P が 1 単位時間の経過により P' に遷移することを表す．時間経過は図 2 の時間経過規則に基づく遷移によってのみ発生する．PAR_T 規則と REP_T 規則の条件により最大進行性が成り立つ [9]．そのため τ による遷移は時間経過による遷移に優先して発生する．

3　時間付き双模倣関係

時間付き π 計算におけるプロセス間の双模倣関係を定義する．時間付き双模倣関係は直観的には以下の条件を満たす関係である．

- 双方のプロセスの入出力アクションの列が同じ

- 双方のプロセスにおいて時間待ちの発生するタイミングと時間待ちの長さが同じ

定義 5 以下を満たす対称な関係 $\mathcal{R}$ を**時間付き強双模倣関係**と呼び，最大の関係を $\sim_{\mathcal{T}}$ と書く．

$(P,Q) \in \mathcal{R}$ の時，

- $P\!\uparrow \implies Q\!\uparrow$
- $P \xrightarrow{\alpha} P' \implies \exists Q'.\ Q \xrightarrow{\alpha} Q' \wedge (P',Q') \in \mathcal{R}$
- $P \rightsquigarrow P'' \implies \exists Q''.\ Q \rightsquigarrow Q'' \wedge (P'',Q'') \in \mathcal{R}$ □

定義 6 以下を満たす対称な関係 $\mathcal{R}$ を**時間付き弱双模倣関係**と呼び，最大の関係を $\approx_{\mathcal{T}}$ と書く．

$(P,Q) \in \mathcal{R}$ の時，

- $P\!\uparrow \implies Q\!\uparrow$
- $P \xrightarrow{\alpha}_{\tau} P' \implies \exists Q'.\ Q \xrightarrow{\alpha}_{\tau} Q' \wedge (P',Q') \in \mathcal{R}$
- $P \rightsquigarrow_{\tau} P'' \implies \exists Q''.\ Q \rightsquigarrow_{\tau} Q'' \wedge (P'',Q'') \in \mathcal{R}$

ただし $\xrightarrow{\alpha}_{\tau} = (\xrightarrow{\tau})^* \xrightarrow{\alpha} (\xrightarrow{\tau})^*$，$\rightsquigarrow_{\tau} = (\xrightarrow{\tau})^* \rightsquigarrow (\xrightarrow{\tau})^*$ とする． □

定理 1 $\sim_{\mathcal{T}}$，$\approx_{\mathcal{T}}$ は等価関係である． □

ここで合同性のためにコンテキストを定義する．

定義 7 コンテキスト $C[\cdot]$ を以下のように定義する．

$$\begin{aligned} C[\cdot] \ ::= \ & [\cdot] \mid x(\widetilde{y}).C[\cdot] \mid \overline{x}\langle\widetilde{z}\rangle.C[\cdot] \mid \tau.C[\cdot] \mid t[n].C[\cdot] \mid C[\cdot]|P \mid P|C[\cdot] \\ & \mid C[\cdot] + P \mid P + C[\cdot] \mid \boldsymbol{\nu} x\, C[\cdot] \mid !\,G[\cdot] \\ G[\cdot] \ ::= \ & x(\widetilde{y}).C[\cdot] \mid \overline{x}\langle\widetilde{z}\rangle.C[\cdot] \mid \tau.C[\cdot] \mid G[\cdot]|Q \mid Q|G[\cdot] \mid G[\cdot] + Q \\ & \mid Q + G[\cdot] \mid \boldsymbol{\nu} x\, G[\cdot] \mid !\,G[\cdot] \end{aligned}$$

ここで P は任意のプロセス，Q はアクションガードつきプロセス式である． □

コンテキスト CXT を構築するために必要な定義 7 の構文規則 $C[\cdot]$ あるいは $G[\cdot]$ を適用する回数を CXT の大きさと呼ぶ．

定理 2 $\sim_{\mathcal{T}}$ は入力プレフィックス以外のコンテキストについて合同性を満たす． □

定理 2 の証明は付録に示す．任意の代入に対して双模倣関係が保存されないため入力プレフィックスについては合同性が成立しない．例えば

$$(\overline{x} \mid y)\{y/x\} \quad \nsim_{\mathcal{T}} \quad (\overline{x}.y + y.\overline{x})\{y/x\}$$

であるが，これは $(\overline{x} \mid y)\{y/x\}$ で τ 遷移が可能になるからである．そこで任意の代入に対しても保存される双模倣関係を与える．

定義 8 P と Q がすべての代入 σ について $P\sigma \sim_{\mathcal{T}} Q\sigma$ を満たすならば，P と Q は timed strong full bisimilar であるといい $P \sim^{c}_{\mathcal{T}} Q$ と書く． □

補題 1 $y \in \mathsf{fn}(P) \cup \mathsf{fn}(Q) \cup \{z\}$ であるようなすべての y に対し $P\{y/z\} \sim_{\mathcal{T}} Q\{y/z\}$ ならば，$x(z).P \sim_{\mathcal{T}} x(z).Q$ である． □

定理 3 $\sim^{c}_{\mathcal{T}}$ は合同性を満たす．

証明： $P \sim^{c}_{\mathcal{T}} Q$ とする．ここで $C \neq x(z).C'$ の場合，$\sim^{c}_{\mathcal{T}}$ の定義より $\sim^{c}_{\mathcal{T}} \subseteq \sim_{\mathcal{T}}$ であり，さらに定理 2 より $C[P] \sim^{c}_{\mathcal{T}} C[Q]$ である．

$C = x(z).C'$ についてはコンテキストの大きさに関する帰納法を用いて $C[P] \sim^{c}_{\mathcal{T}} C[Q]$ を示す．この時，帰納法の仮定より $C'[P] \sim^{c}_{\mathcal{T}} C'[Q]$．$C'[P]\sigma$，$C'[Q]\sigma$ に含まれる，もしくは z である自由な名前 y と任意の代入 σ に対し，定義 8 から $C'[P]\sigma\{y/z\} \sim_{\mathcal{T}} C'[Q]\sigma\{y/z\}$ とできる．補題 1 より $x(z).C'[P]\sigma \sim_{\mathcal{T}} x(z).C'[Q]\sigma$．よって $C[P] \sim^{c}_{\mathcal{T}} C[Q]$． □

4 遅延時間順関係

実時間システムにおける時間制約は，何らかの動作に要する時間の上限や待機時間の下限である．実時間システムはこれらの制約を満たしながら動作する必要がある．そこで制約として与えられた時間内に動作することを示すことができればよい．デッドラインの仕様を S，実装を P，実装がデッドラインを破らないことを表す関係を $\mathcal{R}$ とした時，$(P,S)\in\mathcal{R}$ であることを示せばよい．しかし，時間付き双模倣関係は時間待ちのタイミングや長さが一致することを要求するため関係 $\mathcal{R}$ として利用するには適していない．そこで本節では，遅延時間の長さに着目しプロセスのアクションの発生が早いあるいは遅いを表す順序関係を与える．

定義 9 以下の条件を満たす関係 $\mathcal{R}$ を**遅延時間順関係**と呼び，最大の関係を $\precsim_{\mathcal{I}}$ と書く．

$(P,Q)\in\mathcal{R}$ の時，

$$\begin{array}{rcl} P\!\uparrow & \Longrightarrow & Q\!\uparrow \\ Q\!\uparrow & \Longrightarrow & P\!\uparrow \\ P\xrightarrow{\alpha}P' & \Longrightarrow & \exists Q'.\ Q\rightsquigarrow^{*}\xrightarrow{\alpha}Q'\wedge(P',Q')\in\mathcal{R} \\ P\rightsquigarrow P' & \Longrightarrow & \exists Q'.\ Q\rightsquigarrow Q'\wedge(P',Q')\in\mathcal{R} \\ Q\xrightarrow{\alpha}Q' & \Longrightarrow & \exists P'.\ P\xrightarrow{\alpha}P'\wedge(P',Q')\in\mathcal{R} \\ Q\rightsquigarrow Q' & \Longrightarrow & (P,Q')\in\mathcal{R}\vee(\exists P'.\ P\rightsquigarrow P'\wedge(P',Q')\in\mathcal{R}) \end{array}$$

□

直観的には，$P\precsim_{\mathcal{I}}Q$ ならば図 1 の遷移規則による遷移の列は P と Q のいずれも同じであり，図 2 の時間経過規則による遷移の回数は P の方が少ない．ある一つのアクションに注目すると，P の方が先かあるいは P と Q で同時に実行可能になる．つまり，P であるアクションが実行可能ならば，Q では 0 単位時間以上後に同じアクションが実行可能になる．時間経過遷移の回数は P の方が少ないので，P で時間経過遷移できるならば Q でも時間経過遷移できなければならず，Q で何らかのアクションが実行可能であれば，その時 P では必ず同じアクションが実行可能でなければならない．Q で時間経過遷移できるならば，P は P と Q の関係が保たれるのであれば時間経過遷移してもよい．

補題 2 $P\precsim_{\mathcal{I}}Q$ とする．この時，$P\rightsquigarrow^{n}P'\Longrightarrow\exists Q'.\ Q\rightsquigarrow^{n}Q'\wedge P'\precsim_{\mathcal{I}}Q'$.

証明： n に関する帰納法による． □

定理 4 $\precsim_{\mathcal{I}}$ は擬順序である．

証明： 反射律は定義より明らかに満たす．以下では推移律について示す．$\precsim_{\mathcal{I}}\precsim_{\mathcal{I}}\subseteq\precsim_{\mathcal{I}}$ であること，つまり $\precsim_{\mathcal{I}}\precsim_{\mathcal{I}}$ が遅延時間順関係であることを示せばよい．$P\precsim_{\mathcal{I}}\precsim_{\mathcal{I}}R$ とすると，$P\precsim_{\mathcal{I}}Q$ かつ $Q\precsim_{\mathcal{I}}R$ なる Q が存在する．ここでは $P\xrightarrow{\alpha}P'$ および $R\rightsquigarrow R'$ の場合について述べる．

- $P\xrightarrow{\alpha}P'$ とする．$R\rightsquigarrow^{*}\xrightarrow{\alpha}R'$ かつ $P'\precsim_{\mathcal{I}}\precsim_{\mathcal{I}}R'$ なる R' が存在することを示せばよい．$P\precsim_{\mathcal{I}}Q$ ゆえ $\exists Q'.\ Q\rightsquigarrow^{*}\xrightarrow{\alpha}Q'\wedge P'\precsim_{\mathcal{I}}Q'$．この時 $\exists n,Q''.\ Q\rightsquigarrow^{n}Q''\xrightarrow{\alpha}Q'$ なので，この Q'' に対し $Q\precsim_{\mathcal{I}}R$ と補題 2 より $\exists R''.\ R\rightsquigarrow^{n}R''\wedge Q''\precsim_{\mathcal{I}}R''$．さらに Q' に対して $\exists R'.\ R''\rightsquigarrow^{*}\xrightarrow{\alpha}R'\wedge Q'\precsim_{\mathcal{I}}R'$．ここで $R\rightsquigarrow^{n}R''$ と $R''\rightsquigarrow^{*}\xrightarrow{\alpha}R'$ より $R\rightsquigarrow^{*}\xrightarrow{\alpha}R'$ であり，$P'\precsim_{\mathcal{I}}Q'$ かつ $Q'\precsim_{\mathcal{I}}R'$ なので $P'\precsim_{\mathcal{I}}\precsim_{\mathcal{I}}R'$．以上より $R\rightsquigarrow^{*}\xrightarrow{\alpha}R'$ かつ $P'\precsim_{\mathcal{I}}\precsim_{\mathcal{I}}R'$ なる R' が存在する．

- $R \rightsquigarrow R'$ とする．$P \precsim_{\mathcal{I}} \precsim_{\mathcal{I}} R'$ または $P \rightsquigarrow P'$ かつ $P' \precsim_{\mathcal{I}} \precsim_{\mathcal{I}} R'$ なる P' が存在することを示せばよい．$Q \precsim_{\mathcal{I}} R$ ゆえ $Q \precsim_{\mathcal{I}} R'$ または $\exists Q'.\ Q \rightsquigarrow Q' \wedge Q' \precsim_{\mathcal{I}} R'$．$Q \precsim_{\mathcal{I}} R'$ ならば $P \precsim_{\mathcal{I}} Q$ より $P \precsim_{\mathcal{I}} \precsim_{\mathcal{I}} R'$．そうでなければ Q' に対して $P \precsim_{\mathcal{I}} Q$ ゆえ $P \precsim_{\mathcal{I}} Q'$ または $\exists P'.\ P \rightsquigarrow P' \wedge P' \precsim_{\mathcal{I}} Q'$．$P \precsim_{\mathcal{I}} Q'$ ならば $Q' \precsim_{\mathcal{I}} R'$ なので $P \precsim_{\mathcal{I}} \precsim_{\mathcal{I}} R'$．そうでなければ $P' \precsim_{\mathcal{I}} \precsim_{\mathcal{I}} R'$．以上より $P \precsim_{\mathcal{I}} \precsim_{\mathcal{I}} R'$ または $P \rightsquigarrow P'$ かつ $P' \precsim_{\mathcal{I}} \precsim_{\mathcal{I}} R'$ なる P' が存在する． □

定理 5 制限されたコンテキスト

$$\begin{aligned} C[\cdot] \quad ::= \quad & [\cdot] \mid \overline{x}\langle\widetilde{y}\rangle.C[\cdot] \mid \tau.C[\cdot] \mid t[n].C[\cdot] \mid C[\cdot]|R \mid R|C[\cdot] \\ & \mid C[\cdot] + S \mid S + C[\cdot] \mid \boldsymbol{\nu} x\, C[\cdot] \mid !\, G[\cdot] \\ G[\cdot] \quad ::= \quad & \overline{x}\langle\widetilde{z}\rangle.C[\cdot] \mid \tau.C[\cdot] \mid G[\cdot]|T \mid T|G[\cdot] \mid G[\cdot] + U \mid U + G[\cdot] \\ & \mid \boldsymbol{\nu} x\, G[\cdot] \mid !\, G[\cdot] \end{aligned}$$

に対して，$P \precsim_{\mathcal{I}} Q$ ならば $C[P] \precsim_{\mathcal{I}} C[Q]$ である．ただし R，T，U には $t[n]$ が出現しないとする．

証明： コンテキストの大きさに関する帰納法で証明する．ここでは $C = C' \mid R$ の場合を示す．$\mathcal{R} = \{(C'[P] \mid R, C'[Q] \mid R) \mid P \precsim_{\mathcal{I}} Q\}$ とする．R には $t[n]$ が出現しないため $R \rightsquigarrow R$ であり，$R \rightsquigarrow^* R$ である．

- $C[P]\!\uparrow$ とすると $C'[P]\!\uparrow$ あるいは $R\!\uparrow$ である．$C'[P]\!\uparrow$ ならば帰納法の仮定より $C'[Q]\!\uparrow$ であり，$R\!\uparrow$ ならば明らかに $C[Q]\!\uparrow$ となる．逆も同様である．
- $C'[P] \mid R \xrightarrow{\alpha} P' \mid R$ とする．ここで $C'[P] \xrightarrow{\alpha} P'$ である．
  - $\alpha \neq \tau$ ならば帰納法の仮定より $\exists Q', Q^*.\ C'[Q] \rightsquigarrow^* Q^* \xrightarrow{\alpha} Q' \wedge P' \precsim_{\mathcal{I}} Q'$ である．$C'[Q] \mid R \rightsquigarrow^* Q^* \mid R \xrightarrow{\alpha} Q' \mid R$ とできて，$P' \precsim_{\mathcal{I}} Q'$ ゆえ $(P' \mid R, Q' \mid R) \in \mathcal{R}$.
  - $\alpha = \tau$ ならば帰納法の仮定より $\exists Q'.\ C'[Q] \xrightarrow{\tau} Q' \wedge P' \precsim_{\mathcal{I}} Q'$ である．$\alpha = \tau$ ゆえ時間経過遷移は不可能である．$C'[Q] \mid R \xrightarrow{\tau} Q' \mid R$ とできて，$P' \precsim_{\mathcal{I}} Q'$ ゆえ $(P' \mid R, Q' \mid R) \in \mathcal{R}$.
- $C'[P] \mid R \xrightarrow{\tau} P' \mid R'$ とする．ここで $C'[P] \xrightarrow{\alpha} P'$，$R \xrightarrow{\overline{\alpha}} R'$ である．帰納法の仮定から同様に $C'[Q] \mid R \rightsquigarrow^* Q^* \mid R \xrightarrow{\tau} Q' \mid R'$ とできて $(P' \mid R', Q' \mid R') \in \mathcal{R}$.
- $C'[P] \mid R \xrightarrow{\alpha} C'[P] \mid R'$ とする．この時，α が τ であるか否かに関わらず $C'[Q] \mid R \xrightarrow{\alpha} C'[Q] \mid R'$ とできて，帰納法の仮定から $C'[P] \precsim_{\mathcal{I}} C'[Q]$ であるので $(C'[P] \mid R', C'[Q] \mid R') \in \mathcal{R}$.
- $C'[Q] \mid R \rightsquigarrow Q'' \mid R$ とする．ここで $C'[Q] \rightsquigarrow Q''$ である．帰納法の仮定から $C'[P] \precsim_{\mathcal{I}} Q''$ あるいは $\exists P''.\ C'[P] \rightsquigarrow P'' \wedge P'' \precsim_{\mathcal{I}} Q''$ である．$C'[P] \precsim_{\mathcal{I}} Q''$ ならば $(C'[P] \mid R, Q'' \mid R) \in \mathcal{R}$．そうでなければ $C'[P] \mid R \rightsquigarrow P'' \mid R$ とできて $(P'' \mid R, Q'' \mid R) \in \mathcal{R}$.
- $C'[P] \mid R \rightsquigarrow P'' \mid R$, $C'[Q] \mid R \xrightarrow{\alpha} Q' \mid R$, $C'[Q] \mid R \xrightarrow{\tau} Q' \mid R'$, $C'[Q] \mid R \xrightarrow{\alpha} C'[Q] \mid R'$ についても同様に示すことができる．

以上より $\mathcal{R} \subseteq \precsim_{\mathcal{I}}$ である． □

時間経過アクションを含むプロセスとの並行合成においては遅延時間順関係は保存されない．例えば，$P = a.\mathbf{0}, Q = t[\underline{1}].a.\mathbf{0}, R = t[\underline{2}].b.\mathbf{0}$ とすると $P \precsim_{\mathcal{I}} Q$ である．しかし $P \mid R \xrightarrow{a} \mathbf{0} \mid R$ に対し $Q \mid R \rightsquigarrow \xrightarrow{a} \mathbf{0} \mid t[\underline{1}].b.\mathbf{0}$ であるため $P \mid R \not\precsim_{\mathcal{I}} Q \mid R$ となり関係は成り立たない．$P \precsim_{\mathcal{I}} Q$ の時 $Q \mid R$ の方が Q の次の入出力あるいは τ

アクションまでに多くの時間経過アクションを必要とする．R が時間経過アクションをプレフィックスに持つならば，R の次のアクションまでに必要な時間経過は逆に少なくなる．遅延時間順関係はプロセスが遷移しても常に保たれなければならないため，並行合成するプロセスが時間経過アクションを含むことはできない．同様の理由により，複製演算子を適用するプロセスは時間経過アクションを含むプロセスと並行合成あるいは選択合成されたプロセスではない必要がある．

遅延時間順関係にある 2 つのプロセスは外部からは観測できない τ 遷移も含めて同じアクション列が実行可能である．さらに，τ 遷移以外の外部から観測できるアクションのみに着目した遅延時間順関係を与える．

定義 10 以下の条件を満たす関係 $\mathcal{R}$ を**弱遅延時間順関係**と呼び，最大の関係を $\precsim_{\mathcal{I}}$ と書く．

$(P, Q) \in \mathcal{R}$ の時，

$$\begin{array}{rcl} P\!\uparrow & \Longrightarrow & Q\!\uparrow \\ Q\!\uparrow & \Longrightarrow & P\!\uparrow \\ P \xrightarrow{\alpha}_{\tau} P' & \Longrightarrow & \exists Q'.\ Q \rightsquigarrow_{\tau}^{*} \xrightarrow{\alpha}_{\tau} Q' \wedge (P', Q') \in \mathcal{R} \\ P \rightsquigarrow_{\tau} P' & \Longrightarrow & \exists Q'.\ Q \rightsquigarrow_{\tau} Q' \wedge (P', Q') \in \mathcal{R} \\ Q \xrightarrow{\alpha}_{\tau} Q' & \Longrightarrow & \exists P'.\ P \xrightarrow{\alpha}_{\tau} P' \wedge (P', Q') \in \mathcal{R} \\ Q \rightsquigarrow_{\tau} Q' & \Longrightarrow & (P, Q') \in \mathcal{R} \vee (\exists P'.\ P \rightsquigarrow_{\tau} P' \wedge (P', Q') \in \mathcal{R}) \end{array}$$

□

定理 6 $\precsim_{\mathcal{I}}$ は擬順序である． □

定理 7 制限されたコンテキスト

$$\begin{array}{rcl} C[\cdot] & ::= & [\cdot] \mid \overline{x}\langle\widetilde{y}\rangle.C[\cdot] \mid \tau.C[\cdot] \mid t[n].C[\cdot] \mid C[\cdot]|R \mid R|C[\cdot] \\ & & \mid C[\cdot] + S \mid S + C[\cdot] \mid \boldsymbol{\nu} x\, C[\cdot] \mid !\, G[\cdot] \\ G[\cdot] & ::= & \overline{x}\langle\widetilde{z}\rangle.C[\cdot] \mid \tau.C[\cdot] \mid G[\cdot]|T \mid T|G[\cdot] \mid G[\cdot] + U \mid U + G[\cdot] \\ & & \mid \boldsymbol{\nu} x\, G[\cdot] \mid !\, G[\cdot] \end{array}$$

に対して，$P \precsim_{\mathcal{I}} Q$ ならば $C[P] \precsim_{\mathcal{I}} C[Q]$ である．ただし R には $t[n]$ が出現しないとする． □

5 例

本節ではストリーミング配信の簡単な例について述べる。映像を配信するサーバ、配信される映像を受信して再生するプレイヤ、サーバとプレイヤの間の通信路を確保する 2 つのルータからなるシステムを考える。プレイヤはいずれかのルータに接続して配信を受ける。しかし、もし一定時間配信されてこなければもう一方のルータに接続し直す。ルータに接続してから配信の開始まで一定時間必要であり、ここで必要な時間はルータによって異なるとする。またサーバから送信される映像は圧縮されており、プレイヤで再生する前に解凍しなければならないとする。

サーバは以下のように記述できる。

$$Server \stackrel{def}{=} (\boldsymbol{\nu} g)\,(\overline{g} \mid !\,(g.\overline{v}\langle video\rangle.\overline{g} + g.t[\underline{2}].\overline{v}\langle video\rangle.\overline{g}))$$

サーバは名前 v を通して映像を配信する。名前 $video$ は映像の 1 フレームを表す。実際にはフレームは次々と切り替わっていくが、簡単のためにそれらはすべて同じ名前 $video$ で表す。サーバでは遅延なく送信されるか 2 単位時間遅延して送信されるかがフレームごとに非決定的に選択される。

2 つのルータは

$$\begin{aligned} Router_1 &\stackrel{def}{=} !\,attach(l).\bar{l}\langle \underline{2}, v, rel_1\rangle.rel_1 \\ Router_2 &\stackrel{def}{=} !\,attach(l).\bar{l}\langle \underline{4}, v, rel_2\rangle.rel_2 \end{aligned}$$

と記述できる。名前 $attach$ によりプレイヤから接続要求を受け、配信開始までの時間、映像を送るための名前 v、プレイヤとの接続を切断する rel_i をプレイヤに送信する。その後はプレイヤからの切断を待つ。ここでは配信開始までの時間を $Router_1$ で 2 単位時間、$Router_2$ で 4 単位時間とする。

プレイヤの記述を以下に示す。

$$\begin{aligned} Player \stackrel{def}{=}\ & (\boldsymbol{\nu} l, g, h)\,(\overline{attach}\langle l\rangle.\bar{h} \\ & \mid !\,h.l(n, v, r).t[n].v(video).t[\underline{1}].\overline{play}\langle video\rangle.\bar{g} \\ & \mid !\,g.(v(video).t[\underline{1}].\overline{play}\langle video\rangle.\bar{g} + t[\underline{1}].\tau.\overline{attach}\langle l\rangle.\bar{r}.\bar{h})) \end{aligned}$$

プレイヤは初めに名前 $attach$ によりいずれかのルータに接続要求を出し、配信開始までの時間、映像を受けるための名前 v、ルータとの接続を切断する r を受信する。その後、受信した時間だけ待機する。次に名前 v によりサーバから映像の配信を待機する。配信を受けたら 1 単位時間で圧縮された映像を解凍して名前 $play$ により再生し、次の映像の待機に戻る。もしサーバからの配信が 1 単位時間以内に行われない場合は、名前 $attach$ により再びいずれかのルータに接続要求を出し、r により接続しているルータと切断する。

システム全体は

$$\begin{aligned} System \stackrel{def}{=}\ & (\boldsymbol{\nu} attach, v, video.rel_1, rel_2) \\ & (Server \mid Router_1 \mid Router_2 \mid Player) \end{aligned}$$

と記述する。システムの実行系列の一つを以下に示す。

$$\begin{aligned} System \xrightarrow{\tau}{}^{4} \rightsquigarrow^{2}\ & (\boldsymbol{\nu} attach, v, video, rel_1, rel_2) \\ & (\bar{v}\langle video\rangle \mid Server \mid rel_1 \mid Router_1 \mid Router_2 \\ & \mid v(video).t[\underline{1}].\overline{play}\langle video\rangle.\bar{g} \mid !\,h.\cdots \mid !\,g.\cdots) \\ \xrightarrow{\tau}\rightsquigarrow\ & (\boldsymbol{\nu} attach, v, video, rel_1, rel_2) \\ & (Server \mid rel_1 \mid Router_1 \mid Router_2 \\ & \mid \overline{play}\langle video\rangle.\bar{g} \mid !\,h.\cdots \mid !\,g.\cdots) \\ \xrightarrow{\overline{play}}\xrightarrow{\tau}{}^{2}\ & (\boldsymbol{\nu} attach, v, video, rel_1, rel_2) \\ & (t[\underline{2}].\bar{v}\langle video\rangle \mid Server \mid rel_1 \mid Router_1 \mid Router_2 \mid !\,h.\cdots \\ & \mid v(video).t[\underline{1}].\overline{play}\langle video\rangle.\bar{g} + t[\underline{1}].\tau.\overline{attach}\langle l\rangle.\bar{r}.\bar{h} \mid !\,g.\cdots) \\ \rightsquigarrow\xrightarrow{\tau}\ & (\boldsymbol{\nu} attach, v, video, rel_1, rel_2) \\ & (t[\underline{1}].\bar{v}\langle video\rangle \mid Server \mid rel_1 \mid Router_1 \mid Router_2 \mid !\,h.\cdots \\ & \mid \overline{attach}\langle l\rangle.\bar{r}.\bar{h} \mid !\,g.\cdots) \\ \xrightarrow{\tau}{}^{3} \rightsquigarrow^{4}\ & (\boldsymbol{\nu} attach, v, video, rel_1, rel_2) \\ & (\bar{v}\langle video\rangle \mid Server \mid Router_1 \mid rel_2 \mid Router_2 \\ & \mid v(video).t[\underline{1}].\overline{play}\langle video\rangle.\bar{g} \mid !\,h.\cdots \mid !\,g.\cdots) \\ \xrightarrow{\tau}\rightsquigarrow\ & (\boldsymbol{\nu} attach, v, video, rel_1, rel_2) \\ & (\bar{v}\langle video\rangle \mid Server \mid Router_1 \mid rel_2 \mid Router_2 \\ & \mid \overline{play}\langle video\rangle.\bar{g} \mid !\,h.\cdots \mid !\,g.\cdots) \\ \xrightarrow{\overline{play}}\ & \cdots \end{aligned}$$

この実行系列は以下の動作を表す。

1. 最初に $Router_1$ に接続し映像を受信して再生
2. 映像の次のフレームの待機中にタイムアウトが発生
3. $Router_2$ に接続し次のフレームを受信して再生

$\overline{play}$ による遷移によって表される映像の再生から次の再生までの遅延が最も短くなるのは、同じルータから続けて受信する場合で 1 単位時間の遅延が発生する。逆に遅延が最も長くなるのは、$Router_1$ に接続している状態で映像配信の待機中にタイムアウトし $Router_2$ に接続する場合で 6 単位時間の遅延である。これらのことは

$$(\boldsymbol{\nu} g)\,(\overline{g} \mid !\,g.t[\underline{1}].\overline{play}\langle video\rangle.\overline{g}) \;\precsim_{\mathcal{T}}\; System$$
$$System \;\precsim_{\mathcal{T}}\; (\boldsymbol{\nu} g)\,(\overline{g} \mid !\,g.t[\underline{6}].\overline{play}\langle video\rangle.\overline{g})$$

を示すことで確認できる。

6 関連研究

πRT-calculus [4] では π 計算にタイムアウト演算子 $\overset{t}{\triangleright}$ を導入して時間拡張している．この時間拡張は，離散時間である，アクションと時間経過を分離している，時間を名前の一種とみなしメッセージとして通信できる，といった点で我々の時間付き π 計算と類似している．しかし構文と動作意味は定義しているが，双模倣関係などプロセス間の関係が定義されていない．

Berger らはタイマ，メッセージ消失，プロセスの実行失敗などを導入して π 計算を拡張し，二相コミットプロトコルを記述している [5]．タイマは $\mathsf{timer}^t(Q, R)$ と表されるプロセスとして実現されている．動作は時間ステップ関数と動作意味定義によって定義されており，時間ステップ関数の適用として時間の経過を表現する．時間経過を意味するアクションはなく，入出力アクションや τ アクションによる 1 ステップの遷移で 1 単位時間が経過するとしている．

7 おわりに

本稿では離散時間を導入して拡張した π 計算について，時間付き双模倣関係を定義し合同性を満たすことを示した．時間付き双模倣関係によりアクションの実行列と実行のタイミングからプロセスの等価性を調べることができる．時間付き弱双模倣関係の合同性と full bisimilarity については今後証明する必要がある．時間付き双模倣関係に加え，プロセスの時間待ちの長さの違いを反映する遅延時間順関係を定義した．この関係は連続する 2 つのアクションの間で経過する時間の長さを比較する関係であり，動作が決められた時間内に完了できるか否かなどを調べることが可能である．実際には 5 節に例で示したように τ 遷移を捨象する弱遅延時間順関係を用いる．弱遅延時間順関係が合同性を満たすのは制限されたコンテキストに対してであり並行合成には条件が付くが，時間経過が振舞いに影響するプロセスに着目すればよいため動作検証に有用である．

遅延時間順関係に基づいてある動作が 10 秒以内に開始するか，あるいは 5 秒以上待機してから開始するかを調べることはできる．しかし，5 秒以上 10 秒以内に開始するかどうかを調べることはできない．2 つのプロセス間の時間待ちの長さの大小関係が実行に伴って変化する場合，遅延時間順関係は保たれない．そのため実時間システムの検証への適用を考えると不十分である．テスト等価性 [1] などの手法を用

いた検証手法については今後の課題である．

双模倣関係に基づく動作の等価性判定や順序関係を用いた動作の比較だけでなく，型理論 [2] やモデル検査 [3] などを用いて動作の性質について検証することもシステム開発には有効な手段である．時間付き π 計算に対する型理論の導入や，モデル検査技法の適用は今後の課題である．

謝辞 熱心に議論して頂ける阿草研究室の皆様に感謝致します．本研究の一部は文部科学省科学技術研究費基盤研究 (C)(2) 課題番号 16500027 および基盤研究 (B)(2) 課題番号 14380141 の助成による．

参考文献

[1] Michele Boreale and Rocco De Nicola. Testing equivalence for mobile processes. *Information and Computation*, 120:279–303, 1995.
[2] Luca Cardelli. Type systems. In Allen B. Tucker, editor, *The Computer Science and Engineering Handbook*. CRC Press, 1997.
[3] E.M.Clarke, Orna Grunberg, and Doron Peled. *Model Checking*. The MIT Press, 1999.
[4] Jeremy Y. Lee and John Zic. On modeling real-time mobile processes. In *Proceedings of the twenty-fifth Australasian conference on Computer science*, pages 139–147. Australian Computer Society, Inc., 2002.
[5] M.Berger and K.Honda. The two-phase commitment protocol in an extended π-calculus. In *Preliminary Proceedings of EXPRESS '00*, 2000.
[6] Robin Milner. *Communication and Mobile Systems: the π-Calculus*. Cambridge University Press, 1999.
[7] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, part I/II. *Information and Computation*, 100:1–77, 1992.
[8] Davide Sangiorgi and David Walker. *The π-calculus: A Theory of Mobile Processes*. Cambridge University Press, 2001.
[9] 桑原 寛明, 結縁 祥治, 阿草 清滋. 時間付き π 計算によるリアルタイムオブジェクト指向言語の形式的記述. 情報処理学会論文誌, 45(6):1498–1507, 2004.

A 定理 2 の証明

$\sim_{\mathcal{I}}$ が入力プレフィックス以外のコンテキストについて合同であることを Sangiorgi の up-to テクニック [8] に基づいて証明する．初めに時間付き強双模倣関係に対応した強進行性 (strong progression) を定義する．

定義 11 関係 $\mathcal{R}$ が以下を満たす時 $\mathcal{R}$ は $\mathcal{S}$ に強進行するといい $\mathcal{R} \rightarrowtail \mathcal{S}$ と書く．

$(P, Q) \in \mathcal{R}$ の時，

- $P\!\uparrow \Longrightarrow Q\!\uparrow$
- $P \xrightarrow{\alpha} P' \Longrightarrow \exists Q'.\ Q \xrightarrow{\alpha} Q' \wedge (P', Q') \in \mathcal{S}$
- $P \rightsquigarrow P'' \Longrightarrow \exists Q''.\ Q \rightsquigarrow Q'' \wedge (P'', Q'') \in \mathcal{S}$
- およびこれらの P と Q を置換した条件 □

定義より明らかに $\mathcal{R} \rightarrowtail \mathcal{R}$ ならば $\mathcal{R}$ は時間付き強双模倣関係である．

補題 3 $\mathcal{S}$ が時間付き強双模倣関係である時，$\mathcal{R} \rightarrowtail \mathcal{S}$ ならば $\mathcal{R} \subseteq \sim_{\mathcal{I}}$ である．

証明： $(P, Q) \in \mathcal{R}$，$P \xrightarrow{\alpha} P'$，$P \rightsquigarrow P''$ とすると，$\mathcal{R} \rightarrowtail \mathcal{S}$ より $Q \xrightarrow{\alpha} Q'$ かつ $(P', Q') \in \mathcal{S}$ となる Q'，$Q \rightsquigarrow Q''$ かつ $(P'', Q'') \in \mathcal{S}$ となる Q'' が存在する．ここで $(P, Q) \in \mathcal{R} \cup \mathcal{S}$，$(P', Q') \in \mathcal{R} \cup \mathcal{S}$，$(P'', Q'') \in \mathcal{R} \cup \mathcal{S}$ なので $\mathcal{R} \cup \mathcal{S} \rightarrowtail \mathcal{R} \cup \mathcal{S}$．よって $\mathcal{R} \cup \mathcal{S}$ は時間付き双模倣関係であるので $\mathcal{R} \subseteq \sim_{\mathcal{I}}$．□

補題 4

(1) $\mathcal{R} \subseteq \mathcal{S}$ かつ $\mathcal{S} \rightarrowtail \mathcal{T}$ ならば $\mathcal{R} \rightarrowtail \mathcal{T}$.
(2) $\mathcal{R} \rightarrowtail \mathcal{S}$ かつ $\mathcal{S} \subseteq \mathcal{T}$ ならば $\mathcal{R} \rightarrowtail \mathcal{T}$. □

補題 5 $\{\mathcal{R}_i \mid i \in I\}$ と $\{\mathcal{S}_j \mid j \in J\}$ を関係の集合とする．ただし，$\mathcal{R} \stackrel{def}{=} \bigcup_{i\in I} \mathcal{R}_i$, $\mathcal{S} \stackrel{def}{=} \bigcup_{j\in J} \mathcal{S}_j$ とする．この時,
(1) すべての i に対し $\mathcal{R}_i \rightarrowtail \mathcal{S}_j$ となる j が存在するならば $\mathcal{R} \rightarrowtail \mathcal{S}$.
(2) すべての i に対し $\mathcal{R}_i \rightarrowtail \mathcal{R}_j$ となる j が存在するならば $\mathcal{R}$ は時間付き強双模倣関係である.

証明： 初めに (1) を証明する．$\mathcal{S}$ の定義より $\mathcal{S}_j \subseteq \mathcal{S}$ であり，補題 4(2) よりすべての i に対し $\mathcal{R}_i \rightarrowtail \mathcal{S}$ が成り立つ．よって $\mathcal{R} \rightarrowtail \mathcal{S}$．次に (2) は (1) より $\mathcal{R} \rightarrowtail \mathcal{R}$ であるので $\mathcal{R}$ は時間付き強双模倣関係である． □

次に strongly safe function を定義する.

定義 12 関数 $\mathcal{F}$ について，$\mathcal{R} \subseteq \mathcal{S}$ かつ $\mathcal{R} \rightarrowtail \mathcal{S}$ の時 $\mathcal{F}(\mathcal{R}) \subseteq \mathcal{F}(\mathcal{S})$ かつ $\mathcal{F}(\mathcal{R}) \rightarrowtail \mathcal{F}(\mathcal{S})$ が成り立つならば $\mathcal{F}$ は *strongly safe function* である． □

補題 6 関数 $\mathcal{F}$ が *strongly safe function* かつ $\mathcal{R} \rightarrowtail \mathcal{F}(\mathcal{R})$ ならば $\mathcal{R} \subseteq \sim_{\mathcal{T}}$, $\mathcal{F}(\mathcal{R}) \subseteq \sim_{\mathcal{T}}$ である.

証明： 文献 [8] の Lemma 2.3.8 の証明と同様．ただし $\rightsquigarrow$ は $\rightarrowtail$ と読み替える． □

補題 7 関数 $\mathcal{F}$ が *strongly safe function* かつ $\sim_{\mathcal{T}} \subseteq \mathcal{F}(\sim_{\mathcal{T}})$ ならば $\mathcal{F}(\sim_{\mathcal{T}}) = \sim_{\mathcal{T}}$ である.

証明： $\mathcal{F}(\sim_{\mathcal{T}}) \subseteq \sim_{\mathcal{T}}$ を示せばよい．$\sim_{\mathcal{T}} \rightarrowtail \sim_{\mathcal{T}}$ であり $\mathcal{F}$ は strong safe なので $\mathcal{F}(\sim_{\mathcal{T}}) \rightarrowtail \mathcal{F}(\sim_{\mathcal{T}})$．さらに $\sim_{\mathcal{T}} \subseteq \mathcal{F}(\sim_{\mathcal{T}})$ なので補題 4 (1) より $\sim_{\mathcal{T}} \rightarrowtail \mathcal{F}(\sim_{\mathcal{T}})$ で，補題 6 より $\mathcal{F}(\sim_{\mathcal{T}}) \subseteq \sim_{\mathcal{T}}$ である． □

ここで関数 $\mathcal{F}_{ni}$ を以下のように定義する.

$$\mathcal{F}_{ni}(\mathcal{R}) \stackrel{def}{=} \{(C\eta, C\eta') \mid \forall i.\ (\eta_i, \eta'_i) \in \mathcal{R}\}$$

C は $[\cdot]$ を複数持つ入力プレフィックス以外のコンテキストとする．η はプロセス式のリストであり i 番目の式を η_i と書く．$C\eta$ は，コンテキスト C に含まれる $[\cdot]$ に対し最も左側に現れるものから順に $[\cdot]_1, [\cdot]_2, \ldots$ のように番号を付し，すべての i に対し $[\cdot]_i$ を η_i で置換して得られるプロセス式を表す.

補題 8 関数 $\mathcal{F}_{ni}$ は *strongly safe function* である.

証明： 文献 [8] の Lemma 2.3.21 と同様に証明できる．$\mathcal{R} \subseteq \mathcal{S}$ および $\mathcal{R} \rightarrowtail \mathcal{S}$ を仮定し，プロセス P, Q について $(P, Q) \in \mathcal{R}$ であり，C が入力プレフィックス以外のコンテキストで $C[P] \stackrel{\alpha}{\longrightarrow} P'$，$C[P] \rightsquigarrow P''$ の時，ある Q' に対し $C[Q] \stackrel{\alpha}{\longrightarrow} Q'$ かつ $(P', Q') \in \mathcal{F}_{ni}(\mathcal{S})$，$Q''$ に対し $C[Q] \rightsquigarrow Q''$ かつ $(P'', Q'') \in \mathcal{F}_{ni}(\mathcal{S})$ とできることを示せばよい．ここでは $C = t[\underline{n}].C'$(ただし $n > 0$) の場合について示す.

この時 $C[P] \stackrel{\alpha}{\Rightarrow}$, $C[P] \rightsquigarrow P' = t[\underline{n-1}].C'[P]$ である．さらに $C[Q] \stackrel{\alpha}{\Rightarrow}$, $C[Q] \rightsquigarrow Q' = t[\underline{n-1}].C'[Q]$ であり，仮定より $\mathcal{R} \subseteq \mathcal{S}$ であるので $(t[\underline{n-1}].C'[P], t[\underline{n-1}].C'[Q]) \in \mathcal{F}_{ni}(\mathcal{S})$ とできる． □

定理 2 の証明を以下に示す.

証明： 明らかに $\sim_{\mathcal{T}} \subseteq \mathcal{F}_{ni}(\sim_{\mathcal{T}})$ であり，補題 8 及び補題 7 より $\mathcal{F}_{ni}(\sim_{\mathcal{T}}) = \sim_{\mathcal{T}}$．よって $\sim_{\mathcal{T}}$ は入力プレフィックス以外のコンテキストに対して保たれる． □