パイ計算による仕様を検証する論理体系

竹内 泉[†]

本発表では，拡張されたパイ計算のプロセスの性質を証明する論理体系を提案する．特に，この論理体系はプロセスの活性や停止性を証明する．この論理体系の対象であるパイ計算は，ネピの方法に従って代数仕様の項をパイ計算に付け加えることによって拡張されたものである．代数仕様の項があるので，パイ計算の原形そのままよりも，仕様の記述が便利になっている．パイ計算の不停止は余再帰的枚挙完全であるので，その完全な再帰的公理化は不可能である．そのため，論理体系の設計方針は，健全で，かつ，適当に証明力が強く，適当に簡明である，というものとなる．論理体系は，代数仕様の項をその項とするような様相述語論理である．各論理式はある 1 個のプロセスの満たす性質を表す．様相記号はプロセスの遷移を表す．プロセスの並行合成に対応して，論理式の文法には並行合成を表す論理記号がある．それは線形論理の乗法的連言によく似た性質を持つ論理記号であり，論理規則もまた線形論理によく似たものとなっている．

A Logical System to Verify Specifications Written by Pi-calculus

Izumi Takeuti[†]

This presentation gives a logical system which proves the properties over process-terms of a modified variant of Pi-calculus. Especially, the logical system can prove liveness and termination of processes. The modification of Pi-calculus is to add terms of an algebra to Pi-calculus accoring to the style of Nepi. Because the termination on Pi-calculus is co-recursively-enumerable-complete, one cannot axiomatise it recursively. Therefore, we should make the logical system such that it is sound, has the reasonable power to prove and is reasonably light. The logical system which is given here is a modal predicate logic whose terms are the terms of the algebra. Each formula denotes the property satisfied by a process. Modalities denote transitions. The logical system has a logical symbol to denote parallel composition of processes. The symbol is similar to multiplicative conjunction in linear logic, and the logical rules on it are also similar to those of linear logic.

1. 目 的

パイ計算は動的プロセスの仕様を記述する高い能力があることで知られている[4]．動的プロセスにおいては，プロセス P が遷移してプロセス Q に至るかどうかが問題になる．

プロセス P が遷移 α によってプロセス Q に至ることは $P \rightarrow^{\alpha} Q$ と書こう．もしある論理体系があって，$P \rightarrow^{\alpha} Q$ か否かを任意のプロセス P，Q と任意の遷移 α に対して証明できるならば，動的プロセスの仕様の分析にとって有用である．しかしながら，パイ計算に対してそのような意味で完全な論理体系を作ることはできない．なぜなら，パイ計算は万能チューリング機械を模倣でき，そのため $P \rightarrow^{\tau^*} Q$ は決定不能となるからである．正確にいうならば，$\{(P,Q)|P \rightarrow^{\tau^*} Q\}$ という集合は Σ_1^0 完全であり，その補集合である $\{(P,Q)|P \not\rightarrow^{\tau^*} Q\}$ は Π_1^0 完全である．再帰的公理化可能であるような論理体系の定理集合は Σ_1^0 であり，Π_1^0 完全の元をすべて証明することはできない．

しかし完全ではなくても，ある程度に証明力のある論理体系ならば，それは有用であろう．つまり，$P \rightarrow^{\alpha} Q$ という形のものはすべて証明できて，$P \not\rightarrow^{\alpha} Q$ という形のもののうち，いくつかの重要なものは証明できる，ということである．

すべてを証明しつくすことは不可能なのであるから，証明可能なものを増やしていくためには論理体系に高階論理の操作を付け加えていく必要がある．そうしていくと，論理体系は複雑で難解なものとなる．一般に，証明可能な定理の多さと，論理体系の単純さとは二律背反の関係にある．したがって，目的に応じて，必要な定理は十分に証明できて，論理体系の複雑さは妥当

† 産業技術総合研究所システム検証研究センター
Research Center for Verification and Semantics, National Institute of Advanced Science and Technology

な範囲にある，という妥協点を見付けなければならない．ここで「目的に対して必要な定理は十分に証明できる」も「論理体系の複雑さ」も，どちらもいまだ客観的な概念ではなく，定量的に評価することができない．今後の研究によってこの定量的評価が確立することが望まれる．

本研究では，パイ計算のプロセス項の性質を証明する論理体系を提案する．この論理体系は，たとえば活性や停止性等を証明できる．

実用の際の便宜から，パイ計算に代数仕様の項を付加した π_A を定義した．パイ計算は代数仕様を模倣できるので，表現力の点からは本質的な拡張ではないが，しかし具体的なプロセスを書く際には便利である．代数仕様の項を付加する方法は，Nepi[5] の方法に従った．

2. 代　　数

定義 2.1 （名前対称代数）代数 A が**名前対称**であるとは，以下を満たすことをいう．

– 函数名の集合 Σ は，互いに交わらない 3 つの集合 Σ_F，Σ_N，Σ_E の和集合である．

Σ_F ：通常の函数名の可算個の集合である．各函数名には 0 以上の引数の個数が定まっている．引数 0 個の函数名は定数として働く．

Σ_N ：可算無限個の**名前**の集合．名前はすべて引数 0 個の函数名である．

$\Sigma_E = \{\mathsf{eq}, \mathsf{true}, \mathsf{false}\}$：特別な 3 個の函数名．eq の引数は 2 個，true と false とは引数 0 個．

– **変数**は可算無限個ある．

– **項**は函数名 Σ と変数から通常のように作られる．

– 変数のない項は**基底項**という．基底項全体の集合を $GTerm$ と書く．

– **公理**の集合は，項の等式の再帰的可算な集合．他の条件により，有限ではありえないことがいえる．

– **定理**とは，項の等式であり，公理から通常の一階述語論理によって演繹されるものである．項 t，t' に対し，$t = t'$ が定理であることを $A \vdash t = t'$ と書く．

– V は再帰的可算な集合であって $\Sigma_N \subset V \subset GTerm$ であり，かつ，任意の $t \in GTerm$ に対して唯一の $v \in V$ があって $A \vdash t = v$ となるようなものである．V の元は**値**という．

–（決定性）$A \not\vdash \mathsf{true} = \mathsf{false}$，かつ，任意の相異なる値 v，v' に対して $A \vdash \mathsf{eq}(v, v) = \mathsf{true}$ かつ $A \vdash \mathsf{eq}(v, v') = \mathsf{false}$.

–（対称性）σ は Σ_N 上の任意の置換であるとき，項 t，t' に対して $A \vdash t = t'$ と $A \vdash t\sigma = t'\sigma$ とは同値．

注意 2.2　この代数の典型例は，完備な項書換系である．このとき，V は正規形の項の集合となる．

注意 2.3　決定性と対称性は Nepi[5] に倣った．決定性を満たすために，公理は無限であることが必要となる．

3. パ イ 計 算

定義 3.1 （代数の項を持つパイ計算）名前対称代数 A があるとき，パイ計算に A の項を付け加えた π_A を以下のように定義する．π_A には名前，値項，プロセス項，ガード項という 4 つの構文範疇がある．また補助的な範疇として変数とプロセス変数がある．

変数：x，代数 A の変数

名前：c，代数 A の名前，すなわち Σ_N の元

値項：t，代数 A の項

プロセス変数：X，0 以上の各引数の個数に対して十分多くある．

$$\begin{aligned}
\text{プロセス項：}\ P &::= \ 1 \mid P * P \mid (c)P \mid Xt_1...t_n \mid G \\
\text{ガード項：}\quad G &::= \ \mathsf{out}\ tt'P \mid \mathsf{in}\ txP \mid \tau P \\
&\quad\ \mid \ G + G \mid \mathsf{if}\ tG \\
&\quad\ \mid \ (\lambda x_1...x_n.\rho X.G)t_1...t_n
\end{aligned}$$

ここで X の引数の個数は n

プロセス項全体の集合を $Proc$ と書く．

記法 3.2　以降，x，x_i 等は変数を表し，c 等は名前を表し，v 等は値を表し，t，t'，t_i，u，u' 等は項を表し，P，Q 等はプロセス項を表す．

注意 3.3　プロセス項 1 は正常終了を表す．他の文献の 0 に似ているが，たとえば文献 4) の 0 は単なる不活性であって，正常終了という意味はない．out tt'，in tt' はそれぞれ，文献 4) の $\bar{x}y$，$x(y)$ に相当する．プロセス項 $P * Q$ は P と Q の並列合成を表す．他の文献では $P|Q$ 等と書かれる．プロセス項 $(c)P$ は c の遮蔽，もしくは新しい名前 c の導入を表す．プロセス項 $(\lambda\vec{x}.\rho X.P)\vec{t}$ は再帰を表す．

定義 3.4 （α 同値）以下のプロセス項において，変数 x，x_i，名前 c，プロセス変数 X は束縛される．

$(c)P$, in txP, $(\lambda x_1...x_n.\rho X.G)t_1...t_n$.

α 同値は通常のように定義する．α 同値な表現は同一視する．

定義 3.5 （構造等価）構造等価とは以下の規則から生成される同値関係である．

– $P*Q \sim Q*P$, $(P*Q)*R \sim P*(Q*R)$, $P \sim P*1$

– $P \sim P + P$, $P + Q \sim Q + P$, $(P + Q) + R \sim P + (Q + R)$

– $(\lambda\vec{x}.\rho X.P)\vec{t} \sim P[\vec{t}/\vec{x}, (\lambda\vec{x}.\rho X.P)/X]$

– $(c)P \sim P$ ，ただし c は P に現れない．
– $(c)P[Q] \sim P[(c)Q]$ ，ただし P 中で c は $P[X]$ に現れず，また $P[X]$ 中で X は ρ の有効範囲の中にない．

最後の規則において，表現 $P[\,]$ はただ 1 個の部分式の出現を表す．これは単なる出現であって代入ではない．Q 中の名前や変数は $P[\,]$ 中の束縛子によって束縛されるかもしれない．

注意 3.6 構造等価の最後の規則は，任意の名前の束縛を先頭に移動してよいことを表している．これは文献 4) にはないが，文献 6) によってその正当性が示されている．文献 4) には，束縛された名前を出力する，という概念があるが，この名前束縛を移動する規則により，本研究のパイ計算には，そのような概念が不要になっている．

定義 3.7 （遷移要素）遷移要素とは次の 3 種の形をしたものをいう．

出力	$\mathsf{out}\ cv$
入力	$\mathsf{in}\ cv$
内部	τ

注意 3.8 $\mathsf{out}\ cv$，$\mathsf{in}\ cv$ はそれぞれ，文献 4) の $\bar{x}y$，xy に相当する．文献 4) のパイ計算では，遷移を表す接頭辞に現れる項は名前しかない．一方，π_A には一般の代数の項が現れる．
接頭辞 $\xi tt'$ は，t を評価して名前 c になり t' を評価して値 v になった時点で初めて遷移 ξcv をひき起こす．よって，プロセス項中には一般の項を含む $\mathsf{out}\ tt'$，$\mathsf{in}\ tt'$ が現れるが，遷移には名前 c と値 v だけを含む $\mathsf{out}\ cv$，$\mathsf{in}\ cv$ しか現れない．

定義 3.9 （遷移関係）プロセス項 P からプロセス項 Q への遷移要素 α による**遷移**は，

$$P \xrightarrow{\alpha} Q$$

と書く．これは以下の規則によって定義される．

$$\text{Tau: } \frac{}{\tau P \xrightarrow{\tau} P} \qquad \text{Input: } \frac{}{\mathsf{in}\ cxP \xrightarrow{\mathsf{in}\ cv} P[v/x]}$$

$$\text{Output: } \frac{}{\mathsf{out}\ cvP \xrightarrow{\mathsf{out}\ cv} P}$$

$$\text{Com: } \frac{P \xrightarrow{\mathsf{in}\ cv} Q \quad P' \xrightarrow{\mathsf{out}\ cv} Q'}{P * P' \xrightarrow{\tau} Q * Q'}$$

$$\text{Res: } \frac{P \xrightarrow{\alpha} Q}{(c)P \xrightarrow{\alpha} (c)Q} \text{ ただし } c \text{ は } \alpha \text{ に現れない．}$$

$$\text{Par: } \frac{P \xrightarrow{\alpha} Q}{P * R \xrightarrow{\alpha} Q * R} \qquad \text{Non-Det: } \frac{P \xrightarrow{\alpha} Q}{P + R \xrightarrow{\alpha} Q}$$

$$\text{If: } \frac{P \xrightarrow{\alpha} Q}{\mathsf{if}\ tP \xrightarrow{\alpha} Q} \text{ ただし } A \vdash t = \mathsf{true}$$

$$\text{Calc-I: } \frac{\mathsf{in}\ cxP \xrightarrow{\alpha} Q}{\mathsf{in}\ txP \xrightarrow{\alpha} Q} \text{ ただし } A \vdash t = c$$

$$\text{Calc-O: } \frac{\mathsf{out}\ cvP \xrightarrow{\alpha} Q}{\mathsf{out}\ tt'P \xrightarrow{\alpha} Q} \text{ ただし } A \vdash t = c \text{ かつ } A \vdash t' = v$$

$$\text{Structural: } \frac{P \xrightarrow{\alpha} Q}{P' \xrightarrow{\alpha} Q'} \text{ ただし } P \sim P' \text{ かつ } Q \sim Q'$$

4. 論 理 式

定義 4.1 （前置句）前置句は以下の形をしたものである．

(out, t, t')	out と 2 個の値項 t，t' の三ッ組
(in, t, t')	in と 2 個の値項 t，t' の三ッ組
τ	

前置句 α に対して $[\alpha]$，$\langle\alpha\rangle$ という表現を用いる．$[(\mathsf{in}, t, t')]$ は $[\mathsf{in}, t, t']$ と書き，$\langle(\mathsf{out}, t, t')\rangle$ は $\langle\mathsf{out}, t, t'\rangle$ と書く．
(in, t, t') と (out, t, t') の直観的意味は，$c = t$，$v = t'$ となるような名前 c と値 v においての $\mathsf{in}\ cv$ と $\mathsf{out}\ cv$ である．
遷移要素 $\alpha \equiv \xi cv$ に対して，$[\alpha]$，$\langle\alpha\rangle$ と書いて $[\xi, c, v]$，$\langle\xi, c, v\rangle$ を表す．

定義 4.2 （論理式）述語変数 X は，任意の引数の数 ≥ 0 に応じて十分多くある．

$$\begin{array}{rll}
F ::= & Xt_1 ... t_n \mid 1 \mid t = t \mid \mathsf{Name}(x) & \text{（原子）} \\
\mid & \neg F \mid F \wedge F \mid F \vee F \mid \forall xF \mid \exists xF & \\
& & \text{（加法的）} \\
\mid & (x)F & \text{（遮蔽）} \\
\mid & [\alpha]F \mid \langle\alpha\rangle F & \text{（様相）} \\
\mid & F \multimap F \mid F * F & \text{（乗法的）} \\
\mid & (\lambda x_1 ... x_n \mu XF)t_1 ... t_n & \\
\mid & (\lambda x_1 ... x_n \nu XF)t_1 ... t_n & \text{（再帰）}
\end{array}$$

再帰の構成では，X は F の中に正にしか出現しない．

プロセス変数は，論理式の中にあるときには，引数の数が等しい述語変数と見なす．

記法 4.3 $F \supset F' \equiv (\neg F) \vee F'$, $\top \equiv 1 \supset 1$, $t \neq t' \equiv \neg t = t'$

定義 4.4 （環境）環境 ε とは，各変数 x に対して基底項 $\varepsilon(x)$ を割り当て，引数 n 個の各プロセス変数 X に対して部分集合 $\varepsilon(X) \subset Proc \times GTerm^n$ を割り当てるような函数である．この ε の作用は，値項全体に拡張される．すなわち t に対して $\varepsilon(t) \in GTerm$ を返す函数が自然に作られる．
環境 ε に対して $\varepsilon[t/x]$ とは，$\varepsilon[t/x](x) = t$ でありかつ，$y \neq x$ となる y に対しては $\varepsilon[t/x](y) = \varepsilon(y)$ となるような函数を表す．$\varepsilon[E/X]$ もまた同様である．

定義 4.5 （自由な名前集合） 集合 $C \subset \Sigma_N$ があるとき，C が $E \subset Proc \times GTerm^n$ に対して**自由である**とは，補集合 $\Sigma_N - C$ が有限集合であり，かつ，任意の C 上の置換 σ に対して
$E = \{(P, t_1, t_2, ..., t_n) | (P\sigma, t_1\sigma, t_2\sigma, ...t_n\sigma) \in E\}$
となることをいう．
C が環境 ε に対して**自由である**とは，任意のプロセス変数 X に対して C は $\varepsilon(X)$ に対して自由であり，かつ，任意の変数 x に対して $\varepsilon(x)$ 中に C の元は現れないことをいう．
名前 $c \in \Sigma_N$ が環境 ε に対して**自由**であるとは，ある集合 $C \subset \Sigma_N$ があって $c \in C$ かつ，C が環境 ε に対して自由であることをいう．

定義 4.6 （論理式の解釈） 環境 ε の下での論理式 F の解釈 $[\![F]\!]\varepsilon \subset Proc$ を定義する．これは F による再帰的定義である．ただしここで ε に対して自由な集合 $C \subset \Sigma_N$ が存在することを仮定する．

$[\![Xt_1...t_n]\!]\varepsilon =$
$\{P | \exists Q.\ P \sim Q, (Q, \varepsilon(t_1), ..., \varepsilon(t_n)) \in \varepsilon(X)\}$,
$[\![1]\!]\varepsilon = \{P | P \sim 1\}$

$$[\![t = t']\!]\varepsilon = \begin{cases} Proc, & \varepsilon(t) = \varepsilon(t') \\ \emptyset, & \varepsilon(t) \neq \varepsilon(t') \end{cases}$$

$$[\![\mathsf{Name}(t)]\!]\varepsilon = \begin{cases} Proc, & \varepsilon(t) \in \Sigma_N \\ \emptyset, & \varepsilon(t) \notin \Sigma_N \end{cases}$$

$[\![\neg F]\!]\varepsilon = Proc - [\![F]\!]\varepsilon$, $[\![F \wedge F']\!]\varepsilon = [\![F]\!]\varepsilon \cap [\![F']\!]\varepsilon$,
$[\![F \vee F']\!]\varepsilon = [\![F]\!]\varepsilon \cup [\![F']\!]\varepsilon$

$$[\![\forall x F]\!]\varepsilon = \bigcap_{t \in GTerm} [\![F]\!]\varepsilon[t/x]$$

$$[\![\exists x F]\!]\varepsilon = \bigcup_{t \in GTerm} [\![F]\!]\varepsilon[t/x]$$

$[\![(x)F]\!]\varepsilon = \{P |$
ε に対して自由であり，かつ
F に現れない名前 c と
$Q \in [\![F]\!]\varepsilon[c/x]$ があって $P \sim (c)Q$
$\}$

$[\![[\tau]F]\!]\varepsilon = \{P | P \xrightarrow{\tau} Q$ ならば $Q \in [\![F]\!]\varepsilon\}$

$[\![\langle\tau\rangle F]\!]\varepsilon = \{P |$ ある $Q \in [\![F]\!]\varepsilon$ があって $P \xrightarrow{\tau} Q\}$

$$\begin{cases} [\![[\mathsf{in}, t, t']F]\!]\varepsilon = \\ \quad \{P | P \xrightarrow{\mathsf{in}\ cv} Q \text{ ならば } Q \in [\![F]\!]\varepsilon\} \\ [\![\langle\mathsf{in}, t, t'\rangle F]\!]\varepsilon = \\ \quad \{P | \text{ ある } Q \in [\![F]\!]\varepsilon \text{ があって } P \xrightarrow{\mathsf{in}\ cv} Q\} \\ [\![[\mathsf{out}, t, t']F]\!]\varepsilon = \\ \quad \{P | P \xrightarrow{\mathsf{out}\ cv} Q \text{ ならば } Q \in [\![F]\!]\varepsilon\} \\ [\![\langle\mathsf{out}, t, t'\rangle F]\!]\varepsilon = \\ \quad \{P | \text{ ある } Q \in [\![F]\!]\varepsilon \text{ があって } P \xrightarrow{\mathsf{out}\ cv} Q\} \end{cases}$$

ただしここで c は名前，v は値であって $A \vdash c = \varepsilon(t)$ かつ $A \vdash v = \varepsilon(t')$

$$\begin{cases} [\![[\mathsf{in}, t, t']F]\!]\varepsilon = [\![[\mathsf{out}, t, t']F]\!]\varepsilon = Proc \\ [\![\langle\mathsf{in}, t, t'\rangle F]\!]\varepsilon = [\![\langle\mathsf{out}, t, t'\rangle F]\!]\varepsilon = \emptyset \end{cases}$$

ただしここで $A \vdash c = \varepsilon(t)$ となる名前はない．
$[\![F \multimap F']\!]\varepsilon = \{P | Q \in [\![F]\!]\varepsilon$ ならば $P * Q \in [\![F']\!]\varepsilon\}$
$[\![F * F']\!]\varepsilon = \{P |$ ある Q, R があって
$P \sim Q * R, Q \in [\![F]\!]\varepsilon, R \in [\![F]\!]'\varepsilon\}$
$[\![(\lambda\vec{x}\mu X F)t_1...t_n]\!]\varepsilon = \{P |$
$\{(Q, u_1, ..., u_n) |$
$Q \in [\![F]\!]\varepsilon[E/X, u_1/y_1, ..., u_n/y_n]$
$\} \subset E$
ならば $(P, \varepsilon(t_1), ..., \varepsilon(t_n)) \in E$
$\}$
$[\![(\lambda\vec{x}\nu X F)t_1...t_n]\!]\varepsilon = \{P |$
ある E があって
$E \subset \{(Q, u_1, ..., u_n) |$
$Q \in [\![F]\!]\varepsilon[E/X, u_1/y_1, ..., u_n/y_n]\}$
かつ,$(P, \varepsilon(t_1), ..., \varepsilon(t_n)) \in E$
$\}$

補題 4.7 $P \sim Q \in [\![F]\!]\varepsilon$ ならば $P \in [\![F]\!]\varepsilon$

証明 F に関する帰納法による． ∎

注意 4.8 以下に，プロセス項の論理式への翻訳を定義する．
まず，文献 1) の 3 章にあるような簡単な有限のラベル付遷移系を見てみる．
このラベル付き遷移系では，プロセス P に対して，ちょうどそのプロセスの性質を表す論理式が存在する．

正確には，ある論理式 F が存在してプロセス Q が P と双模倣であることと $Q \models F$ であることが同値になる．

プロセス P_1，P_2，P_3 がそれぞれ各論理式 F_1，F_2，F_3 によってちょうど表現されていて，a，b が相異なるラベルであるとき，プロセス $P = aP_1 + aP_2 + bP_3$ は

$$[a](F_1 \vee F_2) \wedge \langle a\rangle F_1 \wedge \langle a\rangle F_2 \wedge [b]F_3 \wedge \langle b\rangle F_3$$

によってちょうど表現される．

この論理式を作るのは以下の手順による．P から 1 回の遷移で行くプロセスには P_1，P_2，P_3 がある．このうち，a によって行けるプロセスは P_1 と P_2 であり，b によって行けるプロセスは P_3 である．すなわち，ラベルによって行き先のプロセス $\{P_1, P_2\}$ と $\{P_3\}$ とに組分けする．

次に，a で行く際の必然性と可能性を記述する．a では P_1 か P_2 にしか行けないのであるから，$P \models [a](F_1 \vee F_2)$ である．一方で，a では P_1 と P_2 に行けるので，$P \models \langle a\rangle F_1 \wedge \langle a\rangle F_2)$ である．b についても同様の処理を行って $P \models [b]F_3$ と $P \models \langle b\rangle F_3$ を得る．かくして，この 4 個の論理式の連言をとって，先の論理式を得る．

パイ計算の場合にも，プロセスをちょうど表現する論理式を作る際にこの組分けは必要である．しかしこれはそう簡単ではない．$\mathsf{out}\ tt'P$ と $\mathsf{out}\ uu'Q$ は，$t = u \wedge t' = u'$ ならば同じ組に分けられるが，$t \neq u \vee t' \neq u'$ ならば違う組に分けられる．そのどちらになるかは変数に基底項が代入されてみないと決まらない．よって，非決定分岐のあるプロセス項の解釈では，可能なすべての組分けを，その組分けが成立する条件とともに列挙しなければならない．

そのために，ガード項 G に対しては，まず組分けの対象となる分岐成分 $sum(G)$ を定義し，それを使ってガード項 G の翻訳を定める．

定義 4.9 （翻訳）プロセス項 P の論理式への翻訳 $\bar{P}$ を定義する．これはガード項 G の分岐要素 $sum(G)$ に対する翻訳 $\overline{sum(G)}$ と相互再帰的に定義される．

ガード項以外のプロセス項に対しては，

- $\bar{1} \equiv 1$
- $\overline{Xt_1t_2...t_n} \equiv Xt_1t_2...t_n$
- $\overline{P * Q} \equiv \bar{P} * \bar{Q}$
- $\overline{(c)P} \equiv (x)(\overline{P[x/c]})$

以下のガード項に対する定義の中で，$sum(G)$ は後に定義する分岐要素であり，$\overline{sum(G)}$ はその分岐要素に対する翻訳である．

- $\overline{\mathsf{out}\ tt'P} \equiv \overline{sum(\mathsf{out}\ tt'P)}$
- $\overline{\mathsf{in}\ txP} \equiv \forall x.\overline{sum(\mathsf{in}\ txP)}$
- $\overline{\tau P} \equiv \overline{sum(\tau P)}$
- $\overline{G + G'} \equiv \overline{sum(G + G')}$
- $\overline{\mathsf{if}tG} \equiv \overline{sum(\mathsf{if}tG)}$
- $\overline{(\lambda\vec{x}.\rho X.G)\vec{t}} \equiv (\lambda\vec{x}\nu X\overline{sum(G)})\vec{t}$

この定義の中で，再帰の項 G に対する $\overline{sum(G)}$ を参照することはない．

定義 4.10 （分岐要素）ガード項 G に対して，その**分岐要素** $sum(G) \subset Proc$ を定義する．

- $sum(\mathsf{out}\ tt'P) = \{\mathsf{out}\ tt'P\}$
- $sum(\mathsf{in}\ txP) = \{\mathsf{in}\ txP\}$
- $sum(\tau P) = \{\tau P\}$
- $sum(G + G') = sum(G) \cup sum(G')$
- $sum(\mathsf{if}tG) = \{\mathsf{if}tH | H \in sum(G)\}$
- $sum(\ (\lambda\vec{x}.\rho X.G)\vec{t}\ ) = \{H[\lambda\vec{x}.\rho X.G/X, \vec{t}/\vec{x}] | H \in sum(G)\}$

$sum(G)$ の要素はすべて $\mathsf{if}t_1\mathsf{if}t_2...\mathsf{if}t_n\alpha P$ という形をしていて，この P は，再帰の項以外については G よりも小さい．

記法 4.11 項の列 $\vec{t} \equiv t_1, t_2, .., t_n$ と遷移要素 α とプロセス式 P に対して $\mathsf{if}\vec{t}\alpha P \equiv \mathsf{if}t_1\mathsf{if}t_2...\mathsf{if}t_n\alpha P$．

$\vec{t}$ の長さが 0 ならば $\mathsf{if}\vec{t}\alpha P \equiv \alpha P$．

記法 4.12 項の列 $\vec{t} \equiv t_1, t_2, .., t_n$ に対して

$\vec{t} = \mathsf{true} \ \equiv\ t_1 = \mathsf{true} \wedge t_2 = \mathsf{true} \wedge ... \wedge t_n = \mathsf{true}$

$\vec{t} \neq \mathsf{true} \ \equiv\ \neg(\vec{t} = \mathsf{true})$

$\vec{t}$ の長さが 0 ならば $\vec{t} = \mathsf{true} \equiv \top$．

定義 4.13 （分岐要素の翻訳）$\mathsf{if}\vec{t}\alpha P$ という形をしたガード項の有限集合 $S \subset Proc$ に対して，論理式 $\bar{S}$ を定義する．相互再帰的定義の仮定として，各 $\mathsf{if}\vec{t}\alpha P \in S$ に対してすでに論理式 $\bar{P}$ は定義されている．

$\bar{S}$ を定義するために，補助函数 ϕ_o，ϕ_i，ϕ_τ，ψ を用いる．

以下に，S と S' はともに $\mathsf{if}\vec{t}\alpha P$ という形をしたガード項の有限集合である．

このとき，論理式 $\phi_o(S, S')$ は以下のように定義される．

$\phi_o(S, S') \equiv \forall yz.$

$(\bigwedge\{\vec{t} = \mathsf{true} \wedge y = t \wedge z = t' | \mathsf{if}\vec{t}\ \mathsf{out}\ tt'F \in S\}) \supset$

$(\bigwedge\{\vec{t} \neq \mathsf{true} \vee y \neq t \vee z \neq t' | \mathsf{if}\vec{t}\ \mathsf{out}\ tt'F \in S'\}) \supset$

$[\mathsf{out}, y, z](\bigvee\{\bar{P} | \mathsf{if}\vec{t}\ \mathsf{out}\ tt'P \in S\})$

また，論理式 $\phi_i(S, S')$ は以下のように定義される．

$\phi_i(S, S') \equiv \forall yz.$

$(\bigwedge\{\vec{t} = \mathsf{true} \wedge y = t | \mathsf{if}\vec{t}\ \mathsf{in}\ txF \in S\}) \supset$

$(\bigwedge\{\vec{t} \neq \mathsf{true} \vee y \neq t | \mathsf{if}\vec{t}\ \mathsf{in}\ txF \in S'\}) \supset$

$[\mathsf{in}, y, z](\bigvee\{\bar{P}[z/x]|\mathsf{if}\vec{t}\ \mathsf{in}\ txP \in S\})$

ただしここで y と z は $S \cup S'$ に現れない変数である．

論理式 $\phi_\tau(S)$ は以下のように定義される．

$\phi_\tau(S) \equiv (\bigwedge\{\vec{t} = \mathsf{true}|\mathsf{if}\vec{t}\ \tau P \in S\}) \supset [\tau](\bigvee\{\bar{P}|\mathsf{if}\vec{t}\ \tau P \in S\})$

論理式 $\psi(S)$ は以下のように定義される．

$\psi(S) \equiv \wedge\{\vec{t} = \mathsf{true} \supset \langle\alpha\rangle\bar{P}|\mathsf{if}\vec{t}\alpha P \in S\}$

さて，いよいよ $\bar{S}$ を定義する．

$\bar{S} \equiv (\bigwedge\{\phi_o(E, S-E)|E \subset S\}) \wedge (\bigwedge\{\phi_i(E, S-E)|E \subset S\}) \wedge \phi_\tau(S_\tau) \wedge \psi(S)$

補題 4.14 θ は変数に対する代入であって，任意の変数 x に対して $x\theta$ は基底項となるような代入である．$(Q_1, Q_2, ...)$ はガード項の可算無限個の列である．$C = \{c_1, c_2, ...\} \subset \Sigma_N$ は，$\Sigma_N - C$ が有限集合であるような部分集合であり，$X_1, X_2, ...$ はプロセス変数の辞書式順序による列挙である．各 $c_i \in C$，Q_j と任意の変数 x に対して c_i は基底項 $x\theta$ に出現せず，また，プロセス項 Q_j にも出現しない．

ζ は各 X_i に $\lambda x_1 x_2 .. x_{h_i}.\rho X_i.Q_i$ を代入する代入を表す．ここで h_i は X_i の引数の個数である．

ε は，変数 x に $x\theta$ を割り当て，プロセス変数 X_i に集合

$\{(R, t_1, t_2, ..., t_{h_i})|R \sim (X_i\zeta)t_1, t_2, ..., t_{h_i}\}$

を割り当てる環境を表す．

ここで名前 $c \in C$ は ε に対して自由となる．

以上の仮定の許に以下が成り立つ．

任意のプロセス項 P に対して $(c)P[\theta, \zeta] \in [\![\bar{P}]\!]\varepsilon$

証明 P の帰納法による．ここでは $P \equiv (c)Q$ の場合のみを示す．

1. $\bar{P} \equiv (x)\overline{Q[x/c]}$ であるので $[\![\bar{P}]\!]\varepsilon = [\![(x)\overline{Q[x/c]}]\!]\varepsilon$ ．ここで x は Q には現れない．

2. $Q[\theta, \zeta]$ に現れないある名前 $c' \in C$ をとる．この c' は ε に対して自由であり，また $\theta[c'/x]$ という代入は，補題の中の θ に対する条件を満たす．

3. $Q[x/c]$ に対する帰納法の仮定により，

$(Q[x/c])[\theta[c'/x], \zeta] \in [\![\overline{Q[x/c]}]\!]\varepsilon[c'/x]$ ．

よって

$Q[c'/c, \theta, \zeta] \in [\![\overline{Q[x/c]}]\!]\varepsilon[c'/x]$ ．

4. ε に対して自由であり，$Q[x/c]$ に現れない名前 c' によって $Q[c'/c, \theta, \zeta] \in [\![\overline{Q[x/c]}]\!]\varepsilon[c'/x]$ となるので，$(c')(Q[c'/c, \theta, \zeta]) \in [\![(x)\overline{Q[x/c]}]\!]\varepsilon$ ．

5. $P[\theta, \zeta] \equiv (c)Q[\theta, \zeta]$ と $(c')(Q[c'/c])[\theta, \zeta] \equiv (c')(Q[c'/c, \theta, \zeta])$ は α 同値であるので，

$P[\theta, \zeta] \in [\![(x)\overline{Q[x/c]}]\!]\varepsilon = [\![P]\!]\varepsilon$ ． ∎

定理 4.15 （解釈の健全性） P に自由な変数およびプロセス変数がないならば $P \in [\![P]\!]\varepsilon$.

証明 $P \equiv P[\theta, \zeta]$ ．よって補題 4.14 よりただちにいえる． ∎

5. 推 論 規 則

注意 5.1 以下に，証明系を定義する．

論理を公理化することの意義は，$P \models F$ であって $\vdash F \supset G$ が証明可能ならばただちに $P \models G$ が得られることにある．

公理化の方法としては，ゲンツェンの LK と同様の方法をとる．本研究で定義する証明系の式は LK よりも複雑である．たとえば

$F_1, F_2, F_3|F_4, F_5|F_6 \Rightarrow F_7, F_8$

のような式を取り扱う．この式の直観的な意味は，

$P \models F_1 \wedge F_2 \wedge F_3,\ Q \models F_4 \wedge F_5,\ R \models F_6$

ならば

$P * Q * R \models F_7 \vee F_8$

ということである．すなわち，各プロセスの性質から，それを並列合成したものの性質を推論できるようになっている．

定義 5.2 （加法的列）加法的列とは，0 個以上の加法的列を読点「,」でつないだ表現である．加法的列に対しては Γ，Γ'，Γ_i，Δ のような文字を用いる．

定義 5.3 （乗法的列）乗法的列とは，0 個以上の加法的列を縦線「|」でつないだ表現である．乗法的列に対しては $\tilde{\Gamma}$，$\tilde{\Gamma}'$，$\tilde{\Gamma}_i$ のような文字を用いる．

注意 5.4 加法的列 $F_1, F_2, ..., F_n$ の直観的な意味は，ある 1 つのプロセス項に対する n 個の性質である．一方，乗法的列 $\Gamma_1|\Gamma_2|...|\Gamma_n$ の直観的な意味は，n 個のプロセス項 $P_1, P_2, ..., P_n$ があって，各 P_i に対する性質が Γ_i となっているようなものである．

記法 5.5 加法的列 $\Gamma =$「$F_1, F_2, ..., F_n$」に対して $\{\Gamma\}$ は集合 $\{F_1, F_2, ..., F_n\}$ を表す．

記法 5.6 空の加法的列と空の乗法的列を区別しなければならない．空の加法的列は $\emptyset$ と書き，空の乗法的列は何も書かないことによって表す．

定義 5.7 （式）式とは乗法的列と加法的列を $\Rightarrow$ の記号でつないだ，$\tilde{\Gamma} \Rightarrow \Delta$ という形をした表現である．

定義 5.8 （式の解釈）加法的列，乗法的列，式に対して，その解釈を定義する．

加法的列は式の左辺右辺双方に出現するが，その解釈は異なる．$[\![\Gamma]\!]^l\varepsilon \subset Proc$ は式の左辺における Γ の解釈であり，$[\![\Gamma]\!]^r\varepsilon \subset Proc$ は式の右辺における Γ の解釈である．

$[\![F_1, F_2, ..., F_n]\!]^l\varepsilon = \cap_{i=1,2,...,n}[\![F_i]\!]\varepsilon$

$[\![F_1, F_2, ..., F_n]\!]^r\varepsilon = \cup_{i=1,2,...,n}[\![F_i]\!]\varepsilon$

空な加法的列に対しては $[\![\emptyset]\!]^l\varepsilon = Proc$，$[\![\emptyset]\!]^r\varepsilon = \emptyset$ ．

乗法的列に対しては

$[\![\Gamma_1|\Gamma_2|...|\Gamma_n]\!]\varepsilon = \{P_1 * P_2 * ... * P_n | P_i \in [\![\Gamma_i]\!]^l\varepsilon\}$

$\tilde{\Gamma}$ が空ならば $[\![\tilde{\Gamma}]\!]\varepsilon = 1$

環境 ε において式 $\tilde{\Gamma} \Rightarrow \Delta$ が成り立つとは，$[\![\tilde{\Gamma}]\!]\varepsilon \subset [\![\Delta]\!]^r\varepsilon$ となることをいう．

記法 5.9 推論規則を記述するための記法をいくつか定義する．

加法的列 $\Gamma =$「$F_1, F_2, ..., F_n$」に対して，

$[\alpha]\Gamma =$「$[\alpha]F_1, [\alpha]F_2, ..., [\alpha]F_n$」，

$\langle\alpha\rangle\Gamma =$「$\langle\alpha\rangle F_1, \langle\alpha\rangle F_2, ..., \langle\alpha\rangle F_n$」．

乗法的列 $\tilde{\Gamma} =$「$\Gamma_1|\Gamma_2|...|\Gamma_n$」に対して，

$[\alpha]\tilde{\Gamma} =$「$[\alpha]\Gamma_1|[\alpha]\Gamma_2|...|[\alpha]\Gamma_n$」，

$\langle\alpha\rangle\tilde{\Gamma} =$「$\langle\alpha\rangle\Gamma_1|\langle\alpha\rangle\Gamma_2|...|\langle\alpha\rangle\Gamma_n$」．

加法的列 Γ に対して，$\Gamma^{\mathsf{out}\ tt'}$，$\Gamma^{\mathsf{in}\ tt'}$，$\Gamma^{\tau}$ は加法的列であり，次の集合の等式を満たす．加法的列の中の順番は本質に関係ないので，辞書式順序と定義しておく．

$\{\Gamma^{\mathsf{out}\ tt'}\} = \{t \neq u \vee t' \neq u' \vee F | [\mathsf{out}, u, u']F \in \{\Gamma\}\}$

$\{\Gamma^{\mathsf{in}\ tt'}\} = \{t \neq u \vee t' \neq u' \vee F | [\mathsf{in}, u, u']F \in \{\Gamma\}\}$

$\{\Gamma^{\tau}\} = \{F | [\tau]F \in \{\Gamma\}\}$

$P[\lambda x_1 x_2 ... x_n F/X]$ は，P 中の $Xt_1t_2..t_n$ という形をした部分式を $F[t_1/x_1, t_2/x_2, ..., t_n/x_n]$ で置き換えて得られる論理式を表す．

定義 5.10 （推論規則）

$$\frac{}{\Gamma, F \Rightarrow F, \Delta}$$

$$\frac{\Gamma_1|\Gamma_2|...|\Gamma_n \Rightarrow \Delta}{\Gamma_1'|\Gamma_2'|...|\Gamma_n' \Rightarrow \Delta'}$$

ただし各 i に対して $\{\Gamma_i\} \subset \{\Gamma_i'\}$，かつ $\{\Delta_i\} \subset \{\Delta_i'\}$

$$\frac{\Gamma_1|\Gamma_2|...|\Gamma_n \Rightarrow \Delta}{\Gamma_{\sigma 1}|\Gamma_{\sigma 2}|...|\Gamma_{\sigma n} \Rightarrow \Delta}$$

ただし σ は $\{1, 2, ..., n\}$ 上の置換

$$\frac{\tilde{\Gamma} \Rightarrow \Delta}{\tilde{\Gamma}|1 \Rightarrow \Delta} \qquad \frac{}{\Rightarrow 1}$$

$$\frac{\tilde{\Gamma}|\Gamma \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, t = t' \Rightarrow \Delta} \qquad \frac{}{\tilde{\Gamma} \Rightarrow t = t', \Delta}$$

ただし $A \vdash t = t'$

$$\frac{}{\tilde{\Gamma}|\Gamma, \mathsf{true} = \mathsf{false} \Rightarrow \Delta}$$

$$\frac{\tilde{\Gamma}[t'/x]|\Gamma[t'/x] \Rightarrow \Delta[t'/x]}{\tilde{\Gamma}[t/x]|\Gamma[t/x], t = t' \Rightarrow \Delta[t/x]} \qquad \frac{\tilde{\Gamma}|\Gamma|\Gamma', t = t' \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, t = t'|\Gamma' \Rightarrow \Delta'}$$

$$\frac{\tilde{\Gamma}|\Gamma \Rightarrow t = t', \Delta}{\tilde{\Gamma}|\Gamma, t \neq t' \Rightarrow \Delta} \qquad \frac{\tilde{\Gamma}|\Gamma, t = t' \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma \Rightarrow t \neq t', \Delta}$$

$$\frac{\tilde{\Gamma}|\Gamma \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, \mathsf{Name}(t) \Rightarrow \Delta} \qquad \frac{}{\tilde{\Gamma} \Rightarrow \mathsf{Name}(t), \Delta}$$

ただし，ある $c \in \Sigma_N$ があって $A \vdash t = c$

$$\frac{}{\tilde{\Gamma}|\Gamma, \mathsf{Name}(t') \Rightarrow \Delta}$$

ただし，ある $v \in V - \Sigma_N$ があって $A \vdash t = v$

$$\frac{\tilde{\Gamma}|\Gamma|\Gamma', \mathsf{Name}(t) \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, \mathsf{Name}(t)|\Gamma' \Rightarrow \Delta'}$$

$$\frac{\tilde{\Gamma}|\Gamma \Rightarrow \mathsf{Name}(t), \Delta}{\tilde{\Gamma}|\Gamma, \neg\mathsf{Name}(t) \Rightarrow \Delta} \qquad \frac{\tilde{\Gamma}|\Gamma, \mathsf{Name}(t) \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma \Rightarrow \neg\mathsf{Name}(t), \Delta}$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\Gamma, \neg F \Rightarrow \Delta} \qquad \frac{\Gamma, F \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta}$$

$$\frac{\tilde{\Gamma}|\Gamma, F \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, F \wedge G \Rightarrow \Delta} \qquad \frac{\tilde{\Gamma}|\Gamma, F \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, G \wedge F \Rightarrow \Delta}$$

$$\frac{\tilde{\Gamma} \Rightarrow F, \Delta \qquad \tilde{\Gamma} \Rightarrow G, \Delta}{\tilde{\Gamma} \Rightarrow F \wedge G, \Delta}$$

$$\frac{\tilde{\Gamma}|\Gamma, F \Rightarrow \Delta \qquad \tilde{\Gamma}|\Gamma, G \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, F \vee G \Rightarrow \Delta}$$

$$\frac{\tilde{\Gamma} \Rightarrow F, \Delta}{\tilde{\Gamma} \Rightarrow F \vee G, \Delta} \qquad \frac{\tilde{\Gamma} \Rightarrow F, \Delta}{\tilde{\Gamma} \Rightarrow G \vee F, \Delta}$$

$$\frac{\tilde{\Gamma}|\Gamma, F[t/x] \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, \forall x F \Rightarrow \Delta} \qquad \frac{\tilde{\Gamma} \Rightarrow F, \Delta}{\tilde{\Gamma} \Rightarrow \forall x F, \Delta}$$

ただしここで x は下式には現れない

$$\frac{\tilde{\Gamma}|\Gamma, F \Rightarrow \Delta}{\tilde{\Gamma}|\Gamma, \exists x F \Rightarrow \Delta} \qquad \frac{\tilde{\Gamma} \Rightarrow F[t/x], \Delta}{\tilde{\Gamma} \Rightarrow \exists x F, \Delta}$$

ただしここで x は下式には現れない

$$\frac{\tilde{\Gamma}|\Gamma, F, \mathsf{Name}(x) \Rightarrow G_1, ..., G_n, x = t_1, ..., x = t_m, \Delta}{\tilde{\Gamma}|\Gamma, (x)F \Rightarrow (x)G_1, (x)G_2, ..., (x)G_n, \Delta}$$

ここで x は下式に現れず，またどの t_i にも現れない

$$\frac{\tilde{\Gamma}|\Gamma, \mathsf{Name}(x) \Rightarrow F_1, F_2, ..., F_n, x = t_1, ..., x = t_m, \Delta}{\tilde{\Gamma}|\Gamma \Rightarrow (x)F_1, (x)F_2, ..., (x)F_n, \Delta}$$

ここで x は下式に現れず，またどの t_i にも現れない

$$\frac{\tilde{\Gamma}|\Gamma, F|\Gamma', F \Rightarrow \Delta}{\tilde{\Gamma}|[\mathsf{out}, t, t']\Gamma, \langle\mathsf{out}\ tt'\rangle F|[\mathsf{in}, t, t']\Gamma', \langle\mathsf{in}\ tt'\rangle F \Rightarrow \langle\tau\rangle\Delta}$$

$$\frac{\tilde{\Gamma}|\Gamma, F \Rightarrow \Delta}{\tilde{\Gamma}|[\alpha]\Gamma, \langle\alpha\rangle F \Rightarrow \langle\alpha\rangle\Delta}$$

$\tilde{\Gamma}$ 内の各 Γ_i に対して

$$\tilde{\Gamma} - \Gamma_i|\Gamma_i^{\tau} \Rightarrow F, \Delta$$

$\tilde{\Gamma}$ 内の相異なる加法列 Γ_i，Γ_j および $[\mathsf{out}, t, t']F \in \{\Gamma_i\}$ があるような (t, t') の，その任意の組合せに対して

$$\frac{\tilde{\Gamma} - \Gamma_i - \Gamma_j|\Gamma_i^{\mathsf{out}\ tt'}|\Gamma_j^{\mathsf{in}\ tt'} \Rightarrow F, \Delta}{\tilde{\Gamma} \Rightarrow [\tau]F, \langle\tau\rangle\Delta}$$

$$\frac{\tilde{\Gamma}\text{ 中の各 }\Gamma_i\text{ に対して }(\tilde{\Gamma}-\Gamma_i)|\Gamma_i^{\xi tt'}\Rightarrow F,\Delta}{\tilde{\Gamma}\Rightarrow[\xi,t,t']F,\langle\xi,t,t'\rangle\Delta}$$

ここで ξ は out または in

$$\frac{\tilde{\Gamma}\Rightarrow F\qquad\tilde{\Gamma}'|G\Rightarrow\Delta}{\tilde{\Gamma}|\tilde{\Gamma}'|F\multimap G\Rightarrow\Delta}\qquad\frac{\tilde{\Gamma}|F\Rightarrow G}{\tilde{\Gamma}\Rightarrow F\multimap G}$$

$$\frac{\tilde{\Gamma}|F|G\Rightarrow\Delta}{\tilde{\Gamma}|F*G\Rightarrow\Delta}\qquad\frac{\tilde{\Gamma}\Rightarrow F\qquad\tilde{\Gamma}'\Rightarrow G}{\tilde{\Gamma}|\tilde{\Gamma}'\Rightarrow F*G}$$

$$\frac{\tilde{\Gamma}|\Gamma,P[M/X,\vec{t}/\vec{x}]\Rightarrow\Delta}{\tilde{\Gamma}|\Gamma,M\vec{t}\Rightarrow\Delta}\qquad\frac{\tilde{\Gamma}\Rightarrow P[M/X,\vec{t}/\vec{x}],\Delta}{\tilde{\Gamma}\Rightarrow M\vec{t},\Delta}$$

ただしここで $M\equiv\lambda\vec{x}\mu XP$

$$\frac{\Gamma,P[\lambda\vec{x}F/X]\Rightarrow F,\Delta}{\Gamma,M\vec{t}\Rightarrow F[\vec{t}/\vec{x}],\Delta}$$

ただしここで $M\equiv\lambda\vec{x}\mu XP$，かつ，述語変数 X は Γ にも Δ にも現れず，各変数 $\vec{x}$ は下式に現れない

$$\frac{\tilde{\Gamma}|\Gamma,P[N/X,\vec{t}/\vec{x}]\Rightarrow\Delta}{\tilde{\Gamma}|\Gamma,N\vec{t}\Rightarrow\Delta}\qquad\frac{\tilde{\Gamma}\Rightarrow P[N/X,\vec{t}/\vec{x}],\Delta}{\tilde{\Gamma}\Rightarrow N\vec{t},\Delta}$$

ただしここで $N\equiv\lambda\vec{x}\nu XP$

$$\frac{\Gamma,F\Rightarrow P[\lambda\vec{x}F/X],\Delta}{\Gamma,F[\vec{t}/\vec{x}]\Rightarrow N\vec{t},\Delta}$$

ただしここで $N\equiv\lambda\vec{x}\nu XP$，かつ，述語変数 X は Γ にも Δ にも現れず，各変数 $\vec{x}$ は下式に現れない

$$\frac{\tilde{\Gamma}\Rightarrow F,\Delta\qquad\Gamma,F\Rightarrow\Delta'}{\tilde{\Gamma}|\Gamma\Rightarrow\Delta,\Delta'}\qquad\frac{\tilde{\Gamma}\Rightarrow F\qquad\tilde{\Gamma}'|F\Rightarrow\Delta}{\tilde{\Gamma}|\tilde{\Gamma}'\Rightarrow\Delta}$$

定理 5.11 （健全性） $\tilde{\Gamma}\Rightarrow\Delta$ が証明可能ならば $[\![\tilde{\Gamma}]\!]\varepsilon\subset[\![\Delta]\!]^r\varepsilon$.

証明 証明図の帰納法による．ここではこの推論のみを示す．

$$\frac{\tilde{\Gamma}\text{ 中の各 }\Gamma_i\text{ に対して }(\tilde{\Gamma}-\Gamma_i)|\Gamma_i^{\xi tt'}\Rightarrow F,\Delta}{\tilde{\Gamma}\Rightarrow[\xi,t,t']F,\langle\xi,t,t'\rangle\Delta}$$

ここで ξ は out または in

この推論で，上式のすべての i で

$$[\![(\tilde{\Gamma}-\Gamma_i)|\Gamma_i^{\xi tt'}]\!]\varepsilon\subset[\![F,\Delta]\!]\varepsilon$$

かつ，下式において

$$[\![\tilde{\Gamma}]\!]\varepsilon\not\subset[\![[\xi,t,t']F,\langle\xi,t,t'\rangle\Delta]\!]\varepsilon$$

ならば矛盾することを示す．

1. $\tilde{\Gamma}\equiv\Gamma_1|...|\Gamma_n$ ，$\Delta\equiv G_1,...,G_m$ と書く．
$$[\![\tilde{\Gamma}]\!]\varepsilon\not\subset[\![[\xi,t,t']F,\langle\xi,t,t'\rangle\Delta]\!]\varepsilon$$
なので，あるプロセス項 $P\equiv P_1*...*P_n$ があって
– 各 i について $P_i\in[\![\Gamma_i]\!]\varepsilon$
– $P\notin[\![[\xi,t,t']F]\!]\varepsilon$
– 各 j について $P\notin[\![\langle\xi,t,t'\rangle G_j]\!]\varepsilon$
2. $P\notin[\![[\xi,t,t']F]\!]\varepsilon$ なので，ある P' があって $P\overset{\xi cv}{\to}P'\notin[\![F]\!]\varepsilon$ ．ただしここで $A\vdash c=\varepsilon(t)$ ，$A\vdash v=\varepsilon(t')$ ．
3. $P\overset{\xi cv}{\to}P'\notin[\![F]\!]\varepsilon$ なので，ある i があって $P_i\overset{\xi cv}{\to}Q$ かつ $P'\sim P_1*...*P_{i-1}*Q*P_{i+1}*...*P_n$ ．
4. $P_i\in[\![\Gamma_i]\!]\varepsilon$ なので，各 $[\xi,u,u']H\in\Gamma_i$ に対して，$A\vdash\varepsilon(t)=\varepsilon(u)$ ，$A\vdash\varepsilon(t')=\varepsilon(u')$ ならば $P_i\in[\![[\xi,t,t']H]\!]\varepsilon$ ，すなわち $Q\in[\![H]\!]\varepsilon$ ．ゆえに $Q\in[\![(t=u\wedge t'=u')\supset H]\!]\varepsilon$ ．したがって $Q\in[\![\Gamma_i^{\xi tt'}]\!]\varepsilon$ ．
5. $P_1*...*Q*...*P_n\in[\![(\tilde{\Gamma}-\Gamma_i)|\Gamma_i^{\xi tt'}]\!]\varepsilon$ ．ゆえに $P_1*...*P_{i-1}*Q*P_{i+1}*...*P_n\in[\![F,\Delta]\!]\varepsilon$ ．6. 補題 4.7 によって $P'\in[\![F,\Delta]\!]\varepsilon$ ．
7. $P'\notin[\![F]\!]\varepsilon$ なので $P'\in[\![\Delta]\!]\varepsilon$ ．ゆえにある j があって $P\overset{\xi cv}{\to}P'\in[\![G_j]\!]\varepsilon$ ．これは $P\notin[\![\langle\xi,t,t'\rangle G_j]\!]\varepsilon$ に矛盾する．∎

注意 5.12 この体系ではカット除去定理が成り立たないと予想される．

6. 例

write，*read*，*return* という 3 個の名前を使って，このようなプロセス項を定義する．

$P=M0$

$M=\lambda x.\rho X.$

$\quad$ in $write\ y\ Xy+$ in $read\ z$ out $return\ x\ Xx$

すると

$$P\overset{\text{in }write\,t}{\to}Mt\overset{\text{in }read\,u}{\to}\quad\text{out }return\,t\overset{}{\to}Mt$$

という遷移関係が成り立つ．

ここで，

$\bar{P}\Rightarrow\forall yz\ [\mathsf{in},write,y][\mathsf{in},read,z]\langle\mathsf{out},return,y\rangle\top$

が導出可能となる．

7. 関連研究との比較

パイ計算や，その他の動的プロセスに対する論理式の研究にはすでにいくつかある．

文献 1) では，並列合成を含むプロセス代数と，その性質を表す論理式が定義されている．この文献での目的はプロセスの等価性の定義である．2 つのプロセスがあって，すべての論理式に対して同じように振る舞う，すなわち $\forall F.\ P\models F\iff Q\models F$ という関係は同値関係となり，これは強双模倣の関係と一致する．この文献ではこのようなプロセス間の同値関係の，プロセス代数としての公理化が議論されている．本研究の目的はプロセスの同値関係ではないが，本研究の

論理式も強双模倣の関係の同値類を分離する．すなわち π_A の閉プロセス項 P と Q が強双模倣でないことと，ある閉論理式 F があって $P \models F$ かつ $Q \not\models F$ となることとは同値である．また適当に論理式を制限することによって，ちょうど弱双模倣の同値類を分離するような論理式の部分集合を作ることはできる．

文献 2) では，パイ計算のプロセスの性質を表す論理式が定義されている．様相演算子には $[x{=}y]$ ，$\langle \bar{x}y \rangle$ ，$\langle \bar{x}(y) \rangle$ ，$\langle xy \rangle$ ，$\langle x(y) \rangle$ ，$\langle x(y) \rangle^E$ ，$\langle x(y) \rangle^L$ がある．$[x{=}y]$ ，$\langle \bar{x}y \rangle$ ，$\langle xy \rangle$ はそれぞれ本研究の $x = y \supset ...$，$\langle \mathsf{out}, x, y \rangle$，$\langle \mathsf{in}, x, y \rangle$ に相当する．$\langle \bar{x}(y) \rangle$，$\langle x(y) \rangle$，$\langle x(y) \rangle^E$ はそれぞれ本研究では $\forall y.\langle \mathsf{out}, x, y \rangle$，$\exists y.\langle \mathsf{in}, x, y \rangle$，$\forall y.\langle \mathsf{in}, x, y \rangle$ と表現できる．$\langle x(y) \rangle^L$ は本研究の論理式では表現できない．したがって，本研究の論理式で分離できるのは，文献 2) における関係 $\sim_E$ の同値類である．文献 2) の論理式には再帰がなく，よって停止性や活性は記述できない．

文献 3) では，パイ計算のプロセスの性質を表す論理式が定義されている．ここでは論理式に再帰がある．ここでの論理式の構文に代数仕様の項と一階述語論理，そして並列合成に関する論理結合子を加えると，本研究の論理式とほぼ同等になる．また文献 3) では充足問題の決定可能性が議論されている．すなわち，ある種のプロセス P と論理式 F に対して，$P \models F$ か否かは決定可能である．

以上の文献には，論理式の構文に並列合成を表す論理結合子がない．並列合成を表す乗法的論理結合子「$*$」と「$-\!\circ$」が本研究の特長である．

パイ計算の停止性や活性を対象にする演繹体系としては，型体系もまた多く提案されている[4]．多くの型体系では，構成子の数や推論規則の数が小さく，その分表現力が制限されていた．たとえば文献 7) では，型の構成は限定されており，この通信ポートはどのような型の値を送受信するか，ということしか記述できない．

それに対し，本研究の論理体系は一階述語論理の記述をすべて含み，また並列合成も記述できる．

8. 今後の課題

本研究の論理体系は，記述力が豊かである反面，論理記号が多くなり，それに比例して推論規則の量も多く，かつ，煩雑になった．ウェブサービスの設計や検証等の個別問題については，このような記述力をすべて必要とするのではないかもしれない．個別問題に応じ訂正な記述力を持った論理式の部分集合を切り出し，それに対応して論理体系を簡潔なものにすることが今後の課題である．

また，ゲンツェンのシーケント計算流の形式をとりながら，この体系ではカット除去定理が成り立たないであろうと予想される．これは，並列合成に関するものである乗法的論理演算 $-\!\circ$，$*$ と，それ以外の論理演算との証明論的関係が未解明であることによる．この点を明らかにすることもまた今後の課題である．

参 考 文 献

1) Hennessy, M. and Milner, R.: Algebraic laws for nondeterminism and concurrency, *J. ACM*, Vol.32, No.1, pp.137–161 (1985).
2) Milner, R., Parrow, J. and Walker, D.: Modal logics for mobile processes, *Theoretical Computer Science*, Vol.114, pp.149–171 (1993).
3) Dam, M.: Model Checking Mobile Processes, *Information and Computation*, Vol.129, No.1, pp.35–51 (1996).
4) Sangiogi, D. and Walker, D.: *The π-calculus, a therory of mobile processes*, Cambridge University Press (2001).
5) Mano, K. and Kawabe, Y.: Nepi: syntax, semantics and implementation, *5th World Multi-Conference on Systemics, Cybernetics and Informatics* (*SCI '01*) (2001).
6) 河辺義信，真野 健，堀田英一，小暮 潔：パイ計算の名前制限の名前生成による実装の正しさ，信学論，Vol.J85-DI, No.3, pp.249–261 (2002).
7) Kobayashi, N.: Type-Based Information Flow Analysis for the Pi-Calculus, Technical report, TR03-0007, Department of Computer Science, Tokyo Institute of Technology, Tokyo (2003).

(平成 16 年 12 月 20 日受付)
(平成 17 年 4 月 28 日採録)

竹内 泉（正会員）

1968 年生．1993 年東北大学大学院工学研究科情報工学専攻修士課程修了．2005 年より産業技術総合研究所研究員．パラメトリシティの理論，計算可能性数学，様相論理等を研究．