線形論理を用いた逐次型プロセスの並列実行

村 上 昌 己[†]

本稿では逐次的に実行される部分を含む並行プロセスを並列実行する方法について考察する．ここで対象とする並行プロセスは線形論理で記述されたものであり，その操作的意味論は LK 風のシークエント計算の推論規則によって記述される．本稿ではこの体系に並列実行のための規則として，プロセス定義の展開のための規則と値を将来実行される部分に伝搬して分岐を枝刈りするための規則を導入する．これらのプロセスの並列実行のための規則を用いた推論が，線形論理におけるシークエント計算での証明の枠組みの中で記述できることを示す．またこれらの規則を用いた並列実行に対して，同等な計算とみなせる逐次実行が存在することを示す．

Parallel Execution of Sequential Process Using Linear Logic

MASAKI MURAKAMI[†]

This paper presents a method for parallel execution of sequential processes that are represented using linear logic. The operational semantics of processes are provided as a set of inference rules of sequent calculus as LK. A rule for unfolding of process definitions and a rule for pruning of branches and forward propagation of values are introduced. Inference using these new rules for parallel execution of processes is represented as a proof of linear logic. This paper also shows that for every parallel execution there exists an equivalent sequential execution.

1. はじめに

線形論理[1]は，論理にリソースの概念を取り入れたもとの考えられ，並行計算の関係について様々な結果が示され，線形論理を用いて並行・分散環境での計算を記述する様々な方法が提案されてきた[3),4),7)]．

そのような体系でのひとつとして[6)]のように，LK 風のシークエント計算での証明の構築によって並行プロセスの操作的意味論を記述する方法がある．プロセスの起動及びメッセージの発信は $\otimes$ 左導入の推論によって，またメッセージの受信は $\multimap$ 左導入の推論規則によって表現される．このような線形論理による並行計算の記述の特徴のひとつとして，並行計算を構成する基本概念であるメッセージとプロセスが，いずれも論理式の形式で記述されるという点がある．

このような枠組みではさらに，プロセスの定義も論理式によって記述可能であるという特徴もある．すなわち CCS 等の $P \stackrel{\mathrm{def}}{=} a.P + b.Q$ のような記法は，論理式としては $!(P \equiv a \multimap P \mathbin{\&} b \multimap Q)$ のようにあらわすことができる． ここでこの論理式から含意される $P \multimap (a \multimap P \mathbin{\&} b \multimap Q)$ と P から $a \multimap P \mathbin{\&} b \multimap Q$ を推論することによって，P というプロセスが呼び出されたときそれが定義本体 の $a \multimap P \mathbin{\&} b \multimap Q$ に展開されることを表現することができる☆． このような枠組みではプロセスの定義は実行中のプロセスと別の場所に記憶されて必要に応じて参照されるものというより，実行中のプロセスや配送途中のメッセージと同じように存在し，呼び出されたプロセスとプロセス定義が相互に通信を行なうことにより，呼び出されたプロセスがその定義に置き換えられると考えられる．このような状況は，近年の WWW などを用いた分散環境でのプログラムの呼び出しや実行の様子を，自然に表現するものと捉えることができる．したがって，線形論理によって並行計算を記述することにより，近年の分散環境を用いた計算環境の形式化が可能になるであろうことが期待できる．

例えば WWW を通した計算環境の特徴に，プログラムのコードが実行時にデータとして転送される現象がある．さらに転送されたコードが，実行前に最適化されたり実行環境に合わせて書き換えられた後，実行

† 岡山大学 工学部
Faculty of Engineering, Okayama University

☆ 9) 等ではプログラム節を $!\forall\bar{x}(G \multimap A)$ としているが，これは Prolog の `A :- G` に対応しており，$\multimap$ の向きがここで用いる手法では逆になっていることに注意されたい．

されるようなことも考えられる．さらに実行が始まった後に，変更が加わることも一般には考えられる．これらのことより，プロセスの（手続きの）定義が実行時にデータとして扱われることを形式的に表現できることは，分散環境での計算を形式的に表現する体系として有利な特徴あると考えられる．

我々は，先に線形論理を用いた並行プロセスの記述体系において，実行時にプロセスの定義を最適化・変更する方法について考察を進めてきた.[5]

本稿では，同様に線形論理を用いた並行プロセスの記述体系を対象にする．ここでは逐次的に実行される部分を含むプロセスについて，将来起動されるプロセスの実行をあらかじめ開始しておくことにより，並列に実行する方法を提案する．例えば $m_1(x) \multimap (m_2(y) \multimap P(x) \otimes Q(x,y))$ のようなプロセスは，メッセージ m_1, m_2 の両方を受け取ってからでないと $P(x)$ も $Q(x,y)$ も起動することはできない．しかしながら $P(x)$ は，$m_1(x)$ だけを受け取った後に x の値によって，以降どのように実行されるかは部分的に定まっている場合もある．そのような場合，x の値が判明した段階でその値を前方に伝搬し，$P(x)$ をその定義の本体にあらかじめ展開し，可能であれば分岐の枝刈りを行なっておくことにより，m_2 が届いた後の動作の時間を短縮することができることが期待できる．本稿ではこのように，逐次性を含むプロセスを，本来の動作を逸脱しない範囲で先回りできるところはあらかじめ見込み計算を行っておくように操作的意味論を拡張するものである．

本稿で提案する方法は先の [6] の体系をもとにしている．すなわちこの体系に，プロセス定義を論理式として扱う規則を加えたものに，並列実行のための規則として，プロセス定義の展開のための規則と，値の前方伝搬と分岐の枝刈りのための規則を導入する． これらの規則を用いた推論が LL の体系で可能であることから，見込み計算によるプロセスの並列実行が線形論理におけるシークエント計算での証明の枠組みの中で記述できることを示す．またこれらの規則を用いた並列実行に対して，外部から観測した際に同等な計算とみなせる逐次実行が存在することを示す．

2. 準　　備

2.1 構　　文

本稿で用いる線形論理の体系の構文を以下に導入する．個体定数記号，関数記号および変数記号 $x, y, z, x_1, x_2, \ldots$ 等から構成される項は通常のように定義する． 述語記号は $p, q, p_1, p_2, \ldots$ のように表記する．関数記号と述語記号にはそれぞれ引数の数が定められているものとする．論理記号としては 乗法的論理積 $\otimes$, 加法的論理積 $\&$, 含意 $\multimap$, 同値 $\equiv$, 複製 $!$, 任意束縛 $\forall$, 真 1 などを用いる．

ここでは，これらの記号から構成される任意の論理式を扱うわけではない．すなわちプロセス，メッセージ又はプロセス定義として意味をもつ表現のみを扱う．以下では 変数（又は項）の並び $x_1, x_2, \ldots x_n$（又は $t_1, t_2, \ldots t_n$） を $\bar{x}$（又は $\bar{t}$）と表記する．

定義 2.1 $t_1, t_2, \ldots t_n$ を上記の定数, 関数記号, 変数記号から構成される項とする．このとき p を n 引数の述語記号とするとき， $p(t_1, t_2, \ldots, t_n)$ および 1 は素プロセス式である．

$p(t_1, t_2, \ldots, t_n)$ はプロセスの呼び出しを意味する．真をあらわす素論理式 1 は，何も動作しないプロセスである．メッセージは $\mathsf{m}(D, C)$ という 2 引数の特別な述語記号をもつ論理式（以下メッセージ式とよぶ）で表現する．ここで第一引数 D はメッセージの宛名，第二引数はメッセージの内容をあらわす．第一引数が具体化されたメッセージ式をメッセージとよぶ．

定義 2.2

(1) 素プロセス式はプロセス式である．

(2) A がプロセス式，B がプロセス式又はメッセージであるとき，$A \otimes B$ （又は $B \otimes A$） はプロセス式である．

(3) $1 \leq i \leq n$ とする．P_i がプロセス式，m_i がメッセージ式であるとき，
$\forall x(m_1 \multimap P_1 \& \cdots \& m_n \multimap P_n)$
はプロセス式である．

(4) m がメッセージ式であるとき，$!m$ はプロセス式である．

(2) は 直観的には A, B がともにプロセスであるとき，$A \otimes B$ は並行に実行される二つのプロセス A と B を起動するプロセスを意味する．一方がプロセス P で片方がメッセージ m であるとき，$m \otimes P$ は，メッセージ m を発信してそれ以降 P として動作するプロセスと考えられる． 以下では $A \otimes B$ と $B \otimes A$ は同一視することにする．

含意 $\multimap$, 加法的論理積 $\&$ は無制限には使用することなく，(3) のような形で用いる． ここで $\forall x(m_1 \multimap P_1 \& \cdots \& m_n \multimap P_n)$ は，受け取った $m_i\{t/x\}$ に応じて以降の動作を $P_i\{t/x\}$ と定めるプロセスと考えることができる．ここで $\{t/x\}$ は，x にそれぞれ t を代入する操作を意味する．

(4) の $!m$ は，非破壊的に読み出し可能なメモリとし

$$\text{換}\ \frac{\Gamma,\ A,\ B,\ \Delta\ \ \vdash}{\Gamma,\ B,\ A,\ \Delta\ \ \vdash} \qquad \otimes\ \frac{A,\ B,\ \Delta\ \ \vdash}{B \otimes A,\ \Delta\ \ \vdash}$$

$$\&\text{-}\!\multimap\ \frac{P_i\{t/x\},\ \Delta\ \ \vdash}{\mathsf{m}(a_i,t),\ \forall x(\mathsf{m}(a_1,x) \multimap P_1\ \&\ \cdots\ \&\ \mathsf{m}(a_n,x) \multimap P_n),\ \Delta\ \ \vdash}$$

$$\equiv\ \frac{Q\{\bar{t}/\bar{x}\},\ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \Gamma\ \ \vdash}{p(\bar{t}),\ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \Gamma\ \ \vdash} \qquad !\ \frac{\mathsf{m}(a,t),\ !\mathsf{m}(a,t),\ \Gamma\ \ \vdash}{!\mathsf{m}(a,t),\ \Gamma\ \ \vdash}$$

図 1 操作的意味論のための推論規則
Fig. 1 Rules for the operational semantics.

てはたらくプロセスを意味する．したがって，$!m \otimes P$ は，$!m$ に値を格納して，以後 P を実行するプロセスと考えられる．

以下ではプロセス式を $P, Q, P_1, P_2, \ldots$ で，メッセージ式を $m, m_1, m_2, \ldots$ のようにあらわす．

定義 2.3

p を n 引数の述語記号，$\bar{x}$ を変数の並び $x_1, \ldots, x_n$，Q をプロセス式とするとき，$!\forall\bar{x}(p(\bar{x}) \equiv Q)$ はプロセス定義式である．

この論理式によって $p(\bar{t})$ という形のプロセスが，$Q\{\bar{t}/\bar{x}\}$ に置き換えられる． 論理的には $P \equiv Q$ は，$(P \multimap Q)\&(Q \multimap P)$ の略記である．

2.2 操作的意味論

ここで用いる操作的意味論は [6] に基いている．すなわち LK 風のシークエント計算による推論規則を用いて計算を表現している．ここで用いるシークエント計算の規則は文献 [8] の LL のサブセットである．

Γ, Δ をそれぞれ有限個の論理式の並びとするとき，シークエントとは 以下のようなものである．

$\Gamma \vdash \Delta$

ここでは，右辺が空で左辺にのみ論理式があらわれるシークエント $\Gamma \vdash$ によって，現在計算が進んでいるプロセス等の集りを表現する．

2.2.1 推 論 規 則

操作的意味論を定める推論規則は，図 **1** の通りである．

ここで挙げた推論規則は，いずれも LL のいくつかの規則を組み合わせて行なった推論の形をしている．

換の規則と $\otimes$ の規則は，それぞれ LL における換の規則，$\otimes$ 左 の規則の右辺が空の場合に他ならない．$\&\text{-}\!\multimap$ と $\equiv$ の規則は， それぞれ 図 **2**, 図 **3** のような LL の推論をひとつの規則にまとめたものである． ! の規則は LL の ! 減と ! 左を続けて適用したものである．

これらの推論規則によって

$$\frac{\Gamma_2\ \vdash}{\Gamma_1\ \vdash}$$

と推論された場合，これは結論 Γ_1 から前提 Γ_2 に向かって計算が 1 ステップ進んだものと解釈する．

2.2.2 証明の構築としてのプロセスの実行

ここでは，プロセスの計算はひとつのシークエントの証明を構築する過程として表現される．すなわち $\Gamma \vdash$ のようなシークエントを考えたとき，Γ の各論理式は前節で説明したように実行中のプロセス，配送途中のメッセージ，又はプロセスの定義をあらわしている．これらの計算に必要なものがそれぞれ論理式で表現され，並べて $\Gamma \vdash$ として表されることによって，計算のある時点の状態を表現している． 計算とは，結論のシークエントが与えられその証明を前提に向かって構成してゆく過程として表現される．

定義 2.4 シークエント $\Gamma_0 \vdash$ の計算とは， $\Gamma_0 \vdash$ を先頭に持つシークエントの有限又は無限列

$$\begin{array}{c} \cdots \\ \hline \Gamma_i\ \ \vdash \\ \hline \cdots \\ \hline \Gamma_1\ \ \vdash \\ \hline \Gamma_0\ \ \vdash \end{array}$$

であって，各 $i (0 \leq i)$ について Γ_i は Γ_{i+1} を前提として持つ図 1 の推論のいずれかの結論である．

ここでは始式にたどりついた場合，有限ステップで終了した計算を意味する．また始式にたどりつくことなく適用可能な規則が無くなった場合には，計算がデッドロックしたものと考えられる．

例 1

以下のようなプロセスを考える．

$$!(P \equiv \forall x \forall y(\mathsf{m}(a_1, x) \multimap (\mathsf{m}(a_2, y) \multimap (\mathsf{m}(a_3, f(x, y)) \otimes P))))$$

$$!(Q \equiv \forall z(\mathsf{m}(a_{in}, z) \multimap (!\mathsf{m}(a_1, z) \otimes P)))$$

この二つのプロセス定義を Γ とあらわす．この定義のもとでプロセス Q が呼び出され， $\mathsf{m}(a_{in}, c)$ というメッセージが送られてきている様子は，周囲の環境

$$\begin{array}{c r}
P_n\{t/x\}\ ,\ \Delta\ \ \vdash \qquad\qquad \mathsf{m}(a_i,t)\ \vdash\ \mathsf{m}(a_i,t) & \multimap\text{左} \\ \hline
\mathsf{m}(a_i,t),\ \ \mathsf{m}(a_i,t) \multimap P_i\{t/x\},\ \ \ \Delta\ \vdash & \&\text{左} \\ \hline
\mathsf{m}(a_i,t),\ \ \ \mathsf{m}(a_1,t) \multimap P_1\{t/x\}\ \&\ \cdots \&\ \mathsf{m}(a_n,t) \multimap P_n\{t/x\},\ \ \ \Delta\ \vdash & \forall\text{左} \\ \hline
\mathsf{m}(a_i,t),\ \ \ \forall x(\mathsf{m}(a_1,x) \multimap P_1\ \&\ \cdots\ \&\ \mathsf{m}(a_n,x) \multimap P_n),\ \ \ \Delta\ \vdash &
\end{array}$$

図 2 &—o 規則に対応する LL の推論
Fig. 2 LL proof for &—o-rule.

$$\begin{array}{c r}
p(\bar{t}) \vdash p(\bar{t}) \qquad\qquad Q\{\bar{t}/\bar{x}\},\ \ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \ \Gamma\ \vdash & \multimap\text{左} \\ \hline
p(\bar{t}),\ \ p(\bar{t}) \multimap Q\{\bar{t}/\bar{x}\},\ \ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \ \Gamma\ \vdash & \&\text{左} \\ \hline
p(\bar{t}),\ \ p(\bar{t}) \equiv Q\{\bar{t}/\bar{x}\},\ \ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \ \Gamma\ \vdash & \forall\text{左} \\ \hline
p(\bar{t}),\ \ (\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \ \Gamma\ \vdash & !\text{左} \\ \hline
p(\bar{t}),\ \ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \ \Gamma\ \vdash & !\text{減} \\ \hline
p(\bar{t}),\ \ !(\forall\bar{x}(p(\bar{x}) \equiv Q)),\ \ \Gamma\ \vdash &
\end{array}$$

図 3 ≡ 規則に対応する LL の推論
Fig. 3 LL proof for ≡-rule.

をあらわす論理式の列を Δ_0 とあらわすと，以下のようになる．

$$Q, \mathsf{m}(a_{in}, c), \Gamma, \Delta_0 \vdash \qquad (1)$$

このような状態から開始された実行の様子は，先に挙げた推論規則により以下のように推移する．ここで推論規則は結論から前提に向かって適用されていることに注意されたい．

まず ≡ 規則により Q が定義によって展開される．(換の規則の適用は省略する．)

$$\begin{array}{r}
\mathsf{m}(a_{in}, c), \\
\forall z(\mathsf{m}(a_{in}, z) \multimap (!\mathsf{m}(a_1, z) \otimes P)), \\
\Gamma, \Delta_0 \vdash \qquad (2)
\end{array}$$

続いて $\mathsf{m}(a_{in}, c)$ が受信される．&—o の規則と ⊗ の規則より：

$$!\mathsf{m}(a_1, c), P, \Gamma, \Delta_0 \vdash \qquad (3)$$

≡ の規則により P が定義により展開される．

$$\begin{array}{r}
!\mathsf{m}(a_1, c), \forall x \forall y(\mathsf{m}(a_1, x) \multimap (\mathsf{m}(a_2, y) \multimap \\
(\mathsf{m}(a_3, f(x, y)) \otimes P))), \\
\Gamma, \Delta_0 \vdash \quad (4)
\end{array}$$

メモリ $!\mathsf{m}(a_1, c)$ から 値が読み込まれる動作が，! 規則と &—o 規則により：

$$\begin{array}{r}
!\mathsf{m}(a_1, c), \mathsf{m}(a_1, c), \\
\forall x \forall y(\mathsf{m}(a_1, x) \multimap (\mathsf{m}(a_2, y) \multimap \\
(\mathsf{m}(a_3, f(x, y)) \otimes P))), \\
\Gamma, \Delta_0 \vdash \qquad (5)
\end{array}$$

$$\begin{array}{r}
!\mathsf{m}(a_1, c), \\
(\mathsf{m}(a_2, y) \multimap (\mathsf{m}(a_3, f(c, y)) \otimes P))), \\
\Gamma, \Delta_0 \vdash \qquad (6)
\end{array}$$

のように進む．ここで環境 Δ_0 から $\mathsf{m}(a_2, b)$ が届けられるとする．外部環境でも計算が進んで Δ_0 は Δ_1 となる．

$$\begin{array}{r}
!\mathsf{m}(a_1, c), \\
(\mathsf{m}(a_2, y) \multimap (\mathsf{m}(a_c, f(c, y)) \otimes P))), \\
\mathsf{m}(a_2, b), \Gamma, \Delta_1 \vdash \qquad (7)
\end{array}$$

&—o 規則により $\mathsf{m}(a_2, b)$ が受信される．⊗ の規則より：

$$!\mathsf{m}(a_1, c), \mathsf{m}(a_3, f(c, b)), P, \Gamma, \Delta_1 \vdash \qquad (8)$$

この結果 $\mathsf{m}(a_3, f(c, b))$ が最初の出力として発信される．以下 (3) と同様な状態に戻って次の $\mathsf{m}(a_2, y)$ を待って同様の計算をくりかえす．

ここで，$\mathsf{m}(a_2, y)$ が一定時間ごとに届くのではなく，最初の $\mathsf{m}(a_2, y)$ が届くまでに十分に長い待ち時間がありかつ以下の $\mathsf{m}(a_2, y)$ は一度にまとめて届くような場合を想定する．このような場合 $\mathsf{m}(a_1, c)$ が届いてからの実行は，$\mathsf{m}(a_2, y)$ を待っている間処理が停止し，以下 x と y についての処理を逐次的にくりかえすことになる．

3. 並列実行の規則

前節で述べた操作的意味論を定める推論規則に加えて，本節ではさらにいくつかの推論規則を追加し，先の例のような逐次的に実行されるプロセスの並列化可能な部分を，並列に実行するための手法を示す．

定義 3.1

X をプロセスを表現する論理式を値としてとる変数とするとき，コンテクスト $E(X)$ は，以下のように定義される．

(1) 変数 X はコンテクストである．

$$\mathbf{UF}\ \frac{E(Q\{\bar{t}/\bar{x}\}),\ !\forall\bar{x}(p(\bar{x})\equiv Q),\ \Gamma\quad\vdash}{E(p(\bar{t})),\ !\forall\bar{x}(p(\bar{x})\equiv Q),\ \Gamma\quad\vdash}$$

$$\mathbf{FP}\ \frac{!\mathsf{m}(a_i,t),\ E(P_i\{t/x\}),\ \Gamma\quad\vdash}{!\mathsf{m}(a_i,t),\ \ E(\forall x(\mathsf{m}(a_1,x)\multimap P_1\ \&\cdots\&\ \mathsf{m}(a_i,x)\multimap P_i\ \&\cdots\&\ \mathsf{m}(a_n,x)\multimap P_n)),\ \Gamma\quad\vdash}$$

図 4 並列実行の規則
Fig. 4 Rules for Parallel Execution.

(2) l 個の変数 $X_1,X_2,\ldots,X_l$ を $\bar{X}$ とあらわすとき，$E(\bar{X}),E_1(\bar{X}),E_2(\bar{X}),\ldots,E_n(\bar{X})$ を $\bar{X}$ 以外の変数を含まないコンテクスト，$m_1,\ldots,m_n$ をメッセージとするとき:

(a) $E_1(\bar{X})\otimes E_2(\bar{X})$,

(b) $\forall\bar{x}(m_1\multimap E_1(\bar{X})\ \&$
$\cdots\&\ m_n\multimap E_n(\bar{X}))$

はコンテクストである．

新たに導入する推論規則は，図 **4** の通りである．ここで $E(X)$ はコンテクスト，P_i,Q,Γ 等は図 1 と同様である．

定理 3.1

UF 及び FP を用いて可能な推論は，LL の推論規則を用いて実現できる．

証明（概略）

UF について，任意のコンテクスト $E(X)$ について

$$E(p(\bar{t})),\ !\forall\bar{x}(p(\bar{x})\equiv Q)\vdash E(Q\{\bar{t}/\bar{x}\})$$

というシークエントが LL で証明可能であることが E の構造について帰納法を用いて示される．これによって，カットを用いて UF の前提から結論を導びくことができる．

また FP についても同様に，任意のコンテクスト $E(X)$ について

$$!\mathsf{m}(a_i,t),\ \ E(\forall x(\mathsf{m}(a_1,x)\multimap P_1\ \&\cdots\ \&\ \mathsf{m}(a_n,x)\multimap P_n))\ \vdash\ E(P_i\{t/x\})$$

というシークエントが LL で証明可能であることが，E の構造についての帰納法を用いて示される．ここからカットを用いて FP の前提から結論が導かれる．

この定理からわかるように，ここで新たに導入した二つの推論規則も，先に示した図 1 の推論規則と同様，LL での証明可能性の概念を広げるものではない．すなわち，見込み計算による並列化とは，線形論理の証明可能性の枠組みの中で記述できるものであるといえる．

以下では前節で定義した証明図の概念を，先の図 1 の推論規則に図 4 の規則を加えて再定義して用いる．

定義 3.2 以下で $E(X)$ を X を含むコンテクストとする．あるプロセス P について，$E(P)$ が出現する証明図で X の内部に推論が及ぶとは，証明図中に出現する P が，ある箇所で（換以外の）各推論規則の結論に出現し前提には出現しないことをいう．

直観的には，X の内部に推論が及ぶとは，プロセス P の実行が実際に開始されたことを意味する．

UF 又は FP の推論について，それがあるコンテクスト $E(X)$ の X の部分を書き換えるものであったとき，X にかかわる推論であるという．

定義 3.3 $E(P)$ をある証明図中のあるシークエントに出現する式とする．$E(P)$ の X にかかわる UF 又は FP の推論のある出現が消去できるとは，もとの証明図に対して，以下のような条件を満足する証明図が存在することをいう．

(1) X にかかわる UF 又は FP の推論のその出現を含まず，かつ

(2) 証明図から $E(P)$ 及びそれに推論規則を適用して得られた式以外の部分だけを取り出した場合，そこで適用されている推論は，もとの証明図の同じ部分について適用された推論がすべて同じ順序で適用された同じ式の列の並びとなっている．

以下 コンテクスト $E(X)$ を含む任意の証明図で，$E(X)$ の X にかかわる推論が UF であった場合と FP であった場合について，それぞれが消去できることを示す．

補題 3.2 コンテクスト $E(X)$ を含む任意の証明図で，X の内部に推論が及ばないならば，$E(X)$ の X にかかわる UF 及び FP の推論は消去することができる．

証明:

UF であった場合，与えられた証明図より X にかかわる UF の推論の前提より以降に出現するすべての X 部分について，UF によって $p(\bar{t})$ が $Q\{\bar{t}/\bar{x}\}$ に置き換えられている．また その $Q\{\bar{t}/\bar{x}\}$ は，X の内部に推論が及ばないことより，それ以降のすべてのシークエントにそのまま出現している筈である．このような $Q\{\bar{t}/\bar{x}\}$ をすべて $p(\bar{t})$ で置き換えることにより，X にかかわる UF の推論の前提と結論は同じシークエン

トとなる．この重複したシークエントの一方を消去することにより得られるシークエントの列は，やはり証明図となり，X にかかわる UF の推論の出現は含まれなくなる．またこの証明図は，$E(X)$ 以外の部分については置き換える前と全く変わりはないので，定義 3.3 の条件 2 も満足する．

FP の場合も同様である．

次に X の内部に推論が及ぶ場合について考える．

補題 3.3 $E(X)$ を X を含むコンテクストとする．$E(p(\bar{t}))$ を含む任意の証明図について，X の内部に推論が及ぶならば $E(X)$ の X にかかわる UF の推論は消去することができる．

証明:

E の構造についての帰納法を用いて証明する．

E が X のとき，プロセス定義の展開の推論の結論と前提は，ただちに $\equiv$ の結論と前提である．したがって UF を $\equiv$ とみなすことでただちに消去できる．

次に $E(X)$ が $E_1(X) \otimes E_2(X)$ であった場合を考える．推論の一般形は，$!\forall\bar{x}(p(\bar{x}) \equiv Q)$ を $P_{\equiv}$ と略記することにすると以下のような形をしている．

$$
\begin{array}{cr}
E'_1(Q'\{\bar{t}/\bar{x}\}),\ E'_2(Q''\{\bar{t}/\bar{x}\}),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline
E'_1(Q'\{\bar{t}/\bar{x}\}) \otimes E'_2(Q''\{\bar{t}/\bar{x}\}),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline\hline
E_1(Q\{\bar{t}/\bar{x}\}) \otimes E_2(Q\{\bar{t}/\bar{x}\}),\ P_{\equiv},\ \Gamma & \vdash \\
\hline
E_1(p(\bar{t})) \otimes E_2(p(\bar{t})),\ P_{\equiv},\ \Gamma & \vdash
\end{array}
$$

このような推論は X に代入された Q についての FP, UF の推論とそれ以外の推論に分けて，次のように変形することができる．

$$
\begin{array}{cr}
E'_1(Q'\{\bar{t}/\bar{x}\}),\ E'_2(Q''\{\bar{t}/\bar{x}\}),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline\hline
E'_1(Q\{\bar{t}/\bar{x}\}),\ E'_2(Q\{\bar{t}/\bar{x}\}),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline
E'_1(Q\{\bar{t}/\bar{x}\}) \otimes E'_2(Q\{\bar{t}/\bar{x}\}),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline
E'_1(p(\bar{t})) \otimes E'_2(p(\bar{t})),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline\hline
E_1(p(\bar{t})) \otimes E_2(p(\bar{t})),\ P_{\equiv},\ \Gamma & \vdash
\end{array}
$$

このような推論は，$E(X)$ 以外の部分についての推論に影響無くさらに以下のように変形することができる．

$$
\begin{array}{cr}
E'_1(Q'\{\bar{t}/\bar{x}\}),\ E'_2(Q''\{\bar{t}/\bar{x}\}),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline\hline
E_1(Q\{\bar{t}/\bar{x}\}),\ E_2(Q\{\bar{t}/\bar{x}\}),\ P_{\equiv}, \Gamma & \vdash \\
\hline
E_1(Q\{\bar{t}/\bar{x}\}),\ E_2(p(\bar{t})),\ P_{\equiv},\ \Gamma & \vdash \\
\hline
E_1(p(\bar{t})),\ E_2(p(\bar{t})),\ P_{\equiv},\ \Gamma & \vdash \\
\hline
E_1(p(\bar{t})) \otimes E_2(p(\bar{t})),\ P_{\equiv},\ \Gamma & \vdash
\end{array}
$$

これは $E_1(X), E_2(X)$ それぞれについて，X にかかわる UF を含む証明図となっている．帰納法の仮定を用いてこれらの X にかかわる UF は消去することができる．

$E(X)$ が

$$\forall\bar{y}(m_1 \multimap E_1(X) \& \cdots \& m_n \multimap E_n(X))$$

の形のとき，$n = 2$ の場合を示せばあとは同様な論法の繰り返しである．推論の一般形は以下のような形をしている．$\&$–$\multimap$ の規則で第一項が選ばれた場合を示しているが，第二項が選ばれた場合も同様である．

$$
\begin{array}{lr}
\quad E'_1(Q'\{\bar{t}/\bar{x}\})\{\bar{s}/\bar{y}\},\ P_{\equiv},\ \Gamma'' & \vdash \\
\hline
\forall y(m_1 \multimap E'_1(Q'\{\bar{t}/\bar{x}\})\ \& & \\
\quad m_2 \multimap E'_2(Q''\{\bar{t}/\bar{x}\})),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline\hline
\forall y(m_1 \multimap E_1(Q\{\bar{t}/\bar{x}\})\ \& & \\
\quad m_2 \multimap E_2(Q\{\bar{t}/\bar{x}\})),\ P_{\equiv},\ \Gamma & \vdash \\
\hline
\forall y(m_1 \multimap E_1(p(\bar{x}))\ \& & \\
\quad m_2 \multimap E_2(p(\bar{x}))),\ P_{\equiv},\ \Gamma & \vdash
\end{array}
$$

ここで $\otimes$ の場合と同様に，下から二段目の複数回の推論を Q の部分に対する UF, FP とそれ以外に分けて，以下のように変形する．これによって $E(X)$ 以外の部分についての推論の順序は，影響は受けない．

$$
\begin{array}{lr}
\quad E'_1(Q'\{\bar{t}/\bar{x}\})\{\bar{s}/\bar{y}\},\ P_{\equiv},\ \Gamma'' & \vdash \\
\hline\hline
\quad E_1(Q\{\bar{t}/\bar{x}\})\{\bar{s}/\bar{y}\},\ P_{\equiv},\ \Gamma'' & \vdash \\
\hline
\quad E_1(p(\bar{x}))\{\bar{s}/\bar{y}\},\ P_{\equiv},\ \Gamma'' & \vdash \\
\hline
\forall y(m_1 \multimap E_1(p(\bar{x}))\ \& & \\
\quad m_2 \multimap E_2(p(\bar{x}))),\ P_{\equiv},\ \Gamma' & \vdash \\
\hline\hline
\forall y(m_1 \multimap E_1(p(\bar{x}))\ \& & \\
\quad m_2 \multimap E_2(p(\bar{x}))),\ P_{\equiv},\ \Gamma & \vdash
\end{array}
$$

これより帰納法の仮定を用いて $E_1(p(\bar{t}))$ に対する $p(\bar{t})$ の UF は消去することができる．

以上により，UF の推論の場合は消去できることがわかった．

続いて X について FP の推論があった場合を考える．

補題 3.4 コンテクスト $E(X)$ について，$E(\forall x(m_1 \multimap Q_1\ \& \cdots \&\ m_n \multimap Q_n))$ を含む任意の証明図で，X の内部に推論が及ぶならば，$E(X)$ の X にかかわる FP の推論は消去することができる．

証明: E の構造についての帰納法を用いて証明する．

推論規則によれば 一般的には

$$E(\forall x(m_1 \multimap Q_1\ \& \cdots \&\ m_n \multimap Q_n))$$

という形の式がシークエントに登場するが，$n = 2$ の場合を示せば十分である．第一項 Q_1 が選ばれた場合を示しているが，他の場合も同様である．E が変数 X であった場合，推論の一般形は以下のようになる．

$$
\begin{array}{cr}
!m(\bar{t}), Q_1\{\bar{t}/\bar{x}\},\ \Gamma & \vdash \\
\hline
!m(\bar{t}), \forall x(m_1 \multimap Q\ \&\ m_2 \multimap Q_2),\ \Gamma & \vdash
\end{array}
$$

この推論は！と & の規則によって以下のように置き換えることができる．

$$
\begin{array}{l}
!m(\bar{t}),\ Q_1\{\bar{t}/\bar{x}\},\ \Gamma \quad \vdash \\
\hline
!m(\bar{t}),\ m(\bar{t}), \\
\quad \forall x(m_1 \multimap Q \mathbin{\&} m_2 \multimap Q_2),\ \Gamma \quad \vdash \\
\hline
!m(\bar{t}),\ \forall x(m_1 \multimap Q \mathbin{\&} m_2 \multimap Q_2),\ \Gamma \quad \vdash
\end{array}
$$

次に $E(X)$ が $E_1(X) \otimes E_2(X)$ であった場合．推論の一般形は以下のような形をしている．

$$
\begin{array}{l}
!m(\bar{t}),\ E'_1(Q'_1\{\bar{t}/\bar{x}\}),\ E'_2(Q''_1\{\bar{t}/\bar{x}\}),\ \Gamma' \vdash \\
\hline
!m(\bar{t}),\ E'_1(Q'_1\{\bar{t}/\bar{x}\}) \otimes E'_2(Q''_1\{\bar{t}/\bar{x}\}),\ \Gamma' \vdash \\
\hline\hline
!m(\bar{t}),\ E_1(Q_1\{\bar{t}/\bar{x}\}) \otimes E_2(Q_1\{\bar{t}/\bar{x}\}),\ \Gamma \vdash \\
\hline
!m(\bar{t}),\ E_1(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2)) \otimes \\
\quad E_2(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2)),\ \Gamma \quad \vdash
\end{array}
$$

$\otimes$ 以外の推論を Q_i についての UF, FP とそれ以外の推論に分けて，$E(X)$ 以外の部分に影響なく以下のように変形できる．

$$
\begin{array}{l}
!m(\bar{t}),\ E'_1(Q'_1\{\bar{t}/\bar{x}\}),\ E'_2(Q''_1\{\bar{t}/\bar{x}\}),\ \Gamma' \vdash \\
\hline\hline
!m(\bar{t}),\ E_1(Q_1\{\bar{t}/\bar{x}\}),\ E_2(Q_1\{\bar{t}/\bar{x}\}),\ \Gamma' \vdash \\
\hline
!m(\bar{t}),\ E_1(Q_1\{\bar{t}/\bar{x}\}), \\
\quad E_2(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2),\ \Gamma' \quad \vdash \\
\hline
!m(\bar{t}),\ E_1(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2), \\
\quad E_2(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2),\ \Gamma' \quad \vdash \\
\hline
!m(\bar{t}),\ E_1(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2) \otimes \\
\quad E_2(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2),\ \Gamma' \quad \vdash \\
\hline\hline
!m(\bar{t}),\ E_1(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2) \otimes \\
\quad E_2(\forall x(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2),\ \Gamma \quad \vdash
\end{array}
$$

ここで帰納法の仮定より，E_1, E_2 についての FP の規則は消去できる．

最後に $\forall\bar{y}(r_1 \multimap E_1(X) \mathbin{\&} \cdots \mathbin{\&} r_l \multimap E_l(X))$ の場合を考える．$l = 2$ の場合を示せば十分である．推論の一般形は以下のようになる． 第一項が選ばれた場合を示しているが， 第二項の場合も同様である．

$$
\begin{array}{l}
!m(\bar{t}),\ E'_1(Q'_1\{\bar{t}/\bar{x}\})\{\bar{s}/\bar{y}\},\ \Gamma' \quad \vdash \\
\hline
!m(\bar{t}),\ \forall\bar{y}(r_1(\bar{y}) \multimap E'_1(Q'_1\{\bar{t}/\bar{x}\}) \\
\quad r_2(\bar{y}) \multimap E'_2(Q''_1\{\bar{t}/\bar{x}\})),\ r_1(\bar{s}),\ \Gamma' \quad \vdash \\
\hline\hline
!m(\bar{t}),\ \forall\bar{y}(r_1(\bar{y}) \multimap E_1(Q_1\{\bar{t}/\bar{x}\}) \\
\quad r_2(\bar{y}) \multimap E_2(Q_1\{\bar{t}/\bar{x}\})),\ \Gamma \quad \vdash \\
\hline
!m(\bar{t}),\ \forall\bar{y}(r_1(\bar{y}) \multimap E_1(\forall\bar{x}(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2)) \\
\quad r_2(\bar{y}) \multimap E_2(\forall\bar{x}(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2))),\Gamma \vdash
\end{array}
$$

& 以外の推論を $\otimes$ の場合と同様に Q_i についての推論とそれ以外の推論に分けて，$E(X)$ 以外の部分に影響なく以下のように変形できる．

$$
\begin{array}{l}
!m(\bar{t}),\ E'_1(Q'_1\{\bar{t}/\bar{x}\})\{\bar{s}/\bar{y}\},\ \Gamma' \quad \vdash \\
\hline\hline
!m(\bar{t}),\ E_1(Q_1\{\bar{t}/\bar{x}\})\{\bar{s}/\bar{y}\},\ \Gamma' \quad \vdash \\
\hline
!m(\bar{t}),\ E_1(\forall\bar{x}(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2))\{\bar{s}/\bar{y}\},\ \Gamma' \ \vdash \\
\hline
!m(\bar{t}),\ \forall\bar{y}(r_1 \multimap E_1(\forall\bar{x}(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2)) \mathbin{\&} \\
\quad r_2 \multimap E_2(\forall\bar{x}(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2))), \\
\quad\quad r_1(\bar{s}),\ \Gamma' \vdash \\
\hline\hline
!m(\bar{t}),\ \forall\bar{y}(r_1 \multimap E_1(\forall\bar{x}(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2)) \mathbin{\&} \\
\quad r_2 \multimap E_2(\forall\bar{x}(m_1 \multimap Q_1 \mathbin{\&} m_2 \multimap Q_2))),\Gamma \vdash
\end{array}
$$

ここで帰納法の仮定より，E_1 についての FP の規則は消去できる．

以上により，FP の推論の場合は消去できることがわかった．

以上の補題によって以下の定理が示される．

定理 3.2 コンテクスト $E(X)$ の形の式を含む任意の証明図で， $E(X)$ の X にかかわる UF, FP の推論は消去することができる．

この定理は，図 4 の推論規則を導入した場合も，証明体系としての能力は増えることはなく，図 1 の推論規則からなる LL の部分体系と同等な能力であることを示している．さらに補題の証明からわかるように，$E(X)$ 以外の部分の推論に影響なく UF, FP 規則が消去できる．このことは，単に論理的に能力が同等であるだけでなく，プロセスとしての計算を $E(X)$ の外部から観測した際に，図 4 の推論を用いたあらゆる $E(X)$ 動作に対して，図 1 のみの推論による動作で同等なものが存在することを意味する．すなわち $E(X)$ の外部のプロセスは，$E(X)$ が UF, FP を使って動作していても，$E(X)$ は図 1 の規則だけで動作していると考えて実行を進めることができることを意味している．したがって，UF,FP の消去が計算としての意味を変えていないことがわかる．

例 2

先の例 1 で示したプロセスに対して，図 4 の推論規則を適用した場合を示す．

$$
\begin{array}{l}
!\mathsf{m}(a_1, c), \\
\forall x \forall y(\mathsf{m}(a_1, x) \multimap (\mathsf{m}(a_2, y) \multimap \\
(\mathsf{m}(a_3, f(x, y)) \otimes P))), \\
\Gamma,\ \Delta_0 \ \vdash
\end{array}
$$

までは 例 1 の (4) までと同様に実行が進む．ここで P の定義より UF の推論を用いて $\otimes$ の右側の P を定義に展開する．

$$!\mathsf{m}(a_1,c),\ \forall x\forall y(\mathsf{m}(a_1,x) \multimap (\mathsf{m}(a_2,y) \multimap (\mathsf{m}(a_3,f(x,y)) \otimes \forall x\forall y(\mathsf{m}(a_1,x) \multimap (\mathsf{m}(a_2,y) \multimap (\mathsf{m}(a_3,f(x,y)) \otimes P)))))),\ \Gamma,\ \Delta_0 \vdash \quad (9)$$

FP 規則を二箇所の $\mathsf{m}(a_1,x)$ に適用する．

$$!\mathsf{m}(a_1,c),\ \forall y(\mathsf{m}(a_2,y) \multimap (\mathsf{m}(a_3,f(c,y) \otimes \forall y(\mathsf{m}(a_2,y) \multimap (\mathsf{m}(a_3,f(c,y)) \otimes P))))),\ \Gamma,\ \Delta_0 \vdash \quad (10)$$

以上の推論は $\mathsf{m}(a_2,y)$ を待ち続ける間，可能な限り適用される．上記の時点で $\mathsf{m}(a_2,b_1),\mathsf{m}(a_2,b_2)$ という二つのメッセージが届いたとするとき，例 1 における逐次な実行と同様に受信される．

$$!\mathsf{m}(a_1,c),\ \forall y(\mathsf{m}(a_2,y) \multimap (\mathsf{m}(a_3,f(c,y) \otimes \forall y(\mathsf{m}(a_2,y) \multimap (\mathsf{m}(a_3,f(c,y)) \otimes P))))),\ \mathsf{m}(a_2,b_1),\ \mathsf{m}(a_2,b_2),\ \Gamma,\ \Delta_1 \vdash \quad (11)$$

まず $\mathsf{m}(a_2,b_1)$ が受信される．

$$!\mathsf{m}(a_1,c),\ \mathsf{m}(a_3,f(c,b_1) \otimes \forall y(\mathsf{m}(a_2,y) \multimap (\mathsf{m}(a_3,f(c,y)) \otimes P)))),\ \mathsf{m}(a_2,b_2),\ \Gamma,\ \Delta_1 \vdash \quad (12)$$

$\otimes$ の規則により $\mathsf{m}(a_3,f(c,b_1))$ が発信される．

$$\mathsf{m}(a_1,c),\ \forall y(\mathsf{m}(a_2,y) \multimap (\mathsf{m}(a_3,f(c,y)) \otimes P)))),\ \mathsf{m}(a_2,b_2),\ \Gamma,\ \Delta_2 \vdash \quad (13)$$

$\mathsf{m}(a_2,b_2)$ が受信されて処理される．

$$!\mathsf{m}(a_1,c),\ \mathsf{m}(a_3,f(c,b_2)) \otimes P,\ \Gamma,\ \Delta_2 \vdash \quad (14)$$

$\mathsf{m}(a_3,f(c,b_2))$ が発信される．

$$!\mathsf{m}(a_1,c),\ P,\ \Gamma,\ \Delta_3 \vdash \quad (15)$$

4. FP 規則の拡張

ここで示した見込み計算の手法は，$!\mathsf{m}(a,t)$ という形で値 t が記憶されているとき，この値を後で用いる部分をメッセージの到着待ちの時間を利用して先に計算しておくものであった．しかしながら，FP の規則は $!\mathsf{m}(a,t)$ がシークエント中に直接出現した際に初めて適用可能になるものであった．一方以下のような場合は，t の値が書き込まれてから $!\mathsf{m}(a,t)$ がシークエントに直接出現するまでに，メッセージ m の到着を待たねばならないが，t の値は m の到着以前に P の部分で活用することができる．

$(m \multimap (!\mathsf{m}(a,t) \otimes P))$

先に述べた FP の規則は，そのような見込み計算を記述することはできない．そこで FP の規則を以下のように強化した FP^+ という規則を考えることができる．

定義 4.1

変数 X,Y を含むコンテクスト $E(X,Y)$ で，X が Y に先行するとは，以下の (1) から (3) のいずれかの場合である．

(1) $E(X,Y)$ が $X \otimes F(X,Y)$ の形をしているか，

(2) $E_1(X,Y)$ で X が Y に先行していて，かつ $E(X,Y)$ が: $E_1(X,Y) \otimes E_2$ 又は

(3) $E_j(X,Y)$ で X が Y に先行していて，かつ $E(X,Y)$ が
$$\forall \bar{y}(m_1 \multimap E_1 \ \& \cdots \& \ m_j \multimap E_j(X,Y) \ \& \cdots \& \ m_l \multimap E_l)$$
の形をしている．

以上の定義をもとに新たに推論規則 FP^+ を図 **5** のように導入する．次の命題は，この規則を用いた推論についても先に示した *FP* と同様に LL の推論規則を用いて実現が可能であることを示している．

定理 4.1

FP^+ を用いて可能な推論は，LL の推論規則を用いて実現できる．

証明（概略）

任意のコンテクスト $E(X,Y)$ について X が Y に先行するなら，

$$E(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \ \& \cdots \ \& \ \mathsf{m}(a_n,x) \multimap P_n)) \vdash E(!\mathsf{m}(a_i,t), P_i\{t/x\})$$

というシークエントが LL で証明可能であることが，E の構造についての帰納法を用いて示される．ここからカットを用いて FP^+ の前提から結論が導かれる．

このような拡張した推論規則についても，前節で示した場合と同様に UF, FP とともにその適用を消去することができる． FP^+ を適用した $E(X,Y)$ の Y の推論が及ばない場合については，補題 3.2 の FP をそのまま FP^+ と読み換えることによって，FP^+ が消去できることが直ちに示される．

補題 4.1 X が Y に先行するコンテクスト $E(X,Y)$ を含む証明図で，Y の内部に推論が及ぶならば $E(X,Y)$ についての推論 FP^+ は消去することができる．

証明:

補題 3.3 の場合と同様に $n=2$ の場合で第一項が選択された場合を示せば十分である． E の構造についての帰納法を用いる．

$E(X,Y)$ が $X \otimes F(X,Y)$ の場合，推論の一般形は以下のような形をしている．

FP$^+$

$E(X,Y)$ で X が Y に先行するとき:

$$\begin{array}{c}
E(!\mathsf{m}(a_i,t), P_i\{t/x\}),\ \Gamma \quad \vdash \\ \hline
E(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \ \&\cdots\&\ \mathsf{m}(a_i,x) \multimap P_i \ \&\cdots\&\ \mathsf{m}(a_n,x) \multimap P_n)),\ \Gamma \quad \vdash
\end{array}$$

図 5 拡張された FP 規則
Fig. 5 Extended FP-rule.

$$\begin{array}{c}
!m(\bar{t}), F'(Q'_1\{\bar{t}/\bar{x}\}),\ \Gamma' \quad \vdash \\ \hline
!m(\bar{t}) \otimes F'(Q'_1\{\bar{t}/\bar{x}\}),\ \Gamma' \quad \vdash \\ \hline\hline
!m(\bar{t}) \otimes F(Q_1\{\bar{t}/\bar{x}\}),\ \Gamma \quad \vdash \\ \hline
!m(\bar{t}) \otimes F(\forall x(m_1 \multimap Q \ \&\ m_2 \multimap Q_2)),\ \Gamma \quad \vdash
\end{array}$$

この推論は以下のように変形できる.

$$\begin{array}{c}
!m(\bar{t}), F'(Q'_1\{\bar{t}/\bar{x}\}),\ \Gamma' \quad \vdash \\ \hline\hline
!m(\bar{t}) \otimes F(Q_1\{\bar{t}/\bar{x}\}),\ \Gamma \quad \vdash \\ \hline
!m(\bar{t}), F(\forall x(m_1 \multimap Q \ \&\ m_2 \multimap Q_2)),\ \Gamma \quad \vdash \\ \hline
!m(\bar{t}) \otimes F(\forall x(m_1 \multimap Q \ \&\ m_2 \multimap Q_2)),\ \Gamma \quad \vdash
\end{array}$$

すなわち $\otimes$ を先に適用することにより，FP$^+$ を FP の適用で代用することができる．このとき，$E(X,Y)$ 以外の部分の推論のは影響はない．さらに**補題 3.3** より代用した FP の推論は消去することができる． ゆえにこの場合は 図 1 以外の推論は消去することができる.

$E(X,Y)$ が $E_1(X,Y) \otimes F$ かつ E_1 で X が Y に先行している場合，推論の一般形は以下のような形をしている.

$$\begin{array}{l}
E'_1(!\mathsf{m}(a_i,t), P'_i\{t/x\}), F',\ \Gamma' \quad \vdash \\ \hline
E'_1(!\mathsf{m}(a_i,t), P'_i\{t/x\}) \otimes F',\ \Gamma' \quad \vdash \\ \hline\hline
E_1(!\mathsf{m}(a_i,t), P_i\{t/x\}) \otimes F,\ \Gamma \quad \vdash \\ \hline
E_1(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \ \& \\
\qquad\qquad \mathsf{m}(a_2,x) \multimap P_2)) \otimes F,\ \Gamma \quad \vdash
\end{array}$$

このような推論は以下のように $E(X,Y)$ 以外の部分に影響なく変形した後，帰納法の仮定より FP$^+$ の推論を消去することができる.

$$\begin{array}{l}
E'_1(!\mathsf{m}(a_i,t), P'_i\{t/x\}), F',\ \Gamma' \quad \vdash \\ \hline\hline
E_1(!\mathsf{m}(a_i,t), P_i\{t/x\}),\ F,\ \Gamma \quad \vdash \\ \hline
E_1(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \ \& \\
\qquad\qquad \mathsf{m}(a_2,x) \multimap P_2)),\ F,\ \Gamma \quad \vdash \\ \hline
E_1(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \ \& \\
\qquad\qquad \mathsf{m}(a_2,x) \multimap P_2)) \otimes F,\ \Gamma \quad \vdash
\end{array}$$

最後に $E_j(X,Y)$ で X が Y に先行していて，かつ $E(X,Y)$ が

$$\forall \bar{y}(m_1 \multimap E_1 \ \&\cdots\&\ m_j \multimap E_j(X,Y)\ \& \cdots \&\ m_l \multimap E_l)$$

の形をしている場合を考える．ここで $l=1$ の場合を示せば十分である．推論の一般形は以下のような形になる.

$$\begin{array}{l}
E''_1(!\mathsf{m}(a_i,t), P'_1\{t/x\})),\ \Gamma' \quad \vdash \\ \hline
\forall \bar{y}(m_1 \multimap E'_1(!\mathsf{m}(a_i,t), P'_1\{t/x\})),\ m_1,\ \Gamma' \quad \vdash \\ \hline\hline
\forall \bar{y}(m_1 \multimap E_1(!\mathsf{m}(a_i,t), P_1\{t/x\})), \Gamma \quad \vdash \\ \hline
\forall \bar{y}(m_1 \multimap E_1(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \\
\qquad\qquad \mathsf{m}(a_2,x) \multimap P_2))),\ \Gamma \quad \vdash
\end{array}$$

このような場合は，$E(X,Y)$ 以外の部分に影響することなく以下のように推論を変形することができる.

$$\begin{array}{l}
E'_1(!\mathsf{m}(a_i,t), P'_1\{t/x\}),\ \Gamma' \quad \vdash \\ \hline\hline
E_1(!\mathsf{m}(a_i,t), P_1\{t/x\}),\ \Gamma' \quad \vdash \\ \hline
E_1(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \& \\
\qquad\qquad \mathsf{m}(a_2,x) \multimap P_2)))\ \Gamma' \quad \vdash \\ \hline
\forall \bar{y}(m_1 \multimap E_1(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \\
\qquad\qquad \mathsf{m}(a_2,x) \multimap P_2))), \\
\qquad\qquad\qquad m_1,\ \Gamma' \quad \vdash \\ \hline\hline
\forall \bar{y}(m_1 \multimap E_1(!\mathsf{m}(a_i,t), \forall x(\mathsf{m}(a_1,x) \multimap P_1 \\
\qquad\qquad \mathsf{m}(a_2,x) \multimap P_2))), \\
\qquad\qquad\qquad \Gamma \quad \vdash
\end{array}$$

ここで帰納法の仮定より， $E_1(!\mathsf{m}(a_i,t), P_1\{t/x\})$ を得た FP$^+$ の推論は，消去することができる.

以上で，FP$^+$ に拡張した体系でも 図 4, 図 5 の推論を含まないで同等な計算を表現する推論が存在することが示された．すなわち以下の定理が成立する.

定理 4.2 図 1, 図 4, 図 5 の推論規則を用いて作られた任意の証明図について，UF, FP, FP$^+$ を消去した証明図が存在する.

5. まとめ

本稿では，線形論理で表現された並行プロセスの操作的意味論について述べた．特に逐次的に複数のメッセージの到着を待って実行されるプロセスについて,

メッセージの到着待ちの間に見込みで計算を行なう操作的意味論を導入した．そのもとでの任意の計算について，外部から観測した際本来の操作的意味論で同等な計算が存在することを示した．このことによって，線形論理で記述された逐次的なプロセスの並列な実行が，LL の部分体系での推論によって表現できることが示された．

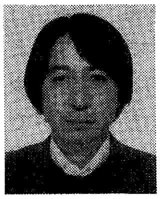
本稿で対象とした線形論理を用いて表現された並行プロセスは，名前の束縛を含まない非同期 π 計算の部分集合に対応する．非同期 π 計算等における名前制限の機能は，非同期のメッセージ通信を基本とした並行計算の記述能力の上で本質的かつ重要なものであることが知られている．したがって，ここで並列実行の対象としたプロセスを名前制限を含むものに拡張することは興味深くかつ重要な課題といえる．線形論理を用いて π 計算の名前制限に相当する機能をもつ並行プロセスを記述する手法は，存在限定記号を使って可能である[5]．今後はこのような記述を含む体系に，本稿での手法を拡張することが考えられる．

謝辞 岡山大学知能情報処理工学研究室の皆様には，有益な議論と研究上の御支援をいただきました．記して感謝します．また本研究は文部省科研費特定領域研究 (A) 10139229 の補助により実施された．

参 考 文 献

1) Girard J-Y., Linear Logic, *Theoretical Computer Science*, 50, 1-102 (1987).
2) Boudol G., Asynchrony and the π-calculus, Thechnical Report 1702, INRIA, Sophia-Antipolis, (1992).
3) Kobayashi N., A. Yonezawa, Logical, Testing and Observation Equivalence for Processes in a Linear Logic Programming, Technical Report, University of Tokyo
4) Kobayashi N., Concurrent Linear Logic Programming, PhD Thesis, Department of Information Science, University of Tokyo. (1996).
5) 村上 昌己: 線形論理を用いた並行プロセスの動的改造手法, 日本ソフトウェア科学会 第 15 回大会, D6-1 (1998).
6) 岡田 光弘: 線形論理に基づく並行計算モデル, − 並行計算の論理的理解の試み − , 情報処理, Vol. 37, No. 4, pp. 327-332 (1996).
7) 清水 智弘, 小林 直樹: 分散線型論理プログラミング, 日本ソフトウェア科学会 第 14 回大会論文集, E5-3, pp305-308 (1997).
8) 竹内 外史: 線型論理入門, 日本評論社 (1995).
9) 田村 直之, 池田 雄一, 線形論理型言語のコンパイラ処理系でのリソース管理方式について, 情処研報, 96-PRO-7-5 (1996).

(平成 10 年 10 月 16 日受付)
(平成 10 年 12 月 16 日採録)



村上　昌己（正会員）

1957 年生．1980 年名古屋工業大学 情報工学科卒業．1982 年名古屋大学大学院 修士課程情報工学 修了．1985 年同 博士課程満了．同年 9 月工学博士．同年 4 月富士通 (株) 入社．同年 6 月より 1989 年 7 月まで新世代コンピュータ技術開発機構に出向．帰社後富士通国際情報社会科学研究所に勤務．1991 年 4 月より岡山大学工学部助教授．並行性の理論，プログラムの意味論，プログラム変換／検証等の研究に従事．高橋奨励賞受賞．