プログラミング—言語・基礎・実践— 5 −11
(1991. 11. 22)

通信プロセスにおける値による条件分岐機構について

結縁 祥治　坂部 俊樹　稲垣 康善

名古屋大学 工学部

〒464–01 名古屋市 千種区 不老町

あらまし　本稿では，同期通信を基本的な計算機構とする並行計算のモデルである通信プロセスにおいて，値の受渡しに関して拡張した体系を提案する．通信プロセスを並行プログラムのモデルとしてとらえたとき，受け渡された値はプロセスの振舞いに影響を与えるのが自然である．このため，値を表現する体系と値によって振舞いを切替える条件分岐の機構を導入し，値の計算も含めた通信プロセスの形式的な意味論について議論する．値の計算を明示的に通信プロセスの振舞いに導入することによって，無限リストなどの無限の構造を持った値の通信について振舞いを定義することができる．このような通信プロセスの動作的な意味論をテスト等価性に基づいて定義し，止まらない値の計算もプロセスの振舞いとして議論する．

A Mechanism for Branching by Values in Communicating Processes

Shoji Yuen　Toshiki Sakabe　Yasuyoshi Inagaki

Faculty of Engineering,
Nagoya University

Furo-cho, Chikusa-ku, Nagoya
Japan, 464–01

Abstract: In this paper, we propose a new framework of communicating processes which is extended in passing values. The mechanism for passing values in communications is the useful and natural requirement in order to model a concurrent program by communicating processes. We introduce the rewriting mechanism to compute values as a behaviour of processes. For this extended framework, we establish a formal semantics by "Testing". It is shown that infinite data objects can be introduced as values, and that our rewriting mechanism reflects the "true non-termination" in value computation as divergence in process behaviour.

1 はじめに

通信プロセスは，複数のプロセスが互いに通信を行ないながら，計算が進行する並行計算のモデルである．通信プロセスの基本的な計算機構は同期通信である．各プロセスは，通信チャネルを通して，他のプロセスとハンドシェークの通信を行なって計算を進める．このモデルはCCS[1]やCSP[3]のような体系として定式化され，形式的な意味論が提案されている．このような通信プロセスの特徴は，並行性に基づく非決定性と停止しない計算である．このため，通信プロセスの意味論にはオートマトンのような逐次的な計算モデルのそれとは異なった性質が要求される．

並行プログラムの実行は通信プロセスの振舞いでモデル化することができる．このことによって，き並行プログラムの意味記述は通信プロセスの体系への変換によって与えることができ，並行プログラムの意味を明確に定義することができるようになる[1][2][8][9][14]．通信プロセスに基づいた並行プログラムの動作は，通信チャネルから値を入力して，通信チャネルに値を出力するととらえるのが自然であるため，並行プログラムを通信プロセスとして定式化するときには，値に関する拡張が必要になる．Full–CCS[1]や文献[2]の体系では値が導入されているが，いずれも値は暗黙に与えられており，通信プロセスとしての振舞いと値の計算とは分離して扱われている．しかし，このために値の計算機構に要求される停止性[5]は，並行プログラムを動作させて振舞いを観測するという観点からすると非常に厳しい条件であり，動作的な意味定義としては不十分であると考えられる．

このような観点に立って，本稿では，並行プログラムのモデルとしての通信プロセスの枠組みの拡張を提案する．この枠組みでは，通信プロセスの枠組みに同期通信の際の値の受渡しとプロセス内での値に対する条件分岐のための機構を導入する．通信の際に受け渡される値は項で表し，項の表す値は，構成子を持つ項書き換え系で計算される代数によって定義する．項書換え系で実際に値を計算することにより，値に関して止まらない計算も通信プロセスの振舞いとして扱うことができるようになる．さらに，無限リストのように，止まらない計算によって表される無限の値に対して，遅延評価を行なって，振舞いを切り替えることができるような条件分岐の機構を導入する．このことによって，無限の値に対しての振舞いが定義できる．

2節で値の計算の枠組みをもつ通信プロセスの記述体系を定義し，3節で値を表す項の意味を定める．4節では値の通信プロセスをテストにもとづいて観測し，動作的意味を定める．5節ではテスト等価性に基づく動作的意味付けの性質について議論し，6節でまとめと今後の課題を述べる．

2 通信プロセス記述言語 PL_v^0

本節では，通信プロセスを記述する言語 PL_v^0 を定める．PL_v^0 は，値を表す項とその書き換え規則の集合，および，プロセスの振舞いを表す動作式から成る．

2.1 値項と値書き換え系

PL_v^0 において値を表す項を値項と呼び，シグニチャ Σ_v と変数の集合 X_v から構成されるとする．ここで，Σ_v は互いに素な二つの部分 Σ_{vc} と Σ_{vf} とに分割されているとする．また，特別な定数記号 ω が存在するものとし，ω は Σ_{vc} にも Σ_{vf} にも含まれないとする．

変数 $X (X \subseteq X_v)$ を含む整合のとれた値項の集合は $T_{\Sigma_v}(X)$ で表す．ここで整合がとれているとは，引数の数とソート[1]に矛盾がないことを示す．以下の議論では，値項はすべて整合がとれているとする．また，$X = \emptyset$ の場合は単に T_{Σ_v} で表し，**値基礎項**と呼ぶ．また，Σ_{vc} だけから構成される値項は $T_{\Sigma_{vc}}(X)$ で表し，$T_{\Sigma_{vc}}(X)$ を**値構成子項**と呼ぶ．

[1]以後，ソートに関しては議論の本質ではないため陽には扱わない．ソートに関する形式的な議論は文献[10]などを参照して頂きたい．

次の形をしている式を値書き換え規則と呼ぶ．

$$t_l \rightarrow t_r$$

ここで，$t_l, t_r \in T_{\Sigma_v}(X)$ であり，t_r に現れる変数は t_l に現れているものとする．値書き換え規則の集合を値書き換え系とよぶ．

値書き換え系 R による書き換え関係を通常の項書き換え系 (TRS) と同様に次のように定める．

値書き換え規則 $r \equiv t_l \rightarrow t_r$ と値項 t_1, t_2 に対して，

$$t_1 = \mathcal{C}[t_l\theta], t_2 = \mathcal{C}[t_r\theta]$$

であるとき，t_1 は r によって t_2 に書き換えられるといい，$t_1 \Rightarrow_r t_2$ と書く．ここで C[] は文脈であり，θ は代入である．また，値書き換え系 R に対して $r \in R$ が存在して，$t_1 \Rightarrow_r t_2$ であるとき，$t_1 \Rightarrow_R t_2$ と書く．

PL_v^0 で扱う値書き換え系は Σ_{vc} と Σ_{vf} に対して，次の3つの条件を満たすものとする．

(i) 無曖昧性:
$t_{li} \rightarrow t_{ri}, t_{lj} \rightarrow t_{rj} \in R$, $i \neq j$ のとき，t_{li} の変数以外の全ての部分項 u_{li} に対し，$u_{li}\theta = t_{lj}$ となる θ が存在しない．

(ii) 左線形性:
$t_l \rightarrow t_r \in R$ において，t_l に現れる変数の出現回数は高々一回である．

(iii) 完全性:
以下の a. と b. を満たす．

a. $t_l \rightarrow t_r \in R$ において，t_l は $f(t_1, \ldots, t_n)$ の形をしていて，以下の条件を満たす．

- $f \in \Sigma_{vf}$
- $t_i \in T_{\Sigma_{vc}}(X)$ for $1 \leq i \leq n$

b. $f \in \Sigma_{vf}$, $t_i \in T_{\Sigma_{vc}}$ $(1 \leq i \leq n)$ のとき，$f(t_1, \ldots, t_n) \Rightarrow_R t$ となる t が必ず存在する．

条件 (1) および (2) により，R は合流性を持ち，値項の意味を書き換えによって定めることができる．条件 (3) により，構成子項でない基礎値項は書き換えられることが保証される．

2.2 動作式

通信チャネルの集合 C が与えられているとき，PL_v^0 の動作式 E は以下のように BNF 風に定義される式である．

$$\begin{aligned} E \ ::= \ & \texttt{NIL} \ [] \ \Omega \\ & [] \ c?x.E \ [] \ c!t.E \\ & [] \ E + E \ [] \ E \oplus E \\ & [] \ [t == u].E \end{aligned}$$

ここで，$c \in C, x \in X_v, t \in T_{\Sigma_v}(X_v), u \in T_{\Sigma_{vc}}(X_v)$ とする．

それぞれの形の動作式は直観的に次のような意味を持つ．

- NIL: 停止したプロセス
- Ω: 発散しているプロセス
- $c?x.E$: チャネル c から値を変数 x に受け取るプロセス
- $c!t.E$: チャネル c から値 t を出力するプロセス
- $E_1 + E_2$: 外部からの制御によって E_1 あるいは E_2 のいずれかが選択されるプロセス
- $E_1 \oplus E_2$: E_1 あるいは E_2 のいずれかが非決定的に選択されるプロセス
- $[t == u].E$: 等式 $t = u$ が成立したら E を実行するプロセス

動作式の変数の出現は以下の2つの規則によって束縛される．

(1) $c?x.E$ の形の部分式において $c?x$ および E に現れる x は束縛されている．

(2) $[t == u].E$ の形の部分式において u に現れる変数 $var(u)$ および E に現れる $var(u)$ は束縛されている．

束縛されない変数の出現は自由である．動作式に対する代入の適用は，自由な変数に対してのみ行なわれる．自由な変数の出現のない動作式をプロセス式と呼ぶことにする．

2.3 PL_v^0 のプログラム

PL_v^0 のプログラムは，値書き換え系 R とプロセス式 P との対 (R,P) である．これを単に P_R と書く．

3 値書き換え系による値項の代数的意味論

本節では，値書き換え系によって与えられる値項の意味論を文献[11][12]に沿って定義する．詳細は文献[11]を参照して頂きたい．

3.1 無限の値を表すための代数構造

$T_{\Sigma_{vc}\cup\{\omega\}}$ 上の関係 $\preceq$ を次のように定める．

(i) 任意の $t \in T_{\Sigma_{vc}\cup\{\omega\}}$ に対して $\omega \preceq t, t \preceq t$

(ii) $t \preceq t'$ かつ $t' \preceq t''$ ならば $t \preceq t''$

(iii) $t_i \preceq t'_i \ (1 \leq i \leq n)$ ならば，$f(t_1,\ldots,t_n) \preceq f(t'_1,\ldots,t'_n) \ (f \in T_{\Sigma_{vc}})$

この関係は $T_{\Sigma_{vc}\cup\{\omega\}}$ 上の順序関係である．

領域として $T_{\Sigma_{vc}\cup\{\omega\}}$，$\Sigma_{vc}$ に属する関数記号および ω は引数とそれ自身から構成される項への写像に割り当てると解釈すると，代数構造 $< T_{\Sigma_{vc}\cup\{\omega\}}, \preceq >$ は Σ 擬領域[4]をなす．この Σ 擬領域はイデアル完備化により一意的に Σ 領域に拡張できる[4]．この Σ 領域を $< T^{\infty}_{\Sigma_{vc}\cup\{\omega\}}, \preceq_\infty >$ で表す．ここで，$\preceq_\infty$ は完備化半順序であり，$< T^{\infty}_{\Sigma_{vc}\cup\{\omega\}}, \preceq_\infty >$ は $< T_{\Sigma_{vc}\cup\{\omega\}}, \preceq >$ を基底とする代数的完備化半順序構造となる．以下では単に $T^{\infty}_{\Sigma_{vc}}$ と書くことにする．

3.2 値項に対する意味の割り当て

T_{Σ_v} から $T_{\Sigma_{vc}\cup\{\omega\}}$ への関数 $\omega_{\Sigma_{vc}}$ を次のように定める．

$t \in T_{\Sigma_v}$ に対して，

$$\omega_{\Sigma_{vc}}(t) = max\{u; u \preceq t, u \in T_{\Sigma_{vc}\cup\{\omega\}}\}$$

構成子 Σ_{vc} を持つ値書き換え系 R において，$t \Rightarrow_R u$ であるとき，R の完全性から，$\omega_{\Sigma_{vc}}(u)$ は t の近似正規形[11]であり，これに基づいて ω を含む値基礎項 $T_{\Sigma_v\cup\{\omega\}}$ に対する割り当て Val_R を次のように定義する．

$$Val_R(t) = \sqcup\{\omega_{\Sigma_{vc}}(u); t \Rightarrow^*_R u\}$$

Val_R は $T_{\Sigma_v\cup\{\omega\}}$ から $T^{\infty}_{\Sigma_{vc}}$ への単調な全域関数である．この割り当ては値書き換え規則 $t_l \rightarrow t_r$ を等式 $t_l = t_r$ と見なした時に生成される自由連続代数のクラスにおける始代数の表示を与えることが知られている[11]．

4 PL_v^0 の動作的意味論

4.1 動作式の導出意味論

イベントによって，通信プロセスが外部と行なう通信を定義する．イベントは入出力イベントと空イベントに分けられる．入出力イベントはチャネルと入出力フラグ(?,!)と $T_{\Sigma_v\cup\{\omega\}}$ の要素との三項組で表す．入力イベントを $< c?t >$，出力イベントを $< c!t >$ のように書く．空イベントは0で表すことにする．

動作式間の関係にはイベント e によってラベル付けされた関係 $\xrightarrow{e}$ を導出関係と呼び，次のような公理と推論規則によって定める．

$$(\Omega) \quad \frac{}{\Omega \xrightarrow{0} \Omega}$$

$$(I) \quad \frac{}{c?x.E \xrightarrow{< c?t >} E\{t/x\}}$$

$$(O) \quad \frac{}{c!t.E \xrightarrow{< c!t >} E}$$

$$(S0_1) \quad \frac{E_1 \xrightarrow{0} E'_1}{E_1 + E_2 \xrightarrow{0} E'_1 + E_2}$$

$$(S0_2) \quad \frac{E_2 \xrightarrow{0} E'_2}{E_1 + E_2 \xrightarrow{0} E_1 + E'_2}$$

$$(S_1)\quad \frac{E_1 \xrightarrow{e} E_1'}{E_1 + E_2 \xrightarrow{e} E_1'},\ (e \neq 0)$$

$$(S_2)\quad \frac{E_2 \xrightarrow{e} E_2'}{E_1 + E_2 \xrightarrow{e} E_2'},\ (e \neq 0)$$

$$(N_1)\quad \frac{}{E_1 \oplus E_2 \xrightarrow{0} E_1}$$

$$(N_2)\quad \frac{}{E_1 \oplus E_2 \xrightarrow{0} E_2}$$

4.2 値テストと値テストシステム

次に通信プロセスに対するテスト [4] [6] の概念に値を導入して，通信プロセスの振舞いに対する観測を定義する．

テスト項 T は次のように BNF 風に定められる項である．

$$\begin{array}{rcl} T & ::= & SUCC \\ & & \mathbin{[\!]}\ c?t.T \mathbin{[\!]} c!t.T \mathbin{[\!]} 1.T \\ & & \mathbin{[\!]}\ T + T \end{array}$$

ここで，$t \in T_{\Sigma_{vc}\cup\{\omega\}}$，$c \in C$ とし，1 は新たな記号であるとする．$SUCC$ はテストが成功したことを表す定数である．また，テスト項全体の集合を $\mathcal{T}$ と書く．

テスト項に対して導出関係を次のように定める．

$$(I_T)\quad \frac{}{c?t.T \xrightarrow{<c?t>}_T T}$$

$$(O_T)\quad \frac{}{c!t.T \xrightarrow{<c!t>}_T T}$$

$$(1_T)\quad \frac{}{1.T \xrightarrow{1}_T T}$$

$$(S_{T1})\quad \frac{T_1 \xrightarrow{e'}_T T_1'}{T_1 + T_2 \xrightarrow{e'}_T T_1'}$$

$$(S_{T2})\quad \frac{T_2 \xrightarrow{e'}_T T_2'}{T_1 + T_2 \xrightarrow{e'}_T T_2'}$$

ここで，e' は入出力イベントである．

プロセス式は，値テストによってテストされる．このためのシステムを値テストシステムと呼ぶ．本稿の枠組みでは値に対して無限の構造を許すが，テスト項 T では有限の構造しか許さない．このため，テストが値に対してどの程度正確であるかという概念を導入する．

[**定義 1**] (値正確度，テスト正確度)
値正確度は $T_{\Sigma_{vc}}$ の対であり，$(t - t')$ と表す．値正確度の 0 個以上の系列をテスト正確度と呼び，Pr で表す． □

値正確度に対して順序 $\preceq_p$ を定義する．

$$(s - s') \preceq_p (t - t') \text{ iff } s \preceq t \text{ かつ } s' \preceq t'$$

直観的には $\preceq_p$ はテストにおける値の近似度を表現している．$\preceq_p$ はテスト正確度に対して次のように拡張される．

(i) $\epsilon \preceq_p Pr$

(ii) 値正確度 p, p' とテスト正確度 Pr, Pr' に対して，$p \preceq_p p'$ かつ $Pr \preceq_p Pr'$ のとき $p \cdot Pr \preceq_p p' \cdot Pr'$

[**定義 2**] (値テストシステム)
PL_v^0 のプログラム P_R とテスト項 T に対する値テストシステムは，遷移システム

$$<(\mathcal{E}\|\mathcal{T}, \mathcal{PR}), (P_R\|T, \epsilon), \rightarrowtail_R>$$

である．ここで，$(\mathcal{E}\|\mathcal{T}, \mathcal{PR})$ は状態集合であり，$\mathcal{E}$,$\mathcal{T}$,$\mathcal{PR}$ は動作式，テスト項，テスト正確度の全体集合である．$(P\|T, \epsilon)$ は初期状態である．遷移関係 $\rightarrowtail_R$ は次の公理と推論規則を満たす最小の関係である．

$$(\mathbf{N_E})\quad \frac{E \xrightarrow{0} E'}{(E\|T, Pr) \rightarrowtail_R (E'\|T, Pr)}$$

$(\mathbf{I_E})$

$$\frac{E \xrightarrow{<c?s>} E',\ T \xrightarrow{<c?t>}_T T'}{(E\|T, Pr) \rightarrowtail_R (E'\|T', Pr \cdot (t - Val_R(s)))}$$

ここで，$Val_R(s) = t$

$(\mathbf{O_E})$

$$\frac{E \xrightarrow{<c!s>} E',\ T \xrightarrow{<c?t>}_T T'}{(E\|T, Pr) \rightarrowtail_R (E'\|T', Pr \cdot (Val_R(s) - t))}$$

ここで，$t \preceq Val_R(s)$

$$(\mathbf{1_E})\quad \frac{T \xrightarrow{1}_T T'}{(E\|T, Pr) \rightarrowtail_R (E\|T', Pr)}$$

$(\mathbf{G_{E1}})$

$$\frac{t \Rightarrow_R t', t \neq u\theta \text{ for all } \theta}{([t == u].E\|T, Pr) \rightarrowtail_R ([t' == u].E\|T, Pr)}$$

$(\mathbf{G_{E2}})$

$$\frac{t \neq \omega, t = u\theta \text{ for some } \theta}{([t == u].E\|T, Pr) \rightarrowtail_R (E\|T, Pr)}$$

$(\mathbf{G_{E3}})$

$$\frac{}{([\omega == u].E\|T, Pred) \rightarrowtail_R ([\omega == u].E\|T, Pr)}$$

□

4.3 値テストに対する観測

ここでは，無限の構造を持つデータに対する値テストについて議論する．このため，値書き換え系は固定して考える．

プログラム P_R を値テストシステムに適用して得られる結果を $Res(P_R, T)$ として次のように定義する．ここで，$\top$ と $\bot$ はそれぞれテストの成功と失敗を表す．

[定義 3] (値テスト結果)
$Res(P_R, T) \subseteq \{(\top, Pr), (\bot, Pr); Pr \in \mathcal{PR}\}$ であり，$Res(P_R, T)$ は次の条件を満たす最小の集合とする．

(i) $(P\|T, \epsilon) \rightarrowtail_R^* (P'\|SUCC, Pr)$ のとき，$(\top, Pr) \in Res(P_R, T)$

(ii) 以下の (a),(b) いずれかが成り立つとき，$(\bot, Pr) \in Res(P_R, T)$

(a) $(P\|T, \epsilon) \rightarrowtail_R^* (P'\|T', Pr)$, $(P'\|T', Pr) \not\rightarrowtail_R$ かつ $T' \neq SUCC$

(b) $(P\|T, \epsilon) \rightarrowtail_R (P'\|T', Pr)$ かつ $(P'\|T', Pr) \rightarrowtail_R^\infty$

ここで，$S \not\rightarrowtail_R$ は $S \rightarrowtail_R S'$ となる S' が存在しないことを示し，$S \rightarrowtail_R^\infty$ は，無限の系列 $S_0, S_1, S_2, \ldots$ が存在して，

$$S = S_0 \rightarrowtail_R S_1 \rightarrowtail_R S_2 \rightarrowtail_R \cdots$$

となることを示す．□

通信プロセスの振舞いは非決定的であるので，一般に $Res(P_R, T)$ の結果は一つに定まらない．また，値に関してより正確なテストをした結果は，他の結果に優先する．結果の正確性を表す関係 $\preceq_p$ を次のように定める．

$$Res(P_R, t) \preceq_p Res(P'_R, t')$$
$$\text{iff}$$
$$\forall (r, Pr) \in Res(P_R, t) \exists (r', Pr') \in Res(P'_R, t')\ \ Pr \preceq_p Pr'$$

次に，無限のデータ構造に対してテスト項の値部分を拡張する．このために，値テストに関して次の順序を導入する．

[定義 4] (テスト順序)
テスト項に対するテスト順序 $\triangleleft$ は以下を満たす最小の関係であるとする．

(i) $SUCC \triangleleft SUCC$

(ii) $t_1 \preceq t_2$ かつ $T_1 \triangleleft T_2$ ならば
$c?t_1.T_1 \triangleleft c?t_2.T_2$,
$c!t_1.T_1 \triangleleft c!t_2.T_2$

(iii) $T_1 \triangleleft T_2$ ならば $1.T_1 \triangleleft 1.T_2$

(iv) $T_1 \triangleleft T'_1$ かつ $T_2 \triangleleft T'_2$ ならば
$T_1 + T_2 \triangleleft T'_1 + T'_2$

□

テスト順序に対してテスト結果は値に関して精密化される．

[命題 5] $T_1 \triangleleft T_2$ ならば，任意の P_R に対して $Res(P_R, T_1) \preceq_p Res(P_R, T_2)$ □

テスト項の集合 $\mathcal{T}$ は $\triangleleft$ の関係で分割され，それぞれの分割を $[T]$ であらわす．すなわち，

$$[T] = \{T'; T' \triangleleft T \text{ or } T \triangleleft T'\}$$

$[T]$ は値に関しては異なるが，通信に関しては同じ形のテストを行なうテスト項の集合である．

値項の出現が全て ω であるようなテスト項の集合を $\mathcal{T}_\omega$ とすると，任意の $T_\omega \in \mathcal{T}_\omega$ に対して，$< [T_\omega], \triangleleft >$ は Σ 擬順序構造をなす．値項に対して行なった場合と同様に，これを Σ 順序構造に一意的に拡張することができる．全ての $[T_\omega]$ に対して，このような拡張を行なった結果得られるテスト項の集合を $\mathcal{T}^\infty$ で表す．

テスト結果から，値に関して最も詳しい結果を観測として取り出す操作を次のように定義する．

[**定義 6**] (値テスト結果の観測 O)
$S \subseteq \{(\top, Pr), (\bot, Pr); Pr \in \mathcal{PR}\}$ に対して，

- $\top \in O(S)$ if
  $\exists(\top, Pr) \in S.p \preceq_p p'$ implies
  $(\top, Pr') \in S$
- $\bot \in O(S)$ if
  $\exists(\bot, Pr) \in S.p \preceq_p p'$ implies
  $(\bot, Pr') \in S$

□

Res の単調性から，$\mathcal{T}^\infty$ におけるテストの観測は有限な値のテストの観測の近似の極限として得られる．

[系 7]　$\mathbf{T}$ を $[T_\omega]$ の有向集合であるとする．このとき，任意の P_R に対して，

$$O(\sqcup Res(P_R, \mathbf{T})) = O(Res(P_R, \sqcup\mathbf{T}))$$

□

4.4　値テストの例

(1) $\Sigma_{vc} = \{s, 0\}$, $\Sigma_{vf} = \emptyset$

$$R_1 = \emptyset$$
$$P_1 = a?x.[x == s(y)].b!y.\mathtt{NIL}$$

プログラム P_{1R_1} は s からなる無限のストリーム $s(s(s(\cdots$ を a から入力して，b から同じストリームを出力するテストに対して成功する．

$$T = \sqcup\{a?s^n(\omega).b!s^{(n-1)}(\omega).SUCC; n \geq 1\}$$

に対して，$O(Res(P_{1R_1}, T)) = \{\top\}$

(2) $\Sigma_{vc} = \{s, 0, cons, nil\}$, $\Sigma_{vf} = \{nlist\}$

$$R_2 = \{nlist(x) \rightarrow cons(x, nlist(s(x)))\}$$
$$P_2 = a?x.[nlist(x) == cons(y, z)].b!y.c!z\mathtt{NIL}$$

プログラム P_{2R_2} は自分で整数リストを生成し，その先頭を b に出力してから，残りを c から出力する．

$$T = \sqcup\{a?0.b!0.c!t_n.SUCC; n \geq 0\}$$

に対して，$O(Res(P_{2R_2})) = \{\top\}$ ここで，

$$t_i = cons(0, cons(1, \ldots, cons(i-1, \omega)\ldots))$$

(3) $\Sigma_{vc} = \{0\}$, $\Sigma_{vf} = \{f\}$

$$R_3 = \emptyset$$
$$P_3 = a?x.[x == y].b!y.\mathtt{NIL}$$

プログラム P_{3R_3} は厳密な条件分岐を表す．すなわち，

$$T = \{a?\omega.b!\omega.SUCC\}$$

に対して，$O(Res(P_{3R_3}, T)) = \{\bot\}$
一方，

$$T' = \{a?0.b!0.SUCC\}$$

に対して，$O(Res(P_{3R_3}, T')) = \{\top\}$ である．

5　テスト意味論

本節では，値テストの結果に基づく意味論を定義し，PL_v^0 において値の計算が通信プロセスの振舞いとしてどのように取り扱われるかを検討する．

通信プロセスと値テストによる観測に対して次の関係を定める．

$\bot \notin O(Res(P_R, T))$ のとき，
　P_R *must* T
$\top \in O(Res(P_R, T))$ のとき，
　P_R *may* T

この関係に基づいて，テスト擬順序 [4] を次のように定める．

[**定義** 8] (テスト擬順序 $\sqsubseteq$)
PL_v^0 のプログラム P_R と P'_R に対して

(i) 任意の $T \in \mathcal{T}^\infty$ に対して，$P_R\ must\ T$ ならば $P'_R\ must\ T$ であるとき，$P_R \sqsubseteq_{MUST} P'_R$

(ii) 任意の $T \in \mathcal{T}^\infty$ に対して，$P_R\ may\ T$ ならば $P'_R\ must\ T$ であるとき，$P_R \sqsubseteq_{MAY} P'_R$

(iii) $P_R \sqsubseteq_{MUST} P'_R$ かつ $P_R \sqsubseteq_{MAY} P'_R$ であるとき，$P_R \sqsubseteq P'_R$

□

[**補題** 9]　自由な変数の出現が s の中だけであるような動作式 $[s\ ==\ t].P$ と値書き換え系 R に対して，s の変数を全て ω で置き換えることによって得られるプログラムを $[s_\omega == t].P$ とする．このとき，$T \in \mathcal{T}^\infty$ に対して、

$$Res(([s_\omega == t].P)_R, T) \preceq_p Res(P_R, T)$$

□

一般的には，値による分岐を加えることにより，テスト意味論では非決定性が増加する．

[**命題** 10]

$$([s_\omega == t].P)_R \sqsubseteq P_R$$

□

とくに任意の値書き換え系 R に対して，

$$[\omega == t].P_R \sqsubseteq \Omega_R$$

となる．

6　おわりに

本稿では，値の計算機構を導入することによって通信プロセスの枠組みの拡張を行なった．このような拡張によって値の計算に対して必ずしも停止性を要求しなくても，値の計算を通信プロセスの振舞いの中に導入することが可能になる．このことは，通信プロセスを並行プログラムの実行モデルとしてとらえるために必要な条件であると考えられる．

ここでは，通信プロセスの意味論としてテスト等価性 [4][6] を拡張した定式化を行なった．この拡張において，止まらない値の計算はプロセスの振舞いにおいて情報を減少させることを示した．また，止まらない値計算は一律に無意味であるとはせず，意味のある無限の値計算は条件分岐に利用できるような機構を導入した．このことにより，ストリームや無限リストのようなデータに対する通信プロセスの意味付けが可能になった．観測等価などの通信プロセスの他の意味論に対しては，テストの概念にいくつかの概念を付け加えることでとらえることができることが示されている [7] ので，本稿の拡張は他の意味論にも適用できると考えられる．

通信プロセスに値の概念を陽に導入した体系として LOTOS[13] が提案され，通信プロトコルの記述などが行なわれている．本稿の枠組みは，LOTOS における等式による値仕様の記述を項書換え系と見なして，実際に証明を行なう実行系であるとも考えられる．LOTOS における値の基本的な意味定義は抽象データ型の始代数意味論であるので，本稿における値項に対する意味定義と同じである．しかし，値の評価を振舞いに含め，途中結果でも必要な情報が得られれば，条件分岐が行なわれる点が異なっている．

PL_v^0 は通信プロセスとしては有限な体系である．今回，通信プロセスに再帰の機構を導入しなかったのは値計算の導入に議論の焦点を絞るためであり，無意味な通信プロセスは無限な振舞いを行なうプロセスの意味定義においてはじめて意味を持つ．このような体系の拡張は今後の課題である．

本稿の議論では値書き換え系を固定しているが，テストおよびその観測という観点からは，全体的な構成子の存在を仮定すれば，値書き換え系を固定して議論する必然性はない．この点の検討は今後の課題である．

謝辞

本研究を進めるにあたり，大変有益な御教示を頂いた岐阜大学 直井徹助教授，御討論とコメントを頂いた NTT 基礎研究所 真野健氏，ならびに，日頃御討論頂く研究室の皆様に感謝致します．

参考文献

[1] Milner,R.:*A Calculus of Communicating Systems*, LNCS Vol.92, (1980)

[2] Milner,R.:*Communication and Concurrency*, Prentice-Hall, (1989)

[3] Hoare,C.A.R.:*Communicating Sequential Processes*, Prentice-Hall, (1985)

[4] Hennessy,M.:*Algebraic Theory of Processes*, MIT Press, (1988)

[5] Hennessy,M. and A.Ingolfsdottir: "A Theory of Communicating Processes With Value-passing", LNCS Vol.443, 209–219, (1990)

[6] De Nicola,R. and M.Hennessy: "Testing Equivalences For Processes", Theoretical Computer Science Vol.34, pp.83–133, (1984)

[7] Abramsky,S.: "Observation Equivalence as a Testing Equivalence", Theoretical Computer Science 53, pp.225–241, (1987)

[8] Walker,D.:"π-calculus Semantics of Object-Oriented Programming Languages", LNCS Vol.526, pp.532–547, (1991)

[9] Smolka,S. and R.Strom:"A CCS Semantics for NIL", in Formal Description of Programming Concepts - III (Wirsing ed.), pp.347–373, North Holland, (1987)

[10] Goguen,J.,J.Thatcher,E.Wagner and J.Wright: "Initial Algebra Semantics and Continuous Algebras", JACM Vol.24, pp.68–95, (1977)

[11] 直井, 稲垣:" 項書き換え系の意味論と自由連続代数 ", 信学論 Vol.J71–D No.6, pp.942–949, (1988)

[12] Naoi,T. and Y.Inagaki:"Recursive Functions on a Completed Set of Natural Numbers", Technical Report of IEICE COMP90–1, (1990)

[13] Brinksma,E.: "A Tutorial on LOTOS", Protocol Specification, testing, and Verification, V (1986), pp.171–194

[14] 結縁, 坂部, 稲垣: "CCS によるモニタの動作の形式的記述 ", 信学論 Vol.J73–D–1, pp.683–692, (1990)