等式による　ＬＯＴＯＳ　プロセスの記述と解釈

１Ｊ－９　　佐々木修二*，富樫敦**，野口正一**

*　ＮＴＴ東北支社
**　東北大学電気通信研究所

１．はじめに

LOTOS(Langage Of Temporal Ordering Specification)は，分散処理システムにおける，OSI アーキテクチャの各層のサービス定義やプロトコル仕様を形式的に記述することを目的とし，ISO によって開発され，国際標準として勧告されているプロセス記述言語である[1].

本稿では，データタイプを含まない Basic LOTOS を対象に，プロセスの振る舞いを Bergstra, Klop の ACPτ [2] のように等式による公理で表現する．また，公理系の一部を左から右への書き換え規則とみなしたとき，LOTOS におけるプロセスの遷移は，項の書き換えによって解釈できることを示す．

２．ＬＯＴＯＳ

LOTOS において，プロセスは，その環境と通信するブラックボックスである．プロセス記述の仕方は，その環境と通信する能力を記述したプロセス表現による．プロセスは”インタラクション”（イベント）によって環境と通信する．イベントは，２つのプロセスの間に存在する同期型通信の単位である．

プロセス定義あるいはプロセス抽象の形式は次で与えられる．

process
<process-identifier> <parameter-list>
:= <behaviour-expression>
endproc

ここで，<process-identifier> は名前であり，この名前によってそのプロセスが参照されうる．<behaviour-expression> はプロセスの観測可能な振る舞いを定義するプロセス表現であり，表１の LOTOS の欄で掲げたプロセス構成演算子によって構成される．<parameter-list> の内容は，プロセスの型を定める．Basic LOTOS の場合，これはプロセスの潜在的な外部イベントのリストである．

LOTOS における観測可能なプロセスの振る舞いは，プロセス表現によって表される．プロセス表現は，イベントが起こりうる順序を形式的に関係づけ，結果的に，これらの表現に対する構成規則が LOTOS の本質となる．

３．等式によるＬＯＴＯＳの記述

LOTOS では，プロセスはその環境とインタラクションによって通信が行われる．本稿では，環境の存在なしにプロセスの通信が行われるものとする．LOTOS プロセスを記述するに当たり，LOTOS で用いられる記号と本稿での記号の対応を表１に示す．ここで，left-composition [L[，middle-composition [L] を新たに導入したのは，技術的な理由による．LOTOS プロセスの振る舞いは，表２の書き換え規則と等式によって定義される．

アトミックアクション（イベント）a,b,c,…,e の有限集合をＡとし，Ａに“内部アクション”τ を加えた集合を A_τ とする．

$$A_\tau = A \cup \{\tau\}.$$

アトミックアクションをａ，プロセスをｐ，p_1，p_2 で表わすと，プロセスは，次の BNF 表現で定義される．

$$\begin{aligned} p = \ & \text{stop} \mid \text{exit} \mid \tau.p_1 \mid a.p_1 \mid \\ & p_1 + p_2 \mid p_1 \vert\vert\vert p_2 \mid p_1 \parallel p_2 \mid \\ & p_1 \vert L \vert p_2 \mid p_1 [L[p_2 \mid p_1 [L] p_2 \mid \\ & p_1 \gg p_2 \mid p_1 [> p_2 \mid \tau_I(p_1) \end{aligned}$$

ここで，”stop”と”exit”は，それぞれ失敗終了（デッドロック）と成功終了を意味するプロセスを表わす．また，τ_I は，集合 $I \subset A$ に含まれるアトミックアクションを，内部アクション τ に名前替えする演算子で，LOTOS の hide 演算子に対応する．

表１　記号の対応

LOTOS	公理系	機能
;	.	action prefix
[]	＋	choice
>>	>>	sequential composition
$\vert\vert\vert$	$\vert\vert\vert$	independent parallel composition
$\parallel$	$\parallel$	dependent parallel composition
$\vert[\ ]\vert$	$\vert L \vert$	general parallel composition
なし	[L[	left-composition
なし	[L]	middle-composition
hide	τ_I	hiding operator
i	τ	internal action(event)
[>	[>	disruption composition
stop	stop	inaction／deadlock
exit	exit	sucessful termination

表２　ＬＯＴＯＳ等式表現の公理系
（ｘ，ｙ，ｚとａ，ｂは，それぞれプロセスとアクション（τ を含む）を値として取る変数であり，$L \subset A$ とする．* 印の等式は公理から得られる定理である．）

A1	$x + y = y + x$
A2	$x + (y + z) = (x + y) + z$
A3	$x + \text{stop} = x$
A4	$x + x = x$
SC1	$\text{stop} \gg x = \text{stop}$
SC2	$\text{exit} \gg x = \tau . x$
SC3	$a.x \gg y = a.(x \gg y)$
SC4	$(x + y) \gg z = x \gg z + y \gg z$
SC5	$x \gg (y \gg z) = (x \gg y) \gg z$
FM1	$x \vert\vert\vert y = x \vert \phi \vert y$
FM2*	$x \vert\vert\vert y = y \vert\vert\vert x$
FM3*	$x \vert\vert\vert (y \vert\vert\vert z) = (x \vert\vert\vert y) \vert\vert\vert z$
SM1	$x \parallel y = x \vert A \vert y$　Ａは全てのイベントのリスト
SM2*	$x \parallel y = y \parallel x$

Description and Interpretation of LOTOS Process using Equations
Syuji SASAKI, Atushi TOGASHI, Syoichi NOGUCHI
* Tohoku Telecommunications Service Region, NTT
** Research Institute of Electrical Communication, Tohoku University

SM3*	$x \parallel (y \parallel z) = (x \parallel y) \parallel z$	
GM1	$x \vert L\vert y = x$ [L[$y + y$ [L[$x + x$ [L] y	
GM2	stop [L[x = stop	
GM3	exit [L[x = exit	
GM4	$a.x$ [L[$y = a.(x \vert L\vert y)$	if $a \notin L$
GM5	$a.x$ [L[y = stop	if $a \in L$
GM6	$(x+y)$ [L[$z = x$ [L[$z + y$ [L[z	
GM7	stop [L] x = stop	
GM8	x [L] stop = stop	
GM9	exit [L] exit = exit	
GM10	$a.x$ [L] $a.y = a.(x \vert L\vert y)$	if $a \in L$
GM11	$a.x$ [L] y = stop	if $a \notin L$
GM12	x [L] $a.y$ = stop	if $a \notin L$
GM13	$(x+y)$ [L] $z = x$ [L] $z + y$ [L] z	
GM13	x [L] $(y+z) = x$ [L] $y + x$ [L] z	
GM14*	$x \vert L\vert y = y \vert L\vert x$	
GM15*	$x \vert L\vert (y \vert L\vert z) = (x \vert L\vert y) \vert L\vert z$	
TI1	$\tau_I(a) = a$	if $a \notin I$
TI2	$\tau_I(a) = \tau$	if $a \in I$
TI3	$\tau_I(x+y) = \tau_I(x) + \tau_I(y)$	
TI4	$\tau_I(a.x) = \tau_I(a).\tau_I(x)$	
TI5	τ_I(stop) = stop	
TI6	τ_I(exit) = exit	
DP1	stop [> $x = x$	
DP2	exit [> x = exit + x	
DP3	$a.x$ [> $y = a.(x$ [> $y) + y$	
DP4	$(x+y)$ [> $z = x$ [> $z + y$ [> z	

4．等式系によるＬＯＴＯＳプロセスの解釈

LOTOSでは，プロセスの意味は LTS(Labeld Trnsition System)によって解釈される．

【定義１】 ラベル付き遷移システム
LTS は４項組 <S,Act,T,s_0>である．ここで，S は状態の集合，Act はアクションの集合，Tは
$T = \{ -a\rightarrow \mid -a\rightarrow \subseteq S \times S, a \in Act \}$
となる遷移関係の集合，$s_0 \in S$は初期状態である．
cur－a→next とは，アクション a の生起による状態 cur から状態 next の遷移を表す． □

表２の公理の一部を左から右への書き換え規則とみなすと，プロセスの振る舞いは項の書き換えで解釈される．

【定義２】 項書き換えシステムR
Rを(A1)～(A4)以外の等式（導出定理は除く）を左辺から右辺への書き換え規則とみなした項書き換えシステム (TRS : Term Rewriting System) とする． □
ここで，(A1)～(A4)の等式を用いないのは，停止性を満たさないからである．
Rを用いて LOTOS のプロセスは "." と"+" のみを用いて表すことができる．

【例１】 次の LOTOS 表現を等式系で書き換える．

```
process  duplex_buffer[in_a,in_b,out_a,out_b]:=
             buffer[in_a,out_a]|||buffer[in_b,out_b]
where
   process  buffer[IN,OUT]:=
               IN;OUT;stop
   endproc
endproc
```

等式表現で書き換える

```
duplex_buffer[in_a,in_b,out_a,out_b]
  →  buffer[in_a,out_a]|||buffer[in_b,out_b]
  →  in_a.out_a.stop|||in_b.out_b.stop
 <等式 FM1,FM2,FM4により書き換える>
  *→ in_a.(out_a.(in_b.out_b.stop)
           +in_b.(out_b.out_a.stop
                 +out_a.out_b.stop))
      +in_b.(out_b.(in_a.out_a.stop
             +in_a.(out_a.out_b.stop
                   +out_b.out_a.stop ))
```

項書き換えシステムRの合流性と停止性を示す．

【定理１】 合流性，停止性
公理 (A1)～(A4) を法とするとき，Rは変数を含まない項に関して合流性と停止性を満たす． □

次に，非決定的な選択"+"を考慮した項書換えシステムを与える．

【定義３】 項書換えシステムR'
R'を，Rに次の書き換え規則
$x + y \triangleright x$
を付け加えた，項書き換えシステムとする． □

項書き換えシステムR'によって，定理２が得られる．

【定理２】 LTS と TRSの等価性
LOTOS のプロセス定義が与えられたとき，P_1，P_2をLOTOS のプロセス表現，P_1'，P_2'を対応する等式によるプロセス表現とすると，以下の(1)，(2)は等価である．
(1) LOTOS において
$P_1 - a_1, \cdots, a_n \rightarrow P_2$　　ただし $a_i \in A\tau$
となる遷移がある．
(2) 公理 (A1)～(A4)を法とする TRS R' のもとで
$P_1' \xrightarrow{*} a_1.\cdots.a_n.P_2'$
と書き換えられる．
なお，stop と exit について bisimulation のもとで等価なプロセスは同一視し，δとというアクションは考慮せず，exit －δ→stop という遷移は排除した． □

5．むすび

LOTOS のプロセスを等式による公理で表現することによって，LOTOS のプロセスの遷移は，項の書き換えと等価であると解釈される．今後は，LOTOS を包含し，演算子をユーザーが定義できるような，プロセス記述言語の設計と支援システムの開発をする予定である．

参考文献

[1] ISO :Information processing - Open Systems Interconnection - LOTOS - a Formal Description Technique Based on the Temporal Ordering of Observational Behaviour,ISO 8807 (1989)

[2] Bergstra,J.A. and Klop,J.W.:Algebra of Communicationg Processes with Abstraction,theor.Comput. Sci.Vol.37,pp.77-121(1985)