

## LOTOS の状態遷移的解釈

4T-6

山口 基志

高橋 薫

白鳥 則郎

野口 正一

東北大学電気通信研究所

## 1. はじめに

通信システム及び通信ネットワークの発展にともない、それらの通信プロトコルの仕様を形式的に記述するための方法であるFDT (Formal Description Technique)の重要性が高まっている。このような状況のもと、プロセス代数に基礎をおくLOTOS<sup>1)</sup>と、拡張型有限状態機械を基礎とするEstelle<sup>2)</sup>という、2つのFDTがISOにおいて提案された。今後、統合的にFDTの研究を進めていくうえで、また、短期的展望としては実装あるいは検証系を実現するためのより現実的なアプローチとして、この2つのFDTに関して、一方のFDTで記述された仕様を他方の仕様に変換することは非常に有益である。

以上の背景に基づき、本稿ではLOTOSのセマンティクスを状態遷移的に解釈することにより、LOTOSの仕様をEstelleの仕様に変換するための1つの手法を提案する。

従来成果としてはLOTOSのアクション木をそのまま状態遷移図に対応づける手法が発表されているが<sup>3)</sup>、筆者らの手法ではより構造的なアプローチとなっている。

## 2. 変換における対象領域の設定

本研究では、変換対象とするFDTの記述的・意味的特性、および技術的な理由により、以下に述べるような制限・拡張をLOTOS, Estelleに加え、変換の対象範囲を設定する。

## &lt;LOTOSへの制限&gt;

- LOTOSのデータ部は扱わない。
- 状態数及び遷移数は有限とする。
- LOTOSでの多重同期は扱わない。(2プロセス間の同期のみを扱う。但し"環境との同期"は考慮する。)
- プロセスの再帰呼び出しの禁止。(Estelleの親子優先権原則による。)
- インタラクションのタイプとしては値の引渡しのみ扱う。値のマッチング、値の生成については、今後の検討課題とする。

## &lt;Estelleへの制限、拡張&gt;

- Estelleの通信は同期通信とする。(queueの廃止)
- 特殊インタラクションポイントenvの導入  
LOTOSでの"イベントの環境との同期"というセマンティクスをEstelleにおいても保存するために、Estelle文法に"環境との同期"用の特殊インタラクションポイントenvの

機能を新たに付加する。

## ● 連動遷移の導入(linkage節)

LOTOSでの同期イベントは、Estelleでは異なるモジュール間の2つの遷移が同期すると解釈する。この2つの遷移を"連動遷移"という。このとき一方の遷移と連動する遷移をEstelle文中で明示するために、linkage節をEstelleのclause-groupに新たに付加する。

[定義]: linkage t1, ..., tn

ただし t1, ..., tn は遷移ラベルリスト

- Estelleのインタラクションポイント名のリラベリングについては、init文中で明示的に行う。

(例: init M with m\_body[a1/g1, ..., an/gn] )

## 3. 変換アルゴリズム

## &lt;基本原則&gt;

- LOTOSのセマンティクスを保存する。
- LOTOSのアクション木を状態遷移図に対応させる<sup>3)</sup>。
- 構造に着目する。そのためにLOTOSのオペレータに着目する。つまりオペレータの両辺をモジュールと解釈し、オペレータはそのモジュール間の関係を表していると解釈する。ただし、[] (choice オペレータ)については別の解釈を与える。

[解釈]: <モジュール> OP <モジュール>

OP: [] 以外のLOTOSのオペレータ

## &lt;構造的変換&gt; (⇒: 変換を表す)

- イベント ⇒ 遷移、インタラクション
- プロセス、プロセス初期化、動作式 ⇒ モジュール (属性: process)
- ゲート ⇒ インタラクションポイントリンク(attach, connect)  
ただしリンクが存在するのは同期ゲートの場合。
- オペレータ ⇒ モジュール間の関係。両モジュールにどのような操作が為されているかを表す。

## &lt;操作的変換&gt;

(1) 遷移 a;B ⇒ 遷移 (<a;B> - a → <B> : <状態>)

## (2) 通信

① A[[g1, ..., gn]]B, A||B の場合

case1 値の引渡しの無い場合

・同期イベント(g1, ..., gn) ⇒ 連動遷移

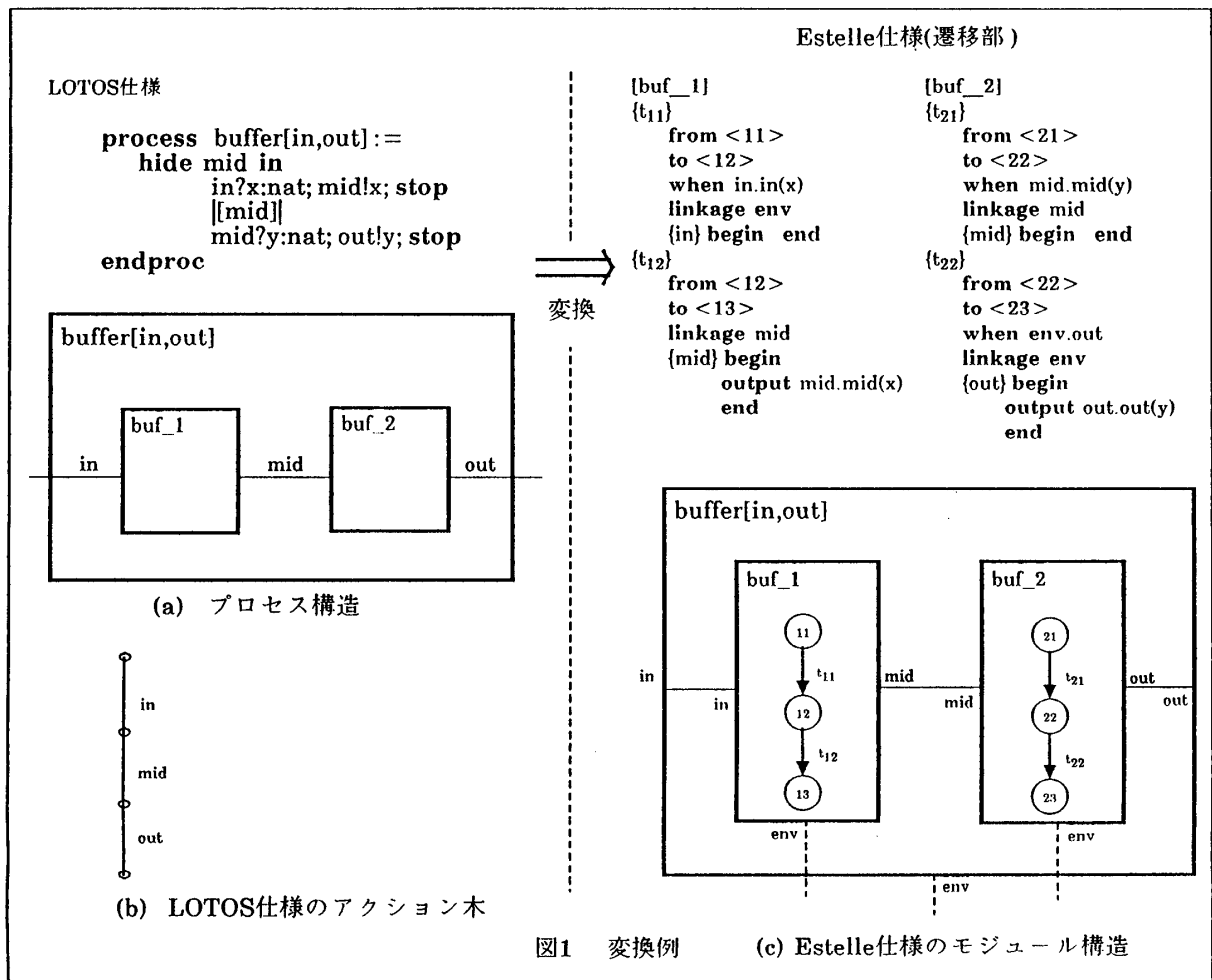
・それ以外のイベント ⇒ 遷移(インターリーブ)

case2 値の引渡しのある場合

例: g!x; stop[[g]g?u:t; stop

⇒ 連動遷移

・インタラクションによるパラメータの授受



- ②  $A \gg B, \Lambda \gg \text{accept } x_1:t_1, \dots, x_n:t_n \text{ in } B$  の場合
  - 連動遷移となる。Aは正常終了状態になった後、**exit**インタラクションをBに送る。Bは**exit**を受け取って初めて自分の遷移がおこなえる。後者は**exit**インタラクションにパラメータが付随する場合である。
- (3)並列 (インターリーブ)
  - $A \parallel B, A [g_1, \dots, g_n] B$  (非同期イベント)
    - A, Bのイベントは遷移に変換されるが、各遷移はどのような順番にインターリーブされてもよい。もちろん各モジュール内での遷移の順番は保存される。
- (4)選択 例:  $a; \text{stop} [b; \text{stop}]$  の場合
  - []オペレータの両辺のモジュール化は行わない。この場合は1つの状態において純粋に遷移条件だけを比較して生起する遷移が決定される。(when節、provided節)もし遷移条件を満たす遷移が複数存在するならば、非決定的にその内の1つが選択され生起する。ただしこの遷移条件には"環境の介入"という要素も含まれてもよい。
- (5)隠蔽  $\text{hide } g_1, \dots, g_n \text{ in } A$  の場合
  - **hiding**されたイベントは遷移に変換されるが、この遷移の遷移条件に環境は介入

- できない。
- (6)自発遷移  $i$ 
  - $i \Rightarrow$  自発遷移。環境は介入できない。

**4.変換例**

LOTOS仕様からEstelle仕様への変換例を図1に示す。

**5.まとめ**

- 本稿で提案したLOTOS仕様をEstelle仕様に変換するアルゴリズムの主な特徴を以下に示す。
- 構造的な対応づけが行える。
  - Estelleの仕様に直接に変換が行える。

**参考文献**

- 1) ISO: "LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour", ISO8807 (1989).
- 2) ISO: "Estelle - A Formal Description Technique Based on Extended State Transition Model", ISO/DIS9074 (1987).
- 3) Günter Karjoth: "Implementing Process Algebra Specifications by State Machines", Protocol Specification, Testing and Verification VIII (1988).