# シミュレーション関係に基づくLOTOS仕様の検証アルゴリズム *

4V−4

山野 敬一郎　太田 正孝 † 高橋 薫 ‡
AIC †　東北大学 ‡

## 1　はじめに

近年、プロトコルなど通信システムの厳密な仕様化を行うために、各種の形式的記述技法(FDT: Formal Description Techniques)が提案されている。

このFDTの1つとして開発されたLOTOS(Language Of Temporal Ordering Specification)[1]は、数学的モデルをベースとしており記述・検証能力が高いため、その活用が期待されている。例えば、開発工程において弱双模倣(Weak-bisimulation)関係[2]の概念を適用することにより、LOTOSで記述された上位レベルと下位レベルの2つの仕様間の無矛盾性の検証を行い、仕様が正しく詳細化されているかどうかを確認することが可能である。

しかし、仕様の段階的な開発においては、複数の上位レベルの仕様を組み合わせて下位レベルの仕様を開発する場合や、下位レベルの仕様に上位レベルの仕様には含まれていない例外処理などの付加的な情報を加える場合がありうる。このような場合には、上記の弱双模倣等価関係は成立せず、むしろ片方向のシミュレーション関係を検証の概念として用いることが適当である。

本報告では、まずシミュレーション関係の概念を示し、次にその判定アルゴリズムを示す。さらに適用例として、この判定アルゴリズムに基づいたLOTOS仕様の検証システムの概要を示す。

## 2　シミュレーション関係の諸定義

本節では、まずLOTOS仕様の形式的なモデルである遷移システム(TS: Transition System)の定義を示す。さらに、この遷移システムに基づき、シミュレーション関係を定義する。

**定義1　遷移システム**　(以下、TSと略記)

TS Sys は4項組 $< S, Act, \longrightarrow, s_0 >$ である。ここで、

S　：状態の集合

Act：外部から観測可能なアクションの集合と観測可能ではない内部アクションiから成るアクション集合

$\longrightarrow$ ：遷移関係 ($\longrightarrow \subseteq S \times Act \times S$)

$s_0$　：Sysの初期状態 ($s_0 \in S$)

である。

$(s, \alpha, q) \in \longrightarrow$ のとき、アクション $\alpha$ による状態sから状態qへの遷移を明確に表すのに、$s \xrightarrow{\alpha} q$ と書く。

TS Sys が有限のグラフとして表現可能であるとき、Sysは有限なTSであるという。 □

なお、上記の遷移システムはtransition導出体系に基づき、LOTOSの動作式から生成することが可能である。以下では、このTSが有限の場合について、話を進める。

**定義2**

$Sys =< S, Act, \longrightarrow, s_0 >$ をTSとする。$t \in Act^*$ のとき、$\hat{t} \in (Act - \{i\})^*$ はtからiのすべてのオカレンスを削ることによって得られる系列である。$t = \alpha_1 \cdots \alpha_n \in Act^*$ のとき、$s \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} s'$ と表す。$t = \varepsilon$ (空系列)のとき、任意の $s \in S$ について、$s \xrightarrow{\varepsilon} s$ である。$t = \alpha_1 \cdots \alpha_n \in Act^*$ のとき、$s(\xrightarrow{i})^* \xrightarrow{\alpha_1} (\xrightarrow{i})^* \cdots (\xrightarrow{i})^* \xrightarrow{\alpha_n} (\xrightarrow{i})^* s'$ ならば、$s \stackrel{\hat{t}}{\Longrightarrow} s'$ と表す。 □

**定義3　シミュレーション関係**

Sys1 と Sys2 をTSとする。

$Sys1 =< S_1, Act, \longrightarrow_1, q_1 >$

$Sys2 =< S_2, Act, \longrightarrow_2, q_2 >$

次の条件を満たす関係 $S \subseteq S_1 \times S_2$ をSys1からSys2へのシミュレーション関係という：

$(s, q) \in S$ ならばすべての $\alpha \in Act$ について、
$s \xrightarrow{\alpha} s'$ のとき、$q \stackrel{\hat{\alpha}}{\Longrightarrow} q'$ なる $q'$ が存在し、
$(s', q') \in S$ □

以上の定義より、シミュレーション関係は観測可能なアクション系列の関係として一般化でき、さらに自然な拡張としてTS間の関係として定義できる。これをもとに、LOTOSの仕様間のシミュレーション関係を次のように定義する。

**定義4　LOTOS仕様間のシミュレーション関係**

LOTOS仕様 Spec1 と Spec2 に対応するTSを、それぞれ Sys1 と Sys2 とする。Sys1 から Sys2 へのシミュレーション関係が成り立つとき、かつそのときに限り Spec2 は Spec1 をシミュレートする、あるいは、Spec1 は Spec2 によってシミュレートされるといい、$Spec1 \ll Spec2$ と書く。 □

---

*A Verification Algorithm for LOTOS Specifications based on the Simulation Relation

†Keiichirou YAMANO, Masataka OHTA

‡Kaoru TAKAHASHI

†Advanced Intelligent Communication System Laboratories.

‡Tohoku University.

## 3 判定アルゴリズム

本節では、TS 間の関係に基づくシミュレーション関係の判定アルゴリズムとその諸性質を示す。判定アルゴリズムは、まず関係 $\mathbf{R}^{(0)}$ を与えることを基本とし、$\mathbf{R}^{(k)}$ を前提として $\mathbf{R}^{(k+1)}$ を導出するという帰納的な定義によって与えられる。

『判定アルゴリズム』

有限な TS Sys1 と Sys2 が与えられたとする。

[基本ステップ] $\mathbf{R}^{(0)}$ を求める
$\mathbf{R}^{(0)}$ として、次のように、全称関係を構成する。

$$\mathbf{R}^{(0)} := S_1 \times S_2$$

[帰納的ステップ] $\mathbf{R}^{(k)}$ から $\mathbf{R}^{(k+1)}$ を求める
$\mathbf{R}^{(k)}$ $(k \geq 0)$ まで求められていたとする。

(1) まず初期的に、$\mathbf{R}^{(k+1)} := \emptyset$ (空集合) とする。

(2) $\mathbf{R}^{(k)}$ の各要素 $(s,q)$ について以下を行う：
各 $\alpha \in Act$ について、$s \xrightarrow{\alpha} s'$ のとき、$q \stackrel{\hat{\alpha}}{\Rightarrow} q'$ なる $q'$ が存在し、$(s',q') \in \mathbf{R}^{(k)}$ が成立するとき、$\mathbf{R}^{(k+1)} := \mathbf{R}^{(k+1)} \cup \{(s,q)\}$ とする。

(3) $\mathbf{R}^{(k+1)} = \mathbf{R}$ ならば (以降、同じ関係しか生成しないから) 終了し、この関係を $\mathbf{R}$ とおく。
$\mathbf{R}^{(k+1)} \neq \mathbf{R}^{(k)}$ ならばこのステップを繰り返す。 □

ここで、Sys1 と Sys2 が有限な TS であれば、この判定アルゴリズムは有限のステップで停止し、帰納的ステップの (3) に示すように関係 R を得る。このとき、関係 R は Sys1 から Sys2 へのシミュレーション関係として最大のものであり、Sys1 から Sys2 への任意のシミュレーション関係 S をその部分集合として含む。逆に、シミュレーション関係が成り立たない場合には、$\mathbf{R} = \emptyset$ が得られる。

## 4 適用例

前節で与えた判定アルゴリズムの適用例を示す。
この例は、上位レベルの LOTOS 仕様 Spec1 と、それを詳細化して例外処理を加えた下位レベルの仕様 Spec2 を与え、Spec2 が Spec1 をシミュレートしているかどうかを判定するものである。

```
Spec1 ≡P[a,b,c,d]
    P[a,b,c,d] := a;b;stop [] a;c;d;P[a,b,c,d]
Spec2 ≡Q[a,b,c,d]
    Q[a,b,c,d] := a; (b;stop [] c;Q1[a,b,c,d])
    Q1[a,b,c,d] := i;Q1[a,b,c,d] [] i;d;Q[a,b,c,d]
```

Spec1 と Spec2 に対応する遷移システム Sys1 と Sys2 は、図 1に示す通りである。
これに判定アルゴリズムを適用すると、次のような R が得られる。

$$\mathbf{R} = \{ \ (p_1,q_1),(p_2,q_2),(p_4,q_1),(p_4,q_2),(q_4,q_3), (p_4,q_4),(p_4,q_5),(p_3,q_2),(p_5,q_4),(p_5,q_5) \ \}$$

従って $Spec1 \ll Spec2$、つまり、pec2 は Spec1 をシミュレートしていることが結論づけられる。

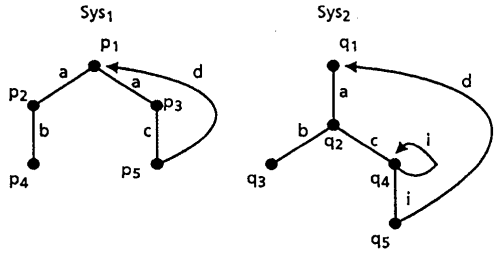


図 1: LOTOS 仕様に対応する遷移システム

## 5 検証システムの開発

以上に述べたような判定アルゴリズムを利用し、現在我々は図 2に示すような LOTOS 仕様の検証システムを開発中である。

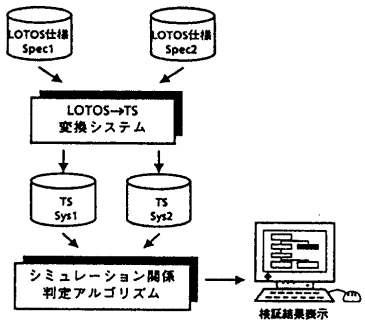


図 2: 仕様検証システム

本システムでは、入力として仕様の開発工程における上位レベルの LOTOS 仕様 Spec1 と下位レベルの仕様 Spec2 を与える。システムの内部では、これらの仕様をそれぞれ対応する遷移システム Sys1 と Sys2 に変換する。次に、これらの遷移システムに判定アルゴリズムを適用し、関係 R を得る。これにより、仕様間の対応関係がわかる。もし、$\mathbf{R} = \emptyset$ となる場合には、下位レベルの仕様が上位レベルの仕様を正しく反映していないことがわかる。最終的に、この検証結果を画面上に表示する。

## 6 おわりに

本報告では、LOTOS 仕様に対するシミュレーション関係を定義し、その判定アルゴリズムを示した。さらに、判定アルゴリズムに基づいた検証システムの概要を示した。現在は、検証システムの開発を行っている。
今後の課題としては、より実用的な検証システムの開発と、実際の通信システム等の仕様記述への適用が考えられる。

## 参考文献

[1] ISO : "Information processing systems - Open Systems Interconnection - LOTOS - A formal description technique based on the temporal ordering of observational behaviour," ISO8807 (1989).

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