プロセス仕様の模倣関係検証

2R－1

鈴木理創　米津光浩　山口文彦　中西正和
慶應義塾大学 大学院 理工学研究科 計算機科学専攻

1. はじめに

従来、システムの仕様記述は自然言語によって記述されてきた。しかし、曖昧さ等の点から、構文と意味が厳格に曖昧なく規定された形式記述言語を用いて記述することが望ましく、またその傾向が高まって来ている。

このような形式記述言語としてプロセス代数という数学的モデルがある。システム仕様をプロセス代数に基づき記述する利点の一つとして、その検証性がある。仕様の段階的な開発過程において、仕様と仕様の間の一貫性を検証するのに、弱双模倣等価性などのプロセスの間の等価性の概念を適用することが可能であり、その判定手続きも提案されている [6]。

しかし、仕様を増補的に開発していく場合には、後続段階の仕様は、前段階の仕様には含まれていない例外処理や新たな付加的な処理を含むことがあり、通常このような仕様間には弱双模倣等価性が成立しない。そこで、新たな検証概念として、仕様間の単方向の模倣関係というものが提案されている [1]。しかし、システムが複雑な場合、煩雑な手続きとなり、従って、自動的に検証するシステムを開発することが要求されて来ている [2]。

本研究では、この模倣関係自動検証システムを実装し、その有効性を検証することを目的とする。

2. プロセス代数

プロセス代数とは、現実世界でのプロセスを、構文論的には形式体系として、意味論的には代数としてモデル化した数学的モデルである。そのモデル化の違いから幾つかに分類されるが、本研究では、LOTOS(Language Of Temporal Ordering Specification) を用いて仕様を記述する。LOTOS では、記述の対象をプロセスと呼び、LOTOS による表現を動作式と呼ぶ。

定義 2..1 (動作式)

$E ::=$	**stop**	（プロセスの停止）
$\mid$	**exit**	（正常終了）
$\mid$	$a;E$	（アクションプレフィックス）
$\mid$	$E[]E$	（選択）
$\mid$	$E\|\|\|E$	（非同期並列）
$\mid$	$E\|\|E$	（同期並列）
$\mid$	$E\|[A]\|E$	（並列合成）
$\mid$	$E[>E$	（割り込み）
$\mid$	$E >> E$	（逐次合成）
$\mid$	**hide** A **in** E	（隠蔽）
$\mid$	$P[g_1,\cdots,g_k](\mathbf{e})$	（プロセス呼び出し）

Verification of simulation relation in specification of process
Riso SUZUKI　Mitsuhiro YONEZU
Fumihiko YAMAGUCHI　Masakazu NAKANISHI
Department of Computer Science, Faculty of Science and Technology, Keio University 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223, Japan

$a \in Act \cup \{i\}$ (Actはすべての観測可能な動作の有限集合)、$A \subset Act$、$k \in N$、eは式のベクトル。

このようにして記述された動作式は、遷移関係を導出することでラベル付き遷移システムを導くことができる。

定義 2..2 (ラベル付き遷移システム)

4 項組 $\langle S, A, T, s_0 \rangle$ で表す。

S	状態の集合
A	アクションの集合
$T(\subset S \times A \times S)$	遷移関係の集合
$s_0(\in S)$	初期状態

3. 検証性

システム仕様はその初期仕様から出発し、段階的に開発されていく。開発過程において、後続段階の仕様は

- 前段階仕様の等価な詳細化
- 例外処理機能や付加機能を加えた増補的なもの

がある。

前者の場合、弱双模倣の概念を用いた仕様間の検証が可能である。一方、後者の場合、次の模倣関係が有効な概念である。

3.1 模倣関係

模倣関係は、ソフトウェアシステムの開発を増補的に行っていくときの正しさの基準として使用される性格を有する。つまり、例外機能などを導入しつつ、前段階仕様の機能を拡充していくようなシステム開発過程でこの関係を用いた検証が効果を発揮する。

定義 3..1 (模倣関係)

$Sys1 = \langle S_1, Act, \rightarrow_1, \sigma_1 \rangle, Sys2 = \langle S_2, Act, \rightarrow_2, \sigma_2 \rangle$ を任意の遷移システムとする。次の条件を満たす関係 $\mathcal{R} \in S_1 \times S_2$ を $Sys1$ から $Sys2$ への模倣関係という。

$(s,q) \in \mathcal{R}$ とは、すべての $\alpha \in (Act - \{i\})^*$ について、$s \overset{\alpha}{\Rightarrow} s'$ である s' が存在するならば $q \overset{\alpha}{\Rightarrow} q'$ である q' が存在し、$(s',q') \in \mathcal{R}$ である。

3.2 判定手続き

関係 $\mathcal{R}^{(0)}$ を全関係として与え、$\mathcal{R}^{(k)}$ を以下のように帰納的に定義する。

基本ステップ　$\mathcal{R}^{(0)} = S_1 \times S_2$(全関係)

帰納的ステップ　$\mathcal{R}^{(k)} (k \geq 0)$ まで求められていたとする。

(1) 初期的に、$\mathcal{R}^{(k+1)} = \phi$(空集合) とする。

(2) $\mathcal{R}^{(k)}$ の各要素 (s,q) について以下を行う。

各 $\alpha \in Act$ について、$s \overset{\alpha}{\Rightarrow} s'$ である s' が存在するとき、$q \overset{\alpha}{\Rightarrow} q'$ なる q' が存在し、$(s',q') \in \mathcal{R}^{(k)}$ が成立するとき、

$\mathcal{R}^{(k+1)} = \mathcal{R}^{(k+1)} \cup (s,q)$ とする。

(3) $\mathcal{R}^{(k+1)} \neq \mathcal{R}^{(k)}$ ならば帰納的ステップを始めから繰り返す。$\mathcal{R}^{(k+1)} = \mathcal{R}^{(k)}$ ならば手続きを終了し、この関係を $\mathcal{R}$ とおく。この関係 $\mathcal{R}$ は $Sys1$ から $Sys2$ への模倣関係であり、$\mathcal{R}$ が初期状態対 (σ_1, σ_2) を含むならば $Sys1 \prec Sys2$ と定義する。なお、$\mathcal{R}$ に含まれる任意の状態対 (s, q) について $s \prec q$ が成り立つ。

4. 模倣関係検証システム

本研究で実装した模倣関係検証システムによる実行例を次に示す。

4.1 遷移システム

図 1 における $Sys1$ から $Sys2$ への検証を行った。

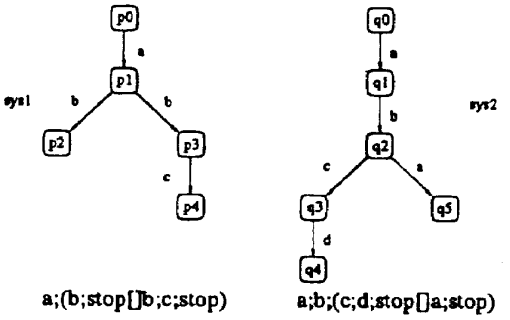

図 1: 遷移システム

```
>*sys1*
((P0 A P1) (P1 B P2) (P1 B P3) (P3 C P4))

>*sys2*
((Q0 A Q1) (Q1 B Q2) (Q2 C Q3) (Q2 A Q5) (Q3 D Q4))

>(main *sys1* *sys2*)
((P4 Q0) (P4 Q1) (P4 Q2) (P4 Q3) (P4 Q5) (P4 Q4) (P2 Q0) (P2 Q1)
 (P2 Q2) (P2 Q3) (P2 Q5) (P2 Q4) (P0 Q0) (P0 Q1) (P0 Q2) (P0 Q3)
 (P0 Q5) (P0 Q4) (P3 Q0) (P3 Q1) (P3 Q2) (P3 Q3) (P3 Q5) (P3 Q4)
 (P1 Q0) (P1 Q1) (P1 Q2) (P1 Q3) (P1 Q5) (P1 Q4))

((P1 Q1) (P3 Q2) (P0 Q2) (P0 Q0) (P2 Q4) (P2 Q5) (P2 Q3) (P2 Q2)
 (P2 Q1) (P2 Q0) (P4 Q4) (P4 Q5) (P4 Q3) (P4 Q2) (P4 Q1) (P4 Q0))

((P4 Q0) (P4 Q1) (P4 Q2) (P4 Q3) (P4 Q5) (P4 Q4) (P2 Q0) (P2 Q1)
 (P2 Q2) (P2 Q3) (P2 Q5) (P2 Q4) (P0 Q0) (P3 Q2) (P1 Q1))
T
```

図 2: 実行例

求められた関係に、初期状態対 (P0 Q0) が存在するので、$Sys1 \prec Sys2$ が成立する。

4.2 包摂アーキテクチャ

複雑なシステムとして、知能の新しい枠組として、Brooks が提唱した包摂アーキテクチャ[3][4][5] について検証を行った。高次のレベルのモジュールを順次付け加えていくという方法論で構成されるため、模倣関係における検証が有効であると考えられる。

図 3 のように、各モジュールは、拡張有限状態機械のネットワークで表される。つまり、互いに非同期に周期的動作をする複数のプロセスのネットワークととらえることができる。

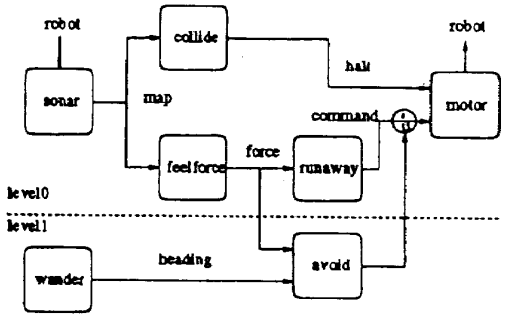

図 3: control system[3]

level0(前段階仕様) に、wander, avoid という付加的な処理を加え、level1(後続段階仕様) とする。

図 3 から求められる状態、遷移関係の数、そしてそれらに判定手続きを適用した結果は次のようになった。

表 1: 包摂アーキテクチャにおける結果

状態と遷移関係

	level0	level1
状態	180	1080
遷移関係	810	7020

関係

	関係の数
$\mathcal{R}^{(0)}$	199400
$\mathcal{R}^{(1)}$	176709
$\mathcal{R}^{(2)}$	176706
$\mathcal{R}^{(3)}$	176706

$\mathcal{R}^2$ と $\mathcal{R}^3$ の数が等しくなったので、判定手続きを終了する。求められた関係において、初期状態対が存在し、従って $level0 \prec level1$ が成立する。

5. 今後の課題

プロセス代数と同様に並行システムを記述するためのモデルとして、ペトリネットがある。ペトリネットの発火によるマーキングの遷移を遷移関係としてとらえることで、ペトリネットの模倣関係が検証できると考えられる。

また、実装した検証システムのオーダーは、$O(n^2)$ である。従って状態数が多くなると、計算量が増えてしまうという問題があり、より効率的な検証方法に関する考察も課題である。

参考文献

[1] 高橋 薫, 神長 裕明 著, コンピュータ通信シリーズ 4 分散システム設計の先端技法 仕様記述言語 LOTOS, 株式会社カットシステム, 1995

[2] 高橋 薫, 山野 敬一郎, 太田 正孝, プロセス仕様の検証のための模倣性判定法, 電子情報通信学会論文誌 D-I, Vol. J76-D-I No. 1 pp. 27–30, 1993

[3] Brooks, R. A. *A Robust Layered Control System for a Mobile Robot*, IEEE Trans. Robotics and Automation, Vol. 2, No.1, pp. 14–23, 1986

[4] Brooks, R. A. *Intelligence without reason*, IJCAI-91, pp. 569–595, 1991

[5] Brooks, R. A. *Intelligence without representation*, Artif. Intell., Intelligence, Vol. 47, pp. 139–159, 1991

[6] 鈴木 理創, プロセス等価性の自動証明器の実装, 第 52 回情報処理学会全国大会講演論文集, Vol. 4, pp. 5–6, 1996