3M-3

時間オートマトンのタイマ付き有限状態機械への変換法とその評価

岡田 嶺 †　　武内 剛 †　　樋口 昌宏 †
近畿大学大学院総合理工学研究科エレクトロニクス系工学専攻 †

1　はじめに

時間オートマトン（TA）[1] は複数のクロックによって時間に依存した動作を表現でき，リアルタイムシステムの仕様記述や検証時の実質的な標準モデルとなっている．一方，タイマ付き有限状態機械（FSM/T）[2] は指定された時間が経過するとタイムアウト通知を行うタイマ機能を利用して時間に依存した動作を表現できる．ソフトウェアによる実装を考えると，OS によって提供されているタイマ機能を利用するのが一般的であり，FSM/T の形式で与えられた仕様に基づいて実装することが望ましい．本稿では，TA を FSM/T に変換する方法を 2 つ提案する．また，それらの変換法に基づく変換システムを構築し，2 つの変換法を比較する為の変換実験を行う．

2　時間オートマトン

TA は有限オートマトンに時間の概念を付加したものであり，有限個の状態と有限個のクロックを持つ．全てのクロックは初期値 0 で，同じ割合で連続的に増加する．クロックは個別に 0 にリセットすることができる．状態，状態遷移にクロック制約（各クロックのクロック値の上限と下限を指定）を与えることができ，それぞれ状態に滞留することができる条件，状態遷移を実行することができる条件を表す．

TA を 6 字組 $A = (S, s_0, \Sigma, X, I, R)$ で定義する．S は状態の有限集合，$s_0 \in S$ は初期状態，Σ はイベントの有限集合，X はクロックの有限集合を表す．$I(s)$ は状態 s に対するクロック制約を表す．R は以下の 5 字組で表される状態遷移 $(s, s', \sigma, \varphi, X_R)$ の有限集合である．

- $s, s' \in S$　：遷移前の状態，遷移後の状態
- $\sigma \in \Sigma$　：イベント
- φ　：クロック制約
- $X_R \subseteq X$　：リセットするクロックの集合

TA は決定的であるとし，状態のクロック制約が満たされない場合，ある状態遷移が実行可能であるとする．TA の動作列は，時間経過と 1 個以上のイベント列が交互に現れる系列として表すことができる．TA の例を **Fig. 1** に示す．

On Translating Timed Automata into Finite State Machine with Timers
† Graduated School of Science and Engineering, Kinki University

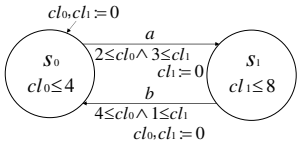

Fig. 1 Timed Automaton

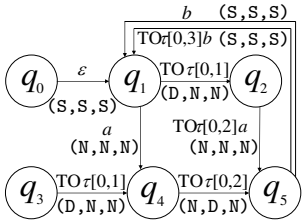

Fig. 2 FSM/T

3　タイマ付き有限状態機械

FSM/T は有限状態機械（FSM）と有限個のタイマによって構成されている．FSM は始動・解除命令によってタイマを始動・解除することができる．タイマは始動時にタイムアウト時間を指定し，始動後解除命令を受けないままタイムアウト時間が経過すると，FSM に対してタイムアウト通知を行う．

FSM/T を 5 字組 $F = (Q, q_0, \Sigma, T, E)$ で定義する．Q は状態の有限集合，$q_0 \in Q$ は初期状態，Σ はイベントの有限集合，T はタイマの有限集合を表す．E は以下の 4 字組で表される状態遷移 $(q, q', \sigma, \overline{p})$ の有限集合である．

- $q, q' \in Q$　：遷移前の状態，遷移後の状態
- $\sigma \in (\Sigma \cup T)\Sigma^*$：イベントまたはタイムアウト通知とそれを契機に連続して実行されるイベントの系列
- $\overline{p} \in \{\mathtt{S}(x), \mathtt{D}, \mathtt{N}\}^{|T|}$：タイマ操作ベクトル

FSM/T は決定的であるとする．タイマ操作ベクトルは各タイマに対する操作を指定し，$\mathtt{S}(x)$ はタイムアウト時間 x の計測の始動，D は解除，N は無操作を表す．FSM/T の動作列は，時間経過と 1 個以上のイベント列が交互に現れる系列として表すことができる．FSM/T の例を **Fig. 2** に示す．

4　変換法

TA $A = (S, s_0, \Sigma, X, I, R)$ を FSM/T $F(A)$ に変換する方法を 2 つ述べる．第 1 変換法で得られる FSM/T を $F1(A) = (Q,\ q_0,\ \Sigma,\ T1,\ E1)$，第 2 変換法で得られる FSM/T を $F2(A) = (Q,\ q_0,\ \Sigma,\ T2,\ E2)$ とする．

各クロック cl_j（$0 \le j < m$）について，クロック制約に現れる上下限値の集合を求め，昇順に並び替えて $V_j = \{t_{j,1}, \cdots, t_{j,jMAX}\}$ とし，cl_j の k 番目の区間 $\mathrm{int}(j,k)$ を

$$\mathrm{int}(j,k) = \begin{cases} (0,\ t_{j,1}) & (k = 0) \\ (t_{j,k},\ t_{j,k+1}) & (1 \le k < jMAX) \\ (t_{j,jMAX},\ \infty) & (k = jMAX) \end{cases}$$

とし，

$$INT_j = \{\mathrm{int}(j,k) \mid 0 \le k \le jMAX\}$$

とする．ただし，$(t_{j,k},\ t_{j,k+1}) = t_{j,k} \le cl_j < t_{j,k+1}$ である．

$Q = \{q_0\} \cup \{(s, \overline{\text{int}} = (\text{int}[0], \cdots, \text{int}[m-1])) \mid s \in S, \text{int}[j] \in INT_j\}$ とする．この時，$(s, \overline{\text{int}}) \in Q$ について $\overline{\text{int}}$ が TA 中の各状態のクロック制約を満たすかどうかは一意に定まる．TA 上での到達可能性解析を行い，Q 中の到達可能でない状態は除去するものとする．

FSM/T のタイマの集合を以下のように定める．

$$T1 = \{\tau[j,\ k] \mid 0 \le j < m,\ 1 \le k \le jMAX\}$$

$$T2 = \{\tau[j] \mid 0 \le j < m\}$$

$T1$ 中のタイマ $\tau[j,k]$ のタイムアウトは，cl_j が区間 $\text{int}(j,k)$ へ進んだことを表す．$T2$ 中のタイマ $\tau[j]$ のタイムアウトは，cl_j が次の区間へ進んだことを表す．$T1$ では，タイムアウト時間は各タイマ毎に固定されているが，$T2$ ではタイマを始動するたびにタイムアウト時間を再設定する必要がある．

TA において cl_j をリセットする時のタイマの操作は，

第 1 変換法　区間 $\text{int}(j,0)$ の状態への遷移時に $\tau[j,k]$ の全てのタイマを始動．（**Fig. 3**）

第 2 変換法　区間 $\text{int}(j,0)$ の状態への遷移時に $\tau[j]$ を再始動．タイムアウト時間は $t_{j,1}$．（**Fig. 4**）

である．また，第 2 変換法においては，タイムアウト遷移を実行した時も $\tau[j]$ を再始動し，タイムアウト時間 $t_{j,k} - t_{j,k-1}$ を再設定する．

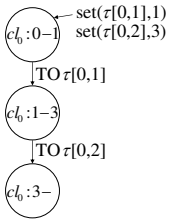

Fig. 3 Timeout transition of $F1$

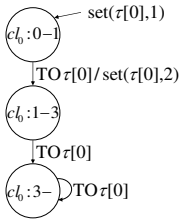

Fig. 4 Timeout transition of $F2$

TA の状態遷移に基づいて $E1$，$E2$ の状態遷移を定める．さらに，TA は初期状態で全てのクロックをリセットしている．FSM/T では，動作開始時には全てのタイマが停止しているので，q_0 から全てのクロックをリセットする操作に対応するタイマの操作を実行し，$(s_0, (\text{int}(0,0), \cdots, \text{int}(m-1,0)))$ へ遷移できるようにする．■

変換前の TA A と変換後の FSM/T $F1(A)$，$F2(A)$ について以下の性質が成り立つ．

定理　TA A において動作列 ω が実行可能である時，かつその時のみ FSM/T $F1(A)$，$F2(A)$ において動作列 ω が実行可能である．■

以上の変換法で得られた FSM/T は一般には非常に冗長なものとなるが，入力によって区別することができない状態の集合を統合する状態数最小化アルゴリズムを用いて簡単化することができる．

5　変換法の比較

タイマに関しては，第 1 変換法では，タイマ数は多くなるが，各タイマ毎にタイムアウト時間を固定することができる．ただし，各クロックの区間が少ない場合，タイマ数の増大はそれほど大きなものとはならない．一方，第 2 変換法では，タイマ数は少なくなるが，タイマを始動するたびにタイムアウト時間を再設定する必要がある．

状態数最小化アルゴリズムに関しては，第 1 変換法は完全定義（全ての状態と入力に対して，遷移先の状態が定義されている）でないので，総当たり的に最小化できる状態を求めなければならない．一方，第 2 変換法は完全定義であるので，状態数最小化が簡単である．

そこで，2 つの変換法を比較する為に，変換法と状態数最小化アルゴリズムに基づく変換システムを構築し，状態数 4，クロック数 2 の TA および状態数 3，クロック数 3 の TA を例にして FSM/T に変換した．その結果を **Table 1** および **Table 2** に示す．

FSM/T の状態数，状態遷移数，タイマ数が少ない方が実装に適していると考えられる．**Table 1** では状態数，状態遷移数はさほど変わらないが，タイマ数は第 2 変換法の方が少ないので，この場合は第 2 変換法を適用するのが妥当である．また，**Table 2** では状態数，状態遷移数，タイマ数共に第 2 変換法の方が少ないので，第 2 変換法を適用するのが妥当である．

Table 1　Evaluation of 4 states and 2 clocks TA

	Not minimized			Minimized		
Translator	$\|Q\|$	$\|E\|$	$\|T\|$	$\|Q\|$	$\|E\|$	$\|T\|$
First	48	90	10	18	55	10
Second	48	90	2	21	60	2

Table 2　Evaluation of 3 states and 3 clocks TA

	Not minimized			Minimized		
Translator	$\|Q\|$	$\|E\|$	$\|T\|$	$\|Q\|$	$\|E\|$	$\|T\|$
First	30	174	7	19	133	7
Second	30	124	3	15	91	3

6　まとめ

本研究では，TA を FSM/T に変換する変換法を 2 つ提案し，それぞれの変換法についての正当性を証明した．また，2 つの変換法の比較を行う為に変換システムを構築し，得られた仕様の状態数，状態遷移数，タイマ数の評価を行った．その結果，タイマを始動するたびにタイムアウト時間を再設定できる場合は第 2 変換法を，できない場合は第 1 変換法を適用するのが妥当であると考えられる．

参考文献

[1] Edmund M. Clarke, Jr., et. al, “Model Checking”, The MIT Press, pp. 265-292(1999)

[2] 森 他, “タイマシステムコールを用いる DFSM プロトコルに対する試験系列生成手法”, 情報処理学会論文誌, Vol. 42, No. 12, pp. 3072-3081(2001)