A_014

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

On Translating Timed Automata into FSM with Timer

岡田 嶺 * 樋口 昌宏 *
Ryo Okada* Masahiro Higuchi*

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（実時間）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を利用して時間に依存した動作を行う．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 から FSM/T への変換法について述べる．

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下限と上限を指定）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 は 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時いずれか 1 つの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 つ以上のイベントの実行が交互に現れる系列として表すことができる．図 1 に TA の例を示す．

*近畿大学大学院総合理工学研究科
Graduated School of Science and Technology, Kinki University

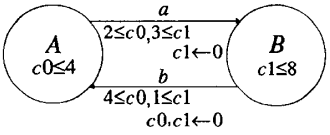

図 1 TA の例
Fig.1 Timed Automata

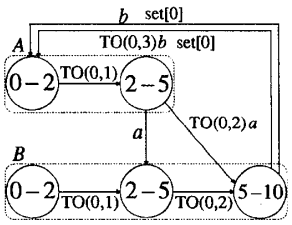

図 2 FSM/T の例
Fig.2 FSM/T

3. タイマ付き有限状態機械

FSM/T は FSM（有限状態機械）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．各タイマにはタイムアウト時間が定まっており，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 は 4 字組 $F = (Q, \Sigma, T, E)$ で定義する．Q は FSM の状態の有限集合，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 $(q, q', \sigma, \bar{p})$ 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
- $\bar{p} \in \{\mathtt{S}, \mathtt{D}, \mathtt{N}\}^{|T|}$ ：タイマ操作ベクトル

タイマ操作ベクトルは各タイマに対する操作を指定し，S は始動，D は解除，N は無操作を表す．

FSM/T の動作列は，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が交互に現れる系列として表すことができる．図 2 に FSM/T の例を示す．

4. 変換法

TA $A = (S, s_0, \Sigma, X, I, R)$ を FSM/T $F = (Q, \Sigma, T, E)$ に変換する方法を述べる．

TA のクロック数を m とする．各クロック c_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 $V_j = (v_j[1], \cdots, v_j[j_{max}])$ とし，

$$\mathrm{int}_j[k] = \begin{cases} 0 \le c_j < v_j[1] & (k = 0) \\ v_j[k] \le c_j < v_j[k+1] & (0 < k < j_{max}) \\ v_j[j_{max}] \le c_j & (k = j_{max}) \end{cases}$$

とする．

$Q = \{(s, \overline{\mathrm{int}} = (\mathrm{int}_0[i_0], \cdots, \mathrm{int}_{m-1}[i_{m-1}])) \mid s \in S, 0 \le i_j < j_{max}\}$ とする．この時，$(s, \overline{\mathrm{int}}) \in Q$ について $\overline{\mathrm{int}}$ が TA 中の各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．

$T_j = \{\tau_j[1], \cdots, \tau_j[j_{max}]\}$ とし，$T = \bigcup T_j$ とする．状態 Q について，次の状態遷移を導入する．

(1) $\overline{\mathrm{int}}$ の第 j 成分 $\mathrm{int}_j[k]$ が $\mathrm{int}_j[j_{max}]$ でない時，

$$((s, \overline{\mathrm{int}}), (s, \overline{\mathrm{int}'}), \mathrm{TO}(j,k), \overline{\mathtt{N}})$$

を E に加える．ただし，$\overline{\mathrm{int}'}$ は $\overline{\mathrm{int}}$ 中の $\mathrm{int}_j[k]$ を $\mathrm{int}_j[k+1]$ に置き換えたものであり，$\overline{\mathtt{N}}$ は全てのタイマが N であるタイマ操作ベクトルである．

(2) $(s, s', \sigma, \varphi, X_R) \in R$ に対して $\overline{\mathrm{int}}$ が φ を満たす時，

$$((s, \overline{\mathrm{int}}), (s', \overline{\mathrm{int}'}), \sigma, \overline{p'})$$

を E に加える．ただし，$X_R = \{c_{x_1}, \cdots, c_{x_m}\}$ とすると，$\overline{\mathrm{int}'}$ は $\overline{\mathrm{int}}$ 中の第 $x_1, \cdots, x_m$ 成分を $\mathrm{int}_{x_1}[0], \cdots, \mathrm{int}_{x_m}[0]$ に置き換えたものであり，$\overline{p'}$ はタイマ $\tau_{x_1}[1], \cdots, \tau_{x_1}[x_{1max}], \cdots, \tau_{x_m}[1], \cdots, \tau_{x_m}[x_{mmax}]$ が S，他は全て N であるタイマ操作ベクトルである．

(3) (1) あるいは (2) において $\overline{\mathrm{int}'}$ が $I(s)$ あるいは $I(s')$ を満たさない場合，TA においていずれか 1 つの状態遷移 $(s, s'', \sigma', \varphi', X'_R)$ あるいは $(s', s'', \sigma', \varphi', X'_R)$ が実行可能である．この場合，(2) と同様にして $\overline{\mathrm{int}''}$ と $\overline{p''}$ を求め，

(1) で加えようとした遷移に変えて

$$((s, \overline{\mathrm{int}}), (s'', \overline{\mathrm{int}''}), \mathrm{TO}(j,k)\sigma', \overline{p''})$$

(2) で加えようとした遷移に変えて

$$((s, \overline{\mathrm{int}}), (s'', \overline{\mathrm{int}''}), \sigma\sigma', \overline{p''})$$

を E に加える．$\overline{\mathrm{int}''}$ が $I(s'')$ を満たさない場合は同様の操作を続ける． □

変換前の TA $A = (S, s_0, \Sigma, X, I, R)$，変換後の FSM/T $F = (Q, \Sigma, T, E)$ 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 $\tau_0 e_0 \tau_1 e_1 \cdots \tau_n$（$\tau_i$ は時間経過，e_i は 1 つ以上のイベント列）が実行可能である時，かつその時のみ FSM/T F で $\tau'_0 e'_0 \tau'_1 e'_1 \cdots \tau'_n$ が実行可能である．ただし，τ'_i は動作列 $\tau'_{i(0)}\mathrm{TO}_0\tau'_{i(1)}\mathrm{TO}_1 \cdots \mathrm{TO}_{x-1}\tau'_{i(x)}$（$\mathrm{TO}_h$ はタイムアウト信号，$\tau'_{i(0)} + \tau'_{i(1)} + \cdots + \tau'_{i(x)} = \tau_i$）であり，$e'_i$ は e_i の前に 0 個または 1 つのタイムアウト信号を付加したものである． □

以上の変換法で得られた FSM は一般には非常に冗長なものとなるが，決定性有限オートマトンの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. 変換法を用いた変換結果

図 1 の TA に上記の変換法を適用して FSM/T に変換した結果を図 3 に示す．図中の点線で囲まれた状態はそれぞれ TA の状態 A と B に対応する FSM/T の状態を表し，A の左上の状態のラベル "0 – 2"，"0 – 1" は int = ($0 \le c_0 < 2$, $0 \le c_1 < 1$) であることを表す．TO(j,k) はタイマ $\tau_j[k]$ のタイムアウト，set[X'] はクロック X' に対応するタイマの始動を表す．また，図 3 の FSM/T に状態数最小化アルゴリズムを適用して得られた FSM/T を図 4 に示す．

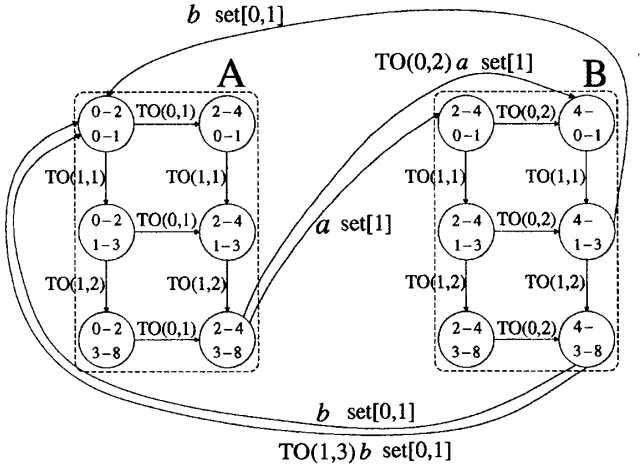

図 3 変換法を適用して得られた FSM/T
Fig.3 Translated FSM/T

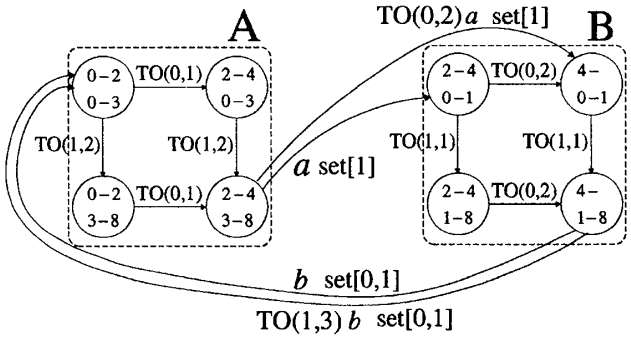

図 4 状態数最小化アルゴリズムを適用して得られた FSM/T
Fig.4 Minimized FSM/T

参考文献

[1] Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled : "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)