SCIENCE CHINA
Information Sciences

• LETTER •

November 2022, Vol. 65 219201:1–219201:2
https://doi.org/10.1007/s11432-020-3161-6

Further results on bilinear behavior formulation of finite state machines

Jumei YUE[1], Yongyi YAN[2*], Zengqiang CHEN[3] & He DENG[2]

[1]*College of Agricultural Engineering, Henan University of Science and Technology, Luoyang* 471000*, China;*
[2]*College of Information Engineering, Henan University of Science and Technology, Luoyang* 471000*, China;*
[3]*College of Artificial Intelligence, Nankai University, Tianjin* 300071*, China*

Received 26 August 2020/Revised 31 October 2020/Accepted 31 December 2020/Published online 13 May 2021

Citation Yue J M, Yan Y Y, Chen Z Q, et al. Further results on bilinear behavior formulation of finite state machines. Sci China Inf Sci, 2022, 65(11): 219201, https://doi.org/10.1007/s11432-020-3161-6

Dear editor,

Linear and nonlinear equations play an important role in modeling control systems governed by the laws of nature. However, they are inadequate for logical dynamic systems such as discrete event systems. A finite state machine (FSM) is a system in which the state evolution is governed by operational rules designed by humans. In the early 1960s, descriptions of the dynamics of FSMs were based primarily on the look-up table method, state transition diagram, and discrete function, which are intuitive and easy to understand. Nevertheless, they are incontinent to analyze and synthesize FSMs mathematically. Recently, the problem of state evolution of FSMs has attracted the attention of scholars in the control field, and some new results have been achieved based on a new mathematical tool called STP of matrices (semi-tensor product of matrices) proposed by Cheng and colleagues [1]. These results can be roughly classified into two methods. In the first, deterministic FSMs and nondeterministic FSMs are modeled separately [2–5], that is, the model is not uniformly applicable to both deterministic and nondeterministic FSMs. In the second, both types of FSMs are modeled uniformly [6–9]. These two methods share the same forms and results of the model, but differ in ideas: the latter defines the state as a vector where the i-th element is the number of different paths from the initial state to the i-th state; the former directly uses the original vector form of a state. Therefore, these two models have different physical meanings. Another difference between them lies in the ability to describe dynamics. The latter describes dynamics in the future 1-time-step, while the former formulates dynamics in the future t-time-step. In short, the latter excels in the uniform of the model, while the former has the advantage of describing the dynamics. This motivates further reconsideration of the problem with the aim of combining the individual advantages of the two models into a single model. Therefore, this study proposes the establishment of a model that not only is uniformly suitable for both deterministic and nondeterministic FSMs but also can adequately describe the dynamics.

Dynamic formulation of FSMs. Consider the FSM $M = (X, \Sigma, f, x_0, X_F)$, where $X = \{x_1, x_2, \ldots, x_n\}$, $\Sigma = \{a_1, a_2, \ldots, a_m\}$. Identify $x_i \in X$ with δ_n^i and call δ_n^i the vector form of x_i, $1 \leqslant i \leqslant n$. Then X can be defined as $X = \{\delta_n^1, \delta_n^2, \ldots, \delta_n^n\}$. Similarly, identifying input symbol $a_j \in \Sigma$ with δ_m^j, δ_m^j is called the vector form of a_j, $1 \leqslant j \leqslant m$, and Σ can be represented by $\Sigma = \{\delta_m^1, \delta_m^2, \ldots, \delta_m^m\}$. Thus the state transition function $f(x_i, a_j) = x_k$ can be rewritten as the form of $f(\delta_n^i, \delta_m^j) = \delta_n^k$. These two types of representations are used as synonyms in this study.

Definition 1 (State packet and reachability). A state packet of the FSM M is a set of states. A t-step reachable state packet of input sequence $u(t)$ from state x_i, defined as $R(x_i, u(t))$, is a state packet in which every state can be reached by $u(t)$ when M reads $u(t)$ at state x_i.

Let a t-step reachable state packet of $u(t)$ from state x_i be $R(x_i, u(t)) = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$, then $f(x_i, u(t)) = x_{i_1}, f(x_i, u(t)) = x_{i_2}, \ldots, f(x_i, u(t)) = x_{i_t}$. The vector form of state packet $R(x_i, u(t)) = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$ is defined as $\delta_n^{i_1, i_2, \ldots, i_k}$.

Definition 2 (Structure matrix of FSMs). The matrix $F = [F_1, \ldots, F_k, \ldots, F_n]$ is called the structure matrix of the FSM M, where F_k $(1 \leqslant k \leqslant n)$ is the k-th block of F and defined as that the element $F_k(i, j)$ takes 1, if $x_i \in R(x_k, a_j)$, takes 0, otherwise.

The physical meaning of the structure matrix of FSM is that if the 1-step reachable state packet of input symbol a_j from state x_k is $R(x_k, a_j) = \{x_{i_1}, x_{i_2}, \ldots, x_{i_l}\}$ if and only if $\mathrm{Col}_j(F_k) = [0 \cdots 0 \;\; 1 \;\; 0 \cdots 0 \;\; 1 \;\; 0 \cdots 0 \;\; 1 \;\; 0 \cdots 0]^{\mathrm{T}}_{1 \times n}$, where the 1s are at the positions of i_1, i_2 and i_l. For a deterministic FSM, $|R(x_k, a_j)| = 1$, the structure matrix is therefore a logical matrix.

Definition 3 (Column-equal-division of a matrix). Let A be a matrix of dimension $n \times m$, where $m = rp$, the r-column-equal-division of A is $A = [\mathrm{Blk}_1^r(A), \ldots, \mathrm{Blk}_i^r(A), \ldots, \mathrm{Blk}_r^r(A)]$, where $\mathrm{Blk}_i^r(A) = [\mathrm{col}_{(i-1)p+1}(A), \ldots, \mathrm{col}_{ip}(A)]$ is called the i-th block of the r-column-equal-division of A.

Theorem 1 (Unified dynamical model of FSMs). The state evolution of FSM M reading a sequence $u(t) =$

* Corresponding author (email: yyyan@mail.nankai.edu.cn)

$a_{i_1}a_{i_2}\cdots a_{i_t}$ at state x_i can be formulated as $x(t+1) = F^t \ltimes x_i \ltimes u(t)$, where $\ltimes$ denotes the STP of matrices (see Appendix A for further details); $x(t+1)$ is the state packet at $x(t+1)$-time step. See Appendix B.1 for the proof.

Remark 1. The state packet contains the complete evolutions from an initial state x_i. Suppose the state packet is $[0,\ldots,0,n_{j_1},0,\ldots,0,n_{j_2},0,\ldots,0,n_{j_r},0,\ldots,0]^{\mathrm{T}}$, then there are n_{j_l} different paths of length t from state x_i to state x_{j_l}, n_{j_2} different paths from state x_i to state x_{j_l}, and so on.

Definition 4 (T-step structure matrix of FSMs)**.** The matrix F^t in Theorem 1 is called the t-step structure matrix of M.

Controllability condition of FSMs. The t-step structure matrix of FSMs facilitates the investigation of some important problems, such as conditions of controllability and reachability of FSMs.

Definition 5 (Controllability of FSMs)**.** Let x_i and x_j be two states of FSM $M = (X,\Sigma,f,x_0,X_F)$. State x_j is called a t-controllable state from state x_i if there is an input string $u(t) = a_{i_1}a_{i_2}\cdots a_{i_t}$ such that M moves to x_j from x_i by $u(t)$. $u(t)$ is called a t-control input.

Theorem 2 (Controllability condition)**.** Consider FSM $M = (X,\Sigma,f,x_0,X_F)$, where $X = \{x_1,x_2,\ldots,x_n\}$, $\Sigma = \{a_1,a_2,\ldots,a_m\}$. State x_j is a t-controllable state from state x_i if and only if $\sum_{k=1}^{m^t}\mathrm{Blk}_i^n(F^t)(j,k) \neq 0$. See Appendix B.2 for the proof.

Algorithm 1 (Searching controllable sequences). Let F be the structure matrix of FSM M. The following procedure produces all the controllable sequences moving M to a target state x_j from a state x_i.

Step 1. Compute the t-step structure matrix of M, F^t.

Step 2. Divide F^t into the form of n-column-equal-division: $F^t = [\mathrm{Blk}_1^n(F^t)\cdots\mathrm{Blk}_i^n(F^t)\cdots\mathrm{Blk}_n^n(F^t)]$.

Step 3. Judge whether there is a non-zero element in $\mathrm{Row}_j\left(\mathrm{Blk}_i^n(F^t)\right)$ or not. If not, there is no input sequence that moves M to x_j from x_i. If yes, construct the set $K = \{k|$ the k-th element of $\mathrm{Row}_j\left(\mathrm{Blk}_i^n(F^t)\right)$ is non-zero$\}$.

Step 4. For each k in K, find $a_i (i = 1,2,\ldots,t)$ by applying Appendix B.4 to $\ltimes_{i=1}^t a_i = \delta_{m^t}^{k_s}$. The resulting a_i $(i = 1,2,\ldots,t)$ construct a sequence $u(t) = a_1a_2\cdots a_t$. The $u(t)$ is an input string moving M to the target state x_j from state x_i.

Step 5. Repeat Step 4 until all the elements in K are executed and all the expected controllable inputs are produced.

Reachability condition between states of an FSM. Using the t-step structure matrix of an FSM, the investigation of the state reachability is also facilitated.

Definition 6 (T-reachability between states)**.** Let $X = \{x_1,x_2,\ldots,x_n\}$ be the state set of an FSM. If there exists a path of length t from state x_i to state x_j, state x_j is said to be t-reachable from state x_i. Otherwise, state x_j is t-unreachable from state x_i.

Remark 2. (i) If state x_j is a t-controllable state from state x_i, then state x_j is t-reachable from state x_i; (ii) if state x_j is t-reachable from state x_i, then state x_j is a t-controllable state from state x_i. The differences between them are as follows. (i) if the t-reachability degree of state x_j from state x_i is greater than one, it does not mean that the t-controllability degree of state x_j from state x_i is also greater than one, in fact, it may exactly be one; (ii) if the t-controllability degree of state x_j from state x_i is one, this does not imply that the t-reachability degree of state x_j from state x_i is also one; in fact it can be greater than one.

Theorem 3 (Reachability condition)**.** Consider FSM $M = (X,\Sigma,f,x_0,X_F)$, where $X = \{x_1,\ldots,x_n\}$, $\Sigma = \{a_1,\ldots,a_m\}$. The t-step reachable state packet of input sequence $u(t)$ from state x_i is $R(x_i,u(t)) = \{x_{i_1},x_{i_2},\ldots,x_{i_k}\}$ if and only if $\prod_{s\in\{i_1,i_2,\ldots,i_k\}}\mathrm{Blk}_i^n(F^t)(s,l) \neq 0$, where $u(t) = a_{i_1}a_{i_2}\cdots a_{i_t} = \delta_{m^t}^l$. See Appendix B.5 for the proof.

Algorithm 2 (Searching t-step reachable state packets). Let F be the structure matrix of M. Consider a state $x_i \in X$ and an input sequence $u(t) = a_{i_1}a_{i_2}\cdots a_{i_t}$ whose vector form is $\delta_{m^t}^l$. The following steps procedure all the t-step reachable state packet of $u(t)$ from state x_i.

Step 1. Calculate F^t and split it into n-column-equal-division by Definition 3, defined as $F^t = [\mathrm{Blk}_1^n(F^t)\cdots\mathrm{Blk}_i^n(F^t)\cdots\mathrm{Blk}_n^n(F^t)]$.

Step 2. Judge whether $\mathrm{Col}_l(\mathrm{Blk}_i^n(F^t))$ has non-zero element or not, if not, the reachable state packet is $R(x_i,u(t)) = \emptyset$ and the algorithm terminates. Otherwise, construct the set $S = \{h|$ the h-th element of $\mathrm{Col}_l(\mathrm{Blk}_i^n(F^t))$ is non-zero$\}$.

Step 3. The t-step reachable state packet of $u(t)$ from state x_i is $R(x_i,u(t)) = \{x_h|h \in S\}$.

Remark 3. (i) The presented controllability and reachability conditions of FSMs show that the proposed dynamic model of FSMs indeed integrates both advantages of the two existing models, that is, the proposed model not only is applicable to both deterministic and nondeterministic FSMs but also adequately formulates the state evolution of the two FSMs. (ii) Algorithms 1 and 2 have the time complexity $O(n)$ (See Appendix B.6 for the analysis).

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. U1804150, 62073124).

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Cheng D, Qi H, Zhao Y. An Introduction to Semi-tensor Product of Matrices and Its Applications. Singapore: World Scientific Publishing Co. Pte. Ltd., 2012. 23–26
2 Yan Y, Chen Z, Liu Z. Semi-tensor product approach to controllability and stabilizability of finite automata. J Syst Eng Electron, 2015, 26: 134–141
3 Yan Y, Yue J, Fu Z, et al. Algebraic criteria for finite automata understanding of regular language. Front Comput Sci, 2019, 13: 1148–1150
4 Han X, Chen Z, Liu Z, et al. The detection and stabilisation of limit cycle for deterministic finite automata. Int J Control, 2018, 91: 874–886
5 Zhang Z, Xia C, Chen S, et al. Reachability analysis of networked finite state machine with communication losses: a switched perspective. IEEE J Sel Areas Commun, 2020, 38: 845–853
6 Xu X, Hong Y. Matrix expression and reachability analysis of finite automata. J Control Theor Appl, 2012, 10: 210–215
7 Xu X R, Hong Y G. Observability analysis and observer design for finite automata via matrix approach. IET Control Theor Appl, 2013, 7: 1609–1615
8 Wang B, Feng J. A matrix approach for the static correction problem of asynchronous sequential machines. Int J Control Autom Syst, 2020, 18: 477–485
9 Dou W, Li H, Alsaadi F. Semitensor product approach to controllability, reachability, and stabilizability of probabilistic finite automata. Math Probl Eng, 2019, 2019: 8021750