SCIENCE CHINA
Information Sciences

• REVIEW •

January 2023, Vol. 66 111201:1–111201:20
https://doi.org/10.1007/s11432-022-3538-4

Survey on applications of algebraic state space theory of logical systems to finite state machines

Yongyi YAN[1*], Daizhan CHENG[2], Jun-E FENG[3], Haitao LI[4] & Jumei YUE[5]

[1]*College of Information Engineering, Henan University of Science and Technology, Luoyang* 471023*, China;*
[2]*Institute of Systems Science, Chinese Academy of Sciences, Beijing* 100190*, China;*
[3]*School of Mathematics, Shandong University, Jinan* 250100*, China;*
[4]*School of Mathematics and Statistics, Shandong Normal University, Jinan* 250014*, China;*
[5]*College of Agricultural Equipment Engineering, Henan University of Science and Technology, Luoyang* 471023*, China*

Received 17 January 2022/Revised 27 April 2022/Accepted 16 June 2022/Published online 23 December 2022

Abstract Algebraic state space theory (ASST) of logical systems, developed based on the semi-tensor product (STP) which is a new matrix analysis tool built in recent ten years, provides an algebraic analysis approach for many fields of science, such as logical dynamical systems, finite-valued systems, discrete event dynamic systems, and networked game systems. This study focuses on comprehensively surveying the applications of the ASST method to the field of finite state machines (FSMs). Some necessary preliminaries on the method are first reviewed. Then the applications of the method in the FSM field are reviewed, including deterministic FSMs, nondeterministic FSMs, probabilistic FSMs, networked FSMs, and controlled and combined FSMs. In addition, other applications related to both STP and FSMs are surveyed, such as the application of FSM to Boolean control networks and the application of graph theory to FSMs. Finally, some potential research directions with respect to the ASST method in the FSM field are predicted.

Keywords Boolean control network, Boolean network, finite state machine, learning system, semi-tensor product (STP), ASST

Citation Yan Y Y, Cheng D Z, Feng J-E, et al. Survey on applications of algebraic state space theory of logical systems to finite state machines. Sci China Inf Sci, 2023, 66(1): 111201, https://doi.org/10.1007/s11432-022-3538-4

1 Introduction

As Deif remarked in the 1990s [1], matrix theory can be called advanced arithmetic and has wonderful applications in almost every field of science and engineering. This is because dealing with complex systems with many components must require a mathematical tool that can combine these components together, and matrix theory can achieve this goal. However, matrix theory is not a panacea; it has its inherent defects. A vivid example can be found in [2].

In 2009, a new type of matrix multiplication, called the semi-tensor product (STP) of matrices, was proposed by Cheng [3]. For two matrices of any dimensions $M \in \mathbb{M}_{m\times n}$ and $N \in \mathbb{M}_{p\times q}$, the STP of M and N is defined as

$$M \ltimes N := (M \otimes I_{s/n})(N \otimes I_{s/p}),$$

where I_k represents the identity matrix of order k, s is the least common multiple of n and p, and $\otimes$ is the Kronecker product of matrices. The STP generalizes the traditional matrix multiplication (TMM) by allowing any two matrices to participate the multiplication. In the framework of the STP, some defects of the TMM are naturally and perfectly overcome, and an overwhelming majority of the properties of the TMM still hold. More importantly, the STP possesses special properties that the TMM does not have, such as the pseudo commutative law with respect to vectors, vector-matrix version of pseudo commutative law, elimination law of vectors, algebraic representation of logical functions, extraction law, and power-reducing law. These beautiful properties have made STP attract many scholars' research interests in

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

Table 1 Notations used in this study

Notation	Meaning	Notation	Meaning
$\mathbb{M}_{m\times n}$	The set of $m \times n$ matrices	$\mathbb{Z}_+$	The set of positive integers
$\mathbb{L}_{m\times n}$	The set of logical matrices of size $m \times n$	D	The set $\{0,1\}$
δ_n^i	The i-th column of the identity matrix I_n	Δ_n	The set $\{\delta_n^i \mid i=1,2,\ldots,n\}$
$M=\delta_n[i_1,i_2,\ldots,i_k]$	$M=[\delta_n^{i_1},\delta_n^{i_2},\ldots,\delta_n^{i_k}]$	$\lfloor A \rfloor$	The matrix $A=(\lfloor a_{ij}\rfloor)\in\mathbb{M}_{m\times n}$
$\lvert S\rvert$	The cardinality of set S	S_1-S_2	The set $\{s \mid s\in S_1, s\notin S_2\}$
$\lfloor x\rfloor$	The largest integer less than or equal to x	$\Theta(x)$	The set $\{y\in\Delta_m \mid y\wedge x=y\}$
$\mathrm{Row}_k(A)$	The k-th row of matrix A	$\mathrm{Row}(A)$	The set $\{\mathrm{Row}_k(A) \mid k=1,\ldots,m\}$
$\mathrm{Col}_k(A)$	The k-th column of matrix A	$\mathrm{Col}(A)$	The set $\{\mathrm{Col}_k(A) \mid k=1,\ldots,n\}$
$\mathbf{1}_n$	The n dimensional vector $[1,1,\ldots,1]$	$\Delta(\eta)$	The set $\{\delta_n^k \mid$ the k-th element of η is nonzero$\}$
E_d	The dummy operator matrix $\delta_2[1,2,1,2]$	$\Omega(M)$	$:=\{\alpha \mid \alpha\in\Delta(\mathrm{col}_i(M)), i=1,2,\ldots,n\}$

various fields, such as automation control systems [4–7], engineering electrical electronics [8–10], artificial intelligence [11–13], information systems [14–16], neurosciences [17,18], and applied mathematics [19–21].

It is worth mentioning that in the framework of algebraic state space theory (ASST) developed based on the STP, finite state machines (FSMs) have been investigated systemically by borrowing the ideas, methods, and techniques from control theory. A series of achievements have been gained, including different modes of FSMs and various problems of the same model. The models of FSMs include deterministic FSMs [22–34], nondeterministic FSMs [35–43], probabilistic FSMs [44–48], and networked FSMs [49,50]. Different problems of the same model consist of stabilization [22–25,31,35,38], simulation [40], simplification [37,51], reachability [28,38,41,46,49,52,53], controllability [31,38,46,53], language recognition [27,28,32,34], observability [39,54], etc.

In the last decade, FSMs have integrated with control theory in many aspects with the help of ASST and STP, such as the ideas (feedback ideas) [23,25,35,55], concepts (reachability, controllability, observability, stabilization) [28,31,38,39,41,46,49,52–54,56], and methods (state feedback control, correction control, model predictive control, fault-tolerant control, adaptive control) [29–31,46,57–59]. These results greatly enrich the control theory of FSMs. This paper aims to present a comprehensive survey of the applications of the ASST method in FSMs.

The remainder of this paper is arranged as follows. Section 2 introduces some commonly used theoretical results on the STP and some concepts on FSMs. Sections 3–7 present applications of the ASST method in various types of FSMs: deterministic FSMs in Section 3, nondeterministic FSMs in Section 4, probabilistic FSMs in Section 5, networked FSMs in Section 6, and controlled and combined FSMs in Section 7. Section 8 is a brief review of other applications related to STP and FSMs in the context of ASST. And finally, several potential research directions are predicted in Section 9. The notations used in this paper are listed in Table 1.

2 STP and finite state machines

This section first briefly looks back at some special properties of STP, which are powerful in studying FSMs and other fields by using ASST and are not possessed by classical matrix multiplication. Then the concepts of various types of FSMs discussed in this paper are briefly introduced.

2.1 Special properties of STP

Remark 1. Since the traditional multiplication of matrices is a special case of the STP, this study uses the STP as the multiplication of matrices, and the STP symbol $\ltimes$ is omitted except for special emphasis.

Proposition 1 (Associative law [60]). Let $A\in\mathbb{M}_{m\times n}$, $B\in\mathbb{M}_{p\times q}$ and $C\in\mathbb{M}_{r\times s}$ be three matrices, and then the associative law $(AB)\,C=A\,(BC)$ holds.

Proposition 2 (Scalar multiplication [60]). Let A be a matrix of size $m\times n$ and r be a real number. Then $rA=Ar=rA$.

Proposition 3 (Pseudo commutative law with respect to vectors [60]). Let X and Y be two column vectors of m and n dimensions, respectively. STP has the pseudo commutative property $W_{[m,n]}XY=YX$ and $W_{[m,n]}YX=XY$, where $W_{[m,n]}$ is an $mn\times mn$ matrix and is called a swap matrix, whose element at position $[(I,J),(i,j)]$ is defined as 1 if $I=i$ and $J=j$, or as 0, otherwise, where (I,J) and (i,j) are double indices used to mark the rows and columns, which are organized by the ordered multiple indices $\mathrm{Id}[j,i;n,m]$ and $\mathrm{Id}[i,j;m,n]$, respectively.

Proposition 4 (Vector-matrix version of pseudo commutative law [61])**.** Let X be a column vector of size $s\times 1$ and $A\in\mathbb{M}_{m\times n}$ be a matrix. Then the pseudo commutative law $XA=(I_s\otimes A)X$ holds.

Proposition 5 (Elimination law of vectors [31])**.** For all pairs of logical variables $x,y\in\Delta$ and $E_d=[1,2,1,2]$, STP has the elimination properties $E_dxy=y$ and $E_dW_{[2,2]}xy=x$.

Proposition 6 (Algebraic representation of logical function [5])**.** Any logic function $f(x_1,\ldots,x_n):\Delta^n\to\Delta$ can be represented as the algebraic form $f(x_1,\ldots,x_n)=M_fx_1x_2\cdots x_n$, where M_f is unique and called the structure matrix of f, $x_1,\ldots,x_n\in\Delta$.

Definition 1 (Column-equal-division of matrix [51])**.** For a matrix A of size $n\times m$, $m=rp$, its r-column-equal-division is $A=[\mathrm{Blk}_1^r(A),\ldots,\mathrm{Blk}_i^r(A),\ldots,\mathrm{Blk}_r^r(A)]$, where $\mathrm{Blk}_i^r(A)=\left[\mathrm{col}_{(i-1)p+1}(A),\ldots,\mathrm{col}_{ip}(A)\right]$ is referred to as the i-th block of r-column-equal-division of A.

Proposition 7 (Extraction law [51])**.** Let A be a matrix of size $p\times qr$. The extraction law $A\delta_r^i=\mathrm{Blk}_i^r(A)$ is true for STP, where $\mathrm{Blk}_i^r(A)$ is given in Definition 1.

Proposition 8 (Power-reducing law [60])**.** Let x be a logical vector of size $n\times 1$. Then $M_{pr}x=x^2$, in which $\mathrm{Col}_i(M_{pr})=\delta_{n^2}^{(n+1)i-n}, i=1,2,\ldots,n$.

2.2 Finite state machines

Finite state machines, also called finite automata or finite-state automata, are devices that can represent languages according to defined rules, or mathematical models of computation.

Definition 2 (FSM)**.** An FSM is formally defined as $M=(X,E,f,x_0,X_F)$, in which X and E are sets of states and events, respectively, f is state transition function, x_0 is the initial state, and X_F is the set of accepting states. An accepting state means that an event sequence can be accepted by M if the event sequence moves M to the accepting state from the initial state x_0.

Figure 1(a) is an illustrative example, where the initial state, state set, event set and accepting state set are $x_0=x_1$, $X=\{x_1,x_2,\ldots,x_9\}$, $E=\{e_1,e_2,e_3,v_1,v_2,v_3\}$, and $X_F=\{x_6\}$, respectively.

Definition 3 (Deterministic FSM)**.** An FSM is called a deterministic one if the state transition function is defined from $X\times E$ to X; i.e., there is only one transition with the same event from a state. $f(x_i,e)=x_j$ means a transition driven by event e from state x_i to state x_j. Figure 1(a) is a deterministic FSM (DFSM).

Definition 4 (Nondeterministic FSM)**.** In Definition 2, if f is defined from $X\times E$ to 2^X, where 2^X is the power set of X, this means that an event can cause multiple transitions from a state. In this case, the FSM is known as a nondeterministic FSM (NFSM).

The FSM M runs as follows. M begins from x_0; when e_i occurs, M transfers to $f(x_0,e_i)$. The process continues according to how the state transition function f is defined. The dynamics of M excited by the event $e=e_1e_2\cdots e_t\in E^*$ is described by $f(e,x)=f(f((\cdots f(x,e_1),e_2)\cdots),e_t)$, where E^* represents the set of all finite event sequences on E. For more detailed description of FSMs, please refer to [62–65].

Remark 2. If we further consider outputs, an FSM can also be defined as $M=(X,E,Y,f,g,x_0,X_F)$, where X denotes a set of states, E is a set of letters or symbols, and Y is a set of output symbols. g is the output function defined as $g:X\times E\to 2^Y$, where 2^Y stands for the power set of Y. If the output function is limited to $g:X\times E\to Y$, the FSM is then called a deterministic FSM with outputs.

In the evolution of deterministic FSMs and nondeterministic FSMs, an event either must happen or must not happen. If we equip the occurrence of events with probabilities, we then get probabilistic FSMs, which are defined as follows.

Definition 5 (Probabilistic FSM)**.** A probabilistic FSM is captured by a six-tuple $M_p=(X,E,P,\Gamma,f,x_0)$, where X,E,f and x_0 have the same meanings as in the deterministic FSMs and nondeterministic FSMs, with two differences that P and Γ. P is a state transition probability function from the Cartesian product $X\times E\times X$ to the probability interval $[0,1]$, defined for $x_i,x_j\in X$ such that $T:=\sum_{x_j\in X}P(x_i,\sigma_k,x_j)=1$ or $T=0$, where $\sigma_k\in E$ is the conditional probability with the occurrence of event σ_k. $T=1$ implies that the same event can drive multiple transitions from a state, while $T=0$ means $P(x_i,\sigma_k,x_j)=0$ for any $\sigma_k\in E$ and such events can be omitted in probabilistic FSMs. P can also be the extension probability function over $s\in E^*$ by $P_{ij}^{s\sigma}=\sum P_{ij}^sP_{jl}^\sigma$. Γ is the feasible event function from X to E defined for $x_i,x_j\in X$, $\sigma_k\in E$ as $\Gamma(x_i)=\{\sigma_k|P_{ij}^k>0\}$, in which $P_{ij}^k=P(x_i,\sigma_k,x_j)$ is the probability distribution. More details on probabilistic FSMs are given in [44,45,66–69].

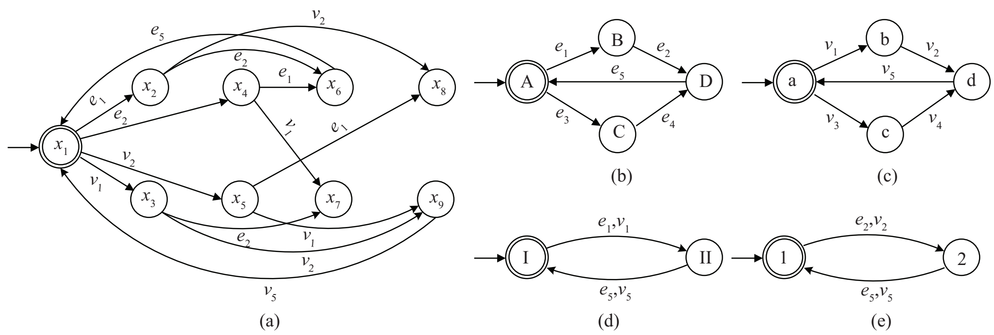

Figure 1 A networked FSM and the component FSMs. (a) A network of FSMs created by composing M_1, M_2, M_3, M_4; (b) component FSM M_1; (c) component FSM M_2; (d) component FSM M_3; (e) component FSM M_4.

Definition 6 (Networked FSM)**.** A networked FSM is a set of FSMs which interact through an operation of composition, called parallel composition, which is defined as follows. Consider two FSMs $M_1 = (X_1, E_1, f_1, x_{01}, X_{m1})$ and $M_2 = (X_2, E_2, f_2, x_{02}, X_{m2})$. The parallel composition of M_1 and M_2 is the FSM $M_1 \| M_2 = (X_1 \times X_2, E_1 \cup E_2, f, (x_{01}, x_{02}), X_{m1} \times X_{m2})$, where

$$f\left(\left(x_1, x_2\right), e\right) = \begin{cases} \left(f_1\left(x_1, e\right), f_2\left(x_2, e\right)\right), & \text{if both } f_1\left(x_1, e\right) \text{ and} f_2\left(x_2, e\right) \text{ are defined,} \\ \left(f_1\left(x_1, e\right), x_2\right), & \text{if only } f_1\left(x_1, e\right) \text{ is defined,} \\ \left(x_1, f_2\left(x_2, e\right)\right), & \text{if only } f_2\left(x_2, e\right) \text{ is defined,} \\ \text{undefined,} & \text{otherwise.} \end{cases} \tag{1}$$

The parallel composition is associative: $(M_1 \| M_2) \| M_3 = M_1 \| (M_2 \| M_3)$. Therefore the parallel composition of a set of n FSMs can be defined by using associativity as $W = M_1 \parallel M_2 \parallel \cdots \parallel M_n = (\cdots((M_1 \parallel M_2) \parallel M_3) \parallel \cdots M_{n-1}) M_n$. Figure 1 is a networked FSM constructed by four FSMs, M_1, M_2, M_3 and M_4, shown in Figures 1(b)–(e), where the states x_i $(i = 1, 2, \ldots, 9)$ of W are simplified representations of the states of the network for the sake of readability. Taking x_1 for example, it stands for the state $(A, a, I, 1)$. Refs. [70–72] provided more details of the networked FSMs.

Definition 7 (Controlled FSM)**.** A controlled FSM is an FSM endowed with a control specification and is formally defined as a five-tuple $M_c = (\Phi \times E, X, f_c, x_0, X_F)$, where the objects E, X, x_0 and X_F have the same meanings as those of deterministic FSMs, $\Phi = \{0, 1\}^{E_c}$, and E_c is called controlled event set, which is defined as follows. Controlled and uncontrolled event sets are subsets of E such that $E_c \subset E$, $E_u \subset E$, $E_c \cup E_u = E$, and $E_c \cap E_u = \emptyset$. A control specification is a Boolean function $\gamma \in E$ satisfying that for any $\sigma \in \Sigma_c$, $\gamma(\sigma) = 1$ means that control specification γ allows event σ to happen, while $\gamma(\sigma) = 0$ means that γ refuses event σ to happen. f_c is the state transition function of the control specification defined as $f_c(\gamma, \sigma, x) = f(\sigma, x)$ if $f(\sigma, x)!$ and $\gamma(\sigma) = 1$; otherwise it is undefined, where $f(\sigma, x)!$ means $f(\sigma, x)$ is well defined. For more details, please refer to [73, 74].

3 Deterministic finite state machines

In 2012, by introducing ASST into the field of FSMs, Xu et al. [41] proposed an algebraic approach to investigate the FSMs. They established the algebraic dynamic equation (2) to model FSMs.

$$x\left(t+1\right) = Fu\left(t\right)x\left(t\right), \tag{2}$$

where $x(t)$ denotes the state at time t, $u(t)$ is the input sequence from the initial time to time t. F is referred to as a transition structure matrix defined as $F = [F_1, \ldots, F_m]$, in which $F_{i(s,t)} = 0$ if $f(x_t, e_i) = x_s$; $F_{i(s,t)} = 1$, otherwise.

And in 2013, the output dynamic of FSMs was also modeled as the following algebraic equation [39]:

$$y(t) = Hu\left(t\right)x\left(t\right), \tag{3}$$

where $y(t)$ is the output at the time t, H is called the output structure matrix determined by the output dynamic function $g(x, e)$ as $H = [H_1, H_2, \ldots, H_m]$, in which $H_{i(s,t)} = 1$, if $g(x_t, e_i) = x_s$; $H_{i(s,t)} = 0$, otherwise.

The combination of (2) and (3) is referred to as a 1-step dynamic model of FSMs, i.e.,

$$x\left(t+1\right)=Fu\left(t\right)x\left(t\right), \qquad y(t)=Hu\left(t\right)x\left(t\right), \tag{4}$$

where F and H are referred to as the 1-step transition matrix and 1-step output matrix, respectively.

Different from (4), Yan et al. [31] proposed a t-step dynamic model of FSMs as shown below:

$$x(t+1)=G^{t}x_{0}u(t), \qquad o(t)=K^{t}x_{0}u(t), \tag{5}$$

where $x(t)$ and $u(t)$ are the same as (4), x_0 is the initial state, $o(t)$ is the output at the time t, G^t is called t-step transition matrix and K^t is t-step output matrix.

Remark 3. Eqs. (4) and (5) share similarities in results as well as differences in ideas. In (4) the state of an FSM is defined as a vector $x(t)=(x^1(t),x^2(t),\ldots,x^n(t))^{\mathrm{T}}$, where $x^i(t)$ is the number of different paths by which the FSM can reach state x_i with an input string of length $t-1$, while in (5) state x_i is simply defined as a vector δ_n^i (δ_n^i is the column vector corresponding to the i-th column of the unit matrix I_n). Eq. (4) can be used to analyze both deterministic FSMs and nondeterministic FSMs, and to describe their 1-step dynamics, while Eq. (5) is only suitable for deterministic FSMs, and formulates the t-step dynamics. Thus Eq. (4) excels in the uniform of modeling FSMs and Eq. (5) has advantages in expressing the dynamics. Based on (5), one can analyze and synthesize FSMs like studying dynamic systems in the control theory. The idea, such as feedback [23,30], the concepts, such as controllability, observability, and stabilization [24,52], and the methods, such as states-feedback control, corrective control, and fault-tolerant control [25,54,75], have been transplanted to FSMs.

As an example, consider the FSM, shown in Figure 2(a), that recognizes the language ending with 000 on $E=\{0,1\}$. Its 1-step transition matrix F and 1-step output matrix H are $F=\delta_4[2,3,4,4,1,1,1,1]$ and $H=\delta_2[2,1,2,2,1,2,1,2]$, respectively. While the 6-step transition matrix G^6 is

$$\begin{aligned}G^6=\delta_{256}[&4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,\\&4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,\\&4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,\\&4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,\\&4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,\\&4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,\\&4,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1].\end{aligned}$$

Based on the dynamic models (4) and (5), many problems of FSMs have been reconsidered or solved by borrowing the methods and ideas from control theory. The first problem that attracts the attention of scholars in the control field is the stabilization of FSMs. Xu et al. [22] introduced the concepts of cycle stability and equilibrium from control theory and defined the stabilization of FSMs. Several necessary and sufficient algebraic conditions for the stabilization were proposed. A new way to design state feedback controllers, just by solving matrix inequalities, was developed. Zhang et al. [23,25] borrowed the notions of equilibrium and output feedback feasible event (OFFE) from the system theory and defined the stabilization of Moore-type FSM, a special type of deterministic FSM. Besides, the static output feedback control (SOFC) was investigated by using the STP, including single and multiple equilibrium-based SOFC. Han et al. [30] discussed the problems of equilibrium point stability and equilibrium point stabilization of FSMs by developing a matrix form method. They proposed a matrix-based criterion to verify the stabilization and designed a polynomial algorithm to characterize all permissible state-feedback controllers. Han et al. [24] studied the stabilization of FSMs from the point of view of topological structure. A method of finding limit cycles of different lengths was presented and two new concepts, the attraction domain of limit cycles and pre-reachability, were proposed. Based on these, several criteria for the stabilization of limit cycles were developed and the corresponding controllers were also designed. Besides, other scholars also discussed the stabilization of FSMs from different perspectives [76,77].

Using ASST, scholars in the control field have done extensive research on FSMs, such as controllability [29,31], observability [54], reachability [28], language recognition [27,32,34], compatible-invariant analysis [26], verification analysis [33], the combination of FSMs [78], and space optimization [51].

It is believed that a large number of problems related to FSMs can be further studied or reconsidered in the framework of the ASST.

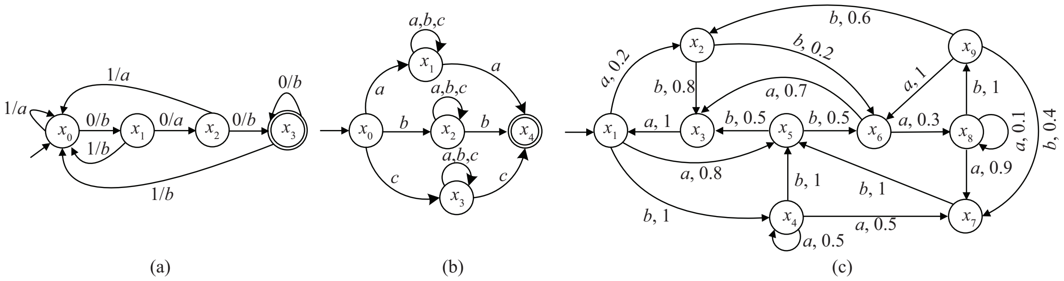

Figure 2 (a) An FSM recognizing the language ending with 000 on alphabet $E = \{0, 1\}$; (b) an NFSM recognizing the languages with the same leading and trailing letters; (c) a model of a probabilistic FSM.

4 Nondeterministic finite state machines

In the evolution of deterministic FSMs, the initial state is only one state, and all transitions are triggered by an event e and determined by a deterministic function. For the convenience of modeling and analysis, these three requirements can be relaxed appropriately. First of all, the transition is excited by an event e at state x to multiple states. Second, an empty event ε is allowed; i.e., the transition between two different states can be caused by an empty event ε. Finally, the initial state can be relaxed to a set of states.

The above factors lead to difficulties in describing the dynamic behaviors of nondeterministic FSMs by models (4) and (5). In 2020, Yue et al. [43] developed a unified model (6) in the ASST framework.

$$x\,(t+1) = F^t x_i u\,(t)\,, \tag{6}$$

where the difference between (6) and models (4) and (5), especially between (5), lies in the way of defining the states of an FSM. The state of (5) is defined as a logical vector, thus causing such type of state can only be applied to represent a single state, while in (6) the state is defined as a "state packet" modeled by an extended logical vector that has multiple elements 1. Therefore the "state" can represent a set of states. For the case of DFSM, the extended logical vector reduces to an ordinary one, and the state packet becomes a single state, and Eq. (6) becomes (5) correspondingly. In other words, Eq. (5) is a special case of (6). Thus the model (6) is applicable to both DFSMs and NFSMs and possesses the advantages of both models (4) and (5), i.e., the model unity of (4) and the dynamic description ability of (5), that is, the ability to formulate the t-step dynamics of FSMs.

Consider the following example of an NFSM, given in Figure 2(b), which recognizes the languages beginning and ending with the same letter. The 5-column-equal-division of the t-step transition matrix (taking $t = 3$ for example) is

$$\mathrm{Blk}_1^5(F^3) = \begin{bmatrix} 000000000000000000000000000 \\ 111111111000000000000000000 \\ 000000000111111111000000000 \\ 000000000000000000111111111 \\ 100100100010010010001001001 \end{bmatrix}, \quad \mathrm{Blk}_2^5(F^3) = \begin{bmatrix} 000000000000000000000000000 \\ 111111111111111111111111111 \\ 000000000000000000000000000 \\ 000000000000000000000000000 \\ 100100100100100100100100100 \end{bmatrix},$$

$$\mathrm{Blk}_3^5(F^3) = \begin{bmatrix} 000000000000000000000000000 \\ 000000000000000000000000000 \\ 111111111111111111111111111 \\ 000000000000000000000000000 \\ 010010010010010010010010010 \end{bmatrix}, \quad \mathrm{Blk}_4^5(F^3) = \begin{bmatrix} 000000000000000000000000000 \\ 000000000000000000000000000 \\ 000000000000000000000000000 \\ 111111111111111111111111111 \\ 001001001001001001001001001 \end{bmatrix},$$

$$\mathrm{Blk}_5^5(F^3) = \delta_5[2, 4, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 5].$$

In the 3-step transition matrix, there are columns having multiple 1s, which are extended logical vectors used to model the state packet representing a set of states. For example, the first column of $\mathrm{Blk}_1^5(F^3)$ has two 1s at positions 2 and 5, respectively, which means that the FSM can move to the second and fifth states from the initial state after 3 steps. In addition, there are 0-columns in the 3-step transition matrix, which shows that there exist states to which the FSM cannot move in 3 steps.

With the t-step transition matrix defined in (6), many problems of NFSMs can be investigated easily, such as controllability, observability, stabilization, reachability and simplification. Yue et al. [37, 42] reconsidered the problem of simplifying FSMs. An algebraic criterion checking different states and a scheme constructing incompatible graphs of FSMs were proposed. Based on them, several sufficient and necessary conditions for formulating three structures of FSMs were established, including compatible coverage sets, representative state sets, and minimum representative sets. These results are applicable to both DFSMs and NFSMs. Li et al. [38] made a comprehensive investigation of reachability, stabilization and controllability of FSMs by introducing the concept of controllability matrix. Xu et al. [41] pointed out that this kind of matrix method may provide a useful way to analyze and compute many interesting and related problems in the area of FSMs. Zhang et al. [35] investigated, under ASST, the control problem of NFSMs which are partially state observable. By providing some algebraic conditions answering the question when an NFSM can be equilibrium-based feedback stabilization, they discussed the range of feasible control events and then designed an algorithm of finding the static feedback matrix. Further, they expected that the corresponding control problem of networked FSMs could be studied and that the mass media of multiplex networked FSMs could also be investigated in the ASST framework. Using ASST, Xu et al. [39] considered the problems of observability and designed an observer of FSMs by viewing FSMs as discrete-time bilinear systems. A matrix form of observability condition and a constructive way to design observers were proposed. They also solved simulation and bi-simulation relations of one or two NFSMs by studying the composition operations of NFSMs and by designing algorithms and criteria for checking the simulation and bi-simulation relations [40].

These researches show that it is reasonable to believe that more problems of NFSMs can be further studied based on model (6).

5 Probabilistic finite state machines

Randomness is an important feature of the world because randomness leads to the uncertainty and unpredictability of the world. Probabilistic FSM (PFSM) is a kind of FSM where there is randomness in its interior or the environment it works in. The main difference between PFSMs and non-PFSMs is that the actions of PFSMs are random.

In 2008, to describe the dynamics of PFSMs, Cassandras et al. [66] proposed the concept of probability transition matrix. The probability transition matrix does not embody the structural information associated with the event-driven. In 2018, Zhang et al. [52] reconsidered the modeling problem of PFSMs from the viewpoint of dynamic systems by using ASST as the mathematical tool. Accordingly, a bilinear-like model was proposed.

Theorem 1. The dynamics of a PFSM M_p excited by a string of length $u(t)$ can be formulated as $E\left[x\left(t+1\right)\right] = Fu\left(t\right)E\left[x\left(t\right)\right]$, in which $E\left[x\left(t\right)\right]$ represents the mathematical expectation of x that can be reached from the initial state with the string, $u(t)$ is the vector form of the string and $F = [F_1, F_2, \ldots, F_m] \in \mathbb{L}_{n\times mn}$ is the transition probabilistic structure matrix, in which $F_i \in \mathbb{M}_{n\times n}$ is associated with event e_k and is defined as follows: $F_{k(j,i)} = P^k_{ij}$, if $x_j \in f(x_i, e_k)$; $F_{k(j,i)} = 0$, otherwise.

The transition probabilistic structure matrix can precisely capture the random information on the stochastic transition of the states during the evolution of a PFSM. This overcomes the defect of the probability transition matrix proposed in [66] that cannot reflect the probabilistic distribution of the event causing a transition.

Consider the PFSM depictured in Figure 2(c). The transition probabilistic structure matrix is

$$F = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0.2 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (7)$$

We can see clearly from (7) that the probabilistic distribution of each event triggering a transfer at a state is contained in the transition probabilistic structure matrix. The essential difference between the transition structure matrix of DFSMs (or NFSMs) and the transition probabilistic structure matrix of PFSMs is that the nonzero elements in these two matrices represent different physical meanings; the former suggests which state(s) can be reached from another state, while the latter indicates the probability that an event raises a state transition.

Based on Theorem 1, many issues of PFSMs were investigated by scholars from the control field. Also, some concepts, methods, and problems were transplanted to PFSMs, such as reachability, controllability, and stabilization. In [52], the reachability problem was discussed. An algebraic form of the necessary and sufficient condition for two types of reachability (ordinary and probabilistic reachability) was reported. Compared with other forms of results, this algebraic form has more theoretical value.

In 2018, Zhang et al. [56] analyzed the reachability of PFSMs in more depth by introducing the notion of F-probabilistic reachability. A sufficient and necessary condition for the F-probabilistic reachability was proposed. In addition, the controllability issue of PFSMs was also discussed in [56]. More precisely, the concept of F-probabilistic controllability was defined, and an algebraic condition was developed to check the F-probabilistic controllability. The approach allows us to convert the state evolutions of PFSMs to discrete-time bilinear systems by which the conditions of probabilistic controllability and probabilistic reachability can be achieved easily. More importantly, the method provides a theoretical way to other problems such as stabilization, observability, and modular control.

Stability and stabilization are fundamental issues for FSMs. For a large-scale automatic system, stabilization means designing a controller to make the system transferable to specific states in finite-time steps. Stabilization and stability of FSMs have been studied extensively from various perspectives [76, 79–82]. Before 2020, there was little report on the stabilization of PFSMs. Zhang et al. [83] addressed the stabilization problem by using ASST and by viewing PFSMs as nondeterministic ones where nondeterministic state transitions are assigned probabilities. The k-controllability with probability one was defined for marked states and the corresponding condition for the controllability was proposed.

Definition 8. Let state $x_m = \delta_n^m$ be a marked state of PFSM M_p. x_m is said to be k-controllable from the initial state $x_0 = \delta_n^p$ with probability one if there is an input s of length k such that $P_{pm}^s = 1$. M_p is said to be k-controllable from the initial state $x_0 = \delta_n^p$ with probability one if for any state x_j there is an input s of length k such that $P_{pj}^s = 1$.

Theorem 2. For a PFSM with dynamic equation described in Theorem 1, a marked state $x_m = \delta_n^m$ is k-controllable from the initial state $x_0 = \delta_n^p$ with probability one iff $\delta_n^m \in \Theta(\lfloor D^k \rfloor)$, where $D = FW_{[n,q]}$, F is the transition probabilistic structure matrix, and $W_{[n,q]}$ is the swap matrix.

The k-controllability of the whole system of PFSMs can be obtained by extending the controllability condition of marked states.

Theorem 3. For the PFSM described in Theorem 2, the system of the PFSM is k-controllable from the initial $x_0 = \delta_n^p$ with probability one iff $\Delta_n \subseteq \Theta(\lfloor D^k \rfloor)$, where D is given in Theorem 2.

On the stabilization of PFSMs, Ref. [83] presented a necessary condition for stabilization and a sufficient condition for stabilization of a closed-loop system.

Definition 9. Probabilistic FSM M_p is called stabilizable to a marked state, taking the state $x_m = \delta_n^m$ for instance, with probability one if there is an input s such that $P_{im}^s = 1$ for any state x_i.

Theorem 4. If the PFSM descried in Theorem 2 is feedback stable to a marked state, taking the state $x_m = \delta_n^m$ for example, with probability one, then there is a k ($k \leqslant n-1$) such that $\Omega_k(m) = \Delta_n$, where $\Omega_k(m) = \Theta(\mathrm{Row}_m(D^k))^{\mathrm{T}}$.

Theorem 5. Assuming that a PFSM satisfies the condition of Theorem 4, there is then a feedback logical matrix such that the closed-loop system constructed by the PFSM can be stabilized to $x_m = \delta_n^m$ with probability one.

In addition, the authors [46] uniformly discussed the problems of reachability, controllability, and stabilization of PFSMs by defining a new type of controllability matrix for PFSMs. The presented approach provides a unified idea for the three problems.

6 Networked finite state machines

Networked FSMs are a useful model to describe cyber-physical systems, discrete event systems, communication systems, smart cities, etc. [84–87]. A networked FSM is built by combining two or more FSMs according to some pre-defined operations on individual FSMs. There are usually two types of operations on FSMs to form a networked FSM. One is the product of FSMs, denoted by $\times$; the other is the parallel composition, denoted by $\|$. The two networked FSMs describe two kinds of shared dynamics of a group of FSMs running at the same time. The critical difference between them is how to handle the private events belonging to component FSMs. Take a networked FSM consisting of two FSMs $M_1 = (X_1, E_1, f_1, x_{01}, X_{m1})$ and $M_2 = (X_2, E_2, f_2, x_{02}, X_{m2})$ for example. The parallel version of networked FSM is defined in (1). The product version, denoted by $M_1 \times M_2 = (X_1 \times X_2, E_1 \cup E_2, f, (x_{01}, x_{02}), X_{m1} \times X_{m2})$, is defined as

$$f\left(\left(x_1, x_2\right), e\right) = \begin{cases} \left(f_1\left(x_1, e\right), f_2\left(x_2, e\right)\right), & \text{if both } f_1\left(x_1, e\right) \text{ and} f_2\left(x_2, e\right) \text{ are defined,} \\ \text{undefined,} & \text{otherwise.} \end{cases} \tag{8}$$

The model (8) suggests that in the product networked FSM, the transitions of M_1 and M_2 must always occur synchronously on common event $e \in E_1 \cup E_2$. Thus $M_1 \times M_2$ describes the clock-synchronization interconnection of M_1 and M_2. In other words, $M_1 \times M_2$ triggers an event iff M_1 and M_2 trigger the event simultaneously. While in the parallel networked FSM, as shown in (1), a common event $e \in E_1 \cup E_2$ is handled only if the common event is executed by M_1 and M_2 simultaneously. Thus the two FSMs M_1 and M_2 are “synchronized” on the common event set $E_1 \cup E_2$. The private events, i.e., those in $E_1 \backslash E_2 \cup E_2 \backslash E_1$, are not subject to such restraints and be allowed to be handled whenever possible.

Comparing (8) with (1), we find that the product networked FSM is restrictive for the fact that it only allows transitions stimulated by shared inputs. However, in practice, when formulating a system that is composed of interacting FSMs, there are usually private events and common events in the event set of each component FSM, where the private events pertain to its own behaviors, the common events are common to other FSMs and capture the couplings among the individual FSMs. Thus a standard networked FSM is built by parallel composition.

Remark 4. From the definitions of networked FSMs, (8) and (1), or the forms of networked FSMs: $M_1 \times M_2 = (X_1 \times X_2, E_1 \cup E_2, f, (x_{01}, x_{02}), X_{m1} \times X_{m2})$ and $M_1 \| M_2 = (X_1 \times X_2, E_1 \cup E_2, f, (x_{01}, x_{02}), X_{m1} \times X_{m2})$, one can see that a networked FSM is in essence an FSM, specifically an NFSM (just viewing $M_1 || M_2$ as M, $X_1 \times X_2$ as X, $E_1 \cup E_2$ as E, (x_{01}, x_{02}) as x, and $X_{m1} \times X_{m2}$ as X_m, respectively). Therefore, we will use the notation of NFSM to express a networked FSM in the sequel of this section.

In advanced manufacturing systems, it is necessary to transfer information using a communication network. The communication network inevitably leads to communication delays or even loss between controllers and plants, which presents a new challenge to manufacturing systems for efficiency and security. Control of networked FSMs has become a very interesting research objective in recent years. Compared with ordinary FSMs, networked FSMs have higher requirements for the communication network, because there is a greater possibility of information loss caused by the more distant communication distance between controllers and plants.

In the past few decades, most studies on networked FSMs have been carried out under the assumption that there is no communication delay or loss. Consequently, how to model, analyze and control networked FSMs with communication delays or loss becomes more and more important. In order to formulate the state evolution of networked FSMs with communication delay or loss, Zhang et al. [49] considered the influence of arbitrary communication loss on the control of networked FSMs from a switched viewpoint. Then a switched model was proposed to describe the dynamics of networked FSMs with communication loss.

Theorem 6. The algebraic form of switched behavior of a networked FSM can be formulated by

$$x\left(t+1\right) = F\theta\left(t\right) u\left(t\right) x\left(t\right), \tag{9}$$

where $x(t)$ denotes the state at time t, F is the switched transition structure matrix, $\theta \in \Delta_2$ is the switching signal, $u(t)$ is the input string.

In order to reduce the computational complexity, a Boolean operation is introduced to (9), thus the algebraic form of switched behavior of a networked FSM can be rewritten as a simpler algebraic equation:

$$x\left(t+1\right) = F\theta_B\left(t\right) u_B\left(t\right) x_B\left(t\right). \tag{10}$$

Based on (10), Zhang et al. [49] further investigated the reachability of networked FSMs with arbitrary communication loss. A set of reachability conditions was presented.

Definition 10. (i) State x_j is k-reachable from initial state x_0 under any communication loss if there is an input s such that $x_j \in f(x_0, s)$ under any switching signals. (ii) The k-reachable set from initial state x_0, $R_k(x_0)$, is the set of k-reachable states from initial state x_0 under arbitrary communication loss. Let $R(x_0)$ denote the set of states reachable from x_0 under arbitrary communication loss. (iii) A networked FSM is called reachable under arbitrary communication loss from initial state x_0 if $R(x_0) = \Delta_n$.

Theorem 7. Consider the networked FSM defined in (1) with communication loss. (i) State x_j is k-reachable from state x_i under any communication loss iff $L^{(k)}_{(j,i)} = 1$, where $L^{(k)}_{(j,i)}$ is the element at the position (j,i) of matrix $L^{(k)} = \vee_{\alpha=1}^{m^k}(D^1_\alpha \wedge D^2_\alpha \wedge \cdots \wedge D^{2^k}_\alpha), D^l_\alpha$ $(l = 1, \ldots, 2^k)$ is 2^k square blocks that are obtained by splitting the right hand of (10), i.e., $F\theta_B(t)\,u_B(t)\,x_B(t)$. (ii) State x_j is reachable from state x_i under arbitrary communication loss iff there is an integer k such that $\vee_{d=1}^{k} L^{(d)}_{(j,i)} = 1$. (iii) A networked FSM is reachable from state x_i under any communication loss iff there is an integer k such that $\vee_{d=1}^{k}\mathrm{Col}_i L^{(d)} = \mathbf{1}_n$.

Finding all the loops and locks of a network of FSMs is a tough and challenging problem. In [51], the problem of how to find loops and lock structures of a networked FSM is investigated by treating a networked FSM as a special kind of NFSM where the states are viewed as "state packets" which corresponds to the state of a networked FSM.

It is convinced that the approach presented to describe the dynamics of networked FSMs may be enlightening to model discrete state systems, such as discrete event systems [88–91], networked games [6, 92, 93], finite-valued systems [75], and Boolean networks [94–96].

7 Controlled and combined finite state machines

Compared with ordinary FSMs, the structure and dynamics of controlled FSMs are more complex, because they allow control specifications to forcibly intervene in the evolution of the systems. Controlled FSMs are one of the four basic modeling and analysis methods for discrete event dynamic systems, and the other three are logic, algebraic, and statistical performance. Therefore, it is of great significance and challenge to study the modeling and analysis of controlled FSMs.

Yan et al. [97] proposed an algebraic form of the model for controlled FSMs based on the algebraic model of ordinary FSMs established in the framework of STP.

Theorem 8. The state evolution of control specification of a controlled FSM $M_c = (\Phi \times E, X, f_c, x_0, X_F)$ can be described as

$$\delta_c\left(f\left(q_j\right), \sigma_i, q_j\right) = \Delta\left(F\delta^j_m\delta^i_n f\left(q_j\right)\left(\sigma_i\right)\right), \tag{11}$$

$$\delta_c\left(f\left(q_j\right), \sigma_i, q_j\right) = \Delta\left(\tilde{F}\delta^i_n\delta^j_m f\left(q_j\right)\left(\sigma_i\right)\right), \tag{12}$$

where $f(q_j)$ denotes the control specification developed from the state q_j, $F = [F_1, \ldots, F_m]$ is defined as follows. $F_{i(s,t)} = 1$, if $\delta^s_n \in f(\delta^t_n, \delta^i_m)$; $F_{i(s,t)} = 0$, otherwise. $\tilde{F} = FW_{[n,m]}$.

Theorem 9. The state evolution of controlled FSM $M_c = (\Phi \times E, X, f_c, x_0, X_F)$ reading an input string $\sigma = \sigma_1\sigma_2\cdots\sigma_t$ can be formulated by

$$X_j = \Delta\left(\tilde{F}^t\delta^i_n u(t)\prod_{j=0}^{t-1} r_j\right), \tag{13}$$

where X_j is the reached state set; $E = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$; $X = \{x_1, x_2, \ldots, x_n\}$; $\tilde{F}$, δ^i_n, and δ^j_m have the same meaning as in Theorem 8; $u(t) = \ltimes_{j=1}^{t}\delta^j_m = \delta^1_m \cdots \delta^t_m$; $r_j = f(\tilde{F}^j\delta^i_n u(j))(\sigma_{j+1})$, where $f \in \Phi, u(j) = \ltimes_{k=1}^{j}\delta^k_m$; $\tilde{F}^0$ and $u(0)$ are defined as I_n.

The reachability of controlled FSMs is similar to that of ordinary FSMs, as defined in the following.

Definition 11. For a controlled FSM $M_c = (\Phi \times E, X, f_c, x_0, X_F)$, a target state x^* is said to be reachable from the initial state x_0 if there exist a control specification $\gamma \in \Phi$ and an input sequence σ such that M_c can move to the target state x^* at least once from the initial state x_0 with the input sequence σ and under the control specification γ. M_c is accessible if every state is reachable from the initial state x_0.

Based on Theorem 9, the reachability problem of controlled FSMs was investigated in [97], and also a reachability condition was presented.

Theorem 10. For the controlled FSM in Theorem 9, a target state $x^* = \delta_n^q$ can be reached from an initial state $x_0 = \delta_n^p$ by a string (the length is t) iff there is $\eta \in \mathrm{col}(R_{x_0}(t))$ such that $\delta_n^q \in \Delta(\eta)$, where $R_{x_0}(t) = \tilde{F}^t \delta_n^i \prod_{j=0}^{t-1} r_j$.

With this reachability theorem, many useful conclusions were obtained. For example, the number of paths of a specified length, taking t for instance, between any two states x_1 and x_2 is $n(x_1, x_2, t)$, where $n(x_1, x_2, t)$ is the number of occurrences of δ_n^q in $\Omega(R_{x_1}(t))$, δ_n^q is the vector form of the state x_2. Another example, if $(R_{x_0}(t)) = X$, a controlled FSM $M_c = (\Phi \times E, X, f_c, x_0, X_F)$ is accessible and vice versa. In addition, the algorithm for finding all the inputs of a specified length that can move a controlled FSM to an expected state was designed (see Algorithm 1).

Algorithm 1 Find inputs moving a controlled FSM to x^* from x_0

1: Check if $\pi = \prod_{j=0}^{t-1} r_j$ is 1. If not, x^* cannot be reached from x_0, and the algorithm ends.
2: If $\pi = 1$, compute $R_{x_0}(t) = \tilde{F}^t \delta_n^p \pi = \prod_{j=0}^{t-1} r_j$.
3: Check if there is an integer $1 \leqslant i \leqslant m^t$ satisfying $\delta_n^q \in \Delta(\mathrm{col}_i(R_{x_0}(t)))$. If not, x^* is unreachable from x_0, and the algorithm ends. Otherwise, set the set $K = \{i | \delta_n^q = \mathrm{col}_i(M), M \in M'(x^0, t)\}$.
4: For each $l \in K$, let $S_{i,m}^t = {E_d}^{m-1} W_{[2^i, 2^{m-i}]}$, $i = 1, 2, \ldots, m$ and compute $\sigma_j := \delta_m^j = S_{j,m}^t {\delta^l}_{m^t}$, $j = 1, 2, \ldots, t$. An input $\sigma = \sigma_1 \sigma \cdots \sigma_t$, corresponding to l, is found. All the expected inputs are $\{\sigma_l | l \in K\}$.

Note: $M \in M'(x^0, t)$ is a matrix defined as follows. If $\mathrm{col}_i(M(x^0, t)) \in \Delta_n, (1 \leqslant i \leqslant m^t)$, then $\mathrm{col}_i(M) = \mathrm{col}_i(M(x^0, t))$. Otherwise, $\mathrm{col}_i(M)$ adopts all the elements of $\Delta(\mathrm{col}_i(M(x^0, t)))$ in a composition order.

A combined FSM is a system of two or more FSMs combined in certain ways to build an integral whole with more powerful functions that a single component FSM cannot achieve. As Gécseg said in [98], in FSM theory, combined FSMs have very important theoretical value and application potential for two reasons: first, for practical applications that can be modeled by FSMs, FSMs are usually composed of different types of FSMs; second, lots of properties of combined FSMs can be deduced from the properties of their component FSMs.

There are three ways of combination: parallel, series, and feedback [62, 99]. Take a combined FSM constructed by two FSMs for example. As shown in Figure 3(a), a combined FSM built in a parallel way is realized by taking its input as the input of each component FSM, and by taking a map of the outputs of each component FSMs as the output of the parallel combined FSM. A serial combined FSM made by FSMs M_1 and M_2, shown in Figure 3(b), means that the input of the combined FSM is the same as the input of M_1. The output of M_2 serves as the output of the combined FSM, wherein the output of M_1 is used as the input of M_2. For a combined FSM constructed by FSMs M_1 and M_2 in feedback mode, shown in Figure 3(c), its input and the output of M_2 are converted by a converter, and the result is used for the input of M_1; the output of M_1 feedbacks to M_2 and is used for the output of the feedback combined FSM.

Compared with regular FSMs, the structure and dynamic behavior of combined FSMs are more complex, as the autonomous operations of each component FSMs are allowed. Therefore, modeling and controlling combined FSMs are significant research subjects. Yan et al. [100] proposed new models for these three kinds of combined FSMs.

Theorem 11. (I) The parallel combined FSM $M = (X, E, W, f, \lambda)$ constructed by $M_1 = (X_1, E, W_1, f_1, \lambda_1)$ and $M_2 = (X_2, E, W_2, f_2, \lambda_2)$ can be described from the following three aspects. (i) $X = \Delta_{n1 \times n2}, E = \Delta_m, W = \Delta_l$. (ii) $f : X \times E \to X$ is computed by $\delta(x_i, e_j) = G x_i (e_j)^2$, where G is referred to as the state transition matrix of M. (iii) $\lambda : X \times E \to W$ is computed by $\lambda(x_i, e_j) = H x_i (e_j)^2$, where H is referred to as the output structure matrix (OSM) of M. For the parallel combined FSM M, the state transition is $x = G_s x_0 {e_1}^2 {e_2}^2 \cdots {e_t}^2$, and the output dynamics is $w^* = H_s x_0 {e_1}^2 {e_2}^2 \cdots {e_t}^2$, where $G_s = G^t, H_s = H G^{t-1}$.

(II) The serial combined FSM $M = (X, E, W, f, \lambda)$ constructed by $M_1 = (X_1, E, W_1, f_1, \lambda_1)$ and $M_2 = (X_2, E, W_2, f_2, \lambda_2)$ can be described as follows. (i) $X = \Delta_{n1 \times n2}, E = \Delta_m, W = \Delta_l$. (ii) $f : X \times E \to X$ is calculated by $f(x_i, e_j) = G(x_i)^2 (e_j)^2$, where G is referred to as the state transition matrix of M. (iii) $\lambda : X \times E \to W$ is computed by $\lambda(x_i, e_j) = H x_i e_j$, where H is referred to as the OSM of M.

(III) The feedback combined FSM $M = (X, E, W, f, \lambda)$ constructed by $M_1 = (X_1, E_1, W, f_1, \lambda_1)$ and $M_2 = (X_2, E, E_2, f_2, \lambda_2)$ can be described by the following. (i) $X = \Delta_{n1 \times n2}, E = \Delta_m, W = \Delta_l$. (ii) $f : X \times \sum \to X$ is calculated by $f(x_i, e_j) = G(x_i)^3 (e_j)^2$, where G is referred to as the state transition matrix

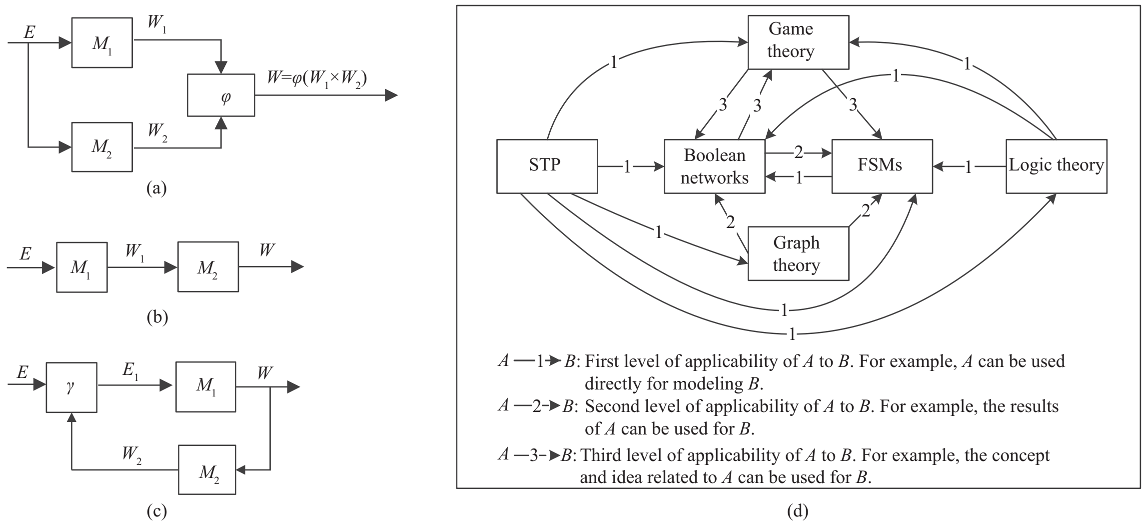

Figure 3 (a) Parallel combination; (b) serial combination; (c) feedback combination; (d) three levels of logical relationships among FSMs, Boolean networks, game theory, logic theory, graph theory, and STP.

of M. (iii) $\lambda : X \times E \to W$ is computed by $\lambda(x_i, e_j) = Hx_ie_j$, where H is referred to as the output structure matrix of M.

Remark 5. The above conclusion has the following advantages. (i) The construction and control problems of combined FSMs can be accurately formulated as matrix form, which is quite different from the existing results. (ii) To construct or control these three kinds of combined FSMs, we only need to perform a series of operations of STP, and then the combined FSMs to be constructed or the desired control inputs can be easily obtained. It should be noted that although this method has the advantages of concise mathematical expression and accurate decisions, it cannot reduce computational complexity.

Based on Theorem 11, Chen et al. [101] investigated the observability analysis of the kinds of combined FSMs, constructed in the manners of parallel, serial, and feedback. Compared with the commonly used formal language method, the advantage of the presented method lies in problem description and solving. In order to solve the problem of the inconsistent framework of these combined FSMs, the structure matrices of these combined FSMs should be optimized first. In addition, to reduce the storage space occupied by a large number of zero elements in the logical matrix, a simplified incidence matrix was constructed by the labeling method. On these bases, the observability analysis of combined FSMs can be conducted with matrix polynomials. In addition, two algorithms were designed for judging the initial and current state observabilities of combined FSMs.

8 Other applications

In addition to the above applications, in the framework of STP, FSM theory and methods have been applied in other fields, such as Boolean networks and Boolean control networks (BCNs). And, in the framework of STP, other theories, such as graph theory, have been successfully applied to the FSM field. This section briefly surveys these research progresses.

8.1 Applications of FSMs to BCNs

Boolean networks provide a discrete dynamic process to formulate gene regulatory networks. Ref. [102] stated that "one of the main research objectives of system biology is to develop a set of control theories for complex biological systems". Therefore, it is of great theoretical and practical significance to study the control of the Boolean network (or Boolean control network). The observability of Boolean control networks is not only a basic problem but an important problem in control theory.

Before 2014, there were four different kinds of definitions for the observability of Boolean control networks [103–110], which were defined from different perspectives or from different emphases and each of them has its own inherent defects. There is no unified way to deal with it, nor is there an answer to the question "Are there unified methods or algorithms to judge whether a Boolean control network is observable with respect to these different notations of observability?"

In 2015, Zhang et al. [111] discussed the observability of Boolean control networks in a general way by using FSM theory under ASST. They made a breakthrough and gave the answer to the aforementioned question. Specifically, the authors proposed a unified method to design algorithms to determine whether a given Boolean control network is observable. To this end, the concept of a weighted pair graph of Boolean control networks was proposed; by using the weighted pair graph, the FSM theory, and the formal language theory, several equivalent criteria for checking multiple types of observabilities were established, and an algorithm was designed to judge whether a Boolean control network is observable with respect to the observabilities. In addition, the implication relationships among the observabilities were also revealed.

A Boolean control network is modeled by a finite number of coupled logical functions:

$$\begin{cases} x_1(t+1) = f_1(u_1(t), \ldots, u_m(t), x_1(t), \ldots, x_n(t)), \\ \cdots \\ x_n(t+1) = f_n(u_1(t), \ldots, u_m(t), x_1(t), \ldots, x_n(t)), \end{cases}$$
$$y_j(t) = h_j(x_1(t), \ldots, x_n(t)), \quad j = 1, \ldots, q, \tag{14}$$

where $u_1, \ldots, u_m$ denote the m inputs of the network, $x_1, \ldots, x_n$ are the n states, $y_1, \ldots, y_q$ represent the q inputs, f_i is a Boolean mapping from D^{n+m} to D, and h_j is also a Boolean mapping from D^n to D.

By identifying $1 \sim \delta_2^1$, $0 \sim \delta_2^2$ and using the STP, Cheng et al. [112] developed an algebraic model of the Boolean control network:

$$x(t+1) = Lu(t)x(t), \qquad y(t) = Hx(t), \tag{15}$$

where u, x and y belong to Δ_m, Δ_n and Δ_q, respectively.

The observability of Boolean control networks has been defined in four different ways [103–110].

Definition 12. Boolean control network (15) is said to be observable, if there is an input such that the initial state $x_0 \in \Delta_n$ can be determined by outputs.

Definition 13. Boolean control network (15) is observable, if there exists an input $u \in (\Delta_m)^p$ such that $(HL)_{x_0}^p u \neq (HL)_{\bar{x}_0}^p u$ for any different states $x_0, \bar{x}_0 \in \Delta_n$.

Definition 14. Boolean control network (15) is an observable network if there exists an input $u \in (\Delta_m)^n$ such that the initial state $x_0 \in \Delta_n$ can be determined by the output $(HL)_{x_0}^n u$.

Definition 15. Boolean control network (15) is observable, if $Hx_0 = H\bar{x}_0$, indicating $(HL)_{x_0}^n u \neq (HL)_{\bar{x}_0}^p u$ holds for any different states $x_0, \bar{x}_0 \in \Delta_n$ and any input $u \in (\Delta_m)^n$.

For the four observability definitions of Boolean control networks, Ref. [111] reported the corresponding observability theorems and their implication relationships.

Theorem 12. (i) Boolean control network (15) is unobservable under Definition 12 iff there exists a state δ_n^i being not a diagonal vertex in the weighted pair graph (WPG) of Boolean control network (15) such that the FSM corresponding to the WPG generates the language $(\Delta_m)^*$. (ii) Boolean control network (15) is unobservable under Definition 13 iff there exists a non-diagonal vertex (δ_n^i, δ_n^j) in the WPG of Boolean control network (15) such that the FSM corresponding to the WPG generates the language $(\Delta_m)^*$. (iii) Boolean control network (15) is unobservable under Definition 14 iff the FSM corresponding to the WPG of Boolean control network (15) generates the language $(\Delta_m)^*$.

The WPG of Boolean control network (15) is a weighted digraph $G = (V, E, W)$, where V is the vertex set, $E \subset V \times V$ is the edge set, and W is the weight function $V = \{(x, x')|x, x' \in \Delta_n, Hx = Hx'\}$, $((x_1, x_1'), (x_2, x_2')) \in E$ iff there is a $u_1 \in \Delta_m$ such that $Lu_1x_1 = x_2$ and $Lu_1x_1' = x_2'$, $W(e) = \{u_1 \in \Delta_m | Lu_1x_1 = x_2, Lu_1x_1' = x_2'\}$.

Remark 6. Determining each type of observability of BCNs is NP-hard. The time complexity of the algorithms designed using Theorem 12 is exponential in time. It is challenging and urgent to reduce computational complexity. The challenge can be expressed as "Is it NP-hard to determine the observability of Boolean control networks?" As for the space complexity, Ref. [113] gave a deep discussion: determining the existence of fixed points of Boolean control networks is NP-complete.

The theory of FSM was also applied to studying the observability of switched BCNs in the framework of STP [114], where the authors proposed the concept of the weighted pair graph of switched Boolean control networks. Based on this concept, switched Boolean control networks were converted into deterministic FSMs. Then the FSM theory was used to investigate the observability of switched Boolean control networks by checking whether the corresponding FSMs are complete or not. Sufficient or necessary conditions for the observability were established, and algorithms for checking the observability were developed. However, the algorithms have a drawback in that the computational space complexity is exponential in the size of the Boolean control network.

8.2 Applications of graph theory to FSMs

In addition to directly using ASST to study the issues of FSMs, as presented in Sections 3–7, there are also indirect applications of ASST in the field of FSMs. For example, ASST was applied to solving some problems in graph theory, and the research results were then applied back to FSMs.

Using ASST as an analysis tool, many issues in the field of graph theory have also been solved more perfectly, such as vertex coloring and the maximum weight stable set [115], robust graph coloring and its application to examination timetabling [116], conflict-free coloring and its application to frequency assignment of communication systems [117], fuzzy graph coloring [118], externally stable sets and cores of graphs [119], hypergraph stable sets and coloring to apply to the storing problem [120], k-internally stable sets of graphs and their application to the k-track assignment problem [121].

Graphs and FSMs also have connections in essence. In fact, an FSM can be represented by its state transition structure diagram, which is a graph. This connection provides a way to study FSM using graph theory. In particular, the results of k-internally stable sets of graphs obtained by using ASST [121] provide support for the construction of incompatible graphs of FSMs.

Definition 16. Given an FSM M, two states $x_i, x_j \in X$ are called k-different states if there exist inputs (or exists an input) of length k making M produce distinct reactions or outputs when M is excited by these inputs at x_i and x_j, respectively; otherwise $x_i, x_j \in X$ are k-identical. A pair of states $x_i, x_j \in X$ is said to be incompatible if they are k-different for all integers $k \geqslant 1$; otherwise, $x_i, x_j \in X$ are said to be a compatible pair. The incompatible graph of M is a graph $G = (X, E)$, where X is the state set of M, $(x_i, x_j) \in E$ iff x_i and x_j are an incompatible pair.

In [122], the authors presented a criterion of k-identical states and algorithms of constructing the incompatible graph of FSMs.

Theorem 13. Supposing that states $x_i, x_j \in X$ of FSM M are k-identical, then they are $(k+1)$-different iff there exists an input $e \in E$ such that $f(e, x_i)$ and $f(e, x_j)$ are k-different.

Theorem 13 and Algorithms 2 and 3 lay the foundation for the optimization of FSMs in the context of STP. The optimization of FSMs is an important problem in the theory of FSMs and has important value in engineering applications as the storage space of the hardware is exponential in the number of the states of an FSM. Optimizing an FSM means minimizing the states without changing its function. The optimization problem has been widely studied from various perspectives and aspects, including the use of different methods to optimize the same type of FSMs [123–125], the same or similar methods to optimize different types of FSMs [126–128], and specific methods to solve specific problems [129,130], etc.

Algorithm 2 Check of r-different of states x_i and x_j for an FSM

1: Set $r = 1$.
2: Set $t = r + 1$.
3: Compute $H^r_{ij} = \{k|\mathrm{Col}_k(K^r_i) \neq \delta^0_l$ and $\mathrm{Col}_k(K^r_j) \neq \delta^0_l\}$, where $K^r_i := G^t \delta^i_n$, $K^r_j := G^t \delta^j_n$, G^t is given in (5), and l is the number of outputs of the FSM.
4: Check whether there is a $k \in H^r_{ij}$ such that $\mathrm{Col}_k(K^r_i) \neq \mathrm{Col}_k(K^r_j)$. If yes or $H^r_{ij} = \emptyset$, x_i and x_j are r-different, then they are t-different and Algorithm 2 terminates; otherwise go to the next step.
5: Let $r = r + 1$. Go to Step 2.

Algorithm 3 Construction of the incompatibility graph of FSMs

1: Build the set $S_0 = \{(x_i, x_j)\,|i, j = 1, \ldots, n,\ i < j\}$, $S_m = \emptyset$.
2: Let $r = 1$.
3: Use Algorithm 2 to obtain all r-different states from S_0 and build: $S_r = \{(x_i, x_j)|(x_i, x_j) \in S_0$, x_i and x_j are r-different$\}$, $S_m = S_m \cup S_r$, $S_0 = S_0 - S_m$.
4: If $S_r \neq \emptyset$, let $r = r + 1$ and go to the next step; otherwise, S_m includes all the incompatible state pairs and go to the next step.
5: $G = (S, S_m)$ is the incompatibility graph.

The authors in [37,122] studied the simplification of FSMs by using the construction method of incompatibility graph of FSMs. Several necessary and sufficient conditions for simplifying FSMs and procedures for reducing FSMs were proposed. In [51], the optimization of FSMs was investigated from a control theory viewpoint by borrowing the concepts of equilibrium point and system convergence, where FSMs were viewed as logical dynamic systems. Two methods for reducing the state space of FSMs were presented.

Remark 7. FSMs and Boolean networks are logical dynamic systems in essence, and their dynamic evolutions are governed by some logical rules. And ASST especially excels in the formulation of logical behavior of systems. These make it possible to study Boolean networks by using FSMs. In fact, as presented above, FSMs possess some inherent advantages in dealing with some problems of Boolean networks. One such advantage is its ability to uniformly consider problems defined from different perspectives. Taking the observability of Boolean networks for instance, before introducing ASST to Boolean networks, there are four types of definitions for the issue, and each has its own inherent shortcomings; there is neither a unified way to handle it nor a solution to judge these observabilities in a uniform manner. FSMs provide a method to study the four kinds of observabilities, and a universal judgment criterion of these observabilities and a unified algorithm to determine them are obtained (see [111] for details).

In the area of graph theory, the advantage of the method is, as Wang et al. [115] pointed out, that some structures of the graph can be expressed in an explicit algebraic form. With the algebraic formulation, the problem of finding all the structures is converted into the problem of computing the STP of a series of matrices. This overcomes the disadvantage of using computer algorithms to find a structure of a graph: only one or more such structures can be found, no guarantee of all these structures.

The disadvantage of the FSM method to investigate problems of Boolean networks and graphs is the computation complexity. The required computer memory space grows exponentially with the number of nodes of a Boolean network or a graph. For an ordinary laptop computer, the method can cope with Boolean networks or graphs with no more than 12 nodes. Fortunately, the drawback may be alleviated with the rapid development of the storage and computing power of computers.

9 Concluding remarks

This study reviews applications of ASST, which suggest that many issues can be considered constructively by using ASST as a mathematical analysis tool from the perspective of mathematics. Although there are still some flaws in these results, we believe, in the wake of developments in science and technology, that more and wider problems in the field of FSMs can be considered or reconsidered by the ASST method in the future.

According to the literature review presented in this survey, especially to the algebraic model of FSMs developed by using the ASST method, we are confident that in the FSM field the ASST method can be used to further study many topics related to FSMs, such as state space refinement, design of observer FSM, determination of equivalent FSMs, safety and blocking, event diagnosis, formal verification, and model checking of discrete event dynamic systems.

A common challenge for the above subjects is to establish the algebraic formulations of logical evolutions involved in the above problems. Specifically, for the state space refinement of FSMs, the difficult tasks are how to find all the equivalent states, how to express the logical connections between these equivalent states, and how to optimize the logical connections which are expected to be expressed as algebraic expressions. Fortunately, the solutions to deal with deterministic FSMs and nondeterministic FSMs, presented in Sections 3 and 4, can be used as a reference scheme.

For the design of observer FSM, the difficulty lies in the way to introduce the empty transition to the nondeterministic FSMs that are served as the observer FSMs. One possible solution is to develop a testing method to compare DFSMs and nondeterministic FSMs in terms of the power of language recognition. Again, the modeling frameworks presented in Sections 3 and 4 may have reference values.

A simulation challenge occurs when judging the equivalence between FSMs because NFSMs can capture the dynamic evolutions of the systems beyond the language recognized by DFSMs. Thus we must study the bi-simulation relationship and isomorphic between DFSMs and nondeterministic FSMs by establishing the bilinear algebraic equations of these two relationships.

To further study the safety and blocking of FSMs in the framework of STP, we are faced with the difficulty that the partially-observed observer of FSMs must be built in advance, where we will confront again the challenges of designing a full-observed observer described above. The idea and method of studying the controlled and combined FSMs, presented in Section 7, may be helpful for the situation of constructing a partially-observed observer.

In the research of event diagnosis of FSMs, a major difficulty may emerge in formulating whether an unobservable event has happened or must have happened when the system executes a string of events. A possible suggestion is to use the special properties of STP to explore the representability of partial variables in the algebraic expression of a logic function, that is, use the semi-tensor product of all variables of the logic function to represent some of their variables that correspond to the unobservable events.

The core task of formal verification is to find, according to a specification, all errors or bugs by exploring all the possible behaviors. Model checking (MC) is a significant method applied to formal verification. MC is applicable to the discrete event dynamic systems (DEDSs) modeled by FSMs and specifications that are expressed by temporal logic propositions. The propositions are used to describe the blocking, safety and diagnosability. Thus the challenges here include all the difficulties encountered in the blocking, safety and diagnosability, which are discussed immediately above. Of course, the corresponding suggestions may also be helpful for the problems. Besides, using a binary decision graph to represent the transition structure of FSM is another feasible method, where the structure of the graph can be modeled by STP.

In addition, some ideas and methods from other fields, such as the ideas of disturbance decoupling, noise sensitivity and model construction of Boolean control networks, and the ideas of localization and the method of identification technique of finite game systems, can be borrowed to investigate FSMs with the help of ASST method.

Concerning further applications of the methods, approaches and results of FSMs obtained by the STP, they can be applied to many fields, for example, discrete event systems. Moreover, since graph theory is essentially related to both FSMs and multi-agent systems, the use of FSMs to solve some problems in agent networks may be a future research direction, such as networked robot manipulators.

One challenge we face with discrete event systems is the establishment of the connection between finite state machines and regular languages and the connection between regular languages and discrete event systems. The first difficulty that must be solved is to create the algebraic expressions of regular languages and their operations, such as concatenation, prefix-closure, post-language, and projection. With this preparation, we can then use the relevant conclusions of FSMs to study discrete event systems. In addition, for large-scale discrete event systems, the computational complexity caused by state space explosion is also a challenge in this direction.

For the networked robot manipulators governed by fuzzy rules, the difficulty to overcome is the conversion of fuzzy control rules presented by fuzzy logic to algebraic equations in the framework of STP. A possible research idea is to discretize the field of fuzzy logic into one of finite-valued logic, for example, $0, 0.1, 0.2, \ldots, 1$. Then use the method that expresses multivalued logic functions and mixed-value logic functions as algebraic forms of matrices to model the fuzzy logic as algebraic expressions.

Moreover, the cross-integration of FSMs, Boolean networks, game theory, logic theory, graph theory, and STP is also a feasible research direction due to the following facts, as shown in Figure 3. (a) Both FSMs and Boolean networks are logically dynamic systems. They have similar essence: the system behavior is logical dynamics. (b) Dynamic evolutions of both FSMs and Boolean networks can be modeled by STP, and the two models are very similar in mathematical expressions. (c) FSMs and Boolean networks are graphs. Naturally, graph theory connects them. (d) The structure of a graph can also be modeled by STP, where the advantage is that all cases of a special structure contained in a graph can be obtained by executing a series of STP operations.

Finally, it is worthwhile pointing out that great effort should be made to reduce the computational complexity of using the ASST method to solve the problems of FSMs, as well as the problems of Boolean networks, games, and graphs, etc.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. U1804150, 62073124, 61973175).

References

1 Deif A S. Advanced Matrix Theory for Scientists ADN Engineers. Boca Raton: CRC Press, 1990
2 Zhang F. Matrix Theory, Basic Results and Techniques. New York: Springer-Verlag, 1999
3 Cheng D Z. Semi-tensor product of matrices and its application to Morgen's problem. Sci China Ser F-Inf Sci, 2001, 44: 195–212
4 Zhong J, Ho D W C, Lu J, et al. Pinning controllers for activation output tracking of Boolean network under one-bit perturbation. IEEE Trans Cybern, 2019, 49: 3398–3408

5 Wu Y H, Sun X M, Zhao X D, et al. Optimal control of Boolean control networks with average cost: a policy iteration approach. Automatica, 2019, 100: 378–387

6 Cheng D Z, Liu T. From Boolean game to potential game. Automatica, 2018, 96: 51–60

7 Yue J M, Yan Y Y. Update law of simplifying finite state machines (FSMs): an answer to the open question of the unmanned optimization of FSMs. IEEE Trans Circ Syst II, 2022, 69: 1164–1167

8 Wu Y H, Shen T L. Policy iteration approach to control residual gas fraction in IC engines under the framework of stochastic logical dynamics. IEEE Trans Contr Syst Technol, 2017, 25: 1100–1107

9 Xu X R, Hong Y G. Matrix approach to model matching of asynchronous sequential machines. IEEE Trans Automat Contr, 2013, 58: 2974–2979

10 Lv Z Y, Wu Y H, Zhao Q, et al. Design and control of a novel coaxial tilt-rotor UAV. IEEE Trans Ind Electron, 2022, 69: 3810–3821

11 Li Y L, Li H T, Ding X Y, et al. Leader-follower consensus of multiagent systems with time delays over finite fields. IEEE Trans Cybern, 2019, 49: 3203–3208

12 Li F F, Yan H C, Karimi H R. Single-input pinning controller design for reachability of Boolean networks. IEEE Trans Neural Netw Learn Syst, 2018, 29: 3264–3269

13 Zhong J, Lu J Q, Huang T W, et al. Controllability and synchronization analysis of identical-hierarchy mixed-valued logical control networks. IEEE Trans Cybern, 2017, 47: 3482–3493

14 Wang Y H, Cheng D Z, Liu X Y. Matrix expression of Shapley values and its application to distributed resource allocation. Sci China Inf Sci, 2019, 62: 022201

15 Wang B, Feng J E. On detectability of probabilistic Boolean networks. Inf Sci, 2019, 483: 383–395

16 Li H T, Zhao G D, Meng M, et al. A survey on applications of semi-tensor product method in engineering. Sci China Inf Sci, 2018, 61: 010202

17 Lu J Q, Li M L, Huang T W, et al. The transformation between the Galois NLFSRs and the Fibonacci NLFSRs via semi-tensor product of matrices. Automatica, 2018, 96: 393–397

18 Yang M, Li R, Chu T G. Construction of a Boolean model of gene and protein regulatory network with memory. Neural Networks, 2014, 52: 18–24

19 Yue J, Yan Y, Chen Z, et al. Identification of predictors of Boolean networks from observed attractor states. Math Meth Appl Sci, 2019, 42: 3848–3864

20 Cheng D Z. On equivalence of matrices. Asian J Math, 2019, 23: 257–348

21 Zhao G D, Wang Y Z. Formulation and optimization control of a class of networked evolutionary games with switched topologies. Nonlinear Anal-Hybrid Syst, 2016, 22: 98–107

22 Xu X R, Zhang Y Q, Hong Y G. Matrix approach to stabilizability of deterministic finite automata. In: Proceedings of American Control Conference (ACC), 2013. 3242–3247

23 Zhang Z P, Chen Z Q, Han X G, et al. On the static output feedback stabilization of deterministic finite automata based upon the approach of semi-tensor product of matrix. Kybernetika, 2018, 54: 41–60

24 Han X G, Chen Z Q, Liu Z X, et al. The detection and stabilisation of limit cycle for deterministic finite automata. Int J Control, 2018, 91: 874–886

25 Zhang Z P, Chen Z Q, Han X G, et al. Static output feedback stabilization of deterministic finite automat. In: Proceedings of the 36th Chinese Control Conference, Dalian, 2017. 2421–2425

26 Zhang Z P, Chen Z Q, Liu Z X. Compatible-invariant subset analysis of deterministic finite automata via semi-tensor product of matrices approach. J China Univ Posts and Telecommun, 2018, 25: 75–82

27 Yue J M, Yan Y Y, Chen Z Q. Language acceptability of finite automata based on theory of semi-tensor product of matrices. Asian J Control, 2019, 21: 2634–2643

28 Yan Y Y, Chen Z Q, Liu Z X. Semi-tensor product of matrices approach to reachability of finite automata with application to language recognition. Front Comput Sci, 2014, 8: 948–957

29 Yan Y Y, Chen Z Q, Yue J M. STP approach to controlliability of finite state machines. In: Proceedings of the 17th IFAC Symposium on System Identification, Beijing, 2015

30 Han X G, Chen Z Q. A matrix-based approach to verifying stability and synthesizing optimal stabilizing controllers for finite-state automata. J Franklin Inst, 2018, 355: 8642–8663

31 Yan Y Y, Chen Z Q, Liu Z X. Semi-tensor product approach to controllability and stabilizability of finite automata. J Syst Eng Electron, 2015, 26: 134–141

32 Yue J M, Yan Y Y, Li Z Q, et al. Matrix conditions of language recognition for finite state machines using the theory of semi-tensor product of matrices. In: Proceedings of the 38th Chinese Control Conference, Guangzhou, 2019. 65–70

33 Yan Y Y, Chen Z Q, Liu Z X. Verification analysis of self-verifying automata via semi-tensor product of matrices. J China Univ Posts Telecommun, 2014, 21: 96–104

34 Yan Y Y, Yue J M, Fu Z M, et al. Algebraic criteria for finite automata understanding of regular language. Front Comput Sci, 2019, 13: 1148–1150

35 Zhang Z P, Xia C Y, Chen Z Q. On the stabilization of nondeterministic finite automata via static output feedback. Appl Math Comput, 2020, 365: 124687

36 Wang H H, Zhao L Y, Li P. Nondeterministic finite automata based on quantum logic: language equivalence relation and robustness. Int J Approximate Reasoning, 2021, 129: 20–40

37 Yue J M, Yan Y Y, Chen Z Q. Three matrix conditions for the reduction of finite automata based on the theory of semi-tensor product of matrices. Sci China Inf Sci, 2020, 63: 129203

38 Li Y L, Dou W H, Li H T, et al. Controllability, reachability, and stabilizability of finite automata: a controllability matrix method. Math Problems Eng, 2018, 2018: 1–6

39 Xu X R, H Y G. Observability analysis and observer design for finite automata via matrix approach. IET Control Theor & Appl, 2013, 7: 1609–1615

40 Xu X R, Hong Y G, Lin H. Matrix approach to simulation and bisimulation analysis of finite automata. In: Proceedings of the 10th World Congress on Intelligent Control and Automation (WCICA), 2012. 2716–2721
41 Xu X R, Hong Y G. Matrix expression and reachability analysis of finite automata. J Control Theor Appl, 2012, 10: 210–215
42 Yan Y Y, Yue J M, Fu Z M, et al. Construction of incompatible graph of finite state machines using the theory of semi-tensor product of matrices. In: Proceedings of the 38th Chinese Control Conference, Guangzhou, 2019. 59–64
43 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: 219201
44 Nan Y, Yi G X, Wang C H, et al. A novel effectiveness evaluation method based on simultaneous probabilistic finite-state machines. IEEE Syst J, 2020, 14: 1611–1622
45 Li J, Tan Y. A probabilistic finite state machine based strategy for multi-target search using swarm robotics. Appl Soft Computing, 2019, 77: 467–483
46 Dou W H, Li H T, Alsaadi F E. Semitensor product approach to controllability, reachability, and stabilizability of probabilistic finite automata. Math Problems Eng, 2019, 2019: 1–7
47 Li Y F, Pan X D, Wu G C, et al. Shape-instability life scatter prediction of 40Cr steel: damage-coupled crystal plastic probabilistic finite element method. Int J Plast, 2016, 79: 1–18
48 Guo T, Frangopol D M, Chen Y W. Fatigue reliability assessment of steel bridge details integrating weigh-in-motion data and probabilistic finite element analysis. Comput Struct, 2012, 112–113: 245–257
49 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
50 Yan Y Y, Deng H, Chen Z Q. A new look at the critical observability of finite state machines from an algebraic viewpoint. Asian J Control, 2022, 24: 3056–3065
51 Yue J M, Yan Y Y, Chen Z Q, et al. State space optimization of finite state machines from the viewpoint of control theory. Front Inform Technol Electron Eng, 2021, 22: 1598–1609
52 Zhang Z P, Chen Z Q, Liu Z X. Modeling and reachability of probabilistic finite automata based on semi-tensor product of matrices. Sci China Inf Sci, 2018, 61: 129202
53 Zhao J T, Chen Z Q, Liu Z X. Reachability and controllability of time-variant k-valued logical control network and finite memories k-valued logical control network. In: Proceedings of the 36th Chinese Control Conference, Dalian, 2017. 2379–2386
54 Gao N, Han X G, Chen Z Q, et al. A novel matrix approach to observability analysis of finite automata. Int J Syst Sci, 2017, 48: 3558–3568
55 Brzozowski J, Ye Y L. Gate circuits with feedback in finite multivalued algebras of transients. J Multiple-Valued Logic Soft Comput, 2010, 16: 155–176
56 Zhang Z P, Chen Z Q, Liu Z C. Reachability and controllability analysis of probabilistic finite automata via a novel matrix method. Asian J Control, 2019, 21: 2578–2586
57 Yang J M. Tolerating permanent state transition faults in asynchronous sequential machines. J Comput Sci Technol, 2016, 31: 1028–1037
58 Yang J M, Kwak S W. Static corrective control for asynchronous sequential machines and its application to on-board computers. IET Control Theor & Appl, 2016, 10: 2526–2533
59 Yang J M. Corrective control of composite asynchronous sequential machines under partial observation. IEEE Trans Automat Contr, 2016, 61: 473–478
60 Cheng D Z, Qi H S, Zhao Y. An Introduction to Semi-Tensor Product of Matrices and Its Applications. Singapore: World Scientific Publishing Co. Pte. Ltd., 2012
61 Cheng D Z, Qi H S. A linear representation of dynamics of Boolean networks. IEEE Trans Automat Contr, 2010, 55: 2251–2258
62 Kohavi Z, Jha N K. Switching and Finite Automata Theory. 3rd ed. New York: Cambridge University Press, 2010
63 Chen W Y. Theory of Finite Automata (in Chinese). Chengdu: University of Electronic Science Technology Press, 2007
64 Barkalov A, Bieganowski J, Titarenko L. Logic Synthesis for Finite State Machines Based on Linear Chains of States: Foundations, Recent Developments and Challenges. Berlin: Springer, 2018
65 Goren S, Ferguson F J. Chesmin: a heuristic for state reduction in incompletely specified finite state machines. In: Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2002
66 Cassandras C G, Lafortune S. Introduction to Discrete Event Systems. 2nd ed. New York: Springer, 2008
67 Shu S L, Lin F, Ying H, et al. State estimation and detectability of probabilistic discrete event systems. Automatica, 2008, 44: 3054–3060
68 Viard K, Fanti M P, Faraut G, et al. Human activity discovery and recognition using probabilistic finite-state automata. IEEE Trans Automat Sci Eng, 2020, 17: 2085–2096
69 Liu C H, Huynh D Q, Reynolds M. Toward occlusion handling in visual tracking via probabilistic finite state machines. IEEE Trans Cybern, 2020, 50: 1726–1738
70 Hsu W C, Lee L W, Tseng K H, et al. Design of feedback control for networked finite-distributed delays systems with quantization and packet dropout compensation. Discrete Dyn Nat Soc, 2015, 2015: 1–15
71 Emek Y, Uitto J. Dynamic networks of finite state machines. Theor Comput Sci, 2020, 810: 58–71
72 Guiraud P, Tanré E. Stability of synchronization under stochastic perturbations in leaky integrate and fire neural networks of finite size. Discrete Cont Dyn Syst-B, 2017, 22: 1–19
73 Meduna A, Zemek P. Controlled finite automata. Acta Inform, 2014, 51: 327–337
74 Farwer B, Jantzen M, Kudlek M, et al. Petri net controlled finite automata. Fundamenta Inform, 2008, 85: 111–121
75 Lu J Q, Li H T, Liu Y, et al. Survey on semi-tensor product method with its applications in logical networks and other finite-valued systems. IET Control Theor & Appl, 2017, 11: 2040–2047
76 Kobayashi K, Imura J, Hiraishi K. Stabilization of finite automata with application to hybrid systems control. Discrete Event Dyn Syst, 2011, 21: 519–545

77 Kari J. Synchronization and stability of finite automata. J Universal Comput Sci, 2002, 8: 270–277
78 Yan Y Y, Chen Z Q, Liu Z X, et al. Modelling combined automata via semi-tensor product of matrices. In: Proceedings of the 33rd Chinese Control Conference, Nanjing, 2014. 6560–6565
79 Češka M, Havlena V, Holík L, et al. Approximate reduction of finite automata for high-speed network intrusion detection. Int J Softw Tools Technol Transfer, 2020, 22: 523–539
80 Hong S P, Kim Y S, Kwon W H, et al. Model and stability of hybrid linear system. In: Proceedings of the DIMACS/SYCON Workshop on Hybrid Systems III: Verification and Control: Verification and Control, 1996. 1066: 424–435
81 Ozveren C M, Willsky A S. Stabilization, tracking, and hierarchical modeling of discrete-event dynamic systems. IFAC Proc Volumes, 1990, 23: 529–533
82 Ozveren C M, Willsky A S. Output stabilizability of discrete-event dynamic systems. IEEE Trans Automat Contr, 1991, 36: 925–935
83 Zhang Z P, Chen Z Q, Han X G, et al. Stabilization of probabilistic finite automata based on semi-tensor product of matrices. J Franklin Inst, 2020, 357: 5173–5186
84 Song H, Srinivasan R, Sookoor T, et al. Smart Cities: Foundations, Principles, and Applications. Hoboken: Wiley, 2017
85 Song H, Rawat D B, Jeschke S, et al. Cyber-Physical Systems: Foundations, Principles and Applications. Orlando: Academic, 2016
86 Li H S, Lai L F, Poor H V. Multicast routing for decentralized control of cyber physical systems with an application in smart grid. IEEE J Sel Areas Commun, 2012, 30: 1097–1107
87 Khan A A, Beg O A, Alamaniotis M, et al. Intelligent anomaly identification in cyber-physical inverter-based systems. Electric Power Syst Res, 2021, 193: 107024
88 Chen Z Q, Zhou Y R, Zhang Z P, et al. Semi-tensor product of matrices approach to the problem of fault detection for discrete event systems (DESs). IEEE Trans Circ Syst II, 2020, 67: 3098–3102
89 Zhang Z P, Chen Z Q, Han X G, et al. On the static output feedback stabilisation of discrete event dynamic systems based upon the approach of semi-tensor product of matrices. Int J Syst Sci, 2019, 50: 1595–1608
90 Wang B, Feng J E, Meng M. Matrix approach to detectability of discrete event systems. J Franklin Inst, 2019, 356: 6460–6477
91 Han X G, Chen Z Q, Su R. Synthesis of minimally restrictive optimal stability-enforcing supervisors for nondeterministic discrete event systems. Syst Control Lett, 2019, 123: 33–39
92 Le S T, Wu Y H, Toyoda M. A congestion game framework for service chain composition in NFV with function benefit. Inf Sci, 2020, 514: 512–522
93 Tu Z Z, Xia X H, Zhu B. Demand-side management and control for a class of smart grids based on game theory. In: Proceedings of the 36th Chinese Control Conference, Dalian, 2017. 10662–10667
94 Lu J Q, Liu R J, Lou J G, et al. Pinning stabilization of Boolean control networks via a minimum number of controllers. IEEE Trans Cybern, 2021, 51: 373–381
95 Han X G, Yang W D, Chen X Y, et al. Detectability vverification of probabilistic Boolean networks. Inf Sci, 2021, 548: 313–327
96 Ding X Y, Li H T. Optimal control of random evolutionary Boolean games. Int J Control, 2021, 94: 144–152
97 Yan Y Y, Chen Z Q, Yue J M, et al. STP approach to model controlled automata with application to reachability analysis of DEDS. Asian J Control, 2016, 18: 2027–2036
98 Gécseg F. Composition of automata. In: Automata, languages and programming. Berlin: Springer, 1974. 351–363
99 Henzinger T A. The Theory of Hybrid Automata. Berlin: Springer, 2000
100 Yan Y Y, Chen Z Q, Yue J M. Algebraic state space approach to model and control combined automata. Front Comput Sci, 2017, 11: 874–886
101 Chen Z Q, Zhou Y R, Zhang Z P, et al. Observability analysis of combined finite automata based upon semi-tensor product of matrices approach. Trans Inst Measurement Control, 2021, 43: 717–727
102 Akutsu T, Hayashida M, Ching W K, et al. Control of Boolean networks: hardness results and algorithms for tree structured networks. J Theor Biol, 2007, 244: 670–679
103 Cheng D Z, Qi H S. Controllability and observability of Boolean control networks. Automatica, 2009, 45: 1659–1667
104 Zhao Y, Qi H S, Cheng D Z. Input-state incidence matrix of Boolean control networks and its applications. Syst Control Lett, 2010, 59: 767–774
105 Cheng D Z, Zhao Y. Identification of Boolean control networks. Automatica, 2011, 47: 702–710
106 Li R, Yang M, Chu T G. Observability conditions of Boolean control networks. Int J Robust Nonlinear Control, 2014, 24: 2711–2723
107 Cheng D Z, Qi H S, Liu T, et al. A note on observability of Boolean control networks. Syst Control Lett, 2016, 87: 76–82
108 Shi W P, Wu B, Han J. A note on the observability of temporal Boolean control network. Abstract Appl Anal, 2013, 2013: 1–9
109 Laschov D, Margaliot M, Even G. Observability of Boolean networks: a graph-theoretic approach. Automatica, 2013, 49: 2351–2362
110 Fornasini E, Valcher M E. Observability, reconstructibility and state observers of Boolean control networks. IEEE Trans Automat Contr, 2013, 58: 1390–1401
111 Zhang K Z, Zhang L J. Observability of Boolean control networks: a unified approach based on finite automata. IEEE Trans Automat Contr, 2016, 61: 2733–2738
112 Cheng D Z, Qi H S, Li Z Q. Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach. London: Springer-Verlag, 2011
113 Chaves M, Tournier L. Analysis tools for interconnected Boolean networks with biological applications. Front Physiol, 2018, 9: 586
114 Zhang K Z, Zhang L J, Xie L H. Finite automata approach to observability of switched Boolean control networks. Nonlinear Anal-Hybrid Syst, 2016, 19: 186–197

115 Wang Y Z, Zhang C H, Liu Z B. A matrix approach to graph maximum stable set and coloring problems with application to multi-agent systems. Automatica, 2012, 48: 1227–1236

116 Xu M R, Wang Y Z, Wei A R. Robust graph coloring based on the matrix semi-tensor product with application to examination timetabling. Control Theor Technol, 2014, 12: 187–197

117 Meirong X U, Wang Y Z. Conflict-free coloring problem with appliction to frequency assignment. J Shandong Univ, 2015, 45: 64–69

118 Xu M R, Wang Y Z, Jiang P. Fuzzy graph coloring via semi-tensor product method. In: Proceedings of the 34th Chinese Control Conference, Hangzhou, 2015. 973–978

119 Zhong J, Lu J Q, Huang C, et al. Finding graph minimum stable set and core via semi-tensor product approach. Neurocomputing, 2016, 174: 588–596

120 Meng M, Feng J E. A matrix approach to hypergraph stable set and coloring problems with its application to storing problem. J Appl Math, 2014, 2014: 1–9

121 Yue J M, Chen Z Q, Yan Y Y, et al. Solvability of k-track assignment problem: a graph approach. Control Theory & Appl, 2017, 34: 457–466

122 Yue J M, Yan Y Y, Chen Z Q. Matrix approach to simplification of finite state machines using semi-tensor product of matrices. Asian J Control, 2020, 22: 2061–2070

123 Wang Y B, Li Y M. Minimization of lattice multiset finite automata. J Intell Fuzzy Syst, 2018, 35: 627–637

124 Li Y M, Pedrycz W. Minimization of lattice finite automata and its application to the decomposition of lattice languages. Fuzzy Sets Syst, 2007, 158: 1423–1436

125 Solov'ev V V. Minimization of mealy finite-state machines by using the values of the output variables for state assignment. J Comput Syst Sci Int, 2017, 56: 96–104

126 Solov'ev V V. Minimization of mealy finite state machines via internal state merging. J Commun Technol Electron, 2011, 56: 207–213

127 Solov'ev V V. Minimization of Moore finite automata by internal state gluing. J Commun Technol Electron, 2010, 55: 584–592

128 Klimowicz A S, Solov'ev V V. Minimization of incompletely specified mealy finite-state machines by merging two internal states. J Comput Syst Sci Int, 2013, 52: 400–409

129 Gören S, Ferguson F J. On state reduction of incompletely specified finite state machines. Comput Electrical Eng, 2007, 33: 58–69

130 Liu D S, Huang Z P, Zhang Y M, et al. Efficient deterministic finite automata minimization based on backward depth information. Plos One, 2016, 11: e0165864