Modeling and Reachability Verification of Controlled Nondeterministic Finite-State Automata

Zipei Wang*, Xiang Ren
College of Electronic Information and Automation, Tianjin University of Science and Technology, 300222， China

*E-mail: *wangzipei0301@163.com*

Abstract

In this paper, we investigate the modeling and state reachability of controlled nondeterministic finite-state automata (NFA). The key feature of a controlled NFA is to admit a controller (also called a supervisor) to intervene the behavior of an original system. We first express the dynamics of a controlled NFA as an algebraic state-space representation in the framework of the semi-tensor product (STP) of matrices. Then, the necessary and sufficient condition for verifying state reachability of controlled NFA is presented. An explicit formula for calculating all paths of any two states is derived. Finally, we use an example to illustrate the application of the proposed theoretical results.

Keywords: Discrete event systems, nondeterministic finite-state automata, reachability, semi-tensor product of matrices

1. Introduction

Discrete event systems (DESs), also called plants, have received considerable attention within the automatic control and computer science communities for many years, see, e.g., [1], [2].

Nondeterministic plants can be viewed as a generalization of deterministic plants. It is useful when designing a system at a higher-level of abstraction so that lower-level details of system are omitted to obtain higher-level models that may be nondeterministic. It is well-known that controlled DESs modeled by controlled finite automata have more complex structures and dynamics than uncontrolled DESs. Therefore, how to model and analyze effectively the dynamics of controlled nondeterministic DESs are still an interesting topic.

In classical DESs, the reachability is a basic and important problem in the study of DESs. For instance, [3] studied the matrix expression and reachability verification of finite automata (including deterministic and nondeterministic). The approach in these two papers are based on algebraic state-space representation. In this paper, we develop a new methodology to investigate how to model controlled nondeterministic finite-state automata (NFA) using the STP of matrices.

The rest of this article is organized as follows. The second section presents some basic notations and concepts needed in this paper. In the third part, we present a matrix-based expression for the dynamics of controlled NFA. In the fourth section, we give a methodology of verifying state reachability of controlled NFA, and an explicit formula of finding all paths of any two states is also provided, if exists. In the fifth section, an example is presented to illustrate the application of the proposed approach. The sixth part summarizes the main content of this paper.

2. Preliminaries

2.1. *Notations*

In this subsection, we introduce some notations, which will be used in the sequel. $\mathbb{R}^n$ is the set of all vectors of dimension n; $|X|$ is the cardinality of set X; $\mathcal{M}_{m\times n}$ is the set of $m\times n$ matrices; $M_{(i,j)}$ is the (i,j) element of matrix M; $Col_j(M)$ is the j-th column of matrix M; $Col(M)$ is the set of all columns of matrix M; $0_n := [\underbrace{0,0,\cdots,0}_{n}]$; $1_n := [\underbrace{1,1,\cdots,1}_{n}]$; $\delta_n^0 := [\underbrace{0,0,\cdots,0}_{n}]^T$; $\delta_n^k := Col_k(I_n)$, $1\le k\le n$; $\Delta_n := \{\delta_n^1,\delta_n^2,\cdots,\delta_n^n\}$; $\tilde{\Delta}_n := \{\delta_n^0,\delta_n^1,\cdots,\delta_n^n\}$; $L\in\mathcal{M}_{m\times n}$ is a logical matrix (resp., generalised logical matrix) if $Col(L)\subseteq\Delta_m$ (resp., $Col(L)\subseteq\tilde{\Delta}_m$). We denote the set of $m\times n$ logical matrices (resp., generalised logical matrix) by $\mathcal{L}_{m\times n}$ (resp., $\tilde{\mathcal{L}}_{m\times n}$); If matrix $L\in\tilde{\mathcal{L}}_{m\times n}$, then it can be expressed as $L\in[\delta_m^{i_1},\delta_m^{i_2},\cdots,\delta_m^{i_n}]$ and it is briefly denoted as $L\in\delta_m[i_1,i_2,\cdots,i_n]$, where $i_k\in\{0,1,\cdots,m\}$, $1\le k\le n$.

2.2. *Semi-tensor product (STP) of matrices*

In this subsection, we give some necessary basic knowledge of the STP of matrices used in the paper.

Definition 2.1 ([4]): Let $A\in\mathcal{M}_{m\times n}$, $B\in\mathcal{M}_{p\times q}$. The STP of A and B is defined as

$$A\ltimes B=(A\otimes I_{t/n})(B\otimes I_{t/p}), \tag{1}$$

where t denotes the least common multiple of n and p, i.e., $t=lcm(n,p)$; $\otimes$ is the Kronecker product.

Remark 2.1: When $n=p$, $A\ltimes B=AB$. Hence, the STP is a generalization of the standard matrix product. Throughout this paper the matrix product is assumed to be the STP. We mostly omit the symbol "$\ltimes$" hereinafter.

Definition 2.2 ([4]): A swap matrix $W_{[m,n]}$ is a $mn\times mn$ logical matrix, which is defined as

$$\begin{aligned} W_{[m,n]} = \delta_{mn}[&1,m+1,2m+1,\cdots,(n-1)m+1,\\ &2,m+2,2m+2,\cdots,(n-1)m+2, \\ &\cdots,m,2m,3m,\cdots,nm]. \end{aligned} \quad (2)$$

Lemma 2.1([4]): Let $X\in\mathbb{R}^m$ and $Y\in\mathbb{R}^n$ be two column vectors. Then

$$W_{[m,n]}XY=YX,\ W_{[n,m]}YX=XY. \quad (3)$$

2.3. *System model*

In this subsection we recall the formalism used in the paper.

A nondeterministic DES is modeled as a NFA $G=(X,\Sigma,\delta,X_0,X_m)$, where X is the finite set of states, Σ is the finite set of events called alphabet or input symbols, $X_0\subseteq X$ is the set of initial states, $X_m\subseteq X$ is the set of marked states (or accepted states), $\delta: X\times\Sigma\rightarrow 2^X$ is the partial transition function(2^X denotes the power set of X), which describes the system dynamics: given states $x,y\in X$ and an event $\sigma\in\Sigma$, $y\in\delta(x,\sigma)$ means the execution of σ from state x takes the system to state y. Note that $\delta(x,\sigma)$ is undefined when the event σ cannot be executed from the state x. $\delta(x,\sigma)!$ denotes $\delta(x,\sigma)$ is well-defined. Obviously, the transition function can be extended to $\delta: X\times\Sigma^*\rightarrow 2^x$ in terms of $\delta(x,e):=\delta(\delta(\cdots\delta(\delta(x,e_{j_1}),e_{j_2})\cdots),e_{jt})$,where $e=e_{j_1}e_{j_2}\cdots e_{j_t}\in\Sigma^*$, Σ^* denotes the set of finite strings on the alphabet Σ, including the empty string ϵ. The objective of this paper is to investigate the controlled NFA. In this regard, the event set Σ can be partitioned into two disjoint subsets, i.e., $\Sigma=\Sigma_c\bigcup\Sigma_{uc}$, where Σ_c denotes the set of controllable events, Σ_{uc} denotes the set of uncontrollable events. We here assume that all events in Σ are observable.

In general, we wish to adjoin a supervisor or a controller S to interact with G in a feedback manner. More precisely, the transition function of G can be controlled by S in the sense that the controllable events of G can be dynamically enabled or disabled by S after each transition. Formally, a *state-feedback supervisor*, denoted by S, is a function $S: X\rightarrow 2^{\Sigma_c}$ that determines the set of events $S(x)\subseteq\Sigma_c$ to be disabled at each state $x\in X$, while events not belonging to the set $S(x)$ remain enabled at state x. The *controlled system* (or called *supervised system*) consisting of G and S, denoted by S/G, is another nondeterministic finite automation given as

$$S/G=(X,\Sigma,\delta_S,X_0,X_m), \quad (4)$$

where X, Σ, X_0 and X_m are as defined above, δ_S is the partial transition function of S/G, i.e.,

$$\delta_S(x,\sigma)=\begin{cases}\delta(x,\sigma), if\ \delta(x,\sigma)!\ and\ \sigma\notin S(x)\\ \text{undefined, otherwise}\end{cases} \quad (5)$$

We use the notation $\mathrm{H}(x)$ to denote the set of feasible events of G at state x. Thus supervisor S is called permissible if for all $x\in X$, $S(x)\subseteq \mathrm{H}(x)\bigcap\Sigma_c$. Note that it is not difficult to see that NFA can be viewed as special case of controlled NFA with $S(x)=\varnothing$ for all $x\in X$. The controlled NFA S/G is depicted in Fig. 1.

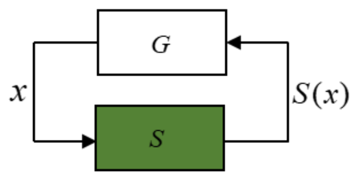

Fig. 1. The controlled NFA S/G

3. Matrix-Based Expression of Controlled NFA

In order to obtain the matrix expression of the dynamics of controlled NFA, let us first give an equivalent description of the controlled system (4). For the NFA G, we define a control pattern as a Boolean function: $\gamma:\Sigma_c\rightarrow\{0,1\}$ and we use the notation $\Gamma=\{0,1\}^{\Sigma_c}$ to denote the set of all Boolean functions on Σ_c. $\gamma\in\Gamma$ is interpreted as follows: for any $\sigma\in\Sigma_c$, $\gamma(\sigma)=1$ means that the control pattern γ allows σ to happen, while $\gamma(\sigma)=0$ means that the control pattern γ refuse σ to happen. Note that it is convenient to extend each $\gamma\in\Gamma$ to a function $\gamma:\Sigma\rightarrow\{0,1\}$ by defining $\gamma(\sigma)=1$ for each uncontrollable event $\sigma\in\Sigma_{uc}$. Further, we define a partial function of the form $f: X\rightarrow\Gamma$, called the *state-feedback control function* or *state-feedback* for short, that maps each state x in X into control pattern γ, i.e., $\gamma(\sigma)=f(x)(\sigma)$ .Thus the controlled NFA consisting of the state-feedback f, control pattern γ and NFA G is described as

$$G_\gamma^f=(X,\Sigma,\delta_\gamma^f,X_0,X_m), \quad (6)$$

where X, Σ, X_0 and X_m are as defined above; δ_γ^f denotes the partial transition function of G_γ^f, which is defined as

$$\delta_\gamma^f(x,\sigma)=\begin{cases}\delta(x,\sigma), \text{ if } \delta(x,\sigma)! \text{ and } f(x)(\sigma)=1\\ \text{undefined, otherwise.}\end{cases} \quad (7)$$

Similarly, the state-feedback f is called permissible if for all $x\in X$, $\sigma\in \mathrm{H}(x)\cap\Sigma_{uc}$, we have $f(x)(\sigma)=1$. There is a bijective correspondence between the state-feedback f and the state-feedback supervisor S, i.e., $\forall x\in X$ and $\forall\sigma\in\mathrm{H}(x)\cap\Sigma_c$, $f(x)(\sigma)=1$ (resp., $f(x)(\sigma)=0$) if and only if $\sigma\notin S(x)$ (resp., $\sigma\in S(x)$). In this regard, we will use for simplicity the controlled system (6) instead of system (4) to present our results. Our objective, in this section, is to model the controlled NFA in the framework of the STP of matrices.

Let us consider the controlled NFA(6), we assume that the set of states is $X=\{x_1,x_2,\cdots,x_n\}$, and the set of events is $\Sigma=\{e_1,e_2,\cdots,e_m\}$. To obtain the dynamics of (6), identifying $x_i \sim \delta_n^i (1\le i\le n)$, $e_j \sim \delta_m^j (1\le j\le m)$, we call δ_n^i and δ_m^j the vector forms of x_i and e_j, respectively. Thus the state set X and event set Σ can be identified with Δ_n and Δ_m, respectively, where $\Delta_n=\{\delta_n^1,\delta_n^2,\cdots,\delta_n^n\}$ and $\Delta_m=\{\delta_m^1,\delta_m^2,\cdots,\delta_m^m\}$. Therefore, $x_s\in\delta_c^f(x_t,e_j)$ can be expressed equivalently as $\delta_n^s\in\delta_c^f(\delta_n^t,\delta_m^j)$, where δ_c^f represents the partial transition function of G_γ^f defined in (7). For brevity, we use the notation r_{ij} to denote the state-feedback $f(x_i)(e_j)$, i.e., $r_{ij}=f(x_i)(e_j)$ where $x_i\in X$ and $e_j\in\Sigma_c$ are the state and controllable event of controlled NFA(6), respectively.

For event $e_j(1\le j\le m)$, we define a $n\times n$ matrix F_j as

$$F_{j(s,t)}=\begin{cases}1, \text{ if } \delta_n^s\in\delta_\gamma^f(\delta_n^t,\delta_m^j)\wedge\delta_m^j\sim e_j\in\Sigma_{uc}\\ r_{tj}, \text{ if } \delta_n^s\in\delta(\delta_n^t,\delta_m^j)\wedge\delta_m^j\sim e_j\in\Sigma_c\\ 0, \text{ otherwise,}\end{cases} \quad (8)$$

where F_j is called transition structure matrix *w.r.t.* event e_j. Thus, the *transition structure matrix* (TSM) of controlled NFA (6) is defined as

$$F=[F_1,F_2,\cdots,F_m], \quad (9)$$

where F is a $n\times mn$ symbol matrix.

Based on the above representations and the STP of matrices, we can obtain the following result on the matrix expression of dynamics of controlled NFA (6).

Theorem 3.1: Given a controlled NFA (6), the dynamics of (6) can be equivalently described as

$$x(t+1)=Fu(t)x(t), \quad (10)$$

where F is the TSM of system (6), which is defined in (9); $x(t)=(x_1(t),x_2(t),\cdots,x_n(t))^T$ is the vector form of state at step t, $x_i(t)$ denotes the number of different paths from the set of initial states to the state $x_i=\delta_n^i$ with a feasible event string of length $t-1$; $u(t)\in\Delta_m$ is vector form of event at step t.

Let us assume (10) holds when $t=p-1$, i.e., $x(p)=Fu(p-1)x(p-1)$. And let $u(p)=\delta_m^k$ for any $1\le k\le m$. Then

$$Fu(p)x(p)=F_k(x_1(p),\cdots,x_n(p))^T$$
$$=(\sum_{i=1}^n F_{k(1,i)}x_i(p),\cdots,\sum_{i=1}^n F_{k(n,i)}x_i(p))^T.$$

Using (8), one has

$$\sum_{i=1}^n F_{k(j,i)}x_i(p)=\sum_{i\in\Lambda_j}F_{k(j,i)}x_i(p), 1\le j\le n,$$

where $\Lambda_j=\{i\mid 1\le i\le n,\delta_n^j\in\delta_\gamma^f(\delta_n^i,\delta_m^k)\}$. From the above assumption, we know that $x_i(p)$ stands for the number of different paths from the initial state $x(1)=\delta_n^{i_0}$ to state δ_n^i with a feasible string of length $p-1$, then $\sum_{i\in\Lambda_j}F_{k(j,i)}x_i(p)$ represents the sum of the number of different paths by which the initial state set $x(1)=\sum_{\lambda=1}^{l}\delta_n^{i_\lambda}$ can reach state δ_n^i with a feasible string of length $p-1$, and δ_n^i can reach δ_n^j with the event $u(p)=\delta_m^k$. Consequently, we define $x_j(p+1):=\sum_{i\in\Lambda_j}F_{k(j,i)}x_i(p)$, which implies $x(p+1)=Fu(p)x(p)$. By mathematical induction, the proof is completed.

By Theorem 3.1, we know readily that verifying whether or not any two states of controlled NFA (6) are reachable can be determined by both the transition function δ defined in G and the state-feedback f. In particular, when the state-feedback f is known, the matrix F becomes a constant matrix. More concretely, F is a generalised logical matrix. In this case, we replace F with F_c. Therefore, by Theorem 3.1, we have the following result.

Corollary 3.1: Given a controlled NFA (6) in which the state-feedback f is known, the dynamics of (6) can be equivalently described by the following equation

$$x(t+1)=F_cu(t)x(t), \quad (11)$$

where $x(t)$ and $u(t)$ have the same interpretation as in Theorem 3.1; F_c is the TSM of (6), which is represented as

$$F_c = \left[F_1^c, F_2^c, \cdots, F_m^c \right] \in \mathcal{L}_{n \times mn}, \quad (12)$$

where the $n \times n$ matrix F_j^c is defined as

$$F_{j(s,t)}^c = \begin{cases} 1, & \text{if } \delta_n^s \in \delta_\gamma^f(\delta_n^t, \delta_m^j) \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Remark 3.1: It should be pointed out that our approach are also suitable for the controlled deterministic DESs in the sense that deterministic DESs can be viewed as special case of nondeterministic DESs. In particular, when the controlled DES (6) is deterministic, i.e., $|X_0| = 1$ and $|\delta(x, \sigma)| \le 1$ for all $x \in X$ and $\sigma \in \Sigma$, the state that is reachable from initial state with a feasible event string $e = \sum_{k=1}^{t-1} e_{jk} \in \Sigma^*$ is unique, which means that there exists only one $1 \le i \le n$ such that $x_i(t) = 1$ in (11). Namely, $x(t) \in \Delta_n$. In contrast, if there is no a state that is reachable from initial state with $e = \sum_{k=1}^{t-1} e_{jk} \in \Sigma^*$, then we have $x(t) = \delta_n^0$.

4. Reachability of Controlled NFA

4.1. *Reachability verification*

We, in this subsection, verify the reachability property of controlled NFA (6) by means of equation (11). We first give the following lemma, which is crucial to find all paths from the set of initial states to any target state for the controlled NFA (6).

Lemma 4.1 ([5]): Let $\ltimes_{i=1}^t \delta_{m_i}^{j_i} = \delta_{m_1 \times m_2 \cdots m_t}^l$, then the formulas $\delta_{m_i}^{j_i} = S_i \ltimes \delta_{m_1 \times m_2 \cdots m_t}^l$, $i = 1, 2, \cdots, t$ hold, where

$$\begin{cases} S_1 = I_{m_1} \otimes 1_{m_2 \times \cdots \times m_t}, \\ S_2 = [\underbrace{I_{m_2} \otimes 1_{m_3 \times \cdots \times m_t}, \cdots, I_{m_2} \otimes 1_{m_3 \times \cdots \times m_t}}_{m_2}], \\ \quad \vdots \\ S_{t-1} = [\underbrace{I_{m_{t-1}} \otimes 1_{m_t}, \cdots, I_{m_{t-1}} \otimes 1_{m_t}}_{m_2 \times \cdots \times m_{t-1}}], \\ S_t = [\underbrace{I_{m_t}, \cdots, I_{m_t}}_{m_2 \times \cdots \times m_t}]. \end{cases} \quad (14)$$

To present the main results on reachability of the controlled NFA (6), we for brevity need to introduce the following notation: let $\alpha = (a_1, a_2, \cdots, a_n)^T$ be a nonnegative column vector of dimension n, we define $\Xi(\alpha) := \{\delta_n^k \mid a_k \ne 0, 1 \le k \le n\}$. For instance, let $\alpha = (1, 2, 0, 3)^T$, then $\Xi(\alpha) = \{\delta_4^1, \delta_4^2, \delta_4^4\}$.

Theorem 4.1: Given a controlled NFA (6) with its dynamics (11), let $X_0 = \{\delta_n^{i_1}, \delta_n^{i_2}, \cdots, \delta_n^{i_l}\}$ and $x^* = \delta_n^q$ be the set of initial states and target state of (6), respectively. Then

1) $x^* = \delta_n^q$ is reachable from X_0 in t steps if and only if there exists a positive integer $k (1 \le k \le m^t)$ such that

$$\delta_n^q \in \Xi(Col_k((F_c W_{[n,m]})^t x(1))), \quad (15)$$

where $x(1)$ represents the vector form of the initial state set X_0, i.e., $x(1) = \sum_{\lambda=1}^{l} \delta_n^{i_\lambda}$.

2) Assume that $L_t^q(X_0)$ denotes the set consisting of all event strings of length t by which the initial state set X_0 can reach $x^* = \delta_n^q$ in t steps, then

$$L_t^q(X_0) = \{e = e_{k_1} e_{k_1} \cdots e_{k_t} \in \Sigma^* \mid \text{there is } k \text{ of satisfying (15) such that } \delta_{m^t}^k = e_{k_1} e_{k_1} \cdots e_{k_t}\}, \quad (16)$$

where the feasible event string of $e = e_{k_1} e_{k_1} \cdots e_{k_t}$ can be easily obtained from the formula $\delta_{m^t}^k = e_{k_1} e_{k_1} \cdots e_{k_t}$ in terms of Lemma 4.1.

Proof: we here omit the proof of Lemma 4.1.

From (11), we have

$$\begin{aligned} x(t+1) &= F_c u(t) x(t) \\ &= F_c W_{[n,m]} x(t) u(t) \\ &= (F_c W_{[n,m]})^2 x(t-1) u(t-1) u(t) \\ &\vdots \\ &= (F_c W_{[n,m]})^t x(1) u(1) u(2) \cdots u(t) \\ &= ((F_c W_{[n,m]})^t x(1)) \ltimes_{j=1}^t u(j). \end{aligned} \quad (17)$$

Note that the following corollary gives a criterion to verify whether any two states of a controlled NFA are reachable or not. Also, an effective approach of finding all paths of any two reachable states is provided.

Corollary 4.1: Given a controlled NFA (6) with its dynamics (11), let $x_p = \delta_n^p$ and $x^q = \delta_n^q$ be any two states of (6). Then

1) $x^q = \delta_n^q$ is reachable from $x_p = \delta_n^p$ in t steps if and only if there exists a positive integer $k (1 \le k \le m^t)$ such that

$$\delta_n^q \in \Xi(Col_k((F_c W_{[n,m]})^t \delta_n^p)). \quad (18)$$

2) Let $L_t^q(p)$ be the set consisting of all event strings of length t by which $x_p = \delta_n^p$ can reach $x^q = \delta_n^q$ in t steps, then

$$L_t^q(p) = \{e = e_{k_1} e_{k_1} \cdots e_{k_t} \in \Sigma^* \mid \text{there exist } k \text{ of satisfying (18) such that } \delta_{m^t}^k = e_{k_1} e_{k_1} \cdots e_{k_t}\}, \quad (19)$$

where the event string $e = e_{k_1} e_{k_1} \cdots e_{k_t}$ can be obtained from $\delta_{m^t}^k = e_{k_1} e_{k_1} \cdots e_{k_t}$ in terms of Lemma 4.1.

Proof: Obviously, here we omit its proof.

Remark 4.1: From Corollary 4.1, we readily know that if the controlled NFA (6) is deterministic, then the number of different paths from state $x_p = \delta_n^p$ to $x^q = \delta_n^q$ in t steps coincide with the number of k satisfying (18). In contrast, when the controlled NFA (6) is nondeterministic, we for convenience assume that $\Upsilon = (\gamma_1, \gamma_2, \cdots, \gamma_n)^T := \sum_{j=1}^{m^t} Col_j((F_c W_{[n,m]})^t \delta_n^p)$. Then the number of different paths from $x_p = \delta_n^p$ to $x^q = \delta_n^q$ in t steps is equal to γ_q.

4.2. *Comparison with the existing approaches*

There are several papers addressing modeling and reachability analysis in DESs in terms of the algebraic state-space approaches, see, e.g., [3] and [6]. Among them, [3] investigated matrix expression and reachability verification of finite automata in which all events are controllable and the cardinality of the set of initial states equals 1. We know that the controlled DESs have more complex structures and dynamics than the uncontrolled DESs as an external control input is allowed to intervene the behaviors of original systems. Again, they did not consider to adjoin a supervisor to interact with an original system. These restrictions significantly simplify the behavior of the system. Obviously, their approaches are different completely from the approach using in this paper since we do not make the above-mentioned these restrictions.

[6] is most closely related to our work, but there are some fatal errors therein. Concretely, the authors asserted that they presented a sufficient and necessary condition to verify whether or not any two states of a controlled DES are reachable (see, e.g., Theorem 2 in [6]). Also, an algorithm was designed to find all paths from one state to another one if they are reachable (see, e.g., Algorithm 1). These results were obtained from Corollary 1 in [6]. It should be pointed out that, however, Corollary 1 addressed the problem of which states are reachable from a given state and a given input string of length t for a controlled DES, which means a contradiction with the previous assertion. Furthermore, the state-feedback control specification was defined as $f: X \to \Gamma$ in [6]. In this regard, $r_j = f(\tilde{F}^t$ 粡 $\delta_n^i \;\; u(j))(\sigma_{j+1})$ is undefined in (11) of [6] since $\tilde{F}^t$ 粡 $\delta_n^i \;\; u(j)$ is not necessarily the vector form of a state for a controlled nondeterministic system, which means that equation (11) in [6] is incorrect. We refer the reader to [6] for the interpretations of these notations. As a part of our work, these errors presented in [6] were corrected and the interesting issues mentioned in [6] were discussed systematically in a more generalized DES.

5. Illustrative Example

In this section, we use an example to illustrate the proposed results.

Example 5.1 : Let us consider the uncontrolled NFA, $G = (X, \Sigma, \delta, X_0, X_m)$ depicted in Fig. 2, where $\Sigma_c = \{e_2, e_3\}$, $\Sigma_{uc} = \{e_1, e_4\}$, $X_0 = \{e_1, e_4\}$, $X_m = \{x_5\}$. The state-feedback supervisor S is given as $S(x_3) = \{e_2, e_3\}$.

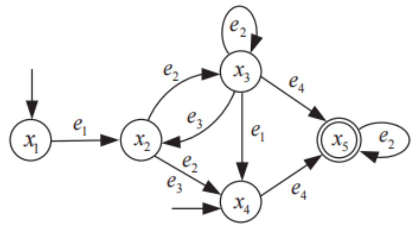

Fig. 2. An uncontrolled NFA.

We now verify the state reachability of the controlled system S/G consisting of the uncontrolled NFA G shown in Fig.2 and the state-feedback supervisor S.

By Corollary 3.1, the dynamics of the uncontrolled system S/G can be expressed as

$$x(t+1) = F_c u(t) x(t), \quad (20)$$

where $x(1) = \sum_{j=1,4} \delta_n^j$,

$$F_c = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&0 \end{bmatrix}.$$

When $t = 1$, we have

$$Col_1(F_c W_{[5,4]} x(1)) = (0,1,0,0,0)^T,$$
$$Col_4(F_c W_{[5,4]} x(1)) = (0,0,0,0,1)^T,$$
$$Col_j(F_c W_{[5,4]} x(1)) = (0,0,0,0,0)^T, \; j = 2,3.$$

By applying Theorem 4.1, states x_2 and x_5 are reachable from the set of initial states X_0 in 1 step, and the corresponding event strings of length 1 are $e = e_1$ and $e = e_4$, respectively. Specifically, we have $x_1 \xrightarrow{e_1} x_2$; $x_4 \xrightarrow{e_4} x_5$.

When $t=2$, we have

$$Col_2((F_cW_{[5,4]})^2 x(1)) = (0,0,1,1,0)^T,$$
$$Col_3((F_cW_{[5,4]})^2 x(1)) = (0,0,0,1,0)^T,$$
$$Col_{14}((F_cW_{[5,4]})^2 x(1)) = (0,0,0,0,1)^T.$$

with all the other columns equal to δ_5^0. By Theorem 4.1, states x_3, x_4 and x_5 are reachable from X_0 in 2 steps, and the corresponding event string of length 2 are $e=\delta_{16}^2=\delta_4^1\delta_4^2$, $e=\delta_{16}^2=\delta_4^1\delta_4^2$ or $e=\delta_{16}^3=\delta_4^1\delta_4^3$, and $e=\delta_{16}^{14}=\delta_4^4\delta_4^2$, respectively. Specifically, we have $x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_2} x_3$; $x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_2} x_4$ or $x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_3} x_4$; and $x_4 \xrightarrow{e_4} x_5 \xrightarrow{e_2} x_5$.

When $t=3$, we have

$$Col_5((F_cW_{[5,4]})^3 x(1)) = (0,0,0,1,0)^T,$$
$$Col_8((F_cW_{[5,4]})^3 x(1)) = (0,0,0,0,2)^T,$$
$$Col_{12}((F_cW_{[5,4]})^3 x(1)) = (0,0,0,0,1)^T,$$
$$Col_{54}((F_cW_{[5,4]})^3 x(1)) = (0,0,0,0,1)^T$$

with all the other columns equal to δ_5^0. Using Theorem 4.1, states x_4 and x_5 are reachable from X_0 in 3 steps, and the corresponding event string of length 3 are $e=\delta_{64}^5=\delta_4^1\delta_4^2\delta_4^1$, $e=\delta_{64}^8=\delta_4^1\delta_4^2\delta_4^4$ or $e=\delta_{64}^{12}=\delta_4^1\delta_4^3\delta_4^4$, and $e=\delta_{64}^{54}=\delta_4^4\delta_4^2\delta_4^2$, respectively. Specifically, we have

$$x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_2} x_3 \xrightarrow{e_1} x_4;\quad x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_2} x_3 \xrightarrow{e_1} x_5;$$
$$x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_2} x_4 \xrightarrow{e_4} x_5,\quad x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_3} x_4 \xrightarrow{e_4} x_5;$$
$$x_4 \xrightarrow{e_4} x_5 \xrightarrow{e_2} x_5 \xrightarrow{e_2} x_5.$$

The cases of $t \geq 4$ are similar to $t=1,2,3$, the details are omitted here for space limitations.

6. Conclusion

In this paper, we proposed a new framework that is matrix-based form to model the dynamics of controlled NFA. Using it, we investigated the verifications of reachability property of controlled NFA. Also, the criteria of verifying this property was presented in terms of the methodology described in this paper.

Future work will concentrate on investigating the problem of synthesizing property-enforcing supervisor for controlled nondeterministic DESs by means of the proposed theoretical framework.

Acknowledgements

This work was supported in part by the Natural Science Foundation of China under Grant 61903274.

References

1. R. Jacob, J.J. Lesage, and J.M. Faure. Overview of discrete event systems opacity: Models, validation, and quantification. Annual Reviews in Control, 41:135–146, 2016.
2. C.G. Cassandras and S. Lafortune. Introduction to disctete event systems. Springer, Switzerland AG, 3rd Edition, 2021.
3. X. R. Xu and Y. G. Hong, “Matrix expression and reachability analysis of finite automata,” J. Control Theory Appl., vol. 10, no. 2, pp. 210–215, 2012.
4. D. Z. Cheng, H. S. Qi, and Y. Zhao, An Introduction to Semi-tensor Product of Matrices and Its Applications. Singapore: World Scientific, 2012.
5. J. J. Wang, X. G. Han, Z. Q. Chen, et al, “Model matching of input-output asynchronous sequential machines based on the semi-tensor product of matrices,” Future Generation Computer Systems: The International Journal of eScience, vol. 83, pp. 468-475, 2018.
6. Y. Y. Yan, Z. Q. Chen, J. Y. Yue, et al, “STP approach to model controlled automata with application to reachability analysis of DEDS,” Asian J. Control, vol. 18, no. 6, pp. 1–10, Nov. 2016.

Authors Introduction

Ms. Zipei Wang

She received the B.S. degree from College of Electronic Information and Automation, Tianjin University of Science and Technology, China in 2023. She is currently a Master course student in Tianjin University of Science and Technology.

Mr. Xiang Ren

He received his B.S. degree from College of Electronic Information and Automation, Tianjin University of Science and Technology, China in 2021. He is currently a Master course student in Tianjin University of Science and Technology.