2010 8th IEEE International Conference on
Control and Automation
Xiamen, China, June 9-11, 2010

Computation for Supremal Simulation-Based Controllable Subautomata

Yajuan Sun, Hai Lin, Fuchun Liu, Ben M. Chen

Abstract— **Bisimulation relation as a well known equivalence relation has been successfully applied to computer science and control theory. In our previous work, we proposed the existence of bisimilarity supervisor by introducing simulation-based controllability. As a continuation, this paper deals with the computation for the supremal simulation-based controllable subautomata with respect to given specifications by lattice theory. In order to achieve the supremal solution, two monotone operators, namely simulation operator and controllable operator, are built upon the established complete lattice, and then we set up the inequalities, whose solutions are simulation-based controllable state pairs. In particular, a sufficient condition is provided to guarantee the existence of supremal simulation-based controllable subautomata. Furthermore, an algorithm is presented for the computation of such subautomata.**

I. INTRODUCTION

Bisimulation relation was introduced in [1] as a behavioral equivalence relationship between dynamical systems. It has been successfully applied to computer science and control theory, see e.g., [2], [5]. Bisimulation relation is stronger than language equivalence because the languages generated by two bisimilar systems are equal to each other, but the systems possessing the same language might not be bisimilar. Therefore, the works on bisimilarity control, which aims to achieve the bisimulation relation between controlled system and specification, have attracted lots of attentions these years.

Komenda and Schuppen [3] [4] proposed a coalgebra to generalize the bisimulation relation to control discrete event systems (DESs) under partial observation and decentralized supervisory control. In [5] [7] [6], controller synthesis is investigated for the abstracted system which is bisimilar to the original system in the form of linear system, nonlinear system or hybrid system respectively. In Zhou's work [11] [12], a small model theorem is established to show that the supervisor which is to execute the control action to ensure the bisimulation relation exists if and only if it exists over the power set of Cartesian product of system and specification state spaces.

In our previous work, we studied the bisimilarity control of DESs from a different aspect. Plant and specification are described as nondeterministic automata, and a supervisor is to execute the control action so that the controlled system is bisimilar to the specification. We extended the language controllability to simulation-based controllability, which is under the framework of automata rather than languages. This simulation-based controllability is much stronger than the language controllability [8] because simulation-based controllability of specification implies the controllability of its generated language, but the reverse does not hold. Furthermore, we showed that the supervisor exists if and only if the specification is simulation-based controllable. However, in most situations, the given specification is usually not simulation-based controllable. Then, an interesting question arose naturally is whether there exists a sub-specification that is simulation-based controllable. If so, can one get an optimal sub-specification in the sense of maximum permissive. To deal with this problem, we will investigate the computation of supremal simulation-based controllable subautomata with respect to given specification.

In this paper, we will reply on lattice theory to solve this problem. First, we establish a complete lattice, following two operators: simulation operator and controllable operator, whose output should meet the simulation condition and the controllable condition separately. After constructing these two operators, an iterative algorithm is provided to calculate the supremal simulation state pairs by applying simulation operator. Meanwhile, the supremal controllable state pairs can be calculated by the controllable operator. Moreover, we can obtain the simulation-based controllable state pairs by combing the inequalities described by these two operators. Then, a sufficient condition is proposed to guarantee the existence of the supremal simulation-based controllable subautomta according to the lattice theory [9]. Finally, an algorithm is provided for computing such supremal simulation-based controllable subautomata.

The rest of the paper is organized as follows. Section 2 gives notation and preliminaries. Section 3 studies method of computing the supremal simulation-based controllable subautomata. An illustrative example is provided in Section 4. The paper concludes with section 5.

II. PRELIMINARIES

In this section, some preliminaries concerning automata, bisimulation relation and lattice theory are introduced.

Yajuan Sun, Hai Lin and Ben M. Chen are with Electrical and Computer Engineering Dept., National University Of Singapore, 117576, Singapore (email: {sunyajuan, elelh, bmchen}@nus.edu.sg)

Fuchun Liu is with Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China (email: liufch8@gmail.com)

A. Discrete Event System

A DES is modelled as an automaton $G = (X, \Sigma, x_0, \alpha, X_m)$, where X is the set of states, Σ is finite set of events, $\alpha : X \times \Sigma \to 2^X$ is the transition function, x_0 is the initial state, $X_m \subseteq X$ is the set of marked states.

The event set can be partition into $\Sigma = \Sigma_{uc} \cup \Sigma_c$, where Σ_{uc} is the set of uncontrollable events and Σ_c is the controllable event set. Σ^* denotes the set of all finite strings over Σ, including the empty string ϵ. The closed language generated by G is defined as L(G) = $\{s \in \Sigma^* \mid \alpha(x_0, s)$ is defined $\}$. Besides, $\Gamma : X \to 2^\Sigma$ is the active function; $\Gamma(x)$ is the active event set of G at state x.

Next, we will introduce the accessible operator [8], which is used to remove the states, which are not accessible from the initial state.

Definition 1: Given an automaton $G = (X, \Sigma, x_0, \alpha, X_m)$, the accessible operator on G is defined as:

$$Ac(G) = (X_{ac}, \Sigma, x_0, \alpha_{ac}, X_{acm}),$$

where $X_{ac} = \{x \in X \mid x \in \alpha_{ac}(x_0, s)$, where $s \in \Sigma^*$ $\}$, $X_{acm} = X_m \cap X_{ac}$, $\alpha_{ac} = \alpha \mid X_{ac} \times \Sigma \to X_{ac}$.

The notation $\alpha \mid X_{ac} \times \Sigma \to X_{ac}$ means that α is restricted to the smaller domain of the accessible states X_{ac}. Besides, the states which are not accessible from the initial state, are not so practical in the real application. Thus, we can remove these states by the accessible operator.

Then, we use bisimulation relation [12] to describe the equivalence between automata as follows.

Definition 2: Let $G_1 = (X_1, \Sigma, x_{01}, \alpha_1, X_{m1})$ and $G_2 = (X_2, \Sigma, x_{02}, \alpha_2, X_{m2})$ be two automata. G_1 is said to be simulated by G_2, denoted by $G_1 \subseteq_\phi G_2$, if there is a binary relation $\phi \subseteq X_1 \times X_2$ such that $(x_{01}, x_{02}) \in \phi$ and for each $(x_1, x_2) \in \phi$,

(1) $x_1^{'} \in \alpha_1(x_1, \sigma)$, where $\sigma \in \Sigma \Rightarrow \exists x_2^{'} \in \alpha_2(x_2, \sigma)$ such that $(x_1^{'}, x_2^{'}) \in \phi$.

(2) $x_1 \in X_{m1}$, then $x_2 \in X_{m2}$.

If $G_1 \subseteq_\phi G_2$, $G_2 \subseteq_\phi G_1$, and ϕ is symmetric, we have ϕ is a bisimulation relation between G_1 and G_2, denoted by $G_1 \simeq_\phi G_2$. Moreover, $G_1 \simeq_\phi G_2$ implies L(G_1) = L(G_2), however, L(G_1) = L(G_2) may not imply $G_1 \simeq_\phi G_2$.

In our earlier work [10], we extended classical language controllability to simulation-based controllability, which can guarantee the existence of supervisor such that there is a bisimulation relation between controlled system and specification.

Definition 3: Given a plant $G = (X, \Sigma, x_0, \alpha, X_m)$ and a specification $R = (Q, \Sigma, q_0, \delta, Q_m)$, R is simulation-based controllable with respect to G and Σ_{uc}, if it satisfies:

(1) (Simulation Condition) There is a simulation relation ϕ such that R $\subseteq_\phi$ G.

(2) (Controllable Condition) ($\forall$ s $\in L(R)$)($\forall$ q $\in \delta(q_0, s)$)($\forall$ $\sigma \in \Sigma_{uc}$)$[s\sigma \in L(G) \Rightarrow \delta(q, \sigma) \neq \emptyset]$.

The simulation-based controllability shows that, R is simulated by G and for any uncontrollable event σ, if σ is defined at a certain state of G reachable from the initial state x_0 along string s, then σ is also defined at all states of R which are reachable from the initial state x_0 along s [10].

Moreover, simulation-based controllability implies language controllability, but the reverse does not hold.

B. Lattice Theory

Definition 4: Given Y $\subseteq$ X, x $\in$ X is said to be the supremal of Y, if

- (upper bound): $\forall$ y $\in$ Y: y $\leq$ x and
- (least upper bound): $\forall$ z $\in$ X : [$\forall$ y $\in$ Y : y $\leq$ z] $\Rightarrow$ [x $\leq$ z].

The notations supY and $\sqcup Y$ are used to denote the supremal of Y.

Given Y $\subseteq$ X, x $\in$ X is said to be the infimal of Y, if

- (lower bound): $\forall$ y $\in$ Y: x $\leq$ y and
- (greatest lower bound)$\forall$ z $\in$ X : [$\forall$ y $\in$ Y : z $\leq$ y] $\Rightarrow$ [z $\leq$ x].

The notations infY and $\sqcap Y$ are used to denote the infimal of Y.

Definition 5: The pair (X, $\leq$), where X is a set and $\leq$ is a partial order over X, is called a partially ordered set or a poset. The poset (X, $\leq$) is said to be a lattice if supY, infY $\in$ X for any finite Y. If supY, infY $\in$ X for arbitrary Y $\subseteq$ X, then (X, $\leq$) is called a complete lattice.

A poset may be a lattice, but it may have a set Y of infinite size for which infY or supY may not exist. However, infY and supY exist for any $Y \subseteq X$ on a complete lattice.

Moreover, monotone functions and disjunctive functions [9] are defined over a complete lattice $(X, \leq)$.

Definition 6: A function f : $X \to X$ is said to be monotone if for any x, y $\in$ X : [x $\leq$ y] $\Rightarrow$ [f(x) $\leq$ f(y)].

Definition 7: A function f : $X \to X$ is said to be disjunctive if for any Y $\subseteq$ X : f($\sqcup_{y \in Y} Y$) = $\sqcup_{y \in Y} f(y)$.

Furthermore, the following lemmas are introduced to obtain the fixed point of the system of inequalities.

Lemma 1: Consider the system of inequalities $\{f_i(x) \leq g_i(x)\}_{i \leq n}$ over a compete lattice (X, $\leq$). Let Y = $\{y \in X \mid \forall i \leq n : f_i(y) \leq g_i(y)\}$ be the set of all solutions of the system of inequalities and $Y_1 = \{y \in X \mid h_1(y) = y\}$ be the set of all fixed points of h_1, where $h_1 = \sqcap_{i \leq n} f_i^\perp(g_i(y))$ and $f_i^\perp(g_i(y))$ is the supremal solution of $f_i(x) \leq g_i(x)$.

If f_i is disjunctive and g_i is monotone, then supY $\in$ Y, $supY_1 \in Y_1$, and supY = $supY_1$.

Lemma 2: Consider the inequalities $\{f_i(x) \leq g_i(x)\}_{i \leq n}, Y = \{y \in X \mid \forall i \leq n : f_i(y) \leq g_i(y)\}$.

If f_i is disjunctive and g_i is monotone. Iterative computation $y_0 = supX$, $\forall k \geq 0, y_{k+1} = h_1(y_k)$ until $y_{m+1} = y_m = supY$.

III. SUPREMAL SIMULATION-BASED CONTROLLABLE SUBAUTOMATA

In this section, we establish a complete lattice and then construct simulation operator, controllable operator and the inequalities of system over such complete lattice respectively. Then, an algorithm is proposed for the computation of the supremal simulation-based controllable subautomtata.

A. Complete Lattice

Definition 8: Given a plant $G = (X, \Sigma, x_0, \alpha, X_m)$ and a specification $R = (Q, \Sigma, q_0, \delta, Q_m)$, a poset is defined as $(2^{Q\times X}, \subseteq)$.

Obviously, the power set lattice $(2^{Q\times X}, \subseteq)$ is a complete lattice. In the next two subsections, we introduce several operators defined over $(2^{Q\times X}, \subseteq)$.

B. Simulation Operator

Definition 9: The simulation operator is defined as a mapping:

$$F_s : 2^{Q\times X} \to 2^{Q\times X},$$

for $Z \subseteq Q \times X$, $(q, x) \in F_s(Z)$ if the following conditions are satisfied:

1. $(q, x) \in Z$.
2. $q^{'} \in \delta(q, \sigma) \Rightarrow [\exists x^{'} \in \delta(x, \sigma)]\ [(q^{'}, x^{'}) \in Z]$.
3. $q \in Q_m \Rightarrow x \in X_m$.

Above definition of $F_s(Z)$ is built upon the complete lattice $(2^{Q\times X}, \subseteq)$, and it evolves from a similar operator in [13].

Proposition 1: If $Z \subseteq Z^{'}$, then $F_s(Z) \subseteq F_s(Z^{'})$.

Proof: For any $(q, x) \in F_s(Z)$, by the definition of $F_s(Z)$, we obtain that: (1) $(q, x) \in Z$, because $Z \subseteq Z^{'}$, we have $(q, x) \in Z^{'}$. (2) If $q^{'} \in \delta(q, \sigma)$ in R, then there exists $x^{'} \in \delta(x, \sigma)$ in G such that $(q^{'}, x^{'}) \in Z$. As $Z \subseteq Z^{'}$, we get $(q^{'}, x^{'}) \in Z^{'}$. Meanwhile, (q, x) also satisfies the third condition in $F_s(Z)$. Thus, $(q, x) \in F_s(Z^{'})$. ∎

The above proposition shows that F_s is monotone. Moreover, this monotonicity of F_s guarantees the existence of supremal solution of inequality $Z \subseteq F_s(Z)$ over the complete lattice $(2^{Q\times X}, \subseteq)$.

Proposition 2: Given a plant $G = (X, \Sigma, x_0, \alpha, X_m)$ and a specification $R = (Q, \Sigma, q_0, \delta, Q_m)$, ϕ is a simulation relation from R to G, if and only if $\phi \subseteq F_s(\phi)$, and $(q_0, x_0) \in \phi$.

Proof: (Necessity) Given a simulation relation ϕ such that $R \subseteq_\phi G$, thus, for any $(q, x) \in \phi$, we have (1) $(q_0, x_0) \in \phi$. (2) For each $(q, x) \in \phi$, if $q^{'} \in \delta(q, \sigma)$ in R, there exits $x^{'} \in \alpha(x, \sigma)$ in G such that $(q^{'}, x^{'}) \in \phi$, where $\sigma \in \Sigma$. (3) If $q \in Q_m$, then $x \in X_m$. Therefore, we can get that $(q, x) \in F_s(\phi)$, i.e $\phi \subseteq F_s(\phi)$.

(Sufficiency) Suppose $\phi \subseteq Q\times X$ and $\phi \subseteq F_s(\phi)$, therefore, for any $(q, x) \in \phi$, we have $(q,x) \in F_s(\phi)$. Then, (q, x) satisfies all the conditions of $F_s(\phi)$, those are, (1) For every $q^{'} \in \delta(q, \sigma)$ in R, $\exists x^{'} \in \alpha(x, \sigma)$ in G such that $(q^{'}, x^{'}) \in \phi$. (2) If $q \in Q_m$, then $x \in X_m$. Furthermore, $(q_0, x_0) \in \phi$, then ϕ is a simulation relation from R and G. ∎

By the definition of $F_s(Z)$, we have already obtained $F_s(Z) \subseteq Z$. If we also have $Z \subseteq F_s(Z)$, Z is a fixed-point of $F_s(Z)$, i.e. $F_s(Z) = Z$. This fixed point is the supremal solution of $Z \subseteq F_s(Z)$ due to the monotonicity of $F_s(Z)$. Moreover, we could achieve the supremal solution by the following iterative algorithm.

Theorem 1: Given a plant $G = (X, \Sigma, x_0, \alpha, X_m)$ and a specification $R = (Q, \Sigma, q_0, \delta, Q_m)$, the supremal simulation relation is the maximal fixed-point Z of the operator F_s if $(q_0, x_0) \in Z$, where $Z \subseteq Q \times X$. Moreover,

$$F_s(Z) = \lim_{i\to\infty} F_s^i(Q \times X),$$

where $F_s^0(Q\times X) = Q\times X$ is defined to be the identity function, and for each $i \geq 0$, $F_s^{i+1}(Q \times X) = F_s(F_s^i(Q \times X))$.

Proof: $(2^{Q\times X}, \subseteq)$ is a complete lattice, and the operator $F_s(Z)$ is the monotone function over such lattice. Thus, Z is the simulation relation if $(q_0, x_0) \in Z$ by proposition 2. Moreover, $F_s(supZ) \subseteq supZ$, therefore, we obtain $F_s(F_s(supZ)) \subseteq F_s(supZ)$ by the monotonicity of F_s. Then, we have the decreasing chain $\{supZ,\ F_s(supZ),\ F_s(F_s(supZ)),\ F_s(F_s(F_s(supZ))),\ ...\}$, where $supZ = Q \times X$. Then, we can obtain

$$\begin{aligned} F_s(Z) &= inf\{supZ, F_s(supZ), F_s(F_s(supZ))...\} \\ &= \lim_{i\to\infty} F_s^i(supZ) \\ &= \lim_{i\to\infty} F_s^i(Q \times X) \end{aligned}$$

∎

C. Controllable Operator

Definition 10: Given a plant $G = (X, \Sigma, x_0, \alpha, X_m)$, for any $s \in L(G)$, we define a nondeterministic state set $X_s = \{x \in X \mid x \in \alpha(x_0, s)\}$. Moreover, for any $x \in X_s$, we define the nondeterministic active event set as $\Gamma_s(x) = \cup_{x_1 \in X_s} \Gamma(x_1)$.

We could obtain all the states which are reachable from x_0 with the same string s by the nondeterministic state set. Besides, the nondeterministic active event set is the union of the active event set of the states in the corresponding nondeterministic state set. In particular, if an uncontrollable event is included in the active event set of any state in nondeterministic state set X_s, it will also contain in the set $\Gamma_s(x)$ for all $x \in X_s$.

Definition 11: Given a plant $G = (X, \Sigma, x_0, \alpha, X_m)$ and a specification $R = (Q, \Sigma, q_0, \delta, Q_m)$, the simulation-based controllable product of R and G is the automaton:

$$R \times_{sc} G = Ac(Q \times X \cup \{(q_m, x_m)\}, \Sigma, q_0 \times x_0, \gamma_{sc}, Q_m \times X_m)$$

where

$$\gamma_{sc}((q,x),\sigma)=\begin{cases}(q_m,x_m) & \sigma\in(\Sigma_{uc}\cap(\Gamma_s(x)-\Gamma(q)));\\ (\delta(q,\sigma),\alpha(x,\sigma)) & \sigma\in\Gamma(x)\cap\Gamma(q);\\ undefined & \text{otherwise.}\end{cases}$$

According to the definition of simulation-based controllable product, a transition that leads to the new states through event σ is allowed if the active event sets of this state pair (q, x) share the event σ. Besides, there will be a transition to the (q_m, x_m) if $\sigma \in (\Sigma_{uc} \cap (\Gamma_s(x) - \Gamma(q))$. Moreover, the state pairs that are not reachable from (q_0, x_0) should be removed by the accessible operator.

Definition 12: The controllable operator is defined as a mapping:

$$F_c : 2^{Q\times X} \to 2^{Q\times X},$$

for some $Z \subseteq Q\times X$, $(q, x) \in F_c(Z)$, if the following condition is satisfied:

$$(q, x) \notin Q_d \times X, Q_d = \cup_{\sigma\in\Sigma_{uc}} Q_{d\sigma},$$

where for any $\sigma \in \Sigma_{uc}$, $Q_{d\sigma} = \{q_{d\sigma} \in Q \mid (\exists x \in X)[\ (q_{d\sigma}, x) \in R \times_{sc} G \wedge (q_m, x_m) \in \gamma_{sc}((q_{d\sigma}, x), \sigma)]\}$.

From the simulation-based controllable product of R and G, we obtain that for any $q \in Q_d$, there exists x such that (q, x) has a transition to (q_m, x_m) along the corresponding uncontrollable event. We can delete those states by controllable operator F_c.

Proposition 3: If $Z \subseteq F_c(Z)$, where $Z \subseteq Q \times X$, then any $(q, x) \in Z$ satisfies the controllable condition.

Proof: Assume that there exists (q, x) $\in Z$ violating the controllable condition, where $Z \subseteq F_c(Z)$ and $Z \subseteq Q \times X$, thus, we have $q \in \delta(q_0, s)$ with $\delta(q, \sigma) = \emptyset$, and $s\sigma \in L(G)$, where $\sigma \in \Sigma_{uc}$. We can obtain that there exists $x^{'} \in \alpha(x_0, s)$ with $\sigma \in \Gamma(x')$. Moreover, (q, x') belongs to the state set of $R \times_{sc} G$ because it is reachable from (q_0, x_0) by the string s. Furthermore, we have $\sigma \in \Sigma_{uc} \cap (\Gamma_s(x') - \Gamma(q))$ as $\delta(q, \sigma) = \emptyset$ and $\sigma \in \Gamma(x')$. Thus, $(q_m, x_m) \in \gamma_{sc}((q, x), \sigma)$ by the definition of simulation-based controllable product. We obtain $q \in Q_d$, therefore, (q, x) $\in Q_d \times X$. On the other hand, we have (q, x) $\in F_c(Z)$ as $Z \subseteq F_c(Z)$. Then, we obtain (q, x) $\notin Q_d \times X$ by the definition of controllable operator. Thus, there is a contradiction which shows the assumption is wrong. Then, we obtain that any (q, x) $\in Z$ satisfies the controllable condition, where $Z \subseteq F_c(Z)$ and $Z \subseteq Q \times X$. ■

Proposition 4: If $Z \subseteq Z'$, then $F_c(Z) \subseteq F_c(Z')$.

Proof: For any $(q, x) \in F_c(Z)$, we obtain (q, x) $\in Z$ and (q, x) $\notin Q_d \times X$. Then, (q, x) $\in Z'$ because $Z \subseteq Z'$. Thus, (q, x) $\in F_c(Z')$. Therefore, we have $F_c(Z) \subseteq F_c(Z')$.

We can obtain that $F_c(Z)$ is monotone from above proposition. Furthermore, this monotonicity of $F_c(Z)$ guarantees the existence of supremal solution of inequality $Z \subseteq F_c(Z)$ over the complete lattice ($2^{Q\times X}$, $\subseteq$). ■

D. Simulation-based Controllable Subautomata

Consider the system of inequalities as follows.
(1) $F(Q_1 \times X_1) \subseteq F_s(Q_1 \times X_1)$;
(2) $F(Q_1 \times X_1) \subseteq F_c(Q_1 \times X_1)$,
where $Q_1 \times X_1 \subseteq Q \times X$, and $F(Q_1 \times X_1) = Q_1 \times X_1$, we have the following proposition.

Proposition 5: Let Y = $\{\ Q_1 \times X_1 \subseteq Q \times X \mid F(Q_1 \times X_1) \subseteq F_s(Q_1 \times X_1)$ and $F(Q_1 \times X_1) \subseteq F_c(Q_1 \times X_1)\ \}$, then $Q_1 \times X_1$ is the set of simulation-based controllable state pairs if $(q_0, x_0) \in Q_1 \times X_1$.

Proof: Because $(q_0, x_0) \in Q_1\times X_1$ and $Q_1\times X_1 \subseteq F_s(Q_1\times X_1)$, we obtain that $Q_1 \times X_1$ is a simulation relation from R to G by proposition 2. Moreover, $Q_1 \times X_1 \subseteq F_c(Q_1 \times X_1)$, then for any $(q, x) \in Q_1 \times X_1$, it satisfies the controllable condition by proposition 3. Therefore, $Q_1 \times X_1$ is the set of simulation-based controllable state pairs. ■

Proposition 6:

$$Y_1 = \{Q_1 \times X_1 \in 2^{Q\times X} \mid h_1(Q_1 \times X_1) = Q_1 \times X_1\}$$

is a set of fixed points of h_1, where $h_1 : 2^{Q\times X} \to 2^{Q\times X}$ is defined as: for any $Q_1 \times X_1 \in 2^{Q\times X}$,

$$\begin{aligned} h_1(Q_1 \times X_1) \quad = \quad & sup\{Q_2 \times X_2 \in 2^{Q\times X} : F(Q_2 \times X_2) \\ & \subseteq F_s(Q_1 \times X_1)\} \cap sup\{Q_3 \times X_3 \in 2^{Q\times X} : \\ & F(Q_3 \times X_3) \subseteq F_c(Q_1 \times X_1)\}\end{aligned}$$

then supY = supY_1.

Proof: From lattice theory, we know that ($2^{Q\times X}$, $\subseteq$) is a compete lattice over which we definite $F_s(Q_1 \times X_1)$ and $F_c(Q_1 \times X_1)$. $F_s(Q_1 \times X_1)$ and $F_o(Q_1 \times X_1)$ are monotone by proposition 1 and proposition 4. F$(Q_2 \times X_2) = Q_2 \times X_2$ and F$(Q_3 \times X_3) = Q_3 \times X_3$ are disjunctive as well as conjunctive, because they are identity functions. Therefore, supY = supY_1 according to Lemma 1 [9]. ■

Furthermore, supY can be calculated by the following procedure.

Proposition 7: Let $y_0 = Q \times X$, $\forall k \geq 0$, $y_{k+1} = h_1(y_k)$ until $y_{m+1} = y_m$, we obtain $y_m = supY$. Then, y_m is the supremal simulation-based controllable state set, if $(q_0, x_0) \in y_m$.

Proof: By Lemma 2, we have $y_m = supY$. Furthermore, if $(q_0, x_0) \in y_m$, we have any (q, x) $\in y_m$ is the simulation-based controllable state pair by Proposition 5. Therefore, y_m is the supremal simulation-based controllable state set. ■

We introduce the concept of subautomata to describe the new automata derived from the original system.

Definition 13: Given an automaton $G = (X, \Sigma, x_0, \alpha, X_m)$, the subautomaton of G is defined as $G_1 = (X_1, \Sigma_1, x_0, \alpha, X_{m1})$, where $X_1 \subseteq X$, $X_{m1} \subseteq X_m$, and $\alpha_1 = \alpha \mid X_1 \times \Sigma \to X_1$.

The subautomaton G_1 has the same initial state and the event set as the original system. Thus, once x $\in X_1$ in G_1, it will have the same transition and active events as those in G. Besides, the states and marked states in G_1 are picked from the corresponding sets in G, therefore, the number of states in state set and marked state set in G_1 is smaller than those in G.

Definition 14: Given an automaton $R = (Q, \Sigma, q_0, \delta, Q_m)$, the subautomata operator is defined as:

$$Rc(Q_R(Z)) = (Q_{rc}, \Sigma, q_0, \delta, Q_{rcm}),$$

where $Z \subseteq Q \times X$, $Q_{rc} = \{q \mid q \in Q_R(Z)\}$, $Q_{rcm} = Q_m \cap Q_{rc}$, and $Q_R : Q \times X \to Q$ defined as $Q_R(q, x) = q$ for any $(q, x) \in Q \times X$.

We have $Q_R(Z) \subseteq Q$, where $Z \subseteq Q \times X$, because $Q_R(Z)$ projects $Q \times X$ to Q. Then, we can construct a subautomton of R from $Q_R(Z)$ by using of subautomata operator.

Theorem 2: If $(q_0, x_0) \in y_m$, the supremal simulation-based controllable subautomaton exists and it equals to $Ac(Rc(Q_R(y_m)))$.

Proof: By proposition 7, we have y_m is the supremal simulation-based state set, if $(q_0, x_0) \in y_m$. Then, we have $Q_1 = Q_R(y_m) \subseteq Q$. Next, we build the subautomton of R according to Q_1. Following this, we remove the states which are not reachable from q_0 in $Rc(Q_R(y_m))$. Therefore, Ac(Rc($Q_R(y_m)$)) is supremal simulation-based controllable. ■

Algorithm 1: The algorithm for computing the supremal simulation-based controllable subautomata as follows.

1. Let $y_0 = Q \times X$, $\forall k \geq 0$, $y_{k+1} = h_1(y_k)$ until $y_{m+1} = y_m$.
2. If $(q_0, x_0) \notin y_m$, the supremal simulation-based controllable subautomaton does not exist, otherwise, if $(q_0, x_0) \in y_m$, $Ac(Rc(Q_R(y_m)))$ is the supremal simulation-based controllable subautomaton.

This algorithm starts from the whole state set $Q \times X$. The number of the state pairs is reduced or unchanged through h_1, thus, the algorithm can be terminated in finite time.

IV. EXAMPLE

To illustrate the proposed method of computing the supremal simulation-based controllable subautomata, we present an example.

Example 1: Consider a plant $G = (X, \Sigma, x_0, \alpha, X_m)$ and a specification $R = (Q, \Sigma, q_0, \delta, Q_m)$ shown in Fig.1, where $X_m = \{x_3, x_5, x_6\}$ and $q_m = \{q_4\}$, we assume $\Sigma_{uc} = \{d\}$ and $\Sigma_c = \{a, b, c\}$.

Firstly, we would like to obtain the supremal simulation relation between the plant and the specification. According to Theorem 1, the state pairs which marked by blue shadow in Fig. 2 are excluded by $F_s(Q \times X)$ in the first around. And the state pair (q_1, x_1), (q_1, x_4) marked by the red shadow in Fig.2, are removed by $F_s^2(Q \times X)$ in the second iteration. Finally, we obtain the fixed-point of $F_s(Q \times X)$: $\{(q_0, x_0), (q_2, x_2), (q_3, x_0), (q_3, x_5), (q_4, x_3), (q_4, x_5), (q_4, x_6)\}$. We can see that (q_0, x_0) belongs to this fixed-point of F_s, therefore, the fixed-point of F_s is the supremal simulation relation.

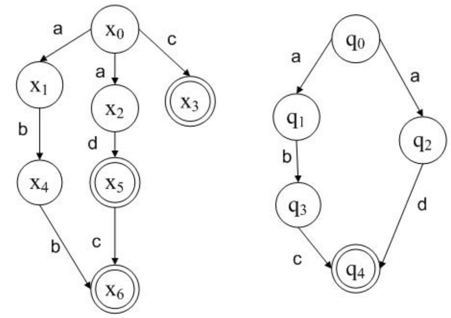

Fig. 1. Plant (Left) and Specification (Right)

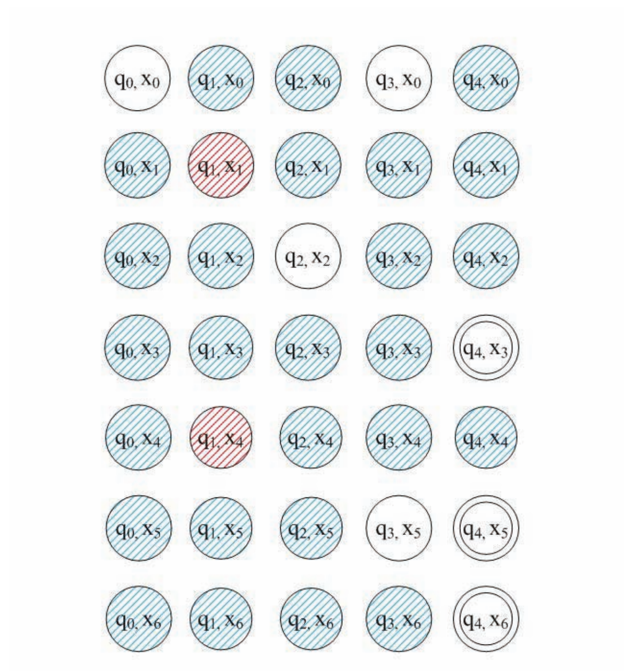

Fig. 2. Iterative algorithm for F_S

In Fig.3, we can obtain that $q_1 \in Q_{dd}$, which violates the controllable condition, therefore, all the states pairs $q_1 \times X$ should be deleted. As a result, the set of controllable state pairs is $Z_c = \{(q, x) \in Q \times X \mid (q, x) \notin q_1 \times X\}$.

According to the Algorithm 1, we have

1. $y_0 = Q \times X$,

$$\begin{aligned} y_1 &= h_1(y_0) \\ &= \{(q_0, x_0), (q_1, x_1), (q_1, x_4), (q_2, x_2), (q_3, x_0), (q_3, x_5), \end{aligned}$$

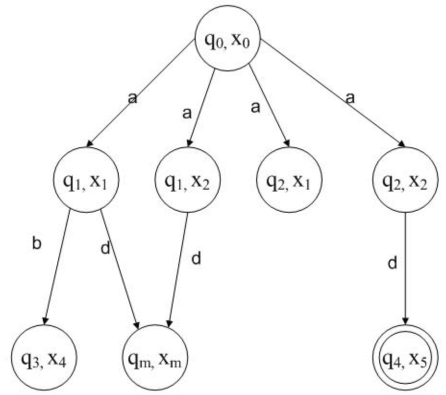

Fig. 3. Simulation-based Controllable Product

$$
\begin{aligned}
& (q_4, x_3), (q_4, x_5), (q_4, x_6)\} \cap \{(q, x) \in Q \times X \mid (q, x) \\
& \notin q_1 \times X\} \\
= \; & \{(q_0, x_0), (q_2, x_2), (q_3, x_0), (q_3, x_5), (q_4, x_3), (q_4, x_5), \\
& (q_4, x_6)\}
\end{aligned}
$$

$$
\begin{aligned}
y_2 \; = \; & h_1(y_1) \\
= \; & \{(q_0, x_0), (q_2, x_2), (q_3, x_0), (q_3, x_5), (q_4, x_3), (q_4, x_5), \\
& (q_4, x_6)\} \cap \{(q_0, x_0), (q_2, x_2), (q_3, x_0), (q_3, x_5), \\
& (q_4, x_3), (q_4, x_5), (q_4, x_6)\} \\
= \; & \{(q_0, x_0), (q_2, x_2), (q_3, x_0), (q_3, x_5), (q_4, x_3), (q_4, x_5), \\
& (q_4, x_6)\} \\
= \; & y_1
\end{aligned}
$$

2. We get that $(q_0, x_0) \in y_1$, which guarantees the existence of supremal simulation-based controllable subautomaton. Then, $Rc(Q_R(y_1))$ is obtained in Fig. 4 (Left). Furthermore, the supremal simulation-based controllable subautomaton $Ac(Rc(Q_R(y_1)))$ is achieved in Fig. 4 (Right).

V. CONCLUSIONS

By resorting to lattice theory, we proposed a computational approach to solve the supremal simulation-based controllable subautomata, where both the plant and the specification are modelled as nondeterministic automata. The obtained solution provides a sufficient condition of the existence of the supremal simulation-based controllable subautomta and an explicit algorithm to calculate such subautomta. Future work will concentrate in development of corresponding algorithm for partial observation case.

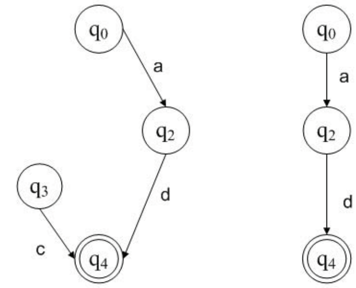

Fig. 4. $Rc(Q_R(y_1))$ (Left) and $Ac(Rc(Q_R(y_1)))$ (Right)

VI. ACKNOWLEDGMENTS

The financial supports from Singapore Ministry of Educations AcRF Tier 1 funding, Temasek Defence Systems Institute, and Temasek Lab are gratefully acknowledged.

The third author thanks the supports from the National Natural Science Foundation of China (60974019), Natural Science Foundation of Guangdong Province of China (9451009001002686), and Foundation of Guangdong University of Technology (083029).

References

[1] R. Milner.*Communication and Concurrency*. Prentice Hall, New York, 1989.

[2] Fernandez, J. "An implementation of an efficient algorithem for bisimulation equivalence," *Sci.Comput. Programming*, vol. 13, pp. 219-236, 1990.

[3] J.Komenda and J. H. van Schuppen, "Control of discrete-event systems with partial observations using coalgebra and coinduction," *Discrete Event Dynamical Systems: Theory and Applications,* 15(3), 257-315, 2005.

[4] J.Komenda and J. H. van Schuppen,"Decentralized supervisory control with coalgebra," *In Proceedings of European Control Conference 2003 (ECC'03)*, Cambridge, September 1-4, 2003, only CD-ROM.

[5] G. J. Pappas, "Bisimilar linear systems", *Automatica*, vol. 39, pp. 2035-2047, 2003.

[6] E. Haghverdi, P. Tabuada, and G. J. Pappas, "Bisimulation relation for dynamical, control, and hybrid systems," *Theoret. Comp. Sci.*,vol.342, pp. 229-261, 2005.

[7] Giordano Pola and Paulo Tabuada, "Symbolic models for nonlinear control systems: Alternating approximate bisimulations," *SIAM Journal on Control and Optimization*, 48(2), 719-733, 2009.

[8] C. G. Cassandtras and S. Lafortune, *Introduction to Discrete Event Systems*. Boston, MA: Kluwer, 1999.

[9] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems*. Kluwer Academic Publishers, Boston, MA, 1995.

[10] F. Liu, D. Qiu, and H. Lin, "Bisimilarity control of nondeterministic discrete event systems,", *Conditionally accepted by IEEE Transactions on Automatic Control*, 2009.

[11] C. Zhou, R. Kumar, and S. Jiang, "A small model theorem for bisimilarity control under partial observation," *IEEE Transactions on Automation Science and Engineering*,vol.4, no.1, pp.93-97, 2007.

[12] C. Zhou, R. Kumar, and S. Jiang, "Control of Nondeterministic Discrete Event Systems for Bisimulation Equivalence,". *IEEE Transactions on Automatic Control*, vol.51, pp. 754-765, 2006.

[13] Paulo Tabuada. *Verification and Control of Hybrid Systems*, Springer, 2009.