Reducing the Supervisory Control of Discrete-Event Systems under Partial Observation

Vahid Saeidi, Ali A. Afzalian, and Davood Gharavian

Abstract—Supervisor reduction procedure can be used to construct the reduced supervisor with a reduced number of states in discrete-event systems. However, it was proved that the reduced supervisor is control equivalent to the original supervisor with respect to the plant; it has not been guaranteed that the reduced supervisor and the original one are control equivalent under partial observation. In this paper, we extend the supervisor reduction procedure by considering partial observation; namely not all events are observable. A feasible supervisor which is constructed under partial observation becomes reduced based on control consistency of uncertainty sets of states, instead of the original supervisor. In order to construct a partial observation reduced supervisor, a partial observation control cover is constructed based on control consistency of uncertainty sets in the supervisor. Four basic functions are defined in order to capture the control and marking information on the uncertainty sets. In the resulting reduced supervisor, only observable events can cause state changes. The results are illustrated by some examples.

Index Terms—control consistency, control cover, discrete-event systems, partial observation, supervisor reduction.

I. INTRODUCTION

The state size and the computational complexity of a monolithic supervisor increase with state sizes of the plant and the specification [1], and may lead to state explosion [2]. However, the application of this theory is restricted, some works are reported on application of this theory in practice, e.g. [3, 4]. Although modular [5, 6] and incremental [7, 8] approaches try to overcome the complexity of the supervisor synthesis, other approaches tend to reduce a supervisor for simple implementation. The supervisor reduction procedure, given by [9], is an evolution of the proposed method in [10]. This procedure reduces the redundant information in the supervisor synthesis without any effect on controlled behavior. A reduced supervisor has some advantages compared to the original supervisor, such as simplicity. Although this procedure is a heuristic method, it has been extended to other applications, e.g. coordination planning for distributed agents [11], supervisor localization procedure with full observation [12], and supervisor localization procedure under partial observation [13]. In [13], the authors employed the concept of relative observability to compute a partial-observation monolithic supervisor, and then they designed a localization procedure using (feasible) partial-observation supervisor to decompose the supervisor into a set of local controllers.

In this paper, we extend supervisor reduction procedure [9], to address the issue of partial observation. At first, we synthesize a partial-observation monolithic supervisor using the concept of relative observability [14]. Relative observability is stronger than observability [15, 16], weaker than normality [15, 16], and the supremal relative observable (and controllable) sublanguage of a given language exists. The supremal sublanguage may be effectively computed, and then implemented by a partial-observation (feasible and non-blocking) supervisor [13, 17]. Then, we suitably extend the supervisor reduction procedure in [9] to reduce a supervisor under partial observation.

In this paper, the partial-observation control cover is introduced. In particular, it is defined on the state set of the partial-observation supervisor; roughly speaking the latter corresponds to the power set of the full-observation supervisor’s state set. As a result, a partial-observation reduced supervisor contains only observable state transitions.

The rest of the paper is organized as follows: In Section II, the necessary preliminaries are reviewed. Reducing the supervisory control under partial observation is proposed in Section III. In Section IV, five examples are given to clarify the proposed method. Finally, concluding remarks are given in Section V.

II. PRELIMINARIES

A discrete-event system (DES) is represented by an automaton $G = (Q, \Sigma, \delta, q_0, Q_m)$, where $Q$ is a finite set of states, with $q_0 \in Q$ as the initial state and $Q_m \subseteq Q$ being the marked states; $\Sigma$ is a finite set of events ($\sigma$) which is partitioned as a set of controllable events $\Sigma_c$ and a set of uncontrollable events $\Sigma_{uc}$. where $\Sigma = \Sigma_c \cup \Sigma_{uc}$. $\delta$ is a transition mapping $\delta: Q \times \Sigma \rightarrow Q$, $\delta(q, \sigma) = q'$ gives the next state $q'$ is reached from $q$ by the occurrence of $\sigma$. $G$ is discrete-event model of the plant. In this context $\delta(q_0, s)$ means that $\delta$ is defined for $s$ at $q_0$. $L(G) = \{s \in L(\Sigma) | \delta(q_0, s)\}$ is the closed behavior of $G$ and $L_m(G) = \{s \in L(G) | \delta(q_0, s) \in Q_m\}$ is the marked behaviour of $G$ [17, 18].

A set of all control patterns is denoted with $\Gamma = \{\gamma \in \text{Pwr}(\Sigma) | \exists s \in \Sigma_{uc}\}$. A supervisory control for $G$ is any map $V: L(G) \rightarrow \Gamma$, where $V(s)$ represents the set of enabled events after the occurrence of the string $s \in L(G)$. The pair $(G, V)$ is written $V/G$, to suggest “ $G$ under the supervision of $V$”. A behavioral constraint on $G$ is given by specification language $E \subseteq \Sigma^*$. Let $K \subseteq L_m(G) \cap E$ be the supremal controllable sublanguage of $E$ w.r.t. $L(G)$ and $\Sigma_{uc}$. i.e. $K = \sup\{L_m(G) \cap E\}$ [17]. If $K \neq \emptyset$, it can be shown as a DES, $\text{SUP} =$

Manuscript received 2 August 2017; accepted 9 October, 2017. Engineering. Abbaspour School of Engineering, Shahid Beheshti University, Tehran, Iran.

Vahid Saeidi (v_saeidi@sibru.ac.ir), Ali A. Afzalian (Afzalian@sibru.ac.ir), and D. Gharavian (d_gharavian@sibru.ac.ir) are with the department of Electrical
\((X, \Sigma, \xi, x_0, X_m)\), which is the recognizer for \(K\). If \(G\) and \(E\) are finite-state DES, then \(K\) is regular language. Write \(|\cdot|\) for the state size of DES. Then \(|\text{SUP}| \leq |G||E|\). In applications, engineers want to employ RSUP, which has a fewer number of states (i.e. \(|\text{RSUP}| < |\text{SUP}|\)) and is control equivalent to SUP w.r.t. G [9], i.e.

\[
\begin{align*}
L_m(G) \cap L_m(\text{RSUP}) &= L_m(\text{SUP}), \\
L(G) \cap L(RSUP) &= L(SUP).
\end{align*}
\]

The natural projection is a mapping \(P: \Sigma' \rightarrow \Sigma_0^\prime\) where

1. \(P(e) = e\) (\(e\) is the empty string),
2. \(P(s) = P(s')\) for\s\(s \in \Sigma', s' \in \Sigma\),
3. \(P(\sigma a) = \sigma P(a)\) if \(\sigma \in \Sigma_0\) and \(P(a) = e\) if \(\sigma \notin \Sigma_0\).

The effect of an arbitrary natural projection \(P\) on a string \(s \in \Sigma^*\) is to erase the events in \(s\) that do not belong to observable events set, \(\Sigma_0\). The natural projection \(P\) can be extended and denoted with \(P: Pwr(\Sigma^0) \rightarrow Pwr(\Sigma_0^0)\). For any \(L \subseteq \Sigma^*\), \(P(L) = \{P(s) | s \in L\}\). The inverse image function of \(P\) is denoted with \(P^{-1}(L) = \{s \in \Sigma^* | P(s) \in L\}\) for any \(L \subseteq \Sigma_0^0\).

Let \(\Sigma'\) be the set of uncertainty in \(\Sigma\), \(\Sigma_0\) be the subset of observable events, \(\Sigma_0\) be the recognizer of supervisor \(K\). Under partial observation, if \(s \in L(\text{SUP})\) occurs, then \(P(s)\) is observed. Let \(U(s)\) be the set of states that may be reached by some \(s'\) that looks like \(s\), i.e.

\[
U(s) = \{x \in X | (\exists \xi' \in \Sigma') P(s) = P(s'), x = \xi(x_0, s')\}.
\]

Let \(U(X)\) be the set of uncertainty sets of all states in \(X\), associated with strings in \(L(SUP)\), i.e.

\[
U(X) = \{U(s) \subseteq X | s \in L(SUP)\}.
\]

The transition function associated with \(U(X)\) is \(\xi: U(X) \times \Sigma_0 \rightarrow U(X)\). \(\xi\) is given by

\[
\xi(U, \sigma) = \bigcup\{(x, u_1 u_2) | x \in U, u_1, u_2 \in \Sigma_0^\prime\}.
\]

Where \(\Sigma_0 = \Sigma - \Sigma_0\). If there exist \(u_1, u_2 \in \Sigma_0^\prime\) such that \(\xi(x, u_1 u_2)\), then \(\xi(U, \sigma)\) is defined and denoted as \(\xi(U, \sigma)\).

Having \(U(X)\) and \(\xi\), partial observation monolithic supervisor SUP can be defined. It is a feasible supervisor, and its synchronization by the plant is control equivalent to the original supervisor w.r.t. the plant. SUP is defined as follows,

\[
\text{SUP} = (U(X), \Sigma_0, \xi, U_0, U_m)
\]

Where \(U_0 = U(\epsilon)\) and \(U_m = \{U \in U(X) | U \cap X_m \neq \emptyset\}\). It is known [13], that \(L(\text{SUP}) = P(L(SUP))\) and \(L_m(\text{SUP}) = P(L_m(SUP))\).

Let \(U \in U(X), x \in U\) be any state in SUP and \(\alpha \in \Sigma\) be a controllable event. We know that 1. \(\alpha\) is enabled at \(x \in U\), if \(\xi(x, \alpha)\), or 2. \(\alpha\) is disabled at \(x \in U\), \(\neg \xi(x, \alpha)\) and \((\exists s \in \Sigma^0)[\xi(x_0, s) \neq x \& \xi(U_0, Ps) = U] \neq \xi(q_0, s\sigma)\) or 3. \(\alpha\) is not defined at \(x \in U\), if \(\neg \xi(x, \alpha)\) and \(\neg \xi(x, \alpha)\) and \((\exists s \in \Sigma^0)[\xi(x_0, s) \neq x \& \xi(U_0, Ps) = U] \neq \xi(q_0, s\sigma)\).

Under partial observation, the control actions after string \(s \in L(SUP)\) depend on the uncertainty set \(U(s) \subseteq U(X)\), i.e. the state of SUP. It was proved that, if \(\alpha\) is enabled at \(x \in U\), then for all \(x' \in U\), either \(\alpha\) is also enabled at \(x' \in U\), or \(\alpha\) is not defined at \(x' \in U\). On the other hand, if \(\alpha\) is disabled at \(x \in U\), then for all \(x' \in U\), either \(\alpha\) is also disabled at \(x' \in U\), or \(\alpha\) is not defined at \(x' \in U\) [13].

In order to propose a supervisor reduction procedure under partial observation, consider the following four functions which capture the control and marking information on the uncertainty sets. Define \(E: U(X) \rightarrow PWR(\Sigma_0)\) according to

\[
E(U) = \{\sigma \in \Sigma_0 | (\exists \xi(x, \sigma))\}
\]

\(E(U)\) denotes the set of events enabled at state \(U\). Also define \(D: U(X) \rightarrow PWR(\Sigma_0)\) according to

\[
D(U) = \{\sigma \in \Sigma_0 | (\exists x \in U) \neg \xi(x, \sigma)) \& (\exists s \in \Sigma^0)[\xi(x_0, s) = x \& \xi(q_0, s\sigma)]\}
\]

\(D(U)\) is the set of events, which are disabled at state \(U\). Next, define \(M: U(X) \rightarrow \{0, 1\}\) according to

\[
M(U) = \begin{cases} 1, & \text{if } (U \in U_m), \\ 0, & \text{otherwise}. \end{cases}
\]

\(M(U) = 1\) if \(U\) is marked in SUP, i.e. \(U\) contains a marked state.
of SUP. Finally define $T : \mathcal{U}(X) \to \{0, 1\}$ according to

$$T(U) = \begin{cases} 1, & \text{if } (\exists s \in \Sigma^*) \xi(x_0, s) \in U \\ \xi(U_0, Ps) = U, \delta(q_0, s) \in Q_m \\ 0, & \text{otherwise.} \end{cases}$$

$T(U) = 1$ if $U$ contains some states that correspond to a marked state of $G$, i.e. $U$ contains a marked state of $G$. Now, the control consistency relation $\mathcal{R}_U \subseteq \mathcal{U}(X) \times \mathcal{U}(X)$ can be defined. $U, U' \in \mathcal{U}(X)$ are control consistent, i.e. $(U, U') \in \mathcal{R}_U$, if

$$E(U) \cap D(U') = E(U') \cap D(U) = \emptyset.$$  

(6)

Thus a pair of uncertainty sets $(U, U')$ satisfies $(U, U') \in \mathcal{R}_U$, if (i) each event is enabled at least at one state of $U$, but is not disabled at any state of $U'$, and vice versa; (ii) $U, U'$ both contain marked states of SUP (both do not contain) provided that they both contain states corresponding to some marked states of $G$ (both do not contain). It is easily verified that $\mathcal{R}_U$ is generally not transitive, thus it is not an equivalence relation. This leads to the partial-observation control cover. Let $I$ be some index set, and $\mathcal{C}_U = \{U_i \subseteq \mathcal{U}(X) | i \in I\}$ be a cover on $\mathcal{U}(X)$. $\mathcal{C}_U$ is a partial observation control cover, if

$$(i) (\forall i \in I) (\forall U' \in U_i)(U, U') \in \mathcal{R}_U,$$

(ii) $(\forall i \in I)(\exists \sigma \in \Sigma_0)(\exists U \in U_i)\xi(U, \sigma) \Rightarrow [(\exists j \in I)(\forall U' \in U_i)\xi(U', \sigma) \Rightarrow \xi(U', \sigma) \in U_j].$  

(8)

A partial observation control cover $\mathcal{C}_U$ lumps the uncertainty sets $U \in \mathcal{U}(X)$ into cells $U_i \in \mathcal{C}_U$, $i \in I$ such that (i) the uncertainty sets $U$ that reside in the same cell $U_i$ must be pairwise control consistent, (ii) for every observable event $\sigma \in \Sigma_0$, the uncertainty set that is reached from any uncertainty set $U' \in U_i$ by one step transition $\sigma$ must be covered by the same cell $U_i$. Obviously, two uncertainty sets $U$ and $U'$ belong to a common cell of $\mathcal{C}_U$, if and only if $U$ and $U'$ are control consistent, and two future uncertainty sets that can be reached respectively from $U$ and $U'$ by a given observable string are again control consistent. $\mathcal{C}_U$ is called a partial-observation control congruence if $\mathcal{C}_U$ happens to be a partition on $\mathcal{U}(X)$, namely its cells are pairwise disjoint. Having $\mathcal{C}_U$, $U_0 = U(e)$ and $x_0 \in U_0$, a generator $I = \{l_1, \Sigma_0, i_0, I_m\}$ can be defined over $\Sigma_0$ as follows,

$$i_0 \in I \text{ such that } U_0 \in U_{i_0},$$

$$l_m = \{i \in I | (\exists U \in U_i)X_m \cap U \neq \emptyset\}$$

$$\zeta : I \times \Sigma_0 \to I \text{ with } \zeta(i, \sigma) = j,$$

$$\text{if } (\exists U \in U_i)\xi(U, \sigma) \in U_j.$$  

(9)

Note that, overlapping of some states results that $i_0$ and $\zeta$ may not be uniquely determined, and $J$ may not be unique. If $\mathcal{C}_U$ is partition on $\mathcal{U}(X)$, $J$ can be determined uniquely and it can be selected as the reduced supervisor, RSUP$_P$.

We prove in Theorem 1, RSUP$_P$ is control equivalent to SUP w.r.t. $G$.

**Theorem 1:** RSUP$_P$ is control equivalent to SUP w.r.t. $G$, i.e.

$$L(G) \cap L(RSUP_P) = L(SUP),$$  

(10)

$L_m(G) \cap L_m(RSUP_P) = L_m(SUP).$  

(11)

**Proof:** We prove the claim in two steps. a. $\subseteq$, b. $\supseteq$.

a. As it was assumed that $L_m(SUP)$ is not empty, it follows that $L(G)$ and $L(RSUP_P)$ are not empty, and as they are closed, the empty string $\varepsilon$ belongs to each. Now, suppose that $s \in L(G) \cap L(RSUP_P)$ implies that $s \in L(SUP)$ and $s \sigma \in L(G) \cap L(RSUP_P)$ such that $\sigma \in \Sigma$. We must prove that $s \sigma \in L(SUP)$. If $s \sigma \in L(SUP)$, because $L(SUP)$ is controllable and observable. Now, assume $s \sigma \in L(G) \cap L(RSUP_P)$. Since $U$ and $U'$ belong to the same cell $U_i$, by definition of partial-observation control cover, they must be control consistent, i.e. $(U, U') \in \mathcal{R}_U$. Thus, $E(U) \cap D(U') = \emptyset$ which implies that $D(U') = \emptyset$. It means that all controllable and observable $\sigma$ that is enabled at $U$, cannot be disabled at $U'$. Thus, $\exists x \in U'$, either $\xi(x, \sigma)$ or $(\forall t \in \Sigma^*)[\xi(x_0, t) = x \& \neg\delta(q_0, tr)]$. Note that, $s \sigma \in L(G)$. Thus $-\delta(q_0, tr)$ is not true. Therefore, $\xi(x, \sigma)$ is true, i.e. $s \sigma \in L(SUP)$.

Now, assume $s \in L_m(G) \cap L_m(RSUP_P)$. It means that $\xi(i_0, s) \in I_m$. From (9), it is obvious $\xi(x_0, s) \in X_m$, i.e. $s \in L_m(SUP)$. b. Suppose that $s \in L(SUP)$ implies that $s \in L(G) \cap L(RSUP_P)$. Assume $s \sigma \in L(SUP)$. If $\sigma \in \Sigma - \Sigma_0$, then it is a self-loop transition at some states in RSUP$_P$. Thus, $s \in L(G) \cap L(RSUP_P) \Rightarrow s \sigma \in L(G) \cap L(RSUP_P)$. If $\sigma \in \Sigma_0$, then (9) implies that $\xi(i, \sigma) = j$. Thus, $s \sigma \in L(G) \cap L(RSUP_P)$.

Now, assume $s \in L_m(SUP)$. It means that $\xi(x_0, s) \in X_m$. From (9), we can write $\xi(i_0, s) \in I_m$. Namely, $s \in L_m(G) \cap L_m(RSUP_P)$. The proof is complete.

**Corollary 1:** Let $G$ be a non-blocking plant, described by closed and marked languages $L(G), L_m(G) \subseteq \Sigma^*$, and $SUP = (X, \Sigma, \xi, x_0, X_m)$ be the recognizer of the supervisor $K$, i.e. $K = L_m(SUP)$. Let RSUP$_P$ be the reduced supervisor under partial observation. If $K$ is relatively observable w.r.t. $(G, P), \text{ where } P : \Sigma^* \to \Sigma_2^*$, and $K \subseteq C \subseteq L_m(G)$, then $P(SUP)$ is control equivalent to $P(SUP)$ w.r.t. $G$, i.e.

$$L_m(G) \cap P^{-1}(L_m(P(SUP))) = L_m(G) \cap P^{-1}(L_m(P(SUP))).$$

$$L_m(G) \cap P^{-1}(L(P(RSUP_P))) = L_m(G) \cap P^{-1}(L(P(SUP))).$$

In order to clarify the proposed method for reducing a supervisor under partial observation, some examples are illustrated in the next section.

**IV. EXAMPLES**

In this section, we consider examples in order to verify the extended theory in Section III. The model construction and supervisor synthesis are carried out by TCT software [20]. A brief description of TCT procedures, which are used in this paper, is given in the Appendix.

**Example 1:** Let $\Sigma = \{1, 2, 3\}$ and $G, SUP$ be the plant and the recognizer of supervisor, respectively (Fig. 1). Obviously, we can find $C_1$ and $C_2$ such that $K = L_m(SUP)$ is relatively observable w.r.t. $(G_1, P_1)$, where $P_1 : \Sigma^* \to \Sigma_1^*$ and $\Sigma_1 = \{1, 3\}$ and $K$ is relatively observable w.r.t. $(G_2, P_2)$, where $P_2 : \Sigma^* \to \Sigma_2^*$ and $\Sigma_2 = \{2, 3\}$. But, we cannot find any $C$ such that $K$ is relatively observable w.r.t. $(G, P_0)$, where $P_0 : \Sigma^* \to \Sigma_0$ and $\Sigma_0 = \{3\}$. We can find uncertainty sets $U_1(X)$ and $U_2(X)$ corresponding to $P_1$ and $P_2$, respectively.
Two vehicles may be in state 0 (at A), state j (while travelling in section j = 1, … , 4), or state 5 (at B). The specification is based on protecting B if rejected, it is returned to B. If a work piece is accepted by TU, it is released from the system; hence it cannot be further reduced, as shown in Fig. 6, 7, respectively. The recognizer of relative observable supervisor, SUP and the partial observation reduced supervisor, corresponding to P0: Σ* → Σ0, Σ0 = Σ – {1,3,5}, are shown in Figs. 8, 9, respectively. We see that, events 1, 3, 5 appear just as self-loop transitions, one at each one state of the reduced supervisor, RSUP0 (Fig. 9). Since the recognizer of partial observation supervisor, SUPO cannot be further reduced, RSUP1 and SUPO are the same.

Example 4: Supervisory control of guide way under partial observation

Consider a guide way with two stations A and B, which are connected by a single one-way track from A to B on a guide way, as shown in Fig. 10. The track consists of 4 sections, with stoplights (*) and detectors (!) installed at various section junctions [17]. Two vehicles V1, V2 use the guide way simultaneously. V1, i = 1, 2 may be in state 0 (at A), state j (while travelling in section j = 1, ..., 4), or state 5 (at B). The
generator of $V_i$, $i = 1, 2$ are shown in Fig. 11. The plant to be controlled is $G = \text{sync}(V_1, V_2)$. To prevent collision, control of the stoplights must ensure that $V_1$ and $V_2$

![Fig. 10. Schematic of a guide way](image)

![Fig. 11. DES model of each vehicle](image)

![Fig. 12. The relative observable supervisor for the guide way $\Sigma_{uo} = \{13,23\}$](image)

![Fig. 13. The feasible supervisor for the guide way $\Sigma_{uo} = \{13,23\}$](image)

![Fig. 14. The partial observation reduced supervisor for the guide way. RSUP$_0$ never travel on the same section of track simultaneously. Namely, $V_i, i = 1, 2$ are mutual exclusion of the state pairs $(i, i), i = 1, \ldots, 4$. Controllable events are odd-numbered and the unobservable events 13, 23 are considered to synthesize the supremal relative observable supervisor, i.e. $P_0: \Sigma^* \rightarrow \Sigma'_0; \Sigma_0 = \Sigma - \{13, 23\}$. The supremal relative observable supervisor, SUP is shown in Fig. 12, and its corresponding partial observation supervisor SUPO is shown in Fig. 13. The reduced supervisor, in which unobservable events 13, 23 are looped at state 1, is shown in Fig. 14. Moreover, events 15, 25 are self-looped at all states of the reduced supervisor (hence, they are not shown). Thus, the supervisor is normal w.r.t. $(L_\text{re}(G), P_0)$, where $P_0: \Sigma^* \rightarrow \Sigma'_0, \Sigma_0 = \Sigma - \{15, 25\}$. It can be checked that $P_0(RSUP_0)$ and $P_0(SUP)$ are isomorphic. Moreover, if the supervisor does not observe events 13, 23, they cannot be disabled at states 0, 2 in RSUP$_0$. It means that, they appear as self-loop transitions at states 0, 2. But the state size of the reduced supervisor does not change.

**Example 5:** Supervisory control of AGV under partial observation

A work cell consists of two machines $M_1, M_2$ and an automated guided vehicle AGV as shown in Fig. 15. AGV can be loaded with a work piece either from $M_1$ (event 10) or from $M_2$ (event 22), which it transfers respectively to $M_2$ (event 21) or to an output conveyor (event 30) [17]. Let $CELL = \text{sync}(M_1, M_2, \text{AGV})$. We can see $CELL$ is blocking in state 9, i.e. the sequence of events reaches to a state from which no further transitions are possible (Fig. 16). To prevent blocking, we define $SPEC = \text{trim}(CELL)$, as an appropriate specification (Fig. 17). The supremal relative observable supervisor, SUP is shown in Fig. 18, and its corresponding partial observation supervisor SUPO is shown in Fig. 19. In Fig. 19, states 0, 3 and states 1, 2 are control consistent, respectively. Thus, the partial observation based reduced supervisor, RSUP$_0$ is as shown in Fig. 20. Assume $P_0: \Sigma^* \rightarrow \Sigma'_0, \Sigma_0 = \Sigma - \{11\}$, we can easily check that Corollary 1 is satisfied for SUP and RSUP$_0$.

**V. CONCLUSIONS**

This paper addresses an extension to supervisor reduction procedure, proposed in [9], by considering partial observation; namely not all events are observable. We reduced a feasible partial observation supervisor instead of the original one. In the resulting reduced supervisor, only observable events can cause state changes. We finally clarified the extended theory by some examples.
In this appendix, a quick review of TCT commands is presented. DES = sync(DES1, DES2, ..., DESk) is the synchronous product of DES1, DES2, ..., DESk.

REFERENCES