ON A GENERALIZATION OF ROBUST SUPERVISORY CONTROL OF DISCRETE EVENT SYSTEMS

Ramavarapu S. Sreenivas [*,1]

[*] *CSL & Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.*

Abstract: In this paper we consider a generalization of the robust supervisory control problem introduced by Lin (Lin, 1993), improved by Takai (Takai, 2000; Takai, 2002), Park and Lim (Park and Lim, 2000; Park and Lim, 2002) and Cury and Krogh (Cury and Krogh, 1999). Following the formalism in (Lin, 1993), we suppose the plant language $L(G) \subseteq \Sigma^*$ belongs to a finite collection of non-empty, prefix-closed languages $L(G) \in \{L_1, L_2, \ldots, L_n\}$, where $L_i (\neq \emptyset) \subseteq \Sigma^*, i \in \{1, 2, \ldots, n\}$. The event-set Σ is partitioned into controllable (Σ_c) and uncontrollable (Σ_u) subsets respectively. We assume all events are observable, and the supervisor has no prior knowledge as to the value of $L(G) \in \{L_1, L_2, \ldots, L_n\}$. For each $L_i \subseteq \Sigma^*$ we suppose there exists a prefix-closed language $K_i \subseteq L_i$. We present three conditions that are necessary and sufficient for the existence of a supervisor that enforces K_i if the plant language $L(G) = L_i$. It is possible that for a given choice of the sets $\{L_1, L_2, \ldots, L_n\}$ and $\{K_1, K_2, \ldots, K_n\}$, the conditions identified in this paper are not satisfied. This calls for finding a $\{\widehat{K}_1, \widehat{K}_2, \ldots, \widehat{K}_n\}$, such that $\forall i \in \{1, 2, \ldots, n\}, \widehat{K}_i \subseteq K_i$ that meets the required conditions, and each $\widehat{K}_i$ satisfies some property that we might be interested in. The search for a satisfactory $\{\widehat{K}_1, \widehat{K}_2, \ldots, \widehat{K}_n\}$ using the notion of *monotone properties* is also presented.

Keywords: Discrete Event Systems, Inductive Learning

1. INTRODUCTION

Inspired by the results in the literature on language identification and query-based learning (Angluin, 1980; Angluin, 1987; Osherson *et al.*, 1986), in this paper we explore the problem of learning supervisory control policies using evidential states in the form of event-strings generated by the plant. The (unknown) plant-behavior belongs to a finite set of language-choices. We associate a (different) desired-behavior for each choice of the unsupervised, plant-behavior that has to be enforced via supervision.

[1] This work was supported in part by the National Science Foundation under grant ECS-0000938 and the Office of Naval Research under grant N00014-99-1-0696.

2. DEFINITIONS AND PROBLEM STATEMENT

Following Ramadge and Wonham (Ramadge and Wonham, 1987), we define a *controlled DEDS* or a *plant* as a 4-tuple $G = (Q, \Sigma, \delta, q_0)$, where Q is a (possibly infinite) set of *states*, $q_0 \in Q$ is the *initial state*, Σ is a finite alphabet used to label transitions between states (or the set of *events*), and $\delta : Q \times \Sigma \rightarrow Q$ is a partial function that describes the dynamics of the system. Events are assumed to be instantaneous and asynchronous. We extend the function $\delta(\bullet, \bullet)$ to a function $\delta^* : Q \times \Sigma^* \to Q$ in the usual way. The language generated by G is denoted by the symbol $L(G)$, where $L(G) = \{\omega \in \Sigma^* \mid \delta^*(q_0, \omega) \in Q\}$. Note that $L(G)$ is prefix-closed. That is, if a string $\omega \in L(G)$ then all of its prefixes, $pr(\omega)$, are also in $L(G)$ (i.e. $pr(\omega) \subseteq L(G)$).

To control the plant we assume the set Σ is partitioned into two sets Σ_c and Σ_u, where Σ_c, is the set of events that can be disabled by an external agent. We let Γ denote the set of *control patterns*, where $\Gamma = \{\gamma \mid \gamma : \Sigma \to \{0, 1\}$ and $\gamma(\sigma) = 1$ for each $\sigma \in \Sigma_u\}$. An event $\sigma \in \Sigma$ is said to be *enabled* by γ when $\gamma(\sigma) = 1$, and is *disabled* otherwise. A supervisor Θ is a system that changes the control pattern dynamically. An automata-based supervisor Θ is a pair (S, ϕ), where $S = (X, \Sigma, \xi, x_0)$ is an automaton with a (possibly infinite) state set X, input alphabet Σ, partial transition function $\xi : X \times \Sigma \to X$, initial state x_0, and $\phi : X \to \Gamma$. The supervisor state transitions are synchronous with identically labelled events in the plant G. At each state of S a control pattern is selected based on the ϕ function. We use the symbol $G \mid \Theta$ to represent the closed-loop plant-supervisor system described above, and the symbol $L(G \mid \Theta)$ to represent the language generated by the system $G \mid \Theta$.

To have a well-defined closed-loop behavior a *completeness condition* (Ramadge and Wonham, 1987) must be imposed on the supervisor Θ, namely, $\forall \omega \in \Sigma^*$ and $\sigma \in \Sigma$ the following statement must be true: $(\omega \in L(G \mid \Theta)) \wedge \omega\sigma \in L(G)) \wedge (\phi(\xi^*(x_0, \omega))(\sigma) = 1) \Rightarrow \omega\sigma \in L(G \mid \Theta)$, where $\xi^* : X \times \Sigma^* \to X$ is the transitive-extension of $\xi : X \times \Sigma \to X$ to strings in Σ^*. A prefix-closed language $K \subseteq \Sigma^*$ is said to be *controllable* with respect to G if

$$K\Sigma_u \cap L(G) \subseteq K.$$

There exists a complete supervisor Θ such that $L(G \mid \Theta) = K$ if and only if the language $K \subseteq \Sigma^*$ is (Ramadge and Wonham, 1987)

(1) prefix-closed and controllable with respect to G, and
(2) $K \subseteq L(G)$.

The supervisor Θ thus prevents the generation of strings in $L(G)$ that are not in K, while making sure all strings in K can be generated by the system $G \mid \Theta$.

A *minimally restrictive* supervisor $\widehat{\Theta}$, where $L(G \mid \widehat{\Theta}) = K$, is one where a controllable event $\sigma_c \in \Sigma_c$ is prevented from occurring following the occurrence of a string of events $\omega \in \Sigma^*$ if and only if $\omega\sigma_c \notin K$ and $\omega\sigma_c \in L(G)$. Alternately, if $\omega\sigma_c \notin L(G)$, or $\omega\sigma_c \in K$, the minimally restrictive supervisor will permit the occurrence of σ_c after the string ω. It is easy to show that there is a minimally restrictive supervisor $\widehat{\Theta}$ such that $L(G \mid \widehat{\Theta}) = K$ if and only if there is a supervisor Θ such that $L(G \mid \Theta) = K$.

In the next section we use supervisors that can switch from enforcing a prefix-closed, language $K_i \subseteq \Sigma^*$ to enforcing a different prefix-closed, language $K_j \subseteq \Sigma^*$ after observing a string $\omega\sigma \in \Sigma^*$, where $\omega \in K_i$ and $\omega\sigma \in K_j$. The semantics of supervision can be informally described as follows – if the event string that has been observed until now is ω, we maintain a "bank" of minimally restrictive supervisors where each member enforces a candidate $K_m \in \{K_1, K_2, \ldots, K_n\}$, where $\omega \in K_m$. Note that since the set $\{K_1, K_2, \ldots, K_n\}$ consists of prefix-closed languages, once a supervisor is eliminated from this "bank," it remains eliminated for all subsequent instants.

The current state of the minimally restrictive supervisor that enforces K_i is used to determine the controllable events that are to be disabled after the string $\omega \in K_i$ has been observed (cf. figure 1). After the occurrence of the string $\omega\sigma \in K_j$ ($\Rightarrow \omega \in K_j$) the minimally restrictive supervisor that enforces K_j is used to determine the controllable events of the plant that are to be disabled. It is possible to describe this "bank" of supervisors and the associated switching logic as a monolithic supervisor in the paradigm of reference (Ramadge and Wonham, 1987). We refrain from presenting this laborious, yet routine construction, in the interest of space.

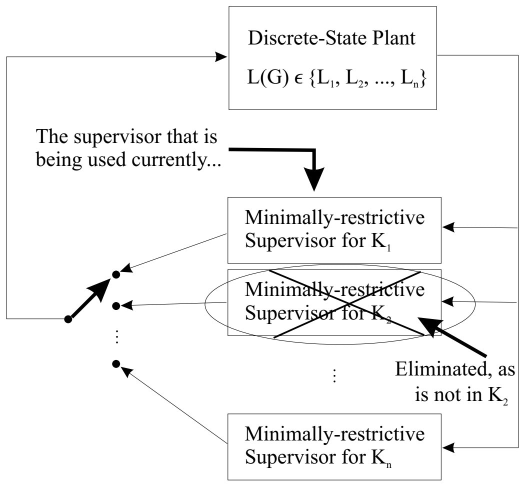

Fig. 1. A pictorial description of the adopted paradigm.

We consider the problem when the prefix-closed, plant language $L(G)$ could take on any value from the set $\{L_1, L_2, \ldots, L_n\}$, where each $L_i \subseteq \Sigma^*(L_i \neq \emptyset)$ $(i \in \{1, 2, \ldots, n\})$ is a different non-empty, prefix-closed language. Let us also suppose that we associate a prefix-closed language $K_i \subseteq \Sigma^*$ for each $i \in \{1, 2, \ldots, n\}$. It is of interest to synthesize a supervisor Θ such that $L(G \mid \Theta) = K_i$ if $L(G) = L_i$. That is, the supervisor Θ should enforce a closed-loop behavior of K_i if the prefix-closed, plant language $L(G)$ is L_i.

Any supervisor that solves the above mentioned problem should be capable of "knowing" what the true value of the plant language $L(G) \in \{L_1, L_2, \ldots, L_n\}$ is, by observing a collection of strings. To facilitate further discussion in this context, we review some relevant results from the area of inductive learning of languages in the next section. These issues are revisited in section 5, where the differences between the paradigms of inductive learning and supervisory control are highlighted.

3. INDUCTIVE LEARNING OF LANGUAGES

The theory of inductive learning of a language $L \subseteq \Sigma^*$ has to do with identifying (the model of) the language $L \subseteq \Sigma^*$ when presented with labelled-evidence of the form $\{(\omega_i, b_i)\}_{i=1}^{\infty}$, where $\omega_i \in \Sigma^*$ is an arbitrary string, and $b_i \in \{0, 1\}$ is a binary-variable that is assigned a value of 1 (0) if the string $\omega_i \in L$ ($\omega_i \notin L$).

The early literature on language identification required the learner to eventually arrive at the correct model for the unknown language $L \subseteq \Sigma^*$ when only positive examples of a language are presented to her (i.e. $\forall i, b_i = 1$). We now present some additional details of the assumptions in this framework. Regarding the data presented to the learner – there might be some repetitions, but in the limit all strings in L are assumed to be presented as evidence. Attention is restricted to a learner who is permitted to vacillate for an unknown period of time during the early stages of learning, but following this vacillation, she has to settle into a hypothesis as to what she thinks $L \subseteq \Sigma^*$ is (cf. (Osherson *et al.*, 1986)). We now present a necessary and sufficient condition for a (possibly infinite) collection of languages $\{L_1, L_2, \ldots\}$ to be identifiable in the above mentioned context.

Proposition 1. (Angluin, 1980) Let Σ be a finite-set of symbols. A (possibly infinite) collection of languages $\{L_1, L_2, \ldots\}(L_i \subseteq \Sigma^*)$, is identifiable if and only if for all $L_i \in \{L_1, L_2, \ldots\}$ there is a finite subset $D_i \subseteq L_i$ such that for all $L_j \in \{L_1, L_2, \ldots\}$, if $D_i \subseteq L_j$, then $L_j \not\subset L_i$.

The formal proof of necessity (cf. section 2.4, (Osherson *et al.*, 1986)) calls upon the fact that there is a specific finite set of evidence, which when presented to any successful learner will cause her to *lock* her hypothesis (Blum and Blum, 1975) regarding the source of presented evidence. There will be no vacillations afterwards. Suppose $\widehat{D}_i \subseteq L_i$ is the finite evidence that locks the learner's hypothesis as the source of evidentiary data to be L_i. A contrapositive proof of necessity assumes there is a L_j such that $\widehat{D}_i \subseteq L_j$ and $L_j \subset L_i$. If the source of evidentiary data is indeed L_j, the learner would have incorrectly locked her hypothesis to L_i, completing the informal proof of the necessity proposition 1.

To establish sufficiency, suppose there is a finite set $D_i \subseteq L_i$ for each $i \in \{1, 2, \ldots\}$ such that $\forall j \in \{1, 2, \ldots\}(D_i \subseteq L_j \Rightarrow L_j \not\subset L_i)$. Also, let us suppose that up to the l-th instant, the learner has been presented with evidentiary data $E_l \subseteq \Sigma^*$. The learner can hypothesize that the source of evidentiary data is L_k, where k is the least index satisfying the requirement $D_k \subseteq E_l \subseteq L_k$, if it exits. If there is no suitable value for k, the learner returns the null hypothesis.

Since all strings from L_i are presented to the learner in the limit, there will be instant $m \in \mathcal{N}$ such that for $l \geq m$, $D_i \subseteq E_l$. If the learner incorrectly hypothesizes the source of evidentiary data to be L_j instead of L_i (say, because $D_j \subseteq D_i$ and $E_l \subseteq L_j$), then it must be that $j < i$. Notice that there will be instant in the future where the learner will be presented with a string from $L_i - L_j$ (as all strings in L_i are presented to the learner eventually). At this point the learner will change her hypothesis that the source of evidentiary data is L_j to some L_m where $m > j$. Note that if $m = i$, the learner does not change her hypothesis on additional evidence. If $m \neq i$, repeating the above argument it can be shown that the index of the hypothesized language gets closer to i, eventually resulting in the learner presenting the correct hypothesis and the identification of the source of evidentiary data will be completed in finite time as a string that is unique to L_i (as compared to all $L_j, j < i$) is presented as evidence.

Variations of the above theme include – learning in environments where negative examples are also presented as evidentiary data; *query-based learning*, where the learner is permitted to ask *membership* and *equivalence-queries* to a *minimally adequate teacher* (cf. (Angluin, 1987)).

4. MAIN RESULT

A necessary and sufficient condition for the existence of Θ is presented in this section. During the early-stages of evidence-gathering, the supervisor will be unsure as to what the true value of $L(G)$ is. But, during this period of uncertainty, it is possible to synthesize a supervisor that does not compromise the requirement that the closed-loop behavior be K_i, if $L(G) = L_i$. After some finite observations are made regarding the plant-language $L(G)$, the supervisor will be able to infer that $L(G) = L_i$ and following this inference the problem becomes an instance of the problem considered by Wonham and Ramadge (Ramadge and Wonham, 1987).

Theorem 2. Let $\Sigma = \Sigma_u \cup \Sigma_c$, where $\Sigma_c \cap \Sigma_u = \emptyset$ be a set of events, where Σ_c (Σ_u) represents the controllable (uncontrollable) subset. Suppose the prefix-closed, plant language $L(G) \subseteq \Sigma^*$ is unknown, and belongs to the set $\{L_1, L_2, \ldots, L_n\}$, where $\forall i \in \{1, 2, \ldots, n\}, L_i \subseteq \Sigma^*$, and let also suppose that $\forall i \in \{1, 2, \ldots, n\}, \exists K_i \subseteq \Sigma^*$. There is a supervisor Θ such that $(L(G) = L_i) \Leftrightarrow (L(G \mid \Theta) = K_i)$ if and only if the following conditions are true:

(1) $\forall i \in \{1, 2, \ldots, n\}, K_i(\neq \emptyset) \subseteq L_i$,
(2) $\forall i \in \{1, 2, \ldots, n\}, K_i$ is controllable with respect to L_i, and
(3) $\forall i, j \in \{1, 2, \ldots, n\}, (K_i - K_j) \cap L_j = \emptyset$.

PROOF. The necessity of the first requirement follows from the fact that if $K_i = \emptyset$, there cannot be a supervisor that enforces K_i when $L = L_i(\neq \emptyset)$.

The necessity of the second requirement can be established by a contrapositive argument. If $K_j \neq \emptyset$ is not controllable with respect to L_j for some value of j, then following the results in (Ramadge and Wonham, 1987), there cannot be a supervisor that enforces K_i if $L = L_i$.

The necessity of the third requirement listed above can be argued as follows – if $\exists \omega \in (K_i - K_j) \cap L_j$, for some i, j, then $\omega \in K_i \cap L_j$ and ω is legal (illegal) if $L = L_i$ ($L = L_j$). Without loss in generality, we can suppose $\omega = \widehat{\omega}\sigma$, and $\widehat{\omega} \in K_i \cap K_j$. If K_j is controllable with respect to L_j, and $\omega(= \widehat{\omega}\sigma) \in L_j - K_j$, it follows that $\sigma \in \Sigma_c$. Any controller Θ we intend to use would be ambivalent about permitting the event $\sigma \in \Sigma_c$ after it has seen $\widehat{\omega}$. If $L = L_i$ ($L = L_j$) then $\sigma \in \Sigma_c$ should (not) be permitted after $\widehat{\omega}$. Since the occurrence of $\widehat{\omega}$ contains insufficient information for any supervisor to determine if $L = L_i$ or $L = L_j$, it follows that the problem does not have a solution.

For sufficiency, we will show that it is possible to synthesize a supervisor Θ that guarantees the closed-loop behavior is K_i when $L = L_i$. Suppose, the supervisor Θ has observed strings in the finite set $pr(\omega) \subseteq L(G \mid \Theta)$ thus far, where $pr(\bullet)$ denotes the prefix-set of a string argument. It hypothesizes that the plant language is $L_m \in \{L_1, L_2, \ldots, L_n\}$, and enforces K_m [2] using a minimally restrictive supervisor, where m is smallest index satisfying the requirement that $\omega \in K_m \Rightarrow pr(\omega) \subseteq K_m$.

Suppose the true, prefix-closed, plant language $L(G)$ is L_i ($i \neq m$), in which case Θ is supposed to enforce K_i in place of K_m.

We now show that any incorrect choice of the closed-loop behavior of K_m in place of K_i does not have a pernicious effect. We have $\omega \in K_m \cap L_i$, and since, $(K_m - K_i) \cap L_i = \emptyset$, it follows that $\omega \in K_i$. If $\omega\sigma$ is permitted by Θ (as $\omega\sigma \in K_m$) and $\omega\sigma \notin K_i$ then it must be that $\omega\sigma \notin L(G)(= L_i)$ (otherwise, $\omega\sigma \in (K_m - K_i) \cap L_i \Rightarrow (K_m - K_i) \cap L_i \neq \emptyset$). That is, even though $\omega\sigma$ is permitted by Θ, it will never be produced by the closed-loop system.

Now, suppose $\omega\sigma \in (K_i - K_m)$. It follows that $\omega\sigma \notin L_m$. This is because, if $\omega\sigma \in L_m$ then $(K_i - K_m) \cap L_m \neq \emptyset$. Since Θ is a minimally restrictive supervisor that enforces K_m, and $\omega\sigma \notin L_m$, the event σ will be permitted by Θ following the event-string ω. Since $K_i \subseteq L_i(= L(G))$ it follows that the event string $\omega\sigma$ can occur under supervision.

Let us suppose $\omega\sigma$ occurs under the supervision of Θ, which has hypothesized that $L(G) = L_m$ after observing $\omega \in K_m(\Rightarrow pr(\omega) \subseteq K_m)$. If $\omega\sigma \in K_m$ then Θ continues with the hypothesis that $L(G) = L_m$ and enforces K_m at the next-step.

If $\omega\sigma \notin K_m$ then it follows that $\omega\sigma \notin L_m$, as $\omega \in K_m$ and K_m is controllable with respect to L_m. In this case, Θ enforces K_l where l is the smallest index satisfying the requirement $\omega\sigma \in K_l(\Rightarrow pr(\omega\sigma) \subseteq K_l)$. The existence of the index l is guaranteed as $l = i$ meets the required condition. Additionally, due to the prefix-closed nature of the languages in the set $\{K_1, K_2, \ldots, K_n\}$, we have $l > m$. Also, if $l = i$ then subsequent observations of event strings under supervision will not cause a change in the hypothesis that $L(G) = L_l = L_i$. If $l \neq i$, then we can effectively replace the index l by m and repeat the above observations, until $l = i$. ♣

5. DISCUSSION

It is important to note that the literature in the inductive learning of languages does not necessarily consider a finite set of prefix-closed languages from which evidentiary data is presented to a learner. Also, the literature in inductive learning assumes that every string is eventually presented as evidence to the learner. This guarantees the fact that any finite subset of a language is covered by evidentiary data in finite time. Un-

[2] Note that $K_m(\neq \emptyset) \subseteq L_m$ and K_m is controllable with respect to L_m, as per the conditions of the theorem

der this paradigm, a finite set of distinct [3] languages $\{L_1, L_2, \ldots, L_n\}$ is always identifiable. To see this, note that for each $L_i \in \{L_1, L_2, \ldots, L_n\}$ we can construct a finite set $D_i \subseteq L_i$ where $D_i \not\subseteq L_j, \forall j \in \{1, 2, \ldots, n\} - \{i\}$, as follows $D_i = \cup_{j \in \{1,2,\ldots,n\}-\{i\}} \{\omega_{i,j}\}$, where $\omega_{i,j}$ is a string in the (non-empty) set $L_i - L_j$.

In the paradigm of supervisory control, evidentiary data is presented in the form of a single string (and its prefix-set). There is no guarantee that all strings in the unknown language will be eventually presented as evidence. Additionally, the supervisor is permitted to (possibly indefinitely) hold an incorrect hypothesis regarding the exact identity of the plant, as long as the desired specifications are not compromised. These aspects differentiate the material presented in this paper from the literature on inductive learning.

As an illustration, suppose $L_1, L_2 (L_1 \neq L_2) \subseteq \Sigma^*$ are two languages such that $L_1 \subset L_2$. Since a string (and its prefix-set) in L_1 could be the only evidence presented, a learner cannot necessarily identify if $L(G) = L_1$ or $L(G) = L_2$ in finite time. Let us suppose that the goal of supervision is to enforce $K_1 \subseteq L_1$ if $L(G) = L_1$ and $K_2 \subseteq L_2$ if $L(G) = L_2$. Let us also suppose that $K_1 (\neq \emptyset) \subseteq L_1$, $K_2 (\neq \emptyset) \subseteq L_2$, K_1 is controllable with respect to L_1 and K_2 is controllable with respect to L_2. Theorem 2 indicates that there is a supervisor Θ that meets the requirements if and only if $(K_2 \cap L_1) \subseteq K_1 \subseteq K_2$.

To see this, note that as $K_1 \subseteq L_1 \subset L_2$ and $K_2 \subseteq L_2$, $(K_1 - K_2) \cap L_2 = \emptyset \Rightarrow (K_1 - K_2) = \emptyset \Rightarrow K_1 \subseteq K_2$, and $(K_2 - K_1) \cap L_1 = \emptyset \Rightarrow K_2 \cap L_1 \subseteq K_1$.

The supervisory strategy that will achieve the required objective will initially enforce K_1 using a minimally restrictive supervisor, under the hypothesis that $L(G) = L_1$, until a string in $K_2 - L_1$ is observed, at which point the supervisor changes its hypothesis to $L(G) = L_2$ and enforces K_2. Note that during the phase when it is hypothesized that $L(G) = L_1$, the minimally restrictive supervisor does not prevent the occurrence of any string in $K_2 - L_1$.

It is possible that for a given choice of the sets $\{K_1, K_2, \ldots, K_n\}$ and $\{L_1, L_2, \ldots, L_n\}$, one or more of the three conditions identified earlier are not satisfied. This calls for finding a $\{\widehat{K}_1, \widehat{K}_2, \ldots, \widehat{K}_n\}$, such that $\forall i \in \{1, 2, \ldots, n\}, \widehat{K}_i \subseteq K_i$ that meets the conditions listed above, and each $\widehat{K}_i$ satisfies some property that we might be interested in. The rest of the paper formalizes this search for a satisfactory $\{\widehat{K}_1, \widehat{K}_2, \ldots, \widehat{K}_n\}$.

[3] $\forall i, j \in \{1, 2, \ldots, n\}, (L_i = L_j) \Rightarrow (i = j)$.

We define a *property* of a prefix-closed language as a function $\mathcal{P} : 2^{\Sigma^*} \rightarrow \{0, 1\}$. The implication being, if for a prefix-closed $\widehat{L} \subseteq \Sigma^*, \mathcal{P}(\widehat{L}) = 1$ (0), then the language $\widehat{L}$ has (does not have) the property $\mathcal{P}$.

We say a property $\mathcal{P}$ is *monotone* (with respect to set-containment) if for two prefix-closed languages $\widehat{L}_1, \widehat{L}_2 \subseteq \Sigma^*$, $(\mathcal{P}(\widehat{L}_1) = 1) \bigwedge (\mathcal{P}(\widehat{L}_2) = 1) \Rightarrow (\mathcal{P}(\widehat{L}_1 \cup \widehat{L}_2) = 1)$. If $\mathcal{P}_1$ and $\mathcal{P}_2$ are two monotone properties, then the conjunction, $\mathcal{P}_{1\wedge 2} : 2^{\Sigma^*} \rightarrow \{0, 1\}$, where $\mathcal{P}_{1\wedge 2}(L) = \mathcal{P}_1(L) \times \mathcal{P}_2(L)$ is also monotone. The disjunction of $\mathcal{P}_1$ and $\mathcal{P}_2$, defined in the usual sense, is also monotone. Given a language L there is a unique, maximal subset of L that satisfies a monotone property $\mathcal{P}$.

An example of a prefix-closed, monotone specification that arises in DEDS could be a liveness specification (cf. section VI, (Sreenivas, 1997*b*)). The notion of non-blocking behaviors can also be represented as a monotone property involving prefix-closed languages. For instance, if a subset $L_m \subseteq L$, of the prefix-closed, plant language are labelled as *marked* strings. We can capture the notion of non-blocking sublanguages of L by a monotone property $\mathcal{P}_{L_m}$, where for a prefix-closed $\widehat{L}$, $(\mathcal{P}_{L_m}(\widehat{L}) = 1) \Leftrightarrow (\forall \omega_1 \in \widehat{L}, \exists \omega_2 \in \Sigma^*, (\omega_1 \omega_2 \in \widehat{L}) \bigwedge (\omega_1 \omega_2 \in L_m))$. The notion of controllable sub-behaviors can also be represented using a monotone property using well-known results in the DEDS-literature. For instance, given a prefix-closed, plant language $L \subseteq \Sigma^*$, and a partition of the event-set as $\Sigma = \Sigma_u \cup \Sigma_c$, we can define a property $\mathcal{P}_{(L,\Sigma_u,\Sigma_c)}(K)$ of a prefix-closed, language $K \subseteq \Sigma^*$ as $(\mathcal{P}_{(L,\Sigma_u,\Sigma_c)}(K) = 1) \Leftrightarrow ((K \subseteq L) \bigwedge (K\Sigma_u \cap L \subseteq K))$. It is important to note that there are prefix-closed specifications that arise in DEDS that are not monotone (cf. (Sreenivas, 1997*a*; Thistle, 1991)).

Given an arbitrary, monotone property $\mathcal{P}$, a prefix-closed plant language $L \subseteq \Sigma^* (\Sigma = \Sigma_c \cup \Sigma_u)$, the conjunction of $\mathcal{P}$ and $\mathcal{P}_{(L,\Sigma_u,\Sigma_c)}$, denoted by $\mathcal{P}_c(\widehat{L}) = \mathcal{P}(\widehat{L}) \times \mathcal{P}_{(L,\Sigma_u,\Sigma_c)}(\widehat{L})$, is a monotone property that assigns a value of unity only to those prefix-closed subsets of L that satisfy property $\mathcal{P}$, and are controllable with respect to L. A monotone property $\mathcal{P} : 2^{\Sigma^*} \rightarrow \{0, 1\}$ can be lifted to a property $\mathcal{P}^n : (2^{\Sigma^*})^n \rightarrow \{0, 1\}$ that returns a value of 0 or 1 for n-tuples of prefix-closed languages as follows $\mathcal{P}^n(\widehat{L}_1, \widehat{L}_2, \ldots, \widehat{L}_n) = \prod_{i=1}^{n} \mathcal{P}(\widehat{L}_i)$. Given two n-tuples of languages $(\widehat{L}_1, \widehat{L}_2, \ldots, \widehat{L}_n)$, $(\widetilde{L}_1, \widetilde{L}_2, \ldots, \widetilde{L}_n)$, we define $(\widehat{L}_1, \widehat{L}_2, \ldots, \widehat{L}_n) \cup (\widetilde{L}_1, \widetilde{L}_2, \ldots, \widetilde{L}_n) = (\widehat{L}_1 \cup \widetilde{L}_1, \widehat{L}_2 \cup \widetilde{L}_2, \ldots, \widehat{L}_n \cup \widetilde{L}_n))$. Under this definition of set-union for n-tuples of languages, we can show that $\mathcal{P}^n$ is monotone, as $\mathcal{P}$ is monotone. That is, $(\mathcal{P}^n(\widehat{L}_1, \ldots, \widehat{L}_n) =$

$1) \wedge (\mathcal{P}^n(\widetilde{L}_1, \ldots, \widetilde{L}_n) = 1) \Rightarrow \mathcal{P}^n((\widehat{L}_1, \ldots, \widehat{L}_n) \cup (\widetilde{L}_1, \ldots, \widetilde{L}_n))(= \mathcal{P}^n(\widehat{L}_1 \cup \widetilde{L}_1, \ldots, \widehat{L}_n \cup \widetilde{L}_n)) = 1$.

Let us suppose we are given an n-tuple of prefix-closed, plant languages $(L_1, L_2, \ldots, L_n)(L_i \neq \emptyset, \forall i \in \{0, 1, \ldots, n\})$. We define a property $\widetilde{\mathcal{P}} : (2^{\Sigma^*})^n \rightarrow \{0, 1\}$, where $\widetilde{\mathcal{P}}(\widehat{L}_1, \ldots, \widehat{L}_n) = \mathcal{P}_c^n(\widehat{L}_1, \ldots, \widehat{L}_n) \times \widehat{\mathcal{P}}(\widehat{L}_1, \ldots, \widehat{L}_n)$ where $(\widehat{\mathcal{P}}(\widehat{L}_1 \ldots) = 1) \Leftrightarrow (\forall i, j \in \{1, 2, \ldots, n\}, (\widehat{L}_i - \widehat{L}_j) \cap L_j = \emptyset)$. First, we note that $\widehat{\mathcal{P}}(\widehat{L}_1, \widehat{L}_2, \ldots, \widehat{L}_n)$ is monotone. To see this, note that if $\widehat{\mathcal{P}}(\widehat{L}_1, \widehat{L}_2, \ldots, \widehat{L}_n) = 1$ and $\widehat{\mathcal{P}}(\widetilde{L}_1, \widetilde{L}_2, \ldots, \widetilde{L}_n) = 1$, for appropriately defined prefix-closed, language arguments, then it follows that $\forall i, j \in \{1, 2, \ldots, n\}, (\widehat{L}_i - \widehat{L}_j) \cap L_j = \emptyset$ and $(\widetilde{L}_i - \widetilde{L}_j) \cap L_j = \emptyset$. This in turn implies that $\forall i, j \in \{1, 2, \ldots, n\}, \{(\widehat{L}_i \cup \widetilde{L}_i) - (\widehat{L}_j \cup \widetilde{L}_j)\} \cap L_j = \emptyset$. That is, $\widehat{\mathcal{P}}(\widehat{L}_1 \cup \widetilde{L}_1, \widehat{L}_2 \cup \widetilde{L}_2, \ldots, \widehat{L}_n \cup \widetilde{L}_n) = 1$. It is well-known that the property $\mathcal{P}_c(\widehat{L})$ is monotone (Ramadge and Wonham, 1987), and in turn this implies that $\mathcal{P}_c^n(\widehat{L}_1, \widehat{L}_2, \ldots, \widehat{L}_n)$ is also monotone. This, along with the fact that $\widehat{\mathcal{P}}(\widehat{L}_1, \widehat{L}_2, \ldots, \widehat{L}_n)$ is monotone implies that the property $\widetilde{\mathcal{P}}$ is also monotone. Since there is a unique maximal-element (with respect to set-containment) for each monotone property, it follows that the property $\widetilde{\mathcal{P}}(K_1, K_2, \ldots, K_n)$ also has a unique maximal-element. It follows that there is a set of n prefix-closed languages $\{\widehat{K}_1, \widehat{K}_2, \ldots, \widehat{K}_n\}$ such that $\forall i \in \{1, 2, \ldots, n\}$ (i) $\widehat{K}_i \subseteq L_i$, (ii) $\widehat{K}_i$ satisfies the property $\mathcal{P}$, and (iii) there exists a supervisor Θ that enforces $\widehat{K}_i$ if $L = L_i$ if and only if the maximal subset of $\{L_1, L_2, \ldots, L_n\}$ under the monotone property $\widetilde{\mathcal{P}}$ does not contain any empty components.

6. CONCLUSION

We considered a supervisory control problem where the prefix-closed, plant language $L(G) \subseteq \Sigma^*$ is unknown, and belongs to the set $\{L_1, \ldots, L_n\}$, where each $L_i \subseteq \Sigma^*, i \in \{1, 2, \ldots, n\}$, is a prefix-closed language, and has an associated, desired, prefix-closed language $K_i \subseteq \Sigma^*$. The objective is to synthesize a supervisor Θ that enforces $L(G \mid \Theta) = K_i$ if $L(G) = L_i$. We presented a set of necessary and sufficient conditions for the existence of a supervisory controller for this problem. In those problem instances where these conditions are violated, it is of interest to find a set of desired behaviors $\{\widehat{K}_1, \widehat{K}_2, \ldots, \widehat{K}_n,\}$ where $\forall i \in \{1, \ldots, n\}, \widehat{K}_i \subseteq K_i$ and the necessary and sufficient conditions identified in the paper hold. Using the notion of *monotone properties* defined in the paper, we present conditions under which there is a unique "maximal-element" in the set of all possible behaviors that satisfy the aforementioned conditions.

REFERENCES

Angluin, D. (1980). Inductive inference of formal languages from positive data. *Information and Control* **45**, 117–135.

Angluin, D. (1987). Learning regular sets from queries and counterexamples. *Information and Control* **75**, 87–106.

Blum, L. and M. Blum (1975). Toward a mathematical theory of inductive inference. *Information and Control* **28**, 125–155.

Cury, J.E.R. and B.H. Krogh (1999). Robustness of supervisors for discrete-event systems. *IEEE Trans. on Automatic Control* **44**(2), 376–379.

Lin, F. (1993). Robust and adaptive supervisory control of discrete event systems. *IEEE Trans. Automatic Control* **38**(12), 1848–1852.

Osherson, D.N., M. Stob and S. Weinstein (1986). *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. The MIT Press Series in Learning, Development and Conceptual Change. MIT Press. Cambridge, MA.

Park, S.J. and J.T. Lim (2000). Robust non-blocking supervisor for discrete event systems with model uncertainty under partial observation. *IEEE Trans. on Automatic Control* **45**(12), 2393–2396.

Park, S.J. and J.T. Lim (2002). Robust and non-blocking supervisory control of nondeterministic discrete event systems using trajectory models. *IEEE Trans. on Automatic Control* **47**(4), 655–658.

Ramadge, P.J.G. and W.M. Wonham (1987). Supervisory control of a class of discrete-event processes. *SIAM Journal on Control and Optimization* **25**(1), 206–230.

Sreenivas, R.S. (1997*a*). On supervisory policies that enforce global fairness and bounded fairness in partially controlled petri nets. *Discrete Event Dynamic Systems* **7**, 191–208.

Sreenivas, R.S. (1997*b*). On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by controlled petri nets. *IEEE Trans. on Automatic Control* **42**(7), 928–945.

Takai, S. (2000). Robust supervisory control of a class of timed discrete event systems under partial observation. *Systems & Control Letters* **39**(4), 267–273.

Takai, S. (2002). Synthesis of maximally permissive and robust supervisors for prefix-closed language specifications. *IEEE Trans. Automatic Control* **47**(1), 132–136.

Thistle, J.G. (1991). Control of Infinite Behaviour of Discrete-Event Systems. Ph.d. thesis. Systems Control Group, Department of Electrical Engineering, University of Toronto. Toronto, Canada.