Towards Decentralized Synthesis: Decomposable Sublanguage and Joint Observability Problems

Liyong Lin, Alin Stefanescu, Rong Su, Weilin Wang and Ahmad Reza Shehabinia

Abstract—This note studies two closely related decision problems from the area of decentralized synthesis. The first problem is about deciding the existence of a non-empty decomposable sublanguage of a regular language, called Decomposable Sublanguage Problem; the second is Joint Observability Problem, which is known in the literature. We provide characterizations of the decidability of both decision problems. Then, we study the undecidability of related problems such as Distributed Supervisory Control Problem and Parametrized Control Synthesis Problem. To address these undecidability results, we also propose a couple of heuristics that can provide solutions to the two studied problems, but also to more general problems from control or trace theory.

Index Terms – discrete-event systems, decomposability, decentralized observation, decidability, trace theory.

I. INTRODUCTION

Decomposability and joint observability are two important notions in decentralized synthesis. The concept of decomposability, called separability in [3], plays an important role in the characterization of the condition under which distributed control can achieve the optimal global behavior. The notion of joint observability is introduced in [15] and is closely related to decentralized supervisory control.

A language $L$ over $\Sigma$ is said to be decomposable with respect to a distribution, i.e., a collection of non-empty subalphabets, of $\Sigma$ if $L$ equals the synchronous product of its projections onto the respective subalphabets. Two disjoint languages are said to be jointly observable with respect to a distribution if any string in one language could be distinguished from any string in the other language, by at least one local observation point. Both the notions of decomposability and joint observability have been studied in the literature, see for example [3], [2] and [6]. In the top-down supervisor synthesis procedure, the global specification is required to be decomposable into local specifications for later synthesis of local supervisors [4]. However, the condition of decomposability is too strict for the assumption to be practical. There are several ways to get around this difficult issue (see [1] and the references therein, for example). For instance, one can change either the distribution or the specification. The first variant is considered in [1] using the notion of conditional decomposability, which provides the freedom to suitably modify the original distribution such that the specification becomes decomposable with respect to the new distribution. The second variant restricts the global specification language to ensure the decomposability of the resulting language for further synthesis, similar to the approach investigated in [7], [5] for trace closed languages. Since there are essentially only a finite number of distributions available in the first approach and it may not be easy to modify the distribution in practice, we will be more interested in the latter alternative and thus investigate its related decision problem, called Decomposable Sublanguage Problem.

Surveying the existing literature, we found some negative results about both problems. In [6], Joint Observability Problem and its prefix closed case, referred to as Prefix Closed Joint Observability Problem, are both shown to be undecidable. The result is then used in the same paper to show the undecidability of Decentralized Supervisory Control Problem and used in [16] to show the undecidability of decentralized diagnosability. It is also known that Decomposable Sublanguage Problem is undecidable [7], [8], following the construction of [5] (which is again based on the construction of [9] and the references therein). In [10], the authors employ a similar construction to establish the undecidability of Decomposable Sublanguage Problem with symmetry constraint and use the result to show that the problem of synthesizing isomorphic and nonblocking controllers, for a finite number of different discrete event systems with a given global specification is undecidable if the global specification is not decomposable with respect to the distribution. For the case when the global specification is decomposable with respect to the distribution, it is shown in [10] that the problem becomes decidable.

The main results of our paper are described below. We first show that Decomposable Sublanguage Problem is decidable if and only if the independence relation is transitive, and Joint Observability Problem is decidable if and only if the independence relation is a transitive forest. These characterizations rely on results on decision problems in regular trace language theory [14], [19]. To the authors’ knowledge, no such characterizations are available in the control theory field. We thus bridge this gap and transfer results between control and computer science theory.

Then, we show that Distributed Supervisory Control Problem is undecidable if the independence relation is not transitive. We also obtain two related results in control theory that are not subsumed by the above characterizations. On the one hand, we prove the undecidability of Specification Template Synthesis Problem, i.e., the problem of synthesizing a specification template from a schematic specification language...
parameterized by the number of local plants, showing the algorithmic infeasibility of top-down parameterized control synthesis. It could also be used to show the undecidability result of [10] and the undecidability of parameterized control synthesis (without the restriction of top-down methodology). On the other hand, as a step towards the characterization result for Prefix Closed Joint Observability Problem, we show another case (different from that of [6]) where the undecidability may arise, by a simple reduction from the undecidability of Joint Observability Problem.

To cope with all these undecidability results, we propose a couple of simple but efficient heuristics for computing a regular under-approximation of the maximal trace closed sublanguage of a regular language and a decomposable sublanguage of a regular language. The heuristics for the maximal trace closed sublanguage could be used to compute approximately the trace closure of a regular language, and naturally lead to heuristics for the decentralized control problems studied above, but also undecidable problems in trace theory [12]. We then prove a general result that the exactness of any regular under-approximation of the maximal trace closed sublanguage (dually, regular over-approximation of the trace closure) of a regular language is decidable if and only if the independence relation is transitive.

The paper is organized as follows. Section II is devoted to mathematical preliminaries. In Section III, we provide characterizations of the decidability of Decomposable Sublanguage Problem and Joint Observability Problem. Related undecidability results are obtained and heuristics are then developed in Section IV and V. We draw the conclusions in the last section. Some of the proofs were left out due to space constraints. They can be found in the full version of this paper at: http://is.gd/acc14_long

II. Preliminaries

The notations used here are standard in the theory of supervisory control and mostly follow that of [17], [18]. In the following, additional notations and terminologies that are necessary to understand the paper are introduced, mostly about the theory of traces.

Let \([1,n]\) denote the set \(\{1,2,\ldots,n\}\). For a given alphabet \(\Sigma\), a distribution of \(\Sigma\) of size \(n\) is an \(n\)-tuple \(\Delta = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n)\) such that \(\emptyset \neq \Sigma_i \subseteq \Sigma\) for \(i \in [1,n]\). A distribution \(\Delta = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n)\) is said to be proper if \(\Sigma = \bigcup_{i=1}^{n} \Sigma_i, \Sigma_i \neq \Sigma\) for any \(i \in [1,n]\), and no subalphabet is a subset of another one. Thus there are only a finite number of proper distributions. In the rest of the paper, when we talk about a distribution \(\Delta\), we will assume it is proper. Given a distribution \(\Delta = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n)\), we have \(n\) projections \(P_i\) from \(\Sigma^*\) to \(\Sigma_i^*\) and \(n\) inverse projections \(P_i^{-1}\) from \(\Sigma_i^*\) to \(\Sigma^*\). We understand that both projections and inverse projections are naturally extended to map between languages. The synchronous product \(\bigcap_{i=1}^{n} P_i^{-1}(L_i)\) of languages \(L_i\) over \(\Sigma_i\) is defined as \(\bigcap_{i=1}^{n} P_i^{-1}(L_i)\). Given a language \(L\) over \(\Sigma\), \(\bigcap_{i=1}^{n} P_i(L)\) is said to be the decomposition closure of \(L\) with respect to \(\Delta\). We denote the complement of a language \(L\) by \(L^c\). \(\overline{L}\) denotes the prefix closure of \(L\). \(A - B\) or \(A\setminus B\) denotes the set theoretic difference of \(A\) and \(B\).

An independence relation \(I \subseteq \Sigma \times \Sigma\) is a symmetric and irreflexive relation. Two strings \(w, w'\) are said to be trace equivalent with respect to \(I\), denoted by \(w \sim_I w'\), if there exist strings \(v_0, \ldots, v_n\) such that \(w = v_0, w' = v_n\) and for each \(i \in [1,n]\), there exist \(u_i, u'_i\) and \(a_i, b_i\) such that \((a_i, b_i) \in I, v_{i-1} = u_ia_ib_i'v_{i-1}'\) and \(v_i = u_ia_ib_i'v_i\). The set of trace equivalent strings of \(s\) for an independence relation \(I\) is called the trace closure of \(s\), denoted by \([s]_I\) or \([s]\) if \(I\) is clear from the context. The trace closure \([L]\) of a language \(L\) is defined to be the set \(\bigcup_{s \in L} [s]\). A language \(L\) is said to be trace closed if \([L] = L\).

A distribution \(\Delta = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n)\) naturally induces an independence relation \(I_{\Delta}\) in the following way. The reflexive, symmetric relation \(D = \{(a,b) \in \Sigma \times \Sigma \mid \exists i \in [1,n], a,b \in \Sigma_i\}\) is called the dependency relation. Then \(I = \Sigma \times \Sigma - D\) is the independence relation induced by the distribution \(\Delta\).

In the rest of the paper, whenever we are given a distribution \(\Delta\), it is implicitly assumed that the independence relation \(I_{\Delta}\) induced by the distribution is given.

An independence relation \(I \subseteq \Sigma \times \Sigma\) is transitive if \(\forall a,b,c \in \Sigma, (a,b) \in I\) and \((b,c) \in I\) implies \((a,c) \in I\). \(I\) is called a transitive forest, if the undirected graph \((\Sigma, I)\), where \(I \subseteq \Sigma \times \Sigma\) is viewed as the set of undirected edges, does not contain the two graphs \(P_4 = \{(a,b), (b,c), (c,d), (d,a)\}\) or \(C_4 = \{(a,b), (b,c), (c,d), (d,a)\}\) as induced subgraphs [14]. Here \(a, b, c, d\) are all different vertices and the set \((a,b)\), represents the edge between node \(a\) and \(b\).

The following folklore lemmas [25], [7] will be used later.

Lemma 1: The trace closure and decomposition closure of a string are equal, i.e., \([s] = \bigcap_{i=1}^{n} P_i(s)\) for any string \(s\) over \(\Sigma\) and distribution \(\Delta = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n)\) of \(\Sigma\).

Lemma 2: The trace closure of \(L\) is a subset of the decomposition closure of \(L\), i.e., \([L] \subseteq \bigcap_{i=1}^{n} P_i(L)\) for any language \(L\) over \(\Sigma\) and distribution \(\Delta = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n)\) of \(\Sigma\).

III. Characterization Results and Application

A. Characterization of Decomposable Sublanguage Problem

Formally, a language \(L\) over \(\Sigma\) is said to be decomposable with respect to a distribution \(\Delta = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n)\) of \(\Sigma\) if \(L = \bigcap_{i=1}^{n} P_i(L_i)\), i.e., \(L\) is equal to its decomposition closure. Given a distribution, if a regular language is not decomposable, an important question is whether it contains a non-empty decomposable sublanguage, called Decomposable Sublanguage Problem. It is not difficult to see that a language has a non-empty decomposable sublanguage with respect to a given distribution if and only if there is a string in that language such that the decomposition closure of that string is a subset of the given language. Since the decomposition closure and trace closure of a string coincide, by Lemma 1, the original problem becomes deciding the existence of a trace closed sublanguage. Unfortunately, in general, the problem of existence of a trace closed sublanguage is undecidable even when the given language is regular [5], [7]. Thus it follows that Decomposable Sublanguage Problem
is undecidable. Indeed, we can obtain characterization of the decidability of Decomposable Sublanguage Problem. The key observation is that there exists a closed form expression for the maximal trace closed sublanguage of a language that links Decomposable Sublanguage Problem to Universality Problem in trace theory. Universality Problem asks whether the trace closure of a regular language over \( \Sigma \) is \( \Sigma^* \). It is known that Universality Problem is decidable iff \( I \) is transitive [11].

To obtain the characterization, we still need the following fact from [19].

**Lemma 3:** For a given independence relation, the unique maximal trace closed sublanguage of \( L \) is \( [L^r]^c \), for any language \( L \) over \( \Sigma \).

Now it is straightforward to see that \( \exists s \in L, [s] \subseteq L \) iff \( [L^r]^c \neq \emptyset \) iff \( [L^r] \neq \Sigma^* \). That is, Decomposable Sublanguage Problem is equivalent to Non-Universality Problem. Thus, using this equivalence and the result from [11], we obtain the following characterization result.

**Theorem 1:** Decomposable Sublanguage Problem is decidable iff \( I \) is transitive.

We shall note that this characterization result is rather strong. That is, for every fixed independence relation that is not transitive, the problem is undecidable. A similar remark also applies to the characterization result of Joint Observability Problem, to be discussed in Section III-B. Further technical remarks can be found in the appendix.

**B. Characterization of Joint Observability Problem**

Formally, two disjoint languages \( G, B \) over \( \Sigma \) are said to be *jointly observable* with respect to a distribution \( \Delta = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n) \) of \( \Sigma \) if \( \forall s \in G, \forall s' \in B, \exists i \in [1, n], P_i(s) \neq P_i(s') \). The problem whether two disjoint regular languages are jointly observable with respect to a distribution is called Joint Observability Problem. As observed in [6], Joint Observability Problem is equivalent to Disjointness Problem in trace theory.

**Disjointness Problem** asks whether the trace closures of two disjoint regular languages are disjoint for a given independence relation, i.e., whether \( [G] \cap [B] = \emptyset \). It is a standard result that Disjointness Problem is decidable iff \( I \) is a transitive forest [11], [14]. The characterization result for Joint Observability Problem then follows.

**Theorem 2:** Joint Observability Problem is decidable iff \( I \) is a transitive forest.

**Remark:** This result subsumes the undecidability result established in [6]. If we use \( \Sigma_1 = \{a, c\} \) to encode the PCP alphabet and \( \Sigma_2 = \{b, d\} \) to encode the set of new symbols \( \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \) of [6], then the independence relation induced by the distribution \( \Delta = (\{a, c\}, \{b, d\}) \) is indeed not a transitive forest [14].

**IV. UNDECIDABILITY OF RELATED PROBLEMS**

**A. Distributed Supervisory Control Problem**

Decomposable Sublanguage Problem is closely related to Distributed Supervisory Control Problem, since every instance of Distributed Supervisory Control Problem explicitly contains a corresponding instance of Decomposable Sublanguage Problem. By a straightforward reduction from Decomposable Sublanguage Problem, we show that Distributed Supervisory Control Problem is undecidable if \( I \) is not transitive. For that purpose, we formally define Distributed Supervisory Control Problem below (for notations and terminologies we refer the reader to [18]).

**Distributed Supervisory Control Problem:** Given \( n \) plants \( G_i \) over subalphabets \( \Sigma_i \), and a specification language \( L \) over \( \Sigma \), we say that there exist \( n \) non-blocking supervisors \( S_i \) over \( \Sigma_i \) (i.e., observing and controlling only a subset of \( \Sigma_i \)) such that:
1. \( \forall i \in [1, n], L_m(S_i/G_i) \) is a controllable sublanguage of \( L(G_i) \).
2. the \( n \) languages \( L_m(S_1/G_1), L_m(S_2/G_2), \ldots, L_m(S_n/G_n) \) are synchronously non-conflicting.
3. \( \emptyset \neq \bigcap_{i=1}^n L_m(S_i/G_i) \subseteq L^2 \)

**Corollary 1:** Distributed Supervisory Control Problem is undecidable if \( I \) is not transitive.

The above result is not difficult to prove, since condition 3) essentially asks for a decomposable sublanguage of \( L \), under a proper setup for \( G_i, \Sigma_i \). There exist some similar results in the literature concerning the undecidability of decentralized and distributed control [6], [24], [23], [26]. For some comparisons between these undecidability results, see [6], [23]. The setup used here is similar to that of [6]. Compared with these previous results, our sufficient condition for the undecidability is different, being based on the independence relation. Moreover, our undecidability result is stronger, using the additional assumption that \( G \) is structured, i.e., a synchronous product of \( n \) distinct interacting local systems [3].

**B. Parameterized Control Synthesis Problem**

The idea of parameterized control synthesis is to use instantiated controllers from a designed control template to control a parameterized family of discrete event plants in a uniform manner. A natural top-down approach for parameterized control synthesis starts with a schematic regular language (user given specification) \( L(n) \) parameterised by \( n \) (see [13]) and a plant template \( T \) (represented by a deterministic finite automaton (DFA) over the template alphabet); then, a non-empty specification template \( S \) is synthesized (step "Synthesis 1") such that the composition of \( n \) isomorphic copies of \( S \) satisfies the specification \( L(n) \), for any \( n \geq 2 \); finally, a control template \( C \) over the template alphabet is synthesized (step "Synthesis 2") according to the specification template \( S \) and plant template \( T \). We prove that the problem whether there exists a specification template \( S \) as above, i.e., Specification Template Synthesis Problem, is undecidable. Indeed, the result holds even when \( L(n) \) is required to be symmetric and the template alphabet consists solely of private events. By a symmetric specification, we

\[ \text{2Note that we explicitly require that } \bigcap_{i=1}^n L_m(S_i/G_i) \neq \emptyset, \text{ since this is the solution space that we are interested in; otherwise, the problem becomes trivial.} \]
mean a language whose projection into each subalphabet follows the same template. This is relevant, since user given specifications for parameterized control synthesis are usually symmetric, but our result suggests that there is no algorithmic procedure for the synthesis even in this restricted case. Note that any reasonable modelling formalism is capable of expressing private events. So our undecidability result is rather strong. By a straightforward reduction from the undecidability of Specification Template Synthesis Problem, we are able to show the undecidability of Parameterized Control Synthesis Problem, to be formally defined below. To that end, we start with the necessary terminology and notations below.

Let $\Sigma_T = \Sigma_g \cup \Sigma_p$ be the template alphabet [22], where $\Sigma_g$ is the global event set and $\Sigma_p$ is the private event set. A distribution $\Delta_T = \langle \Sigma_1, \Sigma_2, \ldots, \Sigma_n \rangle$ of size $n$ based on template alphabet $\Sigma_T$ is a distribution of $\Sigma(n) := \bigcup_{i=1}^n \Sigma_i$, where $\Sigma_i = \Sigma_g \cup \Sigma_p \times \{i\}$ for every $i \in [1, n]$. Let $h_i : \Sigma_T \mapsto \Sigma_i$ be a bijective map that maps events in $\Sigma_T$ to events in $\Sigma_i$ for each $i \in [1, n]$, such that $h_i(\sigma) = \sigma_i$ if $\sigma \in \Sigma_g$ and $h_i(\sigma) = (\sigma, i)$ if $\sigma \in \Sigma_p$. To simplify the notation, we denote (σ, i) by σi. Depending on the context, $h_i$ may also be extended in the natural way to either a bijective map from $\Sigma_T$ to $\Sigma_i$ or a bijective map from $2^{\Sigma_T}$ to $2^{\Sigma_i}$. A language $L$ over $\Sigma(n)$ is said to be symmetric with respect to $\Delta_T$ if for each $i, j \in [1, n]$, $h_i^{-1}(P_i(L)) = h_j^{-1}(P_j(L))$. The problem whether a regular language over $\Sigma(n)$ is symmetric is obviously decidable.

We employ a reduction from the Post Correspondence Problem (PCP), which is a well known undecidable problem, to Specification Template Synthesis Problem. In the next paragraph, we provide the construction.

Let $A, B$ be two disjoint alphabets and $c$ be a new symbol. Let $\Sigma_T = A \cup B \cup \{c\}$ and $h_i : \Sigma_T \mapsto \Sigma_i$ be a bijective map that maps events in $\Sigma_T$ to events in $\Sigma_i$ such that $h_i(\sigma) = \sigma_i$ for any $\sigma \in \Sigma_T$, for any $i \in [1, n]$. That is, $\Sigma_T = \Sigma_p$, (i.e., there are only private events) and for each $i \in [1, n]$, $h_i^{-1}(\{f_i, g_i\}) = \{c\}$ for all $i \in [1, n]$. Let $f, g : A^* \mapsto B^*$ be two homomorphisms and $f_i, g_i$ be two homomorphisms from $A_i^*$ to $B_i^*$ such that $f_i(\sigma_i) = h_i(f(\sigma))$ and $g_i(\sigma_i) = h_i(g(\sigma))$ for each $\sigma \in A$ and $i \in [1, n]$. For $w \in A^+$, let $s_i^f(w) = c_i^f(h_i(w))h_i(w)h_i(f_i(w))$ and $s_i^g(w) = c_i^g(h_i(w))h_i(w)g_i(h_i(w))c_i^g(h_i(w))$ for each $i \in [1, n]$. Since $|f_i(h_i(w))| = |f(i)|$ and $|g_i(h_i(w))| = |g(i)|$, we have $s_i^f(w) = c_i^f(h_i(w))h_i(w)f_i(h_i(w))c_i^f(h_i(w))$ and $s_i^g(w) = c_i^g(h_i(w))h_i(w)g_i(h_i(w))c_i^g(h_i(w))$. The independence relation $I(n)$ is $\{ (x, y) | x \in \Sigma_i$ and $y \in \Sigma_j$ for different $i$ and $j \}$ and $W_f(n) = \{ s_i^f(w) | s_i^f(w) \subseteq A^+ \}$ and $W_g(n) = \{ s_i^g(w) | s_i^g(w) \subseteq A^+ \}$ are constructed. Then we have the following lemmas, which are used to prove the theorem, i.e., the undecidability of Specification Template Synthesis Problem.

Lemma 4: There exist two regular languages $L_f(n)$ and $L_g(n)$ over $\Sigma(n) = \bigcup_{i=1}^n \Sigma_i$ such that $L_f(n)c = [W_f(n)]$ and $L_g(n)c = [W_g(n)]$ with respect to $I(n)$, for each $n \geq 2$.

Lemma 5: For each $n \geq 2$, there exist $L_f(n)$ and $L_g(n)$ such that $L(n) = \Sigma(n)^c \setminus (L_f(n) \cup L_g(n))$ is symmetric with respect to $\Delta_T$.

The following lemma essentially conveys the main proof idea.

Lemma 6: Let $L(n) = \Sigma(n)^c \setminus (L_f(n) \cup L_g(n))$ for $n \geq 2$, then we have $\exists w \in A^n, f(w) = g(w)$ if for all $n \geq 2$, $W_f(n) \cap W_g(n) \neq \emptyset$ iff $\exists w \in A^n, \forall n \geq 2, \|n\|s_i^f(w) \subseteq L(n)$ iff for all $n \geq 2$, there exists a string $s(n) \in L(n)$ such that $\|n\|s_i^f(s(n)) \subseteq L(n)$ and $\forall i, j \in [1, n], h_i^{-1}(P_i(s(n))) = h_j^{-1}(P_j(s(n)))$ if there exists a non-empty $S \subseteq \Sigma_T$, such that for all $n \geq 2$, $\|n\|h_i(S) \subseteq L(n)$.

Now we are able to establish the undecidability of Specification Template Synthesis Problem.

Theorem 3: Specification Template Synthesis Problem is undecidable. That is, given a schematic regular language $L(n) \subseteq \bigcup_{i=1}^n \Sigma_i$ parameterized by $n$, it is undecidable whether there exists a non-empty specification template $S$ over $\Sigma_T$ such that the composition of $n$ isomorphic copies $h_i(S)$ of $S$ satisfies the specification $L(n)$, i.e., $\|n\|h_i(S) \subseteq L(n)$, for any $n \geq 1$. The undecidability holds even when $L(n)$ is symmetric for any $n \geq 2$, that is, for each $n \geq 2, \forall i, j \in [1, n], h_i^{-1}(P_i(L(n))) = h_j^{-1}(P_j(L(n)))$ and $\Sigma_T = \Sigma_p$.

Remark: The main characteristic of our result is that the component plants consist only of private events and are required to be isomorphic; the schematic specification consists only of private events and is required to be symmetric. Note that the model checking problem for this specific type of parameterized family of systems is trivial and our intended solution, i.e., specification template $S$, works for all $n \geq 2$ simultaneously. From above result, one may deduce the undecidability of [10]. Indeed, we could also show that parameterized control synthesis is undecidable in general (without the requirement of top-down methodology), since each instance of the corresponding problem contains a corresponding instance of Specification Template Synthesis Problem. We formally introduce the definition of Parameterized Control Synthesis Problem below.

Parameterized Control Synthesis Problem: Given a plant template $G$ over template alphabet $\Sigma_T$, and a schematic regular specification language $L(n) \subseteq \bigcup_{i=1}^n \Sigma_i$, decide whether there exist a nonblocking supervisor template $S$ over $\Sigma_T$ (i.e., observing and controlling only a subset of $\Sigma_T$) such that for any $n \geq 2$: 1) $\forall i \in [1, n], L_m(S_i/G_i)$ is a controllable sublanguage of $L(G_i)$, 2) the $n$ languages $L_m(S_1/G_1), L_m(S_2/G_2), \ldots, L_m(S_n/G_n)$ are synchronously non-conflicting, 3) $\emptyset \neq \|n\|L_m(S_i/G_i) \subseteq L(n)$. Here each $S_i$ over $\Sigma_i$ is an isomorphic copy of $S$ and $\Sigma_i$ is an isomorphic copy of $\Sigma_T$.

Corollary 2: Parameterized Control Synthesis Problem is undecidable.
C. Prefix Closed Joint Observability Problem

Clearly Prefix Closed Joint Observability Problem is also decidable if $I$ is a transitive forest. With $\Sigma_1 = \{a, c\}$ that encodes the PCP alphabet, $\Sigma_2 = \{b, d\}$ that encodes $\{a_1, a_2, \ldots, a_n\}$ and $\Sigma_3 = \{e, m\}$ that encodes $\{b_1, b_2, \ldots, b_n\}$, the result of [6] indeed shows that every instance of the PCP problem could be encoded as an instance of Prefix Closed Joint Observability Problem if the graph of the independence relation contains an induced subgraph that corresponds to the distribution $\Delta = \{(a, c),\{b, d\},\{e, m\}\}$.

It is possible to show that Prefix Closed Joint Observability Problem as formulated in [6] is equivalent to the following form: Given two disjoint (with the exception of the empty string $e$) prefix closed regular languages $G', B'$, i.e. $G' \cap B' = \{e\}$, whether $[G'] \cap [B'] = \{e\}$? Now let $G, B \subseteq \Sigma^*$ be two disjoint languages, $\Delta_1$ a distribution of $\Sigma_1$ and $\Delta_2 = (\Delta_1, \{e\}, \{m\})$ be a distribution of $\Sigma_2 = \Sigma_1 \cup \{e\} \cup \{m\}$, where $e, m$ are two new symbols not in $\Sigma_1$. Then, the following reduction holds, which permits a straightforward extension of the undecidability result of the general case to the prefix closed case.

Proposition 1: $[G]_{I_1} \cap [B]_{I_2} \neq \emptyset$ iff $[[G \cap B]\{mB\}]_{I_2} \neq \emptyset$.

Here, $I_1$ and $I_2$ are the induced independence relations of $\Sigma_1$ and $\Sigma_2$, respectively. This indeed shows that the Prefix Closed Joint Observability Problem is undecidable if the graph of the independence relation contains an induced subgraph that corresponds to the distribution $\Delta = \{(a, c),\{b, d\},\{e\},\{m\}\}$. From the general case to the prefix closed case, both constructions have to add two extra symbols $e, m$. The characterization result seems to be difficult and still remains open. We remark that with above reduction step, it is also possible to show the undecidability even when one of the prefix closed regular languages is fixed. In fact, with $\Delta = \{(a, c),\{b, d\},\{e\},\{m\}\}$, $G' = c(ab+cd)^n$ and $B' = m(u_1u_1w_1 + \ldots + u_nw_n)^n e$, where $u_i$ is over $\{a, c\}$ and $w_i$ is its isomorphic copy over $\{b, d\}$, together encode the PCP instance, i.e. $(G')\cap[B'] \neq \{e\}$ iff $\exists i_1, i_2, \ldots, i_k$ in $[1,n]$, $u_{i_1}u_{i_2}\ldots u_{i_k}$ is isomorphic to $w_{i_1}w_{i_2}\ldots w_{i_k}$. Note that $G'$ is a fixed prefix closed regular language that is independent of the PCP instance. This is essentially a slight modification of the construction suggested in [14].

V. HEURISTICS

The decidability results of the decision problems about trace closure operation on regular languages easily translate to the corresponding results about the maximal trace closed sublanguages of regular languages. In particular, it is undecidable whether the maximal trace closed sublanguage of an arbitrary regular language is regular, since it is undecidable whether the trace closure of an arbitrary regular language is regular [11]. This implies operations that preserve regularity are in general not sufficient for computing the maximal trace closed sublanguage of a regular language.

To circumvent this difficulty, it is worth studying heuristics for computing regular approximations of it and, for the purpose of further synthesis and verification, it is preferable for the regular approximations to be trace closed as well. Firstly, such a heuristics is useful for the synthesis of control and communication scheme for robot motion planning, where the specification is required to be trace closed [7], by synthesizing a trace closed sub-specification from the given specification. Secondly, it could be directly translated to a heuristics for the computation of the trace closure of a regular language, which finds applications in many decision problems in trace theory, in the partial order reduction based model checking and symbolic verification of some classes of mutual exclusion protocols [12]. In the next subsection, we investigate two such heuristics.

A. Approximation Heuristics

Based on Lemma 2, it is straightforward to see that $\|\|_{i=1}^n P_i(L')^c$, denoted by $L_{amt}$, is a regular under-approximation for the maximal trace closed sublanguage of $L$. Such an under-approximation indeed provides a heuristics for Decomposable Sublanguage Problem. If $\|\|_{i=1}^n P_i(L'^c)^c$, which is effectively computable, is non-empty, then $L$ has a non-empty decomposable sublanguage $[s]$, for any $s \in \|\|_{i=1}^n P_i(L'^c)^c$, with respect to $\Delta$. Note that $L_{amt}$ itself may not be decomposable with respect to $\Delta$, since the class of decomposable languages is not closed under complementation.

In [10], the authors propose a scheme for computing a decomposable sublanguage $\|\|_{i=1}^n L_i$ of $L$, where $L_i = P_i(L) - P_i(\|\|_{i=1}^n P_i(L) - L)$. In the following, we propose an improved heuristic. Let $L_k = P_k(L) - P_k(\|\|_{i=1}^n P_i(L) - L)$ for some $k \in [1,n]$ and $L_i = P_i(L)$ for all $i$ in $[1,n]$ with $i \neq k$. Denote $L_k(\|\|_{i\neq k, i \in [1,n]} L_i)$ by $L_{amdt}$. It is easy to see that the following holds.

Proposition 2: $L_{amdt} \subseteq L$.

Thus, $L_{amdt}$ is a decomposable sublanguage of $L$. There exists one degree of freedom in this scheme, i.e., the particular $k$ could be arbitrary chosen among $[1,n]$. Note that $L_{amdt}$ is also a trace closed regular language. As approximations for the maximal trace closed sublanguage, $L_{amdt} = L_k(\|\|_{i\neq k, i \in [1,n]} L_i)$ and $L_{amt} = (\|\|_{i=1}^n P_i(L'^c)^c$ are incomparable. For example, let $L = \{ab, ba\}$ over $\Sigma = \{a, b\}$ and $\Delta = \{(a, \{b\}\}$, then $\|\|_{i=1}^n P_i(L'^c)^c = \emptyset$ but $L_k(\|\|_{i\neq k, i \in [1,2]} L_i) = L$ for either $k = 1, 2$. Let $L = \Sigma^* - \{ab\}$, then $\|\|_{i=1}^n P_i(L'^c)^c = \Sigma^* - \{ab, ba\}$ but $L_k(\|\|_{i\neq k, i \in [1,2]} L_i) = \Sigma^* - \Sigma^*a\Sigma^*$ or $\Sigma^* - \Sigma^*b\Sigma^*$ for $k = 1$ or 2, respectively. In practice, an approximation for the maximal trace closed sublanguage of a regular language could be chosen from the $n+1$ candidates $L_{amt}, L_{amdt}$ for $k \in [1,n]$. Dually, a regular over-approximation of the trace closure of a regular language $L$ could be chosen from the $n+1$ candidates $\|\|_{i=1}^n P_i(L) - P_k(\|\|_{i=1}^n P_i(L) - L'^c)$ for $k \in [1,n]$. If $L$ is prefix closed, then it is desirable for the component language to
be also prefix closed. For that purpose, we could modify $L_k$ to be either $P_k(L) - P_k(\bigcap_{i=1}^{n} P_i(L) - L)\Sigma^*_k$ or, for a better solution (in the sense of set inclusion), the supremal prefix closed sublanguage of $P_k(L) - P_k(\bigcap_{i=1}^{n} P_i(L) - L)$, with $L_i = P_i(L)$ for $i \in [1, n]$ and $i \neq k$. We note that the supremal prefix closed sublanguage of an arbitrary regular language is computable in linear time.

We say an approximation of the maximal trace closed sublanguage (the trace closure) of a regular language is exact if it is equal to the maximal trace closed sublanguage (the trace closure) of the given language. We show that the exactness of the approximations $L_{amt}$ or $L_{amd}$ is undecidable. This is implied by the following stronger result. Indeed, the exactness of any regular under-approximation of the maximal trace closed sublanguage (any regular over-approximation of the trace closure) of a regular language is decidable if and only if $I$ is transitive. We remark that the undecidability is not implied by the undecidability of the regularity of $[L]$. For example, the problem whether a regular language is trace closed, i.e., $[L] = L$, is decidable [20].

**Proposition 3:** Let $R$ be an operator on languages such that $R$ preserves regularity and a finite automaton that recognizes $R(L)$ is effectively constructible whenever $L$ is regular, and $[L] \subseteq R(L)$, then for an arbitrary regular language $L$ it is decidable whether $[L] = R(L)$ iff $I$ is transitive.

Intuitively, it means that not only we are not able, in general, to compute the trace closure of a regular language (synthesis problem), but we are also not able to check whether any regular approximation is exact in general (verification problem). However, it is possible to show that using only complementation and decomposition closure operators, the regular under-approximation $L_{amt}$ is optimal, i.e., the largest possible under-approximation of the maximal trace closed sublanguage $[L^c]^c$ of a language $L$.

**VI. CONCLUSION**

We have investigated two closely related problems, i.e., Decomposable Sublanguage Problem and Joint Observability Problem, and other similar problems. Two interesting open problems are to obtain characterizations of the decidability of Distributed Supervisory Control Problem in Section IV-A and Prefix Closed Joint Observability Problem in Section IV-C. The heuristics developed in this work could be used to compute approximately the trace closure of a regular language, which is useful for partial reduction based model checking and in the symbolic verification of safety property in some classes of parameterized systems [12] as well as dealing with many undecidable problems in control or trace theory.

**Acknowledgements** We thank Dr. Hendrik Jan Hoogeboom for an email exchange regarding a technical aspect. Second author was supported by Romanian research grant no. PN-II-ID-PCE-2011-3-0688 (project MuVeT).

**REFERENCES**


