

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 Σ is said to be *decomposable* with respect to a distribution, i.e., a collection of non-empty subalphabets, of Σ 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**. 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

L. Lin, R. Su, W. Wang and A. R. Shehabinia are with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. llin5@e.ntu.edu.sg, rsu@ntu.edu.sg, wlwang@ntu.edu.sg, ahmadrez1@e.ntu.edu.sg

Alin Stefanescu is with Univ. of Pitesti and Univ. of Bucharest, Romania. alin@stefanescu.eu

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. Finally, two simple applications of the developed heuristics are illustrated.

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 and provide further details in the appendix.

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 $\{1, 2, \dots, n\}$. For a given alphabet Σ , a *distribution* of Σ of size n is an n -tuple $\Delta = (\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ such that $\emptyset \neq \Sigma_i \subseteq \Sigma$ for $i \in [1, n]$. A distribution $\Delta = (\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ 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 Δ , we will assume it is proper. Given a distribution $\Delta = (\Sigma_1, \Sigma_2, \dots, \Sigma_n)$, we have n projections P_i from Σ^* to Σ_i^* and n inverse projections P_i^{-1} from Σ_i^* to Σ^* . We understand that both projections and inverse projections are naturally extended to map between languages. The *synchronous product* $\prod_{i=1}^n L_i$ of languages L_i over Σ_i is defined as $\bigcap_{i=1}^n P_i^{-1}(L_i)$. Given a language L over Σ , $\prod_{i=1}^n P_i(L)$ is said to be the *decomposition closure* of L with respect to Δ . We denote the complement of a language L by L^c . \bar{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, \dots, 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 $(a_i, b_i) \in I, v_{i-1} = u_i a_i b_i u'_i$ and $v_i = u_i b_i a_i u'_i$. 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 $\Delta = (\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ naturally induces an independence relation I 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 *dependence relation*. Then $I = \Sigma \times \Sigma - D$ is the independence relation induced by the distribution Δ . In the rest of the paper, whenever we are given a distribution Δ , it is implicitly assumed that the independence relation I 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 (Σ, I) , where $I \subseteq \Sigma \times \Sigma$ is viewed as the set of undirected edges, does not contain the two graphs $P_4 = \{\{a, b\}, \{b, c\}, \{c, d\}\}$ 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 [26], [7] will be used later.

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

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

III. CHARACTERIZATION RESULTS AND APPLICATION

A. Characterization of Decomposable Sublanguage Problem

Formally, a language L over Σ is said to be *decomposable* with respect to a distribution $\Delta = (\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ of Σ if $L = \prod_{i=1}^n P_i(L)$, 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 Σ is Σ^* . 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 $[L^c]^c$, for any language L over Σ .

Now it is straightforward to see that $\exists s \in L, [s] \subseteq L$ iff $[L^c]^c \neq \emptyset$ iff $[L^c] \neq \Sigma^*$. 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 Σ are said to be *jointly observable* with respect to a distribution $\Delta = (\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ of Σ 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.¹ 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 $\{a_1, a_2, \dots, a_n\}$ 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 Σ_i , and a specification language L over $\Sigma = \bigcup_{i=1}^n \Sigma_i$, decide whether there exist n non-blocking supervisors S_i over Σ_i (i.e., observing and controlling only a subset of Σ_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 $L_m(S_1/G_1), L_m(S_2/G_2), \dots, L_m(S_n/G_n)$ are synchronously non-conflicting.
- 3) $\emptyset \neq \parallel_{i=1}^n L_m(S_i/G_i) \subseteq L$.²

Corollary 1: Distributed Supervisory Control Problem is undecidable if I is not transitive.

There exist some similar results in the literature concerning the undecidability of decentralized and distributed control [6], [25], [24], [27]. For some comparisons between these undecidability results, see [6], [24]. 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 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 (see Fig. 1, 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 (see Fig. 1, 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 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

²Note that we explicitly require that $\parallel_{i=1}^n L_m(S_i/G_i) \neq \emptyset$, since this is the solution space that we are interested in; otherwise, the problem becomes trivial.

¹However, [6] only shows the undecidability.

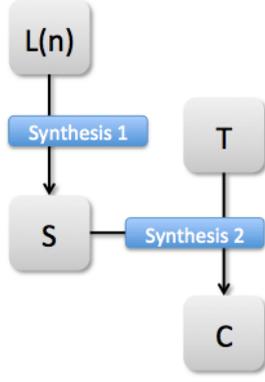


Fig. 1. Top-down control protocol synthesis overview.

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* [23], where Σ_g is the *global event set* and Σ_p is the *private event set*. A distribution $\Delta_T = (\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ of size n based on template alphabet Σ_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 Σ_T to events in Σ_i for each $i \in [1, n]$, such that $h_i(\sigma) = \sigma$, 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 σ_i . Depending on the context, h_i may also be extended in the natural way to either a bijective map from Σ_T^* to Σ_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 Δ_T if $\forall i, j \in [1, n], h_i^{-1}(P_i(L)) = h_j^{-1}(P_j(L))$. 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 Σ_T to events in Σ_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 $\forall i \in [1, n], \Sigma_i = A_i \cup B_i \cup \{c_i\}$. 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 $s_i^f(w) = c_i^{|f_i(h_i(w))|} h_i(w) f_i(h_i(w)) c_i^{|f_i(h_i(w))|} \in (A_i \cup B_i \cup \{c_i\})^*$ and $s_i^g(w) = c_i^{|g_i(h_i(w))|} h_i(w) g_i(h_i(w)) c_i^{|g_i(h_i(w))|} \in (A_i \cup B_i \cup \{c_i\})^*$, for each $i \in [1, n]$. Since $|f_i(h_i(w))| = |f(w)|$ and $|g_i(h_i(w))| = |g(w)|$, we have $s_i^f(w) = c_i^{|f(w)|} h_i(w) f_i(h_i(w)) c_i^{|f(w)|}$ and $s_i^g(w) =$

$c_i^{|g(w)|} h_i(w) g_i(h_i(w)) c_i^{|g(w)|}$. The independence relation $I(n)$ is $\{(x, y) \mid x \in \Sigma_i \text{ and } y \in \Sigma_j \text{ for different } i \text{ and } j\}$ and $W_f(n) = \{s_1^f(w) s_2^f(w) \dots s_n^f(w) \mid w \in A^+\}$ and $W_g(n) = \{s_1^g(w) s_2^g(w) \dots s_n^g(w) \mid w \in A^+\}$ 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 $L(n) = \Sigma(n)^* \setminus (L_f(n) \cup L_g(n))$ is symmetric with respect to Δ_T .

The following lemma essentially conveys the main proof idea.

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

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 $L(n)$ over $\bigcup_{i=1}^n \Sigma_i$ parameterized by n , it is undecidable whether there exists a non-empty specification template S over Σ_T such that the composition of n isomorphic copies $h_i(S)$ of S satisfies the specification $L(n)$, i.e., $\bigcap_{i=1}^n h_i(S) \subseteq L(n)$, for any $n \geq 1$. The undecidability holds even when $L(n)$ is symmetric for any $n \geq 2$, that is, $\forall n \geq 2, \forall i, j \in [1, n], h_i^{-1}(P_i(L(n))) = h_j^{-1}(P_j(L(n)))$ 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 Σ_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 Σ_T (i.e., observing and controlling only a subset of Σ_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 $L_m(S_1/G_1), L_m(S_2/G_2), \dots, L_m(S_n/G_n)$ are synchronously non-conflicting.

3) $\emptyset \neq \bigcap_{i=1}^n L_m(S_i/G_i) \subseteq L(n)$.

Here each S_i over Σ_i is an isomorphic copy of S and Σ_i is an isomorphic copy of Σ_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 $\{a_1, a_2, \dots, a_n\}$ and $\Sigma_3 = \{e, m\}$ that encodes $\{b_1, b_2, \dots, b_n\}$, 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 ϵ) prefix closed regular languages G', B' , i.e. $G' \cap B' = \{\epsilon\}$, whether $[G'] \cap [B'] = \{\epsilon\}$? Now let $G, B \subseteq \Sigma_1^*$ be two disjoint languages, Δ_1 be a distribution of Σ_1 and $\Delta_2 = (\Delta_1, \{e\}, \{m\})$ be a distribution of $\Sigma_2 = \Sigma_1 \cup \{e\} \cup \{m\}$,³ where e, m are two new symbols not in Σ_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: $[G]_{I_1} \cap [B]_{I_1} \neq \emptyset$ iff $[\overline{eGm}]_{I_2} \cap [\overline{mBe}]_{I_2} \neq \{\epsilon\}$.

Here, I_1 and I_2 are the induced independence relations of Σ_1 and Σ_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 $\Delta = (\{a, c\}, \{b, d\}, \{e\}, \{m\})$, $G' = e(ab + cd)^+m$ and $B' = m(u_1w_1 + \dots + u_nw_n)^+e$, where u_i is over $\{a, c\}$ and w_i is its isomorphic copy over $\{b, d\}$, together encode the PCP instance, i.e. $[G'] \cap [B'] \neq \{\epsilon\}$ iff $\exists i_1, i_2, \dots, i_k$ in $[1, n]$, $u_{i_1}u_{i_2} \dots u_{i_k}$ is isomorphic to $w_{i_1}w_{i_2} \dots w_{i_k}$. 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

³ Δ_2 augments the tuples of Δ_1 with $\{e\}, \{m\}$.

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 $(\bigcap_{i=1}^n P_i(L^c))^c$, denoted by L_{amt} , is a regular under-approximation for the maximal trace closed sublanguage of L .⁴ Such an under-approximation indeed provides a heuristics for Decomposable Sublanguage Problem. If $(\bigcap_{i=1}^n P_i(L^c))^c$, which is effectively computable, is non-empty, then L has a non-empty decomposable sublanguage with respect to Δ . Note that L_{amt} itself may not be decomposable with respect to Δ , since the class of decomposable languages is not closed under complementation.

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

Proposition 2: $L_{amd}^k \subseteq L$.

Thus, L_{amd}^k 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_{amd}^k is also a trace closed regular language. As approximations for the maximal trace closed sublanguage, $L_{amd}^k = L_k \parallel (\bigcap_{i \neq k, i \in [1, n]} L_i)$ and $L_{amt} = (\bigcap_{i=1}^n P_i(L^c))^c$ are incomparable. For example, let $L = \{ab, ba\}$ over $\Sigma = \{a, b\}$ and $\Delta = (\{a\}, \{b\})$, then $(\bigcap_{i=1}^n P_i(L^c))^c = \emptyset$ but $L_k \parallel (\bigcap_{i \neq k, i \in [1, 2]} L_i) = L$ for either $k = 1, 2$. Let $L = \Sigma^* - \{ab\}$, then $(\bigcap_{i=1}^n P_i(L^c))^c = \Sigma^* - \{ab, ba\}$ but $L_k \parallel (\bigcap_{i \neq k, i \in [1, 2]} L_i) = \Sigma^* - \Sigma^*a\Sigma^*$ or $\Sigma^* - \Sigma^*b\Sigma^*$ for $k = 1$ or 2 , respectively. In practice, an approximation for

⁴A similar idea has also appeared in [7], but in a slightly different setting.

the maximal trace closed sublanguage of a regular language could be chosen from the $n + 1$ candidates L_{amt}, L_{amd}^k 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 $\|_{i=1}^n P_i(L), [(P_k(L^c) - P_k(\|_{i=1}^n P_i(L^c) - L^c)) \|_{i \neq k, i \in [1, n]} P_i(L^c)]^c$ 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 $P_k(L) - P_k(\|_{i=1}^n P_i(L) - L) \Sigma_k^*$ or, for a better solution (in the sense of set inclusion), the supremal prefix closed sublanguage⁵ of $P_k(L) - P_k(\|_{i=1}^n P_i(L) - L)$, 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.

Remark: There exists another method for computing an upper-approximation of the trace closure of a regular language, namely regular widening technique [28]. Compared with our method, the advantage of regular widening technique is obviously its wider applicability. In dealing with a semi-commutation relation that is not symmetric, the approximation result is expected to be better than our heuristics in general. Our heuristics is more geared towards the computation of the trace closure of a regular language. It could, at the cost of precision, also deal with semi-commutation relations that are not symmetric, by replacing the original semi-commutation relation with its symmetric closure. The outcome of our heuristics is trace closed by design, and the heuristics is simpler than regular widening technique and attacks the problem from a rather interesting point of view, i.e., the duality between the trace closure and the maximal trace closed sublanguage of a regular language. In practice, the heuristics could be used together with regular widening technique for other types of transition relations that contain relations that are not semi-commutation relations.

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}^k 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). One thus might be interested in seeking some “metric”, if it exists, that could measure and bound the degree of exactness of any approximation scheme.

B. Applications

In this section, we provide two simple illustrations of the applications of our heuristics: one for Joint Observability Problem and the other for parameterized model checking.

1) *Application to Joint Observability Problem:* Given two disjoint languages G, B over Σ and a distribution Δ of Σ , we call the intersection of their trace closures $[G] \cap [B]$ the *ambiguous string set* of G and B with respect to Δ . Suppose G represents the “good” language and B the “bad” language and a string x is known to be from $G \cup B$. If $x \in [G] \setminus ([G] \cap [B])$, then x is surely the “good” string, i.e., $x \in G$. Since the problem whether $x \in [G]$ is decidable, a regular upper-approximation K of the ambiguous string set provides a confirmation on the membership of x in G if $x \in [G]$ and $x \notin K$. Similarly, this applies for the identification of “bad” strings. This information is very helpful in making decentralized control decision. If the string is identified to be “bad”, then at least one supervisor needs to prevent its occurrence.

2) *Application to Parameterized Model Checking:* The heuristics for the computation of trace closure of regular languages could be used for the model checking of safety property of certain mutual exclusion protocols, in particular when the transition relation of the parameterized system is a (semi) commutation rewriting system [12]. In this section, we illustrate a simple application of the heuristics for the verification of safety property of a mutual exclusion protocol. In the token passing of a linearly ordered processes [12], the initial configuration of the parameterized system is the regular language ab^* , where a represents the state that the process has a token, and b represents the state that the process does not have a token, and the transition relation is a semi-commutation rule $R : ab \rightarrow ba$ that represents passing a token to the right neighbour. The verification problem is then to check whether $R^*(L) \cap (a + b)^* a (a + b)^* a (a + b)^* = \emptyset$. Clearly, one could check whether $[L] \cap (a + b)^* a (a + b)^* a (a + b)^* = \emptyset$, since $R^*(L) \subseteq [L]$. In this case $L_{amt} = b^* ab^*$ computes exactly $[ab^*]$. Thus $[L] \cap (a + b)^* a (a + b)^* a (a + b)^* = \emptyset$, so the mutual exclusion property is satisfied. For more complex systems that contain (semi) commutation rewriting rules, the heuristics could be used as a subroutine for computing an upper approximation of the commutation closure.

VI. CONCLUSION

We have investigated two closely related problems, i.e., Decomposable Sublanguage Problem and Joint Observability Problem, and other related problems. Decomposable Sublanguage Problem corresponds to the decision problem $[K] \supseteq L$ when $L = \Sigma^*$, and Joint Observability Problem corresponds

⁵The supremal prefix closed sublanguage of an arbitrary language exists, since the class of prefix closed sublanguages is closed under union.

to the decision problem $[K] \subseteq L$. 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.

REFERENCES

- [1] J. Komenda, T. Masopust, J. H. van Schuppen. "On conditional decomposability", *Systems & Control Letters*, 61(12): 1260-1268, 2012.
- [2] F. Lin, W.M. Wonham. "Decentralized supervisory control of discrete-event systems", *Information Sciences*, 44 (3): 199-224, 1988.
- [3] Y. Willner and M. Heymann. "Supervisory control of concurrent discrete event systems", *Int. Journal of Control*, 54(5): 1143-1169, 1991.
- [4] K. Cai, W. M. Wonham. "Supervisor localization: a top-down approach to distributed control of discrete-event systems". *IEEE Trans. of Automatic Control*, 55(3): 605-618, 2010.
- [5] A. Stefanescu. "Automatic synthesis of distributed transition systems", Ph.D. dissertation. University of Stuttgart, 2006.
- [6] S. Tripakis. "Undecidable problems of decentralized observation and control on regular languages", *Information Processing Letters*, 90(1): 21-28, 2004.
- [7] Y. Chen, X. C. Ding, A. Stefanescu, C. Belta. "Formal approach to the deployment of distributed robotic teams", *IEEE Trans. on Robotics*, 28(1): 158-171, 2012.
- [8] L. Lin, R. Su, A. Stefanescu. "Remarks on the difficulty of top-down supervisor synthesis", In Proc. of 12th International Conference on Control, Automation, Robotics, and Vision (ICARCV'12), pp. 270-275, 2012.
- [9] A. Muscholl, H. Petersen. "A note on the commutative closure of star-free languages", *Information Processing Letters*, 57(2): 71-74, 1996.
- [10] R. Su, L. Lin. "Synthesis of control protocols for multi-agent systems with similar actions", manuscript.
- [11] J. Sakarovitch. "The "last" decision problem for rational trace languages". *LATIN'92*, Lecture Notes in Computer Science, Vol. 583, pp. 460-473, 1992.
- [12] A. Bouajjani, A. Muscholl, and T. Touili. "Permutation rewriting and algorithmic verification", *Information and Computation*, 205(2): 199-224, 2007.
- [13] K. R. Apt, D. C. Kozen. "Limits for automatic verification of finite-state concurrent systems", *Information Processing Letters*, 22(6): 307-309, 1986.
- [14] I.J. Aalbersberg, H.J. Hoogboom. "Characterizations of the Decidability of Some Problems for Regular Trace Languages". *Mathematical Systems Theory*, 22(1): 1-19, 1989.
- [15] S. Tripakis. "Undecidable problems of decentralized observation and control", In Proc. of the 40th IEEE Conference on Decision and Control, vol 5, pp. 4104-4109, 2001.
- [16] R. Senguptal, S. Tripakis. "Decentralized diagnosability of regular languages is undecidable", In Proc. of the 41th IEEE Conference on Decision and Control, vol 1, pp. 423-428, 2002.
- [17] J. E. Hopcroft, J. D. Ullman. *Introduction to automata theory, languages, and computation*, Addison-Wesley, Reading, Massachusetts, 1979.
- [18] W. M. Wonham. *Supervisory Control of Discrete-Event Systems*, Systems Control Group, Dept. of ECE, University of Toronto, Toronto, Canada, 2010.
- [19] I.J. Aalbersberg, E. Welzl. "Trace languages defined by regular string languages", *ITA*, 20(2): 103-119, 1986.
- [20] D. Peled, T. Wilke, P. Wolper. "An algorithmic approach for checking closure properties of temporal logic specifications and omega-regular languages", *Theor. Comput. Sci.*, 195(2): 183-203, 1998.
- [21] D. Peled. "A generalized closure and complement phenomenon", *Discrete Math.*, vol. 50, pp. 285-293, 1984.
- [22] K. Nazari, J. G. Thistle. "Blocking in fully connected networks of arbitrary size", *IEEE Trans. on Automatic Control*, 57(5): 1233-1242, 2012.
- [23] K. Rohloff and S. Lafortune. "The verification and control of interacting similar discrete event systems", *SIAM Journal on Control and Optimization*, 45(2): 634-667, 2006.
- [24] A. Arnold, A. Vincent and I. Walukiewicz. "Games for synthesis of controllers with partial observation", *Theor. Comput. Sci.*, 303 (1): 7-34, 2003.
- [25] J.G. Thistle. "Undecidability in decentralized supervision", *Systems & Control Letters*, 54(5): 503-509, 2005.
- [26] S. Tripakis. "Two-phase distributed observation problems". In 5th Intl. Conf. on Application of Concurrency to System Design (ACSD'05), pp. 98-105, 2005.
- [27] P. Madhusudan and P.S. Thiagarajan. "A decidable class of asynchronous distributed controllers" In Proc. of the 13th International Conference on Concurrency Theory (CONCUR'02), pp. 145-160, 2002.
- [28] T. Touili. "Regular Model Checking using Widening Techniques", *Electronic Notes in Theoretical Computer Science*, 50(4): 342-356, 2001.
- [29] K. Rudie and W. M. Wonham. "Think globally, act locally: Decentralized supervisory control", *IEEE Trans. Automatic Control*, 37(11): 1692-1708, 1992.

Acknowledgements: We would like to acknowledge the email correspondence with Dr. Hendrik Jan Hoogboom. Second author was supported by Romanian research grant no. PN-II-ID-PCE-2011-3-0688 (project MuVeT).

APPENDIX

A. Technical Remarks

Remark for Theorem 1: A view of this result, directly from the perspective of Decomposable Sublanguage Problem, is as follows. On the one hand, if I is not transitive, then there exist three letters $a, b, c \in \Sigma$ such that $(a, c), (c, b) \in I$ but $(a, b) \notin I$. It is known that every instance of Post's Correspondence Problem (PCP) can be encoded as an instance of Decomposable Sublanguage Problem with $L \subseteq (a + b + c)^*$ and $\Delta = (\{a, b\}, \{c\})$ [5], [7], [8]. By restricting $L \subseteq (a+b+c)^*$ where $(a, c), (c, b) \in I$ but $(a, b) \notin I$, the problem of existence of a trace closed sublanguage, when I is not transitive, reduces to the case when $\Delta = (\{a, b\}, \{c\})$. Thus if I is not transitive, Decomposable Sublanguage Problem is undecidable. On the other hand, if I is transitive, then for every regular language L there exists a regular language K such that $[L]^c = [K]$ and the finite state automaton that recognizes K is effectively constructible [19]. In this case, the maximal trace closed sublanguage $[L^c]^c$ of L is equal to the trace closure of some regular language M , i.e., $[L^c]^c = [M]$. It is then obvious that the answer for Decomposable Sublanguage Problem is no iff M is empty. Thus if I is transitive, then Decomposable Sublanguage Problem is decidable.

Remark for testing decomposability: With a minor adaptation, the algorithm in [1] for checking conditional decomposability provides a polynomial time algorithm of the same complexity $\mathcal{O}(|\Sigma|m^{n+1})$ for checking decomposability for a given distribution of fixed size n , where Σ is the cardinality of Σ and m is the state size of the DFA G such that $L_m(G) = L$. The essence of the variant algorithm is to use the synchronous product of *non-deterministic finite automata*

(NFA) for language emptiness check and thus avoids the subset construction, i.e. determinization procedure, which may cause exponential state blow-up. More specifically, to check whether $L = \|\|_{i=1}^n P_i(L)$, the algorithm checks whether the NFA $(\|\|_{i=1}^n P_i(G)) \cap G^c$ has a reachable marked state, where G is a DFA such that $L_m(G) = L$. Note that this variant avoids the use of a technical lemma, i.e., Lemma 4 as in [1], on which the algorithm in [1] relies and works slightly more generally. This simple procedure was independently observed in [3]. It is also stated in [3] that if $(\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ forms a partition of Σ , then the complexity⁶ of checking decomposability is $\mathcal{O}(nm^3)$. We now present a very short proof for this. Indeed one could easily check that if $(\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ forms a partition of Σ , then $L = \|\|_{i=1}^n P_i(L)$ iff $\forall k \in [1, n], L = P_{[1, n] \setminus k}(L) \| P_k(L)$, where $P_{[1, n] \setminus k}$ maps from 2^{Σ^*} to $2^{(\cup_{i \in [1, n], i \neq k} \Sigma_i)^*}$. The complexity is now easily seen to be $\mathcal{O}(|\Sigma|nm^3)$.

Remark for heuristic L_{amt} : 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 . Indeed, $(\|\|_{i=1}^n P_i(L^c))^c$ is the interior of L with respect to the semi-topology induced by the decomposition closure operator and there are at most 14 distinct sets formed by the combinations of these two operators [21], thus one could check this fact by exhaustive comparison.

B. Proofs

Proof of Corollary 1 from Section III-A:

Proof: We set $L_m(G_i) = L(G_i) = \Sigma_i^*$ and let each local supervisor S_i control and observe all of Σ_i , i.e., $\Sigma_{i,c} = \Sigma_i$ and $\Sigma_{i,o} = \Sigma_i$. For any language L over Σ and distribution $\Delta = (\Sigma_1, \Sigma_2, \dots, \Sigma_n)$ of Σ , we show that L has a non-empty decomposable sublanguage with respect to Δ iff there exists a solution for the instance of Distributed Supervisory Control Problem with specification L , distribution Δ and the above constructed local plants G_i . We then conclude that Distributed Supervisory Control Problem is undecidable if I is not transitive. Clearly, if the instance of Distributed Supervisory Control Problem has a solution, then the answer for the corresponding instance of Decomposable Sublanguage Problem is yes. If the answer for the instance of Decomposable Sublanguage Problem is yes, then there exist $P_1(s), P_2(s), \dots, P_n(s)$, respectively over $\Sigma_1, \Sigma_2, \dots, \Sigma_n$, such that $\|\|_{i=1}^n P_i(s) \subseteq L$. Clearly, $P_i(s)$ is a controllable sublanguage of $L(G_i)$, since every language over Σ_i is a controllable sublanguage of $L(G_i)$ with the above setup. Thus there exists a supervisor S_i , such that $L_m(S_i/G_i) = P_i(s)$. Since $L_m(G_i) = \Sigma_i^*$, any such supervisor is nonblocking. It is not difficult to see that $P_1(s), P_2(s), \dots, P_n(s)$ are synchronously non-conflicting. Thus the three conditions are all satisfied and there exists a solution to the instance of Distributed Supervisory Control

Problem with specification L , distribution Δ and the above constructed plants G_i . \square

Proof of Lemma 4 from Section IV-B:

Proof: The proof idea follows [9] with suitable modification. The technical detail is omitted due to the tedious construction and limited space. \square

Proof of Lemma 5 from Section IV-B:

Proof: The relative positions of the substrings $s_i^f(w)$ in $s(n) := s_1^f(w)s_2^f(w)\dots s_n^f(w)$ do not make a difference in the $[W_f(n)]$ and $[W_g(n)]$ due to the independence relation $I(n)$. Any procedure to construct $L_f(n)$ could be extended to construct a symmetric $L_f(n)$ by taking the union of $n!$ copies of $L_f(n)$ with indexes permuted. Similarly, $L_g(n)$ can be constructed to symmetric and thus $L(n) = \Sigma(n)^* \setminus (L_f(n) \cup L_g(n))$ is then symmetric with respect to Δ_T . \square

Proof of Lemma 1 from Section IV-C:

Proof: If $[G] \cap [B] \neq \emptyset$, say $s \in G, s' \in B$ and $[s] = [s']$, then $esm \in eGm, ms'e \in mBe$ and $[esm] = [ms'e]$. Thus $[eGm] \cap [mBe] \neq \{\epsilon\}$. If $[eGm] \cap [mBe] \neq \{\epsilon\}$, then $\exists s \in G, s' \in B$ such that $[esm] \cap [ms'e] \neq \{\epsilon\}$. This is only possible iff $[esm] = [ms'e]$ iff $[s] = [s']$. Then $[G] \cap [B] \neq \emptyset$. \square

Proof of Proposition 2 from Section V:

Proof: $L_k \| (\|\|_{i \neq k, i \in [1, n]} L_i)$ is equal to $P_k^{-1}(P_k(L) \cap P_k(\|\|_{i=1}^n P_i(L) \cap L^c)^c) \cap \bigcap_{i \neq k, i \in [1, n]} P_i^{-1}(P_i(L))$ by definition. Now $P_k^{-1}(P_k(L) \cap P_k(\|\|_{i=1}^n P_i(L) \cap L^c)^c) = P_k^{-1}(P_k(L)) \cap P_k^{-1}(P_k(\|\|_{i=1}^n P_i(L) \cap L^c)^c) = P_k^{-1}(P_k(L)) \cap P_k^{-1}(P_k(\|\|_{i=1}^n P_i(L) \cap L^c))^c$. It follows that $L_{amd}^k = P_k^{-1}(P_k(\|\|_{i=1}^n P_i(L) \cap L^c))^c \cap \|\|_{i=1}^n P_i(L) \subseteq (\|\|_{i=1}^n P_i(L) \cap L^c)^c \cap \|\|_{i=1}^n P_i(L) = L$. \square

Proof of Proposition 3 from Section V:

Proof: The proof idea is inspired by [9]. On the one hand, if I is transitive, it is decidable whether the trace closure of a regular language is regular [11]. Moreover, a finite automaton that recognizes $[L]$ is effectively constructible whenever $[L]$ is regular [9]. Thus if I is transitive, it is decidable whether $[L] = R(L)$.

On the other hand, let A, B be two disjoint alphabets and f, g be two homomorphisms from A^* to B^* that encode a given instance of the PCP, such that $\exists w \in A^+, f(w) = g(w)$ iff the answer to that instance is yes. Let $W_f = \{wf(w)c^{|f(w)|} \mid w \in A^+\}, W_g = \{wf(w)c^{|f(w)|} \mid w \in A^+\}$ with $\Delta = (A \cup B, \{c\})$ and c a new symbol. Then $\exists w \in A^+, f(w) = g(w)$ iff $W_f \cap W_g \neq \emptyset$. It is possible to show that there exist two regular languages L_f, L_g such that $[L_f] = [W_f]^c$ and $[L_g] = [W_g]^c$ [9]. Let $L = L_f \cup L_g$. If $W_f \cap W_g = \emptyset$, then $[L] = [L_1 \cup L_2] = ([W_f] \cap [W_g])^c = [W_f \cap W_g]^c = \Sigma^*$. Since $\Sigma^* = [L] \subseteq R(L)$, $[L] = R(L)$. If $W_f \cap W_g \neq \emptyset$, then $[L]$ is not regular [9]. Thus $[L] \neq R(L)$ since $R(L)$ is regular. Any decision procedure for verifying $[L] = R(L)$ then provides a decision procedure for the PCP problem, thus the verification problem $[L] = R(L)$ is undecidable. Note that it is possible to encode the set $A \cup B$ using $\{a, b\}$, thus the undecidability result holds with the distribution $\Delta = (\{a, b\}, \{c\})$ and regular language $L \subseteq (a+b+c)^*$. Thus the undecidability result indeed holds when I is not transitive. \square

⁶The analysis of [3] ignores the factor $|\Sigma|$.