1. AN EXPANDED VERSION OF SECTION III IN THE PUBLISHED MANUSCRIPT

Well-posedness of the considered SC problem: We begin our analysis of the considered SC problem by providing the necessary and sufficient condition for the existence of a correct supervisor. For the systematic characterization of this feasibility condition, we need the following definition:

**Definition 1.** Consider an s-DES \( \mathcal{G} \equiv (X \times S, \Sigma \cup E, \delta, (x_0, s_0)) \), and the supervisor \( \Gamma^0 \) defined by:

\[
\forall (x, s) \in X \times S, \, \Gamma^0(x, s) = \emptyset
\]

Then, for any global state \((x, s) \in X \times S\), the “ uncontrollable reach” \( \text{Reach}^u(x, s; \mathcal{G}) \) of this state is defined by

\[
\text{Reach}^u(x, s; \mathcal{G}) \equiv \text{Reach}((X \times S, \Sigma \cup E, \Gamma^0/\delta, (x, s)))
\]

Also, we define the uncontrollable reach for the execution of a controllable event \( \sigma \in \Sigma^c \) at some state \((x, s) \in X \times S\) as follows:

\[
\text{Reach}^u(x, s, \sigma; \mathcal{G}) = \bigcup_{(x', s') \in \delta(x, s, \sigma)} \text{Reach}^u(x', s'; \mathcal{G})
\]

In view of Definition 1, the complete feasibility condition for the considered SC problem can be expressed as follows:

**Proposition 1.** There exists a correct supervisor for the considered SC problem if and only if (iff) \( \forall (x, s) \in \text{Reach}^u(x_0, s_0; \mathcal{G}) \), \( s \in S_2 \) (or, equivalently, \((x, s) \in \Omega\)).

**Proof:** The necessity of the above condition is obvious. The sufficiency is established by the fact that, under satisfaction of this condition, the supervisor \( \Gamma^0 \) of Definition 1 is a correct supervisor. \( \square \)

In the following, unless otherwise specified, we shall assume that all the addressed instantiations of the considered SC problem are feasible.

Disjunction of correct supervisors and pertinent correct supervisors: Next, we elaborate further on the sought supervisors and their properties. We are especially interesting in a notion of “supervisor disjunction” for the considered SC problem that preserves the correctness of the constituent supervisors, since the availability of such an operation will enable the definition of a notion of “maximally permissive (correct) supervision”, and therefore, a notion of “optimal control” for the considered problem. We begin our analysis with the following proposition.

**Proposition 2.** Let \( \Gamma \) be a correct supervisor for the considered problem, and \( \Gamma' \) be a supervisor such that

\[
\forall (x, s) \in X \times S, \, \Gamma'(x, s) \subseteq \Gamma(x, s)
\]

Then, the more restrictive supervisor \( \Gamma' \) is also a correct supervisor.

**Proof:** It is easy to check that \( \text{Reach}(\Gamma'/\mathcal{G}) \subseteq \text{Reach}(\Gamma/\mathcal{G}) \subseteq \Omega \), where the second inclusion results from the presumed correctness of supervisor \( \Gamma \). Hence, the controlled DES \( \Gamma'/\mathcal{G} \) satisfies the inclusion of Eq. 4 in the main text, and supervisor \( \Gamma' \) is a correct supervisor. \( \square \)

Next, we consider two correct supervisors \( \Gamma_1 \) and \( \Gamma_2 \), and define the “disjunctive” supervisor \( \Gamma_1 \lor \Gamma_2 \) as follows:

**Definition 2.** Given two correct supervisors \( \Gamma_1 \) and \( \Gamma_2 \), the “disjunctive” supervisor \( \Gamma_1 \lor \Gamma_2 \) is obtained by setting

\[
\forall (x, s) \in X \times S, \, (\Gamma_1 \lor \Gamma_2)(x, s) \equiv \Gamma_1(x, s) \lor \Gamma_2(x, s)
\]

It is clear from Definition 2 that \( \text{Reach}(\Gamma_1 \lor \Gamma_2) \supseteq \text{Reach}(\Gamma_i/\mathcal{G}) \), for \( i = 1, 2 \), i.e., the disjunctive supervisor enables a richer behavior than its constituent supervisors. But this new supervisor may not be correct; this is demonstrated by the following example.

**Example 1:** Consider the DES \( \mathcal{G} \) with a single operational mode \( x \) and the corresponding state transition diagram (STD) depicted in Fig. 1. The control predicate is specified by the state set \( S_2 = \{s_0, s_1, s_2, s_3, s_5\} \). Also, in the depicted STD, events \( a \) and \( b \) are controllable, while event \( u \) is uncontrollable. Table 1 provides two correct supervisors \( \Gamma_1 \) and \( \Gamma_2 \) for this DES, the corresponding reachability sets for the controlled DES \( \Gamma_1/\mathcal{G} \) and \( \Gamma_2/\mathcal{G} \), and also the disjunctive supervisor \( \Gamma_1 \lor \Gamma_2 \) and the reachability set of the controlled DES \( \Gamma_1 \lor \Gamma_2/\mathcal{G} \). As it can be seen from the provided data, \( \Gamma_1 \lor \Gamma_2 \) fails to confine the behavior of the underlying DES \( \mathcal{G} \) within the state set \( S_2 \), and therefore, it is not a correct supervisor. \( \square \)

The incorrectness of the disjunctive supervisor \( \Gamma_1 \lor \Gamma_2 \) in the previous example is due to the facts that (i) supervisor \( \Gamma_2 \) enables the transition from state \( s_1 \) to state \( s_2 \), and (ii) the forbidden state \( s_4 \) is uncontrollably reachable from...
Reach a,b

The next proposition establishes that it is reasonable to motivate the following definition:

**Theorem 1**. Consider an s-DES $G$ with required control invariants $I_x$, expressed by the state sets $S_x$, $x \in X$, and two correct pertinent supervisors $\Gamma_1$, $\Gamma_2$. Then, the disjunctive supervisor $\Gamma_1 \lor \Gamma_2$ is a correct pertinent supervisor for the considered problem.

Proof: The pertinence of the supervisor $\Gamma_1 \lor \Gamma_2$ results immediately from the definitions of the disjunctive and the pertinent supervisors, and the pertinence of the constituent supervisors $\Gamma_1$ and $\Gamma_2$. Next, we prove the correctness of this supervisor.

For any state $(x, s) \in \text{Reach}(\Gamma_1 \lor \Gamma_2 \mid G)$, let $\#(x, s)$ denote the minimum number of transitions that are necessary for reaching state $(x, s)$ from the initial state $(x_0, s_0)$ in the dynamics of $\Gamma_1 \lor \Gamma_2 \mid G$. We shall prove the sought result by an induction on $\#(x, s)$.

The base case of $\#(x, s) = 0$ results immediately from the presumed feasibility of the considered supervisory control problem and Proposition 1.

Next, suppose that the correctness condition holds true for every state $(x, s) \in \text{Reach}(\Gamma_1 \lor \Gamma_2 \mid G)$ with $\#(x, s) \leq n$, i.e., for every such state, $s \in S_x$. We shall show that this condition also holds true for every state $(x, s)$ with $\#(x, s) = n + 1$ and suppose that $s \notin S_x$. Then, the presumed feasibility of the considered problem implies that $(x, s) \notin \text{Reach}^n(x_0, s_0 \mid G)$. Hence, every transition sequence leading from $(x_0, s_0)$ to $(x, s)$ must contain at least one controllable event. Consider such a transition sequence and let the triplet $(x', s', \sigma) \in X \times S \times \Sigma^*$ denote the last controllable transition on this path. Also, notice that, by the definition of this sequence, $\#(x', s') \leq n$, and thus, by the inductive hypothesis, $s' \notin S_x$. Finally, since the transition $(x', s', \sigma)$ is enabled by $\Gamma_1 \lor \Gamma_2$, it must be enabled by at least one of the two supervisors $\Gamma_1$ and $\Gamma_2$. On the other hand, the aforementioned assumptions also imply that $(x, s) \in \text{Reach}^n(x', s', \sigma \mid G)$, a fact that when combined with the working hypothesis that $s \notin S_x$, leads to the contradictory conclusion that at least one of the original supervisors $\Gamma_i$, $i = 1, 2$, is not pertinent. □

**Optimal supervision for the considered SC problem**: The closure of the correct pertinent supervisors under disjunction, that was established by Proposition 4, further implies that, for any feasible instantiation of the considered SC problem, the maximally permissive correct pertinent supervisor $^1$ is defined uniquely on the critical states $(x, s) \in X \times S$ with $\text{Reach}^n(x, s \mid G) \subseteq \Omega$ (i.e., the global states that can observe the imposed invariants). In the sequel, we shall denote the maximally permissive supervisor w.r.t. these critical states by $\Gamma^*$, and we shall also refer to it as the “optimal” supervisor. For the remaining states $(x, s) \in X \times S$

---

$^1$ We remind the reader that “maximal permissiveness” is defined in terms of the inclusions of Eq. ??.
that can violate uncontrollably the imposed invariants, the sought optimal supervisor can be left undefined, or, for better specificity, one can set $\Gamma^*(x,s) = \emptyset$, and this is the practice that we shall adopt herein.

A basic iterative algorithm for the computation of $\Gamma^*$: The computation of the optimal correct pertinent supervisor $\Gamma^*$ for any feasible instantiation of the considered SC problem can be performed through the application of a fixed-point iteration that is reminiscent of similar results provided by the classical Ramadge & Wonham SC theory Ramadge and Wonham (1989). More specifically, we define the operator $F$ upon the subsets of $X \times S$, that when applied on any given subset $G$ of this set, returns

$$F(G) = \{(x,s) \in G : \forall q \in \Sigma^u \cup E, \ \delta(x,s,q) \subseteq G\} \quad (7)$$

Also, we use the notation $F^n$ to denote the $n$-fold composition of this operator with itself. Then, we have the following theorem:

**Theorem 1.** Consider an s-DES $\mathcal{G}$ with required control invariants $I_x$ that are expressed by the state sets $S_x$, $x \in X$, and further assume that $V(x,s) \in Reach^u(x_0,s_0;\mathcal{G})$ $(x,s) \in \Omega$. Then, the limit $\lim_i F^n(\Omega)$ is obtained in a finite number of iterations and it is a non-empty subset of the set $\Omega$ containing the initial global state $(x_0,s_0)$. Furthermore, the sought supervisor $\Gamma^*$ can be defined as follows:

$$\forall (x,s) \in X \times S, \ \Gamma^*(x,s) = \begin{cases} \{x' \in \Sigma^u : \delta(x,s,x') \subseteq \lim_i F^n(\Omega)\}, & (x,s) \in \lim_i F^n(\Omega) \\ \emptyset, \text{ otherwise.} \end{cases}$$

**Proof:** The correctness of the supervisor $\Gamma^*$ defined in Theorem 1 can be obtained from its pertinence, once the latter is established, through an argument similar to that followed in the proof of Proposition 4. Hence, to establish the results claimed by Theorem 1, it suffices to establish that (i) the iteration providing $\lim_i F^n(\Omega)$ will terminate in a finite number of steps, (ii) the resultant set $\lim_i F^n(\Omega)$ will contain the initial state $(x_0,s_0)$, (iii) the supervisor $\Gamma^*$ which is subsequently constructed from this set through the equation in Theorem 1 is pertinent, and (iv) the restriction of this supervisor over the global states $(x,s) \in X \times S$ with $Reach^u(x,s;\mathcal{G}) \subseteq \Omega$ is maximally permissive.

These four results can be obtained immediately from Eq 7 and the equation in Theorem 1, the finiteness of the underlying (global) state space of DES $\mathcal{G}$, and the following lemma:

**Lemma 1.** For every $i = 1, 2, \ldots$, a state $(x,s)$ is removed from the set $F^{(i+1)}(\Omega)$ during the computation of $F^i(\Omega)$ iff $Reach^u(x,s;\mathcal{G}) \not\subseteq \Omega$ and any minimal uncontrollable transition sequence that leads from $(x,s)$ to some state $(x',s') \in X \times S \setminus \Omega$ has a length of $i$ steps.

**Proof:** The reader can easily verify that Lemma 1 holds true for $i = 1$. Next, suppose that Lemma 1 holds true for $i \leq n$. We shall show that it must also hold true for $i = n + 1$. Indeed, consider a state $(x,s)$ that is removed from $F^n(\Omega)$ during the computation of $F^{(n+1)}(\Omega)$. According to the logic of Eq. 7, this state is removed from $F^n(\Omega)$ because there is an event $q \in \Sigma^u \cup E$ and a state $(x',s') \in X \times S \setminus \Omega$ such that $(x',s') \in Reach^u(x,s;\mathcal{G})$. Applying the algorithm of Theorem 1, the finiteness of the supervisor $\Gamma^*$ is obtained in a finite number of steps. Hence, state $(x,s)$ has an emanating uncontrollable transition sequence that leads to $X \times S \setminus \Omega$ in no more than $n + 1$ steps. Indeed, consider a feasible instantiation of the considered SC problem in line with the developments presented in this section.

**Example 1 (cont.):** We apply the algorithm of Theorem 1 to compute the maximally permissive, correct, pertinent supervisor $\Gamma^*$ for the example DES of Fig. 1. The computation starts by setting $G_0 = \{s_0, s_1, s_2, s_3, s_5\}$. Applying the operator $F$ on $G_0$, we obtain $G_1 = F(G_0) = \{s_0, s_1, s_3, s_5\}$. State $s_2$ has been removed from $G_1$ since the forbidden state $s_4$ is in the uncontrollable reach of $s_2$. On the other hand, every state in the set $G_1$ enables only controllable events, and therefore, $G_2 = F(G_1) = G_1$. Furthermore, the reader can verify that the application of the equation in Theorem 1 in this case returns the supervisor $\Gamma_1$ of Table 1 as the corresponding maximal supervisor $\Gamma^*$.

2. A DEMONSTRATIVE EXAMPLE

In the considered example, two robots, $R_1$ and $R_2$, move upon the graph $G = (V, E)$ that is depicted in Figure 2. At any time instant, robot $R_i$, $i = 1, 2$, is on some graph vertex $v(R_i)$. Hence, the motion state is defined by $s \equiv (v(R_1), v(R_2))$. The current state changes by each agent deciding to move to a neighboring vertex through an interconnecting edge, or to stay at its current vertex. However, the moves of $R_1$ are controllable while the moves of $R_2$ are uncontrollable, and robot $R_1$ does not know the intended motion of robot $R_2$ when it determines its own move. Hence, the combined move for the two robots can be decomposed to a two-stage move, with robot $R_1$ making first its decision, and robot $R_2$ moving second. Finally, edge $e_5$ in the depicted graph can change its connectivity with respect to one of its two vertices in an uncontrollable manner, as annotated in the figure. Also, for more specificity, we assume that such a switching can take place only between two of the aforementioned motion cycles that determine the system state, and not during the cycle. The current positioning of the two robots on the graph vertices is as indicated in Fig. 2, and we want to minimally restrict the moves of robot $R_1$ at each state $s$ in order to ensure that the robots never get in a state with $v(R_1) = v(R_2)$.

The above problem can be modeled as an invariant-based SC problem in line with the developments presented in
Ramaadge and Wonham (1987). In this basic formulation, the system state is determined by (i) the positioning of the two robots in the graph, (ii) the robot to move next, and (iii) the current placement of the edge $e_5$. Furthermore, in states where robot $R_1$ moves, the feasible events are those that take the robot to a neighboring node or leave it at the same node, and they are controllable; i.e., some of these moves can be disabled at will. In states where robot $R_2$ moves, the corresponding events are defined as in the case of robot $R_1$, but they are uncontrollable; i.e., this robot can determine freely its motion. Also, in states that result from the motion of robot $R_2$, another uncontrollable event is the switching of the edge $e_5$ from its current position to the alternative one. Finally, as already stated, the invariant to be enforced by the sought supervisor is given by the predicate $v(R_1) \neq v(R_2)$.

From the above description of the underlying DES state and the dynamics involved, it is easily checked that the corresponding state space will consist of $7 \times 7 \times 2 = 196$ states. Next, we shall employ the s-DES model for a more efficient representation of the underlying dynamics, and a simple and expedient computation of the sought supervisor.

Hence, in the spirit of the s-DES representations that were discussed in the previous sections of this paper, we associate each graph configuration that results by the two were discussed in the previous sections of this paper, we associate each graph configuration that results by the two robots in the graph, (ii) the robot to move next, and (iii) the current placement of the edge $e_5$. Furthermore, in states where robot $R_1$ moves, the feasible events are those that take the robot to a neighboring node or leave it at the same node, and they are controllable; i.e., some of these moves can be disabled at will. In states where robot $R_2$ moves, the corresponding events are defined as in the case of robot $R_1$, but they are uncontrollable; i.e., this robot can determine freely its motion. Also, in states that result from the motion of robot $R_2$, another uncontrollable event is the switching of the edge $e_5$ from its current position to the alternative one. Finally, as already stated, the invariant to be enforced by the sought supervisor is given by the predicate $v(R_1) \neq v(R_2)$.

From the above description of the underlying DES state and the dynamics involved, it is easily checked that the corresponding state space will consist of $7 \times 7 \times 2 = 196$ states. Next, we shall employ the s-DES model for a more efficient representation of the underlying dynamics, and a simple and expedient computation of the sought supervisor.

Hence, in the spirit of the s-DES representations that were discussed in the previous sections of this paper, we associate each graph configuration that results by the two possible positionings of edge $e_5$, and the corresponding transitional dynamics, with a particular operational mode $x \in \{1, 2\}$. Furthermore, in the case of the considered system, a more efficient representation of the local dynamics materialized in each mode can be based on a “tabular” representation of the local states $s$ that correspond to all those states where robot $R_1$ is about to move. Focusing upon these states is most pertinent since these are the states that are amenable to control. At the same time, the impact of the uncontrollability that results from the free moves of robot $R_2$, after a move of robot $R_1$, can be assessed through a local computation at each of these states. On the other hand, as we shall demonstrate below, the impact of the uncontrollable switching of edge $e_5$ will be captured through the partial computations of the threads that compute the sought supervisors for each of these two modes, and the information (i.e., the messages) exchanged by these threads.

Table 2 introduces the tableau that is employed in the considered computations, and also represents the maximally permissive supervisor for Mode 1 that is computed by the corresponding thread of the algorithm of Fig. 1 in the manuscript while ignoring the uncontrollable transitions from Mode 1 to Mode 2. As already mentioned, each cell in this tableau corresponds to a local state $s$ in Mode 1 that anticipates the move of robot $R_1$, and the content of these cells report the feasible moves of robot $R_1$ that are also admissible at that state. Hence, for instance, the ‘3’ in cell (1,2) reflects the fact that at the considered state, the only move for robot $R_1$ that will avoid with certainty the undesired event of having robots $R_1$ and $R_2$ co-located at the end of the current cycle, is to force robot $R_1$ to node 3. Similarly, the content of the cell (1,5) indicates that, in the considered configuration of graph $G$, and with robot $R_2$ currently located at node 5, robot $R_1$ can essentially decide freely its next move at the current state. On the other hand, the dashes ‘-’ that appear in the principal diagonal of the considered tableau express the fact that these states violate the imposed invariant $I_1$ and therefore they are forbidden.

In the computation that was described in the previous paragraph, additional forbidden states can be generated in the case that the uncontrollability of the moves of robot $R_2$, in combination with the topology of the graph configuration that corresponds to the considered mode, do not allow any feasible and admissible moves for robot $R_1$ at these states. In the particular computation of Table 2 no such new forbidden states arise, since every non-diagonal state has a non-empty set of feasible and admissible moves. But we shall encounter this possibility in the corresponding computation for Mode 2, that is depicted in Table 3. Finally, we also notice that since the computation of Table 2 has generated no new forbidden states (other than the diagonal states), the computational process for Mode 1 goes into its idling mode without sending any message to its counterpart thread for Mode 2.

The initial computation for Mode 2 that corresponds to the computation of Table 2, is depicted in Table 3(a). It

<table>
<thead>
<tr>
<th>$s$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>1</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>4</td>
<td>4.6,7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
</tr>
<tr>
<td>5</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
</tr>
<tr>
<td>6</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
</tr>
<tr>
<td>7</td>
<td>7.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
</tr>
</tbody>
</table>

Table 2. The initial computation of the maximally permissive supervisor for Mode 1; this computation ignores the uncontrollable transitions between the modes.

<table>
<thead>
<tr>
<th>$s$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>4</td>
<td>4.6,7</td>
<td>3.6,7</td>
<td>2.6,7</td>
<td>-</td>
<td>2.3,4.6</td>
<td>2.3</td>
<td>2.3</td>
</tr>
<tr>
<td>5</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
<td>5.7</td>
</tr>
<tr>
<td>6</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
</tr>
<tr>
<td>7</td>
<td>7.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
<td>5.6,7</td>
</tr>
</tbody>
</table>

Table 3. The initial computation of the maximally permissive supervisor for Mode 2; this computation ignores the uncontrollable transitions between the modes.
can be checked that this new table contains non-diagonal empty states. Hence, for instance, we can see that for the state \((5,7)\) that is highlighted in boldface in Table 3(b), and with the edge \(e_5\) interconnecting nodes 4 and 7, it is impossible for robot \(R_1\) to determine a move at this state that will guarantee the corresponding invariant \(\mathcal{I}_2\) once robot \(R_2\) also moves. Furthermore, pronouncing cell \((5,7)\) a forbidden state, also results to the further elimination of the moves that are annotated in cells \((5,4)\), \((5,6)\), \((7,4)\) and \((7,6)\). Since all these eliminations leave the corresponding cells empty, these four cells are pronounced forbidden states, as well.

Processing all the identified forbidden states according to the logic that is described in the previous paragraph, we obtain the final Tableau 3(c). Once this tableau is available, the thread computing the supervisor for Mode 2 sends the following message to the corresponding thread for Mode 1:

\[
M^{(1)}_{2,1} = \{(5,1), (5,2), (5,3), (5,4), (5,6), (5,7), (6,2), (6,3), (6,4), (6,7), (7,1), (7,2), (7,3), (7,4), (7,6)\}
\]

(8)

The states included in the above message correspond to cells with no moves for robot \(R_1\) in Tableau 3(c). And since all these states can be accessed uncontrollably from Mode 1, through the switching of the edge \(e_5\), the corresponding thread must know about the relevant result of Table 3(c).

Upon receiving message \(M^{(1)}_{2,1}\) the computational thread for Mode 1 pronounces all the states in this message as forbidden states, and goes into a local computation that is initialized by the result of Table 2 and is similar to the local computations that were described in the above paragraphs. The result of this computation is depicted in Table 4. We leave the verification of the relevant computations to the reader. Furthermore, upon the generation of Table 4, the thread for the computation of the local supervisor for Mode 1 sends the following message to the thread for the generation of the local supervisor for Mode 2:

\[
M^{(1)}_{1,2} = \{(6,5), (7,5)\}
\]

(9)

Message \(M^{(1)}_{1,2}\) contains the local states \((6,5)\) and \((7,5)\) that were identified as forbidden in the last computation for Mode 1, but they were not contained in message \(M^{(1)}_{2,1}\) and therefore, are considered as safe states in Table 3. Hence, upon receiving this message, the computational thread for Mode 2 defines these two states as forbidden, and proceeds with the necessary updating of the prior result in Table 3(c). It should be clear, however, from all the previous discussion, that this updating will render no further states forbidden in this mode, and therefore, Table 3(c) remains unaltered, and no message is passed from the computational thread for Mode 2 to the corresponding thread for Mode 1. Since Mode 1 is also in an idling mode and there are no further messages requiring processing by any of these two threads, the algorithm terminates.

The maximally permissive supervisor that is returned by the above algorithm upon termination, is represented in a distributed manner, across the underlying operational modes, by Tables 4 and 3(c). An intuitive interpretation of the obtained result is as follows: The topology of the considered graph \(G\) in Mode 1 allows robot \(R_1\) to move intelligently upon the two minimal cycles of this graph, in a way that maintains the enforced invariant \(v(R_1) \neq v(R_2)\). On the other hand, the repositioning of edge \(e_5\) in Mode 2, renders the right minimal cycle of graph \(G\) a pretty dangerous place for robot \(R_1\). In particular, the only admissible states \(s\) in which robot \(R_1\) is located in the right minimal cycle (i.e., nodes 5, 6 to 7), are those that enable robot \(R_1\) to rush to the left cycle without being intercepted by robot \(R_2\). Furthermore, since transitions from Mode 1 to Mode 2 can take place in a very uncontrollable manner and at any local state \(s\), the local states \(s\) where robot \(R_1\) is in the right minimal cycle must be treated as dangerous even by the supervisor of Mode 1. This last effect is attained in the above computation through the exchange of the messages \(M^{(1)}_{2,1}\) and \(M^{(1)}_{1,2}\).

REFERENCES
