Supplement Material for the Paper "On the Existence of Non-blocking Bounded Supervisors for Discrete Event Systems"

I. Proof of Theorem 1

Theorem 1: Let G and R be two automata such that $R \sqsubset G$ as usual. Let S be a supervisor. Supervisor S solves Problem 1 if and only if the following statements hold:

1) The language $P(\mathcal{L}(S/G))$ is bounded.
2) For any $\alpha \in P(\mathcal{L}(S/G))$, $Z_S(\alpha)$ is safe.
3) For any $\alpha \in P(\mathcal{L}(S/G))$, we have $\Lambda(Y_S(\alpha)) \subseteq S(\alpha)$.

Proof: It follows from Lemmas 1 and 2 presented later. □

A. Proof of Lemma 1

Lemma 1: Let G and R be two automata such that $R \sqsubset G$ as usual. Let S be a supervisor. If S solves Problem 1, then the following statements hold:

1) The language $P(\mathcal{L}(S/G))$ is bounded.
2) For any $\alpha \in P(\mathcal{L}(S/G))$, $Z_S(\alpha)$ is safe.
3) For any $\alpha \in P(\mathcal{L}(S/G))$, we have $\Lambda(Y_S(\alpha)) \subseteq S(\alpha)$.

Proof: Assume that S solves Problem 1. Then, Statement (1) holds clearly. We next show that Statement (2) holds. By contradiction, suppose that S solves Problem 1 and Statement (2) does not hold. There exists $\alpha \in P(\mathcal{L}(S/G))$ such that $Z_S(\alpha) = (e, S(\alpha)) \in Z$ is deadlock or unobservable unbounded. Let us consider these two cases.

- $Z_S(\alpha) = (e, S(\alpha))$ is deadlock. There exists $x \in e$ such that $x \notin X_m$ and $\delta(x, \sigma)$ is not defined for all $\sigma \in S(\alpha)$. By Proposition 1 and the definition of the closed-loop language, there exists $s \in \mathcal{L}(S/G)$ such that $\delta(x_0, s) = x \notin X_m$, $P(s) = \alpha$, and $s\sigma \notin \mathcal{L}(S/G)$ for all $\sigma \in \Sigma$. Thus, supervisor S is blocking. This is a contradiction due to the fact that S solves Problem 1.
- $Z_S(\alpha) = (e, S(\alpha))$ is unobservable unbounded. There exist a state $x \in e$ and a string $s \in (S(\alpha) \cap \Sigma_{uo})^* \setminus \{\varepsilon\}$ such that $\delta(x, s) = x$. By Proposition 1 and the definition of the closed-loop language, there exists $t \in \mathcal{L}(S/G)$ with $P(t) = \alpha$ and $\delta(x_0, t) = x$ such that $t(s)^n \in \mathcal{L}(S/G)$ for any $n \in \omega$, where $t(s)^0 = t$, $t(s)^1 = ts$, $t(s)^2 = tss$, and so forth. Thus, supervisor S is unbounded. This is a contradiction due to the fact that S solves Problem 1.

Based on the above discussion, if S solves Problem 1, then Statement (2) holds. Finally, we show that Statement (3) holds. By contradiction, suppose that S solves Problem 1 and Statement (3) does not hold. There exists $\alpha \in P(\mathcal{L}(S/G))$ such that $\Lambda(Y_S(\alpha)) \nsubseteq S(\alpha)$. It follows that there exist $x \in Y_S(\alpha) \cap X_r$, $s \in \Sigma_{uo}^*$, and $\sigma \in \Sigma$ such that $\delta_r(x, s\sigma)!$ and $\sigma \notin S(\alpha)$. For $x \in Y_S(\alpha)$, there exists $t \in \mathcal{L}(S/G)$ such that $\delta(x_0, t) = x$ and $P(t) = \alpha$, due to Proposition 1. Thanks to $x \in X_r$ and $R \sqsubset G$, it holds $\delta_r(x_0, t) = x$. Note that $\delta_r(x, s\sigma)!$. Then $ts\sigma \in \mathcal{L}(R)$ holds. However, as a result of $\sigma \notin S(\alpha)$, we have $ts\sigma \notin \mathcal{L}(S/G)$. It follows $\mathcal{L}(R) \nsubseteq \mathcal{L}(S/G)$. By $\overline{C_s} = \mathcal{L}(R)$, we have $C_s \nsubseteq \mathcal{L}(S/G)$, where $C_s = \mathcal{L}_m(R)$ is the control specification. This is a contradiction due to the fact that S solves Problem 1. Therefore, if S solves Problem 1, then Statement (3) holds. This finishes the proof. □

B. Proof of Lemma 2

Lemma 2: Let G and R be two automata such that $R \sqsubset G$ as usual. Let S be a supervisor. Supervisor S solves Problem 1 if the following statements hold:

1) The language $P(\mathcal{L}(S/G))$ is bounded.
2) For any $\alpha \in P(\mathcal{L}(S/G))$, $Z_S(\alpha)$ is safe.
3) For any $\alpha \in P(\mathcal{L}(S/G))$, we have $\Lambda(Y_S(\alpha)) \subseteq S(\alpha)$.

Proof: Assume that the above Statements (1) to (3) hold. We first show that S is bounded. Without loss of generality, $P(\mathcal{L}(S/G))$ is assumed to be l-bounded for some $l \in \omega$. Let $s \in \mathcal{L}(S/G)$. We next show $||s|| \leq (l+1)(|X|-1)+l$. We write $s = u_1\sigma_1u_2\sigma_2\cdots u_n\sigma_nu_{n+1}$, where $u_j \in \Sigma_{uo}^*$ for $j \in [1, n+1]$ and $\sigma_i \in \Sigma_o$ for $i \in [1, n]$. Further, we write $u_j = \sigma_{j_1}\sigma_{j_2}\cdots\sigma_{j_k}$, where $\sigma_{j_v} \in \Sigma_{uo}$ for $v \in [1, k]$. We need to show $||u_j|| \leq |X|-1$.

String u_j induces a sequence $r = x_{j_1}\sigma_{j_1}x_{j_2}\cdots x_{j_k}\sigma_{j_k}x_{j_{k+1}}$ with $x_{j_h} \in X$ for $h \in [1, k+1]$ such that $x_{j_1} = \delta(x_0, u_1\sigma_1u_2\cdots u_{j-1}\sigma_{j-1})$ and $\delta(x_{j_v}, \sigma_{j_v}) = x_{j_{v+1}}$ for $v \in [1, k]$. By the fact that $u_1\sigma_1u_2\cdots u_{j-1}\sigma_{j-1}u_j \in \mathcal{L}(S/G)$, $u_j \in \Sigma_{uo}^*$, and Proposition 1, it holds $x_{j_h} \in X(Z_S(\sigma_1\sigma_2\cdots\sigma_{j-1}))$ for any $h \in [1, k+1]$. Due to the fact that $Z_S(\sigma_1\sigma_2\cdots\sigma_{j-1})$ is unobservable bounded, in the sequence r, for any $d, f \in [1, k+1]$, if $d \neq f$, then $x_{j_d} \neq x_{j_f}$ holds. Thus, we have $k+1 \leq |X|$, which implies $||u_j|| \leq |X|-1$.

Note that $P(\mathcal{L}(S/G))$ is l-bounded and $P(s) = \sigma_1\sigma_2\cdots\sigma_n \in P(\mathcal{L}(S/G))$. It holds $n \leq l$ and then $||s|| \leq (n+1)(|X|-1)+n \leq (l+1)(|X|-1)+l$. It follows that S is bounded.

We next show that S is nonblocking. By contradiction, suppose that Statements (1) to (3) hold and S is blocking. Note that S is bounded as discussed above. There exists $s \in \mathcal{L}(S/G)$ such that $\delta(x_0, s) = x_s \notin X_m$ and $s\sigma \notin \mathcal{L}(S/G)$ for all $\sigma \in \Sigma$. It follows that for all $\sigma \in S(P(s))$, $\delta(x_s, \sigma)$ is not defined. For x_s, by Proposition 1, we know $x_s \in X(Z_S(P(s)))$. Clearly, $Z_S(P(s))$ is deadlock, which contradicts Statement (2). Therefore, if Statements (1) to (3) hold, then S is nonblocking.

Finally, if Statement (3) holds, by using the similar proof for Lemma 1 in [6], we can deduce $\mathcal{L}(R) \subseteq \mathcal{L}(S/G)$. It follows $C_s \subseteq \mathcal{L}(S/G)$, where $C_s = \mathcal{L}_m(R)$ is the control specification. We have shown that S solves Problem 1 if Statements (1) to (3) hold. We complete the proof. □

II. Proof of Proposition 2

Proposition 2: Let G and R be two automata such that $R \sqsubset G$ as usual. The running time of Algorithm 1 is of $O(2^{|X|+|\Sigma_c|}(|\Sigma_{uo}||X|^2))$.

Proof: In the worst case, Algorithm 1 will encounter at most $2^{|X|}$ Y-states. For each Y-state y encountered, computing $\Lambda(y)$ can be done in time $O(|X_r||\Sigma_{uo}|)$. Moreover, for each y, a maximum of $2^{|\Sigma_c|}$ control decisions γ should be considered. For each γ, computing $UR(y,\gamma)$ can be done in time $O(|X||\Sigma_{uo}|)$. The total running time for Y-states is of $n_1 = O(2^{|X|}(|X_r||\Sigma_{uo}| + 2^{|\Sigma_c|}(|X||\Sigma_{uo}|)))$.

Next, we touch upon Z-states. In the worst case, Algorithm 1 will encounter at most $2^{|X|+|\Sigma_c|}$ Z-states. For each Z-state z encountered, determining whether or not z is safe can be done in time $O(|\Sigma_{uo}||X|^2)$. Moreover, for each z, a maximum of $|\Sigma_o|$ observable events σ should be considered. For each σ, computing $OR(z,\sigma)$ can be done in time $O(|X|)$. The total running time for Z-states is of $n_2 = O(2^{|X|+|\Sigma_c|}(|\Sigma_{uo}||X|^2 + |\Sigma_o||X|))$. Combining the above together, the total running time of the algorithm is of $O(n_1+n_2)$, which can be simplified to $O(2^{|X|+|\Sigma_c|}(|\Sigma_{uo}||X|^2))$. □

III. Proof of Theorem 2

Theorem 2: Let $\mathfrak{R} = (\mathfrak{A}(G,R), \mathcal{W})$ be a reachability game, $\mathbb{W}_0$ be the winning region of Player 0, and $f_{\mathbb{W}_0}$ be a $\mathbb{W}_0$-based positional strategy for Player 0. Problem 1 has a solution if and only if $y_0 \in \mathbb{W}_0$. Moreover, if $y_0 \in \mathbb{W}_0$, then the $f_{\mathbb{W}_0}$-based supervisor $S_{f_{\mathbb{W}_0}}$ solves Problem 1.

Proof: It follows from Lemmas 3 and 4 presented later. □

A. Proof of Lemma 3

Before presenting Lemma 3, we first provide the following property.

Property 1: Let $\mathfrak{R} = (\mathfrak{A}(G,R), \mathcal{W})$ be a reachability game, $\mathbb{W}_0$ be the winning region of Player 0, $f_{\mathbb{W}_0}$ be a $\mathbb{W}_0$-based positional strategy for Player 0, and $S_{f_{\mathbb{W}_0}}$ be the $f_{\mathbb{W}_0}$-based supervisor. If $y_0 \in \mathbb{W}_0$, then the following statements hold:

i) for all $\alpha \in P(\mathcal{L}(S_{f_{\mathbb{W}_0}}/G))$, there exists a generation sequence r of α in $\mathfrak{A}(G,R)$ under $f_{\mathbb{W}_0}$.
ii) the above sequence r is also the estimation sequence of α under $S_{f_{\mathbb{W}_0}}$.

Proof: We show that Statements (i) and (ii) hold by induction on the length of strings $\alpha \in P(\mathcal{L}(S_{f_{\mathbb{W}_0}}/G))$.

Induction basis: Let $||\alpha|| = 0$ (i.e., $\alpha = \varepsilon$). We have $y_0 \in (V_0 \cap \mathbb{W}_0) \setminus \mathcal{W}$, due to $y_0 \notin \mathcal{W}$ and $y_0 \in \mathbb{W}_0 \cap V_0$. By Definition 11, $f_{\mathbb{W}_0}(y_0) = z_0$ such that $z_0 \in y_0^\bullet \cap \mathbb{W}_0$ is a state in V_1. Hence, there exists $\gamma_0 \in \Gamma$ such that $\Theta_{01}(y_0,\gamma_0) = z_0$. Thanks to Algorithm 1, $\Lambda(y_0) \subseteq \gamma_0$ and $z_0 = UR(y_0,\gamma_0)$ hold. Construct a sequence $r_0 = y_0\gamma_0 z_0$. Obviously, r_0 is the generation sequence of ε in $\mathfrak{A}(G,R)$ under $f_{\mathbb{W}_0}$. By Definition 13, it holds $\gamma_0 = S_{f_{\mathbb{W}_0}}(\varepsilon)$. As a result of $z_0 = UR(y_0,\gamma_0)$, the sequence r_0 is also the estimation sequence of ε under $S_{f_{\mathbb{W}_0}}$. The induction basis holds.

Induction hypothesis: Assume that when $||\alpha|| \le n$, the Statements (i) and (ii) hold.

Induction step: Let $\alpha\sigma_{n+1} \in P(\mathcal{L}(S_{f_{\mathbb{W}_0}}/G))$ be a string such that $||\alpha|| = n$ and $\sigma_{n+1} \in \Sigma_o$. We write $\alpha = \sigma_1\sigma_2\cdots\sigma_n$, where $\sigma_i \in \Sigma_o$. By the induction hypothesis, there exists a generation sequence $r_n = y_0\gamma_0 z_0\sigma_1 y_1\gamma_1 z_1\cdots\sigma_n y_n\gamma_n z_n \in (V_0\Gamma V_1)(\Sigma_o V_0\Gamma V_1)^*$ of α in $\mathfrak{A}(G,R)$ under $f_{\mathbb{W}_0}$, and r_n is also the estimation sequence of α under $S_{f_{\mathbb{W}_0}}$. It follows that $UR(y_n,\gamma_n) = z_n$, $\gamma_n = S_{f_{\mathbb{W}_0}}(\alpha) = \Gamma(z_n)$, $Y_{S_{f_{\mathbb{W}_0}}}(\alpha) = y_n$, $Z_{S_{f_{\mathbb{W}_0}}}(\alpha) = z_n$, and $f_{\mathbb{W}_0}(y_n) = z_n$. By $f_{\mathbb{W}_0}(y_n) = z_n$ and Definition 11, $z_n \in \mathbb{W}_0$ holds.

We next show $OR(z_n,\sigma_{n+1}) = \{x \in X | \exists x' \in X(z_n) : x = \delta(x',\sigma_{n+1})\} \neq \emptyset$. Note that $\alpha\sigma_{n+1} \in P(\mathcal{L}(S_{f_{\mathbb{W}_0}}/G))$. There exists $t \in \mathcal{L}(S_{f_{\mathbb{W}_0}}/G)$ such that $P(t) = \alpha$ and $t\sigma_{n+1} \in \mathcal{L}(S_{f_{\mathbb{W}_0}}/G)$. Let $x = \delta(x_0,t)$. It holds $x \in X(z_n)$, due to $Z_{S_{f_{\mathbb{W}_0}}}(\alpha) = z_n$ and Proposition 1. Thanks to $t\sigma_{n+1} \in \mathcal{L}(S_{f_{\mathbb{W}_0}}/G)$, it holds $\delta(x,\sigma_{n+1})!$ and thus $OR(z_n,\sigma_{n+1}) \neq \emptyset$. Let $y_{n+1} = OR(z_n,\sigma_{n+1})$.

Due to $t\sigma_{n+1} \in \mathcal{L}(S_{f_{\mathbb{W}_0}}/G)$ and $P(t) = \alpha$, we have $\sigma_{n+1} \in \gamma_n = S_{f_{\mathbb{W}_0}}(\alpha) = \Gamma(z_n)$. By $\sigma_{n+1} \in \Sigma_o \cap \Gamma(z_n)$, $z_n \in V_1$, $OR(z_n,\sigma_{n+1}) = y_{n+1}$, and Algorithm 1, $\Theta_{10}(z_n,\sigma_{n+1}) = y_{n+1}$ holds. Thanks to $f_{\mathbb{W}_0}(y_n) = z_n \in \mathbb{W}_0$, $y_{n+1} \in z_n^\bullet$, and Algorithm 2, we have $y_{n+1} \in (V_0 \cap \mathbb{W}_0) \setminus \mathcal{W}$. As a result of Definition 11, it holds $f_{\mathbb{W}_0}(y_{n+1}) = z_{n+1}$ for some $z_{n+1} \in y_{n+1}^\bullet \cap \mathbb{W}_0$. Thus, there exists $\gamma_{n+1} \in \Gamma$ such that $\Theta_{01}(y_{n+1},\gamma_{n+1}) = z_{n+1}$. By Algorithm 1, $\Lambda(y_{n+1}) \subseteq \gamma_{n+1}$ and $z_{n+1} = UR(y_{n+1},\gamma_{n+1})$ hold. Construct a sequence $r_{n+1} = r_n\sigma_{n+1}y_{n+1}\gamma_{n+1}z_{n+1}$. Then r_{n+1} is the generation sequence of $\alpha\sigma_{n+1}$ in $\mathfrak{A}(G,R)$ under $f_{\mathbb{W}_0}$. By Definition 13, it holds $\gamma_{n+1} = S_{f_{\mathbb{W}_0}}(\alpha\sigma_{n+1})$. Clearly, r_{n+1} is also the estimation sequence of $\alpha\sigma_{n+1}$ under $S_{f_{\mathbb{W}_0}}$. We complete the induction step. □

Lemma 3: Let $\mathfrak{R} = (\mathfrak{A}(G,R), \mathcal{W})$ be a reachability game, $\mathbb{W}_0$ be the winning region of Player 0, and $f_{\mathbb{W}_0}$ be a $\mathbb{W}_0$-based positional strategy for Player 0. The $f_{\mathbb{W}_0}$-based supervisor $S_{f_{\mathbb{W}_0}}$ solves Problem 1 if $y_0 \in \mathbb{W}_0$.

Proof: The idea of the proof is to show that if $y_0 \in \mathbb{W}_0$, the three statements w.r.t. $S_{f_{\mathbb{W}_0}}$ in Theorem 1 hold. Let $\alpha = \sigma_1\sigma_2\cdots\sigma_n \in P(\mathcal{L}(S_{f_{\mathbb{W}_0}}/G))$, where $\sigma_i \in \Sigma_o$. By Statements (i) and (ii) in Property 1, there exists a generation sequence $r = y_0\gamma_0 z_0\sigma_1 y_1\gamma_1 z_1\cdots\sigma_n y_n\gamma_n z_n \in (V_0\Gamma V_1)(\Sigma_o V_0\Gamma V_1)^*$ of α in $\mathfrak{A}(G,R)$ under $f_{\mathbb{W}_0}$, and r is also the estimation sequence of α under $S_{f_{\mathbb{W}_0}}$. It follows that $z_j = f_{\mathbb{W}_0}(y_j)$, $z_j \in y_j^\bullet$ for $j \in [0,n]$, $y_{j'} \in z_{j'-1}^\bullet$ for $j' \in [1,n]$, $Z_{S_{f_{\mathbb{W}_0}}}(\alpha) = z_n$, $Y_{S_{f_{\mathbb{W}_0}}}(\alpha) = y_n$, $\gamma_n = S_{f_{\mathbb{W}_0}}(\alpha)$, and $z_n = \Theta_{01}(y_n,\gamma_n)$. By $z_n = \Theta_{01}(y_n,\gamma_n)$ and Algorithm 1, we have $\Lambda(y_n) = \Lambda(Y_{S_{f_{\mathbb{W}_0}}}(\alpha)) \subseteq \gamma_n = S_{f_{\mathbb{W}_0}}(\alpha)$ and $z_n = Z_{S_{f_{\mathbb{W}_0}}}(\alpha)$ is safe. Statements (2) and (3) w.r.t. $S_{f_{\mathbb{W}_0}}$ in Theorem 1 hold.

We next still consider the above observation string α and show that $P(\mathcal{L}(S_{f_{\mathbb{W}_0}}/G))$ is bounded. We need to show that in the above sequence r, for any two different $f,g \in [0,n]$, it holds $y_f \neq y_g$. By contradiction, suppose that $y_0 \in \mathbb{W}_0$ and in the sequence r, there exist two different $f, g \in [0, n]$ such that $y_f = y_g$. Without loss of generality, assume $f < g$. Recall that in the sequence r, it holds $z_j = f_{\mathbb{W}_0}(y_j)$ and $z_j \in y_j^\bullet$ for $j \in [0, n]$, and $y_{j'} \in z_{j'-1}^\bullet$ for $j' \in [1, n]$. There exists an infinite play $\pi = y_0 z_0 y_1 z_1 \cdots y_f (z_f y_{f+1} z_{f+1} \cdots y_{f+(g-f-1)} z_{f+(g-f-1)} y_g)^w \in V^w$ starting at y_0 that is consistent with $f_{\mathbb{W}_0}$ and that is not winning for Player 0 by the fact that no terminal state in V_1 occurs in the play. Therefore, $f_{\mathbb{W}_0}$ is not winning for Player 0 from y_0. This is a contradiction due to the fact that $f_{\mathbb{W}_0}$ is winning for Player 0 from y_0 (see Remark 2). It follows that if $y_0 \in \mathbb{W}_0$, in the sequence r, for any two different $f, g \in [0, n]$, it holds $y_f \neq y_g$. Due to $V_0 \subseteq 2^X$, we have $n \leq 2^{|X|}$ and thus $||\alpha|| = n \leq 2^{|X|}$. It follows that if $y_0 \in \mathbb{W}_0$, then $P(\mathcal{L}(S_{f_{\mathbb{W}_0}}/G))$ is $2^{|X|}$-bounded. Statements (1) to (3) w.r.t. $S_{f_{\mathbb{W}_0}}$ in Theorem 1 hold, provided that $y_0 \in \mathbb{W}_0$. It follows that if $y_0 \in \mathbb{W}_0$, $S_{f_{\mathbb{W}_0}}$ solves Problem 1. We complete the proof. □

B. Proof of Lemma 4

Lemma 4: Given a reachability game $\mathfrak{R} = (\mathfrak{A}(G, R), \mathcal{W})$ and the winning region $\mathbb{W}_0$ of Player 0, if $y_0 \notin \mathbb{W}_0$, then Problem 1 has no solution.

Proof: By contradiction, suppose that $y_0 \notin \mathbb{W}_0$ and there exists a supervisor S solving Problem 1. By Theorem 1, the following statements hold: (a) $P(\mathcal{L}(S/G))$ is bounded; (b) for any $\alpha \in P(\mathcal{L}(S/G))$, $Z_S(\alpha)$ is safe; and (c) for any $\alpha \in P(\mathcal{L}(S/G))$, we have $\Lambda(Y_S(\alpha)) \subseteq S(\alpha)$.

We next show that for any $n \in w$, there exists a sequence $r_n = y_0\gamma_0 z_0 \sigma_1 y_1 \gamma_1 z_1 \cdots \sigma_n y_n \gamma_n z_n \in (V_0 \Gamma V_1)(\Sigma_o V_0 \Gamma V_1)^*$ such that: (i) the sequence r_n is generated by $\mathfrak{A}(G, R)$ and $z_n \notin \mathbb{W}_0$; (ii) $\alpha_n = \sigma_1\sigma_2 \cdots \sigma_n \in P(\mathcal{L}(S/G))$; and (iii) the sequence r_n is the estimation sequence of α_n under S. We prove Statements (i) to (iii) by induction on the number of n.

Induction basis: Let $n = 0$ and $\alpha_0 = \varepsilon$. Note that $\alpha_0 \in P(\mathcal{L}(S/G))$. There exists $r_0 = y_0\gamma_0 z_0 \in Y\Gamma Z$ that is the estimation sequence of α_0 under S. It follows that $Y_S(\varepsilon) = y_0$, $Z_S(\varepsilon) = z_0$, $\gamma_0 = S(\varepsilon)$, and $z_0 = UR(y_0, \gamma_0)$. By statements (b) and (c), we have $\Lambda(Y_S(\varepsilon)) = \Lambda(y_0) \subseteq \gamma_0$ and $z_0 = Z_S(\varepsilon)$ is safe. Note that $y_0 \in V_0$. Thanks to Algorithm 1, $\Theta_{01}(y_0, \gamma_0) = z_0 \in V_1$ holds. Thus, the sequence r_0 is generated by $\mathfrak{A}(G, R)$. By $y_0 \notin \mathbb{W}_0$, $z_0 \in y_0^\bullet$, and Algorithm 2, $z_0 \notin \mathbb{W}_0$ holds. The induction basis holds.

Induction hypothesis: Assume that when $n = l$, there exists $r_l = y_0\gamma_0 z_0 \sigma_1 y_1 \gamma_1 z_1 \cdots \sigma_l y_l \gamma_l z_l \in (V_0 \Gamma V_1)(\Sigma_o V_0 \Gamma V_1)^*$ such that: (1) r_l is generated by $\mathfrak{A}(G, R)$ and $z_l \notin \mathbb{W}_0$; (2) $\alpha_l = \sigma_1\sigma_2 \cdots \sigma_l \in P(\mathcal{L}(S/G))$; and (3) r_l is the estimation sequence of α_l under S.

Induction step: Let $n = l+1$. By the induction hypothesis, the sequence r_l is the estimation sequence of α_l under S. It follows that $Y_S(\alpha_l) = y_l$, $Z_S(\alpha_l) = z_l$, $z_l = UR(y_l, \gamma_l)$, and $\gamma_l = S(\alpha_l) = \Gamma(z_l)$. By $\alpha_l \in P(\mathcal{L}(S/G))$ and Statements (b) and (c), it holds $\Lambda(Y_S(\alpha_l)) = \Lambda(y_l) \subseteq \gamma_l$ and $z_l = Z_S(\alpha_l)$ is safe. Thanks to r_l is generated by $\mathfrak{A}(G, R)$, $\Theta_{01}(y_l, \gamma_l) = z_l$ holds. Due to the induction hypothesis, $z_l \notin \mathbb{W}_0$ holds and then $z_l \notin \mathcal{W}$. Thus, z_l is not terminal. By the fact that $z_l \notin \mathbb{W}_0$, z_l is not terminal, and Algorithm 2, there exists $y_{l+1} \in z_l^\bullet$ such that $y_{l+1} \notin \mathbb{W}_0$. Let $\sigma_{l+1} \in \Sigma_o$ be an event such that $\Theta_{10}(z_l, \sigma_{l+1}) = y_{l+1} \in V_0$. As a result of Algorithm 1, $y_{l+1} = OR(z_l, \sigma_{l+1}) \neq \emptyset$ and $\sigma_{l+1} \in \Gamma(z_l)$ hold. We next show $\alpha_l \sigma_{l+1} \in P(\mathcal{L}(S/G))$.

By Proposition 1, we have $X(Z_S(\alpha_l)) = X(z_l) = \{x \in X | \exists s \in \mathcal{L}(S/G) : [x = \delta(x_0, s)] \wedge [P(s) = \alpha_l]\}$. Due to $OR(z_l, \sigma_{l+1}) = \{x \in X | \exists x' \in X(z_l) : x = \delta(x', \sigma_{l+1})\} \neq \emptyset$, there exists $s \in \mathcal{L}(S/G)$ such that $P(s) = \alpha_l$ and $s\sigma_{l+1} \in \mathcal{L}(G)$. Thanks to $\sigma_{l+1} \in \Gamma(z_l) = S(\alpha_l) = S(P(s))$, it holds $s\sigma_{l+1} \in \mathcal{L}(S/G)$ and thus $\alpha_{l+1} = \alpha_l \sigma_{l+1} \in P(\mathcal{L}(S/G))$. Let $\gamma_{l+1} = S(\alpha_{l+1})$, $z_{l+1} = UR(y_{l+1}, \gamma_{l+1})$, and $r_{l+1} = r_l \sigma_{l+1} y_{l+1} \gamma_{l+1} z_{l+1}$. Clearly, r_{l+1} is the estimation sequence of α_{l+1} under S. It follows $Y_S(\alpha_{l+1}) = y_{l+1}$ and $Z_S(\alpha_{l+1}) = z_{l+1}$. By $\alpha_{l+1} \in P(\mathcal{L}(S/G))$ and Statements (b) and (c), z_{l+1} is safe and $\Lambda(y_{l+1}) \subseteq S(\alpha_{l+1}) = \gamma_{l+1}$. Thanks to $y_{l+1} \in V_0$, $z_{l+1} = UR(y_{l+1}, \gamma_{l+1})$, and Algorithms 1, it holds $z_{l+1} \in V_1$ and $\Theta_{01}(y_{l+1}, \gamma_{l+1}) = z_{l+1}$. Clearly, r_{l+1} is generated by $\mathfrak{A}(G, R)$. Moreover, by $y_{l+1} \notin \mathbb{W}_0$ and $z_{l+1} \in y_{l+1}^\bullet$, it holds $z_{l+1} \notin \mathbb{W}_0$. The induction step holds.

Based on the above discussion, if $y_0 \notin \mathbb{W}_0$ and there exists a supervisor S solving Problem 1, then for any $n \in w$, there exists a string $\sigma_1\sigma_2 \cdots \sigma_n \in P(\mathcal{L}(S/G))$, where $\sigma_i \in \Sigma_o$. It follows that $P(\mathcal{L}(S/G))$ is not bounded, which contradicts Statement (a). If $y_0 \notin \mathbb{W}_0$, then Problem 1 has no solution. We complete the proof. □

References

[1] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Cont. Opt.,* vol. 25, no. 1, pp. 206–230, 1987.

[2] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems,* 2nd ed. Springer, 2008.

[3] Y. Brave and M. Heymann, "Stabilization of discrete-event processes," *Int. J. Control*, vol. 51, no. 5, pp. 1101–1117, 1990.

[4] K. W. Schmidt and C. Breindl, "A framework for state attraction of discrete event systems under partial observation," *Inf. Sci.,* vol. 281, no. 10, pp. 265–280, 2014.

[5] X. Yin and S. Lafortune, "A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 61, no. 8, pp. 2140–2154, 2016.

[6] X. Yin and S. Lafortune, "Synthesis of maximally-permissive non-blocking supervisors for the lower-bound containment problem," *IEEE Trans. Autom. Control,* vol. 63, no. 12, 4435–4441, 2018.

[7] H. Cho and S. I. Marcus, "On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation," *Math. Contr. Sig. Syst.,* vol. 2, no. 1, 47–69, 1989.

[8] X. Yin and S. Lafortune, "Synthesis of maximally-permissive supervisors for the range control problem," *IEEE Trans. Autom. Control,* vol. 62, no. 8, pp. 3914–3929, 2017.

[9] E. Gradel, W. Thomas, and T. Wilke, *Automata Logics, and Infinite Games: A Guide to Current Research.* vol. 2500. Springer Science & Business Media, 2002.

[10] K. R. Apt and E. Gradel, *Lectures in Game Theory for Computer Scientists*. Cambridge University Press, 2011.