Revista Ingeniería UC, Vol. 24, No. 1, abril 2017 81 - 90

Supervisory control application to solving optimal control problems for discrete event systems

Guelvis Mata

Departamento de Matemáticas, Facultad de Ciencias, Universidad de los Andes, Márida, Venezuela

Abstract.-

This paper deals with the supervisory control in Discrete Event Systems and its aplication for the problems resolution corresponding to the optimun control, which is related to a situation where the control is execute to satisfy a series of maximal or minimal cualitative specifications on the language generated or marked for a discrete event systems.

Keywords: discrete event systems; automaton; supervisory control

Aplicación de control supervisorio en la resolución de problemas de control óptimo para sistemas de eventos discretos

Resumen.-

Este trabajo trata sobre el control de supervisión en sistemas de eventos discretos y su aplicación para la resolución de problemas correspondiente al control optimum, que se relaciona con una situación en la que el control se ejecuta para satisfacer una serie de especificaciones cualitativas máximas o mínimas en el Lenguaje generado o marcado para sistemas de eventos discretos.

Palabras clave: sistemas de eventos discretos; autómata; control supervisorio

Recibido: octubre 2016
Aceptado: marzo 2017

1. Introducción

La clase de los sistemas de eventos discretos (SED) constituye un área de estudio con importancia creciente dada la necesidad, sofisticación y exigencia del mundo actual. Por lo tanto, es de gran interés desarrollar formalismos adaptados a ésta categoría de sistemas.

Los modelos para representar SED son edificados a partir de dos nociones básicas: estados y eventos; las cuales son expresadas mediante conjuntos. Un estado es una descripción lógica del sistema en el tiempo y un evento es una acción. Así, el comportamiento dinámico (ó dinámica) de los SED queda plenamente determinado por sucesiones finitas de estados o equivalentemente por sucesiones finitas de eventos. Estos SED incluyen a los sistemas de manufactura, a las redes de comunicación, a los sistemas de información, a los sistemas de computación, etc. Adicionalmente, el control inapropiado de la ocurrencia de eventos puede conducir a éstos sistemas a estancamientos, a sobreflujos de capacidad, etc; degradando la ejecución deseable. Estas situaciones deben ser minimizadas en algún sentido o eliminadas ya que necesitamos que los SED sean funcionales o lógicamente correctos en términos de propiedades deseables. De hecho, en este artículo nosotros trataremos el control supervisorio en SED y su aplicación en control óptimo; lo cual corresponde a una situación donde el control es ejercido para satisfacer un conjunto de especificaciones cuali-

Correo-e: `gmata@ula.ve` (Guelvis Mata)

tativas sobre los lenguajes generado y marcado por un SED. El punto de vista adoptado es que algunos de los comportamientos particulares que el sistema puede generar son ilegales, y en consecuencia es necesario que estos sean inhabilitados bajo control. Luego, el control es ejercido para reducir el conjunto de sucesiones de eventos que el sistema puede generar a un subconjunto legal, y también es establecido para eliminar o minimizar estancamiento en algunos SED. Este paradigma fue iniciado por Wonham en 1982 y actualmente es estudiado extensivamente. Nosotros presentaremos algunos resultados básicos de la teoría de control supervisorio focalizados sobre la no controlabilidad, junto con su aplicación a problemas de interés en los SED.

2. Preliminares

Esta sección incluye las definiciones básicas de las teorías de lenguajes y autómatas, necesarias para el estudio de la teoría de control supervisorio.

Sea Σ un conjunto finito no vacío. Consideremos el conjunto Σ^* constituido por las n-uplas,

$$s = (\alpha_1, \alpha_2, \ldots, \alpha_n),$$

$\alpha_i \in \Sigma$, $n \geq 1$, junto con la 0-upla $\theta := (\quad)$.

Sobre Σ^* definimos una operación binaria, llamada concatenación, como sigue: dados $s, t \in \Sigma^*$, con $s = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ y $t = (\beta_1, \beta_2, \ldots, \beta_m)$, entonces $st = (\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_m) \in \Sigma^*$. Así el par $(\Sigma^*, \cdot)$ es un monoide con unidad θ.

Claramente $|st| = |s| + |t|$ y $|\theta| = 0$. Ahora bien, si acordamos escribir α en lugar de la 1–upla (α), entonces $s = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ puede ser escrita como $s = \alpha_1 \cdot \alpha_2 \cdots \alpha_n$ o simplemente $s = \alpha_1\alpha_2 \cdots \alpha_n$, si $n > 0$.

En consecuencia, justificadamente α, s y Σ serán llamadas respectivamente una letra, una palabra y un alfabeto. Finalmente, la convención $\alpha = (\alpha)$ nos permite tratar a Σ como un subconjunto de Σ^*.

Un lenguaje $\mathcal{L}$ sobre Σ es cualquier subconjunto de Σ^*; luego la unión, intersección, diferencia y complemento de lenguajes son lenguajes sobre Σ. Más aún, si $\mathcal{L}_1$ y $\mathcal{L}_2$ son lenguajes sobre un alfabeto Σ entonces

$$\mathcal{L}_1\mathcal{L}_2 := \{s \in \Sigma^* : s = s_1 s_2,\ s_1 \in \mathcal{L}_1,\ s_2 \in \mathcal{L}_2\}$$

y

$$\overline{\mathcal{L}_1} := \{s \in \Sigma^* : \exists\ t \in \Sigma^*,\ st \in \mathcal{L}_1\}$$

son lenguajes sobre Σ.

Sea $s \in \Sigma^*$, $t \in \Sigma^*$ será llamado un segmento de s si $s = utv$, para algunos $u, v \in \Sigma^*$. Si $u = \theta$, entonces t será llamado un prefijo de s. Si $v = \theta$ entonces t será llamado un sufijo de s. Por lo tanto, $\overline{\mathcal{L}_1}$ es el conjunto de todos los prefijos de todas las palabras en $\mathcal{L}_1$. Claramente, por definición, $\mathcal{L}_1 \subseteq \overline{\mathcal{L}_1}$. Cuando $\mathcal{L}_1 = \overline{\mathcal{L}_1}$ entonces $\mathcal{L}_1$ será llamado prefijo–cerrado o simplemente cerrado [1].

Si nosotros queremos modelar un SED mediante un lenguaje, simplemente representamos los eventos con símbolos para generar un alfabeto Σ y la dinámica por un lenguaje $\mathcal{L}$ sobre Σ. Luego, el par $(\Sigma, \mathcal{L})$ es un modelo para el SED.

Un autómata determinístico, denotado $\mathcal{A}$, es un sextuple

$$\mathcal{A} := (Q, \Sigma, \delta, \mathcal{E}, q_0, Q_m),$$

donde Q es un conjunto discreto cuyos elementos serán llamados estados, Σ es un conjunto finito llamado alfabeto y cuyos elementos serán llamados eventos, $\delta : Q \times \Sigma \rightarrow Q$ es una función parcial: definida en algún subconjunto propio de $Q \times \Sigma$; llamada transición: $\delta(q, \alpha) = q'$ significa que hay una transición desde el estado q al estado q' etiquetada por el evento α; $\mathcal{E} : Q \rightarrow 2^\Sigma$ es una función llamada función de eventos activos: $\mathcal{E}(q)$ es el conjunto de todos los eventos α tales que $\delta(q, \alpha)$ está definida; $q_0 \in Q$ es un estado llamado inicial y $Q_m \subseteq Q$ es un subconjunto de estados llamados marcados. Si Q es finito, entonces el autómata determinístico $\mathcal{A}$ será llamado autómata finito determinístico (AFD).

Si δ es la función parcial de transición de un autómata determinístico $\mathcal{A}$, entonces nosotros siempre extenderemos recursivamente δ a la función parcial $\widehat{\delta} : Q \times \Sigma^* \longrightarrow Q$ por $\widehat{\delta}(q, \theta) = q$ y $\widehat{\delta}(q, s\alpha) = \delta(\widehat{\delta}(q, s), \alpha)$ con $q \in Q$, $s \in \Sigma^*, \alpha \in \Sigma$. Por lo tanto, en lo que sigue no se hará distinción entre δ y $\widehat{\delta}$; más aún, se puede pensar a $\mathcal{A}$ como un grafo dirigido: los vértices son los estados de Q y los arcos (q, q') con etiqueta α son transiciones $\delta(q, \alpha) = q'$. En consecuencia, podemos considerar las trayectorias o caminos del grafo que son iniciadas en q_0 y entre estas, todas las trayectorias que finalizan en cualquier estado marcado $q_m \in Q_m$; más precisamente, sea $\mathcal{A}$ un autómata determinístico, los lenguajes

$$\mathcal{L}(\mathcal{A}) := \{s \in \Sigma^* : \delta(q_0, s) \text{ está definida}\}$$

y

$$\mathcal{L}_m(\mathcal{A}) := \{s \in \mathcal{L}(\mathcal{A}) : \delta(q_0, s) \in Q_m\}$$

serán llamados respectivamente lenguaje generado y lenguaje marcado por $\mathcal{A}$.

Note, desde la definición previa, que $\mathcal{L}(\mathcal{A})$ es cerrado mientras que $\mathcal{L}_m(\mathcal{A})$ no necesariamente lo es.

Dado que nosotros siempre interpretaremos a los estados marcados de un autómata $\mathcal{A}$ como estados que determinan completaciones de tareas u operaciones, introducimos la noción de bloqueo.

Desde las definiciones de $\mathcal{L}_m(\mathcal{A})$ y $\mathcal{L}(\mathcal{A})$ es claro que $\mathcal{L}_m(\mathcal{A}) \subseteq \overline{\mathcal{L}_m}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A})$, pero en general $\mathcal{L}(\mathcal{A}) \nsubseteq \overline{\mathcal{L}_m}(\mathcal{A})$. Por ejemplo, un autómata $\mathcal{A}$ podría alcanzar un estado q tal que $q \notin Q_m$ y $\mathcal{E}(q) = \emptyset$. Esta situación indeseable es llamada un estancamiento porque el sistema entra a un estado en el que ningún evento puede ser ejecutado, sin haber completado una tarea. También, el autómata $\mathcal{A}$ podría entrar a un ciclo de estados no marcados sin transiciones fuera de este. En este caso, el sistema siempre podrá ejecutar algún evento pero nunca podrá completar una tarea. Esto es llamado vivencia. Finalmente, cuando estemos en presencia de estancamiento o vivencia simplemente diremos que el sistema es bloqueado. Esta noción de bloqueo es muy importante cuando controlamos un SED: siempre queremos minimizar o evitar el bloqueo. Formalmente, un autómata $\mathcal{A}$ será llamado no bloqueado si $\overline{\mathcal{L}_m}(\mathcal{A}) = \mathcal{L}(\mathcal{A})$.

Sea $\mathcal{A}$ un autómata y consideremos los lenguajes generado y marcado $\mathcal{L}(\mathcal{A})$ y $\mathcal{L}_m(\mathcal{A})$ respectivamente. Si eliminamos del grafo de $\mathcal{A}$ los estados que no son alcanzables o accesibles desde el estado inicial por una palabra en $\mathcal{L}(\mathcal{A})$, entonces el autómata resultante tiene igualmente como lenguajes generado y marcado respectivamente a los lenguajes $\mathcal{L}(\mathcal{A})$ y $\mathcal{L}_m(\mathcal{A})$. Note que eliminar un estado de $\mathcal{A}$ bajo la condición impuesta anteriormente significa igualmente eliminar las transiciones que están atadas a dicho estado.

Por otro lado, nosotros podemos obtener un autómata eliminando del grafo de $\mathcal{A}$ los estados accesibles desde el estado inicial que no pueden ser llevados a un estado marcado por una palabra en $\mathcal{L}_m(\mathcal{A})$. Este autómata posee claramente como lenguaje marcado a $\mathcal{L}_m(\mathcal{A})$, pero su lenguaje generado podría estar contenido propiamente en $\mathcal{L}(\mathcal{A})$. Finalmente, dado un autómata $\mathcal{A}$, las consideraciones de eliminación de estados antes expuestas conducen a la construcción de un autómata determinístico limpio en el sentido que siempre puede completar tareas desde el comienzo de la evolución del sistema.

Notemos que si $\mathcal{A}_a(\mathcal{A})$ y $\mathcal{A}_c(\mathcal{A})$ denotan respectivamente los autómatas accesible: construido desde $\mathcal{A}$ por eliminación de los estdos no alcanzables desde q_0; y coaccesibles: construido desde $\mathcal{A}$ por eliminación de los estados que no conducen a estados marcados desde q_0; entonces, $\mathcal{A}$ es limpio sí, y solo si, $\mathcal{A} = \mathcal{A}_a(\mathcal{A}_c(\mathcal{A})) = \mathcal{A}_c(\mathcal{A}_a(\mathcal{A}))$.

En lo que sigue siempre asumiremos que dado un autómata $\mathcal{A}$ éste es accesible.

Dado un lenguaje infinito $\mathcal{L}$ sobre un alfabeto Σ, siempre podemos construir un autómata $\mathcal{A}$ tal que $\mathcal{L} = \mathcal{L}_m(\mathcal{A})$: por ejemplo, como un árbol cuyo nodo raíz es el estado inicial y donde los nodos, representando los estados marcados, son incorporados por la longitud de las palabras en $\mathcal{A}$.

La clase de lenguajes sobre un alfabeto Σ que tienen representación por autómatas finitos es muy importante: por ejemplo si se quiere almacenar en memoria para ejecutar cálculos o si se necesita almacenar una política de control. Formalmente, un lenguaje $\mathcal{L}$ sobre un alfabeto Σ será llamado regular si existe un AFD $\mathcal{A}$ tal que $\mathcal{L} = \mathcal{L}_m(\mathcal{A})$ [1, 3].

3. Realimentación

Sea Σ un alfabeto representando el conjunto de eventos de un SED. Sean $\mathcal{L} = \overline{\mathcal{L}}$ y $\mathcal{L}_m \subseteq \mathcal{L}$ lenguajes sobre Σ representando, a un nivel lógico de abstracción, el conjunto de sucesiones finitas de eventos que el sistema puede generar y el subconjunto de sucesiones que determinan las completaciones de operaciones o tareas respectivamente. Sea $\mathcal{A}$ un autómata determinístico representando al par de lenguajes $\mathcal{L}$ y $\mathcal{L}_m$: $\mathcal{L} = \mathcal{L}(\mathcal{A})$ y $\mathcal{L}_m(\mathcal{A}) = \mathcal{L}_m$. Así, nosotros nos referiremos al SED como $\mathcal{A}$.

Particionamos a Σ en dos subconjuntos disjuntos $\Sigma = \Sigma_c \cup \Sigma_{nc}$, donde Σ_c representa al conjunto de eventos controlables: eventos cuya ocurrencia puede ser prevenida o inhabilitada bajo control; y Σ_{nc} representa al conjunto de eventos no controlables. Hay muchas razones por las que un evento puede ser modelado como no controlable; por ejemplo, si éste es inherentemente imposible de controlar (evento destrucción), si éste tiene alta prioridad (no puede ser inutilizado), etc.

Ahora bien, supongamos que la función de transición de $\mathcal{A}$ puede ser controlada por un agente externo, en el sentido que los eventos controlables pueden ser permitidos o inhabilitados por un controlador externo. Entonces, el generador $\mathcal{A}$ puede ser conectado con un controlador (supervisor) S en un lazo de realimentación tal como se ilustra a continuación.

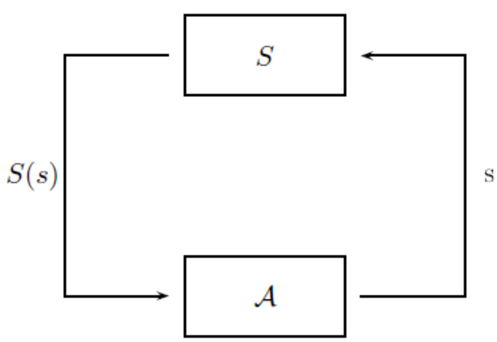

Figura 1: Realimentación dinámica.

Formalmente,

Definición 1. *Sea* $\mathcal{A} = (Q, \Sigma, \delta, \mathcal{E}, q_0, Q_m)$ *un autómata determinístico, y sea* $\Sigma = \Sigma_c \cup \Sigma_{nc}$ *una partición disjunta de* Σ*. Un supervisor* S *para* $\mathcal{A}$ *es cualquier función*

$$S : \mathcal{L}(\mathcal{A}) \to \Gamma := \{\gamma \in 2^{\Sigma} : \Sigma_{nc} \subseteq \gamma\}$$

.

Nosotros ahora acoplaremos las ideas establecidas para control.

Para cada $s \in \mathcal{L}(\mathcal{A})$ bajo el control de S, $S(s) \cap \mathcal{E}(\delta(q_0, s))$ es el conjunto de eventos permitidos que $\mathcal{A}$ puede ejecutar en el estado $\delta(q_0, s)$; es decir, $\mathcal{A}$ no puede ejecutar un evento que este en su conjunto de eventos activos actual si dicho evento no esta también en $S(s)$. Llamaremos a $S(s)$ patrón de control en s. Por definición, el patrón de control $S(s) = \gamma \in \Gamma$ contiene a Σ_{nc}. Esto garantiza que el supervisor S no puede inhabilitar a los eventos no controlables. Por último, notemos que estamos en presencia de una realimentación dinámica en el sentido que el dominio de S es $\mathcal{L}(\mathcal{A})$ y no el conjunto de estados Q. Por lo tanto, el patrón de control puede cambiar en diferentes visitas a un estado. El sistema resultante será denotado $S/\mathcal{A}$ [2, 4].

Definición 2. *Sea* $\mathcal{A}$ *un autómata determinístico con* $\Sigma = \Sigma_c \cup \Sigma_{nc}$ *una partición disjunta de* Σ*. Sea* S *un supervisor para* $\mathcal{A}$*. El lenguaje generado por* $S/\mathcal{A}$ *es dado recursivamente por:*

(*i*) $\theta \in \mathcal{L}(S/\mathcal{A})$;

(*ii*) *Para* $s \in \Sigma^*$ *y* $\alpha \in \Sigma$ *se tiene que:* $s \in \mathcal{L}(S/\mathcal{A})$, $s\alpha \in \mathcal{L}(\mathcal{A})$, $\alpha \in S(s) \iff s\alpha \in \mathcal{L}(S/\mathcal{A})$.

El lenguaje marcado por $S/\mathcal{A}$ *es dado por*

$$\mathcal{L}_m(S/\mathcal{A}) = \mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_m(\mathcal{A}).$$

Desde la definición 2 es claro que $\mathcal{L}(S/\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A})$ es cerrado. Más aún, $\emptyset \subseteq \mathcal{L}_m(S/\mathcal{A}) \subseteq \overline{\mathcal{L}_m(S/\mathcal{A})} \subseteq \mathcal{L}(S/\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A})$. En lo que sigue, con abuso de lenguaje, diremos que un supervisor S para $\mathcal{A}$ es bloqueado si $S/\mathcal{A}$ es bloqueado. Luego, como $\mathcal{L}_m(S/\mathcal{A})$ consiste de los elementos de $\mathcal{L}_m(\mathcal{A})$ que "sobreviven" bajo el control de S, entonces el bloqueo significa que el sistema controlado no puede finalizar la ejecución de la tarea.

4. Planificación

La razón por la que un supervisor S es incluido para un SED $\mathcal{A}$ es el supuesto de que el sistema no controlado $\mathcal{A}$ genera un comportamiento ilegal. Luego, el comportamiento legal o la planificación será dado como un sublenguaje $\mathcal{L}_p \subseteq \mathcal{L}(\mathcal{A})$ o más restrictivamente como un sublenguaje $\mathcal{L}_{p_m} \subseteq \mathcal{L}_m(\mathcal{A})$ (en problemas donde el bloqueo es de interés). Se enfatiza que $\mathcal{L}_p$ ó $\mathcal{L}_{p_m}$ se obtienen después de establecer las especificaciones o requerimientos impuestos sobre el SED. Estas especificaciones son descritas por uno o más lenguajes (posiblemente marcados) $\mathcal{L}_{Ei}$, $i = 1, 2, \ldots, m$. Si los lenguajes $\mathcal{L}_{Ei}$ no son sublenguajes de $\mathcal{L}(\mathcal{A})$ o de $\mathcal{L}_m(\mathcal{A})$ entonces tomamos apropiadamente

$$\mathcal{L}_p = \mathcal{L}(\mathcal{A}) \cap \left(\bigcap_{i=1}^{m} \mathcal{L}_{Ei} \right)$$

ó

$$\mathcal{L}_{p_m} = \mathcal{L}_m(\mathcal{A}) \cap \left(\bigcap_{i=1}^{m} \mathcal{L}_{Ei} \right).$$

Nosotros siempre asumiremos que la planificación es dada; sin embargo, comentamos que expresarla no es tarea fácil. De hecho, no existe una metodología unificada para su construcción

5. Controlabilidad

Estamos interesados en saber bajo que condiciones una planificación $\mathcal{L}_p$ puede ser llevada a cabo por un supervisor S.

Definición 3. *Dado un SED $\mathcal{A}$ con $\Sigma_{nc} \subseteq \Sigma$ su conjunto de eventos no controlables. Sea $K \subseteq \mathcal{L}(\mathcal{A})$, $K \neq \emptyset$*

(i) *K será llamado $\mathcal{L}_m(\mathcal{A})$–cerrado si $K = \overline{K} \cap \mathcal{L}_m(\mathcal{A})$.*

(ii) *K será llamado controlable con respecto a $\mathcal{L}(\mathcal{A})$ y Σ_{nc} si $\overline{K}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{K}$.*

Observación 1. *La condición (ii) de la definición 3 es un concepto fundamental del control supervisorio. De hecho, (ii) significa que un lenguaje puede ser llevado a cabo bajo control si, y solo sí, no hay continuaciones de palabras en el lenguaje por eventos no controlables, físicamente posibles, que estén fuera de dicho lenguaje. Más aún, por definición, K es controlable si, y solo sí, $\overline{K}$ es controlable.*

Teorema 1. *Sea $K \subseteq \mathcal{L}_m(\mathcal{A})$, $K \neq \emptyset$. Sea $\Sigma_{nc} \subseteq \Sigma$ el conjunto de eventos no controlables de $\mathcal{A}$. Existe un supervisor no bloqueado S para $\mathcal{A}$ tal que $K = \mathcal{L}_m(S/\mathcal{A})$ sí, y solo sí, K es $\mathcal{L}_m(\mathcal{A})$–cerrado y controlable con respecto a $\mathcal{L}(\mathcal{A})$ y Σ_{nc}.*

Demostración

Supongamos que existe un supervisor no bloqueado S para $\mathcal{A}$ tal que $K = \mathcal{L}_m(S/\mathcal{A})$. Por definición de $\mathcal{L}(S/\mathcal{A})$, $\mathcal{L}(S/\mathcal{A})\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(S/\mathcal{A})$. Luego, del no bloqueo de S se sigue que

$$\overline{\mathcal{L}_m(S/\mathcal{A})}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{\mathcal{L}_m(S/\mathcal{A})};$$

es decir, $\overline{K}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{K}$. Por otro lado,

$$\begin{aligned} K &= \mathcal{L}_m(S/\mathcal{A}) \\ &= \overline{\mathcal{L}_m(S/\mathcal{A})} \cap \mathcal{L}_m(\mathcal{A}) \\ &= \overline{K} \cap \mathcal{L}_m(\mathcal{A}). \end{aligned}$$

Recíprocamente, supongamos que K es $\mathcal{L}_m(\mathcal{A})$–cerrado y controlable con respecto a $\mathcal{L}(\mathcal{A})$ y Σ_{nc}. Sea S el supervisor para $\mathcal{A}$ definido por $S(s) = \Sigma_{nc} \cup \{\alpha \in \Sigma_c : s\alpha \in \overline{K}\}$.

Afirmación: $\overline{K} = \mathcal{L}(S/\mathcal{A})$. En efecto, por definición de S y controlabilidad de K es claro que $\mathcal{L}(S/\mathcal{A}) \subseteq \overline{K}$. Para la otra dirección procederemos por inducción sobre la longitud de la palabra. Sea $s \in \overline{K}$, con $|s| = 1$; es decir, $s = \alpha$ para algún $\alpha \in \Sigma$, entonces $s \in \mathcal{L}(S/\mathcal{A})$. Dado un lenguaje $\mathcal{L} \subseteq \Sigma^*$ pongamos $\mathcal{L}_i := \{s \in \mathcal{L} : |s| = i\}$, $i = 0, 1, 2, \ldots$ Claramente $\mathcal{L} = \bigcup_{i=0}^{\infty} \mathcal{L}_i$.

Supongamos que $\mathcal{L}_i(S/\mathcal{A}) = \overline{K}_i$, $i = 0, 1, \ldots, n$ (hipótesis de inducción). Sea $s \in \mathcal{L}_n(S/\mathcal{A})$ y consideremos la palabra $s\alpha \in \mathcal{L}(\mathcal{A})$, $\alpha \in \Sigma$. Si $s\alpha \in \overline{K}$ entonces $\alpha \in S(s)$; de donde, $s\alpha \in \mathcal{L}(S/\mathcal{A})$. Por lo tanto, $\mathcal{L}_{n+1}(S/\mathcal{A}) = \overline{K}_{n+1}$. Luego, $\overline{K} \subseteq \mathcal{L}(S/\mathcal{A})$. En consecuencia, $\overline{K} = \mathcal{L}(S/\mathcal{A})$.

Finalmente, desde la afirmación y que K es $\mathcal{L}_m(\mathcal{A})$–cerrado se sigue que

$$\begin{aligned} K &= \overline{K} \cap \mathcal{L}_m(\mathcal{A}) \\ &= \mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_m(\mathcal{A}) \\ &= \mathcal{L}_m(S/\mathcal{A}) \\ & \text{y} \\ \mathcal{L}(S/\mathcal{A}) &= \overline{\mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_m(\mathcal{A})} \\ &= \overline{\mathcal{L}_m(S/\mathcal{A})}. \end{aligned}$$

■

Corolario 1. *Sean $K \subseteq \mathcal{L}(\mathcal{A})$, $K \neq \emptyset$, y sea $\Sigma_{nc} \subseteq \Sigma$ el conjunto de eventos no controlables de $\mathcal{A}$. Existe un supervisor S para $\mathcal{A}$ tal que $K = \mathcal{L}(S/\mathcal{A})$ si, solo sí, K es cerrado y controlable con respecto a $\mathcal{L}(\mathcal{A})$ y Σ_{nc}.*

Demostración

Supongamos que existe un supervisor S para $\mathcal{A}$ tal que $K = \mathcal{L}(S/\mathcal{A})$, entonces obviamente $K = \overline{K}$; es decir, K es cerrado. También, por definición de $\mathcal{L}(S/\mathcal{A})$, $\mathcal{L}(S/\mathcal{A})\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(S/\mathcal{A})$; es decir, $\overline{K}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{K}$. Así, K es controlable con respecto a $\mathcal{L}(\mathcal{A})$ y Σ_{nc}. Recíprocamente, si K es controlable con respecto a $\mathcal{L}(\mathcal{A})$ y Σ_{nc}, y $K = \overline{K}$, entonces tomando el supervisor S definido en la prueba del teorema 1 y procediendo de manera análoga obtenemos que $\overline{K} = \mathcal{L}(S/\mathcal{A}) = K$. ■

Corolario 2. *Sea $K \subseteq \mathcal{L}_m(\mathcal{A})$, $K \neq \emptyset$, y sea $\Sigma_{nc} \subseteq \Sigma$ el conjunto de eventos no controlables del SED $\mathcal{A}$. Existe un supervisor S para $\mathcal{A}$ tal que $\overline{K} = \mathcal{L}(S/\mathcal{A})$ si, solo si, K es controlable con respecto a $\mathcal{L}(\mathcal{A})$ y Σ_{nc}.*

Demostración

Análoga a la prueba del colorario 1 ■

Como ha sido establecido, dados un SED $\mathcal{A}$ y $K \subseteq \mathcal{L}_m(\mathcal{A})$, si existe un supervisor S para $\mathcal{A}$ tal que $\overline{K} = \mathcal{L}(S/\mathcal{A})$ entonces K es un lenguaje controlable con respecto a $\mathcal{L}(\mathcal{A})$ y Σ_{nc} (o simplemente controlable) de $\mathcal{L}(\mathcal{A})$. Adicionalmente (si nuestro interés está centrado en lenguajes marcados), si K es $\mathcal{L}_m(\mathcal{A})$–cerrado, entonces obtenemos que $K = \mathcal{L}_m(S/\mathcal{A})$ donde S es no bloqueado.

Para propósitos de implementación queremos tener una representación de S mediante un autómata determinístico. Esta representación será llamada una realización de S. Por lo tanto, cuando los lenguajes involucrados: $\mathcal{L}(\mathcal{A}), \mathcal{L}_m(\mathcal{A})$ y K; sean regulares entonces la representación será finita y en consecuencia implementable. Así, la realización de S será almacenada en memoria y será suficiente en tiempo real para leer el patrón de control de la traza de eventos actual. Más precisamente, sin pérdida de generalidad supongamos que el dominio de S es $\mathcal{L}(S/\mathcal{A})$.

Sea $\mathcal{R} = (X, \Sigma, \delta_{\mathcal{R}}, \mathcal{E}_{\mathcal{R}}, x_0, X)$ un autómata determinístico limpio tal que $\overline{K} = \mathcal{L}(\mathcal{R}) = \mathcal{L}_m(\mathcal{R})$ (la existencia de $\mathcal{R}$ fue establecida en los preliminares), entonces el autómata producto $\mathcal{A} \times \mathcal{R}$ es tal que

$$\begin{aligned} \mathcal{L}(\mathcal{A} \times \mathcal{R}) &= \mathcal{L}(\mathcal{R}) \cap \mathcal{L}(\mathcal{A}) \\ &= \overline{K} \cap \mathcal{L}(\mathcal{A}) \\ &= \overline{K} = \mathcal{L}(S/\mathcal{A}) \\ & \text{y} \\ \mathcal{L}_m(\mathcal{A} \times \mathcal{R}) &= \overline{K} \cap \mathcal{L}_m(\mathcal{A}) \\ &= \mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_m(\mathcal{A}) \\ &= \mathcal{L}_m(S/\mathcal{A}). \end{aligned}$$

Lo que esto significa es que una vez eliminados de $S(s)$ los eventos no controlables que no están en $\mathcal{E}(\delta(q_0, s))$, el patrón de control $S(s)$ queda inmerso en la estructura de transición de $\mathcal{R}$. Es decir,

$$\begin{aligned} S(s) \cap \mathcal{E}(\delta(q_0, s)) &= \mathcal{E}_{\mathcal{A}\times\mathcal{R}}(\delta_{\mathcal{A}\times\mathcal{R}}((q_0, x_0), s)) \\ &= \mathcal{E}_{\mathcal{R}}(\delta_{\mathcal{R}}(x_0, s)), \end{aligned}$$

donde la última igualdad es consecuencia de que $\overline{K} \subseteq \mathcal{L}(\mathcal{A})$.

Ahora, $\mathcal{A} \times \mathcal{R}$ está definido sin hacer referencia a un mecanismo de control; sin embargo, la interpretación es como sigue: Sea $s \in \mathcal{L}(S/\mathcal{A})$, y sean q y x los estados resultantes de la ejecución de s. Si $\mathcal{A}$ ejecuta un evento $\alpha \in \mathcal{E}(q)$,entonces $\alpha \in \mathcal{E}_{\mathcal{R}}(x)$. Así, $\mathcal{R}$ también ejecuta el evento α (como observador pasivo). Sean q' y x' los estados alcanzados en $\mathcal{A}$ y $\mathcal{R}$ respectivamente después de la ejecución de α, entonces el conjunto de eventos activos de $\mathcal{A}$ en $s\alpha$ es dado ahora por el conjunto de eventos activos de $\mathcal{R}$ en x'.

Finalmente, hemos construido una representación de S que en caso de un lenguaje regular K solo requiere memoria finita.

6. No Controlabilidad

La presente sección es incluida porque posiblemente las planificaciones $\mathcal{L}_p$ y $\mathcal{L}_{p_m}$ no son controlables. Supongamos dado un SED $\mathcal{A}$ junto con su conjunto de eventos no controlables $\Sigma_{nc} \subseteq \Sigma$.

Teorema 2. *Sea $\{K_i\}_{i\in I}$ una familia arbitraria de lenguajes, $K_i \subseteq \mathcal{L}(\mathcal{A})$ para todo $i \in I$.*

(i) *Si los K_i son controlables, entonces $\bigcup_{i\in I} K_i$ es controlable.*

(ii) *Si los K_i son controlables y cerrados, entonces $\bigcap_{i\in I} K_i$ es controlable y cerrado*

Demostración

(i) Si $\overline{K_i}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{K_i}$ para todo $i \in I$, entonces

$$\begin{aligned}\left(\overline{\bigcup_{i\in I} K_i}\right)\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) &= \mathcal{E}_{\mathcal{A}\times\mathcal{R}}(\delta_{\mathcal{A}\times\mathcal{R}}((q_0, x_0), s)) \\ &= \left(\bigcup_{i\in I} \overline{K_i}\right)\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \\ &= \bigcup_{i\in I}(\overline{K_i}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A})) \\ &\subseteq \bigcup_{i\in I} \overline{K_i} \\ &= \overline{\bigcup_{i\in I} K_i}.\end{aligned}$$

Por lo tanto, $\bigcup_{i\in I} K_i$ es controlable.

(ii) Si $K_i = \overline{K_i}$ para todo $i \in I$, entonces $\overline{\bigcap_{i\in I} K_i} = \bigcap_{i\in I} \overline{K_i} = \bigcap_{i\in I} K_i$. Así, $\bigcap_{i\in I} K_i$ es cerrado.

Adicionalmente, si $\overline{K_i}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{K_i}$ para todo $i \in I$, entonces

$$\begin{aligned}\overline{\bigcap_{i\in I} K_i}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) &= \left(\bigcap_{i\in I} \overline{K_i}\right)\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \\ &\subseteq \overline{K_j}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \\ &\subseteq \overline{K_j},\end{aligned}$$

para todo $j \in I$; luego,

$$\overline{\bigcap_{i\in I} K_i}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \bigcap_{i\in I} \overline{K_i} = \overline{\bigcap_{i\in I} K_i}.$$

Por lo tanto, $\bigcap_{i\in I} K_i$ es controlable. ■

Sea $K \subseteq \mathcal{L}(\mathcal{A})$ y consideremos los conjuntos

$$\begin{aligned}C_{\subset}(K) &:= \left\{\mathcal{L} \subseteq K : \overline{\mathcal{L}}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{\mathcal{L}}\right\} \\ &\text{y} \\ C_{\supset}(K) &:= \{\mathcal{L} : K \subseteq \mathcal{L} \subseteq \mathcal{L}(\mathcal{A}), \\ &\qquad \mathcal{L} = \overline{\mathcal{L}}, \overline{\mathcal{L}}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{\mathcal{L}}\}\end{aligned}$$

Claramente $C_{\subset}(K)$ y $C_{\supset}(K)$ son no vacíos ($\emptyset \in C_{\subset}(k)$ y $\mathcal{L}(\mathcal{A}) \in C_{\supset}(K)$). Más aún, $C_{\subset}(K)$ es un conjunto ordenado parcialmente. Luego, por la parte (i) del teorema 2 se tiene que $C_{\subset}(K)$ posee como supremo único a $K_{\sup} = \bigcup_{\mathcal{L}\in C_{\subset}(K)} \mathcal{L}$.

Definición 4. *Sea $K \subseteq \mathcal{L}(\mathcal{A})$. $K_{\sup}$ será llamado el sublenguaje controlable supremo de K.*

Notemos que si K es controlable, entonces $K_{\sup} = K$. Ahora, $K_{\sup}$ no necesariamente es cerrado. Por lo tanto, incluiremos el teorema siguiente.

Teorema 3. (i) *Si $K \subseteq \mathcal{L}(\mathcal{A})$ es cerrado, entonces $K_{\sup}$ es cerrado;*

(ii) *Si $K \subseteq \mathcal{L}_m(\mathcal{A})$ es $\mathcal{L}_m(\mathcal{A})$–cerrado, entonces $K_{\sup}$ es $\mathcal{L}_m(\mathcal{A})$–cerrado.*

(iii) $\overline{K_{\sup}} \subseteq \overline{K}_{\sup}$.

Demostración

(i): Supongamos que $K = \overline{K}$. Entonces, $\mathcal{L} \in C_{\subset}(K)$ sí, y solo sí, $\mathcal{L} \subseteq K$ y $\overline{\mathcal{L}}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{\mathcal{L}}$ sí, y solo sí, $\overline{\mathcal{L}} \subseteq \overline{K} = K$ y $\overline{\overline{\mathcal{L}}}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{\overline{\mathcal{L}}}$ sí, y solo sí, $\overline{\mathcal{L}} \in C_{\subset}(K)$. Luego,

$$\begin{aligned}\overline{K_{\text{sup}}} &= \overline{\bigcup_{\mathcal{L}\in C_{\subset}(K)} \mathcal{L}} \\ &= \bigcup_{\mathcal{L}\in C_{\subset}(K)} \overline{\mathcal{L}} \\ &= \bigcup_{\mathcal{L}\in C_{\subset}(K)} \mathcal{L} \\ &= K_{\text{sup}}.\end{aligned}$$

(*ii*): Supongamos que $K \subseteq \mathcal{L}_m(\mathcal{A})$ es $\mathcal{L}_m(\mathcal{A})$–cerrado. Entonces, $\mathcal{L} \in C_{\subset}(K)$ implica que $\overline{\mathcal{L}} \cap \mathcal{L}_m(\mathcal{A}) \subseteq \overline{K} \cap \mathcal{L}_m(\mathcal{A}) = K$ y $\overline{\mathcal{L}}$ es controlable. Como $\overline{\mathcal{L}} \cap \mathcal{L}_m(\mathcal{A}) \subseteq \mathcal{L}$ entonces

$$\begin{aligned}\overline{K_{\text{sup}}} \cap \mathcal{L}_m(\mathcal{A}) &= \overline{\bigcup_{\mathcal{L}\in C_{\subset}(K)} \mathcal{L}} \\ &\subseteq K_{\text{sup}} \\ &\subseteq \overline{K_{\text{sup}}} \cap \mathcal{L}_m(\mathcal{A}).\end{aligned}$$

(recuerde que $K_{\text{sup}} \subseteq K \subseteq \mathcal{L}_m(\mathcal{A})$). Luego, $K_{\text{sup}} = \overline{K_{\text{sup}}} \cap \mathcal{L}_m(\mathcal{A})$. Así, K_{sup} es $\mathcal{L}_m(\mathcal{A})$–cerrado.

(*iii*): $\mathcal{L} \in C_{\subseteq}(K) \Longrightarrow \mathcal{L} \subseteq K,\ \overline{\mathcal{L}}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{\mathcal{L}} \Longrightarrow \overline{\mathcal{L}} \subseteq \overline{K},\ \overline{\overline{\mathcal{L}}}\Sigma_{nc} \cap \mathcal{L}(\mathcal{A}) \subseteq \overline{\overline{\mathcal{L}}} \Longrightarrow \overline{\mathcal{L}} \in C_{\subset}(\overline{K})$. Luego,

$$\overline{K_{\text{sup}}} = \bigcup_{\mathcal{L}\in C_{\subset}(K)} \overline{\mathcal{L}} \subseteq \bigcup_{P\in C_{\subset(\overline{K})}} P = \overline{K}_{\text{sup}}.$$

■

Ahora, $C_{\supset}(K)$ es un conjunto ordenado parcialmente. Luego, del teorema 2 se sigue que $C_{\supset}(K)$ posee como ínfimo único a $K_{\text{ínf}} = \bigcap_{\mathcal{L}\in C_{\supset}(K)} \mathcal{L}$.

Definición 5. *Sea $K \subseteq \mathcal{L}(\mathcal{A})$. El lenguaje $K_{\text{ínf}}$ será llamado el superlenguaje cerrado y controlable ínfimo de K.*

Notemos que si K es controlable, entonces $K_{\text{ínf}} = K$.

Para finalizar, se pueden verificar sin mayor dificultad, desde las definiciones correspondientes, que $\emptyset \subseteq K_{\text{sup}} \subseteq K \subseteq \overline{K} \subseteq K_{\text{ínf}} \subseteq \mathcal{L}(\mathcal{A})$.

7. Problemas de Control Supervisorio

Con todos los conceptos y propiedades establecidas en las secciones precedentes podemos formular una serie de problemas junto con sus soluciones. Tales problemas constituyen el centro de este artículo y corresponden a necesidades técnicas desde un punto de vista práctico.

Problema Básico de Control Supervisorio:

Dados un SED $\mathcal{A}$, el conjunto de eventos no controlables $\Sigma_{nc} \subseteq \Sigma$ de $\mathcal{A}$ y una planificación $\mathcal{L}_p = \overline{\mathcal{L}_p} \subseteq \mathcal{L}(\mathcal{A})$, construir un supervisor S para $\mathcal{A}$ tal que:

(*i*) $\mathcal{L}(S/\mathcal{A}) \subseteq \mathcal{L}_p$;

(*ii*) Si S' es otro supervisor para $\mathcal{A}$ tal que $\mathcal{L}(S'/\mathcal{A}) \subseteq \mathcal{L}_p$, entonces $\mathcal{L}(S'/\mathcal{A}) \subseteq \mathcal{L}(S/\mathcal{A})$.

Solución:

Como $\mathcal{L}_p$ es cerrado, entonces $(\mathcal{L}_p)_{\text{sup}}$ es cerrado (ver teorema 3, (*i*)). Luego, $(\mathcal{L}_p)_{\text{sup}} \subseteq \mathcal{L}(\mathcal{A})$ es cerrado y controlable. Así, desde el colorario 1 se sigue que existe un supervisor S para $\mathcal{A}$ tal que $\mathcal{L}(S/\mathcal{A}) = (\mathcal{L}_p)_{\text{sup}} \subseteq \mathcal{L}_p$. Más aún, si S' es tal que $\mathcal{L}(S'/\mathcal{A}) \subseteq \mathcal{L}_p$ entonces $\mathcal{L}(S'/\mathcal{A}) \in C_{\subset}(\mathcal{L}_p)$; pero $(\mathcal{L}_p)_{\text{sup}}$ es el más grande de estos. Por lo tanto, $\mathcal{L}(S'/\mathcal{A}) \subseteq \mathcal{L}(S/\mathcal{A})$. En consecuencia, para la solución del problema básico de control supervisorio, basta tomar S tal que $\mathcal{L}(S/\mathcal{A}) = (\mathcal{L}_p)_{\text{sup}}$ siempre que $(\mathcal{L}_p)_{\text{sup}} \neq \emptyset$. Finalmente, si $\mathcal{L}(S/\mathcal{A}) = (\mathcal{L}_p)_{\text{sup}}$ es un lenguaje regular entonces podemos construir una realización del supervisor S, la cual es un AFD que representa a $(\mathcal{L}_p)_{\text{sup}}$.

Problema de Control Supervisorio sin Bloqueo:

Dados un SED $\mathcal{A}$, $\Sigma_{nc} \subseteq \Sigma$ y un lenguaje marcado $\mathcal{L}_{p_m} \subseteq \mathcal{L}_m(\mathcal{A})$ $\mathcal{L}_m(\mathcal{A})$–cerrado, construir un supervisor no bloqueado S tal que:

(*i*) $\mathcal{L}_m(S/\mathcal{A}) \subseteq \mathcal{L}_{p_m}$;

(*ii*) Para cualquier otro supervisor no bloqueado S' tal que $\mathcal{L}_m(S'/\mathcal{A}) \subseteq \mathcal{L}_{p_m}$ se tiene que $\mathcal{L}(S'/\mathcal{A}) \subseteq \mathcal{L}(S/\mathcal{A})$.

Solución:

Como en el problema anterior, usando los resultados de las secciones precedentes la solución es seleccionar S tal que $\mathcal{L}(S/\mathcal{A}) = \overline{(\mathcal{L}_{p_m})_{\text{sup}}}$, siempre que $(\mathcal{L}_{p_m})_{\text{sup}} \neq \emptyset$. Es importante notar que bajo el supuesto que $\mathcal{L}_{p_m}$ es $\mathcal{L}_m(\mathcal{A})$–cerrado, $(\mathcal{L}_{p_m})_{\text{sup}}$ es igualmente $\mathcal{L}_m(\mathcal{A})$–cerrado. Esto garantiza que la elección de S resulta en un sistema $S/\mathcal{A}$ no bloqueado (ver teorema 1). Ahora, si $(\mathcal{L}_{p_m})_{\text{sup}}$ es regular entonces S puede ser realizado por construcción de una representación AFD de $\overline{(\mathcal{L}_{p_m})_{\text{sup}}}$.

Problema de Control Supervisorio Dual:

Dados un SED $\mathcal{A}$, $\Sigma_{nc} \subseteq \Sigma$ y un lenguaje requerido mínimo $\mathcal{L}_{\text{mín}} \subseteq \mathcal{L}(\mathcal{A})$, construir un supervisor S tal que:

(i) $\mathcal{L}_{\text{mín}} \subseteq \mathcal{L}(S/\mathcal{A})$;

(ii) Para cualquier otro supervisor S' tal que $\mathcal{L}_{\text{mín}} \subseteq \mathcal{L}(S'/\mathcal{A})$ se tiene que $\mathcal{L}(S/\mathcal{A}) \subseteq \mathcal{L}(S'/\mathcal{A})$.

Solución: Por el teorema 3, (ii), y la definición de $C_{\supset}(\mathcal{L}_{\text{mín}})$ tenemos que $(\mathcal{L}_{\text{mín}})_{\text{ínf}}$ es controlable y cerrado. Así, desde el colorario 1 se sigue que existe un supervisor S para $\mathcal{A}$ tal que $\mathcal{L}(S/\mathcal{A}) = (\mathcal{L}_{\text{mín}})_{\text{ínf}} \supseteq \mathcal{L}_{\text{mín}}$. Por otro lado, si S' es otro supervisor para $\mathcal{A}$ tal que $\mathcal{L}_{\text{mín}} \subseteq \mathcal{L}(S'/\mathcal{A})$ entonces $\mathcal{L}(S'/\mathcal{A}) \in C_{\supset}(\mathcal{L}_{\text{mín}})$; pero $(\mathcal{L}_{\text{mín}})_{\text{ínf}}$ es el más pequeño de éstos, de donde $\mathcal{L}(S/\mathcal{A}) = (\mathcal{L}_{\text{mín}})_{\text{ínf}} \subseteq \mathcal{L}(S'/\mathcal{A})$. Luego, los requerimientos (i) y (ii) son satisfechos por S. Por lo tanto, para la solución óptima del problema de control supervisorio dual basta tomar S tal que $\mathcal{L}(S/\mathcal{A}) = (\mathcal{L}_{\text{mín}})_{\text{ínf}}$. Notemos que sí $\mathcal{L}_{\text{mín}} \subseteq \mathcal{L}_m(\mathcal{A})$, entonces

$$\begin{aligned}\mathcal{L}_{\text{mín}} \subseteq \overline{\mathcal{L}_{\text{mín}}} \cap \mathcal{L}_m(\mathcal{A}) &\subseteq (\mathcal{L}_{\text{mín}})_{\text{ínf}} \cap \mathcal{L}_m(\mathcal{A})) \\ &= \mathcal{L}_m(S/\mathcal{A}).\end{aligned}$$

Sin embargo, nada asegura que S es no bloqueado. De hecho, la versión sin bloqueo del problema de control supervisorio dual posee dificultades técnicas puesto que la propiedad de controlabilidad no se preserva bajo intersección.

Problema de Control Supervisorio con Tolerancia:

Dados un SED $\mathcal{A}$,$\Sigma_{nc} \subseteq \Sigma$, un lenguaje marcado deseado $\mathcal{L}_d \subseteq \mathcal{L}_m(\mathcal{A})$ y una planificación de tolerancia $\mathcal{L}_t = \overline{\mathcal{L}_t} \subseteq \mathcal{L}(\mathcal{A})$, donde $\overline{\mathcal{L}_d} \subseteq \mathcal{L}_t$; construir un supervisor S para $\mathcal{A}$ tal que:

(i) $\mathcal{L}(S/\mathcal{A}) \subseteq \mathcal{L}_t$;

(ii) Para todo lenguaje cerrado y controlable $K \subseteq \mathcal{L}_t$, $K \cap \mathcal{L}_d \subseteq \mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_d$;

(iii) Para todo lenguaje cerrado y controlable $K \subseteq \mathcal{L}_t$ se tiene que $K \cap \mathcal{L}_d = \mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_d$ implica que $\mathcal{L}(S/\mathcal{A}) \subseteq K$.

Solución:

Sea S tal que $\mathcal{L}(S/\mathcal{A}) = ((\mathcal{L}_t)_{\text{sup}} \cap \mathcal{L}_d)_{inf}$. (La existencia de S está garantizada desde los resultados de las secciones previas).

Ahora, como $(\mathcal{L}_t)_{sup}$ es cerrado y controlable, y $(\mathcal{L}_t)_{sup} \cap \mathcal{L}_d \subseteq (\mathcal{L}_t)_{sup}$ entonces $((\mathcal{L}_t)_{sup} \cap \mathcal{L}_d)_{inf}. \subseteq (\mathcal{L}_t)_{sup} \subseteq \mathcal{L}_t$. Por otro lado, si $K \subseteq \mathcal{L}_t$ es cerrado y controlable entonces $K \subseteq (\mathcal{L}_t)_{\text{sup}}$; de donde $K \cap \mathcal{L}_d \subseteq (\mathcal{L}_t)_{\text{sup}} \cap \mathcal{L}_d$; luego, $K \cap \mathcal{L}_d \subseteq \mathcal{L}(S/\mathcal{A})$. En consecuencia, $K \cap \mathcal{L}_d \subseteq \mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_d$. Finalmente, en adición a las hipótesis establecidas para K, si $K \cap \mathcal{L}_d = \mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_d$ entonces $(\mathcal{L}_t)_{\text{sup}} \cap \mathcal{L}_d \subseteq \mathcal{L}(S/\mathcal{A})$ implica que $(\mathcal{L}_t)_{\text{sup}} \cap \mathcal{L}_d \subseteq \mathcal{L}(S/\mathcal{A}) \cap \mathcal{L}_d = K \cap \mathcal{L}_d \subseteq K$. Por lo tanto, $\mathcal{L}(S/\mathcal{A}) \subseteq K$. Luego, S es solución del problema de control supervisorio con tolerancia. Para finalizar, el supervisor S no necesariamente es no bloqueado. De hecho, la versión sin bloqueo de este problema no posee en general una solución óptima y tiene muchas dificultades técnicas.

8. Conclusiones

Las soluciones de los problemas establecidos en la sección anterior motivan el estudio de las operaciones sup e ínf. El resultado más importante, sin ninguna duda, es que tanto sup como ínf preservan regularidad en el sentido que si $\mathcal{A}$ es un AFD y K es regular, entonces K_{sup} y $K_{\text{ínf}}$ también son regulares. Esta situación es crucial desde un punto de vista de síntesis ya que esto significa que si un SED posee un número finito de estados y el lenguaje de planificación marcado (o lenguaje de planificación) es regular, entonces los supervisores que dan solución a los diferentes problemas de control supervisorio, dados en la sección anterior, pueden ser realizados por AFD, y éstos pueden ser resueltos automáticamente. Justamente, uno de los principales objetivos de la teoría de control supervisorio es desarrollar métodos formales para la síntesis de leyes de control, lo cual garantiza el cumplimiento de un conjunto dado de especificaciones o requerimientos para la planificación. En este sentido, la teoría de control supervisorio es muy útil y permite el manejo de sistemas complejos.

Referencias

[1] Eilemberg S, 1974, Automata, Languages and Machines, Vol. A, Academic Press, New York.

[2] Ramadge P, Wonham W, 1982, Supervisory Control of Discrete Event Processes, Lecture Notes in Control and Inform,v.39, pp.202-214.

[3] Ramadge P, Wonham W, 1984, Supervisory Control of a Class of Discrete Event Processes, Lecture Notes in Control and Inform,v.63, p.477-498.

[4] Ramadge P, Wonham W, 1986, Modular Supervisory Control of Discrete Event Systems,Lecture Notes in Control and Inform,v.83, p.202-214.