xx (101599,1)

CINVESTAV

Centro de Investigación y de Estudios Avanzados del IPN
Unidad Guadalajara

Control Supervisor para Sistemas de Eventos Discretos con Especificaciones de Seguridad y Vivacidad

Tesis que presenta
Ana Gabriela Solorzano García

Para obtener el grado de
Maestro en Ciencias

En la especialidad de
Ingeniería Eléctrica

Guadalajara, Jalisco. Enero 2002

Control Supervisor para Sistemas de Eventos Discretos con Especificaciones de Seguridad y Vivacidad

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:

Ana Gabriela Solorzano García

Lic. en Informática
Universidad de Guadalajara 1993-1997

Becaria del CONACYT, expediente no. 129212

Directores de Tesis
Dr. Arturo del Sagrado Corazón Sánchez Carmona
Dr. Raúl Ernesto González Torres

CINVESTAV del IPN Unidad Guadalajara, Enero 2002

Índice General

Índice de Figuras iv

Índice de Tablas vi

Capítulo 1. Introducción 1

1.1. Objetivos de la Tesis 7

1.2. Contenido de la Tesis 7

Capítulo 2. Preliminares 11

2.1. Preliminares sobre Lenguajes 11

2.2. Autómatas Finitos 18

2.3. Cálculo de Puntos Fijos 20

Capítulo 3. Sistemas de Eventos Discretos y Supervisión 24

3.1. Sistemas de Eventos Discretos (SED) 24

3.2. Supervisores 24

3.3. El Comportamiento del Sistema bajo Supervisión y Controlabilidad 26

3.4. Síntesis del Supervisor 33

3.5. Control Supervisor Modular 41

3.6. Método de Síntesis 48

Capítulo 4. Cálculo del Conjunto Controlable $C^{\mathcal{A}}$ 56

4.1. Conjunto Controlable del Autómata de Rabin. 56

4.2. Propiedades de Vivacidad en el Modelo del Sistema 72

Capítulo 5. Ejemplos 79

5.1. Máquina de Refrescos 79

5.2. Máquinas y Almacenes 89

Capítulo 6. Conclusiones 96

6.1. Ventajas 96
6.2. Desventajas 97
6.3. Trabajos Futuros 97

Bibliografía 99

Índice de Figuras

1.0.1 SED $G = (L, S)$. El estado inicial es i, $\Sigma_u = \{b\}$ y los estados de aceptación son i y t. 5

3.6.1 Método de síntesis de supervisores 49

5.1.1 Modelo del sistema de la máquina de refrescos $MR = (\Sigma, X, \delta, x_0, R)$, con $L = pre(S)$, estado inicial x_0, $\Sigma_c = \{usar, reparar\}$ y conjunto de aceptación $R = \{x_0, x_1, x_2\}$. 80

5.1.2 Autómata de Rabin $(\Sigma, X, \delta^{MR}, x_0, \{(R_p, I_p)\})$ que reconoce el lenguaje ω-controlable supremo del ejemplo máquina de refrescos. 83

5.1.3 Comportamiento del sistema máquina de refrescos generado bajo la supervisión de f, tomando $A = \emptyset$. 84

5.1.4 Autómata de Büchi $MRa = (\Sigma, X_a, \delta_a, a_0, R_a)$, con $X_a = \{a_0, a_1, a_2\}$, $\Sigma_c = \{usar, reparar\}$ y conjunto de aceptación $R_a = \{a_0\}$. 84

5.1.5 Autómata de Streett $MR_{diff} = (\Sigma, X''', \delta''', \{(R_p, I_p) : p \in P\})$, con pares de aceptación $(\{a_0x_0x_0,\ a_1x_1x_1,\ a_2x_2x_2\}, \emptyset)$ y $(\emptyset, \{a_1x_1x_1,\ a_2x_2x_2\})$. 85

5.1.6 $_*$-autómata $(\Sigma, X, \delta^{MR}, x_0, X)$ que reconoce el lenguaje $pre(supC^{\omega}(E))$ del ejemplo de la máquina de refrescos. 86

5.1.7 Comportamiento del sistema máquina de refrescos generado bajo la supervisión de f, tomando como A el ω-lenguaje del autómata de la figura 5.1.4. 87

5.1.8 $_*$-autómata $(\Sigma, X_a, \delta_a, a_0, X_a)$ de la especificación de seguridad. 87

5.1.9 Supervisor $(\Sigma, X_s, \delta_s, s_0, X_s)$ obtenido utilizando la teoría de control de Radmage y Wonham [7]. 88

5.2.1 Operación del sistema máquinas y almacenes. 89

5.2.2 Comportamiento individual de cada máquina M_i y almacén A_i, para $i = 1, 2$. Los estados iniciales son D y V, los estados de aceptación son D, O y V, L, respectivamente. 90

5.2.3 Autómata de Rabin-Büchi $MA = (\Sigma, X, \delta, DDVV, X)$. 91

5.2.4 Comportamiento del sistema máquinas y almacenes bajo la supervisión de f. 95

Índice de Tablas

1 Cálculo del conjunto controlable C^{MA} y la función Φ^{MA} del ejemplo máquinas y almacenes. 93

CAPÍTULO 1

Introducción

Un sistema de eventos discretos (SED) es un sistema dinámico que está compuesto por estados enumerables y diferenciables entre sí. En este trabajo asumimos que los cambios entre estados ocurren de manera asíncrona en respuesta a la ocurrencia de ciertos eventos. La teoría de control supervisor para sistemas de eventos discretos fue iniciada por Ramadge y Wonham [7] y ha sido aplicada a diferentes sistemas, entre ellos, a sistemas de manufactura [16]. Esta teoría de control de SEDs está basada en lenguajes regulares y autómatas finitos, donde un SED es visto como un generador de secuencias finitas de eventos[1] al cual bajo ciertas condiciones se le puede agregar un mecanismo o estructura de control. En esta teoría se considera que el problema de control tiene solución si es posible construir un controlador supervisor *mínimamente restrictivo*[2] que force al sistema controlado a generar algún comportamiento deseado inhabilitando eventos controlables en ciertos estados. Este comportamiento deseado es proporcionado por un lenguaje de especificación[3]. De esta manera, tanto el sistema[4] como la especificación se representan mediante autómatas.

En estos sistemas es común utilizar varios tipos de especificaciones. Alpern y Schneider [1] proponen una clasificación con dos grandes categorías: "seguridad" y "vivacidad". Otras clasificaciones se pueden encontrar en la literatura (e.g. Manna y Pnueli [6]). En este trabajo, se adopta la clasificación de Alpern y Schneider que es la usada también en la teoría de control supervisor [13].

[1]Al conjunto de secuencia finitas de eventos también le llamamos $*$-lenguaje.

[2]En el sentido de que el lenguaje controlado resultante es el comportamiento más grande posible que satisface la especificación.

[3]De ahora en adelante, utilizaremos como sinónimos los términos: comportamiento deseado y lenguaje de especificación.

[4]En la terminología tradicional de control al sistema también se le llama "planta"

Desafortunadamente, las propiedades de vivacidad no pueden ser expresadas utilizando autómatas de secuencias finitas, como se muestra a continuación. Definamos una ejecución de un sistema como una secuencia infinita de estados $\alpha = s_0 s_1 \ldots$. Sea $\alpha = s_0 s_1 \ldots$ una ejecución y sea i un entero no negativo. Entonces, $\alpha_i = s_0 s_1 \ldots s_i$ es una ejecución parcial que contiene los primeros $i + 1$ estados de α. Una propiedad P es un conjunto de ejecuciones.

Sean X un conjunto de estados, X^ω un conjunto infinito de secuencias de estados y X^* un conjunto finito de ejecuciones parciales. Escribimos $\alpha \models P$ si la ejecución α está en la propiedad P.

DEFINICIÓN 1.0.1. Una propiedad P es una propiedad de seguridad si y sólo si para toda ejecución $\alpha \in X^\omega$, si $\alpha \models \neg P$, entonces existe un número natural $i \geq 0$ tal que, para toda ejecución $\beta \in X^\omega$, $\alpha_i \beta \models \neg P$.

De acuerdo a la definición, una propiedad de seguridad estipula que nada "malo ocurrirá" durante la ejecución, y si así fuera, es posible identificar el punto en el cual sucederá. Algunos ejemplos son exclusión mutua, no terminación y primero en solicitar, primero en ser servido.

DEFINICIÓN 1.0.2. Una propiedad P es una propiedad de vivacidad si y sólo si para toda ejecución $\alpha \in X^*$, existe otra ejecución $\beta \in X^\omega$, tal que $\alpha\beta \models P$.

De acuerdo a la definición, esta propiedad estipula que "algo bueno" ocurrirá por lo menos una vez sin importar en qué instante sucederá. Algunos ejemplos son terminación y garantía de servicio.

Alpern y Schneider [1] demostraron que toda propiedad P es la intersección de una propiedad de seguridad y de una propiedad de vivacidad. También encontraron que las clases de seguridad y vivacidad no son disjuntas, es decir, existen propiedades que pertenecen a ambas clases. Consideremos, por ejemplo, el siguiente caso en donde $S = \{b\}$ y $S^\omega = b^\omega$. Entonces, $P = b^\omega$ es una propiedad de seguridad y también una propiedad de vivacidad.

De acuerdo a las definiciones anteriores, podemos representar una propiedad de seguridad estableciendo un estado de la ejecución del sistema (y por lo tanto una secuencia finita) en donde dicha propiedad no se cumple. En cambio una propiedad de vivacidad no hace referencia a algún estado específico de la ejecución de un sistema. Esto trae como consecuencia que las propiedades de vivacidad no puedan ser expresadas en términos de los $_*$-lenguajes. En este sentido, los ω-lenguajes [5] ofrecen mayor poder de expresión que los $_*$-lenguajes. Las propiedades de vivacidad son útiles no sólo en la especificación sino también en el modelado de sistemas. Por ejemplo, en el modelado de propiedades de justicia en sistemas no controlados [4].

Por lo descrito arriba, la teoría de Ramadge y Wonham para lenguajes finitos incorpora de manera natural especificaciones de seguridad. De igual forma, el tratamiento de especificaciones de vivacidad no se puede realizar de manera directa. Es común utilizar especificaciones de seguridad "fuertes" que como consecuencia hacen que el sistema cumpla con las especificaciones de vivacidad de interés. Esto implica un conocimiento profundo de la dinámica del sistema y un trabajo exhaustivo en la labor de especificación.

1.0.1. Extensiones de la Teoría de Control Supervisor a secuencias infinitas. Con la finalidad de incorporar de manera consistente aquellos sistemas con comportamiento infinito, así como especificaciones de seguridad y vivacidad, se han propuesto varias extensiones a la teoría de Control Supervisor de Ramadge y Wonham. A continuación hacemos una revisión en general de estas propuestas.

Ramadge [8] propone una extensión a la teoría de control supervisor para sistemas de eventos discretos, en el marco de ω-lenguajes. Estos lenguajes modelan el comportamiento de sistemas que son diseñados para operar infinitamente. Específicamente Ramadge, modela el sistema como un binomio $G = (L, S)$, donde L representa todo el comportamiento finito posible de eventos que G puede generar y S es el límite de L o un subconjunto de él. Esta representación permite asociar un mecanismo de control a cada secuencia infinita de S a través de L. En este modelo, el comportamiento generado bajo la supervisión se caracteriza por dos propiedades: $_*$-controlabilidad y ω-cerradura. La existencia de supervisores mínimamente restrictivos se deriva de la siguiente proposición.

[5]Los ω-lenguajes, que se definirán formalmente en el capítulo 2, consisten de secuencias infinitas de eventos.

PROPOSICIÓN 1.0.3. [8] *Para todo* $G = (L, S)$ *y todo* $T \subseteq S$ *tal que* $T \neq \emptyset$, *existe un controlador supervisor no bloqueante*[6] *y completo* f *para* G *tal que* $S^f = T$ *si y sólo si*

1. T *es* $_*$*-controlable con respecto a* L *y*
2. T *es* ω*-cerrado con respecto a* S.[7]

Si un ω-lenguaje $T \subseteq S$ satisface las dos condiciones de la proposición anterior, se le llama *comportamiento secuencial controlable*. La existencia de este comportamiento secuencial controlable se deriva de la siguiente proposición.

PROPOSICIÓN 1.0.4. [8] *Si el lenguaje de especificación* $R \subseteq S$ *es* ω*-cerrado con respecto a* S, *entonces existe un único* ω*-lenguaje controlable supremo contenido en* R.

Una característica importante de la proposición 1.0.4 es que podemos garantizar la existencia de supervisores mínimamente restrictivos siempre que el lenguaje de especificación sea ω-cerrado con respecto a S. Pero generalmente, las propiedades de vivacidad no pueden ser expresadas por los lenguajes ω-cerrados como lo demuestra el siguiente ejemplo:

EJEMPLO 1.0.5. Considere el SED $G = (L, S)$ de la figura 1.0.1 [8]. El estado inicial es i, $\Sigma = \{a, b\}$, $S = a^\omega + a^*ba^\omega$, $L = a^* + a^*ba^*$ y $\Sigma_u = \{b\}$. Deseamos derivar un supervisor que genere el comportamiento $R = a^*ba^\omega$. Como el lenguaje de especificación R no es ω-cerrado con respecto a S, la proposición 1.0.4 no garantiza la existencia de un supervisor.

Debido a esta limitante, Young, Spanjol y col. [17] proponen una extensión a la teoría de Ramadge en la que imponen condiciones menos restrictivas al lenguaje de

[6]En el sentido de que el sistema nunca permanecerá bloqueado por generar una secuencia en S. Ramadge utiliza el término "no bloqueante" mientras que Thistle y Wonham [11, 13] utilizan el término "deadlock-free" Lo anterior para diferenciarlo del controlador no bloqueante en la teoría original de Ramadge y Wonham donde no bloqueo significa que el sistema no se bloquea por generar una secuencia en L.

[7]Véase la definición de ω-cerradura más adelante en el capítulo 2.

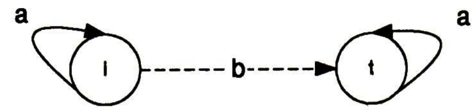


FIGURA 1.0.1. SED $G = (L, S)$. El estado inicial es i, $\Sigma_u = \{b\}$ y los estados de aceptación son i y t.

especificación para la existencia de supervisores tanto deterministas como no deterministas. Sin embargo, a diferencia de [8], en esta extensión no se busca encontrar el comportamiento más grande posible que satisface la especificación. Además se pueden modelar sólo cierta clase de sistemas, ya que se asume que $S = lim(L)$. Bajo estas nuevas condiciones, no es necesario que el lenguaje de especificación sea ω-cerrado con respecto al lenguaje que describe el comportamiento sistema. Por lo tanto, en esta extensión se pueden especificar un número mayor de propiedades de los sistemas, entre ellas, las propiedades de vivacidad.

En este modelo, se considera que el problema de control tiene solución si el lenguaje de especificación es *finitamente estabilizable* [17] y $_*$-controlable con respecto al comportamiento finito del sistema. Bajo estas condiciones, podemos derivar un supervisor que genere solamente secuencias que pertenecen a la ω-cerradura del lenguaje de especificación. Por lo tanto, si el lenguaje de especificación no es $\omega -$cerrado con respecto al lenguaje que describe el comportamiento sistema, entonces el comportamiento generado bajo la supervisión no es exactamente igual al comportamiento especificado, pero es una aproximación. Este lenguaje aproximado se calcula en base a un margen de error dado.

Thistle y Wonham [11, 13] también extienden los resultados de [8] para el caso en el cual el lenguaje de especificación no necesita ser ω-cerrado con respecto al comportamiento del sistema. Definen una nueva propiedad de ω-controlabilidad apropiada a lenguajes que no son ω-cerrados. En este modelo, el comportamiento generado bajo la supervisión se caracteriza por dos propiedades: ω-controlabilidad y ω-cerradura. Determinan también las condiciones suficientes y necesarias para la existencia de supervisores mínimamente restrictivos si el lenguaje de especificación es ω-cerrado con respecto a

S. En caso de que el lenguaje de especificación no sea ω-cerrado con respecto a S, es posible encontrar una solución óptima.

Posteriormente, Barbeau, Kabanza y col. [**5**], proponen un método de síntesis de controladores para especificaciones de seguridad, vivacidad y restricciones de tiempo real[8]. El método de síntesis es aplicable solamente a sistemas en donde $S = lim(L)$. El sistema se representa como un *grafo de transiciones con restricción de tiempo*[9] (GTT) y la especificación es expresada con fórmulas de *lógica temporal métrica*[10] (LTM). Con este tipo de representación, se explotan dos aspectos importantes:

1. La facilidad con la que se puede expresar la dinámica del sistema en términos de lenguajes y autómatas.
2. El poder expresivo y la concisión que tienen las fórmulas de lógica temporal para expresar propiedades de los sistemas.

En este método el controlador se obtiene de manera incremental realizando de manera simultánea la traducción de la fórmula de LTM a un autómata de Büchi y la intersección con el comportamiento controlable del sistema.

Existen varios aspectos importantes respecto a este método de síntesis:

1. La clasificación de propiedades de seguridad y vivacidad es importante en el cálculo del controlador, debido a que una propiedad de seguridad no se satisface en una trayectoria del sistema si nos conduce a algún estado "malo" del sistema, en cambio una propiedad de vivacidad no se satisface en una trayectoria del sistema si nos conduce a un ciclo "malo"
2. Debido a la característica que tienen las propiedades de vivacidad con respecto a la ω-cerradura de lenguajes, si se especifica una propiedad de vivacidad no es posible derivar un controlador que haga que el sistema controlado genere el comportamiento más grande posible que satisface la especificación. Tampoco se especifica ningún procedimiento para derivar un controlador que genere algún

[8]Los ω-lenguajes también tiene la característica de que pueden representar la noción o comportamiento del tiempo.

[9]Las transiciones representan acciones que tienen asociadas una duración de tiempo.

[10]La lógica temporal métrica es una extensión de la lógica temporal lineal que permite asociar una duración a todo operador temporal.

comportamiento cercano al comportamiento más grande posible que satisface la especificación.

Hemos visto algunas extensiones de la teoría de control supervisor de [7] que permiten modelar el comportamiento infinito de los sistemas. Bajo este modelo de control se puede incorporar el uso de especificaciones de vivacidad y tiempo real en los problemas de control. Podemos decir que la teoría propuesta por Thistle y Wonham [13] es más general y puede ser aplicada a una variedad más amplia de sistemas.

1.1. Objetivos de la Tesis

Generalmente, en una especificación completa además de las propiedades de seguridad, también se requiere el uso de propiedades de vivacidad. Como ya hemos visto, estas propiedades sólo pueden ser expresadas utilizando secuencias infinitas de eventos y por lo tanto para sintetizar un controlador supervisor se requiere el uso de una teoría de control en el formalismo de los ω-lenguajes. Por tal motivo, los objetivos principales de esta tesis son:

- Revisar y justificar a detalle la teoría de control supervisor de Thistle y Wonham [13]. Como mencionamos anteriormente, en esta teoría se pueden especificar propiedades de vivacidad y seguridad.
- Proponer un método de síntesis de supervisores basado en la teoría del punto anterior. Este método incluye la captura de especificaciones de seguridad y vivacidad, así como el cálculo del controlador supervisor.
- Mostrar el uso del método de síntesis propuesto en algunos ejemplos.

1.2. Contenido de la Tesis

El resto de la tesis está organizada como sigue. En el capítulo 2 presentamos las bases formales en las que está basada la teoría de control supervisor. En las secciones 2.1 y 2.2 proporcionamos una breve revisión general de la teoría de autómatas y lenguajes formales que son la estructura formal de la teoría de control supervisor para sistemas de eventos discretos. Esta revisión comprende los $_*$- y ω-lenguajes. La aportación que hicimos en la sección 2.1 fue la justificación a detalle de algunas propiedades del *límite* y *la cerradura de un lenguaje*, así como de *el producto* y *el cociente de lenguajes*. Estas propiedades de los lenguajes las utilizaremos más adelante en el capítulo 3 para definir la teoría de control. Finalmente, en la sección 2.3 damos una breve introducción a la teoría de cálculo de puntos fijos que utilizaremos para calcular el lenguaje controlable supremo de un lenguaje.

En el capítulo 3 presentamos la teoría de control supervisor para el comportamiento infinito de los sistemas de eventos discretos de Thistle y Wonham [**13**]. Específicamente, en las secciones 3.1 y 3.2, definimos formalmente el concepto de un SED [**8**] y le agregamos un mecanismo de control al cual denominamos supervisor. En la sección 3.3 definimos el comportamiento generado por el sistema bajo la supervisión, así como las condiciones suficientes y necesarias para la existencia de supervisores. Presentamos de nuevo la proposición 1.0.3 como punto de partida de la teoría de Thistle y Wonham. Como ya hemos dicho, en la teoría de Ramadge [**8**], la existencia de supervisores esta basada en la controlabilidad de lenguajes finitos y no admite especificaciones de vivacidad, ya que el lenguaje de especificación tiene que ser ω-cerrado. Para eliminar esa restricción, Thistle y Wonham extienden la noción de controlabilidad a lenguajes infinitos. En la sección 3.4 demostramos que bajo ciertas condiciones siempre existe un único lenguaje controlable supremo y definimos el supervisor. La aportación que hicimos en esta sección fue demostrar que la cerradura y controlabilidad de lenguajes infinitos tienen diferentes propiedades bajo la unión e intersección de conjuntos. Estas propiedades son importantes ya que a partir de ellas se definen los supervisores mínimamente restrictivos, tal como lo definen Thistle y Wonham. El desarrollo de las demostraciones presentadas en las secciones 3.2, 3.3 y 3.4 fueron tomadas de [**13, 11**]. Algunos pasos de estas demostraciones fueron extendidos con la finalidad de clarificar las demostraciones. En la sección 3.5 extendemos los resultados de la sección 3.2 para incorporar el uso de control supervisor modular. Por último, en la sección 3.6 presentamos el método de síntesis de supervisores propuesto. Este método comprende desde el modelado del sistema y la especificación, hasta la obtención del supervisor.

Un aspecto importante en la síntesis de estos supervisores es el cálculo del ω-lenguaje controlable supremo. Asumiendo que el lenguaje de especificación es modelado con un autómata de Rabin, calcular el ω-lenguaje controlable supremo es equivalente a calcular el conjunto de estados controlables del autómata de Rabin. En el capítulo 4 presentamos el algoritmo de cálculo. Este algoritmo está definido utilizando la teoría de puntos fijos del capítulo 2. En la sección 4.1 presentamos la teoría y procedimiento de cálculo para obtener el conjunto de estados controlables del autómata siempre que el comportamiento del sistema sea ω-cerrado. En la subsección 4.1.1 definimos los operadores de dinámica inversa y alcanzabilidad de un par. En la subsección 4.1.2 definimos el algoritmo para calcular el conjunto de estados controlables, tomado de **[12]**, utilizando los operadores de la sección 4.1.1. En la subsección 4.1.4 redefinimos el algoritmo de la sección 4.1.2 con el fin de reducir su complejidad. Es importante mencionar que el procedimiento de cálculo del conjunto de estados controlables es básico en la síntesis de supervisores, ya que a partir de este procedimiento se define una parte del supervisor. Las aportaciones que hicimos fueron la justificación de las proposiciones del operador de dinámica inversa y del conjunto de estados controlable del autómata, así como la reducción de la complejidad del algoritmo en la subsección 4.1.2. Finalmente, en la sección 4.2 presentamos una extensión de la teoría de la sección 4.1, primeramente propuesta por **[11]**, que incluye el caso en el cual el lenguaje que describe el comportamiento del sistema no es ω-cerrado.

En el capítulo 5 aplicamos a dos ejemplos el método de síntesis de controladores que definimos en el capítulo 3. El primer ejemplo es un sistema "pequeño" que consiste de una máquina de refrescos. En este ejemplo ilustramos los pasos del método de síntesis, desde como modelar el sistema y la especificación, hasta la obtención del supervisor. También, definimos la manera de especificar propiedades de vivacidad y mostramos el uso de un segundo lenguaje de especificación para obtener una solución más aproximada al ω-lenguaje controlable supremo. Posteriormente aplicamos la teoría de Ramadge y Wonham de lenguajes finitos a este mismo ejemplo para obtener un supervisor. Luego comparamos ambos supervisores y el comportamiento generado bajo la supervisión, así como las ventajas y desventajas de ambos. El segundo ejemplo es un sistema de manufactura propuesto en **[11]**, que consiste de dos máquinas y dos almacenes. En este caso mostramos como especificar las propiedades de seguridad.

Para finalizar, en el capítulo 6 enumeramos las ventajas y desventajas del método de síntesis que presentamos en el capítulo 3, así como algunas sugerencias para trabajos futuros.

CAPÍTULO 2

Preliminares

2.1. Preliminares sobre Lenguajes

En esta sección establecemos la notación y terminología para los lenguajes formales que son la base de la teoría de control supervisor para sistemas de eventos discretos. En esta revisión incluimos los $_*$- y ω-lenguajes. Para material de consulta, refiérase a [14].

Sea Σ un alfabeto finito. Entonces, Σ^* es el conjunto de todas las palabras (o cadenas) finitas sobre Σ, incluyendo la cadena vacía 1. La expresión Σ^ω, representa el conjunto de palabras infinitas sobre Σ. La unión de Σ^* y Σ^ω la denotamos como Σ^∞.

Cualquier $L \subseteq \Sigma^*$ es un $_*$-lenguaje sobre Σ y cualquier $S \subseteq \Sigma^\omega$ es un ω-lenguaje sobre Σ. Para toda $k \in \Sigma^*$ y toda $b \in \Sigma^\infty$, kb es la concatenación de las dos palabras.

DEFINICIÓN 2.1.1. (***Producto de lenguajes***). Si K es un $_*$-lenguaje y B es un ω- o $_*$-lenguaje, entonces $KB := \{kb \in \Sigma^\infty : k \in K \text{ y } b \in B\}$ es el *producto* de K y B.

DEFINICIÓN 2.1.2. (***Cociente de lenguajes***). Si K es un $_*$-lenguaje y B es un ω- o $_*$-lenguaje, entonces el *cociente* de B sobre K lo definimos como

$$B/K := \{a \in \Sigma^\infty : ka \in B, \text{ para algún } k \in K\}$$

La cerradura de Kleene K^* de un $_*$-lenguaje K la definimos como $K^* := \bigcup_{i=0}^\infty K^i$, donde K^i es el i-ésimo producto de K consigo mismo. Definimos K^0 como $\{1\}$. De acuerdo a la definición, K^* es el conjunto de todas las concatenaciones de palabras en K. *La ω-cerradura de Kleene* K^ω de un $_*$-lenguaje K es el "producto infinito" de K consigo mismo, es decir, es el conjunto de todas las palabras $s = k_1k_2k_3...$, donde cada $k_i \in K \setminus \{1\}$.

Para toda $k \in \Sigma^*$ y toda $b \in \Sigma^\infty$, escribimos $k \leq b$ si k es un prefijo de b, es decir, si existe un $t \in \Sigma^\infty$ tal que kt es igual a b.

DEFINICIÓN 2.1.3. (*$_*$-Cerradura de un lenguaje*). La *$_*$-cerradura de un lenguaje* $B \subseteq \Sigma^\infty$ es la imagen de B bajo la función *pre*: $2^{\Sigma^\infty} \to 2^{\Sigma^*}$dada por la regla

$$pre(B) := B \mapsto \{k \in \Sigma^* : k \leq b, \text{ para algún } b \in B)\}$$

Si B es un $_*$-lenguaje, entonces pre(B)$\subseteq \Sigma^*$ es el *prefijo* de B. Si B es un ω-lenguaje, entonces pre(B) es la $_*$-cerradura de B. Si B es un $_*$-lenguaje tal que $B = pre(B)$, entonces decimos que B es *$_*$-cerrado*. De acuerdo a la definición, pre(B) es el conjunto de todos los prefijos finitos de las palabras contenidas en B.

La función *pre*: $2^{\Sigma^\infty} \to 2^{\Sigma^*}$ es monótona, como veremos a continuación.

PROPOSICIÓN 2.1.4. *Para cualesquiera $B_1, B_2 \subseteq \Sigma^\infty$, si $B_1 \subseteq B_2$, entonces*

$$pre(B_1) \subseteq pre(B_2).$$

DEMOSTRACIÓN. Supongamos que $B_1 \subseteq B_2$ y sea $k \in pre(B_1)$. Probaremos que $k \in pre(B_2)$. Como $k \in pre(B_1)$, existe un $b \in B_1$ tal que $k \leq b$. Pero como $B_1 \subseteq B_2$, entonces $b \in B_2$. Por lo tanto, $k \in pre(B_2)$. □

PROPOSICIÓN 2.1.5. *Para toda familia $\{B_\alpha\}_{\alpha\in\Lambda}$ de lenguajes en Σ^∞*

a) $pre(\cup_{\alpha\in\Lambda} B_\alpha) = \cup_{\alpha\in\Lambda} pre(B_\alpha)$

b) $pre(\cap_{\alpha\in\Lambda} B_\alpha) \subseteq \cap_{\alpha\in\Lambda} pre(B_\alpha)$

DEMOSTRACIÓN.

Parte a)

($\subseteq$)

Supongamos que $k \in pre(\cup_{\alpha\in\Lambda} B_\alpha)$. Entonces, existe un $b \in \cup_{\alpha\in\Lambda} B_\alpha$ tal que $k \leq b$ y $b \in B_\alpha$ para algún α. Por ser k un prefijo de b, $k \in pre(B_\alpha)$ para algún α. Por lo tanto, $k \in \cup_{\alpha\in\Lambda} pre(B_\alpha)$.

($\supseteq$)

Supongamos que $k \in \cup_{\alpha\in\Lambda} pre(B_\alpha)$. Entonces, $k \in pre(B_\alpha)$ para algún α, $b \in B_\alpha$ y $k \leq b$. Esto implica que $b \in \cup_{\alpha\in\Lambda} B_\alpha$ y por lo tanto $k \in pre(\cup_{\alpha\in\Lambda} B_\alpha)$.

Parte b)

Sea $k \in pre(\cap_{\alpha\in\Lambda} B_\alpha)$. Entonces, existe un $b \in \cap_{\alpha\in\Lambda} B_\alpha$ tal que $k \leq b$, es decir, tenemos que para todo α, $b \in B_\alpha$. Al ser k prefijo de b, deducimos que $k \in pre(B_\alpha)$ para todo α. Esto prueba que $k \in \cap_{\alpha\in\Lambda} pre(B_\alpha)$. □

Definición 2.1.6. (***Límite de un lenguaje***). El *límite* de un $_*$-lenguaje K es la imagen de K bajo la función $lim : 2^{\Sigma^*} \to 2^{\Sigma^\infty}$dada por la regla

$$lim(K) := pre^{-1}(K) \cap \Sigma^\omega,$$

donde $pre^{-1}(K)$ es la preimagen de K, es decir,

$$pre^{-1}(K) := \{b \in \Sigma^\infty : pre(b) \subseteq K\}$$

De acuerdo a la definición, el límite de $K \subseteq \Sigma^*$ es el conjunto de todas las palabras infinitas tales que todos sus prefijos están contenidos en K.

La función $lim : 2^{\Sigma^*} \to 2^{\Sigma^\infty}$ es monótona, como veremos a continuación.

Proposición 2.1.7. *Para cualesquiera $K_1, K_2 \subseteq \Sigma^*$, si $K_1 \subseteq K_2$, entonces*

$$lim(K_1) \subseteq lim(K_2)$$

Demostración. Supongamos que $K_1 \subseteq K_2$ y sea $s \in lim(K_1)$. Probaremos que $s \in lim(K_2)$. Como $s \in lim(K_1)$, entonces $pre(s) \subseteq K_1$. Luego, ya que $K_1 \subseteq K_2$, entonces $pre(s) \subseteq K_2$. Por lo tanto, $s \in lim(K_2)$. □

Proposición 2.1.8. *Para toda familia $\{L_\alpha\}_{\alpha\in\Lambda}$ de lenguajes en Σ^**

a) $\cup_{\alpha\in\Lambda} lim(L_\alpha) \subseteq lim(\cup_{\alpha\in\Lambda} L_\alpha)$

b) $\cap_{\alpha\in\Lambda} lim(L_\alpha) = lim(\cap_{\alpha\in\Lambda} L_\alpha)$

Demostración.

Parte a)

Supongamos que $t \in \cup_{\alpha\in\Lambda} lim(L_\alpha)$. Entonces, $pre(t) \subseteq L_\alpha$ para algún α. Pero $L_\alpha \subseteq \cup_{\alpha\in\Lambda} L_\alpha$. Esto implica que $pre(t) \subseteq \cup_{\alpha\in\Lambda} L_\alpha$. Por lo tanto, $t \in lim(\cup_{\alpha\in\Lambda} L_\alpha)$, por la definición 2.1.6.

Parte b)

($\subseteq$)

Sea $s \in \cap_{\alpha\in\Lambda} lim(L_\alpha)$. Entonces $s \in lim(L_\alpha)$ para todo α. Por la definición 2.1.6, $pre(s) \subseteq L_\alpha$ para todo α, es decir, $pre(s) \subseteq \cap_{\alpha\in\Lambda} L_\alpha$. Esto prueba que $s \in lim(\cap_{\alpha\in\Lambda} L_\alpha)$.

($\supseteq$)

La inclusión recíproca, se prueba de manera análoga.

Por lo tanto, $\cap_{\alpha\in\Lambda} lim(L_\alpha) = lim(\cap_{\alpha\in\Lambda} L_\alpha)$. □

PROPOSICIÓN 2.1.9. *Para toda palabra $m \in \Sigma^*$, cualesquiera lenguajes $K \subseteq \Sigma^*$, $B \subseteq \Sigma^\infty$ y toda familia $\{B_\alpha\}_{\alpha\in\Lambda}$ de lenguajes en Σ^∞,*

a) $pre(lim(K)) \subseteq K$

b) $m(B/m) \subseteq B$

c) $pre(B/K) = pre(B)/K$

d) $\bigcap_{\alpha\in\Lambda}(KB_\alpha) \supseteq K(\bigcap_{\alpha\in\Lambda} B_\alpha)$

e) si K contiene un solo elemento, entonces cambiar a igualdad en la parte (d)

f) $\bigcap_{\alpha\in\Lambda}(B_\alpha/K) \supseteq \bigcap_{\alpha\in\Lambda} B_\alpha/K$

g) si K contiene un solo elemento, entonces cambiar a igualdad en la parte (f)

DEMOSTRACIÓN.

Parte a)

Sea $l \in pre(lim(K))$. Entonces, existe una palabra $l' \in lim(K)$ tal que $l \leq l'$ y $pre(l') \subseteq K$. Por lo tanto, $l \in K$ por la definición 2.1.6.

Parte b)

Supongamos que $a \in m(B/m)$. Entonces, $a = mb$ tal que $b \in B/m$. Luego, por la definición 2.1.2, tenemos que $mb \in B$. Pero $a = mb$ y por lo tanto $a \in B$.

Parte c)

($\subseteq$)

Sea $l \in pre(B/K)$. Entonces, existe un $b \in B/K$ tal que $l \leq b$. Por la definición 2.1.2, existe un $k \in K$ tal que $kb \in B$. Por ser l prefijo de b, entonces $kl \in pre(B)$ y por lo tanto $l \in pre(B)/k$.

($\supseteq$)

Sea $l \in pre(B)/K$. Entonces, existe un $k \in K$ tal que $kl \in pre(B)$. Por la definición 2.1.3, existe un $b \in B$ tal que $kl \leq b$. Esto implica que $kl/k \leq b/k$. Luego, $l \leq b/k$ y $b/k \in B/K$. Por lo tanto, $l \in pre(B/K)$.

Parte d)

Sea $a \in K(\bigcap_{\alpha\in\Lambda} B_\alpha)$. Entonces, $a = kb$ tal que $k \in K$ y $b \in \bigcap_{\alpha\in\Lambda} B_\alpha$. Luego, $b \in B_\alpha$ para todo $\alpha \in \Lambda$. Como $a = kb$, entonces para todo $\alpha \in \Lambda$, $a \in KB_\alpha$ y por lo tanto $a \in \bigcap_{\alpha\in\Lambda}(KB_\alpha)$.

Parte e)

($\supseteq$)

Se obtiene directamente de la parte (d).

($\subseteq$)

Supongamos que $k \in \Sigma^*$ y sea $a \in \cap_{\alpha\in\Lambda}(kB_\alpha)$. Como $a \in \cap_{\alpha\in\Lambda}(kB_\alpha)$, entonces $a \in kB_\alpha$ para todo $\alpha \in \Lambda$. Luego, $a = kb$ tal que $b \in B_\alpha$, para todo $\alpha \in \Lambda$. Esto implica que $a = kb$ y $b \in \cap_{\alpha\in\Delta}B_\alpha$. Por lo tanto, $a \in k(\cap_{\alpha\in\Lambda}B_\alpha)$ por la definición 2.1.1.

Parte f)

Sea $a \in (\cap_{\alpha\in\Lambda}B_\alpha)/K$. Entonces, por la definición 2.1.2, existe un $k \in K$ tal que $ka \in \cap_{\alpha\in\Lambda}B_\alpha$. Luego, $k \in K$ y $ka \in B_\alpha$ para todo $\alpha \in \Lambda$. Por la definición 2.1.2, tenemos que $a \in B_\alpha/K$ para todo $\alpha \in \Lambda$. Por lo tanto, $a \in \cap_{\alpha\in\Lambda}(B_\alpha/K)$.

Parte g)

($\supseteq$)

Se demuestra directamente de la parte (f).

($\subseteq$)

Sea $a \in \cap_{\alpha\in\Lambda}(B_\alpha/k)$. Entonces, $a \in B_\alpha/k$ para todo $\alpha \in \Lambda$. Por la definición 2.1.2, tenemos que $ak \in B_\alpha$ para todo $\alpha \in \Lambda$. Luego, $ka \in \cap_{\alpha\in\Lambda}B_\alpha$. Por lo tanto, $a \in (\cap_{\alpha\in\Lambda}B_\alpha)/k$. □

PROPOSICIÓN 2.1.10. *Para toda palabra $m \in \Sigma^*$ y todo $B \subseteq \Sigma^\infty$*

a) $mlim(pre(B)) = lim(mpre(B))$

b) $lim(pre(B))/m = lim(pre(B)/m)$

DEMOSTRACIÓN.

Parte a)

($\subseteq$)

Sea $s \in mlim(pre(B))$. Entonces, $s = mb$ tal que $b \in lim(pre(B))$. Luego, $b = s/m$ y $pre(b) \subseteq pre(B)$. Como $b = s/m$, entonces $pre(s/m) \subseteq pre(B)$. Por la proposición 2.1.9(c), tenemos que $mpre(s)/m \subseteq mpre(B)$. Esto implica que $pre(s) \subseteq mpre(B)$ y por lo tanto $s \in lim(mpre(B))$.

($\supseteq$)

Sea $s \in lim(mpre(B))$. Entonces, $pre(s) \subseteq mpre(B)$, es decir, se cumple que $pre(s)/m \subseteq mpre(B)/m$. Esto implica que $pre(s)/m \subseteq pre(B)$. Pero, por la proposición 2.1.9(c), tenemos que $pre(s/m) \subseteq pre(B)$. Luego, $s/m \in lim(pre(B))$ y por lo tanto $s \in mlim(pre(B))$.

Parte b)

($\subseteq$)

Sea $s \in lim(pre(B))/m$. Entonces, $s = b/m$ tal que $b \in lim(pre(B))$ y $pre(b) \subseteq pre(B)$, es decir, $pre(b/m) \subseteq pre(B)/m$. Luego, $pre(s) \subseteq pre(B)/m$ por la proposición 2.1.9(c). Por lo tanto, $s \in lim(pre(B)/m)$.

($\supseteq$)

Sea $s \in lim(pre(B)/m)$. Entonces, $pre(s) \subseteq pre(B)/m$, es decir, se cumple que $mpre(s) \subseteq mpre(B)/m$. Esto implica que $mpre(s) \subseteq pre(B)$ y por lo tanto, $ms \in lim(pre(B))$. Esto prueba que $s \in lim(pre(B))/m$. □

DEFINICIÓN 2.1.11. (*ω-Cerradura de un lenguaje*). La *ω-cerradura* de un ω-lenguaje R es la imagen de R bajo la función $clo : 2^{\Sigma^\omega} \to 2^{\Sigma^\omega}$ dada por la regla

$$clo(R) := lim(pre(R)) = pre^{-1}(pre(R)) \cap \Sigma^\omega$$

Si $R = clo(R)$, decimos que R es *ω-cerrado*. Si $R = clo(R) \cap S$, tal que $S \subseteq \Sigma^\omega$, decimos que R es *ω-cerrado con respecto a S*.

De acuerdo a la definición de la función $clo : 2^{\Sigma^\omega} \to 2^{\Sigma^\omega}$, los lenguajes ω-cerrados son completamente determinados por sus prefijos. En otras palabras, clo(R) es el conjunto de todas las palabras infinitas, tales que todos sus prefijos están contenidos en pre(R).

La función $clo : 2^{\Sigma^\omega} \to 2^{\Sigma^\omega}$ es monótona, como lo demuestra la siguiente proposición.

PROPOSICIÓN 2.1.12. *Para cualesquiera $R_1, R_2 \subseteq \Sigma^\omega$, si $R_1 \subseteq R_2$, entonces*

$$clo(R_1) \subseteq clo(R_2)$$

DEMOSTRACIÓN. Supongamos que $R_1 \subseteq R_2$ y sea $t \in clo(R_1)$. Probaremos que $t \in clo(R_2)$.

Como $t \in clo(R_1)$, entonces $t \in lim(pre(R_1))$. Por la definición 2.1.6, tenemos que $pre(t) \subseteq pre(R_1)$ y como $R_1 \subseteq R_2$, entonces $pre(t) \subseteq pre(R_2)$ por la proposición 2.1.7. Por lo tanto, $t \in clo(R_2)$. □

PROPOSICIÓN 2.1.13. *Para todo $R \subseteq \Sigma^{\omega}$ se cumple que*

a) $R \subseteq clo(R)$

b) $clo(clo(R)) = clo(R)$

DEMOSTRACIÓN.

Parte a)

Sea $r \in R$ y supongamos que $r \notin clo(R)$. Entonces, por la proposición 2.1.11, tenemos que $r \notin pre^{-1}(pre(R)) \cap \Sigma^{w}$, es decir, $r \notin pre^{-1}(pre(R))$ o $r \notin \Sigma^{w}$. Si $r \notin pre^{-1}(pre(R))$, entonces existe un $r' \in pre(r)$ tal que $r' \notin pre(R)$. Esto implica que $r \notin R$. Pero esto es una contradicción, ya que $r \in R$.

Por otra parte, si $r \notin \Sigma^{w}$, entonces también tenemos una contradicción, ya que $r \in R$ y $R \subseteq \Sigma^{w}$.

Por lo tanto, $r \in clo(R)$.

Parte b)

($\supseteq$)

Por la parte (a), tenemos que $clo(clo(R)) \supseteq clo(R)$.

($\subseteq$)

Ahora supongamos que $s \in clo(clo(R))$. Entonces, por la definición 2.1.11, tenemos que $s \in lim(pre(clo(R)))$, es decir, $pre(s) \subseteq pre(lim(pre(R)))$. Pero, por la proposición 2.1.9(a), sabemos que $pre(lim(pre(R))) \subseteq pre(R)$. Esto implica que $pre(s) \subseteq pre(R)$. Por lo tanto, $s \in lim(pre(R))$ si y sólo si $s \in clo(R)$. □

PROPOSICIÓN 2.1.14. *Para toda familia $\{B_\alpha\}_{\alpha\in\Lambda}$ de lenguajes en Σ^{ω}, si para toda α se cumple que $R_\alpha = clo(R_\alpha)$, entonces $\bigcap_{\alpha\in\Lambda} R_\alpha = clo(\bigcap_{\alpha\in\Lambda} R_\alpha)$.*

DEMOSTRACIÓN.

($\supseteq$)

$$\begin{aligned}
\textstyle\bigcap_{\alpha\in\Lambda} R_\alpha &= \textstyle\bigcap_{\alpha\in\Lambda} clo(R_\alpha) \\
&= \textstyle\bigcap_{\alpha\in\Lambda} lim(pre(R_\alpha)) \\
&\supseteq lim(\textstyle\bigcap_{\alpha\in\Lambda} pre(R_\alpha)) && \text{(por la proposición 2.1.8 } (b)) \\
&\supseteq clo(\textstyle\bigcap_{\alpha\in\Lambda} R_\alpha) && \text{(por la proposición 2.1.5 } (b))
\end{aligned}$$

($\subseteq$)

Obtenemos directamente que $\bigcap_{\alpha\in\Lambda} R_\alpha \subseteq clo(\bigcap_{\alpha\in\Lambda} R_\alpha)$ de la proposición 2.1.13(a).

Por lo tanto, $\cap_{\alpha\in\Lambda} R_\alpha = clo(\cap_{\alpha\in\Lambda} R_\alpha)$. □

2.2. Autómatas Finitos

En esta sección introducimos algunas definiciones básicas de la teoría de autómatas finitos. Estos autómatas son las estructuras que usaremos para representar los sistemas de eventos discretos.

2.2.1. Autómatas finitos sobre palabras finitas. Los *autómatas finitos sobre palabras finitas* también reciben el nombre de $_*$-autómatas y a continuación los definimos formalmente.

DEFINICIÓN 2.2.1. (*$_*$-autómata*). Un $_*$-autómata es una estructura $\mathcal{A} = (\Sigma, X, \delta, x_0, F)$, donde

- Σ es un alfabeto de entrada,
- X es un conjunto de estados,
- $\delta: \Sigma \times X \to 2^X$ es la función parcial de transición de estados,
- x_0 es el estado inicial y
- $F \subseteq X$ es el conjunto de estados terminales.

Extendemos la función de transición parcial $\delta : \Sigma \times X \to 2^X$ a la función $\delta' : \Sigma^* \times X \to 2^X$ dada por las reglas:

- $\delta'(1, x) := x$,
- $\delta'(k\sigma, x) := \cup\{\delta(\sigma, x_k),$ donde $x_k \in \delta(k, x)\}$ para todo $k \in \Sigma^*$ y $\sigma \in \Sigma$.

De ahora en adelante omitiremos la "′" y escribiremos δ para referirnos a δ'

Cuando escribimos $\delta(k, x)!$ significa que $\delta(k, x)$ está definida. Decimos que $\mathcal{A}$ es determinista si $| \delta(\sigma, x) | \leq 1$, para todo $\sigma \in \Sigma$ y todo $x \in X$.

Un *camino* en $\mathcal{A}$ de una palabra $m \in \Sigma^*$ es una función total $\pi : pre(m) \to X$ tal que $\pi(1) = x_0$ y para todo $k \in pre(m)$ y todo $\sigma \in \Sigma$, si $k\sigma \in pre(m)$, entonces $\pi(k\sigma) \in \delta(\sigma, \pi(k))$.

Un *camino* π de una palabra $k \in \Sigma^*$ *es aceptado* si $\pi(1) = x_0$ y $\pi(k) \in F$. Decimos que una *palabra es aceptada* si tiene un camino que comienza en el estado inicial y termina en un estado terminal.

El $_*$-*lenguaje reconocido* por $\mathcal{A}$, que denotamos como $L(\mathcal{A})$, es el conjunto de todas las palabras finitas sobre Σ que tienen un camino aceptado en $\mathcal{A}$.

2.2.2. Autómatas finitos sobre palabras infinitas. Los *autómatas finitos sobre palabras infinitas* también reciben el nombre de ω-autómatas. A diferencia de los $_*$-autómatas, el criterio de aceptación de los ω-autómatas no depende de un conjunto de estados terminales sino de un conjunto de estados de recurrencia, como veremos a continuación.

DEFINICIÓN 2.2.2. ***(Autómata de Büchi)***. Un autómata de Büchi es una estructura $\mathcal{A} = (\Sigma, X, \delta, x_0, R)$, donde

- Σ es un alfabeto de entrada,
- X es un conjunto de estados,
- $\delta : \Sigma \times X \to 2^X$ es la función parcial de transición de estados,
- x_0 es el estado inicial,
- R es el conjunto de estados de recurrencia.

Extendemos la función de transición parcial $\delta : \Sigma \times X \to 2^X$ a la función $\delta' : \Sigma^* \times X \to 2^X$ dada por las reglas:

- $\delta'(1, x) := x$,
- $\delta'(k\sigma, x) := \cup\{\delta(\sigma, x_k),\ \text{donde } x_k \in \delta(k, x)\}$ para todo $k \in \Sigma^*$ y $\sigma \in \Sigma$.

Igual que en la sección anterior, de ahora en adelante omitiremos la "′" y escribiremos δ para referirnos a δ'

Un *camino* en $\mathcal{A}$ de una palabra $s \in \Sigma^\omega$ es una función total $\pi : pre(s) \to X$ tal que $\pi(1) = x_0$ y para todo $k \in pre(s)$ y todo $\sigma \in \Sigma$, si $k\sigma \in pre(s)$, entonces $\pi(k\sigma) \in \delta(\sigma, \pi(k))$.

Definimos el *conjunto de recurrencia* de un camino π de una palabra $s \in \Sigma^\omega$ como

$$\Omega_\pi := \{x \in X :\ \{\delta(k, x_0) = x\} \text{ es infinito, para } k \leq s\}$$

De acuerdo a la definición, el conjunto de recurrencia es el conjunto de estados que aparecen un número infinito de veces en un camino. Un camino π es aceptado si $\pi(1) = x_0$ y $\Omega_\pi \cap R \neq \emptyset$. Es decir, un camino π es aceptado si comienza en el estado inicial y en el conjunto de recurrencia de ese camino existe por lo menos un estado en R.

Decimos que una palabra $s \in \Sigma^\omega$ es reconocida por $\mathcal{A}$ si s tiene un camino en $\mathcal{A}$. El ω-lenguaje reconocido por $\mathcal{A}$, que denotamos como $L_\omega(\mathcal{A})$, es el conjunto de todas las palabras infinitas sobre Σ que tienen un camino aceptado en $\mathcal{A}$.

DEFINICIÓN 2.2.3. ***(Autómata de Rabin).*** Un autómata de Rabin es un autómata finito no determinista que tiene un conjunto de pares de aceptación. Formalmente lo definimos como $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$, donde

- Σ es un alfabeto de entrada,
- X es un conjunto de estados,
- $\delta : \Sigma \times X \to 2^X$es la función parcial de transición de estados,
- x_0 es el estado inicial,
- $\{(R_p, I_p) : p \in P\}$ es un conjunto de pares de aceptación $(R_p, I_p) \in 2^X \times 2^X$ con subíndices en P.

Un camino π es aceptado por el autómata de Rabin $\mathcal{A}$ si y sólo si $\Omega_\pi \cap R_p \neq \emptyset$ y $\Omega_\pi \subseteq I_p$ para algún $p \in P$. Es decir, un camino es aceptado si para algún $p \in P$, R_p es "continuamente recurrente" e I_p es "eventualmente invariante".

DEFINICIÓN 2.2.4. ***(Autómata de Rabin-Büchi).*** Un autómata de Rabin-Büchi es un autómata de Büchi $\mathcal{A} = (\Sigma, X, \delta, x_0, R)$ al que le agregamos un conjunto de pares de aceptación $(R_p, I_p) \in 2^X \times 2^X$ con subíndices en P para formar un autómata $(\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$.

DEFINICIÓN 2.2.5. ***(Autómata de Streett).*** Definimos a un *autómata de Streett* $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ igual que un autómata de Rabin, pero con diferente condición de aceptación. De manera que un camino π es aceptado por el autómata $\mathcal{A}$ si y sólo si para todo $p \in P$, $\Omega_\pi \cap R_p \neq \emptyset$ o $\Omega_\pi \subseteq I_p$. Es decir, un camino es aceptado si para todo $p \in P$, R_p es "continuamente recurrente" o I_p es "eventualmente invariante"

2.3. Cálculo de Puntos Fijos

En ésta sección presentamos la teoría de cálculo de puntos fijos, tomada básicamente de [11]. Esta notación de puntos fijos permite la anidación de expresiones.

2.3.1. Preliminares.

DEFINICIÓN 2.3.1. *(Funciones monótonas y continuas).* Sea $f : 2^X \to 2^X$ una función en la colección de subconjuntos de X. Entonces,

1. f es una función $\subseteq$-*monótona* si y sólo si $X \subseteq X'$ implica $f(X) \subseteq f(X')$.
2. f es una función $\cup$-*continua* si y sólo si para toda sucesión creciente $X_0 \subseteq X_1 \subseteq X_2 ...$, se cumple que $\bigcup_{i=0}^{\infty} f(X_i) = f(\bigcup_{i=0}^{\infty} X_i)$.
3. f es una función $\cap$-*continua* si y sólo si para toda sucesión decreciente $X_0 \supseteq X_1 \supseteq X_2 ...$, se cumple que $\bigcap_{i=0}^{\infty} f(X_i) = f(\bigcap_{i=0}^{\infty} X_i)$.

Las definiciones anteriores se aplican también a funciones k-arias, es decir, a funciones que tienen k argumentos.

PROPOSICIÓN 2.3.2. *Sea $f : 2^X \to 2^X$ Si f es una función $\subseteq$-monótona, entonces f es $\cup$-continua.*

DEMOSTRACIÓN. Supongamos que f es una función $\subseteq$-monótona y sea $X_1 \subseteq X_2 \subseteq ...$ una secuencia creciente de subconjuntos en X. Probaremos que $f(\cup_{i=0}^{\infty} X_i) = \cup_{i=0}^{\infty} f(X_i)$.

$(\subseteq)$

Sea $x \in f(\cup_{i=0}^{\infty} X_i)$. Entonces, existe $a \in \cup_{i=0}^{\infty} X_i$ tal que $x = f(a)$ y existe $i \in N$ tal que $a \in X_i$. Luego, $f(a) \in f(X_i)$, es decir, $x \in f(X_i)$ para algún $i \in N$. Por lo tanto, $x \in \cup_{i=0}^{\infty} f(X_i)$. Esto prueba que $f(\cup_{i=0}^{\infty} X_i) \subseteq \cup_{i=0}^{\infty} f(X_i)$.

$(\supseteq)$

Sea $x \in \cup_{i=0}^{\infty} f(X_i)$. Entonces, existe $i \in N$ tal que $x \in f(X_i)$ y existe $a \in X_i$ tal que $x = f(a)$. Luego, $a \in \cup_{i=0}^{\infty} X_i$. Como f es $\subseteq$-monótona, entonces $f(a) \in f(\cup_{i=0}^{\infty} X_i)$, pero $x = f(a)$. Por lo tanto, $x \in f(\cup_{i=0}^{\infty} X_i)$. Esto prueba que $f(\cup_{i=0}^{\infty} X_i) \supseteq \cup_{i=0}^{\infty} f(X_i)$. □

PROPOSICIÓN 2.3.3. *Sea $f : 2^X \to 2^X$. Si f es una función $\subseteq$-monótona, entonces f es $\cap$-continua.*

DEMOSTRACIÓN. Supongamos que f es una función $\subseteq$-monótona. Sea $X_1 \supseteq X_2 \supseteq ...$ una secuencia decreciente de subconjuntos en X. Probaremos que $f(\cap_{i=0}^{\infty} X_i) = \cap_{i=0}^{\infty} f(X_i)$

$(\subseteq)$

Sea $x \in f(\cap_{i=0}^{\infty} X_i)$. Entonces, existe $a \in \cap_{i=0}^{\infty} X_i$ tal que $x = f(a)$ y $a \in X_i$ para todo $i \in N$. Por ser f es una función $\subseteq$-monótona, entonces $f(a) \in f(X_i)$, es decir, $x \in f(X_i)$ para todo $i \in N$. Por lo tanto, $x \in \cap_{i=0}^{\infty} f(X_i)$. Esto prueba que $f(\cap_{i=0}^{\infty} X_i) \subseteq \cap_{i=0}^{\infty} f(X_i)$.

$(\supseteq)$

Sea $x \in \cap_{i=0}^{\infty} f(X_i)$. Entonces, $x \in f(X_i)$, para todo $i \in N$, y existe $a \in X_i$ tal que $x = f(a)$. Como X_i es finito para todo $i \in N$, entonces existe un $n \geq 1$ tal que para todo $m > n$, $X_m = X_n$. Luego, $\cap_{i=0}^{\infty} X_i = X_n$. Pero como $f(a) \in X_n$, entonces $f(a) \in \cap_{i=0}^{\infty} X_i$ Por ser $x = f(a)$, entonces $x \in \cap_{i=0}^{\infty} X_i$. Esto prueba que $f(\cap_{i=0}^{\infty} X_i) \supseteq \cap_{i=0}^{\infty} f(X_i)$. □

2.3.2. Cálculo de puntos fijos.

DEFINICIÓN 2.3.4. (***Punto fijo***). Sea X un conjunto y sea f una función. Decimos que X es un punto fijo de f si y sólo si $X \in dom(f)$ y $X = f(X)$.

TEOREMA 2.3.5. ***(Tarski-Knaster).*** [10] *Sea $f : 2^X \to 2^X$ una función $\subseteq$-monótona en 2^X Entonces, X tiene un punto fijo máximo y un punto fijo mínimo.*

1. $$\begin{aligned} \mu Y.f(Y) &= \cap\{Y' \subseteq X : Y' = f(Y')\} \\ &= \cap\{Y' \subseteq X : Y' \supseteq f(Y')\} \end{aligned}$$
2. $$\begin{aligned} \nu Y.f(Y) &= \cup\{Y' \subseteq X : Y' = f(Y')\} \\ & \quad \cup\{Y' \subseteq X : Y' \subseteq f(Y')\} \end{aligned}$$
3. *Si f es una función $\cup$-continua, entonces $\mu Y.f(Y) = \bigcup_{i=0}^{\infty} f^i(\emptyset)$*
4. *Si f es una función $\cap$-continua, entonces $\nu Y.f(Y) = \bigcap_{i=0}^{\infty} f^i(X)$*

El lema que sigue a continuación se deriva del teorema de Tarski y Knaster.

LEMA 2.3.6. *Sean $f_1 : 2^X \to 2^X$ y $f_2 : 2^X \to 2^X$ funciones $\subseteq$-monótonas en 2^X Si $f_1(Y) \subseteq f_2(Y)$, para todo $Y \subseteq X$, entonces*

a) $\mu Y.f_1(Y) \subseteq \mu Y.f_2(Y)$

b) $\nu Y.f_1(Y) \subseteq \nu Y.f_2(Y)$

DEMOSTRACIÓN.

Parte 1)

Como f_1 y f_2 son funciones $\subseteq$-monótonas, f_1 y f_2 son $\cup$-continuas, por el teorema 2.3.2. Ahora bien, por el teorema 2.3.5, tenemos que $\mu Y.f_1(Y) = \bigcup_{i=0}^{\infty} f_1^i(\emptyset)$ y $\mu Y.f_2(Y) = \bigcup_{i=0}^{\infty} f_2^i(\emptyset)$.

Probaremos que $\bigcup_{i=0}^{\infty} f_1^i(\emptyset) \subseteq \bigcup_{i=0}^{\infty} f_2^i(\emptyset)$ utilizando inducción en i.

Caso Base: Si $i = 0$, entonces $f_1(\emptyset) \subseteq f_2(\emptyset)$.

Hipótesis de inducción: Sea $i = n$ y supongamos que $f_1^n(\emptyset) \subseteq f_2^n(\emptyset)$.

Sabemos que $f_1^{n+1}(\emptyset) := f_1(f_1^n(\emptyset))$ y como f_1 es una función $\subseteq$-monótona, entonces por la hipótesis de inducción tenemos que $f_1(f_1^n(\emptyset)) \subseteq f_1(f_2^n(\emptyset))$. Pero $f_1(f_2^n(\emptyset)) \subseteq f_2(f_2^n(\emptyset))$ por hipótesis, tomando $Y = f_2^n(\emptyset)$. Esto implica que $f_1(f_1^n(\emptyset)) \subseteq f_2(f_2^n(\emptyset))$, es decir, $f_1^{n+1}(\emptyset) \subseteq f_2^{n+1}(\emptyset)$.

Sea ahora $x \in \bigcup_{i=0}^{\infty} f_1^i(\emptyset)$. Probaremos que $x \in \bigcup_{i=0}^{\infty} f_2^i(\emptyset)$. Como $x \in \bigcup_{i=0}^{\infty} f_1^i(\emptyset)$, entonces $x \in f_1^i(\emptyset)$ para algún $i \in N$. Puesto que $f_1^i(\emptyset) \subseteq f_2^i(\emptyset)$, luego $x \in f_2^i(\emptyset)$, es decir, $x \in f_2^i(\emptyset)$ para algún $i \in N$. Por lo tanto, $x \in \bigcup_{i=0}^{\infty} f_2^i(\emptyset)$. Esto prueba que $\bigcup_{i=0}^{\infty} f_1^i(\emptyset) \subseteq \bigcup_{i=0}^{\infty} f_2^i(\emptyset)$.

Parte 2)

Sea $z \in \cup\{Y' \subseteq X : Y' = f_1(Y')\}$. Entonces, $z \in Y'$ para algún Y'. Como $f_1(Y') \subseteq f_2(Y')$, entonces $Y' \subseteq f_2(Y')$, es decir, $z \in f_2(Y')$. Por lo tanto, $z \in \cup\{Y' \subseteq X : Y' \subseteq f_2(Y')\}$. $\square$

CAPÍTULO 3

Sistemas de Eventos Discretos y Supervisión

En este capítulo presentamos la teoría de control supervisor para el comportamiento infinito de los sistemas de eventos discretos de Thistle y Wonham [**13**]. Esta teoría de control está basada en [**8**] y [**3**].

Es importante mencionar que el desarrollo de las demostraciones presentadas en las siguientes secciones fueron tomadas de [**13, 11**]. Algunos pasos de estas demostraciones fueron agregados con la finalidad de clarificar las demostraciones.

3.1. Sistemas de Eventos Discretos (SED)

Modelamos un SED como un par $G = (L, S) \in 2^{\Sigma^*} \times 2^{\Sigma^\omega}$ que consiste de un $_*$-lenguaje L que denominamos el $_*$-comportamiento de G y un ω-lenguaje S que denominamos el ω-comportamiento de G [**8**]. Asumimos que el $_*$-comportamiento L es cerrado, es decir, asumimos que $pre(L) = L$. También asumimos que $pre(S) \subseteq L$. Si $pre(S) = L$, entonces decimos que el SED G es no bloqueante.

3.2. Supervisores

Al modelo G le agregamos un modelo de control propuesto en [**3**]. Al conjunto Σ lo dividimos en dos conjuntos disjuntos Σ_u y Σ_c. Los eventos en Σ_u son eventos incontrolables y los eventos en Σ_c se pueden inhabilitar (prevenir que ocurran).

Asociamos al alfabeto Σ una familia no vacía $\mathcal{C} \subseteq 2^\Sigma$ de patrones de control

$$\mathcal{C} = \{\Gamma : \Gamma \in 2^\Sigma \text{ y } \Sigma_u \subseteq \Gamma\}$$

Un supervisor es una función parcial $f : L \rightarrow \mathcal{C}$. Damos por supuesto que $\mathcal{C}$ es cerrado bajo la unión[1] e intersección[2] de conjuntos, es decir, si $\Gamma, \Gamma' \in \mathcal{C}$ entonces $\Gamma \cup \Gamma' \in \mathcal{C}$ y $\Gamma \cap \Gamma' \in \mathcal{C}$.

Para todo SED $G = (L, S)$ y todo supervisor f, el sistema de eventos discretos controlado G^f, representa la acción del supervisor $f : L \rightarrow \mathcal{C}$ sobre el SED G, definimos $G^f = (L^f, S^f)$, donde

1. L^f es el $_*$-lenguaje generado por G bajo la supervisión de f y lo definimos de manera recursiva como sigue:
   (a) $1 \in L^f$.
   (b) Para todo $k \in \Sigma^*$ y todo $\sigma \in \Sigma$,

$$k\sigma \in L^f \text{ si y sólo si } k \in L^f \cap dom(f), k\sigma \in L \quad \text{y} \quad \sigma \in f(k);$$

2. S^f es el ω-lenguaje generado por G bajo la supervisión de f y lo definimos como $S^f := lim(L^f) \cap S$.

De acuerdo a la definición, S^f es el conjunto de todas las palabras infinitas generadas por el SED G que "sobreviven" a la supervisión de f.

DEFINICIÓN 3.2.1. (***Supervisor Completo***). Un supervisor $f : L \rightarrow \mathcal{C}$ es un supervisor *completo* para un SED $G = (L, S)$ si y sólo si $L^f \subseteq dom(f)$.

PROPOSICIÓN 3.2.2. *Para todo SED G y todo supervisor $f : L \rightarrow \mathcal{C}$, $G^f = (L^f, S^f)$ es un SED tal que $pre(L^f) = L^f$ y $pre(S^f) \subseteq L^f$. Además el $_*$-comportamiento y el ω-comportamiento de G^f son sublenguajes de G, es decir, $L^f \subseteq L$ y $S^f \subseteq S$. También S^f es ω-cerrado con respecto a S.*

DEMOSTRACIÓN.

$L^f \subseteq L$ y $S^f \subseteq S$ se obtienen de las definiciones de L^f y S^f respectivamente. Por la definición de L^f, tenemos que $L^f = pre(L^f)$.

---

[1]Esta propiedad asegura la existencia de sublenguajes supremos controlables [3]. Esto se debe a que el conjunto $\mathcal{C}$ forma una *lattice* bajo $\subseteq$, donde Σ_u es el único elemento minimal y Σ es el único elemento maximal.

[2]Esta propiedad se asocia a la síntesis modular de supervisores.

$$\begin{aligned} pre(S^f) &= pre(lim(L^f)) \cap S && \text{(por la definición de } S^f) \\ &\subseteq L^f \cap S && \text{(por la proposición 2.1.9 (a))} \\ &\subseteq L^f \end{aligned}$$

Finalmente, probaremos que $S^f = clo(S^f) \cap S$.

$(\subseteq)$

Por la proposición 2.1.13(a), tenemos que $S^f \subseteq clo(S^f)$ y como $S^f \subseteq S$, entonces $S^f \subseteq clo(S^f) \cap S$.

$(\supseteq)$

De la definición de S^f tenemos que

$$\begin{aligned} S^f &= lim(L^f) \cap S \\ &\supseteq lim(pre(S^f)) \cap S && (pre(S^f) \subseteq L^f) \\ &= clo(S^f) \cap S \end{aligned}$$

lo que completa la demostración. □

DEFINICIÓN 3.2.3. (***Supervisor no bloqueante***). Todo supervisor $f : L \to C$ es un supervisor no bloqueante para un SED $G = (L, S)$ si $pre(S^f) = L^f$.

3.3. El Comportamiento del Sistema bajo Supervisión y Controlabilidad

En la siguiente sección definimos el comportamiento del sistema controlado identificando el $_*$-lenguaje y el ω-lenguaje generados por G bajo la supervisión de f.

3.3.1. El $_*$-comportamiento del sistema bajo supervisión y $_*$-controlabilidad.

Definimos el concepto de $_*$-lenguaje controlable, tomado de [3], para caracterizar el $_*$-comportamiento del sistema bajo supervisión.

DEFINICIÓN 3.3.1. (***Conjunto activo***). Para todo lenguaje $B \subseteq \Sigma^\infty$ y todo $l \in pre(B)$, definimos el *conjunto activo de B después de l*, que denotamos $\Sigma_B(l) \subseteq \Sigma$, como

$$\Sigma_B(l) := \Sigma \cap (pre(B)/l)$$

En otras palabras, Σ_B es el conjunto de todos los $\sigma \in \Sigma$ tales que $l\sigma \in pre(B)$.

DEFINICIÓN 3.3.2. (*$_*$-Controlabilidad*). Sean $B \subseteq \Sigma^\infty$ y $L \subseteq \Sigma^*$ tales que $pre(B) \subseteq L$. Decimos que B es $_*$-controlable con respecto a L si y sólo si para todo $l \in pre(B)$, existe un $\Gamma \in \mathcal{C}$ tal que

$$\Gamma \cap \Sigma_L(l) = \Sigma_B(l)$$

En otras palabras, B es $_*$-controlable con respecto a L si y sólo si las extensiones $l\sigma \in l\Sigma \cap pre(L)$ de todo $l \in pre(B)$ pueden ser restringidas por medio de control a ser iguales a $l\sigma \in pre(B)$.

La definición de $_*$-controlabilidad es equivalente a las siguientes dos condiciones [3]:

1. $\Sigma_B(l) \subseteq \Gamma$ y
2. $l\Gamma \cap L \subseteq pre(B)$

La $_*$-controlabilidad es preservada bajo la unión de conjuntos, como veremos a continuación.

PROPOSICIÓN 3.3.3. *La $_*$-controlabilidad es preservada bajo la unión arbitraria de conjuntos.*

DEMOSTRACIÓN. Sea $L \subseteq \Sigma^*$ y sea $\{B_\alpha\}_{\alpha\in\Lambda}$ una familia de lenguajes en Σ^∞ tales que $pre(B_\alpha) \subseteq L$. Supongamos que B_α es $_*$-controlable, para todo $\alpha \in \Lambda$.

Probaremos que para toda palabra $k \in pre(\cup_{\alpha\in\Lambda} B_\alpha)$ existe un $\Gamma \in \mathcal{C}$ tal que

$$\Gamma \cap \Sigma_L(k) = \Sigma_{\cup_{\alpha\in\Lambda} B_\alpha}(k)$$

($\subseteq$)

Sean $k \in pre(\cup_{\alpha\in\Lambda} B_\alpha)$ y $\sigma_k \in \Sigma_{\cup_{\alpha\in\Lambda} B_\alpha}(k)$. Probaremos que existe un $\Gamma \in \mathcal{C}$ tal que $\sigma_k \in \Gamma \cap \Sigma_L(k)$.

Como $k \in pre(\cup_{\alpha\in\Lambda} B_\alpha)$, entonces por la proposición 2.1.5(a), $k \in \cup_{\alpha\in\Lambda} pre(B_\alpha)$. Como $\sigma_k \in \Sigma_{\cup_{\alpha\in\Lambda} B_\alpha}(k)$, entonces $k\sigma_k \in \cup_{\alpha\in\Lambda} pre(B_\alpha)$. Esto implica que $k\sigma_k \in pre(B_\alpha)$ para algún $\alpha \in \Lambda$. Como $k \leq k\sigma_k$ y B_α es $_*$-controlable, entonces existe $\Gamma \in \mathcal{C}$ tal que $\Gamma \cap \Sigma_L(k) = \Sigma_{B_\alpha}(k)$. Por lo tanto, $\sigma_k \in \Gamma \cap \Sigma_L(k)$.

($\supseteq$)

Sea $k \in pre(\cup_{\alpha\in\Lambda} B_\alpha)$ y supongamos que $\sigma_k \in \Gamma \cap \Sigma_L(k)$. Probaremos que $\sigma_k \in \Sigma_{\cup_{\alpha\in\Lambda} B_\alpha}(k)$.

Como $k \in pre(\cup_{\alpha\in\Lambda} B_\alpha)$, entonces por la proposición 2.1.5 (a), $k \in \cup_{\alpha\in\Lambda} pre(B_\alpha)$. Esto implica que $k \in pre(B_\alpha)$ para algún $\alpha \in \Lambda$. Pero B_α es $_*$-controlable, entonces para toda palabra $k \in pre(B_\alpha)$ existe un $\Gamma \in \mathcal{C}$ tal que $\Gamma \cap \Sigma_L(k) = \Sigma_{B_\alpha}(k)$, es decir, $k\sigma_k \in pre(B_\alpha)$. Por lo tanto, $k\sigma_k \in \cup_{\alpha\in\Lambda} pre(B_\alpha)$ y $\sigma_k \in \cup_{\alpha\in\Lambda} pre(B_\alpha)/k \cap \Sigma$.

Finalmente, por la definición 3.3.1, tenemos que $\sigma_k \in \Sigma_{\cup_{\alpha\in\Lambda} B_\alpha}(k)$. □

PROPOSICIÓN 3.3.4. *Para todo $G = (L, S)$ y todo $_*$-lenguaje $M \subseteq L$ tal que $M \neq \emptyset$, existe un controlador supervisor completo f para G tal que $L^f = M$ si y sólo si M es $_*$-controlable con respecto a L y $_*$-cerrado.*

DEMOSTRACIÓN.

si)

Supongamos que $M \subseteq L$ es $_*$-controlable con respecto a L y $_*$-cerrado. Por $_*$-controlabilidad existe un controlador supervisor $f : pre(M) \to \mathcal{C}$ para $G = (L, S)$ tal que $f(k) = \Gamma_k$, es decir, para toda palabra $k \in pre(M)$ existe un $\Gamma_k \in \mathcal{C}$ tal que:

1) $\Sigma_M(k) \subseteq \Gamma_k$

2) $k\Gamma_k \cap L \subseteq pre(M)$

Mostraremos que $L^f = pre(M)$.

($\subseteq$)

Supongamos que $k\sigma \in L^f$ y probaremos que $k\sigma \in pre(M)$. Como $k\sigma \in L^f$, entonces $k \in L^f \cap dom(f)$, $k\sigma \in L$ y $\sigma \in f(k)$. Asumiendo inductivamente que $k \in pre(M)$ tenemos que $\sigma \in \Sigma_M(k)$. Esto implica que $k\sigma \in pre(M) \cap L$. Por lo tanto, $k\sigma \in pre(M)$.

($\supseteq$)

Supongamos que $k\sigma \in pre(M)$. Entonces, $k\sigma \in L$ ya que $M \subseteq L$. Asumiendo inductivamente que $k \in L^f$, tenemos que $\sigma \in \Sigma_M(k)$ y como $\Sigma_M(k) \subseteq \Gamma_k$, entonces $\sigma \in f(k)$. Por lo tanto, $k\sigma \in L^f$

(sólo si)

Sea f un controlador supervisor tal que $L^f = M$.

Para todo $k \in pre(M)$ mostraremos que

1) $\Sigma_M(k) \subseteq f(k)$

2) $kf(k) \cap L \subseteq pre(M)$

Sea $k\sigma \in pre(M)$. Como $L^f = M$, entonces $k\sigma \in L^f$ y por lo tanto $k \in L^f \cap dom(f)$, $k\sigma \in L$ y $\sigma \in f(k)$. Por ser $k\sigma$ un elemento cualquiera de $pre(M)$, entonces $\Sigma_M(k) \subseteq f(k)$.

Sean $k \in pre(M)$, $\sigma \in f(k)$ y $k\sigma \in L$. Entonces, $k\sigma \in L^f$ y como $L^f = pre(M)$, tenemos que $kf(k) \cap L \subseteq pre(M)$. Por lo tanto, M es $_*$-controlable.

Finalmente, como $M = L^f$ y $L^f = pre(M)$, entonces M es $_*$-cerrado, lo que completa la demostración. □

3.3.2. El ω-comportamiento bajo supervisión y ω-controlabilidad. La $_*$-controlabilidad define la clase de ω-lenguajes que son generados por el sistema bajo la supervisión, como lo demuestra la siguiente proposición que es precisamente la proposición 1.0.3 que analizamos en el capítulo 1.

PROPOSICIÓN 3.3.5. [8]. *Para todo $G = (L, S)$ y todo $T \subseteq S$ tal que $T \neq \emptyset$, existe un controlador supervisor no bloqueante y completo f para G tal que $S^f = T$ si y sólo si*

1. *T es $_*$-controlable con respecto a L y*
2. *T es ω-cerrado con respecto a S.*

DEMOSTRACIÓN.

($\Leftarrow$)

Sea $T \neq \emptyset$ y supongamos que T es $_*$-controlable con respecto a L y ω-cerrado con respecto a S. Como $pre(T)$ es no vacío y $_*$-controlable con respecto a L, existe un controlador supervisor f tal que $L^f = pre(T)$. Por la definición de S^f:

$$\begin{aligned} S^f &= lim(pre(T)) \cap S \\ &= clo(T) \cap S \\ &= T \end{aligned}$$

Finalmente, $pre(S^f) = pre(T) = L^f$ y por lo tanto f es no bloqueante.

($\Rightarrow$)

Supongamos que existe un controlador supervisor no bloqueante f tal que $S^f = T$. Entonces,

$$L^f = pre(S^f) = pre(T)$$

Luego, $pre(T)$ es $_*$-controlable con respecto a L. Si utilizamos la definición de S^f tenemos que

$$\begin{aligned} T &= S^f \\ &= lim(L^f) \cap S \\ &= lim(pre(T)) \cap S \\ &= clo(T) \cap S \end{aligned}$$

Por lo tanto, T es ω-cerrado con respecto a S. □

La proposición anterior de Ramadge garantiza la existencia de supervisores básandose en la $_*$-controlabilidad de ω-lenguajes. A continuación, se extiende esta definición de controlabilidad [**13**].

DEFINICIÓN 3.3.6. (***Prefijos controlables***). Para todo $G = (L, S)$, definimos los *prefijos controlables* $pre_G(T)$ de un ω-lenguaje T como la imagen de T bajo la función $pre_G : 2^s \to 2^{pre(S)}$ dada por la regla

$$\begin{aligned} pre(T) := \{t \in pre(T) :\ & T' \neq \emptyset \text{ , es } * - \text{controlable con respecto a } L/t \text{ y} \\ & \omega - \text{cerrado con respecto a } S/t, \text{ para algún } T' \subseteq T/t\} \end{aligned}$$

El conjunto $pre_G(T)$ son todas las palabras en $pre(T)$ cuyas extensiones infinitas pueden ser controladas para que pertenezcan a T.

LEMA 3.3.7. *Para todo $G = (L, S)$, todo $T \subseteq S$ y todo $t \in pre(T)$*

a) si T es $_*$-controlable con respecto a L, entonces T/t es $_*$-controlable con respecto a L/t

b) si T es ω-cerrado con respecto a S, entonces T/t es ω-cerrado con respecto a S/t.

DEMOSTRACIÓN.

Parte a)

Supongamos que T es $_*$-controlable con respecto a L. Sea $k \in pre(T)$ y sea $l \in pre(T/k) = pre(T)/k$. Por $_*$-controlabilidad existe un $\Gamma \in \mathcal{C}$ tal que $\Gamma \cap \Sigma_L(kl) = \Sigma_T(kl)$, es decir, $\Gamma \cap L/kl = \Sigma \cap pre(T)/kl$. Esto implica que $\Gamma \cap (L/k)l = \Sigma \cap (pre(T)/k)l$ y que $\Gamma \cap (L/k)l = \Sigma \cap (pre(T/k)l)$. Por lo tanto, $\Gamma \cap \Sigma_{L/k}(l) = \Sigma_{pre(T)/k}(l)$.

Como l es un elemento cualquiera de $pre(T/k)$, entonces (T/k) es $_*$-controlable con respecto a L/k.

Parte b)

Supongamos que T es ω-cerrado con respecto a S, entonces se implican las siguientes afirmaciones de manera sucesiva

$T = lim(pre(T)) \cap S$

$\Leftrightarrow T/k = (lim(pre(T)) \cap S)/k$

$\Leftrightarrow T/k = lim(pre(T))/k \cap S/k$ (por la proposición 2.1.9 (f))

$\Leftrightarrow T/k = lim(pre(T)/k) \cap S/k$ (por la proposición 2.1.10 (a))

$\Leftrightarrow T/k = lim(pre(T/k)) \cap S/k$ (por la proposición 2.1.9 (c))

Por lo tanto, T/k es ω-cerrado con respecto a S/k. □

PROPOSICIÓN 3.3.8. *Para cualesquiera $G = (L, S)$, $T, T' \subseteq S$ y $t \in pre(T)$*

a) si $T \subseteq T'$, entonces $pre_G(T) \subseteq pre_G(T')$

b) $pre_{(L/t,S/t)}(T/t) = (pre_{(L,S)}(T))/t$

c) si T es $_*$-controlable con respecto a L y ω-cerrado con respecto a S, entonces

$$pre_G(T) = pre(T)$$

DEMOSTRACIÓN.

Parte a)

Sea $t \in pre_G(T)$ y supongamos que $T \subseteq T'$. Entonces, $t \in pre(T)$ y existe un lenguaje $E' \subseteq T/t$ que no es vacío, $_*$-controlable con respecto a L/t y ω-cerrado con respecto a S/t. Como $t \in pre(T)$ y $T \subseteq T'$, entonces $t \in pre(T')$. Pero $E' \subseteq (T/t) \subseteq (T'/t)$. Por lo tanto, $t \in pre_G(T')$.

Parte b)

$(\subseteq)$

Sea $t' \in (pre_{(L,S)}(T))/t$. Entonces, $tt' \in pre_{(L,S)}(T)$, es decir, $tt' \in pre(T)$ y existe $T' \subseteq T/tt$ no vació tal que T' es $_*$-controlable con respecto a L/tt' y ω-cerrado con respecto a S/tt'. Luego, $t' \in pre(T/t)$ y existe $T' \subseteq (T/t)/t'$ tal que $T' \neq \emptyset$, $_*$-controlable con respecto a $(L/t)/t'$ y ω-cerrado con respecto a $(S/t)/t'$. Por lo tanto, $t \in pre_{(L/t,S/t)}(T/t)$.

$(\supseteq)$

La inclusión recíproca se prueba de manera análoga.

Parte c)

Supongamos que T es $_*$-controlable con respecto a L y ω-cerrado con respecto a S. Por el lema 3.3.7, T/t es $_*$-controlable con respecto a L/t y ω-cerrado con respecto a S/t. Por ser k un elemento cualquiera de $pre(T)$, entonces $pre(T) = pre_G(T)$. □

DEFINICIÓN 3.3.9. (ω-***controlabilidad***). Para todo SED $G = (L, S)$ y todo $T \subseteq S$, T es ω-controlable con respecto a G si T es $_*$-controlable con respecto a L y $pre(T) = pre_G(T)$.

La ω-controlabilidad es preservada bajo la unión de conjuntos, como veremos a continuación.

PROPOSICIÓN 3.3.10. *La ω-controlabilidad es preservada bajo la unión arbitraria de conjuntos*

DEMOSTRACIÓN. Sea $\{B_\alpha\}_{\alpha\in\Lambda}$ una familia de lenguajes en Σ^ω y supongamos que R_α es ω-controlable para todo $\alpha \in \Lambda$.

Probaremos que $pre_G(\cup_\alpha R_\alpha) = pre(\cup_\alpha R_\alpha)$.

($\supseteq$)

Sea $t \in pre(\cup_\alpha R_\alpha)$. Entonces, existe un $t' \in \cup_\alpha R_\alpha$ tal que $t \leq t'$, es decir, $t' \in R_\alpha$ para algún α. Por hipótesis $pre_G(R_\alpha) = pre(R_\alpha)$, para todo $\alpha \in \Lambda$. Luego, $t \in pre_G(R_\alpha)$. Por otra parte, sabemos que $R_\alpha \subseteq \cup_\alpha R_\alpha$. Por lo tanto, por la proposición 3.3.8 (a), tenemos que $pre_G(R_\alpha) \subseteq pre_G(\cup_\alpha R_\alpha)$. Esto prueba que $t \in pre_G(\cup_\alpha R_\alpha)$.

($\subseteq$)

La parte $pre_G(\cup_\alpha B_\alpha) \subseteq pre(\cup_\alpha B_\alpha)$ se obtiene directamente de la definición 3.3.6.

Finalmente, por la proposición 3.3.3, tenemos que la unión de conjuntos $_*$-controlables es $_*$-controlable, lo que completa la demostración. □

PROPOSICIÓN 3.3.11. *Para todo $G = (L, S)$ y todo $T \subseteq S$, si T es ω-cerrado con respecto a S, entonces T es ω-controlable con respecto a (L, S) si y sólo si T es $_*$-controlable con respecto a L.*

DEMOSTRACIÓN.

($\Rightarrow$)

Se demuestra directamente por la definición de ω-controlabilidad.

($\Leftarrow$)

Supongamos que T es $_*$-controlable con respecto a L. Como T es también ω-cerrado con respecto a S, entonces por la proposición 3.3.8 (c), tenemos que $pre(T) = pre_G(T)$. Por lo tanto, T es ω-controlable. □

PROPOSICIÓN 3.3.12. [13]. *Para todo $G = (L, S)$ y todo $T \subseteq S$ existe un controlador supervisor no bloqueante y completo f tal que $S^f = T$ si y sólo si*

1. *T es ω-controlable con respecto a G y*
2. *T es ω-cerrado con respecto a S.*

DEMOSTRACIÓN. Se demuestra directamente por la proposición 1.0.3 y definición 3.3.9. □

Aunque aparentemente la proposición 3.3.12, de Thistle, es más restrictiva que la proposición 3.3.5, de Ramadge, no es así. Esto se debe a la proposición 3.3.7. Esta proposición junto con la definición 3.3.6, nos permite reemplazar la $_*$-controlabilidad de la proposición 3.3.5, por la ω-controlabilidad. La ω-controlabilidad reemplazada por la $_*$-controlabilidad nos permite garantizar la existencia de supervisores aunque el lenguaje de especificación no sea ω-cerrado, como veremos en la siguiente sección.

3.4. Síntesis del Supervisor

El problema del control supervisor de [7], que denotamos como PCS^*, es la base para resolver el problema del control supervisor para ω-lenguajes (PCS^ω).

3.4.1. Problema del control supervisor para ω-lenguajes (PCS^ω). El PCS^ω consiste en lo siguiente:

PROBLEMA 3.4.1. Dado un $G = (L, S)$ y dados los ω-lenguajes $A \subseteq E \subseteq S$ construir un controlador supervisor no bloqueante y completo f para G tal que $A \subseteq S^f \subseteq E$

Por el teorema 3.3.12, la solución del PCS^ω es equivalente a la existencia de un sublenguaje no vacío $T \subseteq S$, ω-cerrado con respecto a S y ω-controlable con respecto a G, tal que $A \subseteq T \subseteq E$.

Los lenguajes ω-controlables y ω-cerrados tienen diferentes propiedades de cerradura bajo la unión e intersección. Específicamente ω-controlabilidad es preservada bajo la unión arbitraria de conjuntos (véase la proposición 3.3.10) pero no por la intersección como lo demuestra el ejemplo 3.4.3. En cambio, la ω-cerradura es preservada bajo la intersección arbitraria de conjuntos (véase la proposición 2.1.14) pero no por la unión como lo demuestra el ejemplo 3.4.2.

EJEMPLO 3.4.2. [8]. Sea $R = a^*ba^\omega$ Para todo $n \geq 1$, el ω-lenguaje $R_n = \{a^kba^\omega, 0 \leq k \leq n\}$ es ω-cerrado. Pero $\bigcup_{n=0}^{\infty} R_n = a^*ba^\omega = R$.

Como R no es ω-cerrado, se deduce que la familia de sublenguajes ω-cerrados de R no es ω-cerrado bajo la unión contable de conjuntos.

EJEMPLO 3.4.3. La ω-controlabilidad puede no ser preservada bajo la intersección de conjuntos.

Sean $S = c^\omega + (c^*a(b+c))^\omega$, $L = pre(S) = c^* + (c^*a(b+c))^*$ y $\mathcal{C} = \{\Gamma \subseteq \Sigma : a \in \Gamma\}$. Los lenguajes $R_1 = c^\omega + (c^*ab)^\omega$ y $R_2 = c^\omega + (c^*ac)^\omega$ son ω-controlables, pero $R_1 \cap R_2 = c^\omega$ no es $*$-controlable, ya que $\Gamma \cap \Sigma_L(c) \neq \Sigma_{(R_1 \cap R_2)}(c)$. Por la definición 3.3.9, puede verse que $R_1 \cap R_2$ no es ω-controlable.

Debido a que la ω-cerradura y la ω-controlabilidad tienen diferentes propiedades bajo la unión e intersección, para definir la solución al PCS^ω es conveniente definir dos sublenguajes separados de la siguiente manera:

DEFINICIÓN 3.4.4. Para todo SED $G = (L, S)$ y cualesquiera ω-lenguajes $A \subseteq T \subseteq E$,

$C^\omega(E) := \{T \subseteq S : T \subseteq E \subseteq S$ y T es ω-controlable con respecto a $G\}$

$F^\omega(A) := \{T \subseteq S : A \subseteq T \subseteq S$ y T es ω-cerrado con respecto a $S\}$

Decimos que un ω-lenguaje es la solución del PCS^ω si no es vacío y pertenece a $C^\omega(E) \cap F^\omega(A)$.

El conjunto $C^\omega(E)$ contiene un único elemento supremo, que denotamos $supC^\omega(E)$. Para definir y demostrar la existencia del elemento supremo $supC^\omega(E)$, necesitamos definir primero el PCS^*.

Por la proposición 3.3.4, el PCS^* es equivalente a la existencia de un sublenguaje no vacío $K \subseteq L$, $_*$-controlable con respecto a L y $_*$-cerrado, tal que $K \subseteq L'$. Entonces, definimos la siguiente clase de lenguajes:

DEFINICIÓN 3.4.5. Para todo $_*$-lenguaje L y todo $L' \subseteq L$,
$CF^*(L') := \{K \subseteq L' \mid K$ es $_*$-controlable con respecto a L y $_*$-cerrado$\}$

El conjunto $CF^*(L')$ tiene un único conjunto supremo, que denotamos $supCF^*(L')$, como lo demuestra la siguiente proposición.

PROPOSICIÓN 3.4.6. *Para todo $_*$-lenguaje L y todo $L' \subseteq L$, el conjunto supremo $supCF^*(L')$ existe en $CF^*(L')$.*

DEMOSTRACIÓN. $CF^*(L')$ no es vacío porque $\{\emptyset\}$ pertenece a él, ya que $\emptyset$ es $_*$-controlable y $pre(\emptyset) = \emptyset$.

Por la proposición 3.3.3, tenemos que la $_*$-controlabilidad es cerrada bajo la unión arbitraria de conjuntos.

Ahora probaremos que $_*$-cerradura es preservada bajo la unión de conjuntos. Sea $\{K_\alpha\}_{\alpha\in\Lambda}$ una familia de lenguajes en Σ^* y supongamos que para todo $\alpha \in \Lambda$ se cumple que $K_\alpha = pre(K_\alpha)$. Entonces,

$$\begin{aligned} \cup_\alpha K_\alpha &= \cup_\alpha pre(K_\alpha) \\ &= pre(\cup_\alpha K_\alpha) \qquad \text{(por la proposición 2.1.5 (a))} \end{aligned}$$

Esto prueba que la $_*$-cerradura es preservada bajo la unión arbitraria de conjuntos.

Por todo lo anterior, el elemento supremo $supCF^*(L') = \cup\{K \mid K \in CF^*(L')\}$, lo que completa la demostración. □

Si el PCS^* tiene solución, entonces $supCF^*(L')$ es la única solución maximal.

El siguiente lema lo utilizaremos más adelante para demostrar la existencia del elemento supremo $supC^\omega(E)$ y hacer la conexión entre PCS^* y PCS^ω.

LEMA 3.4.7. **[13]**. *Para todo $G = (L, S)$ y todo $E \subseteq \Sigma^\infty$ es cierto que*

$$supCF^*(pre_G(E)) \subseteq pre_G(lim(supCF^*(pre_G(E))) \cap E)$$

DEMOSTRACIÓN. Sea $E' := lim(supCF^*(pre_G(E))) \cap E$ y supongamos que $k \in supCF^*(pre_G(E)) \subseteq pre_G(E)$. Entonces, existe un lenguaje $E_k \subseteq E/k$ no vacío, $_*$-controlable con respecto a L/k y ω-cerrado con respecto a S/k. Ya que k es un elemento cualquiera de $pre_G(E)$ es suficiente mostrar que $E_k \subseteq E'/k$.

Sea $k' \in pre(E_k)$. Entonces, $E_k/k' \subseteq E/kk'$ no es vacío, $_*$-controlable con respecto a L/kk' y ω-cerrado con respecto a S/kk'. Luego, por la definición 2.1.3, tenemos que $kk' \in pre_G(E)$. Como k' es un elemento cualquiera de $pre(E_k)$, significa que $kpre(E_k) \subseteq pre_G(E)$.

Como $E_k \subseteq E/k$ es $_*$-controlable con respecto a L/k y $k \in supCF^*(pre_G(E))$, entonces $kpre(E_k)$ es $_*$-controlable con respecto a L y $_*$-cerrado. Esto significa que $kpre(E_k) \subseteq supCF^*(pre_G(E))$, debido a que $kpre(E_k) \subseteq pre_G(E)$. Por la proposición 2.1.10 (a), tenemos que $klim(pre(E_k)) \subseteq lim(supCF^*(pre_G(E)))$. Es decir, $kclo(E_k) \cap S/k \subseteq lim(supCF^*(pre_G(E)))$. Como $k \in supCF^*(pre_G(E))$ y E_k es ω-cerrado con respecto a S/k, entonces $k(E_k) \subseteq lim(supCF^*(pre_G(E))) \cap E$ y por lo tanto $E_k \subseteq E'/k$. Esto completa la demostración. $\square$

PROPOSICIÓN 3.4.8. [13]. *Para todo SED $G = (L, S)$ y cualesquiera ω-lenguajes $A \subseteq E \subseteq S$,*

a) $supC^\omega(E)$ existe en $C^\omega(E)$ tal que

$$supC^\omega(E) = lim(supCF^*(pre_G(E))) \cap E$$

b) $inf\, F^\omega(A)$ existe en $F^\omega(A)$ tal que

$$inf\, F^\omega(A) = clo(A) \cap S$$

DEMOSTRACIÓN.

Parte a)

Sea $E' := lim(supCF^*(pre_G(E))) \cap E$. Entonces,

$$\begin{aligned} pre(E') &\subseteq pre(lim(supCF^*(pre_G(E)))) \cap pre(E) && \text{(por la proposición 2.1.5 }(b)\text{)} \\ &\subseteq supCF^*(pre_G(E)) && \text{(por la proposición 2.1.9 (a))} \\ &\subseteq pre_G(E') && \text{(por el lema 3.4.7)} \\ &\subseteq pre(E') && \text{(por la definición de } pre_G\text{)} \end{aligned}$$

Esto implica que $pre(E') = supCF^*(pre_G(E)) = pre_G(E')$. Por lo tanto, E' es ω-controlable con respecto a G.

Supongamos ahora que $E'' \in C^\omega(E)$. Entonces,

$$
\begin{aligned}
E'' &\subseteq clo(E'') \cap E && \text{(por la proposición 2.1.13}(a)) \\
&= lim(pre(E'')) \cap E && \text{(por la definición de clo)} \\
&= lim(supCF^*(preE'')) \cap E && \text{(por } *-controlabilidad) \\
&= lim(supCF^*(pre_G(E''))) \cap E && \text{(por } \omega - controlabilidad) \\
&\subseteq E' && (E'' \subseteq E)
\end{aligned}
$$

Por lo tanto, E' es el único elemento maximal en $C^\omega(E)$, lo que completa la demostración.

Parte b)

($\subseteq$)

Por la proposición 2.1.13 (a), tenemos que $clo(A) \cap S \subseteq clo(clo(A) \cap S) \cap S$.

($\supseteq$)

$$
\begin{aligned}
clo(clo(A) \cap S) \cap S &= lim(pre(clo(A) \cap S)) \cap S && \text{(por la definición de clo)} \\
&\subseteq lim(pre(clo(A)) \cap pre(S)) \cap S && \text{(por la proposición 2.1.5 }(b)) \\
&= lim(pre(clo(A))) \cap lim(pre(s)) \cap S && \text{(por la proposición 2.1.8 }(b)) \\
&\subseteq lim(pre(clo(A))) \cap S \\
&\subseteq clo(clo(A)) \cap S \\
&= clo(A) \cap S && \text{(por la proposición 2.1.13 }(a))
\end{aligned}
$$

Esto prueba que $clo(A) \cap S$ es ω-cerrado con respecto a S.

Sea ahora $A' \in F^\omega(A)$. Entonces,

$$
\begin{aligned}
A' &= clo(A') \cap S && \text{(por la definición de } F^\omega(A)) \\
&\supseteq clo(A) \cap S && (A \subseteq A' \text{ y clo es monótona})
\end{aligned}
$$

Por lo tanto, A' es el único elemento minimal en $F^\omega(A)$, lo que completa la demostración. □

La proposición anterior muestra que el $supC^\omega$ se deriva a partir de $supCF^*$ Además prueba que $supC^\omega(E)$ es el mayor sublenguaje de E que es ω-controlable con respecto a G y que $infF^\omega(A)$ es el menor superlenguaje de A que es ω-cerrado con respecto a S.

LEMA 3.4.9. *Sean* $A' := inf\, F^\omega(A)$, $E' := sup C^\omega(E)$ *tal que* $A \subseteq E \subseteq \Sigma^\omega$ *y sea* M *el conjunto de todos los elementos de* $pre(E') \setminus pre(A')$ *de longitud mínima. Entonces,*

a) $L^f = pre(A') \cup \bigcup_{m \in M} mpre(E'_m)$,

b) $S^f = A' \cup \bigcup_{m \in M} mE'_m$

DEMOSTRACIÓN.

Parte a)

Probaremos por inducción sobre la longitud de las palabras.

Caso base:

$$\begin{aligned} 1 \in L^f &\Leftrightarrow 1 \in pre(A') \text{ o } 1 \in \textstyle\bigcup_{m \in M} mpre(E'_m) \\ &\Leftrightarrow 1 \in pre(A') \cup \textstyle\bigcup_{m \in M} mpre(E'_m) \end{aligned}$$

Inducción.

$k\sigma \in L^f$

$\Leftrightarrow k\sigma \in L,\ k \in L^f$ y $\sigma \in f(k)$

$\Leftrightarrow k\sigma \in L,\ k \in pre(A') \cup \bigcup_{m \in M} mpre(E'_m)$ y $\sigma \in f(k)$ (hipótesis de inducción)

$\Leftrightarrow k\sigma \in L,\ [k \in pre(A')$ y $\sigma \in f_0(k)$ o $(\exists m \in M)(k \in mpre(E'_m)$ y $\sigma \in f_m(k/m))]$

$\Leftrightarrow k\sigma \in L,\ [k \in L^{f_0} \cap pre(A')$ y $\sigma \in f_0(k)$ o $(\exists m \in M)(k \in m(L/m)^{f_m}$ y $\sigma \in f_m(k/m))]$ (f_m es no bloqueante)

$\Leftrightarrow k\sigma \in L,\ [k\sigma \in L^{f_0} \cap pre(A')\Sigma$ o $(\exists m \in M)(k\sigma/m \in (L/m)^{f_m})]$

$\Leftrightarrow k\sigma \in L,\ [k\sigma \in pre(A')$ o $(\exists m \in M)(k\sigma/m \in pre(E'_m))]$ (f_0 y f_m son no bloqueantes)

$\Leftrightarrow k\sigma \in pre(A') \cup \bigcup_{m \in M} mpre(E'_m)$.

Parte b)

Primero veamos que

$$lim(L^f) = clo(A') \cup \bigcup_{m \in M} mclo(E'_m)$$

($\supseteq$)

$$\begin{aligned} lim(L^f) &= lim(pre(A') \cup \textstyle\bigcup_{m \in M} mpre(E'_m)) \qquad \text{(parte } (a)) \\ &\supseteq lim(pre(A')) \cup lim(\textstyle\bigcup_{m \in M} mpre(E'_m)) \\ &\supseteq clo(A') \cup \textstyle\bigcup_{m \in M} m\, lim(pre(E'_m)) \\ &= clo(A') \cup \textstyle\bigcup_{m \in M} mclo(E'_m)) \end{aligned}$$

($\subseteq$)

Supongamos que $s \in lim(L^f)$. Entonces,

$$pre(s) \subseteq L^f = pre(A') \cup \bigcup_{m \in M} mpre(E'_m).$$

Si $pre(s) \subseteq pre(A')$, entonces $s \in clo(A')$ y la inclusión es verdadera. Ahora bien, si $pre(s) \subseteq mpre(E'_m)$, donde $m \in M$ es el elemento más pequeño de $pre(s) \setminus pre(A')$. Entonces, $s \in mclo(E'_m)$. □

Ahora podemos definir la solución al PCS^ω en términos de $inf\, F^\omega(A)$ y $supC^\omega(E)$.

TEOREMA 3.4.10. [13] *El* PCS^ω *tiene solución si y sólo si* $supC^\omega(E) \neq \emptyset$ *y* $inf\, F^\omega(A) \subseteq supC^\omega(E)$.

DEMOSTRACIÓN.

($\Rightarrow$)

Supongamos que el PCS^ω tiene solución. Entonces, por proposición 3.3.12, existe un controlador supervisor no bloqueante y completo tal que $A \subseteq S^f \subseteq E$, es decir, $S^f \in F^\omega(A)$ y $S^f \in C^\omega(E)$. Por lo tanto, $inf F^\omega(A) \subseteq S^f \subseteq supC^\omega(E)$.

($\Leftarrow$)

Sea $A' := inf\, F^\omega(A)$, sea $E' := supC^\omega(E)$ y supongamos que $A' \subseteq E'$ y $E' \neq \emptyset$. Como E' es $_*$-controlable y no vacío, entonces existe un controlador supervisor no bloqueante y completo $f_0 : L \to \mathcal{C}$ que sintetiza $clo(E') \cap S$ por proposición 1.0.3.

Como E' es ω-controlable, entonces para todo $m \in pre(E')$ existe un ω-lenguaje $E'_m \subseteq E'/m$ no vacío, $_*$-controlable con respecto a L/m y ω-cerrado con respecto a S/m. Sea $f_m : L \to \mathcal{C}$ el correspondiente controlador supervisor para E'_m.

Sea M el conjunto de todos los elementos de $pre(E') \setminus pre(A')$ de longitud mínima. Definimos el controlador supervisor $f : L \to \mathcal{C}$:

$$f(s) := \begin{cases} f_0(s) & \text{si } s \in pre(A') \\ f_m(s/m) & \text{si } s \in mpre(E'_m), \text{ donde } m \in M \\ indefinida & \text{en otro caso} \end{cases}$$

Decimos que f es un controlador supervisor no bloqueante y completo para $G = (L, S)$ y que $A' \subseteq S^f \subseteq E'$ Por lo tanto,

$$
\begin{aligned}
S^f &= lim(L^f) \cap S \\
&= (clo(A') \cup \bigcup_{m \in M} mclo(E'_m)) \cap S \qquad \text{(por el lema 3.4.9)} \\
&= (clo(A') \cap S) \cup \bigcup_{m \in M} (mclo(E'_m) \cap S) \\
&= (clo(A') \cap S) \cup \bigcup_{m \in M} m(clo(E'_m) \cap S/m) \\
&= A' \cup \bigcup_{m \in M} mE'_m
\end{aligned}
$$

Obtenemos que f es completo por la parte (a) y $A' \subseteq S^f \subseteq E'_m$ por la parte (b).

Mostraremos que f es no bloqueante.

$$
\begin{aligned}
pre(S^f) &= pre(A' \cup \bigcup_{m \in M} mE'_m) \qquad \text{(por el lema 3.4.9 } (b)) \\
&= pre(A') \cup \bigcup_{m \in M} mpre(E'_m) \qquad \text{(por la proposición 2.1.5 } (a)) \qquad \square \\
&= L^f
\end{aligned}
$$

El supervisor definido en la prueba de la proposición anterior, aplica la función f_0 hasta la ocurrencia de una palabra en el conjunto $pre(E') \backslash pre(A')$, de ahí en adelante siempre aplica la función f_m.

COROLARIO 3.4.11. [13]. *Para todo SED $G = (L, S)$ y todo $E \subseteq S$, el ω-lenguaje $supC^\omega(E)$ es el comportamiento superior máximo controlable contenido en E; eso es,*

$$
supC^\omega(E) := \bigcup \{R \subseteq E : R \text{ es }_*\text{-controlable con respecto a } L \text{ y } \omega\text{-controlable con respecto a } S\}
$$

DEMOSTRACIÓN.

($\supseteq$)

Se demuestra por la proposición 3.3.11.

($\subseteq$)

Sea $s \in supC^\omega(E)$ y definimos $A := \{s\}$. Entonces, por el teorema 3.4.10, existe un $R \subseteq supC^\omega(E)$, $_*$-controlable con respecto a L y ω-cerrado con respecto a S, tal que $A \subseteq R \subseteq E$. $\square$

COROLARIO 3.4.12. [13]. *Cuando el PCS^ω tiene solución, tiene soluciones maximales si y sólo si $supC^\omega(E)$ es ω-cerrado con respecto a S, esto es, si y sólo si $supC^\omega(E) \in F^\omega(A)$. En ese caso, $supC^\omega(E)$ es la única solución maximal.*

DEMOSTRACIÓN. Se demuestra por el corolario 3.4.11, ya que $C^w(E) \cap F^w(A)$ es cerrado bajo la unión finita de conjuntos. $\square$

COROLARIO 3.4.13. [13]. *El $supC^w(E)$ es ω-cerrado con respecto a S siempre que E sea ω-cerrado.*

DEMOSTRACIÓN. Se demuestra por la proposición 3.4.8. □

De acuerdo con el corolario 3.4.12, si $supC^w(E)$ es ω-cerrado con respecto a S, en caso de haber solución al PCS^ω, el ω-comportamiento generado bajo la supervisión es exactamente igual al lenguaje $supC^w(E)$. En caso contrario, el ω-comportamiento generado bajo la supervisión es un subconjunto, ω-cerrado, de $supC^w(E)$.

3.5. Control Supervisor Modular

En esta sección extendemos la teoría de control supervisor de las secciones anteriores para la síntesis modular de supervisores. En esta sección las tareas se dividen en varias subtareas y el objetivo es derivar un supervisor para cada tarea específica. A este tipo de construcción le llamamos ***síntesis modular*** y al supervisor resultante lo llamamos *supervisor modular* [15]. Una de las ventajas de este tipo de síntesis es que si una tarea del proceso cambia, no es necesario reemplazar todo el supervisor sino solamente el que se encarga de supervisar dicha tarea.

Puesto que el conjunto de patrones de control C es cerrado bajo la unión e intersección de conjuntos, la familia de supervisores para un sistema G es cerrado bajo las siguientes operaciones de disyunción y conjunción:

$$(f \wedge g)(k) = f(k) \cap g(k) \tag{3.5.1}$$

$$(f \vee g)(k) = f(k) \cup g(k) \tag{3.5.2}$$

3.5.1. Conjunción de supervisores. Sean f y g supervisores completos. La *conjunción de supervisores* f y g, la denotamos como $f \wedge g$.

PROPOSICIÓN 3.5.1. *Sean f y g supervisores completos y sea $f \wedge g$ la conjunción de f y g. Entonces,*

a) $L^{f\wedge g} = L^f \cap L^g$

b) $S^{f\wedge g} = S^f \cap S^g$

c) $f \wedge g$ *es un supervisor completo.*

DEMOSTRACIÓN.

Parte a)

Véase [15].

Parte b)

$$\begin{aligned} S^{f\wedge g} &= lim(L^{f\wedge g}) \cap S \\ &= lim(L^f \cap L^g) \cap S & \text{(parte (a))} \\ &= lim(L^f) \cap lim(L^g) \cap S & \text{(por la prosición 2.1.8 }(b)) \\ &= S^f \cap S^g \end{aligned}$$

Parte c)

Sean $k \in L^{f\wedge g}$, $\sigma \in \Sigma$, $k\sigma \in L$ y $\sigma \in f \wedge g$. Entonces, $k \in L^f$ y $k \in L^g$. Como f es un supervisor completo, entonces $k\sigma \in L^f$ y por la misma razón $k\sigma \in L^g$. Por lo tanto, por la parte (a), tenemos que $k\sigma \in L^{f\wedge g}$. □

La propiedad de no bloqueo no siempre se preserva bajo la conjunción de supervisores, es decir, si dos supervisores f y g son no bloqueantes, la conjunción $f \wedge g$ no necesariamente preserva esta propiedad.

DEFINICIÓN 3.5.2. (*$_*$-Lenguajes no conflictivos*) [15]. Dados los lenguajes $A, B \subseteq \Sigma^\infty$, decimos que A y B son lenguajes *no conflictivos* siempre que

$$pre(A) \cap pre(B) = pre(A \cap B)$$

Debido a que siempre $pre(A \cap B) \subseteq pre(A) \cap pre(B)$, dos lenguajes no son conflictivos si la intersección de los prefijos de ambos están contenidos en el prefijo de la intersección de los lenguajes.

La propiedad 3.5.2 determina el no bloqueo de un supervisor $f \wedge g$, como veremos a continuación.

PROPOSICIÓN 3.5.3. *Sea $G = (L, S)$ un DES, sean f y g supervisores no bloqueantes y completos. Si S^f y S^g no son conflictivos, entonces el supervisor $f \wedge g$ es no bloqueante y completo.*

DEMOSTRACIÓN. El supervisor $f \wedge g$ es completo por la proposición 3.5.1 (d).

Probaremos ahora que $f \wedge g$ es no bloqueante.

$(\supseteq)$

$$
\begin{aligned}
L^{f\wedge g} &= L^f \cap L^g \\
&= pre(S^f) \cap pre(S^g) \\
&\supseteq pre(S^f \cap S^g) \qquad \text{(por la proposición 2.1.5 } (b)) \\
&= pre(S^{f\wedge g})
\end{aligned}
$$

$(\subseteq)$

Sea $k \in L^{f\wedge g}$. Probaremos que $k \in pre(S^{f\wedge g})$.

Debido a que $k \in L^{f\wedge g}$, $k \in L^f$, $k \in L^g$ y también a que f y g son no bloqueantes, entonces $k \in pre(S^f)$ y $k \in pre(S^g)$. Por lo tanto $k \in pre(S^{f\wedge g})$, ya que S^f y S^g no son lenguajes conflictivos. □

3.5.2. Problema del control supervisor modular utilizando conjunción ($PCSMC^\omega$). El $PCSMC^\omega$ consiste en lo siguiente:

PROBLEMA 3.5.4. Dados un DES $G = (L, S)$ y los lenguajes S^{f_1}, S^{f_2}, ... ,S^{f_n}, construir un controlador supervisor no bloqueante y completo g tal que

$$S^g = \bigcap_{i=1}^{n} S^{f_i}$$

El $PCSMC^\omega$ tiene solución si y sólo si $S^g = \bigcap_{i=1}^{n} S^{f_i}$ es $\omega-$controlable con respecto a G y $\omega-$cerrado con respecto a S. Para determinar si existe una solución al $PCSMC^\omega$ es suficiente considerar el caso $n = 2$.

Como vimos en el ejemplo 3.4.3, la $\omega-$controlabilidad puede no ser preservada bajo la intersección de conjuntos, por lo tanto nos enfocaremos en determinar bajo qué condiciones sí se puede preservar dicha propiedad.

Primero probaremos que la conjunción de lenguajes $_* -$ controlables no conflictivos siempre es $_* -$ controlable [15].

PROPOSICIÓN 3.5.5. *Sean $T_1, T_2 \subseteq \Sigma^\omega$ y sea $L \subseteq \Sigma^*$ tal que $pre(T_1) \subseteq L$ y $pre(T_2) \subseteq L$. Si los $\omega-$lenguajes T_1 y T_2 son $_*$-controlables y no conflictivos, entonces $T_1 \cap T_2$ es $*$-controlable con respecto a L.*

DEMOSTRACIÓN. Supongamos que T_1 y T_2 son $_*$-controlables y no conflictivos.

Probaremos que para toda $k \in pre(T_1 \cap T_2)$ existe un $\Gamma \in \mathcal{C}$ tal que

$$\Gamma \cap \Sigma_L(k) = \Sigma_{T_1 \cap T_2}(k)$$

($\supseteq$)

Sean $k \in pre(T_1 \cap T_2)$ y $\sigma_k \in \Sigma_{T_1 \cap T_2}(k)$. Probaremos que existe un $\Gamma \in \mathcal{C}$ tal que $\sigma_k \in \Gamma \cap \Sigma_L(k)$.

Como $k \in pre(T_1 \cap T_2)$ y $\sigma_k \in \Sigma_{T_1 \cap T_2}(k)$, entonces $k\sigma_k \in pre(T_1 \cap T_2)$, es decir, $k\sigma_k \in pre(T_1)$ y $k\sigma_k \in pre(T_2)$. Debido a que T_1 y T_2 son $_* -$ controlables, existe un $\Gamma \in \mathcal{C}$ tal que $\Gamma \cap \Sigma_L(k) = \Sigma_{T_1}(k)$ y $\Gamma \cap \Sigma_L(k) = \Sigma_{T_2}(k)$. Por lo tanto, tenemos que $\sigma_k \in \Gamma \cap \Sigma_L(k)$, ya que $k \leq k\sigma_k$.

($\subseteq$)

Sea $k \in pre(T_1 \cap T_2)$ y supongamos que $\sigma_k \in \Sigma_L(k)$ para algún $\Gamma \in \mathcal{C}$. Probaremos que $\sigma_k \in \Sigma_{T_1 \cap T_2}(k)$.

Como $k \in pre(T_1 \cap T_2)$, existe $t \in T_1 \cap T_2$ tal que $k \leq t$. Esto implica que $t \in T_1$ y que $t \in T_2$. Luego, $k \in pre(T_1)$ y $k \in pre(T_2)$. Debido a que T_1 y T_2 son $_*-$controlables, $\Gamma \cap \Sigma_L(k) = \Sigma_{T_1}(k)$ y $\Gamma \cap \Sigma_L(k) = \Sigma_{T_2}(k)$, es decir, $\sigma_k \in \Sigma_{T_1}(k)$ y $\sigma_k \in \Sigma_{T_2}(k)$. Esto significa que $k\sigma_k \in pre(T_1)$ y que $k\sigma_k \in pre(T_2)$. Por hipótesis tenemos que T_1 y T_2 no son lenguajes conflictivos, por lo tanto $k\sigma_k \in pre(T_1 \cap T_2)$ y $\sigma_k \in pre(T_1 \cap T_2)/k$. Esto prueba que $\sigma_k \in \Sigma_{T_1 \cap T_2}(k)$. □

PROPOSICIÓN 3.5.6. *Sea $G = (L, S)$ un DES y sean $T_1, T_2 \subseteq \Sigma^\omega$ Entonces*

$$pre(T_1 \cap T_2) = pre_G(T_1 \cap T_2)$$

DEMOSTRACIÓN.

($\supseteq$)

Tenemos que $pre_G(T_1 \cap T_2) \subseteq pre(T_1 \cap T_2)$ por definición de pre_G.

($\subseteq$)

Sea $k \in pre(T_1 \cap T_2)$ y probaremos que $k \in pre_G(T_1 \cap T_2)$.

Como $k \in pre(T_1 \cap T_2)$, existe una palabra $t \in T_1 \cap T_2$ tal que $k \leq t$. Esto implica que $t \in T_1$ y que $t \in T_2$. Luego, $k \in pre(T_1)$ y $k \in pre(T_2)$. Por hipótesis $pre(T_1) = pre_G(T_1)$ y también $pre(T_2) = pre_G(T_2)$. Esto implica que $k \in pre_G(T_1)$ y que $k \in pre_G(T_2)$.

Como $k \in pre_G(T_1)$, existe un $T'_1 \subseteq T_1/k$ no vacío, $_*$-controlable con respecto a L/k y $\omega -$ cerrado con respecto a S/k. Lo mismo sucede con $k \in pre_G(T_2)$. Entonces, por la hipótesis y la proposición 3.5.5, $(T'_1 \cap T'_2)$ es $_*$-controlable con respecto a L/k; y es $\omega -$ cerrado con respecto a S/k, por la proposición 2.1.14.

Probaremos ahora que $(T'_1 \cap T'_2)$ no es vacío.

Sea $t \in T'_1$ y probaremos que $t \in T'_2$.

Supongamos que no es así, entonces $t \notin T'_2$. Luego, $\{t\}$ no es $_\star$-controlable con respecto a L/k o no es $\omega -$ cerrado con respecto a S/k, pero esto contradice el hecho de que $t \in T'_1$. Para el caso de que $t \in T'_2$, entonces también $t \in T'_1$. Esto prueba que $(T'_1 \cap T'_2)$ no es vacío. Por lo tanto, $k \in pre_G(T_1 \cap T_2)$, lo que completa la demostración.

□

De los resultados anteriores podemos deducir que la ω-controlabilidad de lenguajes también es preservada bajo la intersección de conjuntos siempre que los lenguajes sean no conflictivos, como lo demostramos a continuación.

PROPOSICIÓN 3.5.7. *Sea $G = (L, S)$ un DES y sean $T_1, T_2 \subseteq S$. Si los lenguajes T_1 y T_2 son ω-controlables con respecto a G y no conflictivos, entonces $T_1 \cap T_2$ es ω-controlable con respecto a G.*

DEMOSTRACIÓN. Supongamos que T_1 y T_2 son ω-controlables y no conflictivos. Entonces, por la hipótesis tenemos que T_1 y T_2 son $\quad -$ controlables con respecto a L. Ahora bien, por la proposición 3.5.5, resulta que $T_1 \cap T_2$ es $_\star$-controlable con respecto a L.

Por la hipótesis y la proposición 3.5.6, tenemos que $pre(T_1 \cap T_2) = pre_G(T_1 \cap T_2)$, lo que prueba que $T_1 \cap T_2$ es ω-controlable con respecto a G.

□

El resultado anterior nos lleva a la siguiente conclusión.

PROPOSICIÓN 3.5.8. *Sea $G = (L, S)$ un DES y sean $S^f, S^g \subseteq S$. Entonces el $PCSMC^\omega$ tiene solución siempre que los $\omega-$lenguajes S^f y S^g no sean conflictivos y $S^{f \wedge g} \neq \emptyset$.*

DEMOSTRACIÓN. Supongamos que $S^{f \wedge g} \neq \emptyset$ y que los $\omega-$lenguajes S^f y S^g no son conflictivos. Entonces, por la proposición 3.5.7, $S^{f \wedge g}$ es $\omega -$ controlable y por la proposición 2.1.14, tenemos que $S^{f \wedge g}$ es $\omega -$ cerrado, lo que completa la demostración.

□

3.5.3. Disyunción de supervisores. Sean f y g supervisores completos. La *disyunción de supervisores* f y g, la denotamos como $f \vee g$.

PROPOSICIÓN 3.5.9. *Sean f y g supervisores completos y no bloqueantes; y sea $f \vee g$ la disyunción de f y g. Entonces,*

a) $L^{f \vee g} = L^f \cup L^g$

b) $S^{f \vee g} = S^f \cup S^g$

c) $f \vee g$ *es un supervisor completo y no bloqueante.*

d) $S^{f \vee g}$ *es ω-controlable con respecto a G y ω-cerrado con respecto a S.*

DEMOSTRACIÓN.

Parte a)

Usaremos inducción sobre la longitud de las palabras.

Caso base: $1 \in L^{f \vee g}$ si y sólo si $1 \in L^f \cup L^g$.

Ahora asumimos que toda palabra de longitud n está en $L^{f \vee g}$ si y sólo si está en $L^f \cup L^g$. Sea $m = k\sigma$ una palabra de longitud $n+1$ tal que $k \in L^{f \vee g}$ y $\sigma \in \Sigma$. Entonces, por la hipótesis de inducción tenemos que $k \in L^f$ o $k \in L^g$.

Ahora bien,

$$
\begin{aligned}
k\sigma \in L^{f \vee g} \quad & \text{si y sólo si } k \in L^{f \vee g},\ \sigma \in f \vee g(k) \text{ y } k\sigma \in L \\
& \text{si y sólo si } k \in L^{f \vee g},\ \sigma \in f(k) \cup g(k) \text{ y } k\sigma \in L \\
& \text{si y sólo si } k\sigma \in L^f \text{ o } k\sigma \in L^g
\end{aligned}
$$

Por lo tanto, $L^{f \vee g} = L^f \cup L^g$.

Parte b)

($\supseteq$)

$$
\begin{aligned}
S^{f \vee g} &= lim(L^{f \vee g}) \cap S \\
&= lim(L^f \cup L^g) \cap S && \text{(parte (a))} \\
&\supseteq lim(L^f) \cup lim(L^g) \cap S && \text{(por la proposición 2.1.8 } (a)) \\
&= S^f \cup S^g
\end{aligned}
$$

($\subseteq$)

Sea $s \in lim(L^{f \vee g}) \cap S$. Entonces, $s \in lim(L^{f \vee g})$ y $s \in S$. Por la definición 2.1.6, tenemos que $pre(s) \subseteq L^{f \vee g}$, es decir, $pre(s) \subseteq L^f \cup L^g$. Si $pre(s) \subseteq L^f$, entonces $s \in lim(L^f)$ y por lo tanto $s \in lim(L^f) \cup lim(L^g) \cap S$. Ahora bien, si $pre(s) \subseteq L^g$, entonces $s \in lim(L^g)$ y por lo tanto $s \in lim(L^f) \cup lim(L^g) \cap S$.

Parte c)

Primero mostraremos que $f \vee g$ es completo.

Sean $k \in L^{f \vee g}$, $\sigma \in \Sigma$, $k\sigma \in L$ y $\sigma \in f \vee g$. Entonces, $k \in L^f$o $k \in L^g$. Si $k \in L^f$, entonces $k\sigma \in L^f$, por ser f un supervisor completo. Ahora si $k \in L^g$, entonces por la misma razón $k\sigma \in L^g$. Por lo tanto, tenemos que $k\sigma \in L^{f \vee g}$.

Probaremos ahora que $f \vee g$ es no bloqueante.

$$\begin{aligned} L^{f \vee g} &= L^f \cup L^g \\ &= pre(S^f) \cup pre(S^g) \\ &= pre(S^f \cup S^g) \qquad \text{(por la proposición 2.1.5 } (a)) \\ &= pre(S^{f \vee g}) \end{aligned}$$

Parte d)

Debido a que S^f y S^g son ω-controlable con respecto a G , $S^{f \vee g}$ también lo es por la proposición 3.3.10.

Probaremos ahora que $S^{f \vee g}$ es ω-cerrado con respecto a S. Específicamente, probaremos que $S^{f \vee g} = clo(S^{f \vee g}) \cap S$.

($\subseteq$)

Por la proposición 2.1.13 (a) y la hipótesis, tenemos que $S^{f \vee g} \subseteq clo(S^{f \vee g}) \cap S$.

($\supseteq$)

Sea $t \in clo(S^{f \vee g}) \cap S$ y probaremos que $t \in S^{f \vee g}$

Como $t \in clo(S^{f \vee g}) \cap S$, $t \in lim(pre(S^{f \vee g}))$, es decir, $pre(t) \subseteq pre(S^f \cup S^g)$. Esto significa que $pre(t) \subseteq pre(S^f) \cup pre(S^g)$. Si $pre(t) \subseteq pre(S^f)$, entonces $t \in lim(pre(S^f))$. En otras palabras, $t \in clo(S^f)$. Pero como $clo(S^f) \cap S = S^f$ y $t \in S$, entonces $t \in S^f$. Para el caso de que $pre(t) \subseteq pre(S^g)$, se demuestra igual que en el caso anterior. Por lo tanto, $t \in S^g$.

Esto prueba que $t \in S^f \cup S^g$, lo que completa la demostración. □

Cabe mencionar que el comportamiento generado por la disyunción de supervisores es ω-cerrado con respecto al comportamiento del sistema, ya que esta unión es una unión finita.

3.5.4. Problema del control supervisor modular utilizando disyunción ($PCSMU^{\omega}$).

El $PCSMU^{\omega}$ consiste en lo siguiente:

PROBLEMA 3.5.10. Dados los lenguajes $S^{f_1}, S^{f_2}, \ldots, S^{f_n}$, construir un controlador supervisor no bloqueante y completo g tal que $S^g = \bigcup_{i=1}^{n} S^{f_i}$.

De acuerdo a la proposición 3.5.9, siempre podemos construir un supervisor modular utilizando disyunción de supervisores. Esto se debe a que el comportamiento generado bajo la supervisión, es decir S^g, siempre es ω-controlable con respecto a G y ω-cerrado con respecto a S.

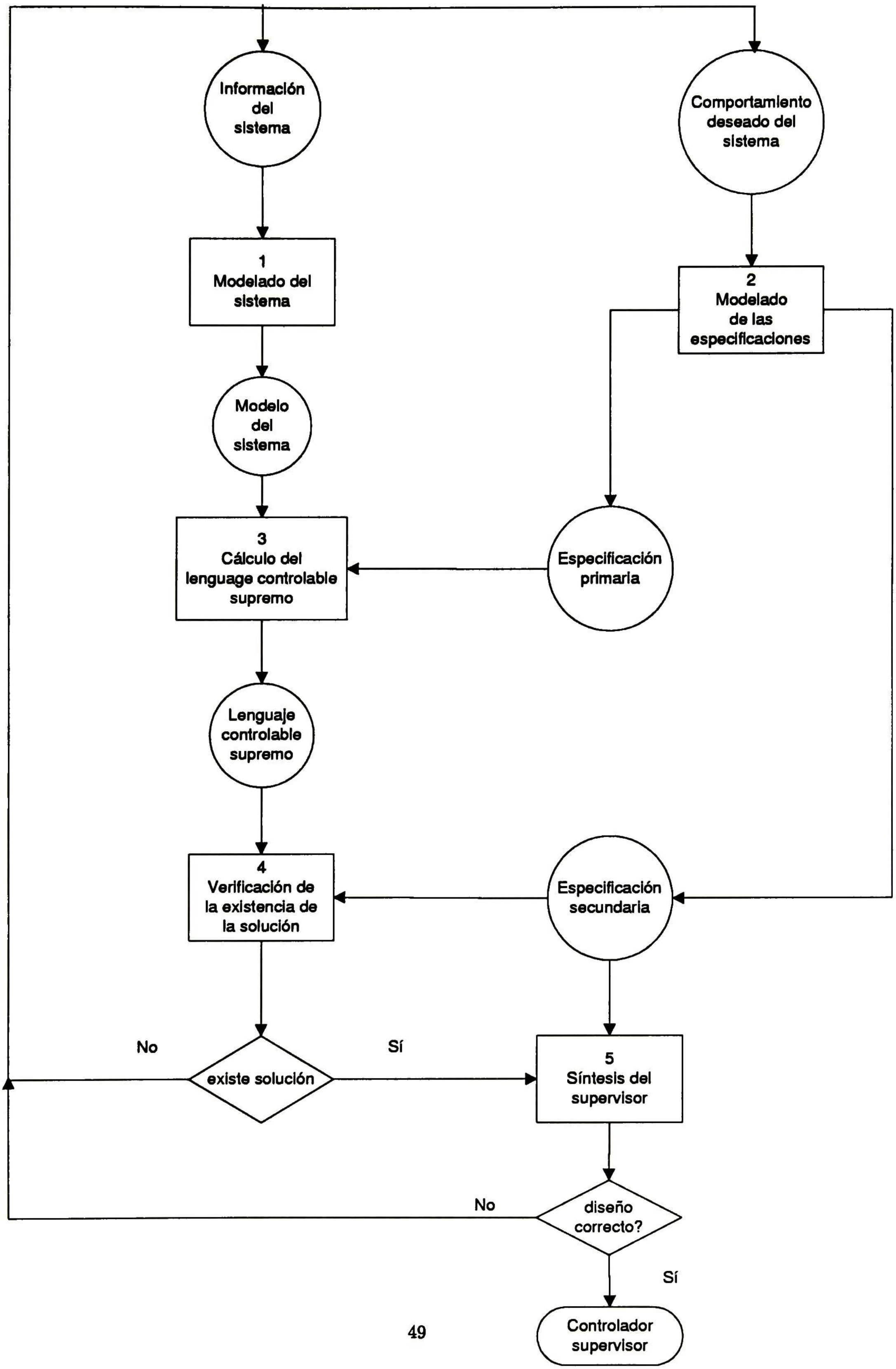
3.6. Método de Síntesis

Utilizando la teoría de control que presentamos en la sección 3.4, proponemos un método de síntesis en el cual modelamos el sistema y la especificación bajo una misma estructura para facilitar el cálculo del lenguaje controlable supremo. Este método se muestra en la figura 3.6.1. Los rectángulos corresponden a los cinco pasos principales del método y los círculos representan las entradas y salidas de información. Los pasos 1 y 2 están asociados al modelado del sistema y la especificación del comportamiento deseado, respectivamente. El paso 3 es la computación del lenguaje ω-controlable supremo. En el paso 4 verificamos la existencia de la solución. Finalmente, el paso 5 es la síntesis del supervisor. Como podemos ver en el diagrama, el método es iterativo.

3.6.1. Paso 1: Modelado del sistema. Modelamos el sistema $G = (L, S)$ como dos autómatas deterministas: $(\Sigma, X, \delta, x_0, X)$ y $(\Sigma, X, \delta, x_0, R)$. Tal que el $_*$-autómata $(\Sigma, X, \delta, x_0, X)$ reconoce el $_*$-lenguaje $L \subseteq \Sigma^*$ y el autómata de Büchi $(\Sigma, X, \delta, x_0, R)$ reconoce el ω-lenguaje $S \subseteq \Sigma^\omega$.

3.6.2. Paso 2: Modelado de las especificaciones. Dividimos la especificación en dos:

- *Especificación primaria*: determinamos esta especificación con base en la información del comportamiento deseado del sistema y la representamos mediante el ω-lenguaje $E \subseteq S \subseteq \Sigma^\omega$ Al autómata de Büchi $(\Sigma, X, \delta, x_0, R)$ le agregamos un conjunto de pares de aceptación $\{(R_p, I_p) : p \in P\}$ donde P es un conjunto de índices y para cada $(R_p, I_p) \in 2^X \times 2^X$ Esto lo hacemos para formar un nuevo autómata de Rabin-Büchi $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ tal que el autómata de Rabin $(\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ reconozca el ω-lenguaje


Información del sistema
Comportamiento deseado del sistema
1
Modelado del sistema
2
Modelado de las especificaciones
Modelo del sistema
3
Cálculo del lenguage controlable supremo
Especificación primaria
Lenguaje controlable supremo
4
Verificación de la existencia de la solución
Especificación secundaria
No
Sí
existe solución
5
Síntesis del supervisor
No
diseño correcto?
Sí
Controlador supervisor

$E \subseteq S \subseteq \Sigma^\omega$. Especificamos este lenguaje de acuerdo a la clasificación seguridad-vivacidad vista en el capítulo 1.

- *Especificación secundaria*: esta especificación, que denotamos como A, debe ser un subconjunto de la especificación primaria y es dada mediante un autómata de Rabin o Büchi. Como lo indica el corolario 3.4.13, esta especificación es necesaria sólo si el lenguaje de especificación E no es ω-cerrado con respecto a S. En tal caso el ω-lenguaje A nos permite lograr una solución óptima al PCS^ω, esto es, que el comportamiento generado bajo la supervisión sea lo más aproximado que se desee al lenguaje ω-controlable supremo. En caso de que E sea ω-cerrado con respecto a S, entonces siempre $A := \emptyset$.

Nótese que los lenguajes $L \subseteq \Sigma^*$, $S \subseteq \Sigma^\omega$, y $E \subseteq \Sigma^\omega$ están en una misma estructura de autómata, pero con diferentes criterios de aceptación.

3.6.3. Paso 3: Computación del lenguaje ω-controlable supremo. Calcular el lenguaje ω-controlable supremo del lenguaje $E \subseteq \Sigma^\omega$, requiere dos pasos: computación de los prefijos controlables $pre_G(E)$ y computación del autómata que reconoce el lenguaje ω-controlable supremo.

Paso 3.1. Computación de los prefijos controlables $pre_G(E)$.

Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ el autómata de Rabin-Büchi que obtuvimos en los pasos 1 y 2. Entonces, con el fin de facilitar el cálculo del $_*$-lenguaje $pre_G(E)$, a continuación definimos $pre_G(E)$ en términos de estados.

DEFINICIÓN 3.6.1. Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi. Definimos el *conjunto de estados controlables* del autómata $\mathcal{A}$ como sigue:

$$C^{\mathcal{A}} := \{x \in X : E' \neq \emptyset \text{ , } * - \text{controlable con respecto a } L_x \text{ y}$$
$$\omega - \text{cerrado con respecto a } S_x, \text{ para algún } E'_x \subseteq E_x\}$$

donde para todo $x \in X$, $E_x \subseteq \Sigma^\omega$ es el lenguaje reconocido por el autómata de Rabin $(\Sigma, X, \delta, x, \{(R_p, I_p) : p \in P\})$, L_x es el lenguaje reconocido por el $_*$-autómata $(\Sigma, X, \delta, x, X)$ y $S_x \subseteq \Sigma^\omega$ es el lenguaje reconocido por el autómata de Büchi $(\Sigma, X, \delta, x, R)$.

Debido a que $\mathcal{A}$ es determinista, entonces para todo prefijo $k \in pre(E)$, se cumple que $k \in pre_G(E)$ si y sólo si $\delta(k, x_0) \in C^{\mathcal{A}}$ [13].

Para el cálculo del conjunto $C^{\mathcal{A}}$, refiérase al capítulo 5.

Paso 3.2. Computación del autómata que reconoce el lenguaje ω-controlable supremo.

Ya que hemos calculado $pre_G(E)$, el siguiente paso es encontrar el ω − lenguaje $supC^{\omega}(E) = lim(supCF^{*}(pre_G(E))) \cap E$. Para esto es conveniente definir las siguientes funciones.

DEFINICIÓN 3.6.2. Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi determinista. Entonces, $\Gamma^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$ es una función parcial definida por la siguiente regla:

$$\Gamma^{\mathcal{A}}(x) := \bigcup\{\Gamma \in \mathcal{C} : \text{ si } \delta(\sigma, x)!, \text{ entonces } \delta(\sigma, x) \in C^{\mathcal{A}}, \text{ para todo } \sigma \in \Gamma\}$$

y $\delta^{\mathcal{A}} : \Sigma \times X \to X$ es una función de transición definida como sigue:

$$\delta^{\mathcal{A}}(\sigma, x) := \begin{cases} \delta(\sigma, x) & \text{si } x \in C^{\mathcal{A}} \text{ y } \Gamma^{\mathcal{A}}(x) \\ indefinida & \text{en otro caso} \end{cases}$$

Esta función se extiende a la función $\delta^{\mathcal{A}} : \Sigma^{*} \times X \to X$ dada por las siguientes reglas:

- $\delta^{\mathcal{A}}(1, x) := x$,
- $\delta^{\mathcal{A}} : (k\sigma, x) := \delta^{\mathcal{A}}(\sigma, \delta^{\mathcal{A}}(k, x))$

La función de transición $\delta^{\mathcal{A}}$ y el conjunto $C^{\mathcal{A}}$ generan el lenguaje $supCF^{*}(pre_G(E))$ como se demuestra a continuación.

PROPOSICIÓN 3.6.3. [13]. *Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi determinista. Sean E el ω-lenguaje reconocido por el autómata de Rabin $(\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ y L el $_*$-lenguaje reconocido por $(\Sigma, X, \delta^{\mathcal{A}}, x_0, X)$. Entonces el $_*$-autómata $sup^{*} = (\Sigma, X, \delta^{\mathcal{A}}, x_0, C^{\mathcal{A}})$ reconoce el $_*$-lenguaje $supCF^{*}(pre_G(E))$.*

DEMOSTRACIÓN. Sea L' el $_*$-lenguaje reconocido por el $_*$-autómata sup^{*} Probaremos que $L' = supCF^{*}(pre_G(E))$. Como $\delta^{\mathcal{A}}$ es una restricción de δ, entonces

$L' \subseteq pre_G(E)$. Esto prueba que $L' \in CF^*(pre_G(E))$. Mostraremos ahora que, si $L'' \in CF^*(pre_G(E))$, entonces $L'' \subseteq L'$. Utilizaremos inducción sobre la longitud de las palabras para demostrar que toda $k \in L''$ está contenida en L'

Caso base: si $1 \in L''$, entonces $1 \in pre_G(E) \Leftrightarrow x_0 \in C^{\mathcal{A}} \Leftrightarrow 1 \in L'$.

Hipótesis de Inducción: Para $|k| = n$ supongamos que si $k \in L''$, entonces $k \in L'$

Por la hipótesis de inducción basta con probar que si $k \in L'' \cap L'$, entonces $\Sigma_{L''}(k) \subseteq \Sigma_{L'}(k)$

Supongamos que no es así. Entonces, existe $\sigma \in \Sigma_{L''}(k)$ tal que $\sigma \notin \Sigma_{L'}(k)$. Esto implica que $\sigma \notin \Gamma^{\mathcal{A}}(\delta^{\mathcal{A}}(k, x_0))$. Como $\delta^{\mathcal{A}}$ es una restricción de δ, entonces tenemos que $\sigma \notin \Gamma^{\mathcal{A}}(\delta(k, x_0))$. Luego, para todo $\Gamma \in \mathcal{C}$, si $\sigma \in \Gamma$, entonces existe un $\sigma' \in \Gamma$ tal que $\delta(\sigma', \delta(k, x_0)) \notin C^{\mathcal{A}}$, es decir, $\sigma' \notin \Sigma_{pre_G(E)}(k)$. Esto implica que $\Gamma \cap \Sigma_L(k)$ no es subconjunto de $\Sigma_{pre_G(E)}(k)$. Pero, por hipótesis $L' \in CF^*(pre_G(E))$. Entonces, $L' \subseteq pre_G(E)$. Luego, $\Gamma \cap \Sigma_L(k)$ no es subconjunto de $\Sigma_{L''}(k)$, lo que contradice el hecho de que L'' es $_*$-controlable.

Por lo tanto, $L' = supCF^*(pre_G(E))$. □

De acuerdo con las definiciones anteriores y la proposición 3.6.3, para construir el autómata que reconoce el lenguaje ω-controlable supremo debemos calcular la función $\delta^{\mathcal{A}} : \Sigma^* \times X \rightarrow X$ relativa al autómata de Rabin-Büchi $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ y construir el autómata de Rabin $\mathcal{A}_{sup} = (\Sigma, X, \delta^{\mathcal{A}}, x_0, \{(R_p, I_p) : p \in P\})$.[3]

El siguiente teorema demuestra que el autómata A_{sup} reconoce el ω-lenguaje $supC^{\omega}(E)$.

TEOREMA 3.6.4. [13]*Sea* $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ *un autómata de Rabin-Büchi determinista. Sean* E *el* ω*-lenguaje reconocido por el autómata de Rabin* $(\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$, L *el* $_*$*-lenguaje reconocido por el* $_*$*-autómata* $(\Sigma, X, \delta, x_0, X)$ *y* $S \subseteq lim(L)$ *el* ω*-lenguaje reconocido por el por el autómata de Büchi* $(\Sigma, X, \delta, x_0, R)$. *Si* $E \subseteq S$, *entonces el autómata de Rabin determinista* $A_{sup} = (\Sigma, X, \delta^{\mathcal{A}}, x_0, \{(R_p, I_p) : p \in P\})$ *reconoce el* ω*-lenguaje* $supC^{\omega}(E)$ *relativo a* $G = (L, S)$.

[3]Nótese que el autómata $\mathcal{A}_{sup}$ tiene la misma estructura que el autómata de Rabin-Büchi $\mathcal{A}$ excepto por la función de transición $\delta^{\mathcal{A}}$

DEMOSTRACIÓN. Se sigue directamente de las proposiciones 3.6.3 y 3.4.8. □

3.6.4. Paso 4: Verificación de la existencia de la solución. Una vez que hemos calculado el ω-lenguaje $supC^{\omega}(E)$, el siguiente paso es verificar si hay solución del PCS^{ω} Esto es, si el ω-lenguaje $supC^{\omega}(E) \neq \emptyset$ y $infF^{\omega}(A) \subseteq supC^{\omega}(E)$.

Para verificar que el ω-lenguaje $supC^{\omega}(E) \neq \emptyset$ es suficiente verificar que el estado inicial del autómata $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ es un elemento del conjunto $C^{\mathcal{A}}$ En caso de que $x_0 \notin C^{\mathcal{A}}$, entonces regresar al paso 1 o 2 del método según sea necesario.

Ahora sólo resta verificar que $infF^{\omega}(A) \subseteq supC^{\omega}(E)$. Si la especificación secundaria $A = \emptyset$, entonces ir directamente al paso 5 del método. En otro caso construimos un $_*$-autómata $\mathcal{A}_{inf} = (\Sigma, X, \delta, x_0, F)$ que reconoce el $_*$-lenguaje $pre(A)$ [4] y junto con el autómata de Büchi determinista $\mathcal{A}_s = (\Sigma, X', \delta', x_0', R')$ que reconoce el ω-lenguaje S y el autómata de Rabin determinista $\mathcal{A}_{sup} = (\Sigma, X'', \delta'', x_0'', \{(R_p, I_p) : p \in P\})$ construimos un autómata de Streett con las reglas siguientes:

$\mathcal{A}_{diff} = (\Sigma, X''', \delta''', x_0''', \{(X \times R \times X'', \emptyset), (F \times X' \times X'', \emptyset\} \cup \{(R_p''', I_p''') : p \in P\})$, donde

$$X''' := X \times X' \times X''$$

$$\delta'''(\sigma, (x, x', x'')) := \begin{cases} (\delta(\sigma, x), \delta'(\sigma, x'), \delta''(\sigma, x'')) & \text{si } \mathcal{A}_{inf} \text{ es determinista} \\ \delta(\sigma, x) \times \{\delta'(\sigma, x') \times \delta''(\sigma, x'')\} & \text{en otro caso} \end{cases}$$

$$x_0''' := (x_0, x_0' x_0'')$$

$$I_p''' := X \times X' \times (X'' \setminus I_p)$$

$$R_p''' := X \times X' \times (X'' \setminus R_p)$$

El lenguaje reconocido por el autómata $\mathcal{A}_{diff}$ es el lenguaje $infF^{\omega}(A) \setminus supC^{\omega}(E)$. Por lo tanto, para verificar que $infF^{\omega}(A) \setminus supC^{\omega}(E) = \emptyset$ es suficiente verificar que

[4] Utilizando el algoritmo propuesto en [2], para verificar si el autómata es vacío.

el autómata $\mathcal{A}_{diff}$ es vacío[5] Si $\mathcal{A}_{diff}$ no es vacío, entonces regresar al paso 1 o 2 del método según sea necesario.

3.6.5. Paso 5: Síntesis del supervisor. Si $A = \emptyset$, calculamos la función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$ siguiendo la demostración del teorema 4.1.15 si utilizamos la definición 4.1.14 para calcular el conjunto $C^{\mathcal{A}}$ o la demostración de 4.2.7 si utilizamos la definición 4.2.6. Posteriormente, definimos un controlador supervisor no bloqueante y completo $f : L \to \mathcal{C}$ como sigue:

$$f(l) := \begin{cases} \Phi^{\mathcal{A}}(\delta(l, x_0)) & \text{si } \delta(l, x_0) \in C^{\mathcal{A}} \\ indefinida & \text{en otro caso} \end{cases}$$

La función f es un controlador supervisor para $G = (L, S)$ que genera un sublenguaje del ω-lenguaje $supC^{\omega}(E)$.

En caso de que $A \neq \emptyset$, podemos utilizar el procedimiento de la prueba del teorema 3.4.10, para sintetizar el supervisor.

Primero construimos un $_*$-autómata $(\Sigma, X'', \delta'', x_0'', F'')$[6] que reconoce el $_*$-lenguaje $pre(supC^{\omega}(E))$ y calculamos la función $\Phi_0 : F'' \to \mathcal{C}$, utilizando la siguiente regla

$$\Phi_0(x') := \{\sigma \in \Sigma : \delta''(\sigma, x') \in F'' \text{ o } \sigma \in \Sigma_u\}$$

Después definimos la función $f_0 : pre(A) \to \mathcal{C}$ que es un controlador supervisor completo y no bloqueante tal que

$$f_0(l) := \Phi_0(\delta''(l, x_0))$$

Ahora calculamos la función total $\Phi^{\mathcal{A}} : X'' \to \mathcal{C}$ siguiendo la demostración del teorema 4.1.15 o 4.2.7, dependiendo si utilizamos la definición 4.1.14 o la definición 4.2.6 para calcular el conjunto $C^{\mathcal{A}}$. Definimos la función $f_1 : pre(supC^{\omega}(E)) \to \mathcal{C}$ que es un controlador supervisor no bloqueante y completo tal que

$$f_1(l) := \Phi^{\mathcal{A}}(\delta''(l, x_0))$$

[5]Podemos verificar en tiempo polinomial si un ω-autómata es vacío utilizando también el algoritmo propuesto en [2].

[6]Nótese que Σ, X'', δ'' y x_0 son tomados del autómata de Rabin $\mathcal{A}_{sup}$.

Finalmente, obtenemos el controlador supervisor completo $f : L \to \mathcal{C}$:

$$f(l) := \begin{cases} f_0(l) & \text{si } l \in pre(A) \\ f_1(l) & \text{si } l \in pre(supC^{\omega}(E)) \setminus pre(A) \end{cases}$$

La función f_0 es un controlador supervisor para $G = (L, S)$ que genera el lenguaje $pre(supC^{\omega}(E))$ y la función f_1 es un controlador supervisor no bloqueante y completo para $G = (L/k, S/k)$ que genera un sublenguaje no vacío de E/k, para todo $k \in pre(supC^{\omega}(E))$.

CAPÍTULO 4

Cálculo del Conjunto Controlable $C^{\mathcal{A}}$

En el paso 3 del método de síntesis dejamos pendiente el cálculo del conjunto controlable $C^{\mathcal{A}}$. En este capítulo definimos el algoritmo para calcular este conjunto utilizando cálculo de puntos fijos.

Dividimos este capítulo en dos partes. En la primera parte introduciremos la teoría y algoritmos para calcular el conjunto controlable de un SED $G = (L, S)$ asumiendo que S es ω-cerrado, es decir, asumimos que el modelo del sistema no contiene propiedades de vivacidad. En la segunda parte hacemos una extensión de las definiciones de la primera parte para calcular el conjunto controlable de un SED G sin importar como es S. Hacemos esta diferencia por motivos de claridad, ya que es más simple el cálculo del conjunto controlable $C^{\mathcal{A}}$ cuando S es ω-cerrado.

4.1. Conjunto Controlable del Autómata de Rabin.

Definimos $\mathcal{A}_x$ como el autómata $\mathcal{A}$ designando $x \in X$ como el estado inicial. Decimos que $v \in \Sigma^\infty$ es generada por $\mathcal{A}_x$ bajo f si v es generada por $\mathcal{A}_x$ y para toda $k\sigma \in pre(v)$, k es generada por $\mathcal{A}_x$, $f(k)$ está definida y $\sigma \in f(k)$.

Recordemos que el conjunto $\mathcal{C} \subseteq 2^\Sigma$ es una familia de patrones de control definida como:

$$\mathcal{C} = \{\Gamma : \Gamma \in 2^\Sigma \text{ y } \Sigma_u \subseteq \Gamma\}$$

De acuerdo a la definición 3.6.1, el conjunto $C^{\mathcal{A}}$ representa el conjunto de estados desde los cuales el autómata $\mathcal{A}$ puede ser controlado para satisfacer su condición de aceptación. Esta interpretación se puede representar también de la siguiente forma.

DEFINICIÓN 4.1.1. (***Conjunto controlable del autómata*** $\mathcal{A}$). Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ un autómata de Rabin. Definimos el *conjunto controlable del autómata* $\mathcal{A}$, que denotamos $P^{\mathcal{A}}$, como el conjunto de todos los $x \in X$ para los cuales existe una función $f : \Sigma^* \to \mathcal{C}$ tal que:

1. toda $s \in \Sigma^\omega$ generada por $\mathcal{A}_x$ bajo f es aceptada por $\mathcal{A}_x$ y
2. para toda $k \in \Sigma^*$ generada por $\mathcal{A}_x$ bajo f, existe un $\sigma \in \Sigma$ tal que $k\sigma$ es generada por $\mathcal{A}_x$ bajo f.

Una forma de medir la complejidad estructural de un autómata de Rabin, es tomando en cuenta el número de pares de aceptación, otra forma es tomando en cuenta el número de estados "vivos".

DEFINICIÓN 4.1.2. Para todo autómata de Rabin $\mathcal{A}$ definimos los siguientes conjuntos:

- $Live(\mathcal{A}) := \{x \in X : \delta(\sigma, x)! \text{ y } \delta(\sigma, x) \neq x, \text{ para algún } \sigma \in \Sigma\}$
- $Deg(\mathcal{A}) := \{x \in X : \delta(\sigma, x)!, \text{ para algún } \sigma \in \Sigma \text{ y si } \delta(\sigma, x)!, \text{ entonces } \delta(\sigma, x) = x, \text{ para todo } \sigma \in \Sigma\}$

Un estado es un elemento del conjunto $Deg(\mathcal{A})$ si existe una transición definida en ese estado y además toda transición, definida en ese estado, conduce al mismo estado. Un estado es un elemento del conjunto $Live(\mathcal{A})$ si existe una transición definida en ese estado que conduzca a otro estado del autómata. Llamamos a los elementos del conjunto $Live(\mathcal{A})$ estados "vivos"

DEFINICIÓN 4.1.3. Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ un autómata de Rabin. Para cualesquiera $x \in X$, $\sigma \in \Sigma$, $X' \subseteq X$ y $p \in P$ definimos las siguientes operaciones sobre $\mathcal{A}$

a) <u>auto-lazo de un conjunto</u>:

$(\mathcal{A} \hookrightarrow X') := (\Sigma, X, \delta', x_0, \{(R'_p, I'_p) : p \in P\})$, donde

$$\delta'(\sigma, x') := \begin{cases} x' & \text{si } x' \in X' \\ \delta(\sigma, x') & \text{en otro caso} \end{cases}$$

$R'_p := (R_p \cup X')$ y $I'_p := (I_p \cup X')$, para todo $p \in P$.

b) <u>restricción a un conjunto</u>:

$(\mathcal{A} \uparrow X') := (\Sigma, X, \delta', x_0, \{(R'_p, I'_p) : p \in P\})$, donde

$$\delta'(\sigma, x') := \begin{cases} \delta(\sigma, x') & \text{si } x' \in X' \\ x' & \text{en otro caso} \end{cases}$$

$R'_p := (R_p \cap X')$ y $I'_p := (I_p \cap X')$, para todo $p \in P$.

c) <u>exclusión de un par</u>:

Sea $(\mathcal{A} \uparrow (I_p \cup Deg(\mathcal{A}))) := (\Sigma, X, \delta', x_0, \{(R'_q, I'_q) : q \in P\})$, donde

$$\mathcal{A} \downarrow p := \begin{cases} (\Sigma, X, \delta', x_0, \{(R'_q, I'_q) : q \in P\}) & \text{si } \mid P \mid = 1 \\ (\Sigma, X, \delta', x_0, \{(R'_q, I'_q) : q \in P \setminus \{p\}\}) & \text{si } \mid P \mid > 1 \end{cases}$$

Las operaciones de la proposición anterior reducen la complejidad estructural del autómata $\mathcal{A}$. Específicamente, las operaciones (a), (b) y (c) disminuyen el número de estados "vivos" del autómata mientras que (c) disminuye el número de pares de la condición de aceptación. Estas operaciones nos serán de utilidad en el cálculo del conjunto $P^{\mathcal{A}}$ para convertir autómatas complejos en autómatas más simples. La siguiente proposición muestra qué papel juega el conjunto $P^{\mathcal{A}}$ bajo estas operaciones.

PROPOSICIÓN 4.1.4. *Sea* $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ *un autómata de Rabin determinista y sea* $X' \subseteq X$. *Entonces,*

a) $P^{\mathcal{A}} \cap Deg(\mathcal{A}) = \bigcup_{p \in P}(R_p \cap I_p) \cap Deg(\mathcal{A})$

b) $P^{\mathcal{A}} \cup X' \subseteq P^{\mathcal{A}(\hookrightarrow X')}$

c) $P^{\mathcal{A} \uparrow X'} \subseteq P^{\mathcal{A}} \cap X'$

d) Para todo $p \in P$, $P^{\mathcal{A} \downarrow p} \subseteq P^{\mathcal{A}} \cap (I_p \cup Deg(\mathcal{A}))$

DEMOSTRACIÓN.

Parte a)

$(\supseteq)$

Sea $x \in \bigcup_{p \in P}(R_p \cap I_p) \cap Deg(\mathcal{A})$. Entonces, $x \in Deg(\mathcal{A})$ y $x \in (R_p \cap I_p)$ para algún $p \in P$. Como $x \in Deg(\mathcal{A})$, entonces para todo $\sigma \in \Sigma$ tal que $\delta(\sigma, x)$ está definida, $\delta(\sigma, x) = x$. Luego, existe un camino π en $\mathcal{A}_x$ que es aceptado, es decir, $x \in \Omega_\pi$ y para todo $i \in N$ tal que $i \geq 0$, tenemos que $\pi(i) = x$. En particular π es aceptado por $\mathcal{A}_x$ bajo la función $f : \Sigma^* \to \mathcal{C}$. Por lo tanto, $x \in P^{\mathcal{A}}$, por la definición 4.1.1. Esto implica que $x \in P^{\mathcal{A}} \cap Deg(\mathcal{A})$.

$(\subseteq)$

Sea $x \in P^{\mathcal{A}} \cap Deg(\mathcal{A})$. Entonces, $x \in Deg(\mathcal{A})$ y $x \in P^{\mathcal{A}}$ Por la definición 4.1.1, toda $s \in \Sigma^\omega$ generada por $\mathcal{A}_x$ bajo la función $f : \Sigma^* \to \mathcal{C}$ es aceptada por $\mathcal{A}_x$. Luego, s tiene un camino π en $\mathcal{A}_x$ tal que $\pi(1) = x$, $\Omega_\pi \cap R_p \neq \emptyset$ y $x \in \Omega_\pi \subseteq I_p$ para algún $p \in P$ Como $x \in Deg(\mathcal{A})$, entonces $x \in \Omega_\pi$ y como $\Omega_\pi \cap R_p \neq \emptyset$ y $x \in \Omega_\pi \subseteq I_p$ para algún $p \in P$, entonces $x \in R_p$ y $x \in I_p$. Por lo tanto, $x \in R_p \cap I_p$ para algún $p \in P$. Esto implica que $x \in \bigcup_{p \in P}(R_p \cap I_p) \cap Deg(\mathcal{A})$.

Parte b)

Sea $x \in P^{\mathcal{A}} \cup X'$ Si $x \in P^{\mathcal{A}}$, entonces toda $s \in \Sigma^{\omega}$ generada por $\mathcal{A}_x$ bajo la función $f : \Sigma^* \to C$ es aceptada por $\mathcal{A}_x$. Luego, s tiene un camino π en $\mathcal{A}_x$ que es aceptado. Por ser π un camino aceptado, entonces existe un $p \in P$, tal que $\Omega_\pi \cap R_p \neq \emptyset$ y $\Omega_\pi \subseteq I_p$. Luego, para todo $x' \in \Omega_\pi \cap R_p$ tenemos que $x' \in \Omega_\pi \cap (R_p \cup X')$ y para todo $x' \in \Omega_\pi \subseteq I_p$ tenemos que $x' \in \Omega_\pi \subseteq I_p \cup X'$ Esto implica que π es aceptado por $\mathcal{A}_x(\hookrightarrow X')$. En particular, π es aceptado por $\mathcal{A}_x(\hookrightarrow X')$ bajo f. Esto prueba que $x \in P^{\mathcal{A}(\hookrightarrow X')}$.

Si $x \in X'$, entonces supongamos que $x \notin P^{\mathcal{A}(\hookrightarrow X')}$. Por la definición 4.1.1, existe una palabra $s \in \Sigma^{\omega}$ generada por $P^{\mathcal{A}(\hookrightarrow X')}$ tal que s tiene un camino π que no es aceptado por $\mathcal{A}_x(\hookrightarrow X')$. Ahora bien, $\pi(1) = x$ y como $x \in X'$, entonces para todo $i \in N$ tal que $i \geq 0$, tenemos que $\pi(i) = x$. Por lo tanto, $x \in \Omega_\pi$. Esto implica que $\Omega_\pi \cap (R_p \cup X') \neq \emptyset$ y también $x = \Omega_\pi \subseteq (I_p \cup X')$ para algún $p \in P$. Luego, π es un camino aceptado por $\mathcal{A}_x(\hookrightarrow X')$, lo que constituye una contradicción. Por lo tanto, $x \in P^{\mathcal{A}(\hookrightarrow X')}$.

Parte c)

Sea $x \in P^{\mathcal{A} \uparrow X'}$ Entonces, por la definición 4.1.1, toda $s \in \Sigma^{\omega}$ generada por $\mathcal{A}_x \uparrow X'$ bajo f es aceptada por $\mathcal{A}_x \uparrow X'$. Luego, existe un camino π de s que es aceptado por $\mathcal{A}_x \uparrow X'$ y $\pi(1) = x$. Por ser π un camino aceptado por $\mathcal{A}_x \uparrow X'$, existe un $p \in P$ tal que $\Omega_\pi \cap R_p \cap X' \neq \emptyset$ y $\Omega_\pi \subseteq I_p \cap X'$, es decir, existe un $p \in P$ tal que $\Omega_\pi \cap R_p \neq \emptyset$ y $\Omega_\pi \subseteq I_p$. Por lo tanto, π es aceptado por $\mathcal{A}_x$. En particular, π es aceptado por $\mathcal{A}_x$ bajo la función $f : \Sigma^* \to C$. Esto prueba que $x \in P^{\mathcal{A}}$.

Supongamos ahora que $x \notin X'$ y como por hipótesis $x \in P^{\mathcal{A} \uparrow X'}$, entonces para todo $i \in N$ tal que $i \geq 0$, tenemos que $\pi(i) = x$. Luego, $x \in \Omega_\pi$ y como π es un camino aceptado por $\mathcal{A}_x \uparrow X'$, entonces $x \in R_p \cap X'$ para algún $p \in P$. Por lo tanto, $x \in X'$, lo que constituye una contradicción.

Por lo tanto, $x \in P^{\mathcal{A}} \cap X'$

Parte d)

Sean $p \in P$ y $x \in P^{\mathcal{A} \downarrow p}$. Entonces, toda $s \in \Sigma^{\omega}$ generada por $\mathcal{A}_x \downarrow p$ bajo la función $f : \Sigma^* \to C$ es aceptada por $\mathcal{A}_x \downarrow p$. Por lo tanto, s existe un camino π que es aceptado por $\mathcal{A} \uparrow (I_p \cup Deg(\mathcal{A}))$ tal que $\pi(1) = x$. Por ser π un camino aceptado por

$\mathcal{A} \uparrow (I_p \cup Deg(\mathcal{A}))$, entonces existe un $r \in P$ tal que $\Omega_\pi \cap (R_r \cap (I_p \cup Deg(\mathcal{A}))) \neq \emptyset$ y $\Omega_\pi \subseteq I_r \cap (I_p \cup Deg(\mathcal{A}))$. Esto implica que $\Omega_\pi \cap R_r \neq \emptyset$ y $\Omega_\pi \subseteq I_r$ para algún $r \in P$ Por lo tanto, π es aceptado por $\mathcal{A}_x$. En particular, π es aceptado por $\mathcal{A}_x$ bajo f. Esto prueba que $x \in P^{\mathcal{A}}$.

Supongamos ahora que $x \notin I_p \cup Deg(\mathcal{A})$. Entonces, $x \notin I_p$ y $x \notin Deg(\mathcal{A})$. Como $\pi(1) = x$ y π es un camino aceptado por $\mathcal{A}_x \uparrow (I_p \cup Deg(A))$, entonces $\delta(\sigma, x) = x$, para todo $\sigma \in \Sigma$ tal que $\delta(\sigma, x)!$, es decir, para todo $i \in N$ tal que $i \geq 0$, tenemos que $\pi(i) = x$. Luego, $x \in Deg(\mathcal{A})$, lo que constituye una contradicción. Por lo tanto, $x \in I_p \cup Deg(\mathcal{A})$.

□

4.1.1. Operador de dinámica inversa y operador de p-alcanzabilidad. Podemos caracterizar a el conjunto $P^{\mathcal{A}}$ en términos de puntos fijos utilizando un operador monótono que llamamos *operador de dinámica inversa* [**12**]. A continuación definimos formalmente este operador.

DEFINICIÓN 4.1.5. (***Operador de dinámica inversa***). Sea $\mathcal{A}$ un autómata de Rabin determinista, definimos el operador de dinámica inversa como la función $\theta^{\mathcal{A}} : 2^X \to 2^X$ dada por la regla

$$\begin{aligned}\theta^{\mathcal{A}}(X') := \{x \in X :\ & \text{existe } \Gamma \in \mathcal{C}, \text{ para el cual si } \delta(\sigma, x)!, \text{ entonces} \\ & \delta(\sigma, x) \in X' \text{ , para todo } \sigma \in \Gamma \text{ y } \delta(\sigma, x), \text{ para algún } \sigma \in \Gamma\}\end{aligned}$$

El operador de dinámica inversa es monótono como lo demuestra la siguiente proposición.

PROPOSICIÓN 4.1.6. *Para todo autómata de Rabin determinista $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ y cualesquiera $X_1, X_2 \subseteq X$, si $X_1 \subseteq X_2$, entonces*

$$\theta^{\mathcal{A}}(X_1) \subseteq \theta^{\mathcal{A}}(X_2)$$

DEMOSTRACIÓN. Sea $x \in \theta^{\mathcal{A}}(X_1)$. Entonces, existe un $\Gamma \in \mathcal{C}$, tal que si $\delta(\sigma, x)!$, entonces $\delta(\sigma, x) \in X_1$, para todo $\sigma \in \Gamma$. Sea $x' = \delta(\sigma, x)$ tal que $x' \in X_1$. Como $X_1 \subseteq X_2$, entonces $x' \in X_2$. Por lo tanto, $x \in \theta^{\mathcal{A}}(X_2)$, ya que $\delta(\sigma, x) = x'$ □

Debido a que el operador de diñamica inversa es monótono, tiene puntos fijos mínimos y puntos fijos máximos. En particular, el conjunto $P^{\mathcal{A}}$ es un punto fijo de $\theta(P^{\mathcal{A}})$.

PROPOSICIÓN 4.1.7. *Sea $\mathcal{A}$ un autómata de Rabin determinista, entonces*

$$P^{\mathcal{A}} = \theta^{\mathcal{A}}(P^{\mathcal{A}})$$

DEMOSTRACIÓN.

($\subseteq$)

Sea $x \in P^{\mathcal{A}}$ Por la definición 4.1.1, existe un controlador supervisor $f : \Sigma^* \to \mathcal{C}$ y para toda $k \in \Sigma^*$ generada por A_x bajo f, existe un $\sigma \in \Sigma$ tal que $k\sigma$ es generada por $\mathcal{A}_x$ bajo f. Por lo tanto, existe un $\Gamma \in \mathcal{C}$ tal que $\sigma \in \Gamma$ y $\delta(k\sigma, x)!$ para toda $k \in \Sigma^*$ En particular $\delta(1\sigma, x) = \delta(\sigma, x)$ está definida.

Mostraremos ahora que para todo $\sigma \in \Gamma$ se cumple que, si $\delta(\sigma, x)$ está definida, entonces $\delta(\sigma, x) \in P^{\mathcal{A}}$.

Supongamos que $\delta(\sigma, x)$ está definida para algún $\sigma \in \Gamma$ y sea $\delta(\sigma, x) = x_1$. Supongamos ahora que $s \in \Sigma^\omega$ es una palabra generada por $\mathcal{A}_{x_1}$ bajo f y probaremos que s es aceptada por $\mathcal{A}_{x_1}$.

Puesto que $x \in P^{\mathcal{A}}$, $\delta(\sigma, x) = x_1$ y s es una palabra generada por $\mathcal{A}_{x_1}$ bajo f, existe una palabra $s' \in \Sigma^\omega$, tal que $s' = \sigma s$, generada por $\mathcal{A}_x$ bajo f que es aceptada por $\mathcal{A}_x$. Por lo tanto, s es aceptada por $\mathcal{A}_{x_1}$.

Por ser s una palabra cualquiera generada por $\mathcal{A}_{x_1}$ bajo f, se deduce que toda $s \in \Sigma^\omega$ generada por $\mathcal{A}_{x_1}$ bajo f es aceptada por $\mathcal{A}_{x_1}$.

Supongamos ahora que k es generada por $\mathcal{A}_{x_1}$ bajo f. Probaremos que $k\sigma'$ es generada por $\mathcal{A}_{x_1}$ bajo f para algún $\sigma' \in \Sigma$.

Como $x \in P^{\mathcal{A}}$ y $\delta(\sigma, x) = x_1$, existe una palabra $k' \in \Sigma^*$ tal que $k' = \sigma k$ y $k'\sigma'$ es generada por $\mathcal{A}_x$ bajo f para algún $\sigma' \in \Sigma$. Esto implica que $k\sigma'$ es una palabra generada por $\mathcal{A}_{x_1}$ bajo f.

Por ser k una palabra cualquiera generada por $\mathcal{A}_{x_1}$ bajo f, se deduce que para toda $k \in \Sigma^*$ generada por $\mathcal{A}_{x_1}$ bajo f, existe un $\sigma' \in \Sigma$ tal que $k\sigma'$ es generada por $\mathcal{A}_{x_1}$ bajo f.

Por lo tanto, $x_1 \in P^{\mathcal{A}}$. Esto prueba que $x \in \theta^{\mathcal{A}}(P^{\mathcal{A}})$.

($\supseteq$)

Sea $x \in \theta^{\mathcal{A}}(P^{\mathcal{A}})$, entonces $\delta(\sigma, x)$ está definida para algún $\sigma \in \Sigma$. En particular $\delta(\sigma, x) \in P^{\mathcal{A}}$. Sea $\delta(\sigma, x) = x_1$ y supongamos que existe una palabra $s \in \Sigma^\omega$ generada por $\mathcal{A}_x$ bajo f Probaremos que s es aceptada por $\mathcal{A}_x$.

Supongamos que $s = \sigma_0\sigma_1\sigma_2....$ Entonces, s tiene un camino π tal que $\pi(1) = x$, $\pi(\sigma_0) = x_1$, $\pi(\sigma_0\sigma_1) = x_2.....$ y $\sigma_j \in \Gamma$ para todo número natural $j \geq 0$. Como $x \in \theta^{\mathcal{A}}(P^{\mathcal{A}})$ y s es generada por $\mathcal{A}_x$ bajo f, entonces $\pi(\sigma_0) \in P^{\mathcal{A}}$ y por lo tanto, existe un $s' = s/\sigma_0$ que es generada por $\mathcal{A}_{x_1}$ bajo f y también aceptada por $\mathcal{A}_{x_1}$. Por ser s' aceptada por $\mathcal{A}_{x_1}$ entonces s es aceptada por $\mathcal{A}_x$.

Por ser s una palabra cualquiera generada por $\mathcal{A}_x$ bajo f se deduce que toda $s \in \Sigma^\omega$ generada por $\mathcal{A}_x$ bajo f es aceptada por $\mathcal{A}_x$.

Mediante un argumento similar obtenemos que para toda $k \in \Sigma^*$ generada por $\mathcal{A}_x$ bajo f, existe un $\sigma' \in \Sigma$ tal que $k\sigma'$ es generada por $\mathcal{A}_x$ bajo f

Por lo tanto, $x \in P^{\mathcal{A}}$. □

La siguiente proposición muestra las propiedades del operador de dinámica inversa bajo las operaciones de la definición 4.1.3.

PROPOSICIÓN 4.1.8. *Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ un autómata de Rabin determinista. Para todo $x \in X$ y cualesquiera $X', X'' \subseteq X$ se cumple que*

a) $\theta^{\mathcal{A}}(X') \cap Deg(\mathcal{A}) = X' \cap Deg(\mathcal{A})$

b) $\theta^{\mathcal{A}(\hookrightarrow X'')}(X') \setminus X'' = \theta^{\mathcal{A}}(X') \setminus X''$

c) $\theta^{\mathcal{A}\uparrow X''}(X') \cap X'' = \theta^{\mathcal{A}}(X') \cap X''$

d) Para todo $p \in P$, $\theta^{\mathcal{A}\downarrow p}(X') \cap (I_p \cup Deg(\mathcal{A})) = \theta^{\mathcal{A}}(X') \cap (I_p \cup Deg(\mathcal{A}))$

DEMOSTRACIÓN.

Parte a)

($\subseteq$)

Sea $x \in \theta^{\mathcal{A}}(X') \cap Deg(\mathcal{A})$. Entonces, $x \in \theta^{\mathcal{A}}(X')$ y $x \in Deg(\mathcal{A})$. Luego, existe un $\sigma \in \Sigma$ tal que $\delta(\sigma, x)!$ y $\delta(\sigma, x) = x$, para todo $\sigma \in \Sigma$. Pero como $x \in \theta^{\mathcal{A}}(X')$, entonces existe un $\Gamma \in \mathcal{C}$ tal que para todo $\sigma \in \Gamma$, $\delta(\sigma, x) \in X'$ Luego, $x \in X'$ Por lo tanto, $x \in X' \cap Deg(\mathcal{A})$.

($\supseteq$)

Sea $x \in X' \cap Deg(\mathcal{A})$. Entonces, $x \in X'$ y $x \in Deg(\mathcal{A})$. Luego, existe un $\sigma \in \Sigma$ tal que $\delta(\sigma, x)!$ y para todo $\sigma \in \Sigma$, $\delta(\sigma, x) = x$. Esto implica que $x \in \theta^{\mathcal{A}}(X')$.

Parte b)

($\subseteq$)

Sea $x \in \theta^{\mathcal{A}(\hookrightarrow X'')}(X') \setminus X''$. Entonces, $x \notin X''$ y existe un autómata de Rabin $\mathcal{A}(\hookrightarrow X'') := (\Sigma, X, \delta', x_0, \{(R'_q, I'_q) : p \in P\})$ tal que para todo $\sigma \in \Sigma$, si $\delta'(\sigma, x)!$, entonces $\delta(\sigma, x) = \delta'(\sigma, x)$. Luego, $x \in \theta^{\mathcal{A}}(X')$ y como $x \notin X''$, entonces $x \in \theta^{\mathcal{A}}(X') \setminus X''$.

($\supseteq$)

Sea $x \in \theta^{\mathcal{A}}(X') \setminus X''$. Entonces, $x \notin X''$ y $x \in \theta^{\mathcal{A}}(X')$. Sea $\mathcal{A}(\hookrightarrow X'') := (\Sigma, X, \delta', x_0, \{(R'_q, I'_q) : p \in P\})$ un autómata de Rabin. Como $x \notin X''$, entonces para todo $\sigma \in \Sigma$ tal que $\delta(\sigma, x)!$, entonces $\delta(\sigma, x) = \delta'(\sigma, x)$. Esto implica que $x \in \theta^{\mathcal{A}(\hookrightarrow X'')}(X')$. Por lo tanto, $x \in \theta^{\mathcal{A}(\hookrightarrow X'')}(X') \setminus X''$

Parte c)

($\subseteq$)

Sea $x \in \theta^{\mathcal{A} \uparrow X''}(X') \cap X''$ Entonces, $x \in X''$ y $x \in \theta^{\mathcal{A} \uparrow X''}(X')$. Luego, existe un autómata de Rabin $\mathcal{A} \uparrow X'' := (\Sigma, X, \delta', x_0, \{(R'_q, I'_q) : p \in P\})$ tal que para todo $\sigma \in \Sigma$ tal que $\delta'(\sigma, x)!$, entonces $\delta(\sigma, x) = \delta'(\sigma, x)$. Esto implica que $x \in \theta^{\mathcal{A}}(X')$, pero $x \in X''$ Por lo tanto, $x \in \theta^{\mathcal{A}}(X') \cap X''$

($\supseteq$)

Sea $x \in \theta^{\mathcal{A}}(X') \cap X''$. Entonces $x \in \theta^{\mathcal{A}}(X')$ y $x \in X''$. Sea $A \uparrow X'' := (\Sigma, X, \delta', x_0, \{(R'_q, I'_q) : p \in P\})$ un autómata de Rabin. Como $x \in X''$, entonces para todo $\sigma \in \Sigma$, si $\delta(\sigma, x)!$ entonces $\delta(\sigma, x) = \delta'(\sigma, x)$. Esto implica que $x \in \theta^{\mathcal{A} \uparrow X''}(X')$. Por lo tanto, $x \in \theta^{\mathcal{A} \uparrow X''}(X') \cap X''$

Parte d)

$$\begin{aligned}
&\theta^{\mathcal{A} \downarrow p}(X') \cap (I_p \cup Deg(\mathcal{A})) \\
&= \theta^{\mathcal{A} \uparrow (I_p \cup Deg(\mathcal{A}))}(X') \cap (I_p \cup Deg(\mathcal{A})) \\
&= \theta^{\mathcal{A}}(X') \cap (I_p \cup Deg(\mathcal{A})) \qquad \text{(parte (c))}
\end{aligned}$$

□

Con el fin de definir una manera de expresar el conjunto alcanzable del autómata $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$, para un par $p \in P$, definimos el siguiente operador.

DEFINICIÓN 4.1.9. (***Opertador de p-alcanzabilidad***) [12].
Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ un autómata de Rabin determinista. Para todo $p \in P$, definimos el operador de *p-alcanzabilidad* como la función $\rho_p^{\mathcal{A}} : (2^X)^2 \to 2^X$ dada por la regla

$$\rho_p^{\mathcal{A}}(X_1, X_2) := \mu X_3.[\theta^{\mathcal{A}}(X_1) \cup [\theta^{\mathcal{A}}(X_1 \cup X_2 \cup X_3) \cap I_p]]$$

De acuerdo a la definición el operador de p-alcanzabilidad es un punto fijo mínimo, donde $\rho_p^{\mathcal{A}}(X_1, X_2)$ representa el conjunto de estados desde los cuales el autómata $\mathcal{A}$ puede ser controlado para alcanzar los estados del conjunto X_1 inmediatamente después de dejar I_p o alcanzar los estados del conjunto X_2 a través de estados que pertenecen al conjunto I_p.

4.1.2. Punto fijo del conjunto $P^{\mathcal{A}}$. En esta sección caracterizamos al conjunto $P^{\mathcal{A}}$ en términos de puntos fijos utilizando los operadores de dinámica inversa y p-alcanzabilidad. Definimos formalmente esta caracterización como sigue:

DEFINICIÓN 4.1.10. [12](***Conjunto de estados controlables del autómata***).
Definimos el *conjunto de estados controlables* $C^{\mathcal{A}} \subseteq X$ como

$$C^{\mathcal{A}} := \begin{cases} \mu X_1.\nu X_2.\rho_p^{\mathcal{A}}(X_1, X_2 \cap R_p) & \text{si } \mid P \mid = 1 \\ \mu X_1.[\bigcup_{p \in P} \nu X_2.\rho_p^{\mathcal{A}}(X_1, C^{\mathcal{A}(\hookrightarrow X_1 \cup (X_2 \cap R_p)) \downarrow p})] & \text{si } \mid P \mid > 1 \end{cases}$$

Si $\mid P \mid = 1$, la idea es primero calcular el conjunto más grande de estados desde los cuales el autómata puede ser controlado para alcanzar los estados del conjunto R_p por medio de I_p, para después calcular el conjunto de estados del autómata que pueden ser controlados para entrar a R_p un número infinito de veces y entrar a el conjunto I_p después de un número finito de pasos.

Si $\mid P \mid > 1$, la idea es calcular el conjunto más grande de estados desde los cuales, para algún $p \in P$, el autómata puede ser controlado para entrar a R_p un número infinito de veces y entrar a I_p después de un número finito de pasos o satisfacer la condición de aceptación excluyendo al par (R_p, I_p).

PROPOSICIÓN 4.1.11. [11] *Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ un autómata de Rabin determinista, para todo $p \in P$, todo $x \in X$ y cualesquiera $X_1, X_2 \subseteq X$ se cumple que*

a) $C^{\mathcal{A}} \cap Deg(\mathcal{A}) = \bigcup_{p \in P}(R_p \cap I_p) \cap Deg(\mathcal{A})$

b) $C^{\mathcal{A}} \cup X_1 \subseteq C^{\mathcal{A}(\hookrightarrow X_1)}$

c) $C^{\mathcal{A}} = C^{\mathcal{A}(\hookrightarrow X_1)}$ *si y sólo si* $X_1 \subseteq C^{\mathcal{A}}$

d) $C^{\mathcal{A}\uparrow X_1} \subseteq C^{\mathcal{A}} \cap X_1$

e) $C^{\mathcal{A}\downarrow p} \subseteq C^{\mathcal{A}} \cap (I_p \cup Deg(\mathcal{A}))$

f) $C^{\mathcal{A}} = \theta^{\mathcal{A}}(C^{\mathcal{A}})$

DEMOSTRACIÓN. Véase [11] apéndice A. □

El siguiente teorema de [11] establece que para todo $x \in C^{\mathcal{A}}$ existe un supervisor f que controla al autómata de Rabin $\mathcal{A}$ a la satisfacción de su condición de aceptación[1] Además es muy importante, ya que utilizaremos una parte de la demostración para definir el supervisor.

TEOREMA 4.1.12. *Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ un autómata de Rabin determinista. Entonces, existe una función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$, tal que para todo $x \in C^{\mathcal{A}}$, la función $f_x : \Sigma^* \to \mathcal{C}$ definida por la regla $f_x(l) := \Phi^{\mathcal{A}}(\delta(l, x))$ es un controlador supervisor completo y no bloqueante para (L_x, S_x) tal que $(S_x)^{f_x} \subseteq E_x$, donde*

- $E_x \subseteq \Sigma^\omega$ es el ω-lenguaje aceptado por el autómata de Rabin $(\Sigma, X, \delta, x, \{(R_p, I_p) : p \in P\})$,
- $L_x \subseteq \Sigma^*$ es el $_*$-lenguaje aceptado por el $_*$-autómata $(\Sigma, X, \delta, x, X)$ y
- $S_x := lim(L_x)$ [2]

DEMOSTRACIÓN. Construiremos la función $\Phi^{\mathcal{A}}$ utilizando inducción sobre $\mid P \mid$. Supongamos que $\mid P \mid = 1$. Entonces, $C^{\mathcal{A}} = \mu X_1.\nu X_2.\rho_p^{\mathcal{A}}(X_1, X_2 \cap R_p)$. Por el teorema 2.3.5, este punto fijo es la mínima cota superior de una secuencia creciente $C_0^{\mathcal{A}} \subseteq C_1^{\mathcal{A}} \subseteq C_2^{\mathcal{A}} \subseteq \ldots$ definida como sigue:

[1]Este resultado nos lleva a la conclusión de que $C^{\mathcal{A}}$ coincide con $P^{\mathcal{A}}$.

[2]Esto se debe a que $S = lim(L)$ y x es el estado de referencia para los autómata que reconocen los lenguajes L, S y E.

$$
\begin{aligned}
C_0^{\mathcal{A}} &:= \emptyset \\
C_{i+1}^{\mathcal{A}} &:= \nu X_2.\rho_p^{\mathcal{A}}(C_i^{\mathcal{A}}, X_2 \cap R_p) \\
&= \rho_p^{\mathcal{A}}(C_i^{\mathcal{A}}, C_{i+1}^{\mathcal{A}} \cap R_p) \\
&= \mu X_1.[\theta^{\mathcal{A}}(C_i^{\mathcal{A}}) \cup [\theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup X_1) \cap I_p]]
\end{aligned}
$$

Cada $C_{i+1}^{\mathcal{A}}$ es la mínima cota superior de una secuencia creciente $C_{i+1,0}^{\mathcal{A}} \subseteq C_{i+1,1}^{\mathcal{A}} \subseteq C_{i+1,2}^{\mathcal{A}} \subseteq ...$ definida como sigue:

$$
\begin{aligned}
C_{i+1,0}^{\mathcal{A}} &:= \emptyset \\
C_{i+1,j+1}^{\mathcal{A}} &:= \theta^{\mathcal{A}}(C_i^{\mathcal{A}}) \cup [\theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup C_{i+1,j}^{\mathcal{A}}) \cap I_p]] \\
&= C_{i+1,j+1,0}^{\mathcal{A}} \cup C_{i+1,j+1,1}^{\mathcal{A}}
\end{aligned}
$$

donde

$$
\begin{aligned}
C_{i+1,j+1,0}^{\mathcal{A}} &:= \theta^{\mathcal{A}}(C_i^{\mathcal{A}}) \\
C_{i+1,j+1,1}^{\mathcal{A}} &:= \theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup C_{i+1,j}^{\mathcal{A}}) \cap I_p.
\end{aligned}
$$

Por lo tanto,

$C^{\mathcal{A}} = \bigcup_{i=1}^{|x|} C_i^{\mathcal{A}} = \bigcup_{i,j=1}^{|x|} C_{i,j}^{\mathcal{A}} = \bigcup_{i,j=1}^{|x|} \bigcup_{k=0,1} C_{i,j,k}^{\mathcal{A}}$.

Definimos una función total $\Phi_{i,j,k}^{\mathcal{A}} : C_{i,j,k}^{\mathcal{A}} \to \mathcal{C}$, para todo componente en $C^{\mathcal{A}}$,

si $k = 0$, entonces definimos $\Phi_{i+1,j+1,0}^{\mathcal{A}} : C_{i+1,j+1,0}^{\mathcal{A}} \to \mathcal{C}$ tal que, para todo $x \in C_{i+1,j+1,0}^{\mathcal{A}}$ se cumple que $\delta(\sigma, x) \in C_i^{\mathcal{A}}$, para todo $\sigma \in \Phi_{i+1,j+1,0}^{\mathcal{A}}(x)$;

si $k = 1$, entonces definimos $\Phi_{i+1,j+1,1}^{\mathcal{A}} : C_{i+1,j+1,1}^{\mathcal{A}} \to \mathcal{C}$ tal que, para todo $x \in C_{i+1,j+1,1}^{\mathcal{A}} \subseteq I_p$ se cumple que $\delta(\sigma, x) \in C_i^{\mathcal{A}} \cup (C_{i+1} \cap R_p) \cup C_{i+1,j}^{\mathcal{A}}$, para todo $\sigma \in \Phi_{i+1,j+1,k}^{\mathcal{A}}(x)$.

Definamos una función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$ por la regla $\Phi^{\mathcal{A}}(x) := \Phi_{i+1,j+1,k}^{\mathcal{A}}(x)$, donde (i, j, k) es la tripleta mínima ordenada de $N \times N \times \{0, 1\}$ tal que $x \in C_{i+1,j+1,k}^{\mathcal{A}}$.

Mostraremos ahora que el supervisor $f_x : \Sigma^* \to \mathcal{C}$ es completo y no bloqueante para (L_x, S_x) tal que $S_x^{f_x} \subseteq E_x$.

Obsérvese que f_x es un supervisor completo ya que para todo $x \in C^{\mathcal{A}}$, se cumple que $\delta(\sigma, x) \in C^{\mathcal{A}}$, para todo $\sigma \in \Phi^{\mathcal{A}}(x)$, y es no bloqueante por el hecho de que para todo $x \in C^{\mathcal{A}}$ se cumple que existe un $\sigma \in \Phi^{\mathcal{A}}(x)$ tal que $\delta(\sigma, x)!$.

Para probar que $S_x^{f_x} \subseteq E_x$, supongamos que $s \in S_x^{f_x}$. Entonces, existe un camino $\pi : pre(s) \to X$ de s sobre $\mathcal{A}$ tal que $\pi(1) \in C^{\mathcal{A}}$ y para toda $k\sigma \in pre(s)$, $\sigma \in \Phi^{\mathcal{A}}(\pi(k))$.

Mostraremos que $s \in E_x$.

Definimos la función $I_p - rango : C^{\mathcal{A}} \to N$ por la regla $I_p - rango(x) := i$, donde i es el menor número natural tal que $x \in C_i^{\mathcal{A}}$.

Por la definición de $\Phi^{\mathcal{A}}$, $(I_p - rango(\delta(\sigma, x))) \leq (I_p - rango(x))$, para todo $x \in C^{\mathcal{A}}$ y $\sigma \in \Phi^{\mathcal{A}}(x)$ Si $x \notin I_p$, entonces $(I_p - rango(\delta(\sigma, x))) < (I_p - rango(x))$, para todo $\sigma \in \Phi^{\mathcal{A}}(x)$. Por lo tanto, $\Omega_\pi \subseteq I_p$.

Ahora definimos la función $R_p - rango : C^{\mathcal{A}} \to N^2$ por la regla $R_p - rango(x) := (i, j)$, donde (i, j) es el menor par de números naturales tal que $x \in C_{i,j}^{\mathcal{A}}$.

Por la definición de $\Phi^{\mathcal{A}}$, si $x \in C^{\mathcal{A}}$, $\sigma \in \Phi^{\mathcal{A}}(x)$ y $\delta(\sigma, x) \notin R_p$, entonces

$$(R_p - rango(\delta(\sigma, x))) < (R_p - rango(x)).$$

Por lo tanto, $\Omega_\pi \cap R_p \neq \emptyset$, es decir, $s \in E_x$.

Ahora supongamos que $| P | > 1$ y supongamos que el resultado se cumple para todo autómata con menor número de pares. Entonces,

$$C^{\mathcal{A}} = \mu X_1.[\bigcup_{p \in P} \nu X_2.\rho_p^{\mathcal{A}}(X_1, C^{\mathcal{A}(\hookrightarrow X_1 \cup (X_2 \cap R_p)) \downarrow p})].$$

Por el teorema 2.3.5, este punto fijo es la mínima cota superior de una secuencia creciente $C_0^{\mathcal{A}} \subseteq C_1^{\mathcal{A}} \subseteq C_2^{\mathcal{A}} \subseteq \ldots$ definida como sigue:

$$\begin{array}{lcl} C_0^{\mathcal{A}} & := & \emptyset \\ C_{i+1}^{\mathcal{A}} & := & \bigcup_{p \in P} \nu X_2.\rho_p^{\mathcal{A}}(C_i^{\mathcal{A}}, C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup (X_2 \cap R_p)) \downarrow p}) \\ & = & \bigcup_{p \in P} C_{i+1,p}^{\mathcal{A}} \end{array}$$

donde

$$\begin{array}{lcl} C_{i+1,p}^{\mathcal{A}} & := & \nu X_2.\rho_p^{\mathcal{A}}(C_i^{\mathcal{A}}, C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup (X_2 \cap R_p)) \downarrow p}) \\ & = & \rho_p^{\mathcal{A}}(C_i^{\mathcal{A}}, C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup (C_{i+1,p}^{\mathcal{A}} \cap R_p)) \downarrow p}) \\ & = & \mu X_3.[\theta^{\mathcal{A}}(C_i^{\mathcal{A}}) \cup [\theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup C_i^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup (C_{i+1,p}^{\mathcal{A}} \cap R_p)) \downarrow p} \cup X_3) \cap I_p]] \end{array}$$

Cada $C_{i+1,p}^{\mathcal{A}}$ es la mínima cota superior de una secuencia creciente $C_{i+1,p,0}^{\mathcal{A}} \subseteq C_{i+1,p,1}^{\mathcal{A}} \subseteq C_{i+1,p,2}^{\mathcal{A}} \subseteq \ldots$ definida como sigue:

$$\begin{array}{lcl} C_{i+1,p,0}^{\mathcal{A}} & := & \emptyset \\ C_{i+1,p,j+1}^{\mathcal{A}} & := & \theta^{\mathcal{A}}(C_i^{\mathcal{A}}) \cup [\theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup (C_{i+1,p}^{\mathcal{A}} \cap R_p)) \downarrow p} \cup C_{i+1,p,j}^{\mathcal{A}}) \cap I_p]] \\ & = & C_{i+1,p,j+1,0}^{\mathcal{A}} \cup C_{i+1,p,j+1,1}^{\mathcal{A}} \end{array}$$

donde

$$C^{\mathcal{A}}_{i+1,j+1,0} := \theta^{\mathcal{A}}(C^{\mathcal{A}}_i)$$
$$C^{\mathcal{A}}_{i+1,j+1,1} := \theta^{\mathcal{A}}(C^{\mathcal{A}}_i \cup C^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p} \cup C^{\mathcal{A}}_{i+1,p,j}) \cap I_p$$

Por lo tanto,

$C^{\mathcal{A}} = \bigcup_{i=1}^{|x|} \bigcup_{p \in P} \bigcup_{j=1}^{|x|} \bigcup_{k=0,1} C^{\mathcal{A}}_{i,p,j,k}$.

Definimos para cada componente $C^{\mathcal{A}}_{i,p,j,k}$ una función total $\Phi^{\mathcal{A}}_{i,p,j,k} : C^{\mathcal{A}}_{i,p,j,k} \to C$, tal que:

si $k = 0$, entonces $\Phi^{\mathcal{A}}_{i+1,p,j+1,0} : C^{\mathcal{A}}_{i+1,p,j+1,0} \to C$ tal que, para todo $x \in C^{\mathcal{A}}_{i+1,p,j+1,0}$ se cumple que $\delta(\sigma, x) \in C^{\mathcal{A}}_i$, para todo $\sigma \in \Phi^{\mathcal{A}}_{i+1,p,j+1,0}(x)$;

si $k = 1$, entonces definimos $\Phi^{\mathcal{A}}_{i+1,p,j+1,1} : C^{\mathcal{A}}_{i+1,p,j+1,1} \to C$ tal que, para todo $x \in C^{\mathcal{A}}_{i+1,p,j+1,1} \subseteq I_p$ se cumple que $\delta(\sigma, x) \in C^{\mathcal{A}}_i \cup C^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p} \cup C^{\mathcal{A}}_{i+1,p,j}$, para todo $\sigma \in \Phi^{\mathcal{A}}_{i+1,p,j+1,1}(x)$.

Ahora definamos la función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$ como sigue:

$$\Phi^{\mathcal{A}}(x) := \begin{cases} \Phi^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p(x)} & \text{si } x \in C^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p} \setminus C^{\mathcal{A}}_i \cup R_p \\ \Phi^{\mathcal{A}}_{i+1,p,j+1,k(x)} & \text{en otro caso} \end{cases}$$

donde (i, p, j, k) es la menor cuádrupla de números naturales tal que $x \in C^{\mathcal{A}}_{i+1,p,j+1,k}$.

Mostraremos ahora que el supervisor $f_x : \Sigma^* \to \mathcal{C}$ es completo y no bloqueante para (L_x, S_x) tal que $S^{f_x}_x \subseteq E_x$.

Obsérvese que f_x es un supervisor completo y no bloqueante por la definición $\Phi^{\mathcal{A}}$.

Para probar que $S^{f_x}_x \subseteq E_x$, supongamos que $s \in S^{f_x}_x$. Entonces, por la hipótesis de inducción, existe un camino $\pi : pre(s) \to X$ de s sobre $\mathcal{A}$ tal que $\pi(1) \in C^{\mathcal{A}}$ y para toda $k\sigma \in pre(s)$, $\sigma \in \Phi^{\mathcal{A}}(\pi(k))$.

Definimos la función $I - rango : C^{\mathcal{A}} \to N \times P$ por la regla $I - rango(x) := (i, p)$, donde (i, p) es el menor número natural tal que $x \in C^{\mathcal{A}}_{i,p}$.

Nótese que

$C^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p} \setminus (C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p) \cup Deg(\mathcal{A}))$

$= \theta^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p}(C^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p}) \setminus (C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p) \cup Deg(\mathcal{A}))$ (por la proposición 4.1.11 (f))

$= [\theta^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p}(C^{\mathcal{A}(\hookrightarrow C^{\mathcal{A}}_i \cup C^{\mathcal{A}}_{i+1,p} \cap R_p))\downarrow p}) \cap I_p] \setminus (C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,p} \cap R_p) \cup Deg(\mathcal{A}))$(por la proposición 4.1.11 (e))

$= [\theta^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup (C_{i+1,p}^{\mathcal{A}} \cap R_p))}(C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup C_{i+1,p}^{\mathcal{A}} \cap R_p))\downarrow p}) \cap I_p] \setminus (C_i^{\mathcal{A}} \cup (C_{i+1,p}^{\mathcal{A}} \cap R_p) \cup Deg(\mathcal{A}))$ (por la proposición 4.1.8 (d))

$= [\theta^{\mathcal{A}}(C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup C_{i+1,p}^{\mathcal{A}} \cap R_p))\downarrow p}) \cap I_p] \setminus (C_i^{\mathcal{A}} \cup (C_{i+1,p}^{\mathcal{A}} \cap R_p) \cup Deg(\mathcal{A}))$ (por la proposición 4.1.8 (b))

$\subseteq C_{i+1,p}^{\mathcal{A}}$

Por la definición de $\Phi^{\mathcal{A}}$ y por la hipótesis de inducción tenemos que para todo $x \in C^{\mathcal{A}}$ y $\sigma \in \Phi^{\mathcal{A}}(x)$, si $\delta(\sigma, x) \notin Deg(\mathcal{A})$, entonces $(I - rango(\delta(\sigma, x))) \leq (I - rango(x))$. Si $I - rango(x) = (i, p)$ y $x \notin I_p \cup Deg(\mathcal{A})$, entonces, por la proposición 4.1.11(e), $(I - rango(\delta(\sigma, x))) < (I - rango(x))$, para todo $\sigma \in \Phi^{\mathcal{A}}(x)$.

Si (i, p) es el menor par de números naturales tal que $\Omega_\pi \cap C_{i+1,p}^{\mathcal{A}} \neq \emptyset$, entonces

$$\Omega_\pi \subseteq C_{i+1,p}^{\mathcal{A}} \cap (I_p \cup Deg(\mathcal{A})).$$

Nótese que si $\Omega_\pi \cap C_{i+1,p}^{\mathcal{A}} \cap Deg(A) \neq \emptyset$, entonces, por la proposición 4.1.11 (a), $s \in E_x$. Luego, $\Omega_\pi \subseteq I_p$.

Ahora sea $R - rango : C^{\mathcal{A}} \to N \times P \times N$ tal que $R - rango(x) := (i, p, j)$, donde (i, p, j) es el menor tripleta de número naturales tal que $x \in C_{i,p,j}^{\mathcal{A}}$.

Por definición de $\Phi^{\mathcal{A}}$, si $x \in C^{\mathcal{A}}$, $\sigma \in \Phi^{\mathcal{A}}(x)$, $I_p - rango(x) = (i + 1, p)$ y $\delta(\sigma, x) \notin C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup C_{i+1,p}^{\mathcal{A}} \cap R_p))\downarrow p}$, entonces $(R_p - rango(\delta(\sigma, x))) < (I_p - rango(x))$.

Si (i, p) es el menor par de números naturales tal que $\Omega_\pi \cap C_{i+1,p}^{\mathcal{A}} \neq \emptyset$, entonces

$$\Omega_\pi \cap C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup C_{i+1,p}^{\mathcal{A}} \cap R_p))\downarrow p} \neq \emptyset.$$

Pero por la hipótesis de inducción, tenemos que $\Omega_\pi \cap R_p \neq \emptyset$, o $\Omega_\pi \subseteq C^{\mathcal{A}(\hookrightarrow C_i^{\mathcal{A}} \cup C_{i+1,p}^{\mathcal{A}} \cap R_p))\downarrow p} \setminus (C_i^{\mathcal{A}} \cup R_p)$. □

4.1.3. La complejidad de computar el conjunto $C^{\mathcal{A}}$. La complejidad de computar $C^{\mathcal{A}}$ es exponencial en el tamaño de la condición de aceptación del autómata de Rabin y polinomial en el tamaño de los estados como lo establece la siguiente proposición.

PROPOSICIÓN 4.1.13. [12] *El conjunto controlable de un autómata de Rabin determinista $\mathcal{A}$ puede ser computado en tiempo $\bigcirc(kl(mn)^{3m})$, donde k es el número de* *patrones de control, l es la cardinalidad el alfabeto, m es el número de pares en la condición de aceptación y n es el número de estados.*

DEMOSTRACIÓN. La demostración es un resultado directo del análisis de la computación de los puntos fijos y de la definición 4.1.10. Para más detalle véase **[11]**. □

4.1.4. Optimización de la computación del conjunto $C^{\mathcal{A}}$. Para reducir la complejidad de la computación del conjunto controlable, podemos redefinir $C^{\mathcal{A}}$ de manera más simple utilizando la estrategia de divide y vencerás. La idea es computar el conjunto controlable de cada par $p \in P$ por separado y posteriormente hacer la unión de estos conjuntos. De esta manera podemos obtener el conjunto controlable completo en menor tiempo. A continuación definimos formalmente esta idea.

DEFINICIÓN 4.1.14. Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi determinista. Definimos el conjunto $C^{\mathcal{A}}$ como

$$C^{\mathcal{A}} := \begin{cases} C_p^{\mathcal{A}} & \text{si } \mid P \mid = 1 \\ \bigcup_{p \in P} C_p^{\mathcal{A}} & \text{si } \mid P \mid > 1 \end{cases}$$

donde $C_p^{\mathcal{A}} := \mu X_1.\nu X_2.\rho_p^{\mathcal{A}}(X_1, X_2 \cap R_p)$ representa el conjunto supremo de estados desde los cuales el sistema puede ser controlado para satisfacer la condición de aceptación del par $p \in P$.

Intuitivamente, el conjunto $\bigcup_{p \in P} C_p^{\mathcal{A}}$ representa el conjunto supremo de estados desde los cuales el sistema puede ser controlado para satisfacer la condición de aceptación del autómata de Rabin $\mathcal{A}$.

Para todo $x \in C^{\mathcal{A}}$ existe un controlador supervisor no bloqueante y completo como lo demuestra el siguiente teorema.

TEOREMA 4.1.15. *Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\})$ un autómata de Rabin determinista. Entonces, existe una función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$, tal que para todo $x \in C^{\mathcal{A}}$, la función $f_x : \Sigma^* \to \mathcal{C}$ definida por la regla $f_x(l) := \Phi^{\mathcal{A}}(\delta(l, x))$ es un controlador supervisor completo y no bloqueante para (L_x, S_x) tal que $(S_x)^{f_x} \subseteq E_x$, donde :*

- $E_x \subseteq \Sigma^\omega$ es el ω-lenguaje aceptado por el autómata de Rabin $(\Sigma, X, \delta, x, \{(R_p, I_p) : p \in P\})$,
- $L_x \subseteq \Sigma^*$ es el $_*$-lenguaje aceptado por el $_*$-autómata $(\Sigma, X, \delta, x, X)$ y
- $S_x := lim(L_x)$[3]

DEMOSTRACIÓN.

Si $P = 1$, la demostración es exactamente igual que la del teorema 4.1.12.

Si $P > 1$, entonces para todo $x \in C^{\mathcal{A}}$, definimos la función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$ por la siguiente regla:

$$\Phi^{\mathcal{A}}(x) := \bigcup_{p \in P} \Phi_p^{\mathcal{A}}(x)$$

donde $\Phi_p^{\mathcal{A}}(x)$ es un controlador supervisor completo y no bloqueante para el par (R_p, I_p) y está definido como en el teorema 4.1.12.

Mostraremos que el correspondiente supervisor $f_x : \Sigma^* \to \mathcal{C}$ es completo y no bloqueante para (L_x, S_x) tal que $S_x^{f_x} \subseteq E_x$.

El supervisor f_x es completo y no bloqueante por la proposición 3.5.9, debido a que $\Phi_p^{\mathcal{A}}(x)$ es un supervisor completo y no bloqueante, para todo $p \in P$.

Ahora probaremos que $S_x^{f_x} \subseteq E_x$.

Sea $s \in S_x^{f_x}$. Entonces, existe un camino $\pi : pre(s) \to X$ de s sobre $\mathcal{A}$ tal que $\pi(1) \in C^{\mathcal{A}}$ y para toda $k\sigma \in pre(s)$, $\sigma \in \Phi^{\mathcal{A}}(\pi(k))$.

Como $x \in C^{\mathcal{A}}$, entonces $x \in \bigcup_{p \in P} C_p^{\mathcal{A}}$. Esto significa que $x \in C_p^{\mathcal{A}}$ para algún $p \in P$. Sea $E_x^p \subseteq \Sigma^\omega$ el ω-lenguaje aceptado por el autómata de Rabin $(\Sigma, X, \delta, x, \{(R_p, I_p)\})$ que consiste del único par $p \in P$. Entonces, $s \in E_x^p$ por el teorema 4.1.12. Como el par $p \in P$ es un par de aceptación del autómata de Rabin completo $(\Sigma, X, \delta, x, \{(R_p, I_p) : p \in P\})$, entonces $s \in E_x$. □

Para probar que $P^{\mathcal{A}}$ corresponde con $C^{\mathcal{A}}$, definamos ahora $P^{\mathcal{A}_p}$ como el conjunto de estados controlables del autómata de Rabin determinista $\mathcal{A}$, designando a p como el único par de aceptación del autómata. Entonces, por lo resultados de **[13, 11]**, tenemos que si $\mid P \mid = 1$, $P^{\mathcal{A}_p} \subseteq C^{\mathcal{A}_p}$ Esto implica que $\bigcup_{p \in P} P^{\mathcal{A}_p} \subseteq \bigcup_{p \in P} C^{\mathcal{A}_p}$. El resultado

[3]Esto se debe a que $S = lim(L)$ y x es el estado de referencia para los autómata que reconocen los lenguajes L, S y E.

anterior junto con el teorema 4.1.15 prueban que nuestra definición es equivalente a la definición 4.1.10.

La complejidad de computar $C^{\mathcal{A}}$ es polinomial en el tamaño de los estados como lo establece la siguiente proposición.

PROPOSICIÓN 4.1.16. *El conjunto controlable de un autómata de Rabin determinista $\mathcal{A}$ puede ser computado en tiempo $\bigcirc(kln^3m)$, donde k es el número de patrones de control, l es la cardinalidad el alfabeto, m es el número de pares en la condición de aceptación y n es el número de estados.*

DEMOSTRACIÓN. La demostración es un resultado directo de la aplicación de la proposición 4.1.13 y de la definición 4.1.14. □

Por lo descrito arriba, la complejidad de computar 4.1.14 no es afectada por el número de pares en la condición de aceptación del autómata de Rabin, como en el caso de 4.1.10, lo que representa una gran ventaja.

4.2. Propiedades de Vivacidad en el Modelo del Sistema

Podemos extender los conceptos de la sección 4.1 para incluir el caso cuando S no es ω-cerrado [**11**]. Bajo estas nuevas condiciones, la idea es agregar al conjunto controlable aquellos estados que forman caminos cuyas palabras no son aceptadas por el autómata que reconoce el lenguaje S.

Asumimos que es dado un autómata de Rabin-Büchi $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ tal que el autómata de Büchi $(\Sigma, X, \delta, x_0, R)$ reconoce el ω-lenguaje S.

Igual que en el caso anterior podemos representar el conjunto $C^{\mathcal{A}}$ en términos de $P^{\mathcal{A}}$

DEFINICIÓN 4.2.1. (*Conjunto controlable del autómata* $\mathcal{A}$). Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi. Definimos $P^{\mathcal{A}}$ como el conjunto de todos los $x \in X$ para los cuales existe una función $f : \Sigma^* \to \mathcal{C}$ tal que:

1. toda $s \in \Sigma^\omega$ generada por $\mathcal{A}_x$ bajo f es aceptada por $\mathcal{A}_x$ y

2. para toda $k \in \Sigma^*$ generada por $\mathcal{A}_x$ bajo f, existe un $\sigma \in \Sigma$ tal que $k\sigma$ es generada por $\mathcal{A}_x$ bajo f.

También tenemos que extender la definición 4.1.3 como sigue.

DEFINICIÓN 4.2.2. Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi. Para todo $X' \subseteq X$ y todo par $p \in P$ extendemos la definición 4.1.3 de la sección anterior como sigue:

a) en la definición <u>auto-lazo de un conjunto</u> $X' \subseteq X$ reemplazamos el conjunto R por $(R \cup X')$,

b) en la definición <u>restricción a un conjunto</u> $X' \subseteq X$ reemplazamos el conjunto R por $R \cup (X \setminus X')$ y

c) en la definición <u>exclusión de un par</u> $p \in P$ reemplazamos el conjunto R por $R \cup (X \setminus (I_p \cup Deg(\mathcal{A})))$.

4.2.1. Operador de dinámica inversa y operador de p-alcanzabilidad. La principal diferencia de la sección anterior y esta sección es la definición del operador de dinámica inversa, es por medio de este operador que incluiremos aquellos estados del autómata que forman caminos que no son aceptados por el autómata $(\Sigma, X, \delta, x_0, R)$.

DEFINICIÓN 4.2.3. Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi determinista. Definimos la función $\theta^{\mathcal{A}} : (2^x)^2 \to 2^x$ por la regla

$$\begin{aligned}\theta^{\mathcal{A}}(X_1, X_2) := \{x \in X : & \text{existe } \Gamma \in \mathcal{C}, \text{ para el cual si } \delta(\sigma, x)!, \text{ entonces} \\ & \delta(\sigma, x) \in X_1 \cup X_2, \text{ para todo } \sigma \in \Gamma, \ \delta(\sigma, x)! \text{ y } \delta(\sigma, x) \in X_1, \text{ para algún } \sigma \in \Gamma\ \}\end{aligned}$$

El conjunto $\theta^{\mathcal{A}}(X_1, X_2)$ consiste de aquellos estados desde los cuales el autómata $\mathcal{A}$ puede ser controlado para inmediatamente alcanzar al conjunto $X_1 \cup X_2$ siempre y cuando alcance al conjunto X_1.

DEFINICIÓN 4.2.4. (*Operador de dinámica inversa*). Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi determinista. Definimos el *operador de dinámica inversa* $\Theta^{\mathcal{A}} : (2^X)^2 \to 2^X$ por la regla:

$$\Theta^{\mathcal{A}}(X_1, X_2) := \nu X_3.\mu X_4.[\theta^{\mathcal{A}}(X_1 \cup (X_4 \setminus R), X_3 \setminus R) \cap X_2]$$

Si $R = X = X_2$, entonces $\Theta^{\mathcal{A}}$ es equivalente al operador de dinámica inversa de la sección anterior.

DEFINICIÓN 4.2.5. (*Operador de p-alcanzabilidad*). Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi determinista. Para todo $p \in P$, definimos el operador de *p-alcanzabilidad* $\rho_p^{\mathcal{A}} : (2^X)^2 \to 2^X$ por la regla:

$$\rho_p^{\mathcal{A}}(X_1, X_2) := \mu X_3.[\Theta^{\mathcal{A}}(X_1, X) \cup \Theta^{\mathcal{A}}(X_1 \cup X_2 \cup X_3, I_p)]$$

De acuerdo a la definición, el operador de p-alcanzabilidad es un punto fijo mínimo, donde $\rho_p^{\mathcal{A}}(X_1, X_2)$ representa el conjunto de estados desde los cuales el autómata $\mathcal{A}$ puede ser controlado para alcanzar los estados del conjunto X_1 inmediatamente después de dejar I_p y entrar a R, o alcanzar los estados del conjunto X_2 a través de estados que pertenecen al conjunto I_p.

El conjunto controlable $C^{\mathcal{A}}$ se define de la misma manera que en la sección anterior.

DEFINICIÓN 4.2.6. Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi determinista. Definimos el conjunto $C^{\mathcal{A}}$ como

$$C^{\mathcal{A}} := \begin{cases} C_p^{\mathcal{A}} & \text{si } \mid P \mid = 1 \\ \bigcup_{p \in P} C_p^{\mathcal{A}} & \text{si } \mid P \mid > 1 \end{cases}$$

donde $C_p^{\mathcal{A}} := \mu X_1.\nu X_2.\rho_p^{\mathcal{A}}(X_1, X_2 \cap R_p)$ representa el conjunto supremo de estados desde los cuales el sistema puede ser controlado para satisfacer la condición de aceptación del par $p \in P$.

TEOREMA 4.2.7. *Sea $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, R)$ un autómata de Rabin-Büchi determinista. Entonces, existe una función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$, tal que para todo $x \in C^{\mathcal{A}}$, la función $f_x : \Sigma^* \to \mathcal{C}$ definida por la regla $f_x(l) := \Phi^{\mathcal{A}}(\delta(l, x))$ es un controlador supervisor completo y no bloqueante para (L_x, S_x) tal que $(S_x)^{f_x} \subseteq E_x$, donde*

- *$E_x \subseteq \Sigma^\omega$ es el ω-lenguaje aceptado por el autómata de Rabin $(\Sigma, X, \delta, x, \{(R_p, I_p) : p \in P\})$,*

- $L_x \subseteq \Sigma^*$ es el $_*$-lenguaje aceptado por el $_*$-autómata $(\Sigma, X, \delta, x, X)$ y
- S_x es el ω-lenguaje aceptado por el autómata de Büchi $(\Sigma, X, \delta, x, R)$.

DEMOSTRACIÓN. Construiremos la función $\Phi^{\mathcal{A}}$ utilizando inducción sobre $\mid P \mid$. Supongamos que $\mid P \mid= 1$. Entonces, $C^{\mathcal{A}} = \mu X_1.\nu X_2.\rho_p^{\mathcal{A}}(X_1, X_2 \cap R_p)$. Por el teorema 2.3.5, este punto fijo es la minima cota superior de una secuencia creciente $C_0^{\mathcal{A}} \subseteq C_1^{\mathcal{A}} \subseteq C_2^{\mathcal{A}} \subseteq \ldots$ definida como sigue:

$$\begin{aligned}
C_0^{\mathcal{A}} &:= \emptyset \\
C_{i+1}^{\mathcal{A}} &:= \nu X_2.\rho_p^{\mathcal{A}}(C_i^{\mathcal{A}}, X_2 \cap R_p) \\
&= \rho_p^{\mathcal{A}}(C_i^{\mathcal{A}}, C_{i+1}^{\mathcal{A}} \cap R_p) \\
&= \mu X_3.[\Theta^{\mathcal{A}}(C_i^{\mathcal{A}}, X) \cup \Theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup X_3, I_p)]
\end{aligned}$$

Cada $C_{i+1}^{\mathcal{A}}$ es la mínima cota superior de una secuencia creciente $C_{i+1,0}^{\mathcal{A}} \subseteq C_{i+1,1}^{\mathcal{A}} \subseteq C_{i+1,2}^{\mathcal{A}} \subseteq \ldots$ definida como sigue:

$$\begin{aligned}
C_{i+1,0}^{\mathcal{A}} &:= \emptyset \\
C_{i+1,j+1}^{\mathcal{A}} &:= \Theta^{\mathcal{A}}(C_i^{\mathcal{A}}, X) \cup \Theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup C_{i+1,j}^{\mathcal{A}}, I_p) \\
&= C_{i+1,j+1,0}^{\mathcal{A}} \cup C_{i+1,j+1,1}^{\mathcal{A}}
\end{aligned}$$

donde

$$\begin{aligned}
C_{i+1,j+1,0}^{\mathcal{A}} &:= \Theta^{\mathcal{A}}(C_i^{\mathcal{A}}, X) \\
&= \nu X_3.\mu X_4.[\theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (X_4 \setminus R), X_3 \setminus R) \cap X] \\
&= \mu X_4.[\theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (X_4 \setminus R), C_{i+1,j+1,0}^{\mathcal{A}} \setminus R) \cap X]
\end{aligned}$$

y

$$\begin{aligned}
C_{i+1,j+1,1}^{\mathcal{A}} &:= \Theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup C_{i+1,j}^{\mathcal{A}}, I_p) \\
&= \nu X_3.\mu X_4.[\theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup C_{i+1,j}^{\mathcal{A}} \cup (X_4 \setminus R), X_3 \setminus R) \cap I_p] \\
&= \mu X_4.[\theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup C_{i+1,j}^{\mathcal{A}} \cup (X_4 \setminus R), C_{i+1,j+1,1}^{\mathcal{A}} \setminus R) \cap I_p]
\end{aligned}$$

Estos dos conjuntos son la mínima cota superior de dos secuencias creciente definidas como sigue:

$$\begin{aligned}
C_{i+1,j+1,0,0}^{\mathcal{A}} &:= \emptyset \\
C_{i+1,j+1,0,l+1}^{\mathcal{A}} &:= \theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1,j+1,0,l}^{\mathcal{A}} \setminus R), C_{i+1,j+1,0}^{\mathcal{A}} \setminus R)
\end{aligned}$$

y

$$\begin{aligned}
C_{i+1,j+1,1,0}^{\mathcal{A}} &:= \emptyset \\
C_{i+1,j+1,1,l+1}^{\mathcal{A}} &:= \theta^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C_{i+1}^{\mathcal{A}} \cap R_p) \cup C_{i+1,j}^{\mathcal{A}} \cup (C_{i+1,j+1,1,l}^{\mathcal{A}} \setminus R), C_{i+1,j+1,1}^{\mathcal{A}} \setminus R) \cap I_p
\end{aligned}$$

Por lo tanto,

$C^{\mathcal{A}} = \bigcup_{i,j,l=1}^{|x|} \bigcup_{k=0,1} C_{i,j,k,l}^{\mathcal{A}}$.

Definimos una función total $\Phi_{i,j,k,l}^{\mathcal{A}} : C_{i,j,k,l}^{\mathcal{A}} \to \mathcal{C}$, para todo componente en $C^{\mathcal{A}}$, si $k = 0$, entonces definimos $\Phi^{\mathcal{A}}_{i+1,j+1,0,l+1} : C^{\mathcal{A}}_{i+1,j+1,0,l+1} \to \mathcal{C}$ tal que, para todo $x \in C^{\mathcal{A}}_{i+1,j+1,0,l+1}$ se cumple que,

para todo $\sigma \in \Phi^{\mathcal{A}}_{i+1,j+1,0,l+1}(x) : \delta(\sigma, x) \in C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,j+1,0,l} \setminus R) \cup (C^{\mathcal{A}}_{i+1,j+1,0} \setminus R)$

y existe $\sigma \in \Phi^{\mathcal{A}}_{i+1,j+1,0,l+1}(x) : \delta(\sigma, x) \in C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1,j+1,0,l} \setminus R)$;

si $k = 1$, entonces definimos $\Phi^{\mathcal{A}}_{i+1,j+1,1,l+1} : C^{\mathcal{A}}_{i+1,j+1,1,l+1} \to \mathcal{C}$ tal que, para todo $x \in C^{\mathcal{A}}_{i+1,j+1,1,l+1} \subseteq I_p$ se cumple que,

para todo $\sigma \in \Phi^{\mathcal{A}}_{i+1,j+1,1,l+1}(x) : \delta(\sigma, x) \in C^{\mathcal{A}}_i \cup (C_{i+1} \cap R_p) \cup C^{\mathcal{A}}_{i+1,j} \cup (C^{\mathcal{A}}_{i+1,j+1,1,l} \setminus R) \cup (C^{\mathcal{A}}_{i+1,j+1,1} \setminus R)$

y existe $\sigma \in \Phi^{\mathcal{A}}_{i+1,j+1,1,l+1}(x) : \delta(\sigma, x) \in C^{\mathcal{A}}_i \cup (C_{i+1} \cap R_p) \cup C^{\mathcal{A}}_{i+1,j} \cup (C^{\mathcal{A}}_{i+1,j+1,1,l} \setminus R)$.

Definamos una función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \to \mathcal{C}$ por la regla $\Phi^{\mathcal{A}}(x) := \Phi^{\mathcal{A}}_{i+1,j+1,k.l}(x)$, donde (i, j, k, l) es la cuadrupla mínima ordenada de $N \times N \times \{0, 1\} \times N$ tal que $x \in C^{\mathcal{A}}_{i+1,j+1,k,l}$.

Mostraremos ahora que el supervisor $f_x : \Sigma^* \to \mathcal{C}$ es completo y no bloqueante para (L_x, S_x) tal que $S_x^{f_x} \subseteq E_x$.

Obsérvese que f_x es un supervisor completo ya que para todo $x \in C^{\mathcal{A}}$, se cumple que para todo $\sigma \in \Phi^{\mathcal{A}}(x)$, $\delta(\sigma, x) \in C^{\mathcal{A}}$.

Para probar que $S_x^{f_x} \subseteq E_x$, supongamos que $s \in S_x^{f_x}$. Entonces, existe un camino $\pi : pre(s) \to X$ de s sobre $\mathcal{A}$ tal que $\pi(1) \in C^{\mathcal{A}}$ y para toda $k\sigma \in pre(s)$, $\sigma \in \Phi^{\mathcal{A}}(\pi(k))$.

Mostraremos que $s \in E_x$.

Definimos la función $I_p - rango : C^{\mathcal{A}} \to N^2$ por la regla $I_p - rango(x) := (i, k)$, donde (i, k) es el menor par en el orden de $N \times \{0, 1\}$ tal que $x \in C^{\mathcal{A}}_{i,j,k,l}$.

Obsérvese que para todo $i \in N$,

(4.2.1) $$\text{Si } (I_p - rango(x)) = (i, 1), \text{ entonces } x \in I_p$$

Por la definición de $\Phi^{\mathcal{A}}$,

(4.2.2) $$(I_p - rango(\delta(\sigma, x))) \leq (I_p - rango(x)), \text{ para todo } x \in C^{\mathcal{A}} \text{ y } \sigma \in \Phi^{\mathcal{A}}(x)$$

Además, si $x \in C^{\mathcal{A}}$, $\sigma \in \Phi^{\mathcal{A}}(x)$,$I_p - rango(x) = (i, 0)$ y $\delta(\sigma, x) \in R_p$, entonces

$$(I_p - rango(\delta(\sigma, x))) < (I_p - rango(x)).$$

Debido a que $\Omega_\pi \cap R \neq \emptyset$ y 4.2.1, tenemos que $\Omega_\pi \subseteq I_p$.

Ahora definimos la función $R_p - rango : C^{\mathcal{A}} \rightarrow N^2$ por la regla $R_p - rango(x) := (i, j)$, donde (i, j) es el menor par de números naturales tal que $x \in C^{\mathcal{A}}_{i,j}$.

Por la definición de $\Phi^{\mathcal{A}}$, si $x \in C^{\mathcal{A}}$, $\sigma \in \Phi^{\mathcal{A}}(x)$ y $\delta(\sigma, x) \notin R_p$, entonces

$$(R_p - rango(\delta(\sigma, x))) \leq (R_p - rango(x)).$$

En particular, si $x \in C^{\mathcal{A}}$, $\sigma \in \Phi^{\mathcal{A}}(x)$ y $\delta(\sigma, x) \in R \setminus R_p$, entonces

$$(R_p - rango(\delta(\sigma, x))) < (R_p - rango(x)).$$

Por lo tanto, $\Omega_\pi \cap R_p \neq \emptyset$, es decir, $s \in E_x$.

Probaremos ahora que f_x es no bloqueante.

Definimos la función $rango : C^{\mathcal{A}} \rightarrow N \times N \times \{0,1\} \times N$ por la regla $rango(x) := (i, j, k, l)$, donde (i, j, k, l) es la menor cuadrupla de números naturales tal que $x \in C^{\mathcal{A}}_{i,j,k,l}$.

Por la definición de $\Phi^{\mathcal{A}}$, toda $k \in L^{f_x}_x$ puede ser extendida a $s \in lim(L^{f_x}_x)$ tal que para todo $l\sigma \in pre(s)$,

(4.2.3) Si $\delta(kl\sigma, x) \notin R_p$, entonces $rango(\delta(kl\sigma, x)) < rango(\delta(kl, x))$

y

(4.2.4) Si $I_p - rango(\delta(kl, x)) = (i, 0)$, entonces $rango(\delta(kl\sigma, x)) < rango(\delta(kl, x))$

Es suficiente con probar que $s \in S_x$. Sea $\pi : pre(s) \rightarrow X$ un camino de s sobre $\mathcal{A}$ tal que $\pi(1) \in C^{\mathcal{A}}$ y para toda $k\sigma \in pre(s)$, $\sigma \in \Phi^{\mathcal{A}}(\pi(k))$. Por la proposición 4.2.3, tenemos que $\Omega_\pi \cap R_p \neq \emptyset$. Además por 4.2.1, 4.2.2 y 4.2.4 tenemos que $\Omega_\pi \subseteq I_p$. Por lo tanto, $s \in E_x \subseteq S_x$.

Esto completa la demostración.

Supongamos que $| P | > 1$ y que el resultado se cumple para todo autómata con menor número de pares. Entonces, para todo $x \in C^{\mathcal{A}}$, definimos la función total $\Phi^{\mathcal{A}} : C^{\mathcal{A}} \rightarrow \mathcal{C}$ por la siguiente regla:

$$\Phi^{\mathcal{A}}(x) := \bigcup_{p \in P} \Phi^{\mathcal{A}}_p(x)$$

donde $\Phi_p^{\mathcal{A}}(x)$ es un controlador supervisor completo y no bloqueante para el par (R_p, I_p) y está definido como en el caso $\mid P \mid = 1$.

Mostraremos que el correspondiente supervisor $f_x : \Sigma^* \to \mathcal{C}$ es completo y no bloqueante para (L_x, S_x) tal que $S_x^{f_x} \subseteq E_x$.

Observemos que el correspondiente supervisor es completo y no bloqueante. Esto se debe a que, por la proposición 3.5.9, $\Phi_p^{\mathcal{A}}(x)$ es un supervisor completo y no bloqueante, para todo $p \in P$

Ahora probaremos que $S_x^{f_x} \subseteq E_x$.

Sea $s \in S_x^{f_x}$. Entonces, existe un camino $\pi : pre(s) \to X$ de s sobre $\mathcal{A}$ tal que $\pi(1) \in C^{\mathcal{A}}$ y para toda $k\sigma \in pre(s)$, $\sigma \in \Phi^{\mathcal{A}}(\pi(k))$.

Como $x \in C^{\mathcal{A}}$, entonces $x \in \bigcup_{p \in P} C_p^{\mathcal{A}}$. Esto significa que $x \in C_p^{\mathcal{A}}$ para algún $p \in P$. Sea $E_x^p \subseteq \Sigma^\omega$ el ω-lenguaje aceptado por el autómata de Rabin $(\Sigma, X, \delta, x, \{(R_p, I_p)\})$ que consiste del único par $p \in P$. Luego, $s \in E_x^p$. Como el par $p \in P$ es un par de aceptación del autómata de Rabin completo $(\Sigma, X, \delta, x, \{(R_p, I_p) : p \in P\})$, entonces $s \in E_x$. □

CAPÍTULO 5

Ejemplos

En este capítulo mostraremos el método de síntesis descrito en el capítulo 3 aplicado a dos ejemplos. Mostraremos como se modelan el sistema y la especificación, así como todos los cálculos que describimos en el método de síntesis hasta la obtención del supervisor.

El primer ejemplo es un sistema pequeño de una máquina de refrescos, que utilizaremos principalmente para mostrar la forma de modelar especificaciones de vivacidad. En este ejemplo primeramente tomaremos como especificación secundaria al lenguaje vacío para obtener el supervisor. Posteriormente incrementaremos este lenguaje y aplicaremos de nuevo el método de síntesis. Esto es con el fin de mostrar como obtener un supervisor óptimo aproximando el lenguaje controlado resultante al lenguaje ω-controlable supremo. En otras palabras, mostraremos que la especificación secundaria junto con el lenguaje ω-controlable supremo, determinan el comportamiento generado bajo la supervisión. Finalmente, aplicaremos la teoría de Ramadge y Wonham de lenguajes finitos a este mismo ejemplo para obtener un supervisor. Luego, compararemos ambos supervisores y los comportamientos que generan bajo la supervisión.

El segundo ejemplo es un sistema de manufactura propuesto en **[11]**, que consiste de dos máquinas y dos almacenes. En este ejemplo, mostraremos la forma de modelar el sistema a partir de componentes básicos y la manera de especificar propiedades de seguridad utilizando la misma estructura del autómata que representa el comportamiento del sistema.

5.1. Máquina de Refrescos

Considere el siguiente ejemplo que es una modificación al ejemplo propuesto por **[17]**. El sistema consiste de una máquina de refrescos. Cuando la máquina falla, entonces se puede reemplazar con otra o se puede mandar a reparar. Si se manda a reparar, una

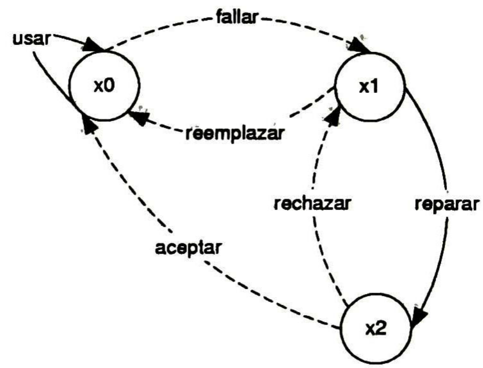


FIGURA 5.1.1. Modelo del sistema de la máquina de refrescos $MR = (\Sigma, X, \delta, x_0, R)$, con $L = pre(S)$, estado inicial x_0, $\Sigma_c = \{usar, reparar\}$ y conjunto de aceptación $R = \{x_0, x_1, x_2\}$.

vez reparada se envía a inspección. Si la máquina pasa la inspección, se pone de nuevo en uso. En caso contrario, se envía de nuevo a reparación o se reemplaza con otra.

De acuerdo a la descripción anterior, el sistema puede encontrarse en tres estados:

- OPERANDO: la máquina está funcionando
- DESCOMPUESTA: la máquina está descompuesta
- EN ESPERA: la máquina está en espera para ser inspeccionada

El sistema controlado debe satisfacer lo siguiente : siempre que la máquina se descomponga, eventualmente se pondrá en uso.

5.1.1. Paso 1: Modelado del sistema. Modelamos el sistema como un SED $G = (L, S)$, donde el ω-comportamiento S, es el lenguaje reconocido por el autómata de Büchi $MR = (\Sigma, X, \delta, x_0, R)$, de la figura 5.1.1, donde $X = \{x_0, x_1, x_2\}$ tal que x_0 = OPERANDO, x_1= DESCOMPUESTA y x_2 = EN ESPERA, Σ_c ={usar, reparar}, el estado inicial es x_0 y el conjunto de aceptación es $R = \{x_0, x_1, x_2\}$. El ω-comportamiento S es ω-cerrado[1] y el $_*$-lenguaje $pre(S)$ define el $_*$-comportamiento L, es decir, $L = pre(S)$.

5.1.2. Paso 2: Modelado de las especificaciones. Siguiendo el método de síntesis dividimos a la especificación en dos.

[1]Esto se debe a que en la descripción del sistema no se asume ninguna restricción.

Especificación primaria:

De acuerdo a la especificación se requiere que la máquina no se quede en el ciclo infinito reparar - rechazar. En consecuencia, al autómata de Büchi $MR = (\Sigma, X, \delta, x_0, R)$ le agregamos la condición de aceptación del autómata de Rabin que consiste del par: $(R_p, I_p) = (\{x_0\}, \{x_0, x_1, x_2\})$ para formar el autómata de Rabin-Büchi $MR = (\Sigma, X, \delta, x_0, (R_p, I_p), R)$. Así, el ω-lenguaje $E \subseteq S$ reconocido por el autómata de Rabin, es el conjunto de todas las palabras infinitas sobre Σ cuyos caminos contienen al estado x_0 un número infinito de veces. Por lo tanto, $E \subseteq S$ captura la especificación. De acuerdo con lo visto en el capítulo 1, esta especificación se clasifica como de vivacidad porque hace referencia a un futuro no especifico, ya que declara que la máquina se pondrá de nuevo en uso sin importar en que instante sucederá.

Especificación secundaria:

Debido a que el lenguaje de especificación $E \subseteq S$ no es ω-cerrado con respecto a S, para obtener una solución óptima tendremos que definir una especificación secundaria. Para este caso, primero tomaremos como especificación secundaria al lenguaje $A := \emptyset$ y luego lo incrementaremos.

5.1.3. Paso 3: Cálculo del lenguaje ω-controlable supremo. Para obtener el lenguaje ω-controlable supremo, primero debemos calcular los prefijos controlables del lenguaje E, que denotamos C^{MR}. Como $\mid P \mid = 1$, entonces $C^{MR} := \mu X_1.f_1(X_1)$, donde f_1 está definido por

$$f_1(X_1) := \nu X_2.\mu X_3.[\theta^{MR}(X_1) \cup [\theta^{MR}(X_1 \cup (X_2 \cap R_p) \cup X_3) \cap I_p]].$$

Sean $f_2(X_2) := \mu X_3.[\theta^{MR}(X_1) \cup [\theta^{MR}(X_1 \cup (X_2 \cap R_p) \cup X_3) \cap I_p]]$ y $f_3(X_3) := [\theta^{MR}(X_1) \cup [\theta^{MR}(X_1 \cup (X_2 \cap R_p) \cup X_3) \cap I_p]]$. Entonces por el teorema 2.3.5, debemos calcular este punto fijo mínimo como sigue:

$$
\begin{aligned}
C_0^{MR} &:= \emptyset \\
C_1^{MR} &:= \nu X_2.\mu X_3.[\theta^{MR}(\emptyset) \cup [\theta^{MR}(\emptyset \cup (X_2 \cap R_p) \cup X_3) \cap I_p]] \\
&\qquad f_2^0(X_2) := X \\
&\qquad f_2^1(X_2) := \mu X_3.[\theta^{MR}(\emptyset) \cup [\theta^{MR}(\emptyset \cup (X \cap R_p) \cup X_3) \cap I_p]] \\
&\qquad\qquad f_3^0(X_3) := \emptyset \\
&\qquad\qquad f_3^1(X_3) := \theta^{MR}(\emptyset) \cup [\theta^{MR}(\emptyset \cup (X \cap R_p) \cup \{q_1\}) \cap I_p] \\
&\qquad\qquad\qquad = \theta^{MR}(R_p) \cap I_p \\
&\qquad\qquad\qquad = \{x_1\} \cap \{x_0, x_1, x_2\} = \{x_1\} \\
&\qquad\qquad f_3^2(X_3) := \theta^{MR}(\emptyset) \cup [\theta^{MR}(\emptyset \cup (X \cap R_p) \cup \emptyset) \cap I_p] \\
&\qquad\qquad\qquad = \theta^{MR}(\{x_0, x_1\}) \cap I_p \\
&\qquad\qquad\qquad = \{x_0, x_1, x_2\} \cap \{x_0, x_1, x_2\} = \{x_0, x_1, x_2\} \\
&\qquad\qquad := \{x_0, x_1, x_2\} \\
&:= \{x_0, x_1, x_2\}
\end{aligned}
$$

Como $C^{MR} = X$, entonces definimos a la función de transición δ^{MR} exactamente igual que δ. Ahora construimos el autómata de la figura 5.1.2. Este autómata reconoce el lenguaje ω-controlable supremo. Nótese que el lenguaje ω-controlable supremo es exactamente igual al lenguaje E.

5.1.4. Paso 4: Verificación de la existencia de la solución. El lenguaje ω-controlable supremo no es vacío ya que $x_0 \in C^{MR}$. Como $A = \emptyset$, pasamos directamente a la síntesis del supervisor.

5.1.5. Paso 5: Síntesis del supervisor. Siguiendo la demostración del teorema 4.1.12, obtenemos la siguiente función

$$
\Phi^{MR}(x) = \begin{cases} \Sigma_u \cup \{usar\} & \text{si } x = x_0 \\ \Sigma_u & \text{si } x = x_1 \\ \Sigma_u & \text{si } x = x_2 \end{cases}
$$

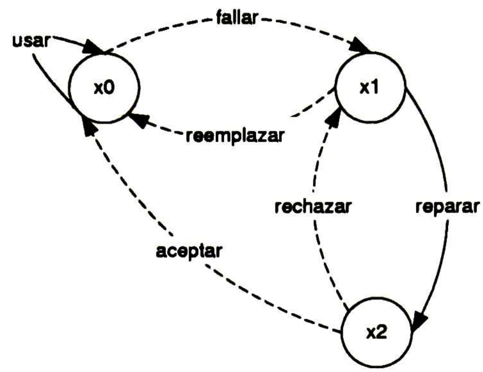


FIGURA 5.1.2. Autómata de Rabin $(\Sigma, X, \delta^{MR}, x_0, \{(R_p, I_p)\})$ que reconoce el lenguaje ω-controlable supremo del ejemplo máquina de refrescos.

para formar el controlador supervisor completo y no bloqueante

$$f(l) := \begin{cases} \Phi^{\mathcal{A}}(\delta(l, x_0)) & \text{si } \delta(l, x_0) \in C^{\mathcal{A}} \\ indefinida & \text{en otro caso} \end{cases}$$

La figura 5.1.3 muestra el comportamiento generado bajo la supervisión de f. Como puede observarse, f siempre inhabilita el evento reparar en el estado x_1 para evitar que el sistema se quede en el ciclo infinito reparar - rechazar. Esto ocasiona que siempre que la máquina se descomponga se reemplace por otra y nunca se envíe a reparación. Debemos observar que dentro del lenguaje ω-controlable supremo existen secuencias que contienen, un número finito de veces, a los eventos reparar rechazar. Como por ejemplo, $usar^* fallarreparar(rechazarreparar)^* aceptarusar^\omega$ Para incluir estas secuencias dentro del comportamiento generado bajo la supervisión, debemos regresar al paso 2 del método de síntesis para modificar la especificación secundaria.

5.1.6. Paso 2: Modelado de las especificaciones. Dejamos igual al lenguaje de especificación E y ahora tomamos como especificación secundaria, que denotamos A, al lenguaje reconocido por el autómata de Büchi $MRa = (\Sigma, X_a, \delta_a, a_0, R_a)$ que se encuentra en la figura 5.1.4. Nótese que este ω-lenguaje consiste de todas las secuencias infinitas de eventos que contienen al evento fallar seguidas por los eventos reemplazar

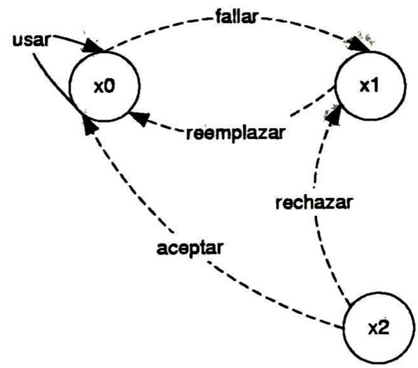


FIGURA 5.1.3. Comportamiento del sistema máquina de refrescos generado bajo la supervisión de f, tomando $A = \emptyset$.

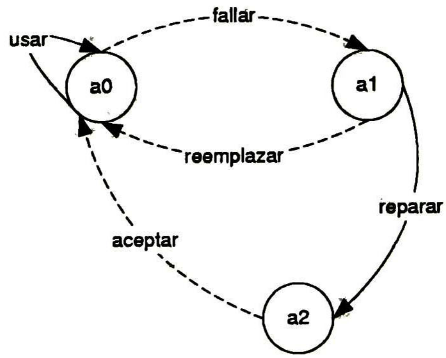


FIGURA 5.1.4. Autómata de Büchi $MRa = (\Sigma, X_a, \delta_a, a_0, R_a)$, con $X_a = \{a_0, a_1, a_2\}$, $\Sigma_c = \{usar, reparar\}$ y conjunto de aceptación $R_a = \{a_0\}$.

y reparar. Por lo tanto, esta especificación se encargará de que la máquina se envíe a reparar por lo menos una vez.

5.1.7. Paso 3: Cálculo del lenguaje ω-controlable supremo. Debido a que el lenguaje de especificación E es el mismo que en el caso anterior, entonces el lenguaje $supC^{\omega}(E)$ también es igual.

5.1.8. Paso 4: Verificación de la existencia de la solución. Para construir el $_*$-autómata que reconoce el $_*$-lenguaje $pre(A)$, tomamos la estructura del autómata de la figura 5.1.4 para construir el $_*$-autómata $MRpre(a) = (\Sigma, X_a, \delta_a, a_0, X_a)$[2].

[2]Nótese que ambos autómatas tienen la misma estructura, pero la condición de aceptación del $_*$-autómata es igual a todo el conjunto de estados. Esto se debe a que este automata es $_*$-cerrado.

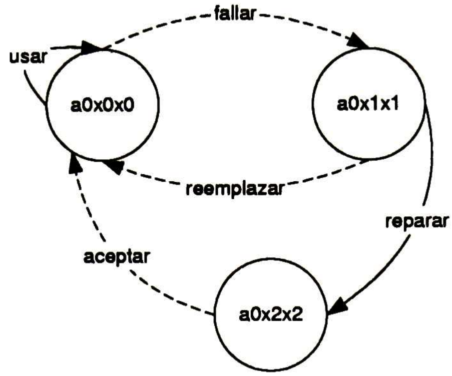


FIGURA 5.1.5. Autómata de Streett $MR_{diff} = (\Sigma, X''', \delta''', \{(R_p, I_p) : p \in P\})$, con pares de aceptación $(\{a_0x_0x_0,\ a_1x_1x_1,\ a_2x_2x_2\}, \emptyset)$ y $(\emptyset, \{a_1x_1x_1,\ a_2x_2x_2\})$.

Luego, construimos el autómata de Streett MR_{diff} de la figura 5.1.5. El estado inicial es $a_0x_0x_0$ y los pares de aceptación son $(\{a_0x_0x_0,\ a_1x_1x_1,\ a_2x_2x_2\}, \emptyset)$ y $(\emptyset, \{a_1x_1x_1,\ a_2x_2x_2\})$. El autómata MR_{diff} es vacío ya que no existe ninguna palabra generada por el autómata que sea aceptada.

5.1.9. Paso 5: Síntesis del supervisor. Construimos el $_*$-autómata $(\Sigma, X, \delta, x_0, X)$, de la figura 5.1.6, que reconoce el lenguaje $pre(supC^{\omega}(E))$ y lo utilizamos para definir la función:

$$\Phi_0(x) := \begin{cases} \Sigma_u \cup \{usar\} & \text{si } x = x_0 \\ \Sigma_u \cup \{reparar\} & \text{si } x = x_1 \\ \Sigma_u & \text{si } x = x_2 \end{cases}$$

Nótese que la función Φ_0 no inhabilita ningún evento, esto se debe a que $supC^{\omega}(E) = E$.

Siguiendo la demostración del teorema 4.1.12, obtenemos la siguiente función:

$$\Phi^{MR}(x) = \begin{cases} \Sigma_u \cup \{usar\} & \text{si } x = x_0 \\ \Sigma_u & \text{si } x = x_1 \\ \Sigma_u & \text{si } x = x_2 \end{cases}$$

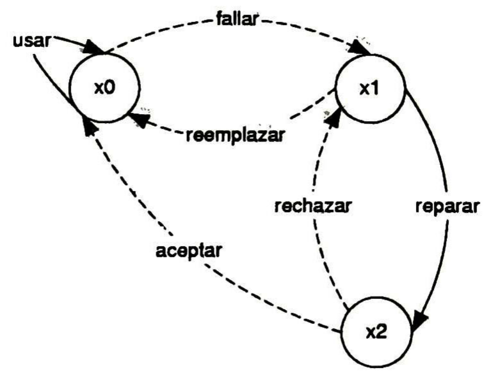


FIGURA 5.1.6. $_*$-autómata $(\Sigma, X, \delta^{MR}, x_0, X)$ que reconoce el lenguaje $pre(supC^{\omega}(E))$ del ejemplo de la máquina de refrescos.

Luego, definimos los supervisores no bloqueantes y completos $f_1(l) := \Phi^{MR}(\delta(l, x_0))$ y $f_0(l) := \Phi_0(\delta, (l, x_0))$.

Finalmente, construimos el controlador supervisor completo f compuesto por f_0 y f_1:

$$f(l) = \begin{cases} f_0(l) & \text{si } l \in pre(A) \\ f_1(l) & \text{si } l \notin pre(A) \text{ y } \delta(l, x_0) \in C^{MR} \\ \text{indefinida} & \text{en otro caso} \end{cases}$$

La figura 5.1.7 muestra el comportamiento del sistema bajo la supervisión de f. Cuando la máquina de refrescos falla, entonces el supervisor f no inhabilita los eventos reemplazar y reparar, de esta manera f permite que por lo menos una sola vez se mande a reparar la máquina satisfaciendo así las especificaciones primaria y secundaria.

Aunque el comportamiento generado bajo la supervisión es más grande que en el caso anterior, podemos aproximar aún más este comportamiento al lenguaje ω-controlable supremo, para el caso que queramos que la máquina se mande a reparar n veces.

Nótese que si la máquina se repara y se rechaza, entonces la máquina ya no se enviará nunca más a reparación. Esto se debe a que Φ^{MR} inhabilita eventualmente el evento *reparar* en el estado x_1 a través de la generación de una palabra que no es un elemento del conjunto $pre(A)$. Una vez generada dicha palabra, siempre todas las palabras que generemos a partir de entonces, tendrán como prefijo a esta palabra y por

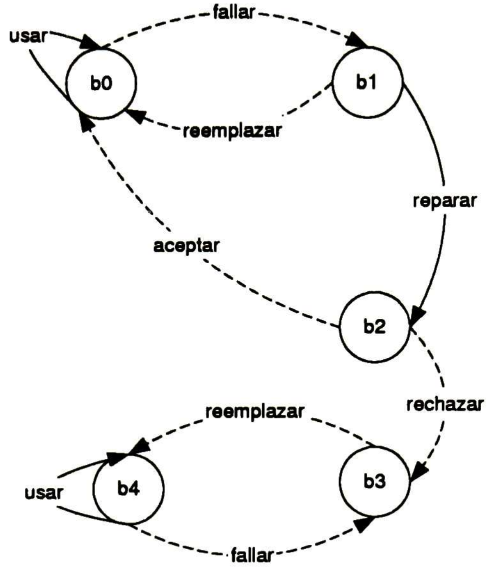


FIGURA 5.1.7. Comportamiento del sistema máquina de refrescos generado bajo la supervisión de f, tomando como A el ω-lenguaje del autómata de la figura 5.1.4.

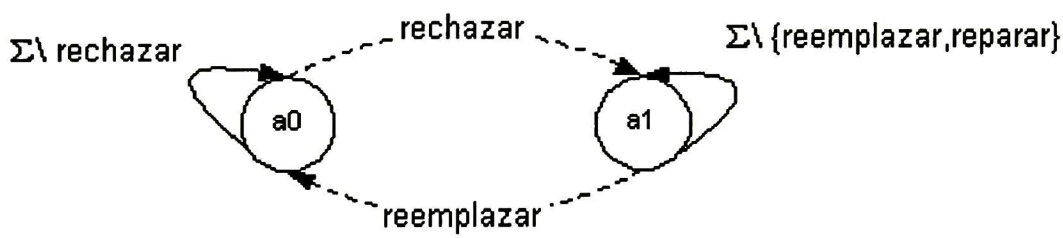


FIGURA 5.1.8. $_*$-autómata $(\Sigma, X_a, \delta_a, a_0, X_a)$ de la especificación de seguridad.

lo tanto tampoco pertenecerán al conjunto $pre(A)$. Esto hace que de ahí en adelante Φ^{MR} inhabilite siempre el evento *reparar*. Esto nos lleva a la conclusión de que por más grande que sea la especificación secundaria no podremos construir un supervisor f de manera tal que haga que la máquina se mande a reparar siempre que esta falle aunque este comportamiento sea parte del lenguaje ω-controlable supremo.

5.1.10. Comparación con la teoría de control supervisor de palabras finitas. Debido a que $L = pre(S)$, podemos utilizar la teoría de Radmage y Wonham de

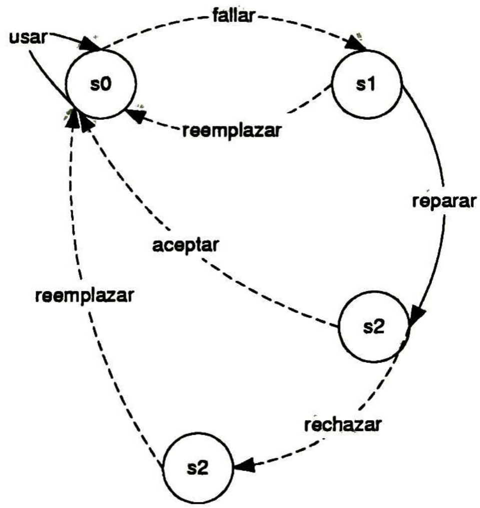


FIGURA 5.1.9. Supervisor $(\Sigma, X_s, \delta_s, s_0, X_s)$ obtenido utilizando la teoría de control de Radmage y Wonham [7].

palabras finitas [7] tomando en cuenta solamente el lenguaje L para tratar de derivar un supervisor que genere el comportamiento especificado.

Como vimos en el capítulo 1, los $_*$-lenguajes no pueden especificar propiedades de vivacidad, pero por la dinámica del sistema, podemos reemplazar esta especificación por una especificación de seguridad (figura 5.1.8) que asegure que siempre que se de el evento rechazar no se de el evento reparar a menos que se de el evento reemplazar.

Una vez aplicada la teoría propuesta en [7] obtenemos el supervisor que se encuentra en la figura 5.1.9.

Ahora la idea es aplicar este controlador supervisor al sistema completo $G = (L, S)$. Obsérvese que el ω-comportamiento generado bajo la supervisión no contiene el ciclo infinito reparar-rechazar y por lo tanto satisface la especificación de vivacidad.

Es importante hacer notar que el comportamiento generado por el sistema $G = (L, S)$ bajo la acción del supervisor de la figura 5.1.9 es más grande que el comportamiento generado por el sistema $G = (L, S)$ bajo la acción del supervisor definido en el paso 5 del método de síntesis.

También cabe aclarar que en sistemas más grandes y complejos no es tan evidente detectar ciclos que no satisfacerán la especificación y no siempre es tan simple tratar de evitarlos por medio de una especificación de seguridad como en el caso de este ejemplo.

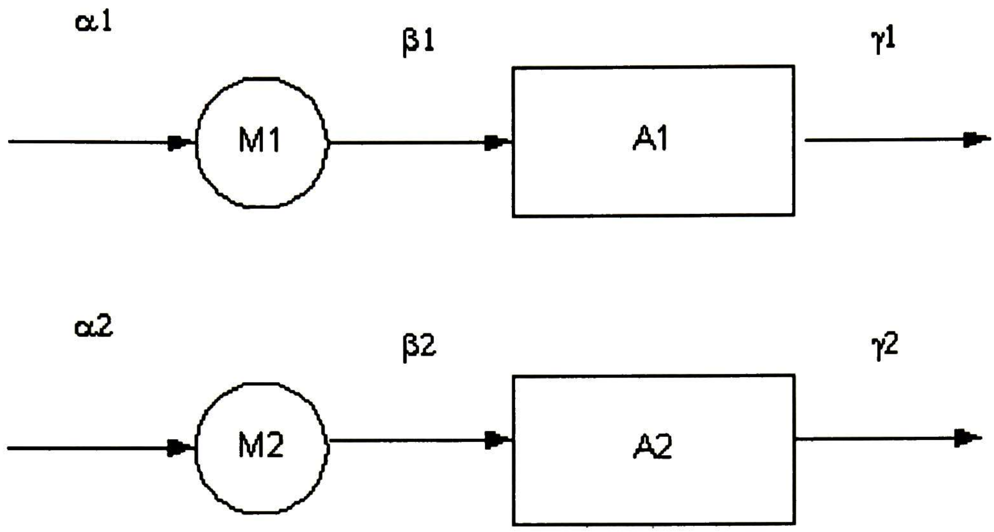


FIGURA 5.2.1. Operación del sistema máquinas y almacenes.

En dichos casos es mejor utilizar la teoría de control que hemos estado viendo a lo largo de la tesis.

5.2. Máquinas y Almacenes

Considere ahora el sistema compuesto por dos máquinas M_1 y M_2 conectadas a dos almacenes A_1 y A_2, como se muestra en la figura 5.2.1. Cada máquina M_i sólo puede procesar una pieza a la vez y cada almacén A_i tiene capacidad para una sola pieza.

Al inicio de la operación los almacenes A_1 y A_2 están vacíos. La máquina M_i adquiere una pieza desde algún lugar de la fabrica mediante el evento α_i. Después de completar su trabajo, deposita la pieza dentro del almacén A_i mediante el evento β_i. Posteriormente la pieza es removida del almacén A_i mediante el evento γ_i. Los eventos α_1 y α_2 son los únicos eventos controlables.

El sistema controlado debe satisfacer las siguientes especificaciones:

1. Ya que las máquinas M_1 y M_2 utilizan la misma fuente de recursos, no se debe permitir que operen simultáneamente (exclusión mutua).
2. No se debe rebasar la capacidad de los almacenes (prevenir desbordamiento).

y el almacén A_2 y formamos el conjunto $D_2 = \{DOVL,\ OOVL,\ DOLL,\ OOLL\}$. Luego, unimos los conjuntos D_1 y D_2 para obtener el conjunto de estados prohibidos $D = D_1 \cup D_2$ de la especificación 2.

Ahora eliminamos del conjunto de estados del autómata los estados que pertenecen al conjunto $EM \cup D$ y formamos el par $(R_p, I_p) = (X \setminus (EM \cup D), X \setminus (EM \cup D))$. Al autómata de Büchi $MA = (\Sigma, X, \delta, x_0, R)$ de la figura 5.2.3 le agregamos el par (R_p, I_p), para formar el autómata de Rabin-Büchi $MA = (\Sigma, X, \delta, x_0, \{(R_p, I_p)\}, R)$. De esta manera, el ω-lenguaje $E \subseteq S$ reconocido por el autómata de Rabin, es el conjunto de todas las palabras infinitas cuyos caminos contienen al conjunto $X \setminus (EM \cup D)$ un número infinito de veces y además $X \setminus (EM \cup D)$ es eventualmente invariante. Esto significa que el ω-lenguaje E contiene algunas palabras cuyos caminos tienen un número finito de veces a estados que pertenecen al conjunto $EM \cup D$. Aunque estos estados no satisfacen la especificación, por la definición 4.1.10, si alguno o algunos de estos estados pertenece al conjunto controlable C^{MA}, entonces Φ^{MA} siempre inhabilitará los eventos que conduzcan a estos estados.

Especificación secundaria:

Como las especificaciones 1 y 2 son de seguridad, tomaremos como especificación secundaria $A := \emptyset$.

5.2.3. Paso 3: cálculo del sublenguaje ω-controlable supremo.

Igual que en el caso anterior, como $\mid P \mid = 1$, entonces $C^{MA} := \mu X_1.f_1(X_1)$, donde f_1 está definido por:

$$f_1(X_1) := \nu X_2.\mu X_3.[\theta^{MA}(X_1) \cup [\theta^{MA}(X_1 \cup (X_2 \cap R_p) \cup X_3) \cap I_p]]$$

El resultado de este punto fijo mínimo se encuentra en la tabla 1. Las columnas 2, 3, 4, 5 y 6 corresponden a las cinco iteraciones necesarias para calcular el punto fijo mínimo $\mu X_1.f_1(X_1)$. En cada columna, los elementos que pertenecen al conjunto $\nu X_2.\mu X_3.[\theta^{MA}(X_1) \cup [\theta^{MA}(X_1 \cup (X_2 \cap R_p) \cup X_3) \cap I_p]]$ están marcados con $\surd$.

Debido a que $C^{MA} = X$, definimos la función de transición δ^{MR} igual que δ. En consecuencia, el autómata que reconoce el sublenguaje ω-controlable supremo es exactamente igual al autómata de Rabin que se encuentra en la figura 5.2.3.

	C_1^{MA}	C_2^{MA}	C_3^{MA}	C_4^{MA}	C_5^{MA}	$\Phi^{MA}(x)$
DDVV	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\Sigma_u \cup \{\alpha_1,\alpha_2\}$
ODVV	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	Σ_u
DOVV	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	Σ_u
DDLV	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\Sigma_u \cup \{\alpha_2\}$
OOVV		$\surd$	$\surd$	$\surd$	$\surd$	Σ_u
DDVL	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\Sigma_u \cup \{\alpha_1\}$
ODLV		$\surd$	$\surd$	$\surd$	$\surd$	Σ_u
DOLV	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	Σ_u
ODVL	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	Σ_u
DOVL		$\surd$	$\surd$	$\surd$	$\surd$	Σ_u
OOLV				$\surd$	$\surd$	Σ_u
DDLL	$\surd$		$\surd$	$\surd$	$\surd$	Σ_u
OOVL				$\surd$	$\surd$	Σ_u
ODLL			$\surd$	$\surd$	$\surd$	Σ_u
DOLL			$\surd$	$\surd$	$\surd$	Σ_u
OOLL					$\surd$	Σ_u

TABLA 1. Cálculo del conjunto controlable C^{MA} y la función Φ^{MA} del ejemplo máquinas y almacenes.

5.2.4. Paso 4: Verificación de existencia de la solución. El sublenguaje ω-controlable supremo no es vacío ya que $x_0 \in C^{MR}$ y como $A = \emptyset$, pasamos directamente a la síntesis del supervisor.

5.2.5. Paso 5: Síntesis del supervisor. Para todo $x \in C^{MR}$, calculamos la función Φ^{MA} que se encuentra en la tabla 1, para luego definir el controlador supervisor f como sigue.

$$f(l) = \begin{cases} \Phi^{MA}(\delta(l, DDVV)) & \text{si } \delta(l, DDVV) \in C^{MA} \\ \text{indefinida} & \text{en otro caso} \end{cases}$$

La figura 5.2.4 muestra el comportamiento del sistema bajo la supervisión de f. Obsérvese que para evitar que las máquinas M_1 y M_2 operen al mismo tiempo, el supervisor f inhabilita al evento α_1 en los estados $DOVV$ y $DOLV$ que son los estados en los cuales la máquina M_1 está ocupada y la ocurrencia del evento α_1 conduciría a los estados prohibidos $OOVV$ y $OOLV$, respectivamente. Igualmente sucede en los estados $ODVV$ y $ODVL$ que son los estados en los cuales la máquina M_2 está ocupada. En este caso, f inhabilita el evento α_2 que conduce a los estados prohibidos $OOVV$ y $OOVL$, respectivamente.

Para evitar el desbordamiento de los almacenes, el supervisor f inhabilita a el evento α_1 en los estados $DDLV$ y $DDLL$ que son los estados en los cuales el almacén A_1 está lleno y la ocurrencia de este evento conduciría a estados prohibidos. Lo mismo sucede en los estados $DDVL$ y $DDLL$. En estos estados el supervisor f inhabilita a el evento α_2.

Es importante hacer notar que el comportamiento del sistema generado bajo la supervisión no garantiza que ambas máquinas se utilizarán, es decir, se puede dar el caso que siempre se utilice nada más la máquina M_1 y nunca la máquina M_2, o viceversa. Si quisiéramos evitar esto, tendríamos que agregar una tercera especificación en la especificación primaria, como por ejemplo, que una vez que se de el evento α_1 no se de nuevo este evento a menos que se de el evento α_2, y viceversa. Pero una vez agregada esta nueva especificación a nuestro problema de control, este no tendría solución ya que el sublenguaje ω-controlable supremo resultante sería vacío, y por lo tanto no podríamos sintetizar un controlador supervisor.

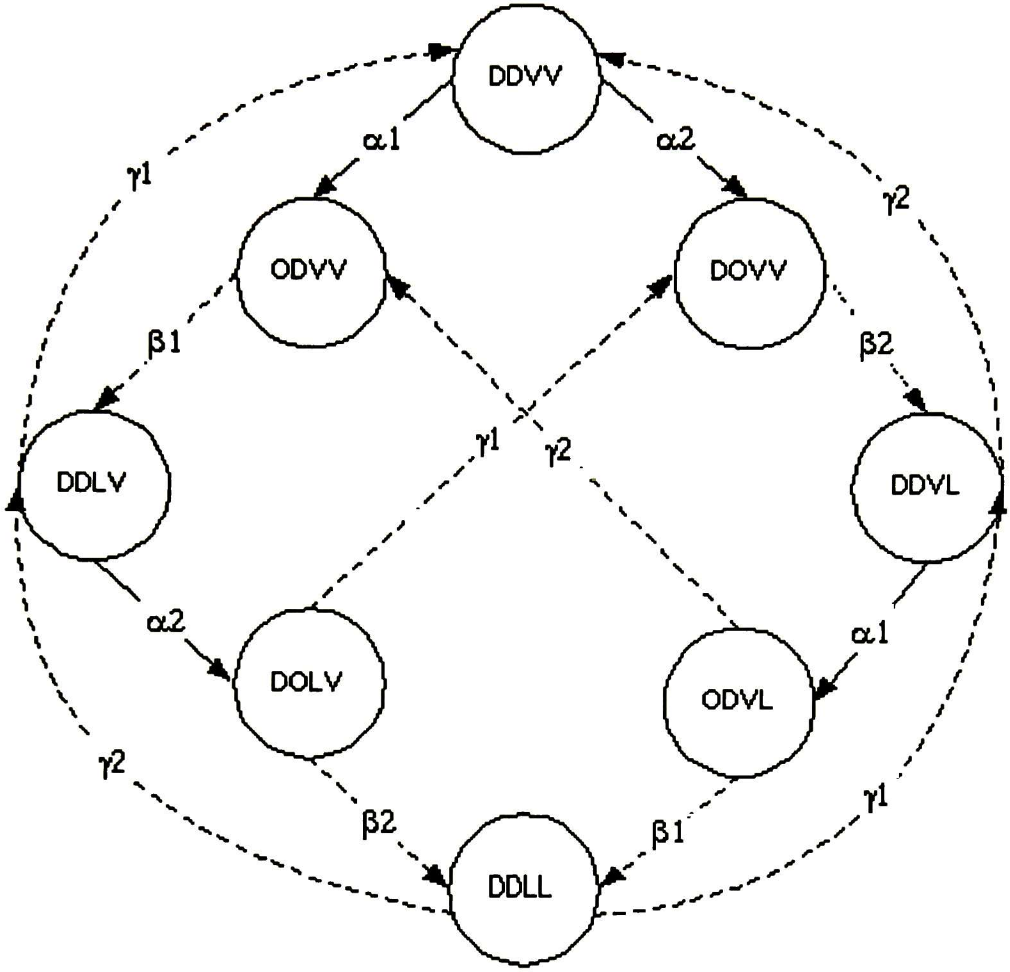


FIGURA 5.2.4. Comportamiento del sistema máquinas y almacenes bajo la supervisión de f.

CAPÍTULO 6

Conclusiones

En esta tesis hemos discutido el modelado y especificación de SED's basados en ω-lenguajes. Estos lenguajes permiten modelar el comportamiento infinito de los sistemas, pero lo más importante es que permiten expresar propiedades de vivacidad.

Como mencionamos anteriormente, las propiedades de vivacidad son exclusivas de los ω-lenguajes, ya que los $_*$-lenguajes no pueden representar dichas propiedades. Para agregar formalmente estas propiedades en los problemas de control es necesaria una teoría de control apropiada que permita representar estas propiedades.

Vimos a grandes rasgos las teorías [**8, 5, 17**]. Revisamos y justificamos a detalle la teoría de control supervisor de Thistle y Wonham [**13**]. Propusimos un método de síntesis de supervisores basado en esta teoría y lo aplicamos a dos ejemplos, mostrando todos los cálculos hasta la obtención del supervisor. Este método de síntesis calcula los prefijos controlables del autómata en tiempo polinomial, a diferencia de [**12, 11**], cuya complejidad es exponencial en el número de pares de la condición de aceptación. También entre las aportaciones de este trabajo se encuentra la implementación en software de los pasos 3, 4 y 5 del método de síntesis.

En las siguientes secciones enumeramos algunas ventajas y desventajas de este método de síntesis, comparado con [**8, 17, 7, 5**].

6.1. Ventajas

- Permite el uso de propiedades de vivacidad y seguridad en la especificación de SEDs a diferencia de [**7, 8**].
- Podemos expresar propiedades de vivacidad en los modelos de sistemas ya, que para todo SED $G = (L, S)$ se asume que $pre(S) \subseteq L$. Es decir, S no tiene que ser un lenguaje ω-cerrado. Las propiedades de vivacidad expresadas en el modelo del sistema son útiles, por ejemplo, para excluir del comportamiento del sistema algunas secuencias de eventos que físicamente no son posibles.

- Todo sublenguaje del lenguaje generado por un SED dado contiene un único lenguaje ω-controlable supremo. Si este lenguaje es ω-cerrado con respecto al comportamiento del sistema, entonces en caso de haber solución, el supervisor resultante es mínimamente restrictivo. Es decir, el comportamiento del sistema bajo la supervisión es exactamente igual al lenguaje ω-controlable supremo. Por el contrario, si el lenguaje ω-controlable supremo no es ω-cerrado con respecto al comportamiento del sistema, en caso de haber solución, es posible derivar un supervisor que genere un sublenguaje aproximado al lenguaje ω-controlable supremo.

6.2. Desventajas

- El comportamiento del sistema y el comportamiento deseado son especificados con diferentes autómatas bajo una misma estructura, lo que dificulta la especificación de algunas propiedades, por ejemplo, precedencia de eventos.
- La teoría no permite especificar restricciones de tiempo real como en el caso de [5].

6.3. Trabajos Futuros

Entre las futuras áreas de trabajo siguiendo esta línea podemos destacar las siguientes:

- Determinar cuándo y bajo qué condiciones se puede reemplazar una propiedades de vivacidad por una de seguridad para resolver problemas de control en el marco de ω-lenguajes utilizando la teoría [7]. Aunque queda claro que la complejidad y dinámica del sistema son determinantes para que se dé esta condición.
- La teoría de control tal como está no permite especificar restricciones de tiempo real, pero como esta teoría está basada en el marco de los ω-lenguajes y estos lenguajes modelan el comportamiento del tiempo, entonces se podría hacer una extensión que permita especificar dichas restricciones.
- Como vimos en el ejemplo de la máquina de refrescos del capítulo 5, no siempre se puede obtener un supervisor que genere el comportamiento que se requiere aunque este sea parte del lenguaje ω-controlable supremo. Esto se debe a que el

supervisor aplica la función f_0 hasta la ocurrencia de una palabra en el conjunto $pre(supC^{\omega}(E)) \setminus pre(A)$ y de ahí en adelante siempre aplica la función f_m. Para solucionar esto, podríamos redefinir el supervisor de manera tal que aplique la función f_0 hasta la ocurrencia de una palabra en el conjunto $pre(supC^{\omega}(E)) \setminus pre(A)$, de ahí que aplique la función f_m y eventualmente que aplique nuevamente la función f_0, y así sucesivamente. De esta manera, el supervisor generaría un comportamiento más grande.

- Mejorar el método de síntesis de controladores expuesto en el capítulo 3, en las siguientes áreas:
  - Computación de los prefijos controlables sin utilizar cálculo de puntos fijos, buscando otras alternativas de representación para tratar de obtener una complejidad menor.
  - Utilización de otros métodos de especificación. En este sentido se puede destacar el uso de la lógica temporal para expresar propiedades de los sistemas.

Bibliografía

[1] B. Alpern and F. B. Schneider. Defining liveness. *Inform. Process. Letter*, pages 181–185, 1985.

[2] E. A. Emerson and C. Lei. Modalities for model checking: Branching time logic strikes back. *journal Sci. Comput. Program*, 8(3):275–306, June 1987.

[3] C. Golaszewski and P. Ramadge. Mutual exclusion problems for discrete event systems with shared events. *Proc. 27th IEEE Conference on Decision and Control*, pages 234–239, December 1988. Austin, TX.

[4] R. P. Malhamé J. G. Thistle. Control of ω-automata under state fairness assumptions. *Systems and control letters*, 33:265–274, 1998.

[5] F. Kabanza M. Barbeau and R. St.-Denis. A method for the synthesis of controllers to handle safety, liveness, and real-time constrains. *IEEE transactions on automatic control*, 43:1543–1559, 1998.

[6] Z. Manna and A. Pnueli. *The Temporal Logic of Reactve and Concurrent Systems: Specification.* Springer-Verlag, 1992.

[7] P. Ramadge and W. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25:206–230, 1987.

[8] P. J. Ramadge. Some tractable supervisory control problems for discrete-event systems modeled by Büchi automata. *IEEE Transactions on Automatic Control*, 34(1):10–19, January 1989.

[9] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proc. IEEE*, 77(1):81–97, January 1989.

[10] A. Tarski. A lattice-theorical fixpoint theorem and its applications. *Pacific Journal of Matematics*, 5:285–309, 1955.

[11] J. G. Thistle. *Control of Infinite Behaviour of Discrete-Event Systems.* Phd. thesis, University of Toronto, 1991.

[12] J. G. Thistle and W. M. Wonham. Control of infinite behaviour of finite automata. *SIAM J.Control and Optimization*, 32(4):1075–1097, July 1994.

[13] J. G. Thistle and W. M. Wonham. Supervision of infinite behaviour of discrete-event systems. *Control and Optimization*, 32(4):1098–1113, July 1994.

[14] W. Thomas. *Automata on Infinite Objects in Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 134–191. J. van Leeuwen, Elsevier, The MIT press, Cambrige, MA edition, 1990.

[15] W. M. Wonham and P. J. Ramadge. Modular supervisory control of discrete event systems. *Mathematics of Control, Signals and Systems*, 1(1):13–30, 1988.

[16] W.M. Wonham. Notes of control of discrete-event systems. Department of Electrical and Computer Engineering, University of Toronto, 1999.

[17] S. Young, D. Spanjol, and V. K. Garg. Control of discrete event systems modeled with deterministic Büchi automata. In *Proc. of 1992 American Control Conference*, pages 2814–2818, Chicago, IL, USA, jun 1992.

Centro de Investigación y de Estudios Avanzados del IPN

Unidad Guadalajara

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis: CONTROL SUPERVISOR PARA SISTEMAS DE EVENTOS DISCRETOS CON ESPECIFICACIONES DE SEGURIDAD Y VIVACIDAD del(a) C. Ana Gabriela SOLORZANO GARCÍA el día 8 de Enero de 2002

Dr. Arturo del Sagrado Corazón Sánchez Carmona
Investigador Cinvestav 3B
CINVESTAV GDL
Guadalajara

Dr. Raul Ernesto González Torres
Investigador Cinvestav 2C
CINVESTAV GDL
Guadalajara

Dr. Félix Francisco Ramos Corchado
Investigador Cinvestav 2A
CINVESTAV GDL
Guadalajara

Dr. José de Jesús Alvarez Ramírez
Profesor Titular C
Universidad Autonoma Metropolitana de Iztapalapa
México, D.F.