UNIVERSIDAD DE LOS ANDES
FACULTAD DE CIENCIAS

# NOTAS DE MATEMATICA

## SISTEMAS DE EVENTOS DISCRETOS

POR

LUZ E. SOLE

DEPARTAMENTO DE MATEMATICA
MERIDA - VENEZUELA

# SISTEMAS DE EVENTOS DISCRETOS

**Mérida, Diciembre 1994**

Las notas presentes corresponden a una redacción realizada por el Lic. Guelvis E. Mata D. del curso, de postgrado de Matemáticas, Sistemas de Eventos Discretos dictado por la prof. Luz E. Solé. Estas constituyen los fundamentos básicos para el desarrollo del Proyecto de Investigación SISTEMAS DE EVENTOS DISCRETOS MODELADOS POR REDES DE PETRI ESTOCASTICAS.

NOTAS DE MATEMATICAS

N° 157

# SISTEMAS DE EVENTOS DISCRETOS

POR

LUZ E. SOLE

NOTAS DE SEMINARIO

UNIVERSIDAD DE LOS ANDES
FACULTAD DE CIENCIAS
DEPARTAMENTO DE MATEMATICAS
MERIDA - VENEZUELA
1995

CONTENIDO

# INTRODUCCION

En estas notas, estudiaremos el control y la observación de una clase de sistemas conocidos como Sistemas de Eventos Discretos. Las caracteristicas principales de estos sistemas es que ellos son discretos, asincrónicos y posiblemente no determinísticos. Como ejemplos de estos sistemas tenemos a las redes de computación, sistemas flexibles de manufactura, sistemas expertos y procedimientos "start-up", y "shut-down" de plantas industriales. Numerosas proposiciones han aparecido en la literatura de simulación para el modelaje de los Sistemas de Eventos Discretos. Una muestra general de éstos incluyen a los modelos Booleanos, redes de Petri, lenguajes formales y lógica temporal, entre otros. Todos éstos se refieren, de una u otra manera, al problema de cómo verificar el orden en que se presentan los eventos en el sistema. Para esto es usada la lógica, la teoría del lenguaje y la teoria de Autómatas. La variedad de proposiciones reflejan la diversidad de áreas en las cuales los Sistemas de Eventos Discretos juegan un papel importante.

Nuestro objetivo más importante será el determinar las caracteristicas estructurales cualitativas de los problemas de control básicos más relevantes, y la observabilidad en los Sistemas de Eventos Discretos.

La organización de las notas es como sigue. En el capítulo 0 damos las herramientas básicas, que constituyen los fundamentos para los capítulos posteriores. Este incluye algunas nociones de la teoría de Generadores, la teoría de Grafos, y la teoría del Lenguaje formal. En el capítulo 1 definimos los Sistemas de Eventos Discretos y los sistemas producto, dando a éstos una representación por Autómatas finitos. En el capítulo 2 definimos una clase de procesos controlados y controladores (supervisores) de interés, y discutimos varios lenguajes formales asociados con éstos; luego introducimos un criterio que garantiza la existencia de un supervisor, para un sistema controlado de lazo cerrado satisfaciendo algunos requerimientos lingüísticos dados. Posteriormente, damos la noción de supervisor completo, y estudiamos dos problemas de síntesis de supervisores: el problema de marcaje supervisorio (PMS) y el problema de control supervisorio (PCS). Finalmente, definimos una proyección de supervisores, e introducimos un resultado de mucha importancia como lo es el teorema de la estructura cociente. Para finalizar, en el capítulo 3 introducimos las nociones de control minimal y observación minimal, para un sistema de lazo cerrado, dando algunas condiciones para la existencia de la observación minimal y algunos algoritmos para sus cálculos.

# CAPITULO 0.

# PRELIMINARES.

Introducimos en este capítulo algunas nociones de la teoría de generadores, la teoría de lenguajes y la teoría de grafos, las cuales nos permitirán un mejor desarrollo de nuestro propósito en los capítulos posteriores.

## 1. GENERADORES Y LENGUAJES.

Sea $\Sigma$ un conjunto no vacío y consideremos el conjunto de todas las combinaciones finitas formadas con elementos de $\Sigma$, incluyendo la combinación que no contiene símbolos $\theta$, el cual denotamos por $\Sigma^*$. Los elementos de $\Sigma^*$ serán llamados palabras. Cada elemento de $\Sigma$ será llamado letra y $\Sigma$ será llamado alfabeto.

$\Sigma^*$ Tiene estructura de monoide con la operación $\bullet$ llamada concatenación, la cual es definida como sigue: para

$a= (a_1a_2...a_n)$, $b= (b_1b_2...b_m)$ en $\Sigma^*$,

$a\bullet b= (a_1a_2...a_n)\bullet (b_1b_2...b_m) = a_1a_2... a_nb_1b_2...b_m$.

**DEFINICION 0.1.1.** Una palabra $X \in \Sigma^*$ es llamada un prefijo de $Y \in \Sigma^*$, denotado $X \underline{\alpha} Y$, si existe una palabra $Z \in \Sigma^*$ tal que $Y = X \bullet Z$, donde $\bullet$ es la operación de concatenación sobre $\Sigma^*$. Si $Z=\theta$, diremos que $X$ es un prefijo propio de $Y$ y lo denotaremos por $X \alpha Y$.

**OBSERVACION 1.** La relación $\underline{\alpha}$ dada en la definición 0.1.1. es una relación de orden parcial sobre $\Sigma^*$.

**DEFINICION 0.1.2.** Todo subconjunto $L \subseteq \Sigma^*$ es llamado un lenguaje sobre $\Sigma$.

**DEFINICION 0.1.3.** El subconjunto de todos los prefijos de palabras de un lenguaje $L \subseteq \Sigma^*$ es llamado la clausura de $L$ y es denotado por $\bar{L}$, es decir, $\bar{L} = \{X \in \Sigma^* /$ existe $Y \in \Sigma^*, X \bullet Y \in L\}$.

**OBSERVACION 2.** Si $L=\emptyset$, entonces $\bar{L}=\emptyset$, y si $L=\emptyset$, entonces $\theta \in \bar{L}$.

**DEFINICION 0.1.4.** Un lenguaje $L \subseteq \Sigma^*$ es llamado cerrado si $L=\bar{L}$.

Sea $\mathbb{N}=\{1,2,\ \ldots\}$ el conjunto de los enteros positivos y sea $\Sigma$ un alfabeto. Consideremos el conjunto de todas las sucesiones en $\Sigma$ denotado por $\Sigma^{\mathbb{N}}$, es decir, $\Sigma^{\mathbb{N}}=\{X / X: \mathbb{N} \longrightarrow \Sigma\}$.

Para $X \in \Sigma^{\mathbb{N}}$ y $j \in \mathbb{N}$, sea $X(j)$ denotando el j-ésimo elemento de $X$ y $X^j = X(1) \ldots X(j)$ denotando la palabra que consiste de los primeros $j$ elementos de $X$.

**DEFINICION 0.1.5.** $u \in \Sigma^*$ es llamado un prefijo de $X \in \Sigma^{\mathbb{N}}$, denotado $u < X$, si para algún $j \in \mathbb{N}$, $u = X^j$.

**DEFINICION 0.1.6.** Todo subconjunto $B \subseteq \Sigma^{\mathbb{N}}$ es llamado un w-lenguaje sobre $\Sigma$.

Sea $\mathcal{C}$ el conjunto de todos los w-lenguajes sobre $\Sigma$. El prefijo de $B \subseteq \Sigma^{\mathbb{N}}$ es el conjunto que consiste de la palabra $\Theta$ junto con cada palabra en $\Sigma^*$ que sea un prefijo de una sucesión en $B$.

Sea $\mathcal{L}$ el conjunto de todos los lenguajes sobre $\Sigma^*$.
Definimos el operador

$Pr: \mathcal{C} \longrightarrow \mathcal{L}$ por $Pr(B) = \{\Theta\} \cup \{u / u \in \Sigma^*, u < X, X \in B\}$

$= \bigcup_{j>1} \bigcup_{X \in B} X^j = \{e^j \, / \, e \in B, \; j \geq 0\}$.

Se puede probar que dada una sucesión $\{u_k\}$, con $u_1 \underline{\alpha} \; u_2 \underline{\alpha} \ldots$ de elementos de $\Sigma^*$, existe un elemento único $X \in \Sigma^{\mathbb{N}}$ tal que $X^j = u_k$, para $j = |u_k|$, $K \in \mathbb{N}$. Llamaremos a $X$ el límite de $\{u_k\}$.

**DEFINICION 0.1.7.** La Adherencia o límite de $L \subseteq \Sigma^*$ es el w-lenguaje

$L^\infty = Adh(L) = \{X \ / \ X \in \Sigma^{IN}, \ X^j \in L$ para infinitos $j \in IN\}$

De la definición 0.1.7. se sigue que $e \in L^\infty$, si y solo si, existe una sucesión creciente $\{u_k\}$ de elementos de L tal que $e = \lim_{K\to\infty} u_k$. Mas aun, si L es cerrado, entonces $e \in L$, si y sólo si, $e^j \in L$, para todo $j \in IN$.

**DEFINICION 0.1.8.** Un generador es un quíntuple $G = (Q, \Sigma, \delta, q_o, F)$ donde Q es un conjunto cuyos elementos serán llamados estados, $\Sigma$ es un alfabeto cuyos elementos serán llamados entradas, $\delta : Q \times \Sigma \longrightarrow Q$ es una función parcial, es decir, para cada $q \in Q$ fijo, $\delta(q,\sigma)$ es definida solo para algún subconjunto de $\Sigma$ que depende de $q, q_o \in Q$ será llamado estado inicial y $F \subseteq Q$ es un subconjunto de estados llamado conjunto de estados finales (o marcados).

Dado un generador G asumiremos, de acuerdo a nuestro interés, que $\Sigma$ es un alfabeto finito pero no necesariamente Q o F lo son.

Si $\delta(q,\sigma) = p$ está definida, llamaremos al triple $(q,\sigma,p)$ un evento de G.

Para definir el comportamiento de un generador $G=(Q,\Sigma,\delta,q_0,F)$ en elementos de $\Sigma^*$, es necesario extender la función parcial $\delta$ a una función parcial $\tilde{\delta}:Q\times\Sigma^* \longrightarrow Q$. Para esto definimos $\tilde{\delta}:Q\times\Sigma^* \longrightarrow Q$ acordando que $\tilde{\delta}(q,\theta)= q$, para todo $q \in Q$ y $\tilde{\delta}(q,wa)=\delta(\tilde{\delta}(q,w),a)$, para todo $w \in \Sigma^*$, para todo $a \in \Sigma$ y para todo $q \in Q$. Claramente $\tilde{\delta}$ es una extensión de $\delta$, pues $\tilde{\delta}(q,\theta a)=\delta(\tilde{\delta}(q,\theta),a)=\delta(q,a)$, para todo $a \in \Sigma$.

En lo que sigue, no haremos distinción entre $\delta$ y $\tilde{\delta}$, siempre y cuando ambas sean definidas.

**DEFINICION 0.1.9.** Dado un generador $G= (Q,\Sigma,\delta,q_0,F)$. Llamáremos el lenguaje generado por G, denotado L(G), al conjunto $L(G)=\{X \in \Sigma^* \ / \ \delta(q_0,x) \text{ está definida}\}$.

**OBSERVACION 3.** Dado un generador $G=(Q,\Sigma,\delta,q_0,F)$, su lenguaje generado L(G) es cerrado, es decir, $L(G)= \bar{L}(G)$.

**DEFINICION 0.1.10.** Diremos que una palabra $X \in \Sigma^*$ es aceptada por un generador $G= (Q,\Sigma,\delta,q_0,F)$, si $\delta(q_0, x) \in F$. El conjunto de todas las palabras aceptadas por G, denotado $L_{ac}(G)$, es $L_{ac}(G)=\{X \in L(G) \ / \ \delta(q_0,x) \in F\}$, y será llamado el lenguaje aceptado por G.

**EJEMPLO 0.1.1.** Consideremos el generador $G=(Q,\Sigma,\delta,q_o,F)$, donde $Q=\{q_o, q_1, q_2, q_3\}$, $\Sigma=\{0,1\}$, $F=\{q_2\}$ y $\delta$ es definida en la tabla 1.

| | Entradas 0 | 1 |
|---|---|---|
| Estados | | |
| $q_o$ | $q_1$ | $q_2$ |
| $q_1$ | $q_o$ | $q_3$ |
| $q_2$ | $q_3$ | $q_1$ |
| $q_3$ | $q_2$ | $q_o$ |

TABLA 1

La palabra 100110 esta en $L_{ac}(G)$.

En efecto, como $\delta(q_o,1)= q_2$ y $\delta(q_2,0)=q_3$, entonces $\delta(q_o,10)=\delta(\delta(q_o,1),0) = \delta(q_2,0) = q_3$. Ahora, $\delta(q_o,100)=\delta(\delta(q_o,10),0) = \delta(q_3,0)=q_2$, de donde $\delta(q_o,1001)=\delta(\delta(q_o,100),1)= \delta(q_2,1)=q_1$, de donde $\delta(q_o,10011)=\delta(\delta(q_o,1001),1)= \delta(q_1,1)=q_3$, finalmente $\delta(q_o,100110)=\delta(\delta(q_o,10011),0)= \delta(q_3,0)=q_2$. Por lo tanto, $100110 \in L_{ac}(G)$.

**DEFINICION 0.1.11.** Un generador $G=(Q,\Sigma,\delta,q_o,F)$ es llamado Accesible si todo estado de Q es alcanzable desde $q_o$, es decir, para cada $q \in Q$ existe $r \in \Sigma^*$ tal que $\delta(q_o,r)=q$.

Si un generador $G=(Q,\Sigma,\delta,q_o,F)$ no es accesible podemos construir, a partir de G, un nuevo generador tomando el conjunto de todos los estados accesibles desde $q_o$. Denotemos a este subconjunto de Q por $Q_{ac}(G)$. Entonces $Q_{ac}(G)=\{q \in Q$ / existe $r \in \Sigma^*, \delta(q_o,r)=q\}$. Pongamos $Q_{ac,F}=Q_{ac} \cap F$. Si definimos $\delta_{ac}=\delta/Q_{ac} \times \Sigma$ (la restricción de $\delta$ en $Q_{ac} \times \Sigma$), entonces el generador $Ac(G)=(Q_{ac},\Sigma,\delta_{ac},q_o,Q_{ac,F})$ es accesible y es llamado la componente accesible de G. Note que un generador G es accesible si G= Ac(G), es decir, $Q=Q_{ac}$.

**EJEMPLO 0.1.2.** El generador definido en el ejemplo 0.1.1. es un generador accesible.

**EJEMPLO 0.1.3.** El generador $G=(Q,\Sigma,\delta,q_o,F)$, donde $Q=\{q_o,q_1,q_2\}$, $\Sigma=\{0,1\}$, $F=\{q_o\}$ y $\delta$ es definida en la tabla 2(a) no es accesible. Su componente accesible es el generador $Ac(G)=(Q_{ac},\Sigma,\delta_{ac},q_o,F)$, donde $Q_{ac}=\{q_o,q_1\}$ y $\delta_{ac}$ es definida en la tabla 2(b).

| Estados\ Entradas | 0 | 1 |
|---|---|---|
| $q_o$ | $q_1$ | $\emptyset$ |
| $q_1$ | $\emptyset$ | $q_o$ |
| $q_2$ | $\emptyset$ | $q_1$ |

(a)

| Estados \ Entradas | 0 | 1 |
|---|---|---|
| $q_o$ | $q_1$ | $\emptyset$ |
| $q_1$ | $\emptyset$ | $q_o$ |

(b)

TABLA 2

**DEFINICION 0.1.12.** Un generador $G=(Q,\Sigma,\delta,q_o,F)$ es llamado coaccesible si para cada $x \in L(G)$ existe $y \in \Sigma^*$ tal que $xy \in L_{ac}(G)$.

**EJEMPLO 0.1.4.** El generador definido en el ejemplo 0.1.1 es coaccesible.

**DEFINICION 0.1.13.** Un generador G es llamado trim si G es accesible y coaccesible.

**EJEMPLO 0.1.5.** El generador definido en el ejemplo 0.1.1. es trim (ver ejemplos 0.1.2. y 0.1.4).

**OBSERVACION 4.** Dado un generador $G=(Q,\Sigma,\delta,q_o,F)$. si $Q=Q_{co}$, donde $Q_{co}=\{q \in Q$ / existe $r \in \Sigma^*$, $\delta(q,r) \in F\}$, entonces G es coaccesible. Además, si G es trim, entonces $L(G)=\bar{L}_{ac}(G)$.

## 2. GENERADORES Y GRAFOS

**DEFINICION 0.2.1.** Un grafo, denotado por G, es un par $G=(V,E)$, donde V es un conjunto finito no vacío y E es una colección finita de pares no ordenados de $V \times V$ que pueden repetirse. Los elementos de V y E serán llamados respectivamente vértices (o nodos) y aristas (o líneas).

Un grafo G puede ser representado gráficamente identificando los elementos de V con círculos y los elementos de E con líneas.

**EJEMPLO 0.2.1.** Consideremos el grafo $G=(V,E)$, donde $V=\{u,v,w,z\}$ y $E=\{(u,v),(v,v),(v,v),(v,w),(v,w),(v,w),(u,w),(u,w),(w,z)\}$, entonces gráficamente G es dado en la figura 1.

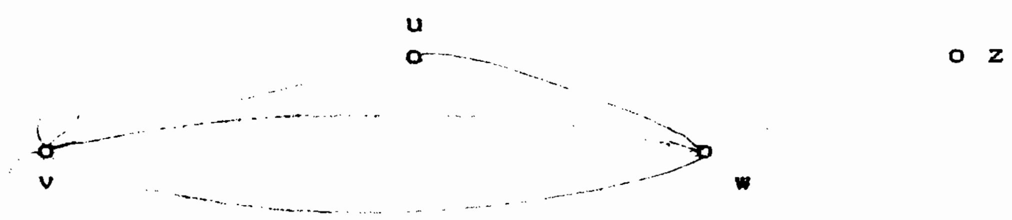



FIGURA 1

**DEFINICION 0.2.2.** Un grafo dirigido D es un par $D=(V,E)$, donde V es como en la definición 0.2.1. y E es una colección finita de pares ordenados de $V \times V$ que pueden repetirse. Los elementos de E serán llamados arcos. Un arco $(u,v) \in E$ es llamado un arco de u a v y lo denotamos por $u \longrightarrow v$.

Un grafo dirigido D puede ser representado gráficamente identificando los elemento de V con círculos y los arcos con líneas dirigidas.

**EJEMPLO 0.2.2.** Consideremos el grafo dirigido $D=(V,E)$, donde $V=\{P,Q,R,S,T\}$ y $E=\{(P,Q),(Q,R),(R,S),(P,S)(P,T), (Q,T),(S,T), (Q,S),(S,Q)\}$. Entonces las representación gráfica de D es dada en la figura 2.

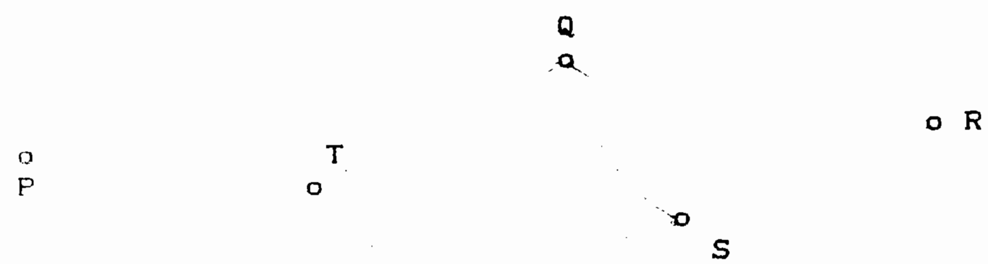



FIGURA 2

**DEFINICION 0.2.3.** Sea G un grafo. Una sucesión de aristas en G, es una sucesión finita de aristas de la forma $(V_o,V_1),(V_1,V_2),\ldots,(V_{m-1},V_m)$, la cual denotamos por $V_o \longrightarrow V_1 \longrightarrow \cdots \longrightarrow V_m$.

Una sucesión de aristas determina una sucesión finita de vértices $V_o, V_1, \ldots, V_m$. Llamarémos a $V_o$ vértice inicial, a $V_m$ vértice final y diremos que la sucesión va desde $V_o$ hasta $V_m$.

**DEFINICION 0.2.4.** Sea G un grafo y sea $(V_o,V_1),(V_1,V_2),\ldots,(V_{m-1},V_m)$ una sucesión de aristas en G. LLamaremos la longitud de la sucesión al número de aristas que determinan dicha sucesión.

**DEFINICION 0.2.5.** Dado un grafo G. Una sucesión de aristas $(V_o,V_1),(V_1,V_2),\ldots,(V_{m-1},V_m)$ es llamada una trayectoria en G, si todas las aristas son distintas. Además, si $V_i \neq V_{i+1}$, para todo $i(1 \leq i \leq m)$ (excepto, posiblemente $V_o = V_m$), entonces la trayectoria será llamada una cadena en G.

**EJEMPLO 0.2.3.** Sea G el grafo cuya representación gráfica es dada en la figura 3. Entonces $v \longrightarrow w \longrightarrow x \longrightarrow y \longrightarrow z \longrightarrow z \longrightarrow x$ es una trayectoria en G y $v \longrightarrow w \longrightarrow x \longrightarrow y \longrightarrow z$ es una cadena en G.

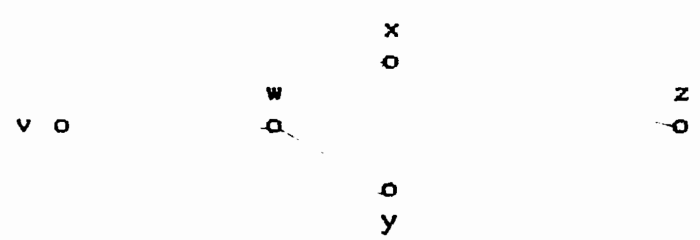



FIGURA 3.

**DEFINICION 0.2.6.** Dado un grafo G. Una trayectoria $(V_o,V_1),(V_1,V_2),\ldots,(V_{m-1},V_m)$ en G es llamada cerrada si $V_o=V_m$.

**EJEMPLO 0.2.4.** En el ejemplo 0.2.3. la trayectoria

$v \longrightarrow w \longrightarrow x \longrightarrow y \longrightarrow z \longrightarrow x \longrightarrow v$ es cerrada.

**DEFINICION 0.2.7.** Un grafo infinito G es un par (V,E), donde V es un conjunto infinito cuyos elementos serán llamados vértices y E es una colección infinita de pares no ordenados cuyos elementos serán llamados aristas. Si V y E son infinitos numerables, entonces G será llamado un grafo numerable.

**DEFINICION 0.2.8.** Un árbol es un grafo dirigido que tiene las propiedades siguientes:

(i) Existe un vértice, llamado raíz, que no tiene predecesor y del cual hay una trayectoria a cada vértice.

(ii) Cada vértice distinto de la raíz tiene exactamente un predecesor.

(iii) Los sucesores de cada vértice son ordenados de izquierda a derecha.

Un grafo dirigido, el cual llamaremos diagrama de transición, es asociado con un generador como sigue: los vértices del grafo corresponden a los estados del generador. Si existe una transición del estado q al estado p con la entrada s, entonces existe un arco etiquetado s del estado q al estado p en el diagrama de transición.

**EJEMPLO 0.2.5.** Sea G= $(Q,\Sigma.\delta,q_o,F)$ un generador, donde $Q=\{q_o,q_1,q_2,\ldots,q_n\}$, $\Sigma=\{\sigma_1,\sigma_2\ldots,\sigma_n\}$, $F=\{q_n\}$ y $\delta$ es definida en la tabla 3. Entonces el diagrama de transición asociado a G es dado en la figura 4.

Entradas.

| Estados \ | $T_1$ | $T_2$ | $T_3$ | . . . | $T_{n-1}$ | $T_n$ |
|---|---|---|---|---|---|---|
| $q_o$ | $q_1$ | $\emptyset$ | $\emptyset$ | . . . | $\emptyset$ | $\emptyset$ |
| $q_1$ | $\emptyset$ | $q_2$ | $\emptyset$ | . . . | $\emptyset$ | $\emptyset$ |
| $q_2$ | $\emptyset$ | $\emptyset$ | $q_3$ | . . . | $\emptyset$ | $\emptyset$ |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | | $q_{n-1}$ | . |
| $q_{n-1}$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | . . . | $\emptyset$ | $q_n$ |
| $q_n$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | . . . | $\emptyset$ | $\emptyset$ |

TABLA 3.

$$\longrightarrow \underset{q_0}{\circ} \xrightarrow{\sigma_1} \underset{q_1}{\circ} \xrightarrow{\sigma_2} \underset{q_2}{\circ} \xrightarrow{\sigma_3} \cdots \xrightarrow{\sigma_n} \underset{q_n}{\circ} \longrightarrow$$

FIGURA 4

**OBSERVACION 5.** La flecha entrante en $q_o$ y la saliente en $q_n$ son usadas para indicar respectivamente el estado inicial y el estado final del Generador.

## 0. EJERCICIOS PROPUESTOS

0.1 Pruebe que si $\Gamma: \Sigma^{\mathbb{N}} \times \Sigma^{\mathbb{N}} \longrightarrow \mathbb{R}$ es definida por

$$\Gamma(x,y) = \begin{cases} 1/j \ , & \text{si } x^{j-1} = y^{j-1},\ x^j \neq y^j, \\ 0 \ , & \text{si } x=y \end{cases} ,$$

entonces el par$(\Sigma^{\mathbb{N}}, \Gamma)$ es un espacio métrico.

0.2 Pruebe que el operador $P_r$ es monótono, es decir, $B_1 \subseteq B_2 \Rightarrow P_r(B_1) \subseteq P_r(B_2)$ y que para cada familia indizada de w-lenguajes $\{B_\alpha \subseteq \Sigma^{\mathbb{N}},\ \alpha \in \Omega\}$, $P_r(\cup B\alpha) = \cup P_r(B_\alpha)$ y $P_r(\cap B_\alpha) \subseteq \cap P_r(B_\alpha)$.

0.3 Pruebe que si $u_1 \underline{\alpha}\ u_2 \underline{\alpha} \ldots$ es una sucesión creciente de elementos de $\Sigma^*$, entonces existe un único elemento $X \in \Sigma^{\mathbb{N}}$ tal que $X^j = u_k$ para $j= |u_k|$, $K \in \mathbb{N}$.

0.4 Pruebe que el operador $L^\infty$ es monótono $(K_1 \subseteq K_2 \Rightarrow K_1^\infty \subseteq K_2^\infty)$ y que para cada familia indizada de lenguajes $\{ L_\alpha \subseteq \Sigma^*,\ \alpha \in \Omega \}$ se tiene que $\cup L^\infty_\alpha \subseteq (\cup L_\alpha)^\infty$ y $(\cap L_\alpha)^\infty \subseteq \cap L^\infty_\alpha$.

0.5 Pruebe en el ejercicio 0.4 que $\cup L^\infty_\alpha = (\cup L_\alpha)^\infty$ si $\Omega$ es finito, y que $\cap L^\infty_\alpha = (\cap L_\alpha)^\infty$ si $L_\alpha$ es cerrado para cada $\alpha \in \Omega$.

0.6 Pruebe lo siguiente: para $B \subsetneq \Sigma^{\mathbb{N}}$ y $K \subsetneq \Sigma^{*}$ cerrado se tiene que $B \subseteq P_r(B)^{\infty}$ y $P_r(K^{\infty}) \subseteq K$ ¿ cuándo $K = P_r(K^{\infty})$?

0.7 Denotamos a la topología clausura de un w-lenguaje $B \subseteq \Sigma^{\mathbb{N}}$, con respecto a la métrica definida en el ejercicio 0.1, por $\bar{B}$. Pruebe que $\bar{B} = P_r(B)^{\infty}$.

0.8 Pruebe que en el ejercicio 0.6 $P_r(B)^{\infty} = B$, sí y sólo sí, B es un conjunto cerrado en la topología métrica.

# CAPITULO 1

# SISTEMAS DE EVENTOS DISCRETOS Y SU RELACION CON LOS GENERADORES DE ESTADO FINITO

## 1. SISTEMAS DE EVENTOS DISCRETOS

Un Sistema de Eventos Discretos (SED) consiste de un conjunto de eventos junto con una especificación de los ordenes posibles en los cuales estos eventos pueden ocurrir. Para formalizar estas nociones. Sea $\Sigma$ un conjunto finito de eventos. El comportamiento de un SED puede ser modelado como un lenguaje cerrado $L \subseteq \Sigma^*$. En este modelo L representa el conjunto de todas las palabras de eventos que el SED puede generar. Una extensión natural de este modelo es considerar la adherencia de L, o un subconjunto de ésta, para discutir el comportamiento limite del SED. Para esto modelamos un SED como un par $A=(L,S)$, donde L es un lenguaje cerrado de $\Sigma^*$ y S es un subconjunto de $L^\infty$. Modelando el comportamiento de un SED con un conjunto de sucesiones tenemos algunas ventajas, por ejemplo modelar procesos no deterministicos y distinguir elementos persistentes y transitorios del comportamiento. También encontramos que toda palabra en L es un prefijo de alguna sucesión en S. Esto puede ser interpretado diciendo que el sistema nunca está bloqueado permanentemente.
Cuando $P_r\ (S) = L$ dirémos que A es no bloqueado.

## 2. SISTEMAS PRODUCTO

En esta sección introducimos una clase de estructura de SED's llamada sistemas producto. Un sistema producto está compuesto de un conjunto finito de componentes asincrónicas interactuando. Estos sistemas se presentan cuando modelamos operaciones concurrentes de algunos Sistemas de Eventos Discretos asincrónicos. Una de las dificultades en los sistemas producto es que el número de estados aumenta exponencialmente con el número de componentes.

Sean $A_i = (L_i, S_i)$, $i = 1,2,\ldots, p$, Sistemas de Eventos Discretos sobre alfabetos disjuntos $\Sigma_i$, $i = 1,2,\ldots, p$. Para cada $A_i$ asumiremos $S_i \neq \emptyset$ y $P_r(S_i) = L_i$.

Sea $\Sigma = \bigcup_{i=1}^{p} \Sigma_i$. Definimos la proyección $P_i: \Sigma^* \rightarrow \Sigma^*_i$ como sigue.

Si $w \in \Sigma^*$ y $|W| = 1$, entonces, $P_i(w) = \begin{cases} w \text{ si } w \in \Sigma_i \\ \theta \text{ si } w \notin \Sigma_j,\ j=i \end{cases}$ .

Si $w \in \Sigma^*$ y $w=w_1\ldots w_k$, entonces $P_i(w) = P_i(w_1)\ldots P_i(w_k)$.

Si $w \in \Sigma^*$, $|w|=k$ y $b \in \Sigma$, entonces $P_i(wb)=P_i(w)P_i(b)$.

**DEFINICION 1.2.1** Una sucesión $X \in \Sigma^{IN}$ es llamada $\Sigma_i$-recurrente si $X(j) \in \Sigma_i$, para infinitos $j \in IN$.

En la definición 1.2.1. $X_i$ denota la única subsucesión de X consistiendo de los elementos de $\Sigma_i$.

Extendemos la proyección $P_i$ a una función parcial

$$\bar{P}_i(X) = \begin{cases} X_i, & \text{si } X \text{ es } \Sigma_i\text{- recurrente} \\ \text{indefinida en otro caso.} \end{cases}$$

El producto $A = \prod_{i=1}^{p} A_i$ es definido como en el SED (L,S), donde $L = \{ w \,/\, w \in \Sigma^*,\ P_i(w) \in L_i,\ i=1,2,\ldots,p \}$ y $S = \{ X / X \in \Sigma^{IN},\ \bar{P}_i(X) \in S_i,\ i = 1,2,\ldots,p \}$, es decir, $X \in S$, si y sólo si, para cada i, $X(t) \in \Sigma_i$, para infinitos $t \in IN$. Si $X_i$ es la subsucesión de X consistiendo de todos los valores en $\Sigma_i$, entonces $X_i \in S_i$.

Como $S_i \neq \emptyset$, $i=1,2,\ldots,p$ se tiene que $S \neq \emptyset$ . Además, de la definición de S y sabiendo que $P_r(S_i)=L_i$, $i=1,2,\ldots,p$.Se sigue que $P_r(S)=L$. Así, el sistema producto A es no bloqueado.

# 3. SISTEMAS DE EVENTOS DISCRETOS Y GENERADORES

Para propósitos de computación es necesario tener una representación finita de los lenguajes L y S. Esto puede ser hecho de distintas maneras, por ejemplo por redes de Petri, por procesos generalizados semi-markovianos y por autómatas finitos (generadores de estado finito). Usaremos los Autómatas finitos para representar Sistemas de Eventos Discretos. Esto no quiere decir que esta representación sea la mejor, ya que son claras las limitaciones de un Autómata finito.

**DEFINICION 1.3.1** Un generador para un lenguaje $L \subseteq \Sigma^{*}$ es un Autómata de estado finito (o Autómata finito) $G=(Q,\Sigma,\delta,q_0,F)$.

Asumiremos que G en la definición 1.3.1 es accesible.

Todo lenguaje cerrado $L \subseteq \Sigma^{*}$ tiene una representación por autómatas finitos. El comportamiento límite S es especificado como sigue: para cada sucesión de eventos $X \in L(G)^{\infty}$ hay una única trayectoria de estados $S_x: \mathbb{N} \longrightarrow Q$ satisfaciendo $S_x(j)=\delta(X^j,q_0)$. La sucesión X y la trayectoria $S_x$ son llamadas admisibles si $S_x$ visita a F infinitas veces. El conjunto de sucesiones de eventos generadas por G es entonces definido por $S(G)=\{X \ / \ X \in L(G)^{\infty},\ S_X$ es admisible$\}$.

Un w-lenguaje S puede ser representado de esta manera, si y sólo si, S es la adherencia de un lenguaje regular. Este constituye un subconjunto propio de los w-lenguajes regulares.

## 4. REPRESENTACION DE SISTEMAS PRODUCTO

Asumiremos que cada Sistema de Eventos Discretos $A_i=(L_i,S_i)$, $i=1,2,\dots,p$ tiene una representación por un autómata finito $G_i=(Q_i,\Sigma_i,\delta_i,q_{0i},F_i)$. Sean $|Q_i|$ el cardinal de $Q_i$ y $n=\text{máx}\ \{|Q_i| \ / \ 1 \le i \le P\}$. El generador producto $G=\prod_{i=1}^{p} G_i$ es definido por $G=(Q,\Sigma,\delta,q_o,F)$, donde $\Sigma=\bigcup_{i=1}^{p} \Sigma_i$, $Q=\prod_{i=1}^{p} Q_i$, $q_o=(q_{01},q_{02},\dots,q_{op})$ y $\delta: Q \times \Sigma \longrightarrow Q$ es definida por

$$\delta((q_1,\dots,q_i,\dots,q_p),\sigma)=\begin{cases}(q_1,\dots,\delta_i(q_i,\sigma),\dots,q_p), \text{si } \delta_i(q_i,\sigma)\\ \text{indefinida en otro caso.}\end{cases}$$

y $F=\{q \ / \ q \in Q,\ q_i \in F_i,\ i=1,2,\dots,p\}$.

Para $i=1,2,\dots,p$ sea $Y_i=\{q \ / \ q \in Q,\ q_i \in F_i\}$. Estos conjuntos jugarán el papel de los conjuntos $F_i$ para el generador $G_i$.

Para cada sucesión de eventos $e \in L(G)^{\infty}$ hay una única trayectoria $S_e$. La sucesión e y la trayectoria $S_e$ son admisibles, si y sólo si, para cada $i=1,2,...,p$, $e$ es $\Sigma_i$-recurrente y $S_e$ es $y_i$-recurrente, es decir, $S_e$ visita a $y_i$ infinitas veces. Luego, el comportamiento de G es definido por $S(G)=\{ e \ / \ e \in L(G)^{\omega}, S_e \text{ es admisible}\}$.

## 1. EJERCICIOS PROPUESTOS

1.1 Pruebe que en $S(G) \subseteq L(G)^{\infty}$ la igualdad es cierta si $F=Q$.

1.2 Pruebe que $P_r(S(G)) \subseteq L(G)$.

1.3 Pruebe que $P_r(S(G))=L(G)$, si y sólo si, para todo $q \in Q-F$ existe $w \in \Sigma^*$ tal que $\delta(q,w) \in F$, es decir, G es coaccesible, y para todo $q \in F$ existe un evento $\sigma \in \Sigma$ tal que $\delta(q,\sigma)$ está definida.

1.4 Pruebe que $L(G)=\{w \ / \ w \in \Sigma^*, P_i(w) \in LG_i), i=1,2,...,p\}$ en la representación de sistemas producto.

1.5 Pruebe que $S(G)=\{e \ / \ e \in \Sigma^{IN}, P_i(e) \in S(G_i), i=1,2,...,p\}$ en la representación de sistemas productos.

CAPITULO 2

# CONTROL SUPERVISORIO DE UNA CLASE DE SISTEMAS DE EVENTOS DISCRETOS

En este capítulo introducimos una clase de Sistemas de Eventos Discretos controlables junto con algunos conceptos y resultados generales, relacionados con su control o "supervisión". Nuestro gran resultado será el llamado "Teorema de la Estructura Cociente".

## 1. SISTEMAS DE EVENTOS DISCRETOS CONTROLADOS

Para un generador $G=(Q,\Sigma,\delta,q_0,F)$, introducimos un significado de control, es decir, G será interpretado como una "planta" (objeto a ser controlado) en teoría de control. Tomemos $\Sigma_c \subsetneq \Sigma$ un subconjunto distinguido del alfabeto. Diremos que un evento $(q,\sigma,p)$, etiquetado $\sigma$, es un evento controlado si $\sigma \in \Sigma_c$. Sea $\Gamma=\{0,1\}^{\Sigma_c}$ el conjunto de todas las asignaciones binarias para los elementos de $\Sigma_c$. Cada asignación $r \in \Gamma$, es decir, cada función $r:\Sigma_c \longrightarrow \{0,1\}$ es un modelo o patrón de control. Un evento etiquetado $\sigma$ será llamado posible por r si $r(\sigma)=1$ o no posible si $r(\sigma)=0$. Extendemos cada $r \in \Gamma$

a una función $r:\Sigma \longrightarrow \{0,1\}$ definiendo $r(\sigma)=1$, para cada $\sigma \in \Sigma-\Sigma_c$. Si $\delta$: $Q \times \Sigma \longrightarrow Q$ es la función de transición de G, definimos una función de transición argumentada

$\delta_c$: $Q \times \Sigma \times \Gamma \longrightarrow Q$ por

$$\delta_c(q,\sigma,r)=\begin{cases}\delta(q,\sigma), \text{ si } \delta(q,\sigma) \text{ está definida y } r(\sigma)=1\\ \text{indefinida en otro caso.}\end{cases}$$

Formalmente, el objeto $G_c=(Q,\Sigma\times\Gamma,\delta_c,q_0\ F)$ es un generador construido a partir de G por una especificación de $\Sigma_c$; por lo tanto, interpretamos a $G_c$ como una versión de G que admite control externo, como sigue: para cada $r \in \Gamma$ fijo, hay un generador G(r) formado eliminando del diagrama de transición de G aquellos eventos etiquetados $\sigma$ tales que $r(\sigma)=0$, es decir, aquellos eventos que el patrón de control r inhabilita. Por lo tanto, la acción del control externo podría consistir simplemente en encender el patrón de control a través de una sucesión $r, r', r'', r''', \ldots$ de elementos de $\Gamma$. Una estructura $G_c$ como fué descrita anteriormente será llamada un Sistema de Eventos Discreto Controlado (SEDC).

**EJEMPLO 2.1.1** Un usuario de un recurso puede ser modelado como un SEDC, con tres estados O(ocioso), P(petición) y U (uso), y con transiciones dadas en la figura 1; con alfabeto $\Sigma=\{\alpha,\beta,\tau\}$ y $\Sigma_c=\{\beta\}$.

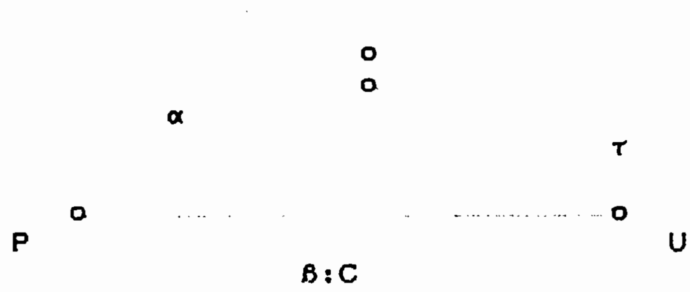



FIGURA 1

los dos modelos o patrones de control corresponden a la evaluación C=0 ó C=1, de la variable de control C. Una trànsición de P a U puede ocurrir solamente cuando C=1.

## 2. SUPERVISORES

En esta sección diseñaremos un controlador que ponga en funcionamiento los patrones de control en cada una de las formas de un SEDC dado. Como es sabido, un SEDC funciona en obediencia a varios constreñimientos. Este controlador será llamado supervisor. Formalmente, un supervisor $\mathcal{S}$ es un par $\mathcal{S}=(S,\phi)$, donde $S=(X,\Sigma,\$,X_o,F^1)$ es un generador y $\phi:X \longrightarrow \Gamma$ es una función que mapea estados de X en patrones de control r; para cada $x \in X$, $r=\phi(x) \in \{0,1\}^{\Sigma c}$. Como antes, extendemos cada $\phi(x)$ a una función $\phi(x):\Sigma \longrightarrow \{0,1\}$, con $\phi(x)(\sigma)=1$, para cada $\sigma \in \Sigma-\Sigma_c$. Asumiremos que S es accesible y llamaremos a $\phi$ la función de retroalimentación de estado. Interpretamos a S como un aparato que ejecuta una sucesión de transiciones de estados (de acuerdo a $) en respuesta a una apropiada palabra de entrada $w \in \Sigma^*$. Así, podríamos acoplar $\mathcal{S}$ a $\mathcal{S}_c$ en un lazo de retroalimentación permitiendo las transiciones de estados de S forzadas por $\mathcal{S}_c$, y requiriendo que $\mathcal{S}_c$ sea construida por los patrones de control sucesivos, determinados por los estados de S. Formalmente, definimos la función parcial

$\mathcal{S}_c \times \$ : Q \times X \times \Sigma \longrightarrow Q \times X$ por

$(\mathcal{S}_c \times \$)(q,x,\sigma)=(\mathcal{S}_c(q,\sigma,\phi(x)),\$(x,\sigma))$. Es claro que

$(\mathcal{S}_c \times \$)(q,x,\sigma)$ es definida, si y sólo si, $\mathcal{S}(q,\sigma)$ y $\$(x,\sigma)$ están definidas, y $\phi(x)(\sigma)=1$. Esto produce el generador $(Q \times X,\Sigma,\mathcal{S}_c \times \$,(q_o,x_o),F \times F^1)$. Definimos el Sistema de Eventos Discreto supervisado (SEDS), denotado por $\mathcal{S}/G_c$, como el generador accesible $\mathcal{S}/G_c=Ac(Q\times X,\Sigma,\mathcal{S}_c\times\$,(q_o,x_o),F\times F^1)$. Asumiremos que $\mathcal{S}_c \times \$$ ha sido extendida a elementos de $\Sigma^*$ como en el capitulo 0 (sección 1). Queremos interpretar al lenguaje $L(\mathcal{S}/G_c)$ como el conjunto de todas las palabras posibles de eventos que pueden ocurrir, cuando $\mathcal{S}$ es acoplado a $\mathcal{S}_c$. Para esto es necesario asegurar que las transiciones de S sean definidas, siempre que éstas ocurran en G y sean permitidas por $\phi$. Para formalizar estas relaciones diremos que $\mathcal{S}$ es completo con respecto a $G_c$ si se cumple lo siguiente: para todo $s \in \Sigma^*$ y para todo $\sigma \in \Sigma$ se tiene que

(i) $s \in L(\mathcal{S}/G_c)$,

(ii) $s\sigma \in L(G)$,

(iii) $[(\phi o \$)(X_o,s)](\sigma)=1$; (i),(ii) y (iii) implican que

(iv) $s\sigma \in L(\mathcal{S}/G_c)$

## 3. LENGUAJES DE $\mathcal{S}/G_c$

Sea $G_c$ un SEDC construido a partir de un generador $G=(Q,\Sigma,\delta,q_o,F)$. Nos referiremos a $L(G)$ y a $L_{ac}(G)$ como los lenguajes del SED no controlado. Sean $\mathcal{S}$ un supervisor de G, $L(\mathcal{S}/G_c)$ el lenguaje generado por $\mathcal{S}/G_c$ y $L_{ac}(\mathcal{S}/G_c)$ el lenguaje aceptado por $\mathcal{S}/G_c$.

**DEFINICION 2.3.1** El lenguaje $L_c(\mathcal{S}/G_c)=L(\mathcal{S}/G_c)\cap L_{ac}(G)$ será llamado lenguaje controlado por $\mathcal{S}$ en G, es decir, $L_c(\mathcal{S}/G_c)$ consiste de aquellas palabras de $L_{ac}(G)$ que sobreviven ante la presencia del supervisor.

Es claro, a partir de la definición 2.3.1, que $L_{ac}(\mathcal{S}/G_c) \subseteq L_c(\mathcal{S}/G_c) \subseteq L_{ac}(G)$. Más aún, si G es trim, entonces $\bar{L}_{ac}(\mathcal{S}/G_c) \subseteq \bar{L}_c(\mathcal{S}/G_c) \subseteq \bar{L}(\mathcal{S}/G_c)=L(\mathcal{S}/G_c)\subseteq L(G)=\bar{L}_{ac}(G)$.

**EJEMPLO 2.3.1** Sea $G=(Q,\Sigma,\delta,q_o,F)$ un generador donde $Q=\{q_o,q_1\}$, $\Sigma=\{\alpha,\beta,\tau\}$, $F=\{q_o\}$ y $\delta$ es dada en la figura 2. Entonces, $L_{ac}(G)=(\alpha\tau^*\beta)^*$.

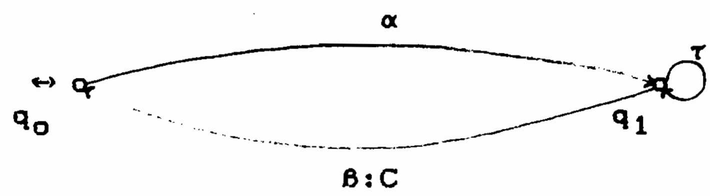



FIGURA 2

Consideremos dos supervisores diferentes, cada uno especificado por su diagrama de transición como sigue:

(i) El primer supervisor es dado en la figura 3.

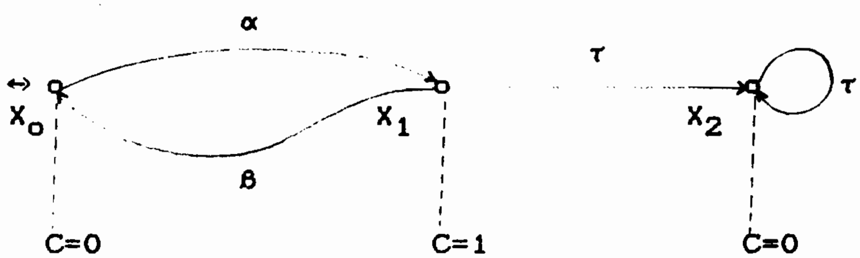



FIGURA 3

luego, el diagrama de transición para $S/G_c$ es dado en la figura 4.

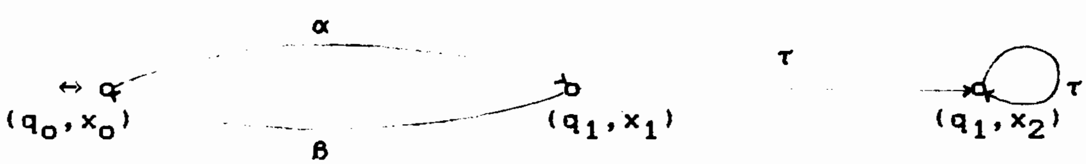



FIGURA 4

Ahora, $L(\mathcal{S}/G_c)=(\alpha\beta)^*(\theta+\alpha\tau^*)$, donde + significa unión,

$L_c(\mathcal{S}/G_c)= L(\mathcal{S}/G_c)\cap L_{ac}(G)=(\alpha\beta)^*=L_{ac}(\mathcal{S}/G_c)$ y

$L_c(\mathcal{S}/G_c)=(\alpha\beta)^*(\theta+\alpha)\ L(\mathcal{S}/G_c)$.

(ii) El segundo supervisor es dado en la figura 5.

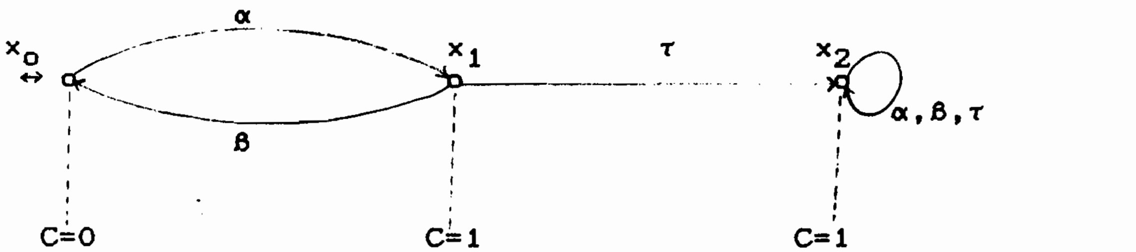
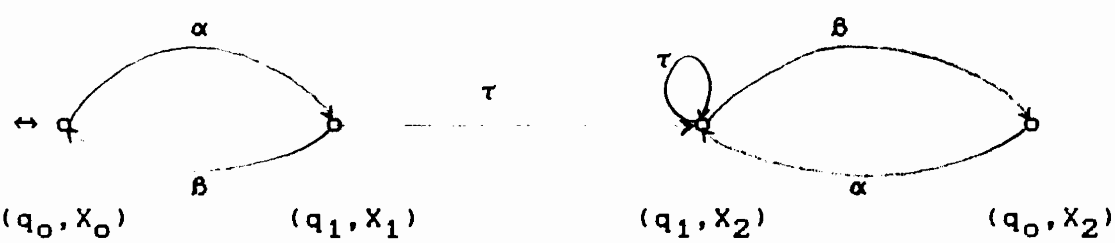



FIGURA 5.

luego, el diagrama de transición para $\mathcal{S}/G_c$ es dado en la figura 6.




Aquí, $L_c(\mathcal{S}/G_c)=L_{ac}(G)=(\alpha\tau^*\beta)^*$, $L_{ac}(\mathcal{S}/G_c)=(\alpha\beta)^*\subseteq L_c(\mathcal{S}/G_c)$ y

$\bar{L}_{ac}(\mathcal{S}/G_c)\subseteq \bar{L}_c(\mathcal{S}/G_c)$.

## 4. MARCACION Y CONTROL

Un supervisor $\mathcal{S}$ ejecuta dos tareas esencialmente independientes, estas son, marcación (descrita por $L_{ac}(\mathcal{S}/G_c)$) y control (descrito por $L_c(\mathcal{S}/G_c)$ y $L(\mathcal{S}/G_c)$). Si un comportamiento controlable dado es una realización, entonces cualquier marcación es simultáneamente una realización que es consistente con el comportamiento controlable.

Sin pérdida de generalidad asumiremos que el generador G es trim, es decir, $L(G)=\bar{L}_{ac}(G)$.

**DEFINICION 2.4.1** Sea $L \subseteq \Sigma^*$. Un reconocedor M para L es un generador accesible G tal que $L_{ac}(G)=L$.

**OBSERVACION 1** Aunque un reconocedor y un generador accesible no son formalmente diferentes, nosotros interpretamos un reconocedor M como un quintuple $M=(Q,\Sigma,\delta,q_0,F)$, el cual es un aparato, como un supervisor, que es activado por palabras en $\Sigma^*$; esta acción es entonces para "reconocer" las palabras de un lenguaje L, consideradas como palabras de entradas para M.

**PROPOSICION 2.4.1** (i) para cada sublenguaje $K \subseteq L_{ac}(G)$ existe un supervisor completo $\mathcal{S}$ tal que, para $\mathcal{S}/G_c$, se tiene que $L(\mathcal{S}/G_c)=L(G)$ y $L_{ac}(\mathcal{S}/G_c)=K$.
(ii) Sea L un sublenguaje cerrado de $L(G)$. Si existe un supervisor completo $\mathcal{S}$ para el cual $L(\mathcal{S}/G_c)=L$, entonces para todo sublenguaje K de $L \cap L_{ac}(G)$ existe un supervisor completo $\mathcal{S}_k$ tal que $L(\mathcal{S}_k/G_c)=L$ y $L_{ac}(\mathcal{S}_k/G)=K$.

**PRUEBA** (i) Sea $S=(X,\Sigma,\$,X_o,F^1)$ un reconocedor para K. Agregamos un nuevo estado en X, si es necesario, para arreglar que $\$(x,\sigma)$ sea definida para todo $(x,\sigma) \in X \times \Sigma$. Definimos $\phi: X \longrightarrow \{0,1\}^{\Sigma}$ por $\phi(x)(\sigma)=1$, $x \in X, \sigma \in \Sigma$.
Es claro que $\mathcal{S}=(S,\phi)$ tiene las propiedades requeridas.

(ii) Sea $T=(Y,\Sigma,\Phi,Y_o,F^1)$ un reconocedor para K. Como en (i), agregamos un estado adicional en Y, si es necesario,para arreglar que $\Phi(y,\sigma)$ sea definida, para todo $(y,\sigma) \in Y \times \Sigma$. Sea $\mathcal{S}=(S,\phi)$ un supervisor completo para el cual $L(\mathcal{S}/G_c)=L$. Definimos el supervisor $\mathcal{S}_k=(S',\Phi')$, con $S'=(X',\Sigma,\$',X_o',F')$ donde $X'=X \times Y$, $\$'(x,y,\sigma)=(\$((x,\sigma),\Phi(y,\sigma))$, $X_o'=(X_o,Y_o)$, $F'=X \times F^1$ y $\phi'(x,y)=\phi(X)$. Ya que la acción del control de $\mathcal{S}_k$ es la misma que la de $\mathcal{S}$, es claro que $L(\mathcal{S}_k/G_c)=L$, y obviamente $L_{ac}(\mathcal{S}_k/G_c)=K$. Además, $\$'(x,y,\sigma)$ es definida solamente cuando $\$(x,\sigma)$ es definida. Por lo tanto, $\mathcal{S}_k$ es completo con respecto a $G_c$.

## 5. CONTROLABILIDAD

En esta sección, introducimos una definición de controlabilidad que jugará un papel importante en la caracterización de lenguajes generados por estructuras $\mathcal{S}/G_c$, con un SEDC $G_c$ dado, y una escogencia apropiada del supervisor completo $\mathcal{S}$.

Sea $G_c$ SEDC fijo, construido a partir de un generador $G=(Q,\Sigma,\delta,q_0,F)$ y supongamos que G es trim. Pongamos $\Sigma_u=\Sigma-\Sigma_c$, es decir, $\Sigma_u$ es el conjunto de etiquetas de eventos que no pueden ser inhabilitados. Sean K y L dos lenguajes arbitrarios de $\Sigma^*$.

**DEFINICION 2.5.1.** Dirémos que K es L-cerrado si $K=\bar{K}\cap L$.

**DEFINICION 2.5.2** Dirémos que K es $(\Sigma_u,L)$-invariante si $\bar{K}\Sigma_u\cap L \subseteq \bar{K}$.

**DEFINICION 2.5.3** Dirémos que K es controlable si $K \subseteq L(G)$ y K es $(\Sigma_u,L(G))$-invariante, es decir, $\bar{K}\Sigma_u\cap L(G) \subseteq \bar{K}$.

De la definición 2.5.1 se sigue que un sublenguaje K de L es L-cerrado, si y sólo si, cualquier prefijo de K, que es una palabra de L, es también una palabra de K.

El lenguaje $\bar{K}\Sigma_u \cap L$ consiste de todas las palabras $s'=s\sigma$, donde $s' \in L$, $s \in K$ y $\sigma \in \Sigma_u$. Si pensamos a L como la representación del "comportamiento posible fisicamente", y a K como el "comportamiento admisible legalmente", entonces la palabra $s\sigma$ es una palabra formada por una palabra admisible legalmente s, seguida de una letra (o símbolo) no controlado $\sigma$ que es posible fisicamente. K es precisamente $(\Sigma_u, L)$-invariante cuando todas estas palabras son admisibles legalmente, es decir, ciertas instancias del comportamiento no controlado son legales.

Finalmente, pensando en L(G) como el lenguaje del sistema no controlado, es decir, como el comportamiento no controlado posible fisicamente de nuestro SEDC, tenemos que K es controlable si todo prefijo $s \in \bar{K}$ es posible fisicamente, y toda palabra $s\sigma$ posible fisicamente, con $S \in K$ y $\sigma$ no controlado está en $\bar{K}$.

**PROPOSICION 2.5.1** Sean $K_1, K_2 \subseteq L_{ac}(G)$ y $K_3 \subseteq L(G)$, con $K_3$ no vacío. Existe un supervisor completo $\mathcal{S}$ tal que, para el sistema $\mathcal{S}/G_c$, $L_{ac}(\mathcal{S}/G_c)=K_1$, $L_c(\mathcal{S}/G_c)=K_2$, y $L(\mathcal{S}/G_c)=K_3$, si y sólo si, $K_1 \subseteq K_2$, $K_2=K_3 \cap L_{ac}(G)$ y $K_3$, es cerrado y controlable.

**PRUEBA** Sea $\mathcal{S}$ un supervisor completo tal que $L_{ac}(\mathcal{S}/G_c)=K_1$, $L_c(\mathcal{S}/G_c)=K_2$, y $L(\mathcal{S}/G_c)=K_3$. Como $L_{ac}(\mathcal{S}/G_c) \subseteq L_c(\mathcal{S}/G_c) \subseteq L_{ac}(G)$, entonces $K_1 \subseteq K_2$; y por la definición 2.3.1 tenemos que $K_2=K_3 \cap L_{ac}(G)$. Para probar que $K_3$ es controlable, supongamos que $s\sigma \in L(G)$, con $s \in \bar{K}_3=K_3= L(\mathcal{S}/G_c)$ y $\sigma \in \Sigma_u$. Si $\mathcal{S}=(S,\phi)$, con $S=(X,\Sigma,\$,X_o,F^1)$, entonces $(q,x)=(\delta_c \times \$)(q_o,x_o,s)$ es definida. Como $\sigma \in \Sigma_u$, se tiene que $\phi(x)(\sigma)=1$, y como $s\sigma \in L(G)$ se sigue que $q'=\delta(q,\sigma)$ está definida. Por lo tanto, $\delta_c(q,\sigma,\phi(x))=q'$. Ahora, como $\mathcal{S}$ es completo con respecto a $G_c$, tenemos que $x'= \$(x,\sigma)$ está definida. De esto se sigue que $(\delta_c \times \$)(q,x,\sigma)=(\delta_c(q,\sigma,\phi(x)),\$(x,\sigma))=(q',x')$ está definida. Luego, $s\sigma \in L(\mathcal{S}/G_c)=K_3=\bar{K}_3$; así $K_3$ es controlable.

Recíprocamente supongamos que $K_1 \subseteq K_2$, $K_2=K_3 \cap L_{ac}(G)$ y $K_3$ es cerrado y controlable, entonces por la proposición 2.4.1 es suficiente construir un supervisor completo $\mathcal{S}$ tal que $L(\mathcal{S}/G_c)=K_3$. Para esto, sea $S=(X,\Sigma,\$,X_o,F^1)$ un reconocedor trim para $K_3$, entonces $F^1=X$ (recuerde que $K_3$ es cerrado). Por lo tanto, $\$(x_o,s)$ está definida, si y sólo si, $s \in K_3$. Para cada $x \in X$, sean

$\Sigma_x{}^0=\{\sigma$ / existe $s \in k_3$, $\$(x_o,s)=x$, $s\sigma \in L(G)$, $s\sigma \notin k_3\}$ y

$\Sigma_x{}^1=\{\sigma$ / existe $s \in K_3$, $\$(x_o,s)=x$, $s\sigma \in K_3\}$.

Afirmamos que $\Sigma^0{}_x \cap \Sigma^1{}_x=\emptyset$, para cada $x \in X$. En efecto, la afirmación se sigue por ser $S$ un generador trim para $K_3$.

Supongamos que $\sigma \in \Sigma^{o}{}_{x} \cap \Sigma^{1}{}_{x}$, con $s^{o} \in K_3$, $\$(x_o,s^{o})=x$, $s^{o}\sigma \notin K_3$, $s' \in K_3$, $\$(x_o,s')= x$ y $s'\sigma \in K_3$. Entonces,
$\$(x,\sigma)=\$(\$(x_o,s^{o}),\sigma)=\$(x_o,s^{o}\sigma)$ lo cual es contradictorio, ya que $s^{o}\sigma \notin K_3$. Por otro lado,
$\$(x,\sigma)=\$(\$(x_o,s'),\sigma)=\$(x_o,s'\sigma)$ está definida ya que $s'\sigma \in K_3$, pero nuevamente encontramos contradicción.

$\Sigma^{o}{}_{x} \subseteq \Sigma_c$, pues $K_3$ es controlable. Sea $\phi:X \longrightarrow \{0,1\}^{\Sigma}$ cualquier función tal que, si $\phi(x)= r$, entonces $r(\Sigma^{o}{}_{x})=0$, $r(\Sigma^{1}{}_{x})=1$ y $r(\Sigma_u-\Sigma^{1}{}_{x})=1$.

Es claro que la afirmación probada anteriormente es requerida para que $\phi$ esté bien definida. Ahora, sea $\mathbb{S}=(S,\phi)$. Probaremos que $L(\mathbb{S}/G_c)=K_3$. Por la definición de S es claro que $L(\mathbb{S}/G_c) \subseteq K_3$. Para la otra inclusión, usaremos inducción sobre la longitud de s ($|s|$) de las palabras s que están en $L(G)$. Si $|s|=1$, es decir, $s=\sigma$, para algún $\sigma \in \Sigma$, entonces $\sigma \in \Sigma_{xo}{}^{1}$ si $\sigma \in K_3$; así $\sigma \in L(\mathbb{S}/G_c)$ si $\sigma \in K_3$. Para un lenguaje $L \subseteq \Sigma^{*}$ escribimos $L^{(j)}=\{s \ / \ s \in L, |s|=j\}$, $j= 0,1,2,\ldots$; note que $L= \bigcup_{j=0}^{\infty} L^{(j)}$. Supongamos por inducción que $L^{(i)}(\mathbb{S}/G_c)=K_3^{(i)}$, $i=0,1,\ldots,j$. Sea $s \in L^{(j)}(\mathbb{S}/G_c)$ y consideremos la palabra $s\sigma \in L(G)$. Ahora $x=\$(x_o,s)$ está definida, y así $\sigma \in \Sigma_x{}^{o} \cup \Sigma_x{}^{1}$. Por lo tanto, $s\sigma \in K_3$ implica que $\sigma \in \Sigma_x{}^{1}$, $\$(x,\sigma)$ definida implica que $\phi(x)(\sigma)=1$ y $s\sigma \in L(\mathbb{S}/G_c)$. Por lo tanto, $L^{(j+1)}(\mathbb{S}/G_c)=K_3{}^{(j+1)}$.

Sólo falta probar que § es completo con respecto a $G_c$. Para esto, sea $s \in L(§/G_c)=K_3$, $s\sigma \in L(G)$ y $[(\phi_{\emptyset} \circ \$)(x_o,s)](\sigma)=1$. Si $s\sigma \notin K_3$, entonces debemos tener que $\sigma \in \Sigma^o{}_{\$(xo,s)}$. Pero esto implica que $[(\phi \circ \$)(x_o,s)](\sigma)=0$; una contradicción. Así $s\sigma \in K_3$ y en consecuencia § es completo.

## 6. SUPERVISORES PROPIOS

Las condiciones más estrictas para especificar el comportamiento controlado deben ser establecidas en los lenguajes $L_{ac}(§/G_c)$, $L_c(§/G_c)$ y $L(§/G_c)$ que describen al sistema $§/G_c$. Introducimos en esta sección una clase de supervisores, llamados supervisores propios, los cuales nos permitirán especificar el comportamiento controlado en una forma satisfactoria intuitivamente.

**DEFINICION 2.6.1** Un supervisor § será llamado no bloqueado si $\bar{L}_c(§/G_c)=\overline{L(§/G_c)\cap L_{ac}(G)}=L(§/G_c)$.

**DEFINICION 2.6.2** Un supervisor será llamado no rechazado si $\bar{L}_c(§/G_c)=\bar{L}_{ac}(§/G_c)$.

Por definición $\bar{L}_c(§/G_c) \subseteq L(§/G_c)$. Si § bloquea, es decir, falla para ser no bloqueado, entonces existe una palabra $s \in L(§/G_c)$ que no puede ser completada a una palabra st en

$L_c(\mathcal{S}/G_c)$, es decir, $s \notin L_c(\mathcal{S}/G_c)$. En este sentido el SEDC puede ser bloqueado para que siempre complete una "tarea". Esta situación indeseable es ilustrada por el supervisor (i) del ejemplo 2.3.1, acá la palabra $\alpha\tau \in L(\mathcal{S}/G_c)-\bar{L}_c(\mathcal{S}/G_c)$.

Si $\mathcal{S}$ rechaza, es decir, falla para ser no rechazado, entonces existe una palabra $s \in \bar{L}_c(\mathcal{S}/G_c)$, que puede ser completada para una "tarea" en $L_c(\mathcal{S}/G_c)$, pero nunca para una "tarea" que sea marcada. Sí $\bar{L}_c(\mathcal{S}/G_c)=\bar{L}_{ac}(\mathcal{S}/G_c)$, entonces $L_{ac}(\mathcal{S}/G_c) \subseteq L_c(\mathcal{S}/G_c) \subseteq \bar{L}_{ac}(\mathcal{S}/G_c)$. Así, para todo $s \in L_c(\mathcal{S}/G_c)$ existe un $t \in \Sigma^*$ tal que $st \in L_{ac}(\mathcal{S}/G_c)$. En consecuencia, $st \in L_c(\mathcal{S}/G_c)$.

El supervisor (ii) del ejemplo 2.3.1 rechaza solamente palabras de la forma $(\alpha\beta)^*$ que son marcadas, mientras $L_c(\mathcal{S}/G_c)=(\alpha\tau^*\beta)^*$ representa el conjunto de "tareas" que pueden ser ejecutadas.

**DEFINICION 2.6.3** Un supervisor $\mathcal{S}$ será llamado propio si $\mathcal{S}$ es completo, no bloqueado y no rechazado, es decir, $\mathcal{S}$ es completo y $\bar{L}_{ac}(\mathcal{S}/G_c)=\bar{L}_c(\mathcal{S}/G_c)=L(\mathcal{S}/G_c)$.

**TEOREMA 2.6.1** Sea $K \subseteq L_{ac}(G)$, con K no vacío.

(i) Existe un supervisor propio $\mathcal{S}$ tal que $L_{ac}(\mathcal{S}/G_c)=K$, si y sólo si, K es controlable. En este caso, $L_c(\mathcal{S}/G_c)=L_{ac}(G)\cap\bar{K}$.

(ii) Existe un supervisor propio $\mathcal{S}$ tal que $L_c(\mathcal{S}/G_c)=K$, si y sólo si, K es controlable y $L_{ac}(G)$-cerrado.

**PRUEBA** (i) K es controlable, si y sólo si, $\bar{K}$ es cerrado y controlable, si y sólo si,el triple $(K_1,K_2,K_3)=(K,\bar{K}\cap L_{ac}(G),\bar{K})$ satisface que $K_1 \subseteq K_2$, $K_2=K_3\cap L_{ac}(G)$ y $K_3$ es cerrado y controlable.(Ver proposición 2.5.1), si y sólo si, existe un supervisor completo $\mathcal{S}$ tal que

$(L_{ac}(\mathcal{S}/G_c),L_c(\mathcal{S}/G_c),L(\mathcal{S}/G_c))=(K,\bar{K}\cap(L_{ac}(G),\bar{K})$, si y sólo si,$\mathcal{S}$ es propio.

(ii) K es controlable y $(L_{ac}(G)$-cerrado, si y sólo si, el triple $(K_1,K_2,K_3)=(K,K,\bar{K})$ satisface las condiciones de la posición 2.5.1, si y sólo si, existe un supervisor completo $\mathcal{S}$ tal que $(L_{ac}(\mathcal{S}/G_c),L_c(\mathcal{S}/G_c),L(\mathcal{S}/G_c))=(K,K,\bar{K})$, si y sólo si, $\mathcal{S}$ es propio.

# 7. PROBLEMAS DE SINTESIS DE SUPERVISORES

Dados dos lenguajes $L_a, L_g \subseteq \Sigma^*$, con $\emptyset \neq L_a \subseteq L_g \subseteq L_{ac}(G)$. Interpretamos a $L_g$ como "el comportamiento legal", es decir, cada palabra de $L_g$ es una "tarea legal", y a $L_a$ como "el comportamiento aceptado minimal", es decir, el control del SEDC G en tal forma que un lenguaje generado que sea más pequeño que $L_a$ es considerado inadecuado.

Introducimos en esta sección el problema del marcaje supervisorio (PMS), es decir, el problema de construir un supervisor propio $\mathcal{S}$ para G tal que $L_a \subseteq L_{ac}(\mathcal{S}/G_c) \subseteq L_g$. Análogamente, definimos el problema de control supervisorio (PCS), es decir, el problema de construir un supervisorio propio $\mathcal{S}$ para G tal que $L_a \subseteq L_c(\mathcal{S}/G_c) \subseteq L_g$.
Si el PCS es soluble, entonces por la prueba del teorema 2.6.1 (ii) podemos acomodar que $L_{ac}(\mathcal{S}/G_c)=L_c(\mathcal{S}/G_c)$, así que el PMS es soluble también. Recíprocamente, considere el caso especial donde $L_g$ es $L_{ac}(G)$-cerrado, es decir, $L_g=\bar{L}_g \cap L_{ac}(G)$. Entonces $L_g \subseteq L_{ac}(G)$ tiene la propiedad que si una palabra $st \in L_g$ y $s \in L_{ac}(G)$, entonces $s \in L_g$. Ahora, si el PMS es soluble, entonces el lenguaje $L_{ac}(\mathcal{S}/G_c)$ satisface que $L_a \subseteq L_{ac}(\mathcal{S}/G_c) \subseteq L_g$;en consecuencia $L_a \subseteq L_{ac}(\mathcal{S}/G_c) \subseteq L_c(\mathcal{S}/G_c)$. Además, como $\mathcal{S}$ es propio, entonces $\bar{L}_{ac}(\mathcal{S}/G_c)=\bar{L}_c(\mathcal{S}/G_c)$, así

$L_c(\S/G_c) \subseteq \bar{L}_c(\S/G_c) = \bar{L}_{ac}(\S/G_c)$  $L_g$. Pero por definición $L_c\ (\S/G_c) \subsetneq L_{ac}(G)$, es decir, $L_c(\S/G_c) \subsetneq \bar{L}_g \cap L_{ac}(G) = L_g$. Por lo tanto, $L_a \subseteq L_c(\S/G_c) \subseteq L_g$ y en consecuencia el PCS es soluble también.

Cuando el PMS o el PCS es soluble, consideramos deseable que la solución sea minimal, en el sentido que $L_{ac}(\S/G_c)$ o $L_c(\S/G_c)$ sea considerado como un sublenguaje de $L_{ac}(G)$ tan grande como sea posible y sujeto a los constreñimientos para que sea un sublenguaje de $L_g$. Las soluciones son posibles, en principio, debido a una propiedad de los semi-retículos.

Sea $L \subseteq L(G)$ un sublenguaje arbitrario de $L(G)$ y sean $\varsigma_G(L) = \{K/K \subseteq L, K \text{ controlable}\}$, $F_G(L) = \{K/K \subseteq L,\ K = \bar{K} \cap L_{ac}(G)\}$.

**PROPOSICION 2.7.1** $\varsigma_G(L)$ y $F_G(L)$ son clases no vacías de lenguajes, cerradas bajo la unión arbitraria.

**PRUEBA** Sea $\emptyset$ el lenguaje vacío de $\Sigma^*$. Claramente $\emptyset \in \varsigma_G(L)$ y $\emptyset \in F_G(L)$, así $\varsigma_G(L)$ y $F_G(L)$ son clases no vacías.

Si $K_\alpha \in \varsigma_G(L)$, para $\alpha$ en algún conjunto de índices $A$, entonces $\bar{K}_\alpha \Sigma_u \cap L(G) \subseteq \bar{K}_\alpha$, $\alpha \in A$. Por la definición de clausura se tiene que $\bigcup_{\alpha\in A} \bar{K}_\alpha = \overline{\bigcup_{\alpha\in A} K_\alpha}$. Por lo tanto, $(\overline{\bigcup_{\alpha\in A} K_\alpha})\Sigma_u \cap L(G)$

$$=(\bigcup_{\alpha\in A} \bar{K}_\alpha)\Sigma_u \cap L(G) = \bigcup_{\alpha\in A} (\bar{K}_\alpha \Sigma_u) \cap L(G) = \bigcup_{\alpha\in A} (K_\alpha \Sigma_u \cap L(G)) \subseteq \bigcup_{\alpha\in A} \bar{K}_\alpha =$$

$= \overline{\bigcup_{\alpha\in A} K_\alpha}$, así $\bigcup_{\alpha\in A} K_\alpha \in \mathcal{C}_G(L)$. La prueba para $F_G(L)$ es análoga.

Por la proposición 2.7.1, $\mathcal{C}_G(L)$ y $F_G(L)$ contienen un único supremo con respecto a la inclusión, los cuales denotamos respectivamente por $SUP\mathcal{C}_G(L)$ y $SUP\ F_G(L)$. En efecto, $\mathcal{C}_G(L)$ y $F_G(L)$ son sub semi-retículos completos del semi-retículo de todos los sublenguajes de L, ordenado parcialmente por la inclusión.

Basados en el teorema 2.6.1 y la proposición 2.7.1 obtenemos nuestro primer gran resultado.

**TEOREMA** 2.7.1 (i) El PMS es soluble, si y sólo si,

$L_a \quad SUP\mathcal{C}_G(L)$.

(ii) El PCS es soluble, si y sólo si,

$L_a \subseteq SUP\{\mathcal{C}_G(L_s) \cap F_G(L_s)\}$.

En cada caso del teorema 2.7.1, el supervisor correspondiente es restrictivo minimalmente.

## 8. PROYECCIONES DE SUPERVISORES

Sean $\mathcal{S}=(S,\phi)$ y $\mathcal{S}'=(S',\phi')$ supervisores para un generador G, con $S=(X,\Sigma,\$,X_o,F^1)$, $\phi:X \longrightarrow \{0,1\}^\Sigma$ y $S'=(X',\Sigma,\$',X_o',F^{1\prime})$.

**DEFINICION 2.8.1** Una proyección $\pi$ de $\mathcal{S}$ en $\mathcal{S}'$, denotado por $\pi:\mathcal{S} \longrightarrow \mathcal{S}'$, es una función $\pi$: $X \longrightarrow X'$ tal que

(i) $\pi$ es sobreyectiva,

(ii) $\pi(x_0)=x_0'$ y $F^1=\pi^{-1}(F^{1\prime})$,

(iii) $[\delta' o(\pi x Id_\Sigma)](x,\sigma)=(\pi o \delta)(x,\sigma)$, para todo $(x,\sigma) \in X x \Sigma$ tales que $\delta(x,\sigma)$ está definida,

(iv) $\phi' o \pi = \phi$.

Bajo las condiciones de la definición 2.8.1 nos referiremos a S' como el cociente de S bajo $\pi$. Esta situación es descrita en la figura 7.

$$
\begin{tikzcd}
X \times \Sigma \arrow[r, "\delta"] \arrow[d, "\pi"', "Id_\Sigma"] & X \arrow[d, "\pi"] \\
X' \times \Sigma \arrow[r, "\delta'"] & X'
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[dr, dashed, "\phi"] \arrow[d, "\pi"'] & \\
X' \arrow[r, dashed, "\phi'"'] & \{0,1\}^\Sigma
\end{tikzcd}
$$

FIGURA 7

Extendemos $\pi x Id_\Sigma$ a una función $\pi x Id_\Sigma : X x \Sigma^* \longrightarrow X' x \Sigma^*$ de manera natural.

**PROPOSICION 2.8.1** Sea $\mathcal{S}$ un supervisor completo con respecto a un generador G, y sea $\pi:\mathcal{S} \longrightarrow \mathcal{S}'$ una proyección. Entonces

(i) $\pi$ es única,

(ii) $L_{ac}(\mathcal{S}/G_c) = L_{ac}(\mathcal{S}'/G_c$, $L_c(\mathcal{S}/G_c = L_c(\mathcal{S}'/G_c)$ y $L(\mathcal{S}/G_c = L(\mathcal{S}'/G_c)$,

(iii) $\mathcal{S}'$es completo con respecto a G,

(iv) $\mathcal{S}$ es no bloqueado (respectivamente, no rechazado y propio), si y sólo si, $\mathcal{S}'$ es no bloqueado (respectivamente, no rechazado y propio).

**PRUEBA** Pongamos $\mathcal{S}=(S,\phi)$,con $S=(X,\Sigma,\delta,X_o,F^1)$, y $\mathcal{S}'=(S',\phi')$, con $S'=(X',\Sigma,\delta',X_o',F^{1\prime})$. Recordamos que, por definición, S y S' son accesibles.

(i) tenemos que $\pi(x_o)=x_o'$. Si $x \in X$, entonces, por ser S accesible, existe $s \in \Sigma^*$ tal que $x=\delta(x_o,s)$, asi $\pi(x)=(\pi o \delta)(x_o,s)=[\delta' o(\pi x Id_\Sigma)](x_o,s)=\delta'(x_o',s)$. Por lo tanto $\pi$ es única.

(ii) pongamos $L=L(\mathcal{S}/G_c)$ y $L'=L(\mathcal{S}'/G_c)$. Si $s \in L$ con $|s|=1$, entonces $\phi(x_o)(s)=1=\phi(x_o)(\sigma)$, y

$(\mathcal{E}_c x \delta)(q_o,x_o,\sigma)=(\mathcal{E}_c(q_o,\sigma,\phi(x_o)),\delta(x_o,\sigma))=(\mathcal{E}(q_o,\sigma),\delta(x_o,\sigma))$

está definida. Como $\phi'(x_o')(\sigma)=1$ y

$(\mathcal{E}_c x \delta')(q_o,x_o',\sigma)=(\mathcal{E}(q_o,\sigma),\delta'(x_o',\sigma))=(\mathcal{E}(q_o,\sigma),(\pi o \delta)(x_o,\sigma))$

$=[(\pi x Id_Q)o(\mathcal{E}_c x \delta)](q_o,x_o,\sigma)$está definida, entonces $L^{(1)} \subseteq L'^{1}$.

Por inducción sobre la longitud de $s$ es fácil probar que $L^{(j)} \subseteq L'^{(j)}$, $j=0,1,\ldots$ por lo tanto $L$ $L'$ para probar la otra inclusión supongamos que $(\mathcal{S}_c \times \$')(q_0, x_0', \sigma)$ está definida, entonces $\sigma \in L'$ y $\phi'(X_0')(\sigma)=1$. Así, $\phi(x_0)(\sigma)=(\phi' o \pi(x_0))(\sigma)=\phi'(x_0')(\sigma)=1$. Además, $\mathcal{S}(q_0,\sigma)=\mathcal{S}_c(q_0,\sigma,\phi'(x_0'))$ está definida;en consecuencia $\$(x_0,\sigma)$ está definida. Por lo tanto, $(\mathcal{S}_c \times \$)(q_0,x_0,\sigma)=(\mathcal{S}(q_0,\sigma),\$(x_0,\sigma))$ está definida, y $L'^{(1)} \subseteq L^{(1)}$. Supongamos que $L'^{(i)} \subseteq L^{(i)}$ $(i=0,1,\ldots,j)$. Sea $s \in L'^{(j)}$ y consideremos $s\sigma \in L'^{(j+1)}$. Entonces $s \in L^{(j)}$,así $x=(\mathcal{S}_c \times \$)(q_0,x_0,s)$y $x'=(\mathcal{S}_c \times \$')(q_0,x_0',s)$ están definidas, y $\pi(x)=x'$. Aplicado este mismo argumento a $(x,x')$ en lugar de $(x_0,x_0')$ obtenemos que $s\sigma \in L^{(j+1)}$. Por lo tanto $L' \subseteq L$ y en consecuencia $L'=L$. Luego, $L_c(\$/G_c)=L(\$/G_c)\cap L_{ac}(G)=L(\$'/G_c)\cap L_{ac}(G)=L_c(\$'/G_c)$.

Finalmente, $s \in L_{ac}(\$/G_c)$,si y sólo si, $(\mathcal{S}_c \times \$)(q_0,x_0,s)$ está en $F \times F^1$, si y sólo si, $\$(x_0,s) \in F^1$ y $s \in L_c(\$/G_c)$, si y sólo si,$\$(x_0,s) \in \pi^{-1}(F'^1)$ y $s \in L_c(\$/G_c)$, si y sólo si, $(\pi o \$)(x_0,s) \in F'^{1}_{1}$ y $s \in L_c(\$'/G_c)$, si y sólo si, $\$'(x_0',s) \in F'^{1}_{1}$ y $s \in L_c(\$'/G_c)$, si y sólo si, $(\mathcal{S}_c \times \$')(q_0,x_0',s) \in F \times F'^{1}_{1}$,si y sólo si, $s \in L_{ac}(\$'/G_c)$.

(iii) para probar que $\$'$ es completo con respecto a G, sea $s \in L(\$'/G_c)$, $s\sigma \in L(G)$ y $(\phi' o \$')(x'_0,s)(\sigma)=1$. Entonces, por (ii), $s \in L(\$/G_c)$, y

$(\emptyset \circ \$)(x_0, s)(\sigma) = (\emptyset' \circ \pi \circ \$)(x_0, s)(\sigma) = (\emptyset' \circ \$')(x'_0, s)(\sigma)$

Como $\mathcal{S}$ es completo, entonces $\$(x_0, s\sigma)$ está definida. Además, como $\pi$ es una proyección, se tiene que $\$'(x'_0, s\sigma)$ está definida. Por lo tanto $\mathcal{S}'$ es completo.

(iv) la prueba es inmediata a partir de (ii) y (iii).

## 9. SUPERVISOR EFICIENTE

En esta sección daremos una caracterización abstracta de un supervisor "construído eficientemente", para un lenguaje no vacío $K \subseteq \Sigma^*$ que tenga la propiedad de ser $L_{ac}(G)$-cerrado y controlable. La existencia es dada por el teorema 2.6.1. Más aún si $S=(\mathcal{S}, \emptyset)$ es el supervisor propio para K, construido como en la prueba del teorema 2.6.1, entonces $K = L_{ac}(S/G_c) = L_c(S/G_c)$ y $K = L(S/G_c)$. Por lo tanto, como en la construcción usada en la prueba de la proposición 2.5.1, podemos arreglar que, para una palabra $s \in K$, el estado x rechazado por $\mathcal{S}$ sea tal que, para todo $\sigma \in \Sigma_c$,

$$\emptyset(x)(\sigma) = \begin{cases} 0, \text{si } s\sigma \notin \bar{K}, \\ 1, \text{si } s\sigma \in \bar{K} \end{cases}$$

En base a esta definición, introducimos una relación de equivalencia sobre $\Sigma^*$ de la manera siguiente: diremos que $s, s' \in \Sigma^*$ son control-equivalente, escrito $s \sim s'$, si para todo $\sigma \in \Sigma_c$ se tiene que $s\sigma \in \bar{K}$, si y sólo si, $s'\sigma \in \bar{K}$.

Recordando de la teoria de autómatas que una relación de equivalencia e sobre $\Sigma^*$ es invariante a derecha si, para todo $s,s' \in \Sigma^*$ y $s \equiv s' \mathrm{mod}(e)$ se tiene que, para todo $t \in \Sigma^*$, $st \equiv s' \mathrm{mod}(e)$. Sea $\{e_\alpha \ / \ \alpha \in A\}$ una familia arbitraria no vacia de relaciones de equivalencia sobre $\Sigma^*$. Definimos una relación de equivalencia sobre $\Sigma^*$, denotada $e = \mathrm{SUP}\{ e_\alpha \ / \alpha \in A \}$, como sigue: para $s,s' \in \Sigma^*$, $s \equiv s' \mathrm{mod}(e)$, si existen $K \geq 1$ (K-entero), $\alpha_0, \alpha_1, \ldots, \alpha_k \in A$ y $s_1, s_2, \ldots, s_k \in \Sigma^*$ tal que $s \equiv s_1 \mathrm{mod}(e_{\alpha_0})$, $s_1 \equiv s_2 \mathrm{mod}(e_{\alpha_1}), \ldots, s_{k-1} \equiv s_k \mathrm{mod}(e_{\alpha_{k-1}})$, $s_k \equiv s' \mathrm{mod}(e_{\alpha_k})$. Es fácil probar que si, en particular, las $e_\alpha$ son congruentes a derecha, entonces e también lo es. El orden de la teoria de reticulos para las relaciones de equivalencia sobre $\Sigma^*$ está definido como sigue: $e_1 \leq e_2$ si, para todo $s,s' \in \Sigma^*$. $s \equiv s' \mathrm{mod}(e_1)$ implica que $s \equiv s' \mathrm{mod}(e_2)$; $e_1$ es llamada más fina que $e_2$, en este caso.Por lo tanto, $e = \mathrm{SUP}\{e_\alpha \ / \ \alpha \in A\}$ es la relación de equivalencia más fina sobre $\Sigma^*$ que es menos fina que cada $e_\alpha$, $\alpha \in A$.

En general, la relación de equivalencia control-equivalente $\sim$ no es congruente a derecha. Por lo tanto, si definimos $s \equiv s' \mathrm{mod}(0)$ si $s = s'$, entonces trivialmente $0$ es congruente a derecha y $0 \leq \sim$; de acá que la relación de equivalencia $\approx = \mathrm{SUP}\{e \ / \ e$ es congruente a derecha sobre $\Sigma^*$, $e \leq \sim\}$ existe, y es la congruencia a derecha que es la menos fina sobre $\Sigma^*$ pero más fina que $\sim$.

Para $s \in \Sigma^*$, sea $[s]$ la clase de equivalencia de s módulo $\approx$. Definimos el autómata $\bar{S}=(\bar{X},\Sigma,\bar{\$},\bar{x}_0,\bar{X})$, donde $\bar{X}=\{[s]/s \in \bar{K}\}$, $\bar{x}_0=[\theta]$ y $\bar{\$}(\bar{x},\sigma)=\bar{x}'$ si $\bar{x}=[s]$, $s \in \bar{K}$, $s\sigma \in \bar{K}$ y $[s\sigma]=\bar{x}'$; en otro caso $\bar{\$}$ no está definida. Definimos $\bar{\phi}:\bar{X} \longrightarrow \{0,1\}^{\Sigma}$ acordando que $\bar{\phi}(\bar{x})(\sigma)=0$ si $\sigma \in \Sigma_c$, existe $s \in \bar{K}$ con $\bar{x}=[s]$ y $s\sigma \notin K$; en otro caso $\bar{\phi}(\bar{x})(\sigma)=1$. Ahora podemos definir el supervisor eficiente $\bar{\mathbb{S}}=(\bar{S},\bar{\phi})$. Evidentemente $\bar{\mathbb{S}}$ es "eficiente" en el sentido de que cualquier autómata $\hat{S}$, que soporte la acción de control

$$\phi(x)(\sigma)=\begin{cases} 0, & \text{si } s\sigma \notin \bar{K} \\ 1, & \text{si } s\sigma \in \bar{K} \end{cases}$$

tal como fué definida al comienzo de esta sección, debe tener una estructura de estado (congruencia a derecha sobre $\Sigma^*$) al menos tan fina como la de $\bar{S}$. Es fácil ver que $\bar{\mathbb{S}}$ es completo, ya que $s \in \bar{K}$, $[s]=\bar{x}$, $\bar{\phi}(\bar{x})(\sigma)=1$ y $s\sigma \in L(G)$ implican que $s\sigma \in \bar{K}$ y por lo tanto $\bar{\$}(\bar{x},\sigma)$ está definida. Procediendo como en la prueba de la proposición 2.8.1 se tiene que $L(\bar{\mathbb{S}}/G_c)=\bar{K}=L(\mathbb{S}/G_c)$. Finalmente como $X=F^1$, entonces $L_{ac}(\bar{\mathbb{S}}/G_c)=L(\bar{\mathbb{S}}/G_c)\cap L_{ac}(G)=\bar{K}\cap L_{ac}(G)=K$ (recuerde que K es $L_{ac}(G)$-cerrado); así $K=L_{ac}(\bar{\mathbb{S}}/G_c)=L_c(\bar{\mathbb{S}}/G_c)$ y $\bar{\mathbb{S}}$ es propio. Así $\bar{\mathbb{S}}$ ejecuta la misma acción de control sobre G, tal como el supervisor con el cual comenzamos.

Definimos sobre $\Sigma^*$ una relación de equivalencia, denotada por $\equiv \bmod(K)$, llamada $\bar{K}$-equivalencia, de la manera siguiente: para $s,s' \in \Sigma^*$, $s\equiv s' \bmod(\bar{K})$ si para todo $t \in \Sigma^*$, $st \in \bar{K}$, si y sólo si, $s't \in \overline{K}$. Claramente $\equiv \bmod(\overline{K})$ es una congruencia a derecha. Además si $s \equiv s' \bmod(\overline{K})$, entonces, en particular, para todo $\sigma \in \Sigma_c$, $s\sigma \in \overline{K}$, si y sólo si $s'\sigma \in \overline{K}$. Así, $\equiv \bmod(\overline{K}) \leq \approx$, es decir, para todo $s,s' \in \Sigma^*$, $s \equiv s' \bmod(\overline{K})$ implica que $s \approx s'$. En particular, si $s,s' \in \overline{K}$ y $s \equiv s' \bmod(\overline{K})$, entonces $\bar{\xi}(\bar{x}_0,s)=\bar{\xi}(\bar{x}_0,s')$. Nos referiremos a esta última propiedad diciendo que el autómata $\bar{S}$ es $\overline{K}$-reducido. Para su construcción, $\bar{S}$ es también $\overline{K}$-trim, es decir, para todo $\bar{x} \in \bar{X}$ existe $s \in \overline{K}$ tal que $\bar{\xi}(\bar{x}_0,s)=\bar{x}$.

En la próxima sección se muestra que cualquier supervisor con estas dos propiedades puede ser proyectado desde un supervisor basado en un reconocedor para $\overline{K}$.

## 10. TEOREMA DE LA ESTRUCTURA COCIENTE

Estamos preparados para probar el segundo gran teorema de este capítulo. Estableceremos que "todo supervisor construído eficientemente" es un cociente del comportamiento deseado de $\mathbb{S}/G_c$.

**TEOREMA 2.10.1** Sea $\mathbb{S}=(S,\phi)$ un supervisor completo para G. Pongamos $K_1=L_{ac}(\mathbb{S}/G_c)$, $K_3=L(\mathbb{S}/G_c)$ y supongamos que $S$ es $K_3$-reducido y $K_3$-trim. Sea $S^o=(X^o,\Sigma,\$^o,X^o{}_o,X^o_m)$ un reconocedor trim para $K_3$. Entonces, existe un subconjunto $F^o \subseteq X^o$ y una función de retroalimentación de estado $\phi^o:X^o \longrightarrow \{0,1\}^\Sigma$ con las siguientes propiedades:

(i) El supervisor $\mathbb{S}^o=(S^o,\phi^o)$, con $S^o=(X^o,\Sigma,\$^o,x^o{}_o,F^o)$ es un supervisor completo para G tal que $L_{ac}(\mathbb{S}^o/G_c)=K_1$ y $L(\mathbb{S}^o,/G_c)=K_3$.

(ii) Existe una proyección $\pi:\mathbb{S}^o \longrightarrow \mathbb{S}$.

(iii) Si $\mathbb{S}$ es propio, entonces $\mathbb{S}^o$ es propio.

**PRUEBA** Pongamos $S=(X,\Sigma,\$,X_o,F)$. Sea $x^o \in X^o$; como $S^o$ es trim, entonces existe $s \in K_3$ tal que $\$^o(x_o{}^o,s)=x^o$. Sea $\$(x_o,s)=x \in X$ y definamos $\pi:X^o\longrightarrow X$ acordando que $\pi(x^o)=x$. Afirmamos que $\pi$ está bien definida. En efecto, sea $t \in K_3$, $\$^o(x^o{}_o,t)=x^o$, y sea $\$(x_o,t)=y \in X$. Ya que $S^o$ es un reconocedor para $K_3$ y $\$^o(x^o{}_o,s)=\$^o(x^o{}_o,t)$, tenemos que $s\equiv t \bmod(K_3)$. Ahora, como S es $K_3$-reducido, entonces $\$(x_o,s)=\$(x_o,t)$, es decir, $x=y$.

Afirmamos que $\pi$ es una proyección. En efecto, sea $x \in X$. Como S es $K_3$-trim, entonces existe un $s \in K_3$ tal que $\$(x_o,s)=x$. Sea $\$^o(x^o{}_o,s)=x^o$, entonces $\pi(x^o)=x$; así $\pi$ es sobreyectiva. Ahora, sea $\$^o(x^o,\sigma)=y^o$. Tenemos que $x^o=\$^o(x^o{}_o)$ para algún $s \in K_3$; así $s\sigma \in K_3$. Ya que $K_3=L(§/G_c)$, se sigue que $\$(\pi(x^o),\sigma)$ está definida y coincide con $(\pi o \$^o)(x^o,\sigma)$.Para establecer que $\pi$ es una proyección sólo resta definir $F^o=\pi^{-1}(F)$ junto con $\emptyset^o= \emptyset$ o $\pi$.

Queremos probar que $L\ (§^o/G_c)=K_3$. Usando el argumento de la prueba de la proposición 2.5.1 tenemos que es suficiente mostrar que: (i) Para todo $\sigma$ y para todo $x^o \in X^o$, con $\$^o(x^o{}_o,s)=x^o$, $s\sigma \in L(G)$, $s\sigma \notin K_3$, para algún s implica que $\emptyset^o(x^o)(\sigma)=0$.

(ii) Para todo $\sigma$ y para todo $x^o \in X^o$, con $\$^o(x^o{}_o,s)=x^o$, $s\sigma \in K_3$, para algún s implica que $\emptyset^o(x^o)(\sigma)=1$.

Para la prueba de (i), sea $\$^o(x^o{}_o,s)= x^o$, $s\sigma \in L(G)$, $s\sigma \in K_3$. Claramente $s \in K_3$. Si $\$(x_o,s)=x$, entonces $\emptyset(x)(\sigma)=0$. Por lo tanto,$\emptyset^o(x^o)(\sigma)=(\emptyset$ o $\pi)(x^o)(\sigma)=\emptyset(x)(\sigma)=0$. La prueba de (ii) es análoga.

Para probar que $\S^o$ es completo con respecto a $G_c$ note que $s \in L(\S^o/G_c)$, $s\sigma \in L(G)$ y $(\phi^o o \$^o)(x^o_o,s)(\sigma)=1$, sí y sólo si, $s \in L(\S/G_c)$, $s\sigma \in L(G)$ y $(\phi o \$)(x_o,s)(\sigma)=1$. Pero como $\S$ es completo, entonces $s\sigma \in L(\S/G_c)=L(\S^o/G_c)$. Finalmente, sea $s \in K_3$. Entonces, $s \in K_1$, sí y sólo si, $\$(x_o,s) \in F^1$ y $\delta(q_o,s) \in F$, sí y sólo si, $(\pi o \$^o)(x^o_o,s) \in F^1$ y $\delta(q_o,s) \in F$, sí y sólo si, $\$^o(x^o_o,s) \in F^o$ y $\delta(q_o,s) \in F$, sí y sólo si, $s \in L_{ac}(\S^o/G_c)$. Así $L_{ac}(\S^o/G_c)=L_{ac}(\S/G_c)$. En particular, si $\S$ es propio, entonces $\S^o$ es propio.

**EJEMPLO 2.10.1** En un sistema de manufactura consideremos dos máquinas $M_1, M_2$ conectadas en serie y separadas por un almacén A (figura 8). Cada máquina $M_i$ es modelada como un SEDC sobre el alfabeto $\{\alpha_i, \beta_i, \tau_i, \mu_i\}$ y con variables de control $u_i$ y $v_i$ (figura 9). Los estados de $M_i$ son $A_i$ (apagada), $F_i$ (funcionando) y $D_i$ (dañada).

$$\longrightarrow M_1 \longrightarrow A \longrightarrow M_2 \longrightarrow$$

FIGURA 8

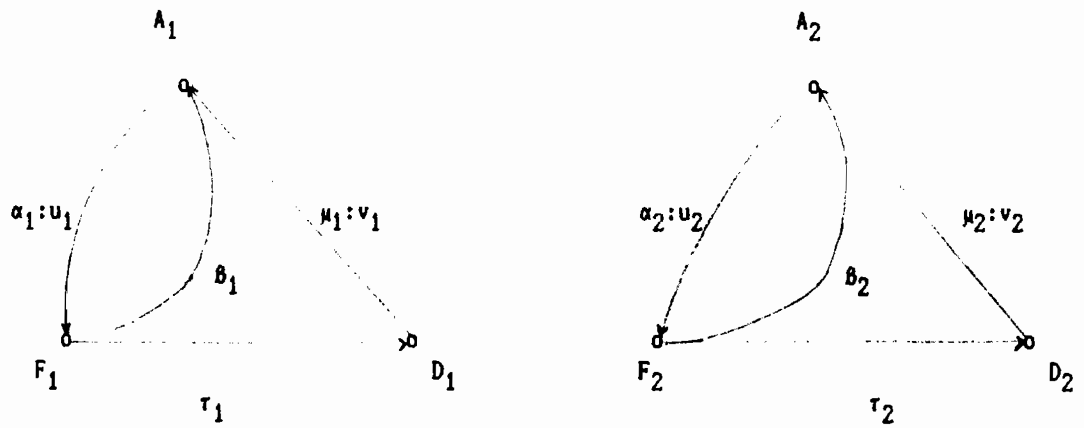



FIGURA 9

El control $u_i$ habilita o inhabilita la transición de $A_i$ a $F_i$ ($u_i=1$ permite que $M_i$ acepte una pieza de trabajo); mientras que $v_i$ habilita o inhabilita la transición de $D_i$ a $A_i$ ($v_i=1$ significa, cuando $M_i$ está en el estado $D_i$, que $M_i$ está en reparación). El almacén A tiene dos estados, estos son, vacío (V) y no vacío (NV). A no es un SEDC pero puede ser visto como un autómata dirigido por $M_1$ y $M_2$ (figura 10). El sistema opera de la manera siguiente:la máquina $M_1$ toma una pieza de trabajo (evento $\alpha_1$). Si la máquina completa su proceso de tratamiento correctamente, la pieza es almacenada (evento $\beta_1$). Ahora, si se rompe el proceso, la máquina descarta la pieza de trabajo (evento $\tau_1$), pero en este caso la máquina es reparada (evento $\mu_1$). La máquina $M_2$ opera análogamente, pero toma sus piezas de trabajo del almacén A, en caso de haber una.

Ahora, nuestro problema consiste en manipular los controles de tal manera que sean satisfechos los cuatro requerimientos siguientes:

(i) $M_1$ ejecuta $\alpha_1$ solamente si A está en V.

(ii) $M_2$ ejecuta $\alpha_2$ solamente si A está en NV.

(iii) $M_1$ no puede ejecutar $\alpha_1$ mientras $M_2$ esté en $D_2$ .

(iv) Si $M_1$ está en $D_1$ y $M_2$ está en $D_2$, entonces $v_1=0$.

La condición (iv) significa que si las dos máquinas están dañadas simultáneamente, entonces $M_2$ debe ser reparada antes que $M_1$. No formalizaremos los cuatro requerimientos, antes mencionados, ni daremos los detalles de como el lenguaje legal $L_g$ es derivado de éstos, pero daremos el resultado. El lenguaje $L_g$ para nuestro sistema es dado en la figura 11. El reconocedor correspondiente define un supervisor $\S^o$ tal que $L_g=L(\S/G_c)$; los patrones de control son tabulados en la tabla 1. Puede verificarse que $\S^o$ admite el cociente dado en la figura 12; la proyección requerida es tabulada en la tabla 1. El cociente representa una reducción de doce estados a solamente seis estados.

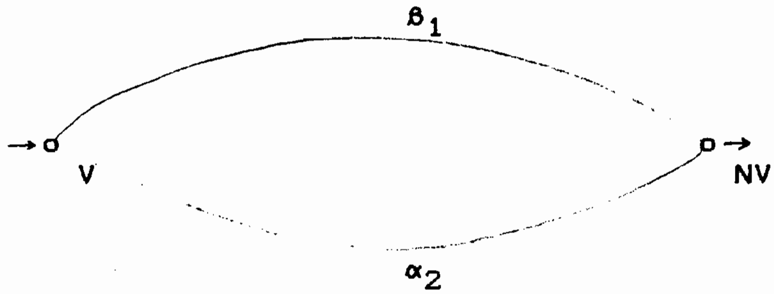



FIGURA 10.

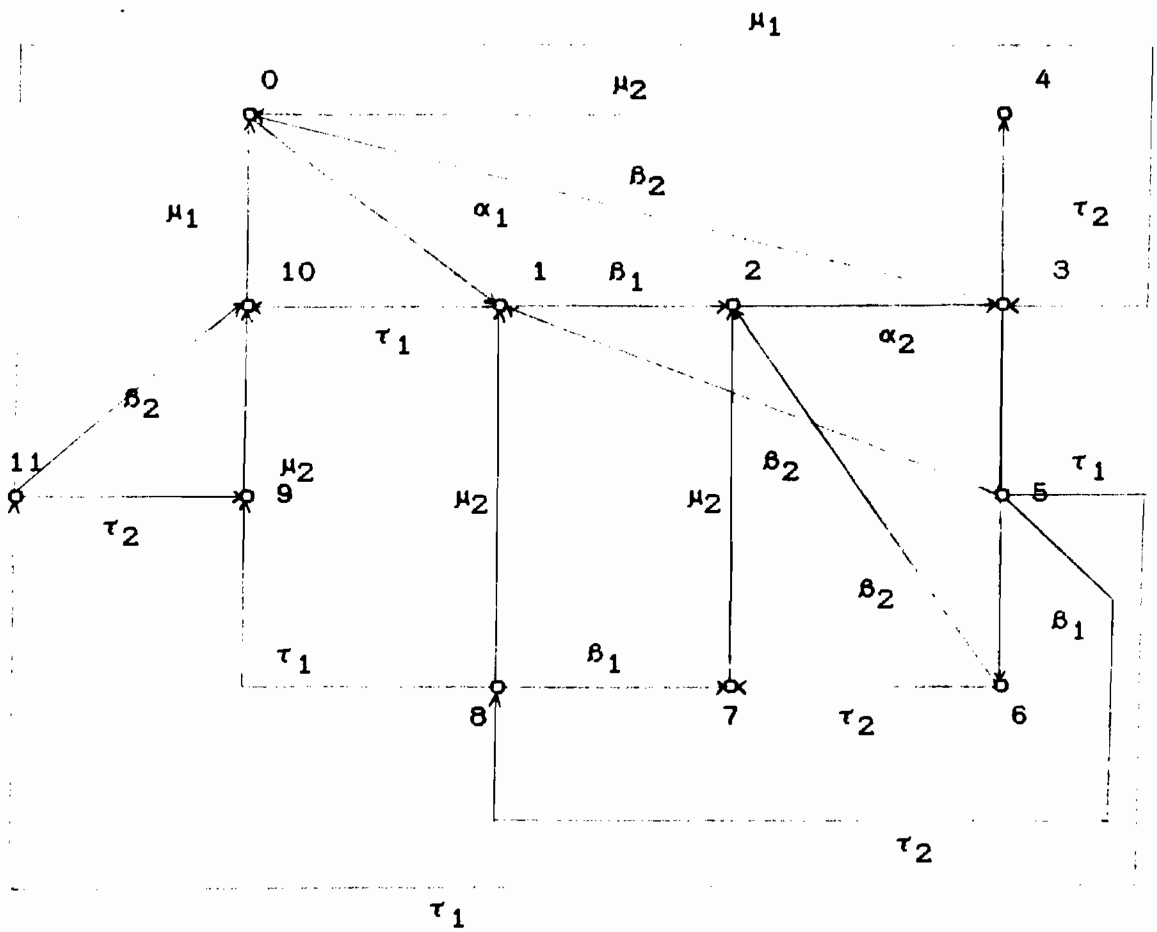



FIGURA 11

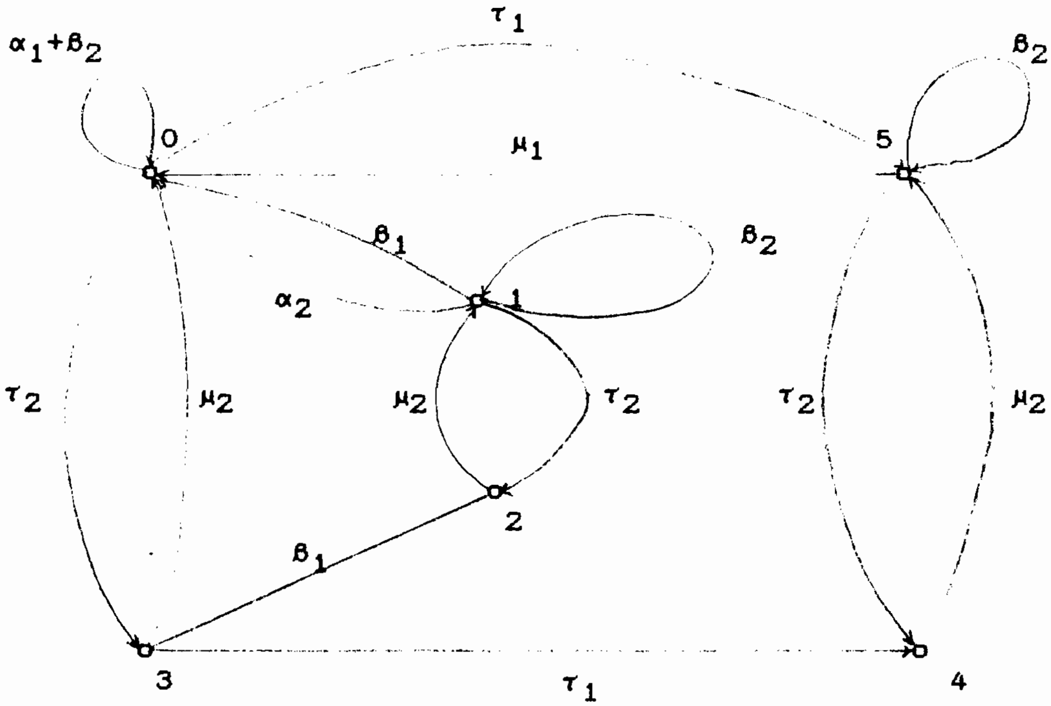



FIGURA 12

| $x^o$ | $x$ | $u_1$ | $v_1$ | $u_2$ | $v_2$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | – | 0 | – |
| 1 | 0 | 1* | – | 0 | – |
| 2 | 1 | 0 | – | 1 | – |
| 3 | 0 | 1 | – | 0 | – |
| 4 | 3 | 0 | – | 0 | 1 |
| 5 | 0 | 1* | – | 0 | – |
| 6 | 1 | 0 | – | 1* | – |
| 7 | 2 | 0 | – | – | 1 |
| 8 | 3 | 0 | – | 0 | 1 |
| 9 | 4 | 0 | 0 | 0 | 1 |
| 10 | 5 | – | 1 | 0 | – |
| 11 | 5 | – | 1 | 0 | – |

TABLA 1

En la tabla 1 el símbolo (–) es usado para significar una asignación arbitraria.

CAPITULO 3

# CONTROL MINIMAL Y OBSERVACION MINIMAL DE SISTEMAS DE EVENTOS DISCRETOS

Una consecuencia inmediata de la proposición 2.5.1 es que, dado un lenguaje cerrado $K \subseteq L(G)$, si todos los eventos en $\Sigma$ pueden ser controlados, entonces K es controlable y $K=L(\S/G_c)$. En realidad existen muchos subconjuntos $\Sigma_c$ de $\Sigma$ que garantizan que K es controlable. En este capítulo nuestro interés está centrado en encontrar el subconjunto minimal de eventos controlados, para el cual K es controlable y el subconjunto minimal de eventos observables, para el cual K es observable. Damos algunas condiciones de necesidad y suficiencia que garantizan la existencia de la observación minimal (el control minimal siempre existe), y algunos algoritmos eficientes para calcular el control minimal y la observación minimal.

## 1. EL CONTROL MINIMAL

**DEFINICION 3.1.1** Sea $K \subseteq \Sigma^*$ cerrado y sea $L(G)$ el lenguaje del sistema no controlado. Diremos que $\Sigma_c \in 2^\Sigma$ es K controlable, si $K(\Sigma-\Sigma_c)\cap L(G) \subseteq K$.

Sea $C(\Sigma)=\{\Omega/\Omega \in 2^\Sigma, K\Omega\cap L(G) \subseteq K\}$.

**PROPOSICION 3.1.1** $(C(\Sigma),\cup,\cap)$ es un retículo completo.

**PRUEBA** Seán $\Omega_1$, $\Omega_2 \in C(\Sigma)$, entonces $K\Omega_1\cap L(G) \subseteq K$ y $K\Omega_2\cap L(G) \subseteq K$. Por lo tanto,

$K(\Omega_1\cup\Omega_2)\cap L(G)=(K\Omega_1\cup K\Omega_2)\cap L(G)=(K\Omega_1\cap L(G))\cup(K\Omega_2\cap L(G))\subseteq K$.

Luego, $\Omega_1\cup\Omega_2 \in C(\Sigma)$. Además,

$K(\Omega_1\cap\Omega_2)\cap L(G) \subseteq (K\Omega_1\cap K\Omega_2)\cap L(G)=(K(\Omega_1\cap L(G))\cap(K\Omega_2\cap L(G))$ $K$; así $\Omega_1\cap\Omega_2) \in C(\Sigma)$. En consecuencia, $(C(\Sigma),\cup,\cap)$ es un retículo. Ahora, ya que $C(\Sigma)$ es finito se tiene que $(C(\Sigma),\cup,\cap)$ es un retículo completo.

Como $(C(\Sigma),\cup,\cap)$ es un retículo completo, entonces $SUPC(\Sigma)$ existe.

**TEOREMA 3.1.1** El subconjunto de eventos K-controlable minimal $\Sigma^m{}_c$ existe, y $\Sigma^m{}_c= \Sigma-SUPC(\Sigma)$.

**PRUEBA** Claramente $\Sigma^m{}_c$ es K-controlable, pues $K(\Sigma-\Sigma^m{}_c)\cap L(G)= KSUPC(\Sigma)\cap L(G)\subseteq K$. Para la minimalidad, sea

$\Sigma^{o} \subsetneq \Sigma^{m}_{c}$ K-controlable, entonces $K(\Sigma-\Sigma^{o}) \cap L(G) \subseteq K$; así $\Sigma-\Sigma^{O} \in C(\Sigma)$, de donde $\Sigma-\Sigma^{O} \subsetneq SUPC(\Sigma)=\Sigma-\Sigma^{m}_{c}$. Por lo tanto, $\Sigma^{m}_{c} \subsetneq \Sigma^{o}$. Luego, $\Sigma^{o}=\Sigma^{m}_{c}$. En consecuencia $\Sigma^{m}_{c}$ es minimal.

El teorema 3.1.1 nos dice que para calcular $\Sigma^{m}_{c}$, primeramente calculamos SUPC($\Sigma$) y luego hacemos la diferencia.

**PROPOSICION 3.1.2** Dado $\sigma \in \Sigma$. Entonces, $\sigma \in SUPC(\Sigma)$, si y sólo si, $K\sigma \cap L(G) \subsetneq K$.

**PRUEBA** Supongamos que $\sigma \in SUPC(\Sigma)$, entonces $K\sigma \cap L(G) \subsetneq KSUPC(\Sigma) \cap L(G) \subsetneq K$.

Recíprocamente, supongamos que $K\sigma \cap L(G) \subsetneq K$, entonces $\{\sigma\} \subsetneq SUPC(\Sigma)$. Por lo tanto, $\sigma \in SUPC(\Sigma)$.

## 2. LA OBSERVACION MINIMAL

**DEFINICION 3.2.1** Sea $K \subseteq \Sigma^{*}$ cerrado y sea L(G) el lenguaje del sistema no controlado. Diremos que $\Sigma_{o} \in 2^{\Sigma}$ es K-observable, si $KerP \leq act(K)$, donde $P:\Sigma^{*} \longrightarrow \Sigma^{*}_{o}$ es inducida por $\Sigma_{o}$ como sigue: $P(\theta)=\theta$, $P(\sigma)=\theta$ si $\sigma \in \Sigma - \Sigma_{o}$, $P(\sigma)=\sigma$ si $\sigma \in \Sigma_{o}$ y $P(s\sigma)=P(s)P(\sigma)$.

Sea $O(\Sigma)=\{\Gamma/\Gamma\in 2^{\Sigma}, KerP_{\Gamma} \leq act(K)\}$, donde $P_{\Gamma}$ es inducida por $\Gamma$. En algunos casos $O(\Sigma)$ no es cerrado con respecto a la intersección. Tomese como ejemplo:

$\Sigma=\{\alpha,\beta,r,\sigma,\sigma_1,\sigma_2\}$, $L(G)= \overline{r} + \overline{\alpha\beta} + \overline{\alpha\sigma_1\sigma_2\beta} + \overline{\sigma\sigma_1} + \overline{\sigma\sigma_2}$,

$K=\overline{\alpha\beta} + \overline{\alpha\sigma_1\sigma_2} + \overline{\sigma\sigma_1} + \overline{\sigma\sigma_2}$ , $P_1=\{r,\alpha,\beta,\sigma_1\}$ y $P_2=\{r,\alpha,\beta,\sigma_2\}$.

En consecuencia, el elemento minimal de $O(\Sigma)$ no necesariamente existe.

**PROPOSICION 3.2.1** Sean $\Sigma_1 \subseteq \Sigma_2 \subseteq \Sigma$. Si $\Sigma_1$ es K-observable, entonces $\Sigma_2$ es también K-observable.

**PRUEBA.** Supongamos que $\Sigma_1$ es K-observable y sean $P_1,P_2$ "las proyecciones" inducidas respectivamente por $\Sigma_1$ y $\Sigma_2$, entonces $KerP_2 \leq KerP_1$. Así, $KerP_1 \leq act(K)$. Por lo tanto, $KerP_2 \leq act(K)$. Luego, $\Sigma_2$ es K-observable.

**PROPOSICION 3.2.2** El subconjunto de eventos K-observable minimal $\Sigma_o{}^m=INFO(\Sigma)$ existe, si y sólo si, $O(\Sigma)$ es cerrado bajo la intersección de conjuntos.

**PRUEBA** Supongamos que $\Sigma_o{}^m=INFO(\Sigma)$ existe, entonces $\Sigma_o{}^m \subseteq \Gamma$, para todo $\Gamma \in O(\Sigma)$. En consecuencia, $\Sigma_o{}^m \subseteq \Gamma_1\cap\Gamma_2$, para todo $\Gamma_1,\Gamma_2 \in O(\Sigma)$; así $\Gamma_1\cap\Gamma_2$ es K-observable (ver proposición 3.2.1). Por lo tanto, $\Gamma_1\cap\Gamma_2 \in O(\Sigma)$.

Recíprocamente, supongamos que $O(\Sigma)$ es cerrado bajo la intersección de conjuntos, entonces obviamente $\Sigma_o^m = INFO(\Sigma)$ existe, pues $\bigcap_{\Gamma \in O(\Sigma)} \Gamma$ es K-observable, y es el subconjunto de eventos K-observable minimal.

Pongamos $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, $\Sigma_i = \Sigma - \{\sigma_i\}$ y $P_i$ la función inducida por $\Sigma_i$, $i=1,2,\ldots,n$. Consideremos la relación entre la K-observabilidad de $\Sigma_i$ y el subconjunto de eventos K-observable minimal. En el caso en que ninguno de los $\Sigma_i$ sea K-observable, se tiene claramente que $INFO(\Sigma) = \Sigma$.

**TEOREMA 3.2.1** Supongamos que $\Sigma_i$ es K-observable, para todo $i \in \{1,2,\ldots,m\}$ $(1 \leq m < n)$. Entonces, el subconjunto de eventos K-observable minimal existe, si y sólo si, $\bigcap_{i=1}^{m} \Sigma_i$ es K-observable.

**PRUEBA.** Supongamos que el subconjunto de eventos K-observable minimal existe, entonces $O(\Sigma)$ es cerrado bajo la intersección de conjuntos (ver proposición 3.3.2), así $\bigcap_{i=1}^{m} \Sigma_i$ es K-observable.

Recíprocamente, supongamos que $\bigcap_{i=1}^{m} \Sigma_i$ es K-observable. Afirmamos que $\bigcap_{i=1}^{m} \Sigma_i$ es minimal. En efecto, supongamos que existe un subconjunto de eventos K-observable $\Sigma^o$ tal que $\Sigma^o \subsetneq \bigcap_{i=1}^{m} \Sigma_i$, entonces existe $\sigma \in \Sigma, \sigma \in \bigcap_{i=1}^{m} \Sigma_i$ y $\sigma \notin \Sigma^o$. Como $\Sigma^o$ es K-observable, entonces $\bigcap_{i=1}^{m} \Sigma_i - \{\sigma\}$ es K-observable. Por lo tanto, $\Sigma-\{\sigma\}$ es K-observable. Pero, como $\sigma\in\Sigma_i$, para todo $i\in\{1,2,\ldots,m\}$, entonces $\Sigma-\{\sigma\}$ no es K-observable, lo cual es contradictorio. Luego, $\bigcap_{i=1}^{m}\Sigma_i$ es minimal.

**COROLARIO 3.2.1** Si $\bigcap_{i=1}^{m}\Sigma_i$ es K-observable, entonces

$\mathrm{INFO}(\Sigma)=\bigcap_{i=1}^{m}\Sigma_i$.

## BIBLIOGRAFIA


1. Lin F. and Wonham,W.M.," On observability of discrete event systems" . inform. Sci.,1988, vol.44,Nº3,pp 137-198.

2. Kvmar, R. et al., " On controllability and normality of discrete event dinamical systems " . systems and control letters, 1991 (17),pp. 157-168.

3. Tsitsik,j., " On de control of discrete event dinamical systems " . math. of control signals, and systems, 1989, Nº 2,pp 95-107.

4. Aveyard,R, " A boolean model for a class of discrete event systems ". IEEE trans. syst. man and eyb.,SMC-4, 1974,pp 249-258.

5. Beauquier,J and M. Nivat, " Application of formal languaje theory to problems of security and synchronization, in formal lenguage theory-perspective and open problems " . R.V. book,ed.,Academie press, New York, 1980,pp 407-454.

6. Peterson,J,L, " Petri Net Theory and the modelins of

systems ". 1981, prentice-hall, englewood,NJ.

7. Park,D, " Concorrency and automata on infinite secuences " in theoretical computer science, lecture notes in computer science 104, 1983 springer-verlas. New York,pp 167-183.

8. Eilenberg,S, " Automata,languages,and machines " , Vol A, 1974, Academic press, New York.

9. Fishman, G, S, " Principles of discrete event simulation " 1978, John Wiley, New York.

10. Harrison,M,A, " Introduction to switchins and automata theory " , 1965, Mc Graw-Hill, New York.

11. Hoare,C,A,R," Notes on comunicatins sequential processes" 1983,tech. monograph PRG-33, programmins research group, Oxford Univ. Computing laboratory, Oxford.

12. Milne, G and Milmer,K, " Concurrent processes and their syntax ". 1979, J. Assoc. comput. mach,26,pp2302-321.

13. Hopcroft,J,E, and Ullman, J,D, " Introduction to automata theory, languages, and computation ". 1979, Adisson-Wesley,Reading,MA.