

TEMA II: INTRODUCCIÓN A LA TEORÍA DE AUTÓMATAS FINITOS

Haciendo una comparación con la rama combinacional de Electrónica Digital, el Álgebra de Boole era la herramienta matemática para poder llevar a cabo la reducción de las fórmulas lógicas, y por lo tanto, del circuito de conmutación.

En el caso de los circuitos secuenciales, esta herramienta no nos basta debido a la dependencia temporal. Por lo tanto, necesitamos otra herramienta para minimizar la dependencia temporal de estos circuitos; esta herramienta es la *teoría de autómatas finitos*.

1. Introducción. Definición.

En primer lugar, vamos a definir lo que se entiende por autómata finito. Una posible definición de autómata finito es la siguiente:

Un **autómata finito** es un vector de tres elementos

$$M = (I, S, \delta, F)$$

donde I es el conjunto finito de entradas, S es el conjunto finito de estados (no vacío), δ es la función de transición de estados y F es el conjunto finito de estados finales (incluidos en S).

El hecho de que todos los conjuntos sean finitos, le otorga a este elemento el atributo de finito. También se puede desprender de la definición que un autómata finito es un tipo especial de máquina secuencial, en la cual no existen señales de salida como tal sino que sólo hay señales de entrada y estado, como sucedía en la máquina de Moore.

Debido a este motivo, las máquinas secuenciales cumplen todos los axiomas de los autómatas finitos. Siendo ésta la razón por la cual empezaremos el estudio de la teoría desarrollada alrededor de los autómatas finitos.

Al igual que en las máquinas secuenciales, la representación de los autómatas finitos puede llevarse a cabo de dos formas diferentes: mediante un diagrama de estados; o mediante una tabla de estados. De estas dos representaciones, la más intuitiva para una traducción partiendo de unas especificaciones de diseño es el diagrama de estados. No obstante, el trabajo con los diagramas de estado no es sencillo ni intuitivo. Por lo tanto, para un posterior procesamiento utilizaremos la tabla de estados.

2. Diagramas de Estado.

En los diagramas de estado podemos encontrar dos elementos: estados y transiciones. Los estados son las letras o símbolos enmarcados (dentro de un círculo generalmente). En cambio, las transiciones son arcos dirigidos que llevan asociadas una/s etiquetas.

Los estados se pueden definir como las posibles situaciones a las que puede llegar el autómatas que estamos describiendo. Dentro de estos estados podemos distinguir entre estados estables y estados inestables. Cuando existe alguna transición para la cual se llega al mismo estado desde el que se parte, se dice que es un **estado estable**. Mientras que si no existe ninguna transición que cumpla la condición anterior, se dice que es un **estado inestable**. En el caso de la máquina de ventas de refresco, los posibles estados pueden ser:

- No hay dinero en el interior de la máquina
- Existe el dinero suficiente para sacar un refresco
- En el interior de la máquina están cada una de las cantidades permitidas. Por ejemplo si sólo se admiten monedas de 10, 20 y 50 c., el refresco cuesta 60 c. y no se devuelve dinero, las cantidades pueden ser 10, 20, 30, 40, 50 y 60 c.

Las transiciones corresponderán a los eventos en las entradas que producirán los cambios de estado en el sentido de la flecha del arco. El cambio de estado se producirá si la condición de entrada coincide con la etiqueta asociada a dicha transición. De nuevo, en el caso de la máquina de refresco, las posibles condiciones de entrada podrían ser:

- Introducir las diferentes monedas permitidas: 10, 20 y 50 c.
- Pulsar el botón para obtener el refresco.

Así pues, una posible parte del diagrama del autómatas se puede ver en la figura 2.1:

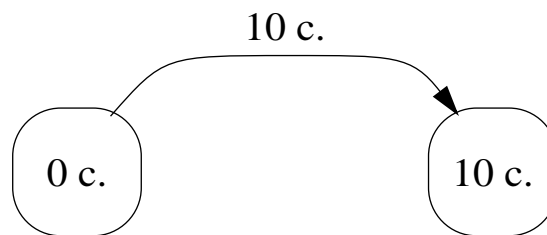


Figura 2.1.- Porción del diagrama de estado correspondiente a la máquina de refresco.

Dicha porción del diagrama se leería del siguiente modo:

... si no hay dinero almacenado en la máquina (estado de 0c.) e introducimos una moneda de 10 c. (evento), pasaremos a la situación de tener almacenado 10 c. (estado de 10c.)...

Si realizamos la descripción a través de la tabla de estado, la porción anterior del diagrama se mostraría en la tabla 2.1.

	10 c.	
0 c.	10 c.	

Tabla 2.1. Porción de la tabla de estado correspondiente a la máquina de refresco.

Por lo tanto, un posible diagrama de estado para la máquina de refresco se muestra en la figura 2.2. En este diagrama distinguimos 8 situaciones posibles: cada una de las cantidades que se pueden almacenar según las monedas aceptadas (0, 10, 20, 30, 40, 50 y 60 c.); y una situación en la que ya es posible dar el refresco. Los eventos involucrados en dicha máquina serán la inserción de cada una de las monedas (10, 20 y 50 c.), y pulsar el botón de refresco. Las transiciones de estado son las siguientes:

- Cuando introduzcamos una moneda, pasaremos al estado con la cantidad actualizada; pero en el caso de que sobrepasemos la cantidad del refresco, nos quedaremos en los 60 c. ya que no se devuelve cambio y no es necesario almacenar dicha cantidad superior.
- Cuando pulsemos el botón de refresco, B, no sucederá nada a menos que nos encontremos en el estado de 60 c., en cuyo caso se dará el refresco.

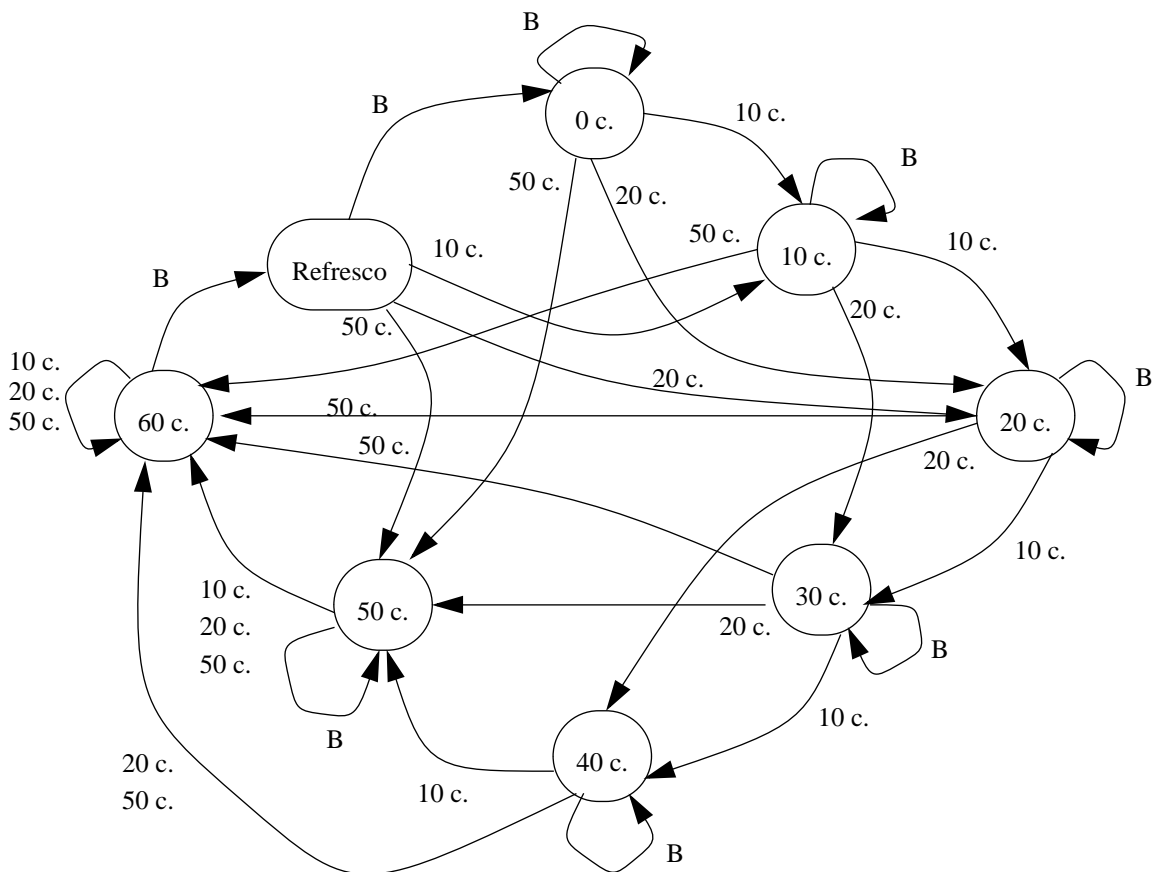


Figura 2.2.- Diagrama de estado correspondiente a la máquina de refresco.

3. Teoremas y Definiciones.

Como nuestro interés está centrado en el análisis y diseño, vamos a orientar los teoremas y definiciones de la teoría de autómatas finitos hacia los sistemas secuenciales. Una vez realizada esta aclaración, en esta teoría encontramos las siguientes definiciones:

Equivalencia: Una máquina M es equivalente a otra M^* , si para cualquier secuencia de entrada es posible encontrar algún estado inicial tal que la secuencia de salida sea la misma. Esta relación es denotada por $M = M^*$.

Compatibilidad: Una máquina M es compatible con otra M^* , si para cualquier secuencia de entrada es posible encontrar un estado inicial tal que la secuencia de salida sea la misma, siempre y cuando ésta esté especificada. Esta relación se denota por $M \sim M^*$.

Máquina completamente especificada: es una máquina en la cual no existen ningún estado total para los que no esté especificada la función de próximo estado y/o de salida. En caso contrario, se dice que la **máquina** está **incompletamente especificada**.

A continuación vamos a poner ejemplos de máquinas completas e incompletas. una primera máquina podría ser la siguiente:

A través de una línea serie va llegando bits, se desea realizar una máquina que indique la llegada de la secuencia $1 \rightarrow 0 \rightarrow 1$.

En esta máquina, la señal de entrada es la línea serie mencionada anteriormente, con las posibles combinaciones de '1' y '0'. Como ambas combinaciones son posibles, no podemos encontrar ninguna situación de no especificación. Por lo tanto, la máquina será completamente especificada, ya que tanto la salida como los próximos estados deberán estar especificados para todas las secuencias de entradas.

Una segunda máquina podría ser la descrita a continuación:

La figura 2.3 muestra el esquema de una máquina trituradora. La máquina es cuestión debe gobernar los motores de trituración según las siguientes especificaciones: cuando no exista nada que triturar, los motores deben estar apagados; cuando la máquina esté llena, ambos motores deben estar funcionando; cuando la máquina esté medio vacía, sólo debe funcionar un motor, y en particular aquel que en el estado anterior estuviera apagado (con el fin de garantizar una mayor vida activa de ambos motores).

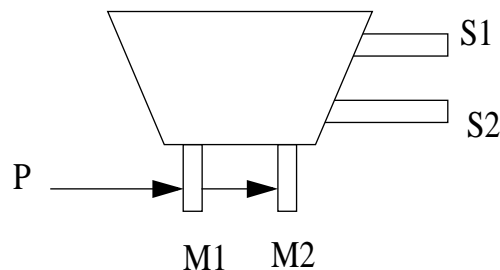


Figura 2.3.- Esquema de una máquina trituradora.

Esta máquina es un ejemplo claro de máquina incompleta ya que los sensores de presencia S1 y S2 nunca pueden tener la combinación S1 S2 = "1 0", ya que es físicamente imposible. Por lo tanto, para esta combinación de entrada, ni la salida ni el próximo estado estarán especificado porque no importará cuales sean al no llegar nunca dicha combinación. No obstante, podemos encontrarnos máquinas en los que la salida sí estará especificada pero el próximo estado no, y viceversa.

Según las definiciones de compatibilidad y de equivalencia, se comprueba el siguiente teorema:

Teorema 1. Se verifica que si las máquinas M y M* están especificadas de forma:

- completa: $M = M^* \Leftrightarrow M \sim M^*$
- incompleta: $M = M^* \Rightarrow M \sim M^*$

Demostración.- De las definiciones de equivalencia y compatibilidad se desprende que la primera engloba a la segunda por lo que la dirección hacia la derecha de las implicaciones queda demostrada. En cuanto a la dirección de la izquierda, sólo se puede verificar a priori cuando las dos definiciones sean iguales, caso que sólo se puede dar cuando se trata de máquinas completamente especificadas.

Cubrimiento. Una máquina M* cubre a otra M si cualquier secuencia de entrada aplicada a M* produce la misma secuencia de salida que si se aplica a M, siempre y cuando M esté completamente especificada. Esta relación es denotada por $M \subset M^*$.

Estados compatibles: Dos estados internos S_i y S_j se denominan compatibles, si para cualquier secuencia de entrada, tomando ambos estados como iniciales, las dos secuencias de salida correspondientes resultan idénticas siempre que ambos estados de salida estén completamente especificados. Esta relación es denotada por $S_i \sim S_j$. En caso contrario, los estados se denominan **incompatibles**.

La definición de compatibilidad se reduce a comprobar que tienen el mismo valor de salida y sus próximos estados son compatibles para cualquier condición de entrada, ya que la aplicación de la definición de forma directa es inviable.

Clase de compatibilidad o clases o compatibles: Consiste en un conjunto de estados, los cuales son compatibles dos a dos.

Clase máxima o máximo compatible: Será aquella clase que no es un subconjunto de ninguna otra clase, y por lo tanto no está contenida totalmente en ninguna otra.

Colección de clases cerrada: Es un conjunto de clases, en el cual los próximos estados correspondientes a cada clase están contenidos al menos en una clase de la colección, para cualquier secuencia de entradas.

Cubrimiento de una tabla de estados: Se trata de una colección de clases que cubre a una tabla de estados, si cada estado interno de la tabla está contenido en una clase de la colección como mínimo.

4. Minimización del Número de Estados.

El número de estados es el conjunto de situaciones que es necesario almacenar. Por lo tanto, nuestro interés en la teoría de autómatas finitos se centra en la reducción del número de estados. De esta forma obtendremos un diseño mínimo en cuanto a elementos de memoria, que son los elementos críticos en el diseño semicustom.

El problema de la minimización del número de estados para una máquina M es equivalente a obtener una colección cerrada de clases con un número mínimo que cubra su tabla de estados. A esta colección la denominaremos **cubrimiento mínimo**, y los estados internos de la máquina reducida serán las clases de dicho cubrimiento.

Llegado este momento, debemos preguntar si la minimización de una máquina incompletamente especificada puede reducirse al de una máquina completamente especificada. Para ello podríamos suponer que la máquina mínima sería la de una máquina completa dándole un determinado valor a las inespecificaciones.

Consideremos como ejemplo la máquina de Moore especificada por la tabla de estados mostrada en la tabla 2.2:

	a	b	S
A	A	B	0
B	B	C	-
C	A	C	1

Tabla 2.2. Ejemplo de máquina incompletamente especificada.

Al utilizar todas las combinaciones de salida del estado B, nos quedarían las tablas de estado mostradas en la tabla 2.3 y tabla 2.4:

	a	b	S
A	A	B	0
B	B	C	0
C	A	C	1

Tabla 2.3. Máquina completamente especificada obtenida a partir del ejemplo de la tabla 2.2.

	a	b	S
A	A	B	0
B	B	C	1
C	A	C	1

Tabla 2.4. Máquina completamente especificada obtenida a partir del ejemplo de la tabla 2.2.

En la tabla 2.3 se verifica que los tres estados son incompatibles por las siguientes razones:

- Los estados A y C son incompatibles por tener diferentes valores de salida.
- Los estados B y C son incompatibles por tener diferentes valores de salida.
- Los estados A y B son incompatibles ya que, a pesar de tener el mismo valor de salida, los próximos estados de la combinación de entrada b (B - C) son incompatibles.

En la tabla 2.4 se verifica que los tres estados son incompatibles por las siguientes razones:

- Los estados A y C son incompatibles por tener diferentes valores de salida.
- Los estados A y B son incompatibles por tener diferentes valores de salida.
- Los estados B y C son incompatibles ya que, a pesar de tener el mismo valor de salida, los próximos estados de la combinación de entrada b (A - B) son incompatibles.

No obstante, en la tabla 2.4 se verifica que los estados A - B y B - C son compatibles por las siguientes razones:

- Los estados A y C son incompatibles por tener diferentes valores de salida.
- Los estados A y B serán compatibles si lo son los estados B y C.
- Los estados B y C serán compatibles si lo son los estados A y B.
- Como no se han encontrados incongruencias en la compatibilidad de los estados A y B, y B y C, ambas parejas se consideran compatibles.

Por lo tanto, si elegimos dos nuevos estados α y β como los correspondientes a los estados AB y BC, nos encontramos ante una máquina equivalente a la original; y por tener un menor número de estados, una máquina más eficiente (que para este caso particular será mínima). La tabla de estados correspondiente a esta nueva máquina será la mostrada en la tabla 2.5:

	a	b	S
α	α	β	0
β	α	β	1

Tabla 2.5. Máquina mínima del ejemplo de la tabla 2.2.

Por lo tanto, llegamos a la conclusión de que debemos tratar de forma diferente a las máquinas completamente e incompletamente especificadas.

4.1. Máquinas completamente especificadas.

En este tipo de máquinas se cumplen los siguientes teoremas.

Teorema 2. La relación de compatibilidad se denomina equivalencia, donde cada clase de compatibilidad representa una clase de equivalencia.

Demostración.- Utilizando las definiciones de compatibilidad y equivalencia, sabemos que la única diferencia existente radica en la necesidad de que las salidas y próximos estados sean completamente especificados en el primer caso. Como en las máquinas que estamos tratando, esta restricción se cumple para todos los estados y secuencias de entrada, ambas definiciones coinciden.

Teorema 3. El conjunto de máximos compatibles es el cubrimiento mínimo cerrado.

Demostración.- En primer lugar, en el conjunto de máximos compatibles se encuentran representados todos los estados de la máquina en cuestión, por lo cual este conjunto será un cubrimiento de dicha máquina. Además, por la definición de máximo compatible sabemos que no serán ningún subconjunto de otro máximo, por lo cual todos ellos deberán estar representados en la máquina. Por lo tanto, el número mínimo de clases necesarias será el formado por todos los máximos compatibles.

Por lo tanto, la reducción del número de estados consistirá en la obtención de los máximos compatibles correspondientes a la máquina considerada. Para llegar a este fin, primero tenemos que obtener los estados compatibles, para después ver cuáles de ellos forman los diferentes máximos compatibles de la máquina secuencial.

4.1.1. Obtención de los estados compatibles

Para obtener los estados compatibles, un posible método es el denominado como **tabla de pares compatibles**. Esta tabla tiene la forma mostrada en la figura 2.4, donde $S_1 - S_n$ son los estados de nuestra máquina secuencial. La tabla se rellena de la siguiente forma:

- Cuando los estados son incompatibles, ya que existe combinaciones de entrada para las que el valor de las señales de salida es diferente, se identifica con un aspa (X).
- Cuando los estados son compatibles, ya que para todas las combinaciones de entrada tanto el valor de las salidas como los próximos estados son iguales, se identifica con una onda (~).
- Cuando no se cumple ninguna de las condiciones anteriores, es decir, todas las combinaciones de entrada tienen el mismo valor de salida pero diferentes próximos estados, se identifica con la pareja de los próximos estados, ya que estos nos determinarán si realmente existe compatibilidad o no.

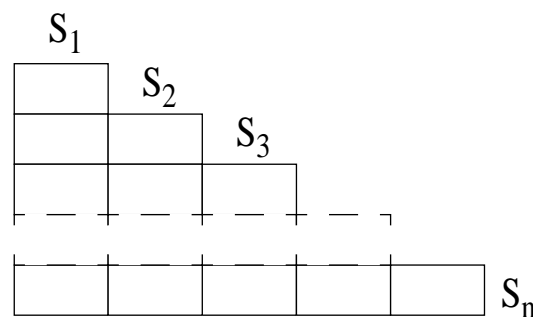


Figura 2.4.- Apariencia de la tabla de pares compatibles.

Una vez completada la tabla, determinaremos la compatibilidad de todos los estados siguiendo las siguientes reglas (que sólo tendrán valor para el tercer punto):

- En el caso de que los próximos estados sean incompatibles, los estados presentes también serán incompatibles.
- En el caso de que los próximos estados sean compatibles, los estados presentes también serán compatibles.
- En el caso de que los próximos estados sean los estados presentes, éstos serán compatibles.
- Una vez que hayamos tachado todos los estados incompatibles, el resto se considerarán compatibles.

Por ejemplo, si consideramos la máquina representada por el diagrama de estados mostrado en la figura 2.5, junto con su tabla de estados. En ella podemos distinguir un total de siete

estados (desde el estado A al estado G), una señal de entrada y una señal de salida. La tabla de pares compatibles también es mostrada en la misma figura.

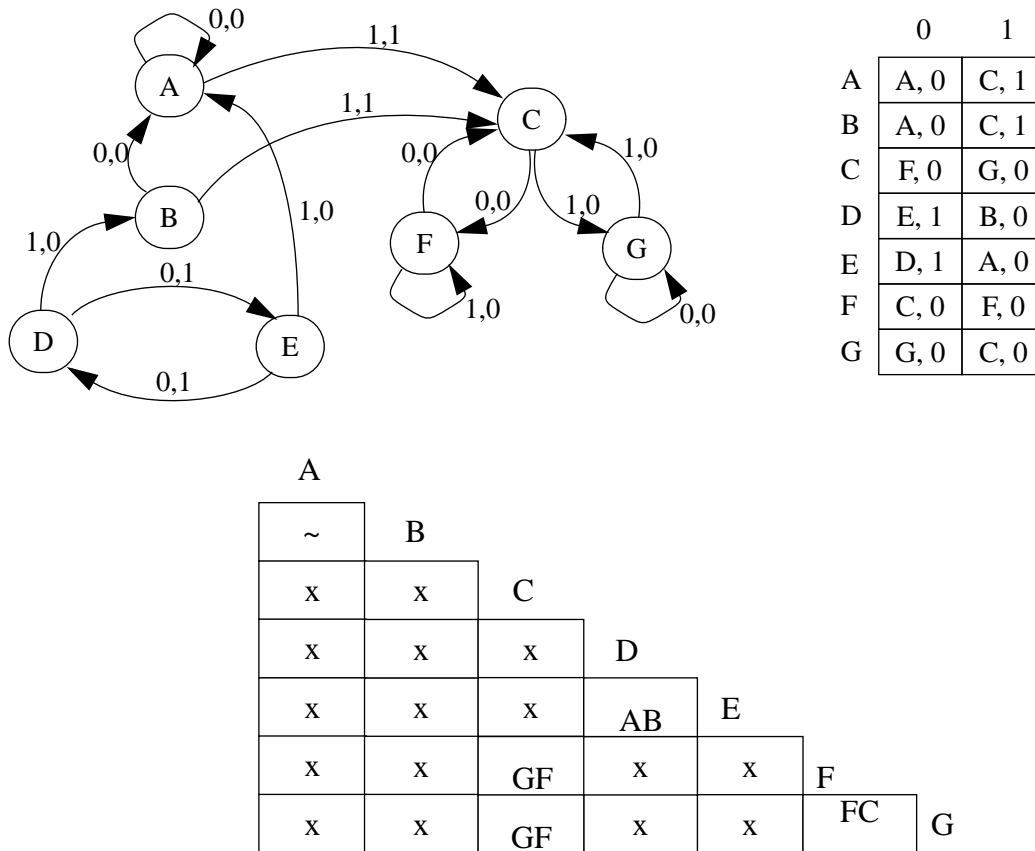


Figura 2.5.- Diagrama de estados del ejemplo de la máquina completamente especificada.

Los únicos estados sobre los que se debe prestar una especial atención (no se ha podido determinar en una primera pasada su compatibilidad) son los pares C-F, C-G, D-E y F-G. Si empezamos por el par D-E, observamos sus próximos estados: D-E para x=0 (que por ser el mismo par no aparece en la tabla), y A-B para x=1; luego podemos afirmar que este par es compatible ya que los próximos estados para todas las combinaciones de entrada son compatibles. En el caso de los pares C-F, C-G y F-G, vemos que la compatibilidad de un par depende de la de otro formando un ciclo como podemos ver a continuación:

$$C \sim F \Rightarrow G \sim F \Rightarrow F \sim C$$

$$C \sim G \Rightarrow G \sim F \Rightarrow F \sim C \Rightarrow G \sim F$$

Como no se ha llegado a ninguna condición de incompatibilidad, puede ser cierto ambas condiciones: que todos los estados sean incompatibles o que sean compatibles. Pero como nuestro interés está en la reducción del número de estados para garantizar la minimización del circuito secuencial equivalente, siempre se considerará que todos son compatibles. Por lo tanto, los pares compatibles que encontramos en la máquina de partida son los siguientes: A~B, D~E, C~F, C~G y F~G.

4.1.2. Obtención de los máximos compatibles

Una vez obtenidos los pares compatibles, el siguiente paso consiste en determinar los máximos compatibles. Para ello disponemos de dos métodos diferentes: un método tabular basado en los estados compatibles; y un método algebraico basado en los pares incompatibles.

El método tabular está basada en una tabla con tres columnas: una primera en la que colocamos los estados; otra en la que colocamos los estados compatibles con el de la primera columna; y un tercero donde se agrupan los estados compatibles dos a dos, por lo que la última celda de esta columna contendrá al cubrimiento cerrado de máximos compatibles. Para evitar la duplicidad, en la columna de estados compatibles sólo se considerarán los estados ya vistos, es decir, si empezamos por el último estado de la tabla de pares compatibles, la compatibilidad sólo se considerará con los estados de la columna del estado en cuestión. Si continuamos con el ejemplo anterior, obtendríamos la tabla 2.6.

Estados	CEC	LC
G		G
F	G	FG
E	-	E, FG
D	E	DE, FG
C	F,G	DE, CFG
B	-	B, DE, CFG
A	B	AB, DE, CFG

Tabla 2.6. Tabla de la lista de compatibles del ejemplo de la figura 2.5.

Por lo tanto, la lista de máximos compatibles para el ejemplo será: AB, DE, CFG

En el segundo método, se obtiene una función de conmutación como expresión de producto de sumas tal que cada término suma es la suma de los pares incompatibles. A partir de esta fórmula se obtiene una expresión mínima en forma de suma de productos utilizando los teoremas del Álgebra de Boole, entre los que tendrán una especial importancia los teoremas de absorción y de idempotencia:

- $a + a = a$
- $a \cdot a = a$
- $(a + b)(a + c) = a + b \cdot c$
- $a + a \cdot b = a$

Una vez obtenida la expresión mínima en suma de productos, se verifica que por cada término producto habrá un máximo compatible, que estará formado por los estados que no aparezcan en dicho término. Si aplicamos este método a la máquina anterior, para la cual los pares incompatibles eran: A-C, A-D, A-E, A-F, A-G, B-C, B-D, B-E, B-F, B-G, C-D, C-E, D-F, D-G, E-F, E-G; la función a minimizar es la siguiente:

$$F = (A+C)(A+D)(A+E)(A+F)(A+G)(B+C)(B+D)(B+E)(B+F)(B+G)(C+D) \\ (C+E)(D+F)(D+G)(E+F)(E+G)$$

$$F = (A+CDEFG)(B+CDEFG)(C+DE)(D+FG)(E+FG)$$

$$F = (AB+CDEFG)(C+DE)(DE+FG) = (AB+CDEFG)(CFG+DE)$$

$$F = ABCFG + ABDE + CDEFG$$

Una vez tenemos todos los términos productos, habrá tantos máximos compatibles como términos productos. Estos máximos compatibles estarán formados por los estados que faltan en dichos términos.

- Del término ABDE, obtenemos el máximo compatible CFG.
- Del término ABCFG, obtenemos el máximo compatible DE.
- Del término CDEFG, obtenemos el máximo compatible AB.

4.1.3. Obtención de la máquina mínima

El proceso finaliza obteniendo la tabla mínima. Ésta se obtiene de la siguiente forma:

- Asociando a cada máximo compatible un estado.
- Redefiniendo los próximos estados en el nuevo alfabeto.

Así para nuestro ejemplo, la tabla mínima de estado sería la mostrada en la tabla 2.7:

	x=0	x=1
α	$\alpha,0$	$\chi,1$
β	$\beta,1$	$\alpha,0$
χ	$\chi,0$	$\chi,0$

Tabla 2.7. Tabla de estados de la máquina mínima del ejemplo de la figura 2.5.

que se corresponderá con el siguiente diagrama de estados:

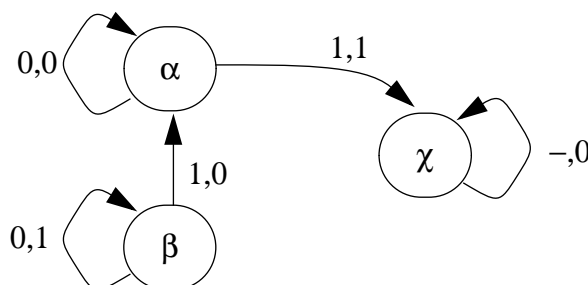


Figura 2.6.- Diagrama de estados de la máquina mínima del ejemplo de la figura 2.5.

4.2. Máquinas incompletamente especificadas (Apéndice).

Ya se ha discutido que este tipo de máquinas debe tener un tratamiento diferente a las máquinas completas. Este hecho viene motivado principalmente porque podemos encontrar que un estado es compatible con otros dos que no lo sean entre sí. En este tipo de máquinas, encontramos las siguientes definiciones:

Sea C_α un compatible y $C_{\alpha i}$ el compatible formado por los próximos estados de los estados C_α para una condición de entrada X_i . Entonces se dice que **$C_{\alpha i}$ es implicado por C_α** .

Para cualquier compatible C_α , el conjunto de compatibles implicados, P_α , es el conjunto de todos los compatibles $C_{\alpha i}$ de implicados por C_α tal que:

- $C_{\alpha i}$ tenga más de un elemento
- $C_{\alpha i} \subsetneq C_\alpha$
- $C_{\alpha i} \not\subset C_{\alpha j}$ si $C_{\alpha j} \in P_\alpha$

Un compatible C_β se dice **excluido** por otro C_α si:

- $C_\alpha \supset C_\beta$
- $P_\alpha \subseteq P_\beta$

Un **compatible prima** es aquel que no es excluido por ningún otro compatible. Serán equivalentes a los máximos compatibles en el caso de máquinas completamente especificadas.

Con esta última definición concuerdan todos los máximos compatibles. Además, se verifica que un compatible que implique un conjunto de compatibles vacío excluye a todos sus subcompatibles.

Teorema 4. Existe al menos un cubrimiento mínimo cuyos miembros son compatibles primos.

Demostración.- En el caso de las máquinas completamente especificadas, el cubrimiento mínimo era aquel formado por los máximos compatibles. Y como en el cubrimiento formado por los compatibles primas, se encontrarán todos los máximos compatibles; existirá un cubrimiento mínimo formado por compatibles primas.

De esta forma, debemos hallar todos los compatibles primas, de los cuales se elegirán los que estarán en el cubrimiento mínimo. La selección debe satisfacer tres condiciones:

- Las compatibles seleccionados deben cubrir todos los estados internos.
- El conjunto de compatibles seleccionado debe ser cerrado.
- El número de compatibles debe ser mínimo.

Por lo tanto, la reducción de la tabla de estados correspondiente a una máquina incompletamente especificada conlleva los siguientes pasos:

- En primer lugar debemos obtener las clases primas. Para lo cual descomponemos los máximos compatibles en subclases, y después verificaremos cuáles son primas.
 - Las lista inicial de clases primas se constituye con los máximos compatibles mayores.
 - Después se añaden a la lista los siguientes máximos compatibles, inferiores en tamaño, que no sean excluidos por ningún máximo existente ya en la lista.
 - El proceso se repite hasta que ya no queden compatibles.

Vamos a ver como ejemplo la reducción de la máquina mostrada en la tabla 2.8. En esta tabla encontramos un total de ocho estados posibles con siete condiciones de entrada. Las inespecificaciones son tales que pueden estar inespecificadas la salida, el próximo estado o ambos.

	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇
a	a,-	-, -	d,0	e,1	b,0	a,-	-, -
b	b,0	d,1	a,-	-, -	a,-	a,1	-, -
c	b,0	d,1	a,1	-, -	-, -	-, -	g,0
d	-, -	e,-	-, -	b,-	b,0	-, -	a,-
e	b,-	e,-	a,-	-, -	b,-	e,-	a,1
f	b,0	c,-	-,1	h,1	f,1	g,0	-, -
g	-, -	c,1	-, -	e,1	-, -	g,0	f,0
h	a,1	e,0	d,1	b,0	b,-	e,-	a,1

Tabla 2.8. Ejemplo de minimización de una máquina incompleta.

La obtención de los máximos compatibles se realiza de la misma forma que en el caso de las máquinas completas. Es decir, en primer lugar hallamos la tabla de pares compatibles:

A

AD	B						
x	~	C					
BE	DE AB	DE AG	D				
AB AD	DE AB	x	~	E			
x	x	DC	x	CE DE EG	F		
~	x	DC GF	EC DE AF	x	EH	G	
x	x	x	~	AB AD	x	x	H

Figura 2.7.- Tabla de pares compatibles del ejemplo de la tabla 2.8.

seguidamente hallamos la lista de máximos compatibles:

Estados	CEC	LC
		H
G	--	H, G
F	G	H, FG
E	H	EH, FG
D	E, H	DEH, FG
C	D, F, G	CD, DEH, CFG
B	C, D, E	BCD, BDE, DEH, CFG
A	B, D, E, G	ABDE, BCD, DEH, CFG, AG

Es decir, los máximos compatibles de dicha máquina son $MC = \{abde, ag, bcd, cfg, deh\}$. Por lo tanto,

- la lista de compatibles primas empezará por “abde”, cuyo conjunto de implicantes será $P = \{-\}$. Esto implica que dicho compatibles excluye a todas sus subclases.
- Añadiremos los máximos compatibles en orden (de tres estados) y las combinaciones de tres estados de los compatibles superiores cuyas lista de implicados no sea el conjunto vacío, por lo que la lista quedará:

$$LC = \{abde, bcd, cfg, deh\}$$

$$P_{bcd} = \{de, ab, ag\}$$

$$P_{cfg} = \{dc, eh\}$$

$$P_{deh} = \{ab, ad\}$$

- A continuación, añadiremos los compatibles de dos estados y las combinaciones de dos estados de los compatibles superiores cuyas lista de implicados no sea el conjunto vacío:

$$LC = \{abde, bcd, cfg, deh, ag, bc, bd, cd, cf, cg, fg, de, dh, eh\}$$

$$P_{ag} = \{-\}$$

$$P_{bc} = \{-\} \quad P_{bd} = \{de, ag\} \quad P_{cd} = \{de, ab\}$$

$$P_{cf} = \{cd\} \quad P_{cg} = \{cd, fg\} \quad P_{fg} = \{eh\}$$

$$P_{de} = \{-\} \quad P_{dh} = \{-\} \quad P_{eh} = \{ab, ad\}$$

- Por último añadiremos los estados (compatibles de un solo estado) que no estén en compatibles superiores cuyas listas de implicados sean el conjunto vacío.

$$LC = \{abde, bcd, cfg, deh, ag, bc, bd, cd, cf, cg, fg, de, dh, eh, f\}$$

De todos estos compatibles, vemos que quedan excluidos bd (por abde), de (por abde) y eh por (deh). Además los compatibles ag, bc, de y dh excluyen a todas sus subclases ya que tienen un conjunto de implicantes vacío. Por lo tanto, el conjunto de clases primas sería el siguiente:

$$LC = \{abde, bcd, cfg, deh, ag, bc, cd, cf, cg, fg, dh, f\}$$

- Una vez que hemos obtenido todos los compatibles primas, debemos seleccionar aquellas que cumplan las tres restricciones que vimos anteriormente. Para realizar esta selección, podemos utilizar las denominadas tablas de cubrimiento y cierre o tablas CC. En estas tablas, las filas corresponden a los compatibles primas, mientras que las columnas están divididas en dos partes: una parte de cubrimiento, cada columna corresponde a cada estado de la máquina inicial, en la que tachamos los estados que son cubiertos por los compatibles primas; una parte de cierre, cada columna corresponde a un próximo estado de un compatible prima, de tal forma que se coloca un \cdot (punto) en la celda correspondiente al estado presente y una x (cruz) en la celda correspondiente al próximo estado.

La tabla correspondiente al ejemplo anterior sería la mostrada en la tabla 2.9:

	Cubrimiento								Cierre													
	a	b	c	d	e	f	g	h	ab	ag	de	cd	eh	ab	ad	de	ag	cd	cd	fg	eh	
abde	x	x		x	x				x		x			x	x	x						
bcd		x	x	x					.	.	.	x						x	x			
cfg			x			x	x					.	.								x	
deh				x	x			x			x		x	.	.	x						x
ag	x							x		x							x					
bc		x	x																			
cd			x	x								x				.	.	x	x			
cf			x			x											.					
cg			x					x										.	.			
fg						x	x													x	.	
dh				x				x														
f						x																

Tabla 2.9. Tabla de cubrimiento y cierre (CC) correspondiente al ejemplo de la tabla 2.8.

- Una vez obtenida la tabla CC, pasamos a aplicar una serie de reglas para tratar de eliminar filas y/o columnas. Para lo cual, daremos dos definiciones de dominación:

Una fila r_i es x -dominada por otra r_j , si r_j tiene todas las cruces de r_i y no tiene puntos.

Una columna u_i es dominada por otra u_j , si u_j tiene todas las cruces y el punto (si lo hubiera) de u_i .

Si la fila r_i es x -dominada por otra, entonces r_i puede ser eliminada así como todas las columnas en las que tenga un punto.

Si una columna u_i sólo tiene una cruz en la fila r_j y blancos en todas las demás, se debe seleccionar el compatible r_j . Después eliminamos dicha fila y todas las columnas en las que tenga cruces.

Si una columna u_i es dominada por otra u_j , podemos eliminar la columna u_i .

Si la fila r_i sólo tiene puntos, puede ser eliminada con todas las columnas donde tenga los puntos.

Cuando no sea posible reducir más la tabla, diremos que nos encontramos ante una tabla cíclica.

Utilizando estas reglas podemos ver que la columna “de” (de “bcd”) domina a la columna “ab” (de “bcd”), por lo que podemos eliminarla. También vemos que la columna “ab” (de “deh”) domina a la columna “ad” (de “deh”), por lo que podemos eliminarla. Y que la columna “d” domina a la columna “e”, por lo que podemos eliminarla. Resultando la tabla CC reducida mostrada en la tabla 2.10.

- Una vez que hayamos reducido la tabla, pasamos a aplicar el método de la función de Petryck. Esta función es una función lógica en producto de sumas, en la que existe un término suma por cada columna en la que haya algún punto o alguna cruz. Estos términos estarán formados por cada compatible sin complementar, si hay una cruz, o complementado, si hay un punto. Esta función es representada en forma de suma de

	Cubrimiento							Cierre										
	a	b	c	e	f	g	h	ab	ag	cd	eh	ad	de	ag	cd	cd	fg	eh
abde	x	x		x				x				x	x					
bcd		x	x					.	.	x					x	x		
cfg			x		x	x				.	.						x	
deh				x			x				x	.	x					x
ag	x					x			x					x				
bc		x	x															
cd			x							x		.	.	x	x			
cf			x		x									.				
cg			x			x									.	.		
fg					x	x										x	.	
dh							x											
f					x													

Tabla 2.10. Tabla de cubrimiento y cierre (CC) correspondiente al ejemplo de la tabla 2.8 minimizada.

productos. Así, la función mínima será aquella formada por los compatibles del término producto en el que existan el menor número de compatibles sin complementar.

Por lo tanto, si denominamos C_1, \dots, C_{12} a los compatibles $abde, \dots, f$, la función de Petryck será:

$$\begin{aligned}
 P = & (C_1+C_5)(C_1+C_2+C_6)(C_2+C_3+C_6+C_7+C_8+C_9)(C_1+C_4)(C_3+C_8+C_{10}+C_{12}) \\
 & (C_3+C_5+C_9+C_{10})(C_4+C_{11})(C_1+C_2')(C_2'+C_5)(C_2+C_3'+C_7)(C_3'+C_4)(C_1+C_4') \\
 & (C_1+C_4+C_7')(C_5+C_7')(C_2+C_7+C_8')(C_2+C_7+C_9')(C_3+C_9'+C_{10})(C_4+C_{11}') =
 \end{aligned}$$

Aplicando las leyes del álgebra de conmutación.

$$\begin{aligned}
 P = & C_1C_4C_3C_5C_2 + C_1C_4C_3C_5C_7 + C_1C_4C_{10}C_5C_2 + C_1C_4C_{10}C_5C_7 + C_1C_4C_{10}C_5C_8'C_9'C_3'C_6 \\
 & + C_1C_4C_{10}C_2'C_7'C_8'C_9'C_3'C_6
 \end{aligned}$$

De aquí podemos comprobar que el término con menos variables sin complementar es el último con un total de cuatro compatibles, de tal forma que podemos determinar que la máquina con un menor número de estados será la compuesta por los compatibles $C_1=abde$, $C_4=deh$, $C_6=bc$ y $C_{10}=fg$. Por último, la tabla reducida se muestra en la tabla 2.11:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
C_1	$C_1,$	$C_4,1$	$C_1,0$	$C_1,1$	$C_1,0$	$C_1,-$	$C_1,-$
C_4	$C_1,1$	$C_{1(4)},0$	$C_1,1$	$C_{1(6)},0$	$C_{1(6)},1$	$C_{1(4)},-$	$C_1,1$
C_6	$C_{1(6)},0$	$C_{1(4)},1$	$C_1,1$	$-, -$	$C_1,-$	$C_1,1$	$C_{10},0$
C_{10}	$C_{1(6)},0$	$C_6,1$	$-,1$	$C_4,1$	$C_{10},1$	$C_{10},0$	$C_{10},0$

Tabla 2.11. Tabla de estados reducida