SIMPLIFICACIÓN DE GRAFOS DE ESTADOS:
UNA METODOLOGÍA CONCEPTUAL ALTERNATIVA

T. POLLÁN Y J.M. LÓPEZ
Escuela Universitaria de Ingeniería Técnica Industrial de Zaragoza.
Departamento de Ingeniería Electrónica y Comunicaciones. Universidad de Zaragoza. España

Consideramos que la forma clásica de simplificación de estados, mediante tablas de transiciones o tablas de fases, es poco útil desde un punto de vista práctico e ingenieril y proponemos una forma alternativa basada en detectar, en términos de especificaciones de cada problema, y agrupar aquellos estados que son «iguales» en cuanto a salidas y transiciones de salida y aquellos que son «distinguibles (ellos y sus transiciones) desde fuera» por medio de alguna variable de entrada .

1. Introducción y conclusiones

En la representación de circuitos secuenciales mediante grafos de estados, resulta conveniente simplificar, en lo posible, tales grafos; pues la reducción del número de estados que los conforman facilita su diseño y reduce el número de variables de estado, las expresiones algebraicas de su evolución y el circuito digital necesario para su implementación .

El tratamiento que dan la mayor parte de los textos a la agrupación de estados de un grafo, utilizando tablas de transición de estado o tablas de fases, es correcto e interesante desde un punto de vista matemático pero poco pertinente y escasamente útil en cuanto al diseño de un sistema de control.

A lo largo de nuestra experiencia como docentes de electrónica digital hemos desarrollado otra forma de abordar el problema, a nuestro entender más eficiente, práctica y conceptual. Una forma que requiere mayor capacidad de razonamiento y «meterse en el problema» con mayor profundidad; pero eso es, precisamente, desde una perspectiva ingenieril, lo que debe hacer un diseñador cuando se enfrenta a la descripción del comportamiento de un sistema que requiere memoria. El grafo de estados debe ser el resultado de la comprensión profunda y razonada de las «situaciones» que el sistema necesita «recordar» [1].

Existen dos mecanismos que permiten agrupar dos o más estados en uno solo y que se refieren a dos situaciones conceptuales: estados que no necesitan diferenciarse entre sí y estados que ya se distinguen por variables exteriores:

a) son agrupables aquellos estados que no precisan diferenciarse entre sí, por tener las mismas salidas, es decir, los mismos vectores de salida y las mismas transiciones desde ellos;

b) también son agrupables los estados que, tanto ellos como las transiciones que se producen desde ellos, pueden diferenciarse mediante variables exteriores, o sea mediante valores de las entradas.

Los estados se refieren a situaciones (secuencias de vectores de entrada) que el sistema necesita conservar en su memoria en forma diferenciada. No es preciso diferenciar aquellas situaciones que producen los mismos efectos: los mismos vectores de salida y las mismas transiciones. Tampoco es necesario distinguir internamente aquellas que se diferencian por el valor de alguna de las variables de entrada. En el primer caso, se evita diferenciar situaciones que son idénticas por serlo sus efectos; en el segundo, puede prescindirse de distinguir en memoria lo que puede diferenciarse externamente.

La reducción del número de estados de un grafo es ciertamente útil cuando permite construirlo con menor número de variables de estado; en ocasiones, el grafo resultante pasa a ser de tipo Mealy y las funciones de activación de las salidas son más complejas que en el grafo inicial.

2. Ejemplos de simplificación: estados con los mismos efectos

Son agrupables (reducibles a uno solo) aquellos estados que producen los mismos efectos, es decir, tienen los mismos vectores de salida y las mismas transiciones desde ellos.

Consideremos un ejemplo muy simple*: un motor con giro en sentido único controlado a través de dos pulsadores; activando uno cualquiera de ellos el motor se pone a girar y se para al presionar ambos pulsadores a la vez.*

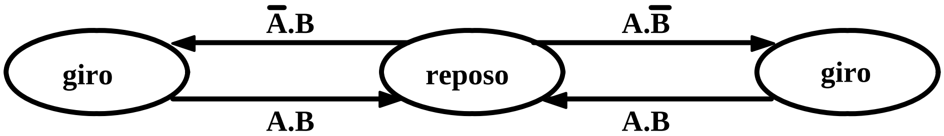

Figura 1. Grafo de estados de un motor controlado por dos pulsadores A y B.

Ahora bien, en el grafo de la figura 1, los dos estados de *giro* tienen el mismo vector de salida (el motor girando en el mismo sentido) y la misma transición de salida (con A.B pasan al reposo); equivalen a un solo estado (figura 2).

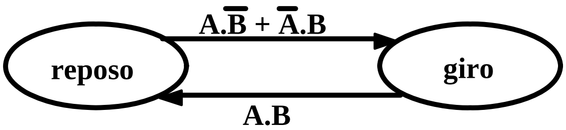

Figura 2. Grafo de estados simplificado de un motor controlado por dos pulsadores A y B.

Este ejemplo presenta un problema: al activar ambos pulsadores el motor se para, pero al soltarlos lo más probable es que un pulsador pase a **0** antes que el otro produciéndose la condición de activación del motor; el grafo adecuado para evitar esto requiere tres estados.

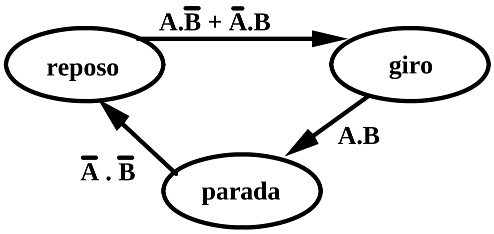

Figura 3. Grafo de estados que evita la activación del motor al soltar ambos pulsadores.

Un segundo ejemplo: *semáforo de aviso de paso de tren en un cruce de vía única bidireccional con un camino; la vía posee, a ambos lados del cruce y a una distancia adecuadamente grande, sendos detectores de paso de tren* ***a*** *y* ***b****; los trenes circulan por ella en ambas direcciones y se desea que el semáforo señale presencia de tren desde que éste alcanza el primer sensor en su dirección de marcha hasta que pasa por el segundo sensor tras abandonar el cruce.*

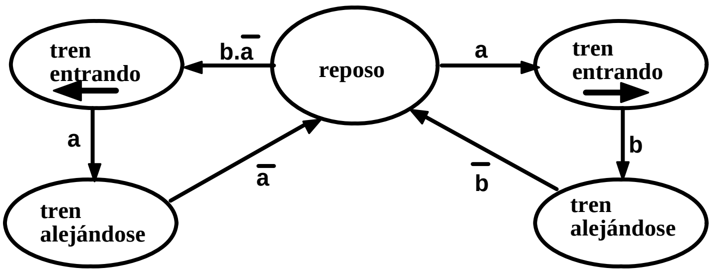

Figura 4. Grafo de estados para el control de un semáforo de aviso de paso de tren.

El estado relativo a *tren alejándose* es necesario pues, si se prescinde de dicho estado, un mismo tren al alejarse lleva al sistema al estado de reposo pero, inmediatamente después, será considerado como tren entrando en la dirección opuesta y causará una evolución incorrecta.

El grafo anterior, tal como está representado, no es simplificable, pero podemos modificar la condición booleana de las transiciones que salen de ambos estados inferiores (*tren alejándose*), de forma que ambas se produzcan con $\overline{a} \, . \, \overline{b}$ (cuando el tren abandona el cruce los detectores estarán a **0**) y, entonces, dichos estados son agrupables. Aunque difieren en un solo estado, el primero de los grafos necesita, al menos, tres variables de estado, mientras que el segundo puede codificarse con dos.

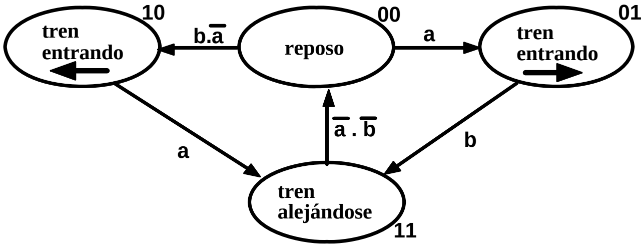

Figura 5. Grafo de estados simplificado para el control de un semáforo de aviso de paso de tren.

Consideremos un tercer ejemplo: *sea un timbre que, para filtrar llamadas molestas, dispone de dos pulsadores **A** y **B**, pero que solamente suena si se ejecuta la siguiente secuencia sobre ellos: **A**, **A** y **B**, **B**, **A** y **B** (se pulsa **A**; sin soltar, se pulsan ambos, **A** y **B**; se suelta **A**; y, por, último, sin soltar **B**, se vuelven a pulsar ambos, **A** y **B**).* Su grafo de estados puede ser el representado en la figura 16.

Para evitar el tener que acordarse por cuál de los pulsadores hay que iniciar la secuencia, pueden admitirse las dos secuencias simétricas: (**A**, **A y B**, **B**, **A y B**) o, también, (**B**, **A y B**, **A**, **A y B**).

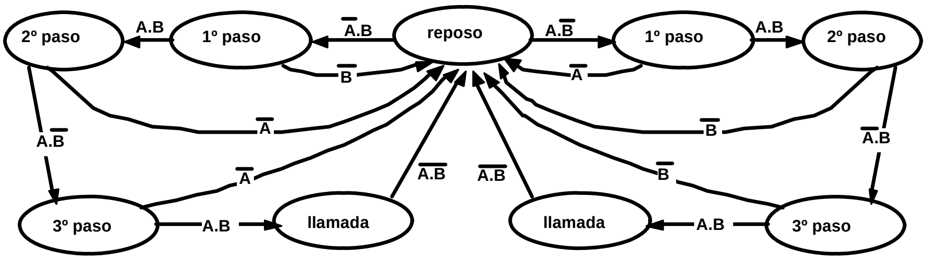

Figura 6. Grafo de estados de activación de un timbre a través de secuencias simétricas de dos pulsadores.

Los dos estados que corresponden a «llamada» en una y otra secuencia se pueden reducir a uno solo; el diagrama resultante puede ser codificado con tres variables de estado, mientras que el grafo sin simplificar necesitaba, al menos, cuatro variables.

Como mero ejercicio operativo se deja a consideración del lector la simplificación del siguiente grafo de estados de tipo Moore con una sola salida **Y** que se activa en los estados **6** y **9** y permanece inactiva en todos los demás; los 9 estados del grafo pueden reducirse a solamente 6.

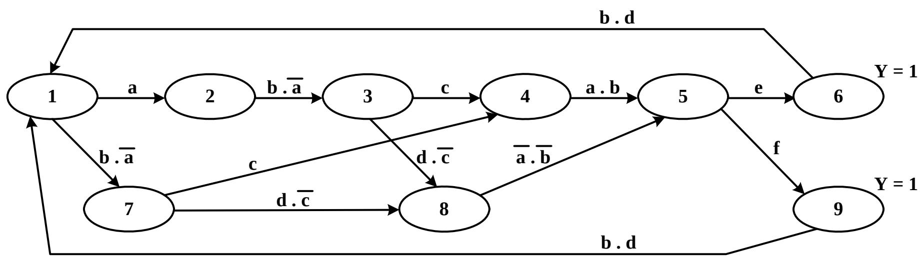

Figura 7. Grafo de estados de simplificable por agrupación de estados que «hacen lo mismo».

En la práctica, en el diseño de un sistema secuencial a partir de unas especificaciones funcionales, apenas suele presentarse este tipo de estados con las mismas salidas; es raro que, al formular el correspondiente grafo de estados, el diseñador incluya dos estados que presenten los mismos efectos, pues generalmente los considerará ya de entrada como el mismo estado.

3. Ejemplos de simplificación: estados distinguibles «desde fuera»

También son agrupables (reducibles a uno solo) aquellos estados que, tanto ellos como las transiciones que desde ellos se producen, son distinguibles por los valores de las variables de entrada.

Consideremos una puerta de garaje con el siguiente funcionamiento: *la puerta ha de abrirse al accionar una llave **Ll** y debe permanecer arriba durante un tiempo prefijado, dado por una temporización **T**, transcurrido el cual la puerta se cierra; si durante la bajada una célula fotoeléctrica situada en la zona inferior **Cf** detecta la presencia de un objeto o persona, la puerta debe volver a subir reiniciando el ciclo de apertura*.

Las entradas al sistema de control de esta puerta serán: la llave **Ll**, los topes superior $\mathbf{T_s}$ e inferior $\mathbf{T_i}$ y la célula fotoeléctrica $\mathbf{C_f}$, así como la salida de un circuito auxiliar de temporización **T**; las salidas serán los movimientos hacia arriba **S** y hacia abajo **B** y el disparo de la temporización **DT**.

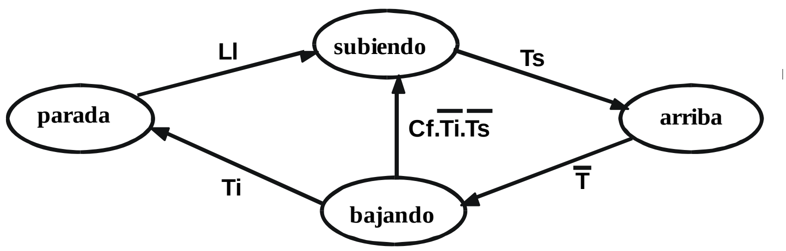

Figura 8. Grafo de estados de control de una puerta de garage con temporizaqción.

En realidad no son necesarios cuatro estados: la necesidad de memoria se reduce a distinguir entre subida y bajada. El estado de reposo y el de bajada pueden difernciarse por el valor de la variable $\mathbf{T_i}$ ($\mathbf{T_i=1}$, reposo; $\mathbf{T_i=0}$, bajada) y, de igual forma, la variable $\mathbf{T_s}$ permite distinguir el estado de subida ($\mathbf{T_s=0}$) y el de temporización ($\mathbf{T_s=1}$).

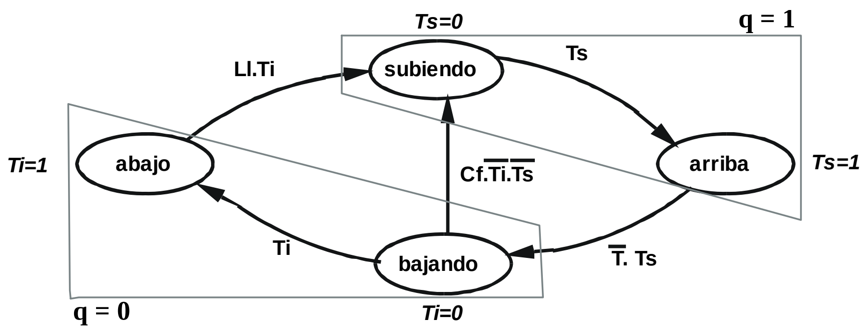

Figura 9. Simplificación del grafo de estados de control de la puerta de garage.

Para agrupar los estados es preciso comprobar que las transiciones que salen de ellos pueden diferenciarse, de forma que se produzcan correctamente:

- es necesario condicionar con T_i la transición que produce Ll para que solamente se realice desde la situación de puerta parada abajo
- la transición que produce C_f se encuentra adecuadamente condicionada con $\overline{T_i}$, de forma que no se efectúa cuando la puerta se encuentra ya parada abajo

- y, también, hay que añadir T_S a la transición determinada por $\overline{T}$ para que no se realice mientras la puerta se encuentra subiendo. De esta forma, las transiciones entre los dos nuevos estados, resultantes de la agrupación de los cuatro iniciales, se producen en las mismas situaciones que en el anterior grafo de cuatro estados.

Basta una variable de estado **q** para diferenciar los dos estados necesarios, que corresponden a la puerta subiendo o parada arriba (**q=1**) y puerta bajando o abajo (**q=0**) y la activación de las salidas vendrá dada por las siguientes funciones:

S (subir) $= q \cdot \overline{T_S}$ DT (disparo de la temporización) $= T_S \uparrow$ (al pasar de **0** a **1**)

B (bajar) $= \overline{q} \cdot \overline{T_i}$

Otro ejemplo semejante: *sea un pequeño carrito motorizado que recorre linealmente el camino entre dos puntos A y B, en los cuales existen sendos contactores «fin de camino» **a** y **b**; al activar un pulsador **P**, el carrito circula desde A hasta B y vuelve nuevamente a A.*

Un primer grafo del comportamiento de dicho carrito puede incluir tres estados: reposo, movimiento hacia B y movimiento hacia A, pero el primero y el último son simplificables, ya que los diferencia la variable exterior **a** (reposo: **a=1**, movimiento hacia A: **a=0**). Basta, pues, una variable de estado **q**, cuyo marcado y borrado se producirán con **P.a** y con **b**, respectivamente; el movimiento hacia B coincidirá con la variable de estado **q** y el movimiento hacia A se producirá con $\overline{q} \cdot \overline{a}$.

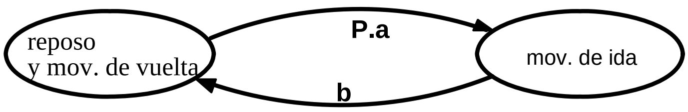

Figura 10. Grafo de estados simplificado de control de movimiento de ida y vuelta.

Supóngase que, *además del carrito anterior que se mueve entre A y B, un segundo carrito circula entre C y D (contactos «fin de carrera»: **c** y **d**), de forma que ambos carritos inician el movimiento desde A y C con el pulsador **P** y el primero en alcanzar el otro extremo B o D, espera a que el otro alcance el suyo, para iniciar juntos el movimiento de vuelta.* Un grafo detallado de este sistema de dos carritos puede incluir siete estados:

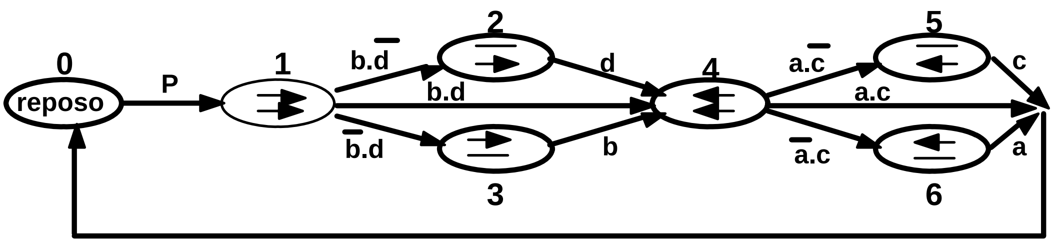

Figura 11. Grafo de estados de control de movimiento de ida y vuelta de dos carritos.

El análisis de la posibilidad de reducir el número de estados puede desarrollarse sistemáticamente considerando:

1 estados agrupables
2 variables que los diferencian
3 posibilidad de diferenciar correctamente las transiciones desde el nuevo estado.

En el anterior grafo de estado son posibles las siguientes agrupaciones:

1.1 estados agrupables: los estados 1, 2 y 3
1.2 variables que los diferencian: **b** y **d** : **00**, **10**, **01**, respectivamente
1.3 posibilidad de diferenciar correctamente las transiciones desde el nuevo estado:
d debe producirse desde el estado 2 (**b=1**) : **d.b**
b debe producirse desde el estado 3 (**d=1**) : **b.d**

el nuevo estado agrupado tendrá una transición de salida **d.b** (cuando se alcancen los dos topes de la derecha), lo cual es correcto.

2.1 estados agrupables: los estados 4, 5, 6 y 0
2.2 variables que los diferencian: **a** y **c** : **00**, **10**, **01**,**11**, respectivamente
2.3 posibilidad de diferenciar correctamente las transiciones desde el nuevo estado:
P debe producirse desde el estado 0 (**a=1 c=1**) : **P.a.c**

el estado agrupado tendrá una transición de salida **P.a.c**, lo cual es correcto.

Obsérvese que solamente es necesario considerar las transiciones que salen desde el nuevo estado agrupado y no las que se producen en el interior del mismo entre los estados que se agrupan.

En este caso, conforme al análisis desarrollado, son suficientes dos estados ya que el resto de las situaciones son distinguibles mediante las variables exteriores.

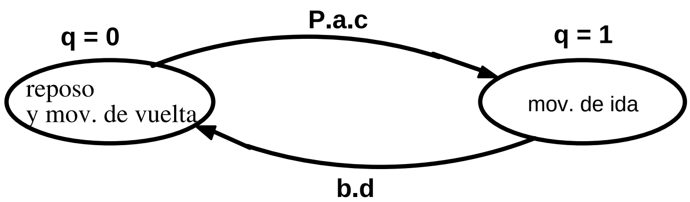

Figura 12. Grafo de estados simplificado de control de movimiento de ida y vuelta de dos carritos.

Las funciones de salida correspondientes a los movimientos de cada carrito serán:

- primer carrito: $\rightarrow$ = mov. AB = $q \cdot \overline{b}$ $\leftarrow$ = mov. BA = $\overline{q} \cdot \overline{a}$
- segundo carrito: $\rightarrow$ = mov. CD = $q \cdot \overline{d}$ $\leftarrow$ = mov. DC = $\overline{q} \cdot \overline{c}$.

Este grafo corresponde a la necesidad de memoria que se limita a distinguir entre dos situaciones: el movimiento hacia B y D del movimiento hacia A y C. Además, el grafo es el mismo (con solamente dos estados) aunque el número de carritos aumente (sirve para cualquier número **n** de carritos); solamente se requiere diferenciar el movimiento de ida del de vuelta: las salidas referidas a cada uno de los carritos dependerán, además del estado (variable q) de la situación de los correspondientes contactos «fin de carrera».

Este grafo presenta un interés particular, por su generalización a **n** mecanismos, y muestra su mayor potencia simplificadora frente a las redes de Petri: en caso de 10 mecanismos bastan 2 estados en el grafo (el no simplificado tendría 2047) mientras que la correspondiente red de Petri precisa de 21 «lugares».

Otro ejemplo de simplificación de estados que se distinguen por los valores de las variables de entradas: *un depósito se llena con una mezcla de cuatro líquidos diferentes, para lo cual dispone de cuatro electroválvulas **A, B, C, D** que controlan la salida de dichos líquidos y de cinco detectores de nivel **n1, n2, n3, n4, n5** siendo **n1** el inferior y **n5** el de llenado máximo. Solamente cuando el nivel del depósito desciende por debajo del mínimo **n1** se produce un ciclo de llenado: primero con el líquido A hasta el nivel **n2**, luego el líquido B hasta el nivel **n3**, el líquido C hasta **n4** y, finalmente, D hasta completar el depósito **n5**.*

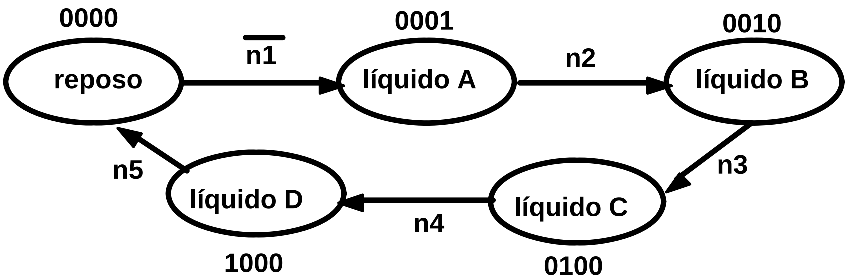

Figura 13. Grafo de estados de control de electroválvulas para mezcla de cuatro líquidos.

En el grafo anterior se utiliza para los estados el código de «un solo uno» con cuatro variables de estado **q4 q3 q2 q1**; de esta forma cada variable de estado coincide con una de las variables de salida (**q1** = llenado con líquido **A**; **q2** = líquido **B**; **q3** = líquido **C**; **q4** = líquido **D**). Los cuatro estados correspondientes a la salida de líquidos pueden diferenciarse por las entradas detectoras de nivel (n2, n3, n4, n5), de forma que el grafo se reduce a dos estados.

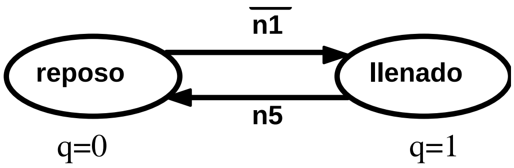

Figura 14. Grafo de estados simplificado de control de electroválvulas para mezcla de cuatro líquidos.

No es necesario añadir a la transición con **n5** más condiciones para que solamente se produzca desde la situación de *liquido D*, ya que solamente desde ella se puede alcanzar dicho detector de nivel. Se necesita solamente una variable de estado **q**, pero las funciones de activación de las salidas resultan más complejas, pues dependen de las entradas, de la información que aportan los detectores de nivel:

electroválvula A	$= q \cdot \overline{n_2}$	*líquido A hasta que se alcanza el nivel n2*
electroválvula B	$= q \cdot n_2 \cdot \overline{n_3}$	*líquido B desde el nivel n2 hasta el nivel n3*
electroválvula C	$= q \cdot n_3 \cdot \overline{n_4}$	*líquido C desde el nivel n3 hasta el nivel n4*
electroválvula D	$= q \cdot n_4$	*líquido D por encima del nivel n4*

Aunque no siempre, la agrupación de estados diferenciables por las entradas suele pasar de un autómata de Moore a otro de Mealy, en que las salidas dependerán de las entradas. Es lo que sucede en los cuatro ejemplos considerados anteriormente.

Un autómata de Mealy presenta siempre menos estados que el correspondiente autómata de Moore, ya que éste tiene que diferenciar con estados distintos aquellas situaciones que corresponden a un mismo estado de Mealy pero con diferentes vectores de salida. Tales vectores de salida, y los correspondientes estados de Moore, son distinguibles por vectores de entrada diferentes.

En ocasiones no es posible agrupar estados que se diferencian por variables de entrada, debido a que las transiciones desde ellos no pueden diferenciarse adecuadamente. Por ejemplo, *una sirena que se activa y se desconecta con un mismo pulsador **a*** (funcionamiento análogo al de una lámpara de estudio de tipo «flexo»).

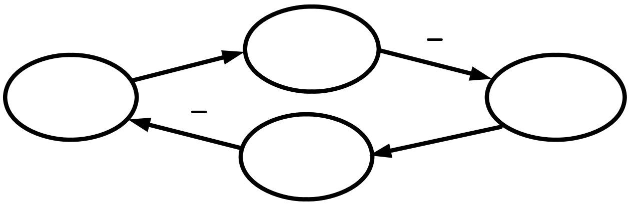

Figura 15. Grafo de estados de control de la sirena.

Los estados 1 y 2 (y, también, 3 y 0) se diferencian por **a** pero no pueden agruparse pues la condición de salto **a** no puede ser multiplicada por «**a negada**» (a = 0, estado 2). Sucede que la variable que las diferencia e, precisamente, la que determina la transición de salida desde uno de ellos.

Es conveniente insistir en la importancia que tiene el diferenciar adecuadamente las transiciones desde el estado agrupado. Volviendo al caso de *dos pulsadores que activan un timbre a través de una secuencia **A**, **A y B**, **B**, **A y B***, cuyo grafo puede ser el siguiente.

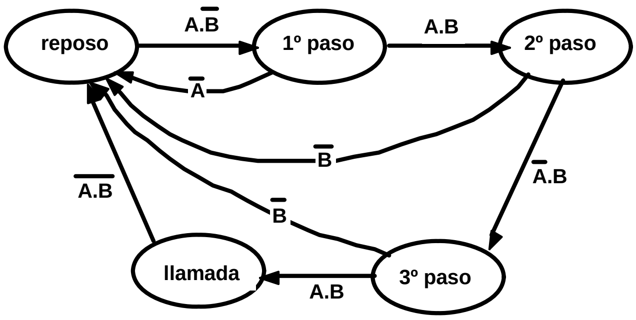

Figura 16. Grafo de estados de activación de un timbre a través de una secuencia de dos pulsadores.

Los estados correspondientes a *1º paso* y a *2º paso* pueden distinguirse a través de la variable **B** (**B=0** y **B=1**, respectivamente) pero al agruparlos no es posible diferenciar la transición producida por $\overline{B}$ de forma que solamente se efectúe desde el *2º paso* (**B=1**: $\overline{B}$. B transición imposible).

También los estados de *reposo* y *1º paso* pueden distinguirse a través de la variable **A** (**A=0** *reposo* y **A=1** *1º paso*) pero al agruparlos no puede diferenciarse la transición producida por A . B de manera que sólo se efectúe desde el *1º paso* (por ejemplo, podríamos pulsar B y luego A . B, con lo cual la transición al *2º paso* se realizaría directamente desde el *reposo,* lo cual es incorrecto).

Ello se debe a que, precisamente, la misma variable exterior que distingue los estados es la que produce la transición. En este caso la necesidad de memoria se extiende a todos los pasos de la secuencia (**A**, **A y B**, **B**, **A y B**); al agrupar dos de ellos se reduce la longitud de la secuencia (si acaso podría prescindirse del último de los estados, el de llamada, y activar el timbre directamente desde el estado que corresponde a *3º paso* cuando la entrada sea **A.B**).

4. Ejemplo con los dos tipos de simplificación de estados

*Un determinado mecanismo se mueve a lo largo de un riel entre dos posiciones A y B que se detectan mediante sendos sensores **a** y **b** y se controla mediante un pulsador **P** de la siguiente manera:*
*- cuando el mecanismo se encuentra en A y se activa **P**, se inicia el movimiento hacia B al soltar el pulsador **P***
*- de la misma forma, cuando se encuentra en B se inicia el movimiento hacia A al dejar libre el pulsador **P** (una vez activado)*
*- cuando el mecanismo se encuentra entre A y B, si se pulsa **P** el móvil se detiene y al soltar **P** continúa su movimiento anterior.*

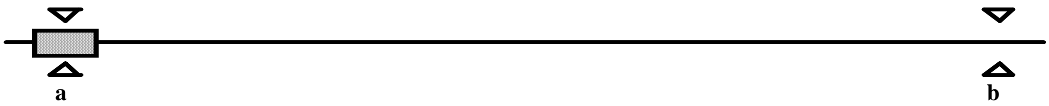

Figura 17. Mecanismo con movimiento de ida y vuelta entre dos detectores fin de camino.

Un posible grafo de estados, en forma de autómata de Moore con detalle de todas las situaciones posibles, en el que las salidas corresponden directamente a los estados de *ida* y *vuelta*, es el siguiente.

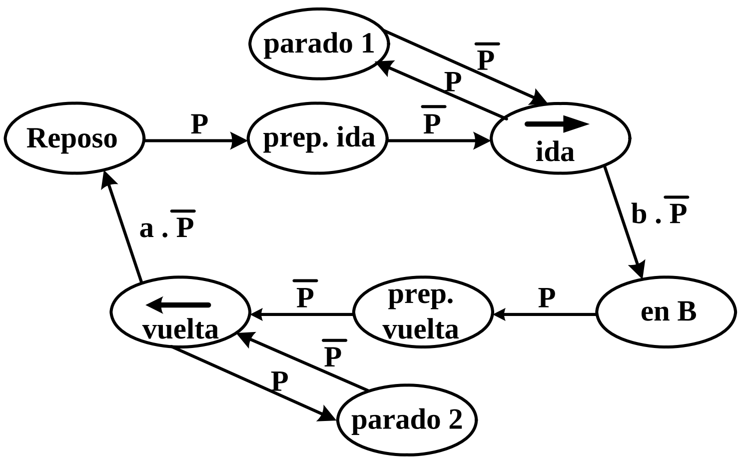

Figura 18. Grafo de estados «completo» del mecanismo con movimiento de ida y vuelta.

Pero los estados *preparado ida* y *parado 1* son idénticos ya que tienen los mismos efectos (salida nula y transición hacia el estado *ida* con **P** negado) y lo mismo sucede con los estados *preparado vuelta* y *parado 2* (salida nula y transición hacia el estado *vuelta* con **P** negado); de forma que ambas parejas de estados pueden agruparse según el grafo siguiente, que continúa siendo de Moore (salidas en correspondencia con los estados de *ida* y *vuelta*):

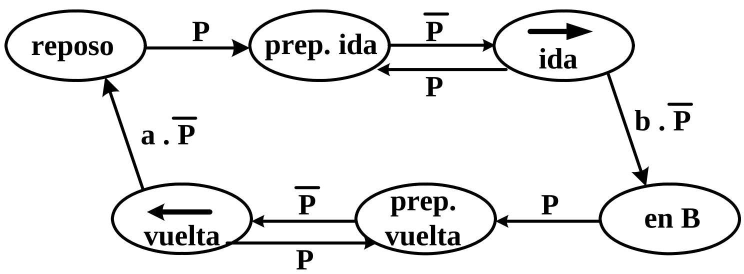

Figura 19. Primera simplificación del grafo de estados del mecanismo con movimiento de ida y vuelta.

Ahora bien, los estados *preparado ida* e *ida* pueden diferenciarse por el valor de la entrada **P** (**P** = **1** *preparado ida*, **P** = **0** *ida*) y lo mismo sucede con los estados *preparado vuelta* y *vuelta* (**P** = **1** *preparado vuelta*, **P** = **0** *vuelta*); lo cual permite agruparlos.

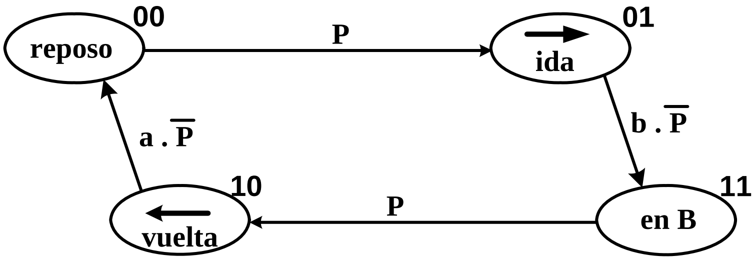

Figura 20. Segunda simplificación del grafo de estados del mecanismo con movimiento de ida y vuelta.

En este caso, el autómata es de Mealy y las funciones de activación de las salidas son: ida = $\overline{q_2} \,.\, q_1 \,.\, \overline{P}$ vuelta = $q_2 \,.\, \overline{q_1} \,.\, \overline{P}$.

Pero también, en este último grafo, los estados *ida* y *en B* pueden diferenciarse por el valor de una entrada (**b** = **0** *ida*, **b** = **1** *en B*); para agrupar ambos estados es necesario multiplicar la transición **P** que sale del nuevo estado por **b**, a fin de que se produzca desde la situación *en B*. Igualmente sucede con los estados *vuelta* y *reposo* (**P** = **0** *vuelta*, **P** = **1** *reposo*); ambos pueden agruparse multiplicando la transición **P** que sale del nuevo estado por **a**, para que se produzca desde *Reposo*.

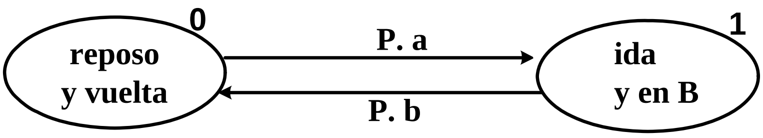

Figura 21. Tercera simplificación del grafo de estados del mecanismo con movimiento de ida y vuelta.

Las funciones de activación de las salidas son, en este caso de máxima simplifiación: ida = $q \,.\, \overline{P} \,.\, \overline{b}$ vuelta = $\overline{q} \,.\, \overline{P} \,.\, \overline{a}$.

Referencias

[1] T. Pollán *Electrónica Digital II. Sistemas secuenciales.* Prensas Universitarias de Zaragoza. Colección Textos docentes nº 104. Universidad de Zaragoza 2004; en particular el capítulo 12.
[2] Página web: http://www.unizar.es/euitiz/digital.htm
[3] Página web: http://www.unizar.es/euitiz/areas/aretecel/docencia/digitel/digitelib.htm , capítulo 12