12 INTRODUCCIÓN A LOS SISTEMAS SECUENCIALES II: DISEÑO

El proceso de diseño de un sistema secuencial parte de la delimitación de los estados necesarios para describirlo, para lo cual resultan sumamente útiles los grafos de estados. La reducción, cuando es posible, del número de estados, contribuye a simplificar el grafo y, en consecuencia, las funciones de evolución del estado. Dos son las situaciones que permiten la agrupación de varios estados en uno solo: cuando los estados tienen los mismos efectos (incluyendo en ellos, tanto las salidas como las transiciones que se producen desde ellos) y cuando no es preciso distinguir los estados entre sí, ya que se diferencian por los valores de algunas variables de entrada.

Una vez establecidos los estados de un sistema, el siguiente paso es «dar nombre» a cada estado: asignarle una palabra binaria. Luego, cada uno de sus dígitos (cada variable de estado) puede almacenarse sobre un biestable, de forma que el diseño secuencial consiste en obtener las funciones de marcado y de borrado de los mismos (biestables RS), junto con las funciones que producen el vector de salida. El sistema secuencial queda distribuido en tres partes: los biestables que contienen las variables de estado, las funciones de activación de dichos biestables y las funciones de activación de las salidas.

El diseño secuencial equivale a un recorrido a través de los diversos niveles de descripción del sistema: a partir de las especificaciones o requisitos se plantea un grafo de estados, desde el cual se obtienen las funciones de evolución de las variables de estado (marcado y borrado de sus biestables) y las de activación de las salidas del sistema para llegar al correspondiente circuito digital de biestables y puertas lógicas.

La síntesis de las funciones de marcado y borrado de los «biestables de estado» puede hacerse a través de la correspondiente tabla de verdad (entradas, estado → nuevo estado), según el método habitual de construcción de funciones booleanas; o bien, directamente desde el grafo de estados, identificando las transiciones que modifican el valor de cada variable de estado. La elaboración de la tabla funcional obliga a considerar todas y cada una de las situaciones posibles (para cada estado, cada uno de los vectores de entrada), muchas de las cuales no quedan reflejadas, en forma expresa, en el grafo de estado; además facilita la simplificación de las funciones.

Este capítulo, además de tratar en detalle la simplificación de estados y exponer el proceso de diseño de sistemas secuenciales, incluye un variado y significativo conjunto de ejemplos de diseño de los mismos.

12.1. Simplificación de estados con los mismos efectos

La representación de la evolución del estado del sistema sobre un grafo de estados sirve para determinar los estados necesarios; una vez establecido un grafo de estados, conviene considerar la posibilidad de simplificarlo, disminuyendo el número de estados que contiene.

La reducción del número de estados de un sistema secuencial, cuando tal cosa es posible, supone una simplificación de las funciones booleanas de evolución del estado, la cual resulta particularmente importante cuando permite disminuir el número de variables de estado (ello sucede siempre en codificación con «un solo uno» pero tiene aún mayor interés en código Gray y demás códigos de longitud mínima).

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:

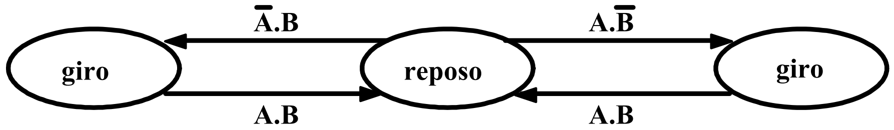
a) son agrupables aquellos estados que no precisan diferenciarse entre sí, por tener las mismas salidas, es decir, los mismos vectores de salida y las mismas transiciones desde ellos;

b) también son agrupables aquellos estados que, tanto ellos como las transiciones que se producen desde ellos, pueden diferenciarse mediante variables exteriores, es decir, mediante los valores de las variables de entrada.

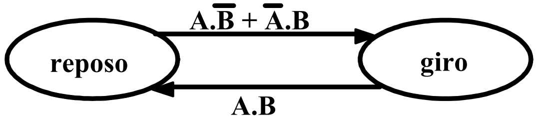
Los estados se refieren a situaciones (secuencias de vectores de entrada) que el sistema necesita conservar en su memoria en forma diferenciada (distinguiendo una situación de otra). Obviamente, no es preciso diferenciar aquellas situaciones que producen los mismos efectos: los mismos vectores de salida y las mismas transiciones. Tampoco es necesario distinguir internamente aquellas situaciones que se diferencian entre sí por el valor de alguna de las variables de entrada (o de varias de ellas). En el primer caso, se evita diferenciar situaciones que son idénticas por serlo sus efectos; en el segundo, puede prescindirse de distinguir en la memoria del sistema aquello que puede diferenciarse externamente.

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.

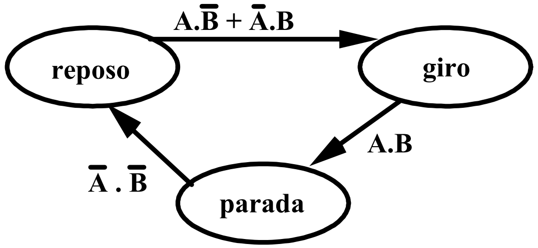
Consideremos un ejemplo muy simple*: un motor con giro en sentido único controlado a través de dos pulsadores; activando uno cualquiera de ellos el motor se pone a girar y se para al presionar ambos pulsadores a la vez.* Un posible grafo de estados sería el siguiente:



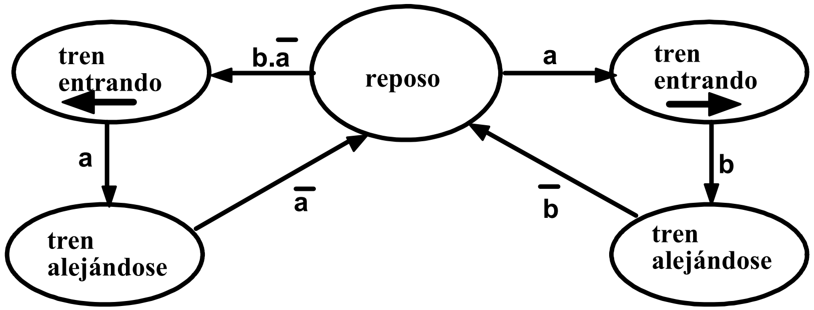
Ahora bien, los dos estados de *giro* tienen el mismo vector de salida (el motor girando en el mismo sentido) y la misma transición de salida (con A.B pasan al reposo); por ello, equivalen a un solo estado:



Se ha propuesto este primer ejemplo por su sencillez, pero presenta un problema sobre el cual conviene razonar: 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 (los circuitos electrónicos son sensibles a diferencias de nanosegundos) y, entonces, se produce la condición de activación y el motor se pone a girar. El grafo adecuado para evitar esto requiere tres estados:

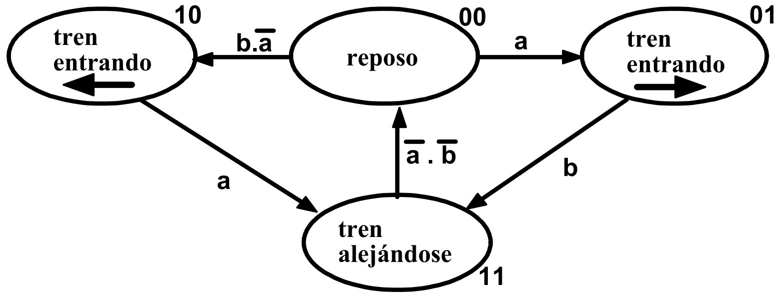


Un segundo ejemplo: *semáforo de aviso de paso de tren en un cruce de vía única bidireccional con un camino; la vía posee, a ambos lados del cruce y a una distancia adecuadamente grande, sendos detectores de paso de tren **a** y **b**; los trenes circulan por ella en ambas direcciones y se desea que 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.*



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.

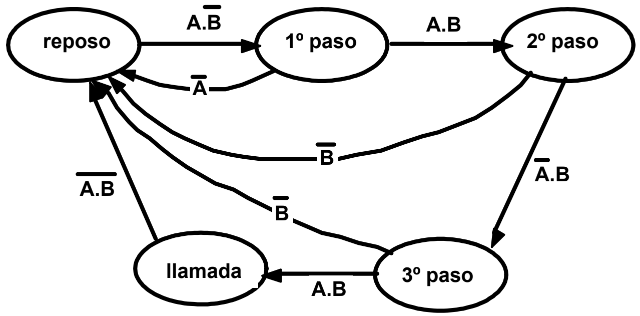
El grafo anterior, tal como está representado no es simplificable. Pero podemos modificar la condición booleana de las transiciones que salen de ambos estados inferiores (*tren alejándose*), de forma que ambas se produzcan con $\overline{a}\,.\,\overline{b}$ (en el momento en que el tren abandona el cruce ambos detectores estarán a **0**) y, entonces, ambos estados son reducibles a uno solo.



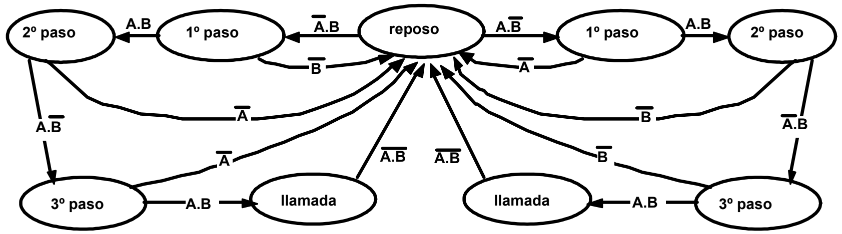
Aunque ambos grafos difieren en un solo estado, el primero de ellos necesita, al menos, tres variables de estado, mientras que el segundo puede codificarse con dos variables de estado.

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.

Consideremos un ejemplo que utilizaremos posteriormente para describir las formas de codificar estados: *sea un timbre que 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 siguiente:*



Supongamos que, para evitar el tener que acordarse por cuál de los pulsadores hay que iniciar la secuencia, se admiten las dos secuencias simétricas posibles: (**A**, **A y B**, **B**, **A y B**) o, también, (**B**, **A y B**, **A**, **A y B**).



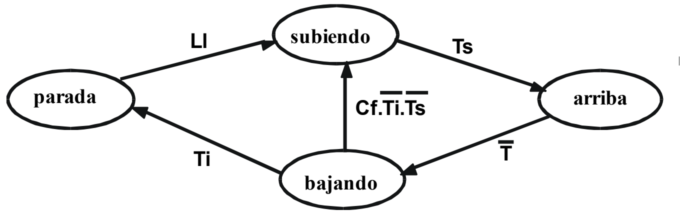
Los dos estados que corresponden a «llamada» en una y otra secuencia se pueden reducir a uno solo; el diagrama resultante puede ser codificado con tres variables de estado, mientras que el grafo sin simplificar necesitaba, al menos, cuatro variables.

12.2. Agrupación de estados que se distinguen por variables de entrada

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 de funcionamiento un poco más complejo que el detallado en el apartado 11.1: *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 **Cf**, así como la salida de un circuito auxiliar de temporización **T**. Las salidas de dicho control serán los movimientos hacia arriba **S** y hacia abajo **B** y el disparo de la temporización **DT**.

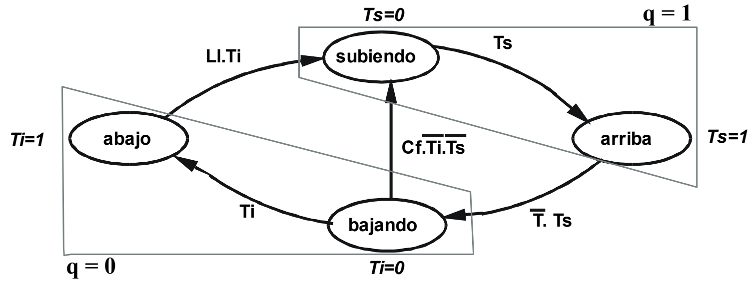


Los cuatro estados de este grafo requieren para su codificación, al menos, dos variables de estado $\mathbf{q_2\,q_1}$.

Pero en realidad no son necesarios cuatro estados: la necesidad de memoria se reduce a distinguir entre subida y bajada, pudiendo diferenciarse los otros dos estados mediante variables exteriores. El estado de reposo y el de bajada pueden distinguirse por el valor de la variable $\mathbf{T_i}$ ($\mathbf{T_i=1}$, reposo; $\mathbf{T_i=0}$, bajada); de igual forma la variable $\mathbf{T_s}$ permite diferenciar el estado de subida ($\mathbf{T_s=0}$) y el de temporización ($\mathbf{T_s=1}$).

Una vez agrupados tales 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**):

q: $S = LL.T_i + Cf.\overline{T_i}.\overline{T_s}$ $R = \overline{T}.T_s$

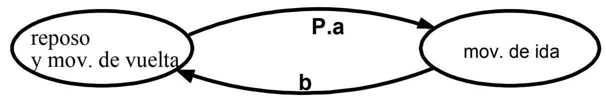
La activación de las salidas vendrá dada por las siguientes funciones:

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

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

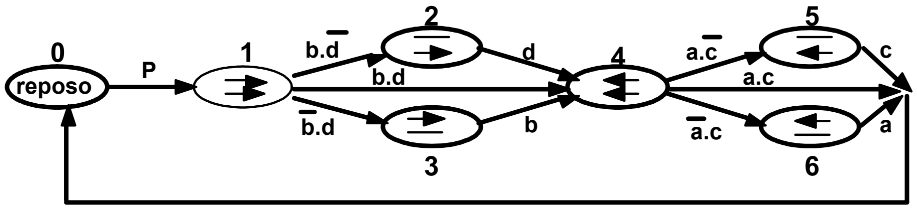
Otro ejemplo semejante: *sea un pequeño carrito motorizado que recorre linealmente el camino entre dos puntos A y B, en los cuales existen sendos contactores «fin de camino» **a** y **b**; al activar un pulsador **P**, el carrito circula desde A hasta B y vuelve nuevamente a A.*

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 $\overline{q}\,.\,\overline{a}$.



*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:



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:

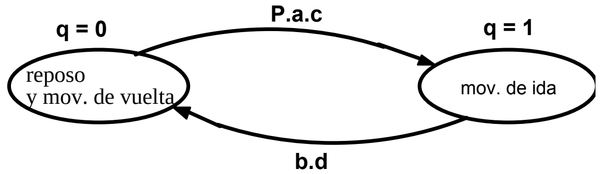
P debe producirse desde el estado 0 (**a=1 c=1**) : **P.a.c**

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.

Resulta un grafo del tipo siguiente:



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.

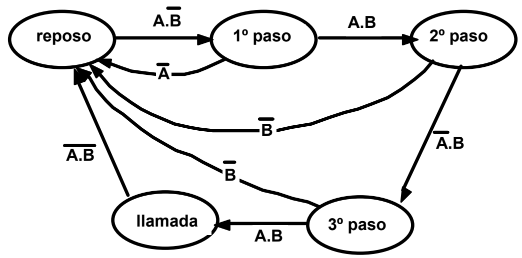
Las funciones de salida correspondientes a los movimientos de cada carrito serán:

- 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 de dos estados, en forma de autómata de Mealy, sirve también 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».

Otro ejemplo de simplificación de estados que se distinguen por los valores de las variables de entradas es el considerado en el último apartado del capítulo anterior (apartado 11.4, páginas 30 – 32, depósito que se llena con una mezcla de cuatro líquidos diferentes): su grafo como autómata de Moore presenta cinco estados, que se reducen a dos al tratarlo como autómata de Mealy, debido a que cuatro de dichos estados pueden diferenciarse por las entradas detectoras de nivel (n_2, n_3, n_4, n_5).

Es conveniente insistir en la importancia que tiene el diferenciar adecuadamente las transiciones desde el estado agrupado. Volviendo al grafo de los dos pulsadores que activan un timbre a través de una secuencia **A**, **A y B**, **B**, **A y B**, (apartado anterior, página 36)



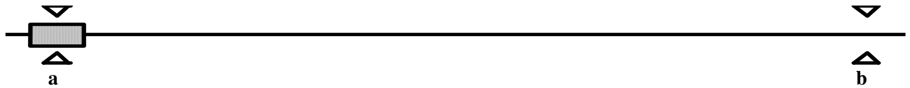
los estados correspondientes a *1º paso* y a *2º paso* pueden distinguirse a través de la variable **B** (**B=0** y **B=1**, respectivamente) pero al agruparlos no es posible diferenciar la transición producida por $\overline{B}$ de forma que solamente se efectúe desde el *2º paso* (**B=1**: $\overline{B} \cdot B$ 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**, tal como veremos en el próximo apartado 12.3.2).

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**:*



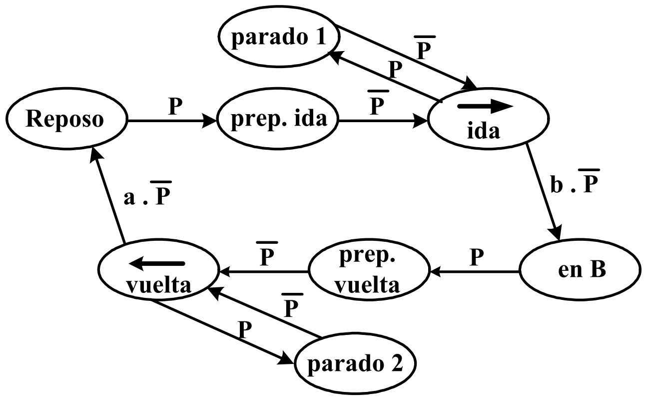
*el movimiento se controla mediante un pulsador **P** de la siguiente manera:*

*- cuando el mecanismo se encuentra en A y se activa **P**, se inicia el movimiento hacia B al soltar el pulsador **P***

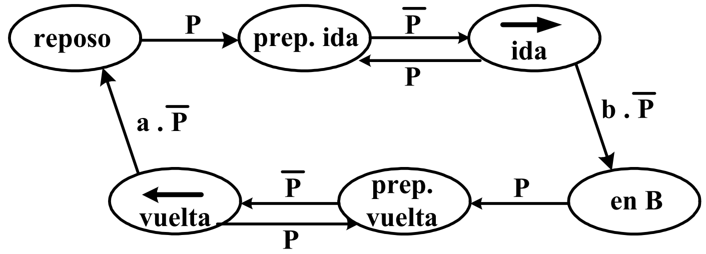
*- de la misma forma, cuando se encuentra en B se inicia el movimiento hacia A al dejar libre el pulsador **P** (una vez activado)*

*- 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.*

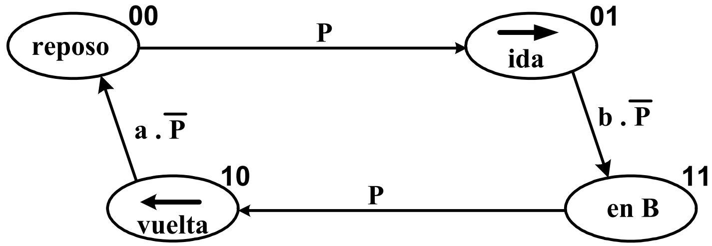
Un posible grafo de estados, en forma de autómata de Moore con detalle de todas las situaciones posibles, es el siguiente, en el que las salidas corresponden directamente a los estados de *ida* y *vuelta*, respectivamente.



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*):

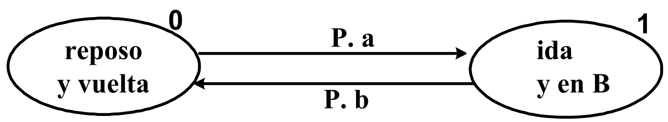


Ahora bien, los estados *preparado ida* e *ida* pueden diferenciarse por el valor de la entrada **P** (**P** = **1** *preparado ida*, **P** = **0** *ida*) y lo mismo sucede con los estados *preparado vuelta* y *vuelta* (**P** = **1** *preparado vuelta*, **P** = **0** *vuelta*); lo cual permite agrupar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



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

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*. Resulta un grafo de Mealy, con solo dos estados:



Las funciones de activación de las salidas son:

$$\text{ida} = q\,.\,\overline{P}\,.\,\overline{b} \qquad\qquad \text{vuelta} = \overline{q}\,.\,\overline{P}\,.\,\overline{a}$$

y la evolución de la variable de estado corresponde a las siguientes funciones de marcado y borrado:

$$S = P\,.\,a \qquad\qquad R = P\,.\,b\ .$$

12.3. Codificación de los estados

El grafo de estados establece los estados necesarios; conocidos los estados del sistema secuencial, el siguiente paso será asignar un código binario (un conjunto de valores de las variables de estado) a cada uno de los estados.

La codificación de los estados, es decir, la asignación de vectores de estado a los diversos estados, es un proceso arbitrario, con la única limitación de que a estados diferentes deben corresponder códigos binarios (vectores de estado) distintos. En principio el diseñador puede asignar libremente a los diversos estados las palabras binarias que desee o que considere más adecuadas.

Ahora bien, entre las múltiples codificaciones posibles de los estados, existen dos de particular interés:

a) asignar a cada estado (excepto al estado inicial) una variable de estado, de forma que cada estado se corresponda biunívocamente con una de las variables de estado, resultando un código de «un solo uno» (cada vector de estado contiene un solo **1**);

b) utilizar para numerar los estados el código Gray, de forma que dos estados consecutivos difieran en una sola variable de estado (en cada transición se produce «un solo cambio» en las variables de estado).

La codificación con «un solo uno» (cada vector de estado contiene un solo **1**) supone utilizar un biestable para cada estado (excepto para el estado inicial que será el **000…**); en cambio, para código Gray en cada transición se produce «un solo cambio» en las variables de estado y, consiguientemente, solamente conmuta uno de los biestables.

El código de «un solo uno» resulta muy sencillo de manejar si el número de estados es reducido, mientras que el código Gray confiere una gran seguridad al funcionamiento del sistema secuencial, evitando evoluciones erróneas. Además, ambas formas de codificación de estados reducen la lógica necesaria para la activación de los biestables (reducen las funciones booleanas de evolución de las variables de estado).

Aplicaremos ambas formas de codificar estados al ejemplo del timbre con dos pulsadores **A** y **B** que suena al aplicar la secuencia: **A**, **A y B**, **B**, **A y B**.

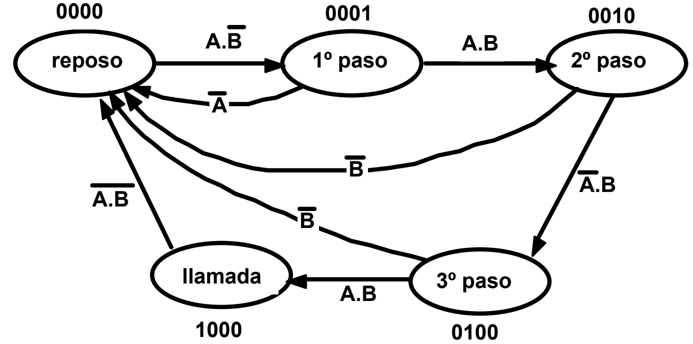
12.3.1. Codificación con un solo uno

El estado inicial o de reposo se codifica con el valor **000…** y para el resto de los estados se hace coincidir una variable de estado con cada uno de ellos, de forma que el valor de dicha variable sea **1** para tal estado y **0** para el resto de ellos.

De esta forma se requiere un número de variables igual al número de estados menos uno; en cambio, se consigue que las funciones booleanas, tanto las de evolución de estado como las de salida, sean relativamente simples y fáciles de deducir directamente del grafo de estados.

Con esta codificación el ejemplo relativo al timbre con dos pulsadores requiere cuatro variables de estado $\mathbf{q_4\,q_3\,q_2\,q_1}$ para representar sus cinco estados:

0000 reposo, **0001** primer paso, **0010** segundo paso, **0100** tercer paso, **0001** llamada.



La activación de la salida (sonido del timbre) coincide con el último estado y, por tanto, con una de las variables de estado $\mathbf{q_4}$. Para configurar el sistema secuencial con biestables RS, se asigna un biestable a cada variable de estado y las condiciones de marcado y borrado pueden obtenerse directamente del grafo de estados.

Resulta útil introducir la variable $\mathbf{q_0} = \boldsymbol{Nor}(\mathbf{q_4,q_3,q_2,q_1})$, que corresponde al estado inicial **0000**, es decir, adopta el valor **1** cuando $\mathbf{q_4=q_3=q_2=q_1=0}$; de esta forma, todos los estados, incluido el inicial, se encuentran en correspondencia biunívoca con una variable de estado.

- evolución del estado:

$\mathbf{q_1}$: $\quad S = q_0\,.\,A\,.\,\overline{B} \qquad R = q_1.(\overline{A} + A.B) = q_1.(\overline{A} + B)$

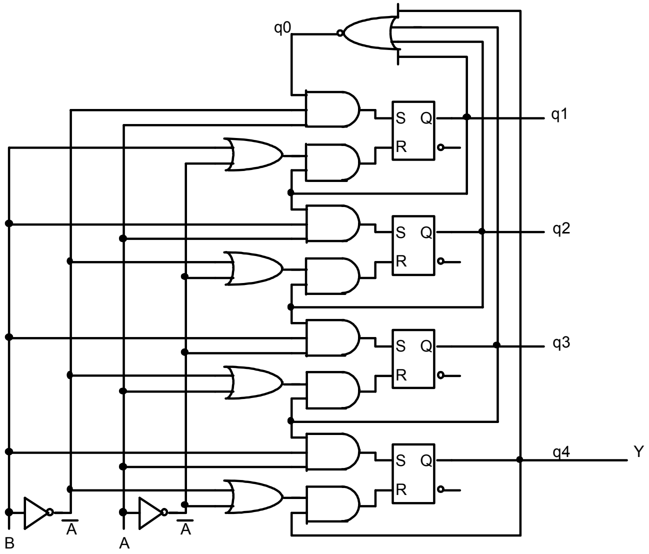
$\mathbf{q_2}$: $\quad S = q_1\,.\,A\,.\,B \qquad R = q_2.(\overline{B} + \overline{A}.B) = q_2.(\overline{A} + \overline{B})$

$\mathbf{q_3}$: $\quad S = q_2\,.\,\overline{A}\,.\,B \qquad R = q_3.(\overline{B} + A.B) = q_3.(A + \overline{B})$

$\mathbf{q_4}$: $\quad S = q_3\,.\,A\,.\,B \qquad R = q_4.(\overline{A.B})$

- variable de salida: $\quad$ **timbre =** $\mathbf{q_4}$

La siguiente figura representa el circuito correspondiente a dichas funciones:



Obsérvese que las condiciones de marcado deben incluir la información sobre el estado anterior desde el que se produce el marcado; precisamente dicha información viene dada por la variable que corresponde al estado anterior y en el caso del estado inicial por el producto de las variables de estado negadas: ***Nor*(q4,q3,q2,q1) = q0**.

Lo mismo sucede con las condiciones de borrado que, en general, deben incluir la información sobre el propio estado desde el que se produce el borrado; dicha información se reduce a la variable propia del estado, que es la que pasa a **0**.

Se debe poner particular atención a las transiciones entre estados, considerando la posibilidad de que, por error, se marquen dos o más variables de estado o que la variable de estado anterior se borre tan rápidamente que no llegue a marcarse el estado siguiente.

Así puede suceder que al pasar de un estado al siguiente (por ejemplo de **q1** a **q2**) el primero de los estados se borre tan rápidamente (**q1** con **A.B**) que no llegue a marcarse el siguiente (**q2**); esto puede corregirse condicionando el borrado de una variable al hecho de que la variable de estado siguiente se haya marcado (para **q1**, $R = \overline{A} + B.q_2$).

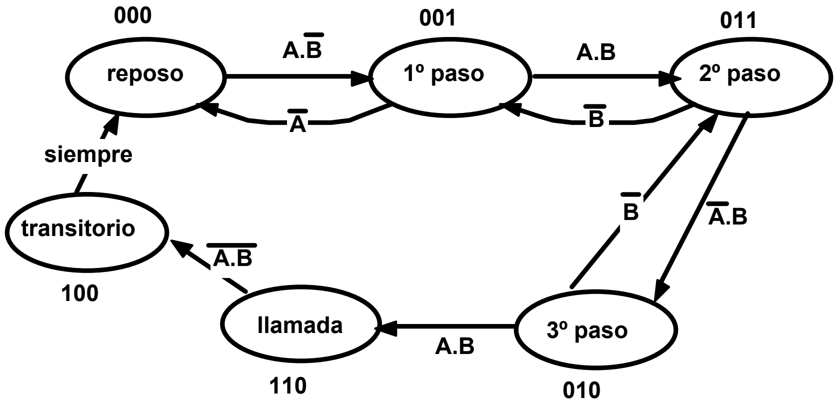
Ahora bien, esta forma de asegurar que el borrado de una variable no se produce hasta que no se marca la siguiente aumenta el riesgo de que se marquen dos variables a la vez. Todos estos problemas son propios de los sistemas asíncronos y no es posible resolverlos globalmente, sino que, en cada caso deben analizarse los riesgos de que tales problemas se produzcan; la forma de evitar todo este tipo de posibles errores pasa por codificar n código Gray o, aún mejor, por efectuar un diseño síncrono (que será estudiado en el capítulo 14).

12.3.2. Codificación con un solo cambio

La utilización del código Gray para numerar los estados permite que dos estados consecutivos se diferencien en el valor de una sola variable de estado, es decir, que en cada transición cambie solamente una variable de estado.

Con esta codificación el ejemplo anterior requiere tres variables de estado **q3 q2 q1** para representar sus cinco estados:

000 reposo, **001** primer paso, **0011** segundo paso, **0010** tercer paso, **110** llamada.



Ha sido preciso reorganizar las transiciones del grafo e introducir un estado adicional para asegurar que en todas las transiciones solamente cambia de valor una de las variables de estado. Puede comprobarse que el anterior grafo de estados es funcionalmente correcto y que cualquiera de las transiciones expresadas en el mismo modifica un sola variable de estados.

Asignando un biestable a cada una de las variables de estado, para obtener las condiciones de marcado y borrado ha de tenerse en cuenta el subconjunto de estados en que la correspondiente variable se encuentra a **1** y considerar las transiciones que conducen y que abandonan, respectivamente, dicho subconjunto.

Tanto en las condiciones de marcado como en las de borrado debe incluirse la información sobre el estado anterior desde el que se produce dicho marcado o borrado.

- evolución del estado:

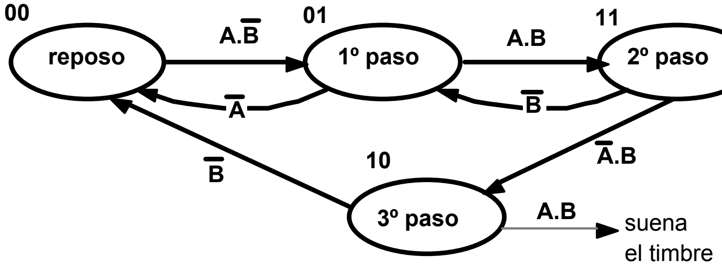
q1: $S = \overline{q_3}.\overline{q_2}.\overline{q_1}.A.\overline{B} + \overline{q_3}.q_2.\overline{q_1}.\overline{B}$ $\quad$ $R = \overline{q_3}.\overline{q_2}.q_1.\overline{A} + \overline{q_3}.q_2.q_1.\overline{A}.B$

q2: $S = \overline{q_3}.\overline{q_2}.q_1.A.B$ $\quad$ $R = \overline{q_3}.q_2.q_1.\overline{B} + q_3.q_2.\overline{q_1}.\overline{A.B}$

q3: $S = \overline{q_3}.q_2.\overline{q_1}.A.B$ $\quad$ $R = q_3.\overline{q_2}.\overline{q_1}$

- variable de salida: **timbre = q3**

Hubiera sido correcto utilizar para este mismo sistema secuencial un grafo más reducido, conforme a la siguiente figura, ya que no es estrictamente necesario memorizar el último estado de la secuencia; basta con que se active el timbre desde el estado correspondiente a *3º paso* cuando la entrada sea **A.B**,



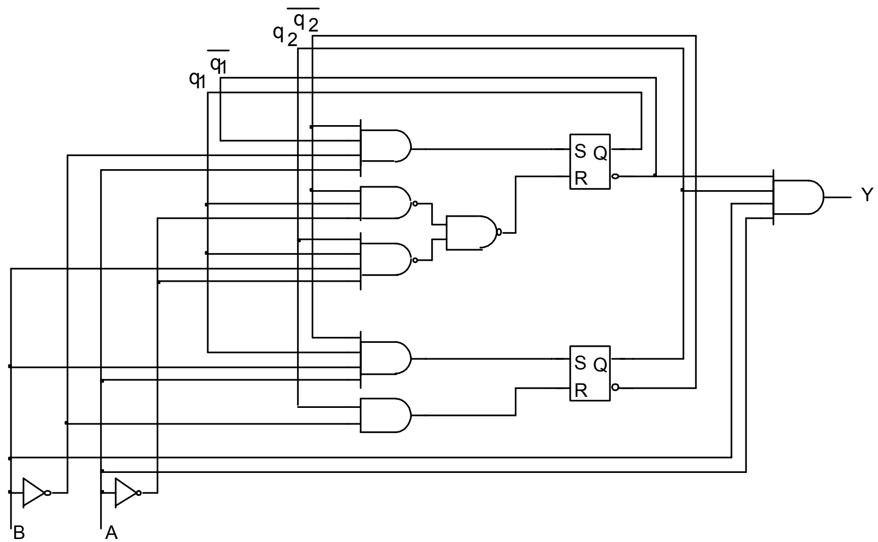
Obsérvese que en el estado final, si después de activar el timbre se deja de pulsar **A** y se mantiene pulsado **B**, permanece el mismo estado y puede activarse de nuevo el timbre al pulsar **A**; en esto se diferencia este grafo del anterior, en el cual, al dejar de activar uno de los pulsadores **A** o **B**, el estado de llamada pasa directamente al estado de reposo.

- evolución del estado:

q_1: $S = \overline{q_2}.\overline{q_1}.A.\overline{B}$ $\qquad R = \overline{q_2}.q_1.\overline{A} + q_2.q_1.\overline{A}.B$

q_2: $S = \overline{q_2}.q_1.A.B$ $\qquad R = q_2.\overline{q_1}.\overline{B} + q_2.q_1.\overline{B} = q_2.\overline{B}$

- variable de salida: $\text{timbre} = q_2.\overline{q_1}.A.B$.



Cuando la codificación se ajusta por completo a un código Gray (un solo cambio en cada transición) se evita en gran medida la posibilidad de evoluciones erróneas debidas a diferencia en los tiempos de propagación de las funciones que marcan o borran las diversas variables.

Si el código no es Gray, en el paso de un estado a otro en que cambie el valor de más de una variable de estado puede suceder que la modificación de una de ellas sea más rápida que la del resto y se produzca un estado intermedio a partir del cual el sistema desarrolle una evolución errónea. Por ejemplo, en un sistema con dos variables de estado **q_2 q_1**, en el cual para **b=1** se pase del estado **01** al **10**, siendo la variable **q_1** más rápida en modificarse, se pasará primero al estado **00**; puede suceder que dicho estado intermedio **00** no deba cambiar con **b=1** y la transición a **10** no se complete, dando lugar a una transición errónea del **01** al **00**.

La codificación en código Gray requiere, en ocasiones, modificar el grafo de estados, añadiendo estados auxiliares para asegurar que en todas las transiciones se modifica una sola variable de estado; ello siempre es posible pero, en algunos casos, resulta muy complejo y supone un aumento significativo del número de estados. La alternativa, en tal caso, pasa por el sincronismo como mecanismo conceptual que proporciona seguridad funcional (al evitar la formación de estados intermedios).

12.4. Diseño de circuitos secuenciales con biestables RS

El proceso de síntesis o «construcción digital» de una función parte del enunciado o especificaciones de la misma, para configurar la «tabla de verdad» de la función y obtener, a través de ella, su expresión algebraica; una vez simplificada, dicha expresión puede ser directamente trasladada a un esquema de puertas como representación gráfica del circuito digital que «hace efectiva» dicha función.

enunciado $\rightarrow$ *tabla funcional* $\rightarrow$ *expresión algebraica* $\rightarrow$ *esquema de puertas*

De forma que el proceso de diseño de un sistema combinacional consiste en recorrer sucesivamente los diferentes niveles de descripción del mismo.

Un sistema secuencial admite, también, varios niveles y tipos de descripción:

• Nivel 1: Enumeración de las especificaciones o requisitos que se le exigen; es una descripción en términos de enunciado del correspondiente problema o proyecto, con indicación de todas aquellas prestaciones que interesan pero sin detallar los estados necesarios ni la evolución entre ellos.

• Nivel 2: Grafo de estados; descripción gráfica de los estados y de la evolución entre ellos, correspondiendo un círculo anotado a cada estado y un arco orientado a cada posible transición, con indicación sobre cada arco de la condición booleana (término de entrada) que produce la transición.

Por su propia definición conceptual dos estados no pueden estar en activo (marcados) a la vez; en cada momento el sistema se encuentra en uno de los estados del grafo, es decir, debe encontrarse marcado uno y solo uno de los estados del grafo y la *marca* pasará de un estado a otro a través de las transiciones (sin posibilidad de duplicarse ni de desaparecer).

Además del grafo de estados será necesario indicar la correspondencia de los estados con las salidas (que, si es autómata de Mealy, también dependerán de las entradas).

• Nivel 3: Funciones booleanas; expresión algebraica de las funciones de evolución de los estados y de activación de las salidas.

• Nivel 4: Configuración circuital en términos de biestables y de puertas lógicas; constituye el objetivo del proceso de diseño del sistema secuencial y consiste en la construcción circuital de las expresiones algebraicas de activación de las salidas y de las funciones de evolución del estado (con los correspondiente biestables).

El proceso de diseño de un sistema secuencial consiste en recorrer sucesivamente los cuatro niveles de descripción:

especificaciones/ /requisitos → **grafo de estados** → **funciones booleanas** → **realización circuital**

El paso de grafo de estados a funciones booleanas contiene dos etapas previas: la simplificación del número de estados por agrupación de estados equivalentes y la codificación binaria de los estados:

→ **grafo de estados** → *simplificación de estados* → *codificación de los estados* → **funciones booleanas** →

En muchos casos la acertada realización de estas etapas, simplificación y codificación, puede reducir en gran medida la complejidad de las funciones booleanas.

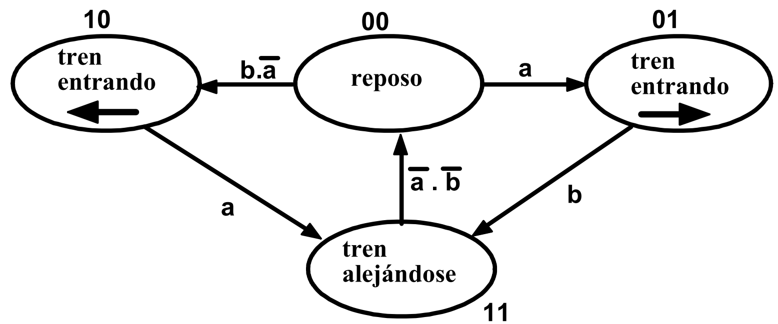
Si se utiliza un biestable RS para almacenar cada variable de estado, las funciones de evolución del estado se dividen en dos partes:

- las funciones de activación de los biestables (marcado)
- y las funciones de desactivación de los mismos (borrado).

A tales funciones es preciso añadir las funciones de activación de las salidas. Los tres conjuntos de funciones son de tipo combinacional (sin realimentación: la memoria es conformada por los biestables).

Las funciones de marcado y borrado de los biestables que contienen las variables de estado pueden obtenerse directamente de los arcos que representan las transiciones en el grafo de estados: para cada variable de estado ha de tenerse en cuenta el subconjunto de estados en que la correspondiente variable se encuentra a **1** y considerar las transiciones que conducen (marcado) y que abandonan (borrado) dicho subconjunto.

Consideremos el ejemplo de un semáforo de aviso de paso de tren en un cruce de vía única bidireccional con un camino (apartado 12.1):



La variable $\mathbf{q_1}$ está activada en los estados **01** y **11**, de forma que se marca cuando pasa a ellos desde los otros estados (**00** y **10**) y se borra al pasar del estado **11** al **00**: se marca con a desde **00** y, también, con a desde **10** y se borra con $\overline{a}\,.\,\overline{b}$ desde **11**.

q1: $S = \overline{q_2}.\overline{q_1}.a + q_2.\overline{q_1}.a = \overline{q_1}\,.\,a$ $\qquad R = q_2.q_1.\overline{b}.\overline{a}\,.$

La variable $\mathbf{q_2}$ se encuentra activa en los estados **10** y **11**: se marca con $b.\overline{a}$ desde **00** y con b desde **01** y se borra con $\overline{a}\,.\,\overline{b}$ desde **11**.

q2: $S = \overline{q_2}.\overline{q_1}.b.\overline{a} + \overline{q_2}.q_1.b = \overline{q_2}.b.(\overline{a} + q_1)$ $\qquad R = q_2.q_1.\overline{b}.\overline{a}\,.$

La salida (el semáforo) debe encontrarse activa en los estados **01**, **10** y **11**:

semáforo en rojo $= \overline{q_2}.q_1 + q_2.\overline{q_1} + q_2.q_1 = q_2 + q_1\,.$

Ahora bien, si el número de variables (entrada + estado) no es elevado conviene obtener las condiciones de marcado y borrado a partir de la tabla funcional correspondiente a la evolución de las variables de estado: **Q = F(X, Q)**.

En la misma tabla funcional de evolución de los estados pueden expresarse las funciones de marcado y borrado de cada una de las variables de estado; tales funciones se deducen a partir de los cambios que sufren las correspondientes variables de estado según las siguientes consideraciones:

- cuando el valor previo de $\mathbf{q_i}$ es **0** y dicho valor permanece, se requiere que $\mathbf{S_i=0}$ y, en cambio, no importa que valor adopte $\mathbf{R_i}$ (con **0** continúa el valor anterior y con **1** se borra, lo cual es lo mismo);
- cuando el valor previo de $\mathbf{q_i}$ es **1** y dicho valor permanece, es necesario que $\mathbf{R_i=0}$ y, en cambio, no importa que valor adopte $\mathbf{S_i}$ (con **0** continúa el valor anterior y con **1** se marca);
- cuando el valor previo de $\mathbf{q_i}$ es **0** y pasa a valor **1**, se requiere que $\mathbf{S_i=1}$ y $\mathbf{R_i=0;}$
- cuando el valor previo de $\mathbf{q_i}$ es **1** y pasa a valor **0**, se requiere que $\mathbf{S_i=0}$ y $\mathbf{R_i=1}$.

q_i:	**$0 \rightarrow 0$**	**R_i =**	X	**S_i =**	0
	$0 \rightarrow 1$		0		1
	$1 \rightarrow 0$		1		0
	$1 \rightarrow 1$		0		X

Para el ejemplo anterior, semáforo de aviso de paso de tren, la tabla de evolución del estado, ampliada con las columnas de marcado **S** y borrado **R**, es la siguiente:

q_2	q_1	b a	q_2^+	q_1^+	R_2	S_2	R_1	S_1
0	**0**	**0 0**	**0**	**0**	**X**	**0**	**X**	**0**
		0 1	**0**	**1**	**X**	**0**	**0**	**1**
		1 0	**1**	**0**	**0**	**1**	**X**	**0**
		1 1	**0**	**1**	**X**	**0**	**0**	**1**
0	**1**	**0 0**	**0**	**1**	**X**	**0**	**0**	**X**
		0 1	**0**	**1**	**X**	**0**	**0**	**X**
		1 0	**1**	**1**	**0**	**1**	**0**	**X**
		1 1	**1**	**1**	**0**	**1**	**0**	**X**
1	**0**	**0 0**	**1**	**0**	**0**	**X**	**X**	**0**
		0 1	**1**	**1**	**0**	**X**	**0**	**1**
		1 0	**1**	**0**	**0**	**X**	**X**	**0**
		1 1	**1**	**1**	**0**	**X**	**0**	**1**
1	**1**	**0 0**	**0**	**0**	**1**	**0**	**1**	**0**
		0 1	**1**	**1**	**0**	**X**	**0**	**X**
		1 0	**1**	**1**	**0**	**X**	**0**	**X**
		1 1	**1**	**1**	**0**	**X**	**0**	**X**

La elaboración de la tabla de las funciones de evolución del estado obliga a decidir para cada estado y para cada vector de entrada, es decir, para cada una de las situaciones posibles, cuál es la transición más adecuada, evitando errores de diseño por no haber previsto todas las situaciones que pueden presentarse; además, la citada tabla permite realizar las posibles simplificaciones a que hubiere lugar.

A partir de la tabla funcional, se obtienen las expresiones algebraicas de las funciones de marcado y borrado de cada uno de los biestables que conforman las variables de estado del sistema, pudiendo simplificarlas mediante mapas de Karnaugh.

En la tabla anterior, dichas funciones, una vez simplificadas serán:

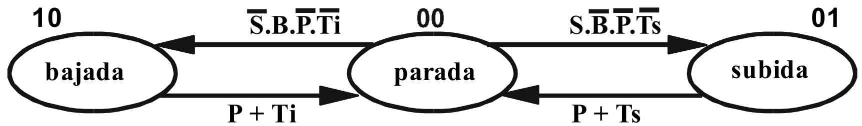
q_1: $S_1 = a$ $\qquad R_1 = q_2.\overline{b}.\overline{a}$

q_2: $S_2 = b.(\overline{a} + q_1)$ $\qquad R_2 = q_1.\overline{b}.\overline{a}$.

Estas funciones son algo más reducidas que las expresadas anteriormente, ya que la construcción de la correspondiente tabla funcional facilita su simplificación.

12.5. Ejercicios de diseño secuencial

12.5.1. *Montacargas con pulsadores de subida **S**, bajada **B** y paro **P**: para modificar el sentido de la marcha es preciso activar previamente el pulsador de paro y no responde a la activación simultánea de varios pulsadores; sendos topes fin de carrera **T_i** y **T_s** le impiden continuar subiendo o bajando cuando alcanza los extremos del recorrido.*



La codificación de los estados utiliza «un solo uno», correspondiendo cada variable a una de las direcciones del movimiento del montacargas (subida = **q_2** , bajada = **q_1**). A partir de este grafo de estados pueden expresarse directamente las funciones de marcado y de borrado de los biestables correspondientes a ambas variables de estado **q_1** y **q_2**.

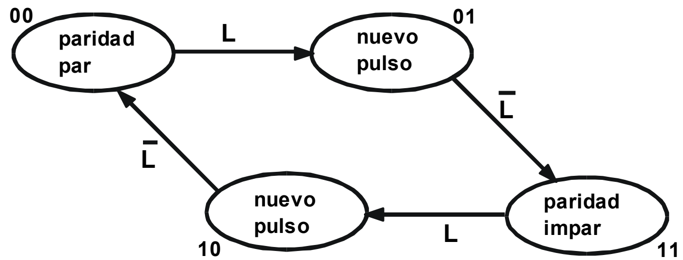
Sea **$q_0 = q_2 \Delta q_1$**, cuyo valor en el estado de **parada** es **1**

q_1: $S_1 = q_0.S.\overline{B}.\overline{P}.\overline{T_s}$ $\qquad R_1 = P + T_s$

q_2: $S_2 = q_0.\overline{S}.B.\overline{P}.\overline{T_i}$ $\qquad R_2 = P + T_i$.

En este caso, las funciones de borrado no necesitan incluir la correspondiente variable de estado.

12.5.2. *Indicador de paridad relativo al número de pulsos que llegan por una determinada línea **L**, de forma que indica paridad par **p=0** cuando ha llegado un número par de pulsos y paridad impar **p=1** cuando dicho número es impar; los pulsos se cuentan una vez que han finalizado, es decir, en su bajada (↓).*



Se han codificado los estados en código Gray; aunque algunos estados son distinguibles por la variable exterior **L** (**01** y **11** se distinguen por **L=1** y **L=0** e igualmente **10** y **00**) no se puede reducir el grafo de estados ya que es precisamente dicha variable la que produce las transiciones.

• función de salida: $\qquad p = q_2$.

• evolución del estado:

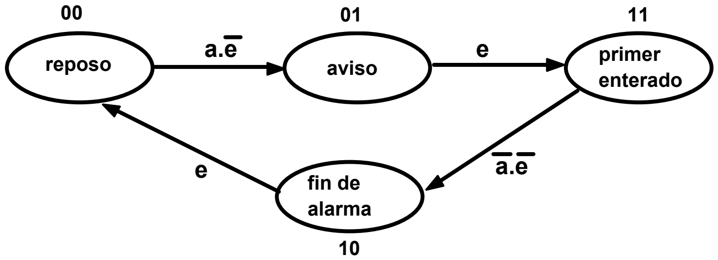
q1: $S_1 = \overline{q_2}.\overline{q_1}.L$ $\qquad R_1 = q_2.q_1.L$

q2: $S_2 = \overline{q_2}.q_1.\overline{L}$ $\qquad R_2 = q_2.\overline{q_1}.\overline{L}$

Este grafo de estados coincide con el de control de una lámpara de mesa (*flexo*) considerado en el apartado 11.2 (página 19); asimismo corresponde al grafo de estados del biestable tipo **T**.

12.5.3. *Un sistema secuencial actúa de la forma siguiente: cuando aparece una señal de alarma (entrada **a=1**) suena un claxon **c** y se enciende una lámpara **l** hasta que el operario pulsa una entrada de enterado **e**; con ello se apaga el claxon **c** pero, si la alarma sigue activa (**a=1**), la lámpara **l** sigue encendida hasta que desaparezca la alarma. En todo caso, cuando desaparece la alarma (**a=0**) la lámpara **l** se pone intermitente hasta que el operario vuelve a pulsar por segunda vez el contacto de enterado **e**. Este comportamiento permite al operario, una vez enterado de la situación de alarma y si ésta persiste, dedicarse a resolver tal situación; finalizada la misma, la lámpara se mantiene intermitente para recordar al operario que debe anotar la hora de la misma en un libro de registros, antes de pulsar un segundo enterado.*

El grafo correspondiente a la evolución de los estados puede reducirse a cuatro estados (agrupando adecuadamente aquellos que se diferencian por la variable **e**):



Los estados han sido codificados en código Gray con dos variables; construyendo la tabla de evolución de estados y, sobre ella, las columnas **Ri** y **Si**, se obtienen las siguientes funciones de evolución del estado simplificadas:

q1: $S_1 = \overline{q_2}.a.\overline{e}$ $\qquad R_1 = q_2.\overline{a}.\overline{e}$

q2: $S_2 = q_1.e$ $\qquad R_2 = \overline{q_1}.e\,.$

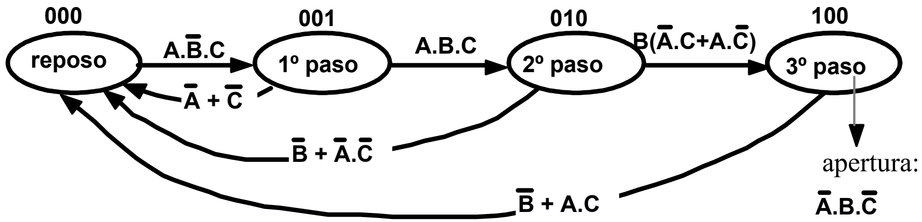
Las funciones de activación de las salidas serán las siguientes:

$$\text{claxon } c = \overline{q_2}.q_1 \qquad \text{luz } l = a + i.(q_2 + q_1);$$

la variable **i** corresponde a una entrada auxiliar que genera la intermitencia (una onda cuadrada de baja frecuencia, por ejemplo, medio segundo en **1** y otro medio en **0**).

12.5.4. *Secuencia sobre tres pulsadores para abrir una puerta: la apertura de una puerta está controlada por tres pulsadores **A**, **B** y **C**, de forma que se abre cuando se ejecuta sobre ellos la siguiente secuencia: 1º **A y C** (pulsadores de los extremos), 2º **A y B y C** (los tres pulsadores), 3º **B y C** o **A y B** indistintamente (dos pulsadores contiguos), 4º **B** (pulsador central); se progresa en la secuencia sin «soltar» los pulsadores y la puerta permanece abierta mientras se mantiene **B** pulsado.*

El grafo de estados necesario puede reducirse a cuatro estados, agrupando en el último de ellos el tercer y cuarto paso de la secuencia:



Las transiciones de vuelta al estado inicial son las siguientes: el primer paso se anula cuando no se produce $A.\overline{B}.C$ ni $A.B.C$; el segundo paso lo hace cuando no están presentes $A.B.C$ ni tampoco $B.(\overline{A}.C + A.\overline{C})$**;** el tercer estado pasa a reposo cuando $B = 0$ o cuando se pulsan conjuntamente A y C (ya que, en otro caso, se encontrará bien en una de las condiciones de transición a dicho estado, o bien en la situación de apertura de la puerta). Se ha utilizado la codificación con «un solo uno».

- **q_0 = *Nor*(q_3, q_2, q_1)** corresponde al estado **000**

• evolución del estado:

q1: $S_1 = q_0.A.\overline{B}.C$ $\qquad R_1 = q_1.(\overline{A} + \overline{C} + A.B.C) = q_1.(\overline{A} + B + \overline{C})$

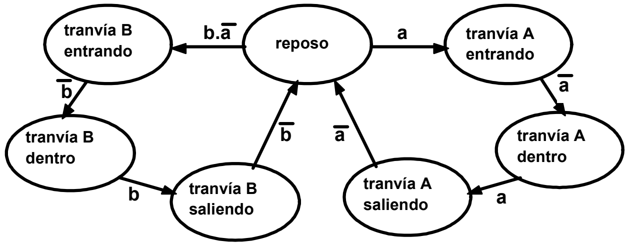
q2: $S_2 = q_1.A.B.C$ $\qquad R_2 = q_2.(\overline{A} + \overline{B} + \overline{C})$

q3: $S_3 = q_2.B.(\overline{A}.C + A.\overline{C})$ $\qquad R_3 = q_3.(\overline{B} + A.C)$

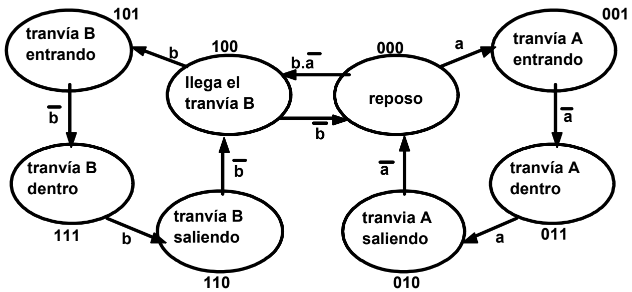
• función de salida: apertura de la puerta = $q_3.\overline{A}.B.\overline{C}\,.$

12.5.5. *Árbitro para la utilización de un recurso común: dos vías de tranvía (de ida y vuelta cada una de ellas), procedentes respectivamente de A y de B se unen en un punto C, de forma que tienen un tramo común desde C hasta D (dicho tramo CD es terminal: una vez que es ocupado por un tranvía, no puede entrar ningún otro, ya que el primero ha de salir por C); previamente al punto C existen sendos detectores de presencia de tranvía (**a** y **b**) y sendos semáforos (**S_a y S_b**) para detener a un tranvía cuando el trayecto común se encuentra ocupado por el otro.*

Un grafo de estados de la evolución de este sistema de arbitraje puede ser:



Este grafo incluye un número de estados relativamente alto; para su codificación, caso de utilizar una variable para cada estado (*un solo uno*), son necesarias seis variables de estado. En cambio, utilizando código Gray bastarán tres variables de estado; para ello, (para que en cada transición solamente cambie el valor de una de las variables) es necesario añadir un estado adicional, según el siguiente grafo de estados:



Se ha añadido el estado **100** que separa la evolución correspondiente al tranvía A de la que corresponde al tranvía B.

• funciones de salida:

semáforo **S_a** en rojo (vía común ocupada por B: detención del tranvía A) $S_a = q_3$

semáforo **S_b** en rojo (vía común ocupada por A: detención del tranvía B) $S_b = \overline{q_3}$.

El grafo da prioridad al tranvía A: el semáforo **S_b** se encontrará en rojo en reposo y pasará a verde cuando llegue el tranvía B a su detector, si el otro tranvía no ha llegado al suyo ni tampoco se encuentra dentro del tramo común. La asignación de prioridad evita colisiones en caso de que ambos tranvías alcancen a la vez sus respectivos sensores.

• evolución del estado:

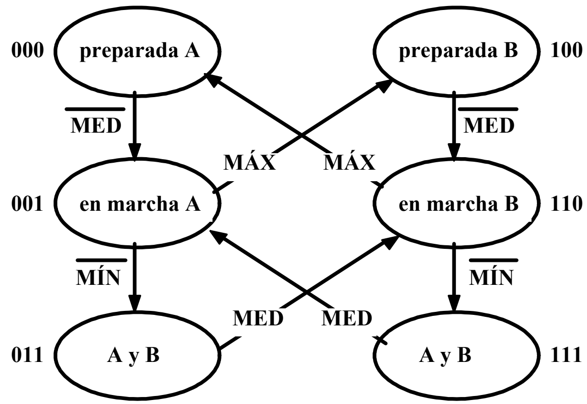
q_1: $S_1 = \overline{q_3}.\overline{q_2}.\overline{q_1}.a + q_3.\overline{q_2}.\overline{q_1}.b$ $\quad R_1 = \overline{q_3}.q_2.q_1.a + q_3.q_2.q_1.b$

q_2: $S_2 = \overline{q_3}.\overline{q_2}.q_1.\overline{a} + q_3.\overline{q_2}.q_1.\overline{b}$ $\quad R_2 = \overline{q_3}.q_2.\overline{q_1}.\overline{a} + q_3.q_2.\overline{q_1}.\overline{b}$

q_3: $S_3 = \overline{q_3}.\overline{q_2}.\overline{q_1}.b.\overline{a}$ $\quad R_3 = q_3.\overline{q_2}.\overline{q_1}.\overline{b}$.

12.5.6. *Un depósito de agua dispone de dos bombas **A y B** para su llenado y de tres detectores de nivel **MÁX**imo, **MED**io y **MÍN**imo. Cuando el nivel cae por debajo del nivel medio entra en funcionamiento una de las bombas hasta que alcanza el máximo; por debajo del mínimo actúan ambas bombas hasta el nivel medio y luego una sola de ellas hasta el máximo. Para equilibrar el desgaste de las bombas, cuando funciona una sola, lo hacen alternativamente (es decir, si la vez pasada lo ha hecho la bomba **A**, entrará en funcionamiento la **B** y viceversa).*

El grafo de estados debe ser simétrico respecto a la actuación de las bombas **A** y **B**; debe recordar cual de ambas ha actuado anteriormente y, por tanto, a cual de ellas le corresponde la siguiente puesta en marcha:



La codificación de estados indicada en este grafo (que no es ni Gray, ni de «un solo uno») resulta adecuada porque permite identificar las variables **q_1** y **q_2** con las bombas de agua **A** y **B**: activación de **A** = **q_1** ; activación de **B** = **q_2** .

• evolución del estado:

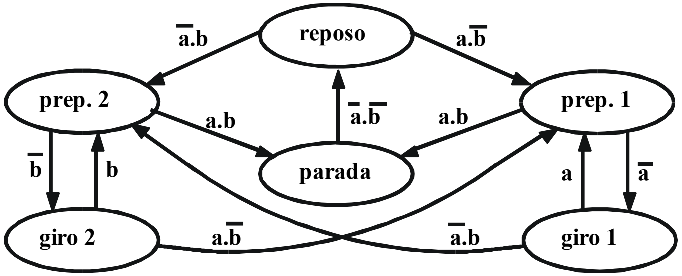
q_1: $S_1 = \overline{q_3}.\overline{q_2}.\overline{q_1}.\overline{\text{MED}} + \overline{\text{MÍN}}$ $\quad R_1 = \overline{q_3}.\text{MED} + \text{MÁX}$

q_2: $S_2 = q_3.\overline{q_2}.\overline{q_1}.\overline{\text{MED}} + \overline{\text{MÍN}}$ $\quad R_2 = q_3.\text{MED} + \text{MÁX}$

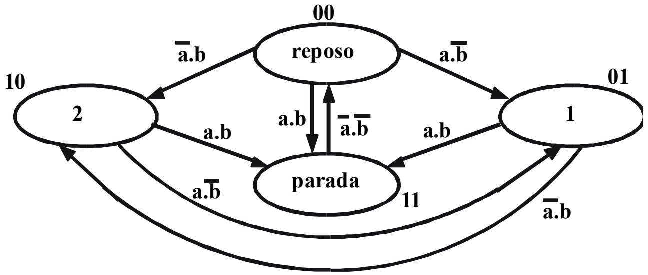
q_3: $S_3 = \overline{q_3}.\text{MÁX} + \overline{q_3}.q_2.q_1.\text{MED}$ $\quad R_3 = q_3.\text{MÁX} + q_3.q_2.q_1.\text{MED}$.

12.5.7. *Un motor puede girar en ambas direcciones, controlado por dos pulsadores **a** y **b** en la forma siguiente: al pulsar **a** se detiene el motor (si es que estaba en movimiento) y, al soltar dicho pulsador, el motor se pone a girar en el sentido de las agujas del reloj; lo mismo sucede al pulsar **b**, pero al soltarlo, el motor girará en sentido contrario a las agujas del reloj; la forma de detener el movimiento del motor es pulsar **a** y **b** a la vez, en cuyo caso el motor se para y, al soltar los pulsadores, continuará parado.*

El grafo de estados, como autómata de Moore, requiere 6 estados:



Pero los estados *prep. 1* y *giro 1* pueden distinguirse por el valor de la variable **a** e, igualmente, los estados *prep. 2* y *giro 2* pueden distinguirse por el valor de la variable **b**, de manera que el grafo puede reducirse, en forma de autómata de Mealy, a 4 estados:



Las funciones de salida correspondientes a los dos giros se producen en los estados **01** y **10** pero, solamente, cuando no están activados los pulsadores:

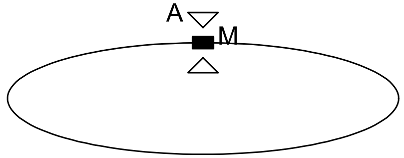
$$\text{giro } 1 = \overline{q_2}.q_1.\overline{a} \qquad \text{giro } 2 = q_2.\overline{q_1}.\overline{b}.$$

Las funciones de evolución del estado se han obtenido a través de su correspondiente tabla funcional (en la cual, para facilitar la simplificación, se ha hecho que el estado **00** si se activan ambos pulsadores a la vez pase al estado **11**); una vez construidas las columnas **R**

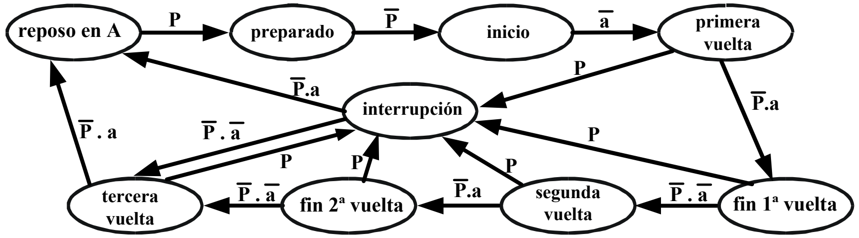
q_1: $S_1 = a \qquad R_1 = q_2.\overline{a}.\overline{b} + \overline{q_2}.\overline{a}.b = \overline{a}.(q_2 \oplus b)$

q_2: $S_2 = b \qquad R_2 = q_1.\overline{a}.\overline{b} + \overline{q_1}.a.\overline{b} = \overline{b}.(q_1 \oplus a).$

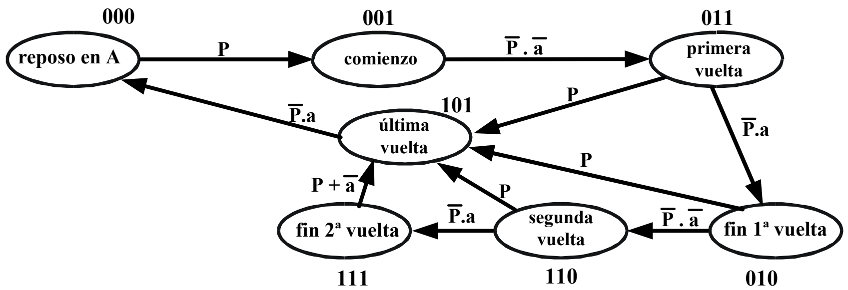
12.5.8. *Cierto mecanismo **M** se desplaza en el sentido de las agujas del reloj por un riel elipsoidal a partir de la posición **A**, en la cual existe un sensor que detecta la presencia del mismo: si se activa un pulsador **P**, **M** inicia su movimiento al soltar **P** y da 3 vueltas completas, tras lo cual se detiene en **A**; si durante el transcurso del movimiento se vuelve a pulsar **P**, **M** se para y, al soltar el pulsador, completa la vuelta que esta dando y se detiene al llegar a **A** (en todo caso, al menos da una vuelta).*



Un posible grafo de estados sería el siguiente (una vez iniciado el movimiento, si se pulsa **P** se va al estado de interrupción y desde dicho estado, al soltar **P**, se completa la vuelta hasta **A**):



Los estados *preparado* e *inicio* pueden agruparse ya que se diferencian por el valor de **P** y, por lo mismo, pueden agruparse los estados *interrupción* y *tercera vuelta* (de esta forma quedan 7 estados, codificables con 3 variables de estado):



La tabla de evolución de estados presenta solamente 5 variables: 3 de estado **q_3 q_2 q_1** y 2 de entrada **P a** (se deja como ejercicio para el lector). La salida (movimiento del mecanismo) debe activarse en todos los estados menos en el de *reposo*; en los estados de *comienzo* y de *última vuelta* se activa solamente cuando **P = 0**:

$$\text{movimiento de M} = q_2 + q_1.\overline{P}.$$