

Examen de S.E.T.I.

1er curso de Ingeniería Electrónica
 19 de septiembre de 2001

Cuestión teórica.

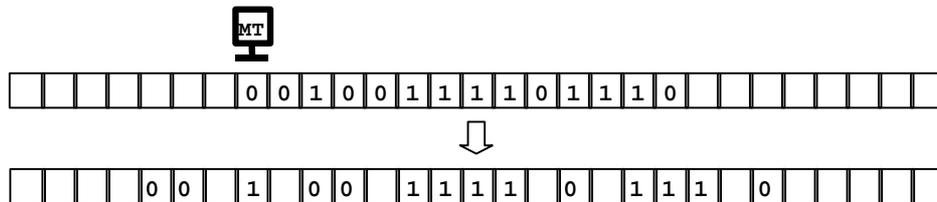
1 punto

- 1 Comente la siguiente afirmacion (puede ser cierta o no):

Es imposible construir una Máquina de Turing para resolver ciertos problemas

2 puntos

- 2 Escriba la tabla de una máquina de Turing que, dada una cinta en blanco con una sección conexas de unos y ceros distribuidos aleatoriamente, introduce un espacio en blanco entre cada dos elementos distintos. Considérese que la máquina comenzará a funcionar alineada con el elemento más a la izquierda.



Programación.

4 +3 puntos

- 3 Escriba una rutina (y si es necesario otras subrutinas) para el $\mu P68000$ que ejecute la función de la Máquina de Turing conforme al siguiente prototipo:

```

;RUTINA TM
;FUNCION: Máquina de Turing
;ENTRADA: A0(L) dirección de la tabla
;         A1(L) dirección de la posicion inicial en la cinta
;         D0(B) estado inicial
;SALIDA:
;MODIFICA: no altera ningún registro.
  
```

Podrá manejar 256 símbolos (caracteres) y un máximo de 127 estados más el de parada, de modo que cada regla de la tabla que se le proporcione constará de 4 bytes:

byte	significado
1	carácter de lectura
2	estado en la lectura (Si ==0 es marcador de fin de tabla)
3	carácter de escritura
4	estado tras la escritura [bits 0..6] (0=halt) + movimiento (bit 7. 0 = Dcha., 1 = Izqda.)

La rutina no dispondrá del número de reglas contenidas en la tabla, sino que el final de ésta se encontrará marcado por una "regla nula" caracterizada por tener a cero el byte 2 (téngase en cuenta que el estado no está permitido en tiempo de lectura ya que indica parada —

Ejemplo de tabla para la máquina:

Estad.	Car.	Car.	mov.	Estad.	byte 1	byte 2	byte 3	byte 4	
1	0	1	I	1	48	1	49	128+1	DC.B 48, 1, 49, 129
1	0	1	I	1	48	1	49	128+1	DC.B 48, 1, 49, 129
1	#	#	D	2	32	1	32	2	DC.B 32, 1, 32, 2
2	0	0	D	2	48	2	48	2	DC.B 48, 2, 48, 2
2	1	1	D	3	49	2	49	3	DC.B 49, 2, 49, 3
2	#	#	I	H	32	2	32	128+0	DC.B 32, 2, 32, 128
3	1	1	D	3	49	3	49	3	DC.B 49, 3, 49, 3
3	0	1	I	4	48	3	49	128+4	DC.B 48, 3, 49, 132
3	#	#	I	H	32	3	32	128+0	DC.B 32, 3, 32, 128
4	1	1	I	4	49	4	49	128+4	DC.B 49, 4, 49, 132
4	0	0	D	5	48	4	48	5	DC.B 48, 4, 48, 5
4	#	#	D	5	32	4	32	5	DC.B 32, 4, 32, 5
5	1	0	D	2	49	5	48	2	DC.B 49, 5, 48, 2
					0	0	0	0	DC.B 0, 0, 0, 0

4 El planteamiento anterior de la Máquina de Turing es en realidad inadecuado porque no contempla las posibles situaciones de error. Alterar la rutina anterior para añadirle los siguientes controles:

- Se terminará la ejecución con una marca de error en caso de que no exista en la tabla una regla necesaria.
- Se tendrá en cuenta un número máximo de pasos, de modo que la ejecución termine al realizarlos aún cuando no se haya alcanzado el estado "halt"
- Se tendrán en cuenta limitaciones en los extremos de la cinta, es decir, no se podrá acceder a posiciones de memoria fuera de un intervalo indicado. En caso de que la máquina llegue a "necesitar" escribir fuera del intervalo, la ejecución se detendrá con la consiguiente marca de error.

```

;RUTINA TM
;FUNCION: Máquina de Turing
;ENTRADA: A0(L) dirección de la tabla
;         A1(L) dirección de cinta
;         A2(L) dirección mínima perteneciente a la cinta
;         A3(L) dirección máxima perteneciente a la cinta
;         D0(B) estado inicial
;         D1(W) número máximo de pasos permitido.
;SALIDA:  Z=1 ejecución terminada normalmente
;         N=1 salida por límite de iteraciones
;         V=1 Falta regla necesaria
;MODIFICA: no altera ningún registro.
  
```

Examen de S.E.T.I.

1er curso de Ingeniería Electrónica
 19 de septiembre de 2001

SOLUCIONES

1 "Es imposible construir una Máquina de Turing para resolver ciertos problemas cierta. La Máquina Universal de Turing es capaz de resolver cualquier problema que tenga solución, o lo que es lo mismo, es posible construir una Máquina de Turing para cualquier problema con solución. Esto en realidad no es más que una conjetura (la conocida como conjetura de "Church-Turing") ya que la demostración de la equivalencia del cálculo lambda de Alonzo Church y la Máquina de Turing (y algunos otros sistemas menos conocidos actualmente) llevó a postular que los problemas resolubles con estos mecanismos coincidirían con los problemas con solución, pero no existe una demostración formal. Lo que sí se ha demostrado es la existencia de problemas sin solución y la imposibilidad (evidentemente) de ser resueltos por cualquiera de los anteriores mecanismos. Por ello podemos asegurar que hay problemas que para los que no puede construirse una Máquina de Turing que los resuelva. El conocido como "problema de la parada" es una demostración sencilla de este hecho.

2 Un "algoritmo" muy sencillo para resolver el problema puede ser el que se explica con todo detalle a

- (1) Insertar en todas las secuencias 01 y 10 un marcador (p.ej. 0X1 y 1X0).
- (2) Cambiar todos los marcadores por blancos.

"0" ; "1" ; "X" ; " " .

Veamos ahora como ejecutar cada una de las fases:

- (1.1) Moverse hacia la derecha buscando un cambio 01 ó 10 y si se llega al blanco terminar
- (1.2) poner un marcador a la izquierda (entre ambos símbolos) recordando lo que se quita
- (1.3) mientras no se llegue al principio poner a la izquierda lo recordado y recordar lo quitado repetir

Esta primera fase termina con la máquina en el extremo derecho de la cinta, de modo que la segunda puede llevarse a cabo de derecha a izquierda.

- (2.1) Moverse hacia la izquierda cambiando "X" por " " hasta encontrar un " " .

Concretemos ahora el paso (1.1). Para detectar un cambio 01 ó 10, necesitamos en cada instante saber cual ha sido el símbolo anterior de modo que necesitaremos tres estados que se muestran a continuación junto con las reglas correspondientes a todos los símbolos posibles:

[Nota: como (1.1) requiere los estados 1, 2 y 3, el paso a (1.2) será al estado 4. La terminación de la fase (1) para pasar a la (2) requiere ir a un estado que se determina más adelante, tras ver toda la fase (1) al completo, y que resulta ser el 8. Se representa como {8} para indicar que no se ha podido determinar inmediatamente]

(1.1) Estado 1: El carácter anterior ha sido un "0" .		Esqr.	Mov.	Est.
Carácter	Acción			
"0"	Seguir hacia la derecha sin más.	"0"	D	1
"1"	Ir a la izquierda y avanzar al paso (1.2).	"1"	I	4
"X"	Seguir hacia la derecha recordando que se ha visto una "X".	"X"	D	3
" "	Terminar el paso (1) .	" "	I	{8}

(1.1) Estado 2: El carácter anterior ha sido un "1" .

Carácter	Acción	Escr.	Mov.	Est.
"0"	Ir a la izquierda y avanzar al paso (1.2).	"0"	I	4
"1"	Seguir hacia la derecha sin más.	"1"	D	2
"X"	Seguir hacia la derecha recordando que se ha visto una "X".	"X"	D	3
" "	Terminar el paso (1)..	" "	I	{8}

(1.1) Estado 3: El carácter anterior ha sido una "X" .

Carácter	Acción	Escr.	Mov.	Est.
"0"	Seguir hacia la derecha recordando que se ha visto un "0".	"0"	D	1
"1"	Seguir hacia la derecha recordando que se ha visto un "1".	"1"	D	2
"X"	Situación imposible.			
" "	Situación imposible.			

El paso (1.2) se corresponderá con el estado 4, en el que se escribe un marcador y se recuerda lo que se quita, de modo que ello implica pasar a dos estados diferentes en función de lo borrado:

(1.2) Estado 4: Poner "X" e ir a la izquierda recordando lo borrado

Carácter	Acción	Escr.	Mov.	Est.
"0"	Seguir hacia la derecha recordando que se ha visto un "0".	"X"	I	5
"1"	Seguir hacia la derecha recordando que se ha visto un "1".	"X"	I	6
"X"	Situación imposible.			
" "	Situación imposible.			

El paso (1.3) es el desplazamiento a izquierda "arrastrando" los símbolos, por lo que necesita de nuevo de tres estados que recuerden cada uno de los posibles símbolos borrados. Dos de ellos son el 5 y el 6, que recuerdan ceros y unos, y necesitaremos uno más para recordar las equis.

(1.3) Estado 5: A la derecha había un "0" .

Carácter	Acción	Escr.	Mov.	Est.
"0"	Poner "0" y recordar "0".	"0"	I	5
"1"	Poner "0" y recordar "1".	"0"	I	6
"X"	Poner "0" y recordar "X".	"0"	I	7
" "	Poner "0" y repetir desde (1.1) recordando el "0" puesto.	"0"	D	1

(1.3) Estado 6: A la derecha había un "1" .

Carácter	Acción	Escr.	Mov.	Est.
"0"	Poner "1" y recordar "0".	"1"	I	5
"1"	Poner "1" y recordar "1".	"1"	I	6
"X"	Poner "1" y recordar "X".	"1"	I	7
" "	Poner "1" y repetir desde (1.1) recordando el "1" puesto.	"1"	I	2

(1.3) Estado 7: A la derecha había una "X" .

Carácter	Acción	Escr.	Mov.	Est.
"0"	Poner "X" y recordar "0".	"X"	I	5
"1"	Poner "X" y recordar "1".	"X"	I	6

"X" Situación imposible.
 " " Situación imposible.

Terminada la fase (1) vemos la (2) que, como ya se ha adelantado comienza con el estado 8 y es tan simple que no necesita de ningún otro.

(2) Estado 8: Todo a la izquierda cambiando "X" por " " .

Carácter	Acción	Escr.	Mov.	Est.
"0"	Izquierda sin más.	"0"	I	8
"1"	Izquierda sin más.	"1"	I	8
"X"	Izquierda tras cambiarlo por " " .	" "	I	8
" "	Terminar.	" "	D	H

La tabla completa será por tanto:

Estado			Escr.	Mov.	Est.
1	"0"		"0"	D	1
1	"1"		"1"	I	4
1	"X"		"X"	D	3
1	" "		" "	I	8
2	"0"		"0"	I	4
2	"1"		"1"	D	2
2	"X"		"X"	D	3
2	" "		" "	I	8
3	"0"		"0"	D	1
3	"1"		"1"	D	2
4	"0"		"X"	I	5
4	"1"		"X"	I	6
5	"0"		"0"	I	5
5	"1"		"0"	I	6
5	"X"		"0"	I	7
5	" "		"0"	D	1
6	"0"		"1"	I	5
6	"1"		"1"	I	6
6	"X"		"1"	I	7
6	" "		"1"	I	2
7	"0"		"X"	I	5
7	"1"		"X"	I	6
8	"0"		"0"	I	8
8	"1"		"1"	I	8
8	"X"		" "	I	8
8	" "		" "	D	H

3 Este problema "poco realista" por la falta de control en las situaciones de error se resuelve como sigue: la Máquina de Turing realiza un proceso cíclico hasta llegar al estado 0 en la rutina "TM". Dentro del ciclo llama a una sub-rutina para localizar la regla aplicable en cada momento de acuerdo con el estado [D0] y el carácter en la cinta [(A1)]. Localizada la regla [A4] se escribe el nuevo carácter en la cinta y se toma el dato de "nuevo estado y movimiento" en el registro de estado [D0]. Al escribir el nuevo carácter se hace un postincremento, lo que supone un movimiento a derecha del cabezal de la máquina; a continuación se borra el bit de movimiento del registro de estado dejándolo listo para el siguiente paso y afectando a los códigos de condición, lo que permite mover el cabezal dos posiciones atrás en caso de que se tratase de movimiento a izquierda.

La rutina de búsqueda de regla "LOCALIZA_REGLA" presupone que la regla buscada existe y se limita a hacer un ciclo hasta encontrarla comparando en cada caso el estado y carácter actuales con los presentes en la tabla.

```

;RUTINA TM
;FUNCION: Máquina de Turing
;ENTRADA: A0(L) dirección de la tabla
;         A1(L) dirección de cinta
;         D0(B) estado inicial
;SALIDA:
;MODIFICA: no altera ningún registro.

TM      MOVE.B  (A1),D2          ;Toma carácter de la cinta
        BSR.S  LOCALIZA_REGLA  ;Localiza la regla en la tabla
        MOVE.B 2(A4),(A1)+      ;Escribe car. la cinta (pre-ejecuta dcha.)
        MOVE.B 3(A4),D0        ;y toma nuevo estado (incluye mov.)
        BCLR   #7,D0           ;limpia (y testea) bit de movimiento.
        BEQ.S  CONT            ;si dcha. ya está hecho.
IZQDA   SUBQ.L #2,A1           ;izqda=-2 porque se pre-ejecuto dcha
CONT    TST.B  D0              ;si el estado era halt
        BNE.S  TM
TERMINA RTS

;RUTINA LOCALIZA_REGLA
;FUNCION: Dada la tabla de una máquina de Turing y un estado y carácter,
;         localiza la correspondiente regla
;ENTRADA: A0(L) dirección de la tabla
;         D0(B) estado
;         D2(B) carácter.
;SALIDA: A4(L) apuntando a la regla (debe existir)
;MODIFICA: no altera ningún registro.

LOCALIZA_REGLA
        MOVE.L A0,A4          ;Usara A4 para recorrer la tabla
CICLO_LR CMP.B  (A4),D2        ;Compara el carácter
        BNE.S  AVANZA         ;Si no coincide, avanza 1 regla
        CMP.B 1(A4),D0        ;Compara el estado
        BEQ.S  LOCALIZADA     ;Si coincide ya está localizada la regla
AVANZA  ADDQ.L #4,A4          ;sino avanza a la siguiente
        BRA.S  CICLO_LR       ; y continúa buscando
LOCALIZADA RTS
  
```

4 El primer punto en que añadiremos un control de error será en la rutina "localiza_regla" que debe tener en cuenta la posibilidad de que no exista la regla buscada. Esto se indicará con el código de

```

;RUTINA LOCALIZA_REGLA
;FUNCION: Dada la tabla de una máquina de Turing y un estado y caracter,
; localiza la correspondiente regla
;ENTRADA: A0(L) dirección de la tabla
; D0(B) estado
; D2(B) caracter.
;SALIDA: A4(L) apuntando a la regla en caso de que exista
; Z=1 (EQ) ejecución terminada normalmente
; Z=0 (NE) Falta regla necesaria
;MODIFICA: no altera ningún registro.

LOCALIZA_REGLA
MOVE.L A0,A4 ;Usara A4 para recorrer la tabla
CICLO_LR TST.B 1(A4) ;Si la segunda columna es cero
BEQ.S NOREGLA ;se han acabado las reglas sin encontrarla
CMP.B (A4),D2 ;Compara el caracter
BNE.S AVANZA ;Si no coincide, avanza 1 regla
CMP.B 1(A4),D0 ;Compara el estado
BEQ.S LOCALIZADA ;Si coincide ya esta localizada la regla
AVANZA ADDQ.L #4,A4 ;sino avanza a la siguiente
BRA.S CICLO_LR ; y continua buscando
NOREGLA MOVE #0,CCR ;error regla no existe (NE)
LOCALIZADA RTS
  
```

De este modo la rutina "TM" debe tener en cuenta esta condición después de llamar a la sub-rutina.

Además el ciclo deberá ser controlado ahora por el contador de iteraciones máximas [D1] de manera que se comienza entrando en la zona de control de finalización (etiquetada CONT). Por último debe añadirse el control de los límites de la cinta que se lleva a cabo después de realizar el movimiento comparando la nueva posición del cabezal [A1] con los límites establecidos como parámetros.

```

;RUTINA TM
;FUNCION: Máquina de Turing
;ENTRADA: A0(L) dirección de la tabla
; A1(L) dirección de cinta
; A2(L) dirección mínima perteneciente a la cinta
; A3(L) dirección máxima perteneciente a la cinta
; D0(B) estado inicial
; D1(W) número máximo de pasos permitido.
;SALIDA: Z=1 Ejecucion terminada normalmente
; N=1 salida por limite de iteraciones
; V=1 Falta regla necesaria
; nota: si no es N ni V pero no se ha terminado normalmente (Z=0)
; entonces se ha producido un error de "limite en la cinta"
;MODIFICA: no altera ningún registro.

TM BRA.S CONT
CICLO MOVE.B (A1),D2 ;Toma caracter de la cinta
BSR.S LOCALIZA_REGLA ;Localiza la regla en la tabla
BNE.S NO_REGLA ;Si no existe termina (NE Z=0)
MOVE.B 2(A4),(A1)+ ;Escribe car. la cinta (pre-ejecuta dcha.)
MOVE.B 3(A4),D0 ;y toma nuevo estado (incluye movimiento.)
BCLR #7,D0 ;limpia (y testea) bit de mov.
BNE.S IZQDA ;si iqda retrocedera A2(dcha pre-ejecutado)
CMP.L A3,A1 ;si se sobrepasa el máximo en la cinta
BHI.S TERMINA ;error. Z=V=N=0 porque A1-A3=1
BRA.S CONT
IZQDA SUBQ.L #2,A1 ;izqda=-2 porque se pre-ejecuto dcha
  
```

```

          CMP.L   A1,A2           ;si se sobrepasa el máximo en la cinta
          BHI.S   TERMINA        ;error. Z=V=N=0 porque A2-A1=1
CONT     TST.B   D0             ;si el estado era halt, termina (EQ Z=1)
          BEQ.S   TERMINA
          DBRA   D1,CICLO        ;Da otro paso si ni pasa el limite
          OR.B   #$08,CCR        ;activa N para indicar limite iteraciones
TERMINA  RTS
NO_REGLA MOVE   #$02,CCR        ;Pone el bit de overflow [000xnzVc]
          RTS
  
```

Un programa ejemplo de uso de estas rutinas en el 68fil puede ser el siguiente:

SECTION 0

PRUEBA:

```

LEA     MAQUINA1,A0
LEA     CINTA,A1
LEA     MINCINTA,A2
LEA     MAXCINTA-1,A3
MOVE.B  #1,D0
MOVE.W  #1000,D1
BSR     TM
TRAP    #15
DC.W    0
  
```

SECTION 1

;Aqui las rutinas

SECTION 2

MAQUINA1:

```

DC.B    48, 1, 49, 129
DC.B    48, 1, 49, 129
DC.B    32, 1, 32, 2
DC.B    48, 2, 48, 2
DC.B    49, 2, 49, 3
DC.B    32, 2, 32, 128
DC.B    49, 3, 49, 3
DC.B    48, 3, 49, 132
DC.B    32, 3, 32, 128
DC.B    49, 4, 49, 132
DC.B    48, 4, 48, 5
DC.B    32, 4, 32, 5
DC.B    49, 5, 48, 2
DC.B    0,0,0,0
  
```

SECTION 3

```

MINCINTA DC.B    '
CINTA    DC.B    '00100110101'
          DC.B    '
MAXCINTA
  
```