Tema 2: Máquinas de Turing

3. Diseño de máquinas de Turing con objetivos prefijados.

A continuación vamos a diseñar máquinas de Turing que realicen tareas concretas. En lo que sigue, salvo que se indique lo contrario, denotaremos una casilla vacía en la cinta de la máquina por el símbolo s_0. Llamaremos **estado de partida** de una máquina de Turing al estado en el que se encuentra el dispositivo cuando comienza a actuar.

Definición. Sea $\mathfrak{T} = (S, E, \mathfrak{C})$ una máquina de Turing. Se llama **descripción instantánea de** $\mathfrak{T}$ a una palabra $\alpha = Pe_is_jP'$, donde $P, P' \in \Omega_S$, $e_i \in E$ y $s_j \in S$.

Intuitivamente, una descripción instantánea debe entenderse de la manera siguiente: "los símbolos de la cinta son las letras que aparecen en P s_j y P' (escritos éstos en celdas contiguas y casillas en blanco en el resto) y la cabeza lectora-inscriptora se encuentra en el estado e_i examinando la casilla que contiene s_j".

Por ejemplo, la descripción instantánea $\alpha = s_2e_3s_1s_5s_7s_2$ la interpretamos: "los símbolos de la cinta son $s_2s_1s_5s_7s_2$ y la máquina se encuentra en el estado e_3 examinando una casilla que contiene a s_1".

Definición. Sea $\mathfrak{T} = (S, E, \mathfrak{C})$ una máquina de Turing y $\alpha = Pe_is_jP'$ una descripción instantánea de $\mathfrak{T}$. Decimos que α es **terminal** si no existe un cuádruple en $\mathfrak{C}$ que comience por e_is_j.

Definición. Sea $\mathfrak{T} = (S, E, \mathfrak{C})$ una máquina de Turing y α y β dos descripciones instantáneas. Escribiremos $\alpha \to \beta$ siempre que existan $P, P' \in \Omega_S$ de manera que se verifique una de las siguientes condiciones:

i) $\begin{array}{rcl} \alpha & = & Pe_is_jP' \\ \beta & = & Pe_ls_kP' \end{array}$ y el cuádruple $e_is_js_ke_l \in \mathfrak{C}$

ii) $\begin{array}{rcl} \alpha & = & Pe_is_js_kP' \\ \beta & = & Ps_je_ls_kP' \end{array}$ y el cuádruple $e_is_jDe_l \in \mathfrak{C}$

iii) $\begin{array}{rcl} \alpha & = & Pe_is_j \\ \beta & = & Ps_je_ls_0 \end{array}$ y el cuádruple $e_is_jDe_l \in \mathfrak{C}$

iv) $\begin{array}{rcl} \alpha & = & Ps_ke_is_jP' \\ \beta & = & Pe_ls_ks_jP' \end{array}$ y el cuádruple $e_is_jIe_l \in \mathfrak{C}$

v) $\begin{array}{rcl} \alpha & = & e_i s_j P' \\ \beta & = & e_l s_0 s_j P' \end{array}$ y el cuádruple $e_i s_j I e_l \in \mathfrak{C}$.

Diremos que $\alpha \to \beta$ es un **movimiento básico** de la máquina T.

Definición. Una **computación** de una máquina de Turing $\mathfrak{T} = (S, E, \mathfrak{C})$ es una sucesión finita de descripciones instantáneas $\alpha_1, \alpha_2, \ldots, \alpha_t$ tales que $\alpha_i \to \alpha_{i+1}$ para $i = 1, 2, \ldots, t-1$ y α_t es terminal.

Ejemplo. Consideramos la máquina $\mathfrak{T} = (S, E, \mathfrak{C})$ con $S = \{s_0, s_1, s_2\}$, $E = \{e_0, e_1, e_2, e_3\}$ y $\mathfrak{C} = \{e_0 s_0 I e_0, e_0 s_1 D e_1, e_0 s_2 D e_1, e_1 s_0 D e_1, e_1 s_2 s_1 e_1, e_1 s_1 D e_2, e_2 s_0 s_0 e_3, e_2 s_2 s_2 e_2\}$. Entonces, la sucesión de descripciones instantáneas $\alpha_1 = s_2 e_1 s_2 s_1$, $\alpha_2 = s_2 e_1 s_1 s_1$, $\alpha_3 = s_2 s_1 e_2 s_1$ es una computación. En cambio, si partimos de $\alpha_1 = s_2 e_1 s_2 s_2$, tenemos $\alpha_2 = s_2 e_1 s_1 s_2$ y $\alpha_i = s_2 s_1 e_2 s_2$ para $i \geq 3$, es una sucesión infinita de descripciones instantáneas tales que $\alpha_i \to \alpha_{i+1}$ que no es una computación.

El ejemplo anterior nos sugiere el planteamiento de la siguiente cuestión: si consideramos una máquina de Turing que tenga en su cinta escrita una determianada palabra con las letras del conjunto S y la cabeza lectora-inscriptora se encuentra en un estado $e \in E$ examinando una de las casillas, ¿podemos saber si va a existir una computación que comience con la descripción instantánea que presenta la máquina? A esto es lo que se conoce como **problema de parada** y la respuesta es en general no.

Nos planteamos ahora el diseño de máquinas de Turing que realicen tareas determinadas. Obviamente, para que el resultado ofrecido por la máquina sea el deseado, tendremos que pedir condiciones iniciales, como por ejemplo, fijar un estado de partida o que las palabras escritas en la cinta no contengan casillas en blanco, etc...

Ejemplo 1. Deseamos diseñar una máquina de Turing que al serle introducida una sucesión finita de "1", colocados en celdas contiguas y estando situada la cabeza sobre uno cualquiera de ellos en el estado de partida e_0, nos permita contar cuántos "1" aparecen escrito en notación decimal. Está claro que los símbolos que al menos debe admitir la máquina serán los dígitos 0 a 9 y el símbolo de casilla en blanco. Además, pueden ser necesarios introducir símbolos adicionales dependiendo de la estrategia a seguir. El algoritmo que emplearemos será el siguiente:

1) En primer lugar, nos situamos al comienzo de la sucesión de "1", moviendonos hacia la izquierda. Una vez localizado, nos trasladamos dos celdas más hacia la izquierda y marcamos la segunda de las casillas vacías con "0" (en ésta, colocaremos la cifra de las unidades del número de "1" que aparecen en la sucesión).

2) A continuación, nos dirigimos hacia la derecha hasta toparnos con el primer "1" que tenga la sucesión. Lo sustituimos por "a" y volvemos hacia atras para añadir una unidad más a la cifra que teníamos anotada.

3) El proceso anterior finalizará cuando se extinga la sucesión de "1". En ese instante, cambiaremos "a" por s_0 y tras esto, la máquina se parará, devolviendonos el número de "1" que había en la sucesión.

Por tanto, el conjunto de símbolos que tiene la máquina es $S = \{s_0, a, 0, \ldots, 9\}$, los estados en los que puede estar son: $E = \{e_0, \ldots, e_8\}$ y los cuádruples que la forman:

$$\mathfrak{C} = \{e_0 1 I e_0, e_0 s_0 I e_1, e_1 s_0 0 e_2, e_2 0 D e_2, e_2 1 D e_2, e_2 2 D e_2, e_2 3 D e_2, e_2 4 D e_2,$$

$$e_2 5 D e_2, e_2 6 D e_2 e_2 7 D e_2, e_2 8 D e_2, e_2 9 D e_2, e_2 s_0 D e_3, e_3 a D e_3, e_3 1 a e_4,$$

$$e_4 a I e_4, e_4 s_0 I e_5, e_5 0 1 e_2, e_5 1 2 e_2, e_5 2 3 e_2 e_5 3 4 e_2, e_5 4 5 e_2, e_5 5 6 e_2, e_5 6 7 e_2,$$

$$e_5 7 8 e_2, e_5 8 9 e_2, e_5 9 0 e_6, e_6 0 I e_5, e_5 s_0 1 e_2, e_3 s_0 I e_7, e_7 a s_0 e_8, e_8 s_0 I e_7\}$$

Veamos cómo actúa la máquina diseñada en un ejemplo concreto:

MOVIMIENTO	*ACCION*	*CINTA* *ESTADO*
Inicio	–	1 1 e_0
1	$e_0 1 I e_0$	s_0 1 1 e_0
2	$e_0 s_0 I e_1$	s_0 s_0 1 1 e_1
3	$e_1 s_0 0 e_2$	0 s_0 1 1 e_2
4	$e_2 0 D e_2$	0 s_0 1 1 e_2
5	$e_2 s_0 D e_3$	0 s_0 1 1 e_3
6	$e_3 1 a e_4$	0 s_0 a 1 e_4
7	$e_4 a I e_4$	0 s_0 a 1 e_4
8	$e_4 s_0 I e_5$	0 s_0 a 1 e_5
9	$e_5 0 1 e_2$	1 s_0 a 1 e_2
10	$e_2 1 D e_2$	1 s_0 a 1 e_2

MOVIMIENTO	*ACCION*	*CINTA* *ESTADO*
11	$e_2s_0De_3$	1 s_0 a 1 e_3
12	e_3aDe_3	1 s_0 a 1 e_3
13	e_31ae_4	1 s_0 a a e_4
14	e_4aIe_4	1 s_0 a a e_4
15	e_4aIe_4	1 s_0 a a e_4
16	$e_4s_0Ie_5$	1 s_0 a a e_5
17	e_512e_2	2 s_0 a a e_2
18	e_22De_2	2 s_0 a a e_2
19	$e_2s_0De_3$	2 s_0 a a e_3
20	e_3aDe_3	2 s_0 a a e_3
21	e_3aDe_3	2 s_0 a a s_0 e_3
22	$e_3s_0Ie_7$	2 s_0 a a s_0 e_7
23	$e_7as_0e_8$	2 s_0 a s_0 s_0 e_8
24	$e_8s_0Ie_7$	2 s_0 a s_0 s_0 e_7
25	$e_7as_0e_8$	2 s_0 s_0 s_0 s_0 e_8
26	$e_8s_0Ie_7$	2 s_0 s_0 s_0 s_0 e_7

La máquina se para puesto que no existe ningún cuádruple que comience por e_7s_0 y el resultado que ofrece, como cabía esperar, es 2.

Ejemplo 2. Construimos una máquina de Turing que cuando introducimos un número natural n, escrito en notación unaria, la máquina nos devuelve ese número multiplicado por 3, también en notación unaria. En la inicialización, la máquina tiene su cabeza lectora-inscriptora examinado el primer “1” de la sucesión y el estado de partida es el e_0. Utilizaremos dos símbolos adicionales “a” y “b‘”. “a” aparecerá en lugar de un “1”, para indicarnos que el “1” que ocupaba esa posición ya ha sido o está siendo triplicado y “b” nos señalará cuantos “1” debemos añadir para conseguir nuestro objetivo.

La idea intuitiva en que nos basamos para diseñar la máquina es la siguiente:

1) Tomamos el primer “1” de la sucesión, lo sustituimos por “a”, localizamos el final de la sucesión de “1” y añadimos dos “b”.

2) Buscamos el primer “1” que aparezca en la sucesión. Si no lo hay, se va al paso 4. En otro caso, lo sustituimos por “a”, localizamos el final de la sucesión de “1” y “b”. Tras ello, añadimos dos “b” a la sucesión de éstas.

3) Reiteramos 2) hasta que se acaben los “1”.

4) Sustituimos “a” y “b” por “1” y el proceso se acaba.

Una máquina de Turing que realice las operaciones indicadas será aquella que tiene $S = \{s_0, 1, a, b\}$, $E = \{e_i \mid i = 1, \dots, 7\}$ y

$$\mathfrak{C} = \{e_1 1 a e_2, e_2 a D e_2, e_2 1 D e_2, e_2 b D e_2, e_2 s_0 b e_3, e_3 b D e_4,$$

$$e_4 s_0 b e_5, e_5 b I e_5, e_5 1 I e_5, e_5 a D e_1, e_1 b b e_6,$$

$$e_6 a I e_6, e_6 b I e_6, e_6 s_0 D e_7, e_7 a 1 e_7, e_7 1 D e_7, e_7 b 1 e_7\}$$

En el caso particular de introducir una sucesión con dos “1”, los movimientos que realizará la máquina diseñada son los siguientes:

MOVIMIENTO	*ACCION*	*CINTA* *ESTADO*
Inicio	–	$1 \quad 1$ e_0
1	$e_0 1 I e_0$	$1 \quad 1$ e_0
2	$e_0 s_0 D e_e$	$1 \quad 1$ e_1
3	$e_1 1 a e_2$	$a \quad 1$ e_2
4	$e_2 a D e_2$	$a \quad 1$ e_2
5	$e_2 1 D e_2$	$a \quad 1 \quad s_0$ e_2
6	$e_2 s_0 b e_3$	$a \quad 1 \quad b$ e_3

MOVIMIENTO	*ACCION*	*CINTA* *ESTADO*
7	e_3bDe_4	$a\ 1\ b\ s_0$ e_4
8	$e_4s_0be_5$	$a\ 1\ b\ b$ e_5
9	e_5bIe_5	$a\ 1\ b\ b$ e_5
10	e_5bIe_5	$a\ 1\ b\ b$ e_5
11	e_51Ie_5	$a\ 1\ b\ b$ e_5
12	e_5aDe_1	$a\ 1\ b\ b$ e_1
13	e_11ae_2	$a\ a\ b\ b$ e_2
14	e_2aDe_2	$a\ a\ b\ b$ e_2
15	e_2bDe_2	$a\ a\ b\ b$ e_2
16	e_2bDe_2	$a\ a\ b\ b\ s_0$ e_2
17	$e_2s_0be_3$	$a\ a\ b\ b\ b$ e_3
18	e_3bDe_4	$a\ a\ b\ b\ b\ s_0$ e_4
19	$e_4s_0be_5$	$a\ a\ b\ b\ b\ b$ e_5
20	e_5bIe_5	$a\ a\ b\ b\ b\ b$ e_5
21	e_5bIe_5	$a\ a\ b\ b\ b\ b$ e_5
22	e_5bIe_5	$a\ a\ b\ b\ b\ b$ e_5
23	e_5bIe_5	$a\ a\ b\ b\ b\ b$ e_5
24	e_5aDe_1	$a\ a\ b\ b\ b\ b$ e_1
25	e_1bbe_6	$a\ a\ b\ b\ b\ b$ e_6
26	e_6bIe_6	$a\ a\ b\ b\ b\ b$ e_6
27	e_6aIe_6	$a\ a\ b\ b\ b\ b$ e_6

MOVIMIENTO	*ACCION*	*CINTA* *ESTADO*
28	e_6aIe_6	s_0 a a b b b b e_6
29	$e_6s_0De_7$	s_0 a a b b b b e_7
30	e_7a1e_7	s_0 1 a b b b b e_7
31	e_71De_7	s_0 1 a b b b b e_7
32	e_7a1e_7	s_0 1 1 b b b b e_7
33	e_71De_7	s_0 1 1 b b b b e_7
34	e_7b1e_7	s_0 1 1 1 b b b e_7
35	e_71De_7	s_0 1 1 1 b b b e_7
36	e_7b1e_7	s_0 1 1 1 1 b b e_7
37	e_71De_7	s_0 1 1 1 1 b b e_7
38	e_7b1e_7	s_0 1 1 1 1 1 b e_7
39	e_71De_7	s_0 1 1 1 1 1 b e_7
40	e_7b1e_7	s_0 1 1 1 1 1 1 e_7
41	e_71De_7	s_0 1 1 1 1 1 1 s_0 e_7

La máquina se para puesto que no hay ningun cuádruple que comience por e_7s_0. Por tanto, ha finalizado la computación y el resultado que ofrece es 6 '1', tal y como cabía esperar.

Ejemplo 3. La máquina que aquí presentamos nos permite "traducir" un número natural a su expresión por marcas, esto es, si en la cinta de la máquina escribimos un número natural cualquiera en el sistema decimal y hacemos que la máquina actúe, aparece escrito en su cinta tantos "1" como el número que hayamos introducido, estando el resto de las casillas vacías. En las etapas intermedias se deja una casilla en blanco entre los lugares reservados para los dígitos del número y la correspondiente sucesión de "1" que se va generando a la derecha de la casilla vacía indicada.

El proceso que vamos a seguir para diseñar una máquina que logre este objetivo se divide en varias etapas:

1) Localizamos la cifra de las unidades del número N.

2) Escribimos $N-1$ en las casillas reservadas a los dígitos de N y el "1" que hemos restado lo escribimos a la derecha dejando una casilla sin ningún símbolo entre las cifras de $N-1$ y este "1", para evitar confusiones posteriores.

3) Examinamos si el número que aparece escrito a la izquierda de la casilla en blanco es o no 0. Si lo es, se sustituye "0" por s_0 y se acaba el proceso. En caso contrario, se pasa al siguiente paso.

4) Localizamos la cifra de las unidades del número que está escrito a la izquierda de la casilla vacía.

5) Restamos una unidad a dicho número y añadimos un "1" a la sucesión de éstos.

6) Reiteramos los pasos 3), 4) y 5).

La máquina que buscamos tiene por símbolos $S = \{0, 1, \ldots, 9, s_0\}$, estados $E = \{e_0, \ldots e_7\}$ y cuádruples:

$$\begin{gathered}\mathfrak{C} = \{e_00De_0, e_01De_0, e_02De_0, e_03De_0, e_04De_0, e_05De_0, e_06De_0,\\ e_07De_0, e_08De_0, e_09De_0, e_0s_0Ie_1, e_198e_2, e_187e_2, e_176e_2,\\ e_165e_2, e_154e_2, e_143e_2, e_132e_2, e_121e_2, e_110e_2e_109e_7, e_79Ie_1,\\ e_20De_2, e_21De_2, e_22De_2, e_23De_2, e_24De_2, e_25De_2, e_26De_2,\\ e_27De_2, e_28De_2, e_29De_2, e_2s_0De_3, e_31De_3, e_3s_01e_4, e_41Ie_4,\\ e_4s_0Ie_5, e_50Ie_5, e_511e_0, e_522e_0, e_533e_0, e_544e_0, e_555e_0,\\ e_566e_0, e_577e_0, e_588e_0, e_599e_0, e_5s_0De_6, e_60s_0e_7, e_7s_0De_6\}\end{gathered}$$

Para que la máquina funcione de manera adecuada, debe encontrarse en el estado e_0 con su cabeza situada sobre alguna de las cifras del número natural que hayamos anotado en la cinta.

Esta última máquina en combinación con las anteriores nos sirve para construir otras máquinas de Turing que multipliquen números naturales por un número fijo. Así,

Fase 1: Convertir el número natural en su expresión por marcas.

Fase 2: Aplicar las mismas acciones que conducen a la multilplicación por 3 (ó en general por k).

Fase 3: Contar cuántos "1" aparecen en el resultado obtenido en la fase 2 y anotar en la cinta dicho número.

Únicamente se precisará ser un poco cuidadoso con los índices de los estados, de manera que encajen adecuadamente (habrá que realizar un corrimiento en dichos índices).