

Tema 2: Máquinas de Turing

1. Antecedentes históricos.

En el siglo XVII, Leibniz planteó la búsqueda de un procedimiento efectivo que permitiera resolver cualquier problema matemático; esto es, se intentaba buscar un método general que nos diera la solución a un problema arbitrario. A principios del siglo XX, David Hilbert resumió la búsqueda de este método general en una conjetura que afirmaba que las Matemáticas en su conjunto podían axiomatizarse. Intentó buscar un algoritmo que permitiese determinar la validez o no de cualquier proposición matemática. Sin embargo, ante los infructuosos intentos de construcción, se empezó a pensar que quizás el procedimiento buscado no existía, es decir, que era posible la existencia de problemas para los cuales no era factible crear un algoritmo resolutivo. Siguiendo esta nueva línea, en 1931 el lógico austriaco Kurt Gödel publicó un artículo en el que se recogía su famoso teorema de incompletitud, que probaba la no existencia del procedimiento efectivo buscado. Así, Gödel construyó una fórmula en el cálculo de predicados aplicada a los enteros cuya definición establecía que no podía ser probada su validez o falsedad dentro del sistema lógico. Esto constituyó un gran avance para la lógica matemática, pues permitió formalizar la noción intuitiva de procedimiento efectivo y abandonar los intentos de construcción del algoritmo general que se habían realizado hasta entonces.

En este marco surgieron las máquinas de Turing que eran un mecanismo que permitían resolver problemas concretos. Simultáneamente, Church, Kleene y Post presentaron otros modelos que tenían una capacidad de computación semejante al modelo de Turing, aunque este último ha sido el más conocido e investigado.

Para diseñar su máquina, Turing se basó en las calculadoras que en aquella época existían. Gracias a ellas, se dió a la computabilidad un significado más concreto y preciso y se describió lo que hoy conocemos como **algoritmo**: una serie de instrucciones que al seguir las resuelven el problema para el que han sido creadas. De manera rápida, podemos visualizarlas como una computadora idealizada en la que las operaciones elementales se descomponen hasta el límite. De hecho, se pueden considerar el dispositivo teórico precursor de los computadores actuales, con dos salvedades: la cantidad de memoria que dispone es infinita y las operaciones más simples se descomponen lo más posible. Precisamente, esta última cuestión es la que marca las diferencias más claras entre los ordenadores y el dispositivo teórico creado por Turing: entre otros, los objetivos que presentan los computadores actuales es el de procesamiento de la información de la manera más rápida posible y la realización de complejas operaciones matemáticas con el menor coste operativo y con la simplicidad de la máquina de Turing el coste operativo que se tiene es alto, si entendemos por tal el número de pasos que se deben dar para conseguir la solución deseada.

Por último, para finalizar este párrafo debemos mencionar que ideas desarrolladas por Turing en su concepción de las máquinas de Turing son en la actualidad empleadas en programación. Así, la composición en paralelo de máquinas que presentó Turing es similar a la que hoy se está empleando en programación. Por otro lado, sería erróneo pensar que la única finalidad de las máquinas de Turing se centra en ser dispositivos teóricos cuyas ideas sirvieron para diseñar computadores. También han sido empleados en la lógica matemática para demostrar resultados como el Teorema de Markov-Post, relacionadas con los subconjuntos recursivamente enumerables.

2. Descripción de una máquina de Turing.

Intuitivamente, podemos interpretar una máquina de Turing como cabeza lectora-inscriptora a través de la cual pasa una cinta que consta de una serie de casillas contiguas y es tan larga a derecha y a izquierda como se desee. La cabeza lectora-inscriptora es capaz de imprimir un número finito de símbolos s_0, s_1, \dots, s_m y de estar en un número finito de estados e_0, e_1, \dots, e_n . En un momento dado, la cabeza se encuentra sobre una casilla determinada que contiene un único símbolo s_j en un estado e_i y dependiendo de las instrucciones que posea realizará una de las siguientes acciones: moverse una casilla a la derecha, moverse una casilla a la izquierda, cambiar el símbolo s_j por s_k ó parar. Opcionalmente, en las tres primeras posibilidades variará su estado interno y continuará el proceso descrito.

Procedemos a dar la definición formal de máquina de Turing. Salvo que se indique lo contrario, trabajaremos con $S = \{s_0, s_1, \dots, s_m\}$ conjunto finito cuyos elementos denominamos símbolos y con $E = \{e_0, \dots, e_n\}$ conjunto finito cuyos elementos llamamos estados.

Necesitamos introducir previamente la noción de cuádruple:

Definición. Un **cuádruple** es una 4-tupla de uno de los tres tipos siguientes:

- i) $e_i s_j s_k e_l$
- ii) $e_i s_j D e_l$
- iii) $e_i s_j I e_l$,

donde $e_i, e_l \in E$, $s_j, s_k \in S$ y D e I denotan derecha e izquierda, respectivamente.

Por abuso del lenguaje, hemos omitido el símbolo $(, , ,)$, entendiendo que dos expresiones del tipo anterior son iguales si y sólo si tienen iguales sus componentes. Cada uno de estos cuádruples corresponden a un tipo de movimiento de la descripción intuitiva dada anteriormente. Esto es:

- i) $e_i s_j s_k e_l$ se interpreta de la manera siguiente: al examinar el símbolo s_j en el estado e_i , borra el s_j , imprime s_k y entra en el estado e_l .
- ii) $e_i s_j D e_l$ significa: cuando examines el símbolo s_j en el estado e_i , mueveté una casilla hacia la derecha y pasa al estado e_l .
- iii) $e_i s_j I e_l$ significa: cuando examines el símbolo s_j en el estado e_i , mueveté una casilla hacia la izquierda y pasa al estado e_l .

Dos cuádruples serán distintos cuando difieran en una de las componentes del mismo.

Definición. Sean $S = \{s_0, \dots, s_m\}$, $E = \{e_0, \dots, e_n\}$ y \mathfrak{C} tres conjuntos finitos no vacíos cuyos elementos son símbolos, estados y cuádruples, respectivamente. Diremos que $\mathfrak{T} = (S, E, \mathfrak{C})$ es una **máquina de Turing** si \mathfrak{C} satisface la propiedad siguiente: no existen dos cuádruples distintos en \mathfrak{C} que comiencen por $e_i s_j$ para algunos $i \in \{0, \dots, n\}$ y $j \in \{0, \dots, m\}$.

Equivalentemente, una máquina de Turing consiste en dar una correspondencia

$$f: E \times S \longrightarrow (S \cup \{I, D\}) \times E$$

$$(e_i, s_j) \longmapsto f(e_i, s_j)$$

que cumple que a cada (e_i, s_j) le corresponde a lo sumo un $f(e_i, s_j)$.

Observemos que:

- i) El conjunto imagen de esta correspondencia consiste de elementos del tipo (s_k, e_l) , (D, e_l) ó (I, e_l) .
- ii) La correspondencia anterior no es en general una aplicación en el sentido que conocemos, puesto que no todos los elementos de $E \times S$ tienen una imagen por f .

La condición de que no existan dos cuádruples que comiencen con el mismo par de elementos de $E \times S$, esto es, para cada $(e_i, s_j) \in E \times S$ se verifica $|\{f(e_i, s_j)\}| \leq 1$, supone que en cada instante el movimiento a realizar por la máquina está totalmente determinado, no cabe ambigüedad sobre lo que tiene que hacer.

La máquina se parará si se encuentra en un estado e_i examinando un símbolo s_j y no existe ningún cuádruple que comience por $e_i s_j$.

Ejemplo. Se consideran los conjuntos $S = \{s_0, s_1, s_2, s_3\}$ y $E = \{e_0, e_1, e_2\}$. Estos conjuntos junto con los siguientes cuádruples:

$$\mathfrak{C} = \{e_0 s_0 D e_1, e_1 s_1 s_2 e_2, e_0 s_1 I e_2, e_2 s_0 I e_2, e_0 s_2 I e_1, e_2 s_3 s_1 e_0\}$$

constituyen una máquina de Turing.

Interpretando la máquina anterior como una correspondencia tenemos:

$$\begin{array}{lcl}
 f : E \times S & \longrightarrow & (S \cup \{I, D\}) \times E \\
 (e_0, s_0) & \longmapsto & (D, e_1) \\
 (e_0, s_1) & \longmapsto & (I, e_2) \\
 (e_0, s_2) & \longmapsto & (I, e_1) \\
 (e_1, s_1) & \longmapsto & (s_2, e_2) \\
 (e_2, s_0) & \longmapsto & (I, e_2) \\
 (e_2, s_3) & \longmapsto & (s_1, e_0)
 \end{array}$$

También podemos visualizar las acciones de una máquina de Turing mediante una tabla. Así, los cuádruples del ejemplo anterior vienen dados por:

	s_0	s_1	s_2	s_3
e_0	De_1	Ie_2	Ie_1	
e_1		s_2e_2		
e_2	Ie_2			s_1e_0

Para elaborar este cuadro hemos utilizado el siguiente criterio: Cuando la máquina se encuentre en el estado e_i examinando una casilla que lleve el símbolo s_j , en la casilla (i, j) correspondiente aparece el movimiento a realizar. Si no hay nada, significa que la máquina se para.

Del hecho de que para cada par (e_i, s_j) pueda haber a lo sumo un cuádruple que comience por él, se sigue que podemos contar el número de máquinas de Turing distintas que se pueden dar con un mismo conjunto de símbolos y estados, entendiendo que dos máquinas serán diferentes si presentan, al menos, un cuádruple distinto.

Teorema 2.1. *El número de máquinas de Turing diferentes que se pueden crear con m símbolos y n estados es $(n(m+2)+1)^{nm} - 1$,*

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 $\alpha = Pe_i s_j P'$, 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 $\alpha = s_2 e_3 s_1 s_5 s_7 s_2$ la interpretamos: “los símbolos de la cinta son $s_2 s_1 s_5 s_7 s_2$ 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 $\alpha = P e_i s_j P'$ una descripción instantánea de \mathfrak{T} . Decimos que α es **terminal** si no existe un cuádruple en \mathfrak{C} que comience por $e_i s_j$.

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

- i) $\alpha = P e_i s_j P'$
 $\beta = P e_l s_k P'$ y el cuádruple $e_i s_j s_k e_l \in \mathfrak{C}$
- ii) $\alpha = P e_i s_j s_k P'$
 $\beta = P s_j e_l s_k P'$ y el cuádruple $e_i s_j D e_l \in \mathfrak{C}$
- iii) $\alpha = P e_i s_j$
 $\beta = P s_j e_l s_0$ y el cuádruple $e_i s_j D e_l \in \mathfrak{C}$
- iv) $\alpha = P s_k e_i s_j P'$
 $\beta = P e_l s_k s_j P'$ y el cuádruple $e_i s_j I e_l \in \mathfrak{C}$
- v) $\alpha = e_i s_j P'$
 $\beta = e_l s_0 s_j P'$ y el cuádruple $e_i s_j I e_l \in \mathfrak{C}$.

Diremos que $\alpha \rightarrow \beta$ 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 $\alpha_1, \alpha_2, \dots, \alpha_t$ tales que $\alpha_i \rightarrow \alpha_{i+1}$ para $i = 1, 2, \dots, t-1$ 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 $\alpha_i \rightarrow \alpha_{i+1}$ 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 determinada 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”, moviéndonos 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 atrás 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, devolviéndonos el número de “1” que había en la sucesión.

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

$$\begin{aligned} \mathcal{C} = \{ & e_0 1 I e_0, e_0 s_0 I e_1, e_1 s_0 0 e_2, e_2 0 D e_2, e_2 1 D e_2, e_2 2 D e_2, e_2 3 D e_2, e_2 4 D e_2, \\ & e_2 5 D e_2, e_2 6 D e_2, e_2 7 D e_2, e_2 8 D e_2, e_2 9 D e_2, e_2 s_0 D e_3, e_3 a D e_3, e_3 1 a e_4, \\ & e_4 a I e_4, e_4 s_0 I e_5, e_5 0 1 e_2, e_5 1 2 e_2, e_5 2 3 e_2, e_5 3 4 e_2, e_5 4 5 e_2, e_5 5 6 e_2, e_5 6 7 e_2, \\ & e_5 7 8 e_2, e_5 8 9 e_2, e_5 9 0 e_6, e_6 0 I e_5, e_5 s_0 1 e_2, e_3 s_0 I e_7, e_7 a s_0 e_8, e_8 s_0 I e_7 \} \end{aligned}$$

Veamos cómo actúa la máquina diseñada en un ejemplo concreto:

Proyecto OCW de la UPV/EHU. M.A.García y T. Ramírez

<i>MOVIMIENTO</i>	<i>ACCION</i>	<i>CINTA ESTADO</i>
Inicio	-	1 1 e ₀
1	e ₀ 1Ie ₀	s ₀ 1 1 e ₀
2	e ₀ s ₀ Ie ₁	s ₀ s ₀ 1 1 e ₁
3	e ₁ s ₀ 0e ₂	0 s ₀ 1 1 e ₂
4	e ₂ 0De ₂	0 s ₀ 1 1 e ₂
5	e ₂ s ₀ De ₃	0 s ₀ 1 1 e ₃
6	e ₃ 1ae ₄	0 s ₀ a 1 e ₄
7	e ₄ aIe ₄	0 s ₀ a 1 e ₄
8	e ₄ s ₀ Ie ₅	0 s ₀ a 1 e ₅
9	e ₅ 0Ie ₂	1 s ₀ a 1 e ₂
10	e ₂ 1De ₂	1 s ₀ a 1 e ₂
11	e ₂ s ₀ De ₃	1 s ₀ a 1 e ₃
12	e ₃ aDe ₃	1 s ₀ a 1 e ₃
13	e ₃ 1ae ₄	1 s ₀ a a e ₄
14	e ₄ aIe ₄	1 s ₀ a a e ₄
15	e ₄ aIe ₄	1 s ₀ a a e ₄
16	e ₄ s ₀ Ie ₅	1 s ₀ a a e ₅
17	e ₅ 12e ₂	2 s ₀ a a e ₂
18	e ₂ 2De ₂	2 s ₀ a a e ₂
19	e ₂ s ₀ De ₃	2 s ₀ a a e ₃
20	e ₃ aDe ₃	2 s ₀ a a e ₃

<i>MOVIMIENTO</i>	<i>ACCION</i>	<i>CINTA ESTADO</i>
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 examinando 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

$$\mathcal{C} = \{e_11ae_2, e_2aDe_2, e_21De_2, e_2bDe_2, e_2s_0be_3, e_3bDe_4, \\ e_4s_0be_5, e_5bIe_5, e_51Ie_5, e_5aDe_1, e_1bbe_6,$$

$$e_6aIe_6, e_6bIe_6, e_6s_0De_7, e_7a1e_7, e_71De_7, e_7b1e_7\}$$

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

<i>MOVIMIENTO</i>	<i>ACCION</i>	<i>CINTA ESTADO</i>
Inicio	–	1 1 e ₀
1	e ₀ 1Ie ₀	1 1 e ₀
2	e ₀ s ₀ De _e	1 1 e ₁
3	e ₁ 1ae ₂	a 1 e ₂
4	e ₂ aDe ₂	a 1 e ₂
5	e ₂ 1De ₂	a 1 s ₀ e ₂
6	e ₂ s ₀ be ₃	a 1 b e ₃
7	e ₃ bDe ₄	a 1 b s ₀ e ₄
8	e ₄ s ₀ be ₅	a 1 b b e ₅
9	e ₅ bIe ₅	a 1 b b e ₅
10	e ₅ bIe ₅	a 1 b b e ₅
11	e ₅ 1Ie ₅	a 1 b b e ₅
12	e ₅ aDe ₁	a 1 b b e ₁
13	e ₁ 1ae ₂	a a b b e ₂
14	e ₂ aDe ₂	a a b b e ₂
15	e ₂ bDe ₂	a a b b e ₂
16	e ₂ bDe ₂	a a b b s ₀ e ₂
17	e ₂ s ₀ be ₃	a a b b b e ₃
18	e ₃ bDe ₄	a a b b b s ₀ e ₄

3. Diseño de máquinas de Turing

<i>MOVIMIENTO</i>	<i>ACCION</i>	<i>CINTA ESTADO</i>
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
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_7aIe_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_7aIe_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_7bIe_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_7bIe_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_7bIe_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_7bIe_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 ningún 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 $S = \{0, 1, \dots, 9, s_0\}$, estados $E = \{e_0, \dots, e_7\}$ y cuádruples:

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

3. Diseño de máquinas de Turing

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 multiplicación 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).