Máquina de Turing: Multiplicación de dos números

Introducción

En Teoría de la computabilidad, la **recursión primitiva** permite definir una clase de funciones que forman un importante paso en la formalización de la noción de computabilidad. Se definen usando como principales operaciones la recursión y composición de funciones y forman un subconjunto estricto de las funciones recursivas, que son precisamente las funciones computables. Las funciones recursivas se definen agregándole a la recursión primitiva el operador de búsqueda no acotada que permite definir funciones parciales.

La operación de una recursión primitiva se muestra en la siguiente figura.

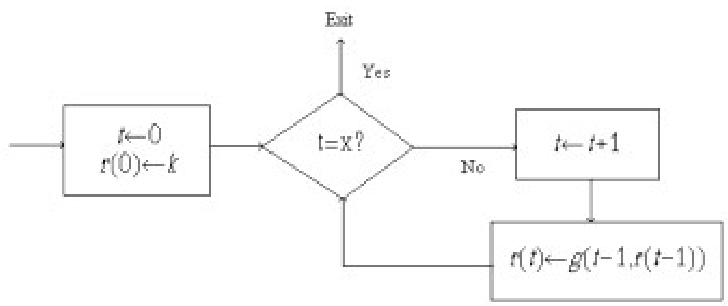


Figura 1. Recursión Primitiva.

Ejemplos de funciones recursivas primitivas son la suma y la multiplicación:

1. $\sum(x,y)=x+y$

$$\begin{cases} \sum(0,y)=y \\ \sum(x+1,y)=S\left(\sum(x,y)\right) \end{cases}$$

2. $\Pi(x,y)=x\cdot y$

$$\begin{cases} \Pi(0,y)=0 \\ \Pi(x+1,y)=\Pi(x,y)+y=\sum\left(\Pi(x,y),U_2^2(x,y)\right) \end{cases}$$

Planteamiento de la multiplicación para la máquina de Turing

Si analizamos un ejemplo de una suma, usando funciones recursivas parciales tenemos que se mandan llamar operaciones de sucesión:

$\sum(2,3)$

Sucesión del anterior

$\sum(0,3) = 3$
$\sum(1,3) = S(\sum(0,3)) = 4$
$\sum(2,3) = S(\sum(1,3)) = 5$

Con este ejemplo de suma podemos ver como trabajaría la cinta de la Máquina de Turing cumpliendo con la recursividad:

****x* 0 *y* 0 contador***

Sucesión de 111

1110111101
11110111101 1
11111011110111
1111110111101111

Si y ≠ contador, se incrementa contador en uno

x = 5
Resultado

Donde 1 = 0, 11=1, 111 = 2, 1111 = 3…

Mientras "*y*" sea diferente del *contador* se aplicará la sucesión de "*x*" (dato anterior) y se incrementará en uno el "*contador*", hasta que "*y*" y *contador* sean iguales se terminará la operación dando como resultado *x*. Por lo tanto tenemos que $\sum(2,3) = 5$.

Si nos planteamos como se puede llevar a cabo una multiplicación utilizando el principio anterior de la suma, es decir, seguir utilizando funciones recursivas parciales tenemos que para $\prod(3,4) = 12$ la operación la siguiente:

$\prod(0,4) = 0$
$\prod(1,4) = 0 + 4 = \sum(0,4) = 4$
$\prod(2,4) = 4 + 4 = \sum(4,4) =$ $\quad \sum(0,4) = 4$
$\sum(1,4) = S(\sum(0,4)) = 5$
$\sum(2,4) = S(\sum(1,4)) = 6$
$\sum(3,4) = S(\sum(2,4)) = 7$
$\sum(4,4) = S(\sum(3,4)) = 8$
$\prod(3,4) = 8 + 4 = \sum(8,4) =$ $\quad \sum(0,4) = 4$
$\sum(1,4) = S(\sum(0,4)) = 5$
$\sum(2,4) = S(\sum(1,4)) = 6$
$\sum(3,4) = S(\sum(2,4)) = 7$
$\sum(4,4) = S(\sum(3,4)) = 8$
$\sum(5,4) = S(\sum(4,4)) = 9$
$\sum(6,4) = S(\sum(5,4)) = 10$
$\sum(7,4) = S(\sum(6,4)) = 11$
$\sum(8,4) = S(\sum(7,4)) = 12$

Vemos que las sumatorias se pueden resolver como la anterior máquina de Turing de la suma, sólo resta controlar la multiplicación.

En una máquina de Turing podríamos controlar lo anterior de la siguiente manera, cumpliendo con el requisito de recursividad:

Para el producto de *X* y *Y* tendríamos el siguiente formato:

$* \; x \; 0 \; contX \; 0 \; resultado \; 0 \; y \; 0 \; contY \; *$

En la parte de $0 \; res \; 0 \; y \; 0 \; contY \; *$ se controla la suma, y en la parte de $* \; x \; 0 \; contX \; 0$ se controla que hasta que x y $contX$ sean iguales se terminará el proceso, dando por terminado el proceso de multiplicación.

Primero se debe hacer la comparación por parte de x y $contX$ y si son desiguales entonces se hará la comparación de y y $contY$ (como dijimos anteriormente, controla la suma).

Pero para simplificarnos los posibles corrimientos de los datos que tendríamos que hacer en el anterior formato de entrada de la cinta, **es conveniente que de entrada solo se tengan los dos números a multiplicar** y posteriormente inicializar la cinta como se muestra a continuación:

Siguiendo el ejemplo anterior de ∏(3,4) = 12

x 0 y

1110111111 *Formato de entrada*

***contX 0 x 0 y 0 contY * 0..0 * resultado ***

*10111101111101*00000*1* *Inicialización*

El número de ceros debe ser igual a la cantidad de 1s en y. Se garantiza que no se sobrescribirá el resultado

Desarrollando la multiplicación tenemos:

2

1

*10111101111101*00000*1*

*101111011111011*0000*11*

*1011110111110111*000*111* 3

*10111101111101111*00*1111*

4 *101111011111011111*0*11111*

*110111101111101*00000*11111*

*1101111011111011*0000*111111*

*11011110111110111*000*1111111*

*110111101111101111*00*11111111*

*1101111011111011111*0*111111111*

*1110111101111101*00000*111111111*

*11101111011111011*0000*1111111111*

*111011110111110111*000*11111111111*

*1110111101111101111*00*111111111111*

*11101111011111011111*0*1111111111111*

5 *11110111101111101*00000*1111111111111*

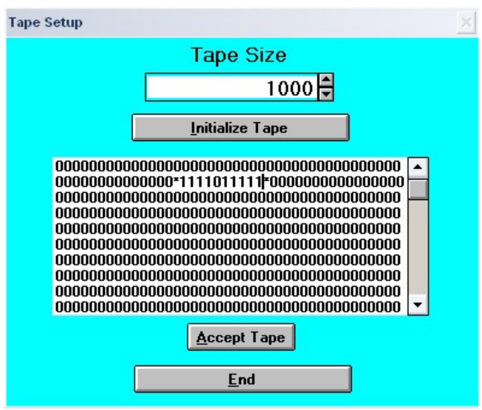
6 *111111111111*

En donde en

1) Si *contX* es diferente de “*x”*, comparamos “*y*” y *contY*
2) Si “*y*” y *contY* son diferentes, incrementamos en uno a *contY* y a *resultado*
3) Llevamos a cabo la operación de suma.
4) Si “*y*” y *contY* son iguales, incrementamos en uno a *contX*, inicializamos en cero a *contY* y volvemos a obtener el espacio de ceros de la inicialización.
--Se repiten los pasos del 1 al 4 hasta que *contX* y “*x*” sean iguales.--
5) Si *contX* y “*x*” son iguales, hemos terminado la multiplicación y *resultado* ya contiene el resultado final.
6) Se procede a limpiar la cinta para mostrar solo el resultado final.

<u>Funcionamiento de la Máquina de Turing de Multiplicación</u>

- Guardar el archivo MULX_Y.NTM en la misma carpeta que el simulador.
- Indicar el formato de entrada de la cinta: ***x 0 y*.*** Esto se hace al presionar TAPE SETUP.

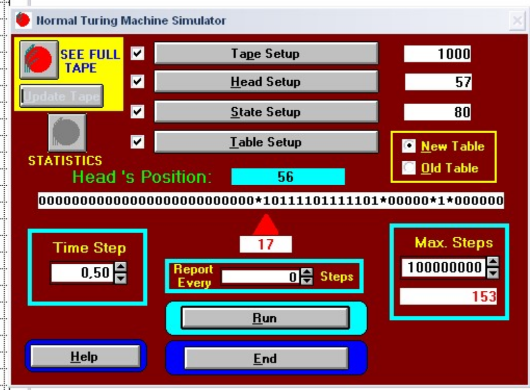


*Formato de entrada *x 0 y**

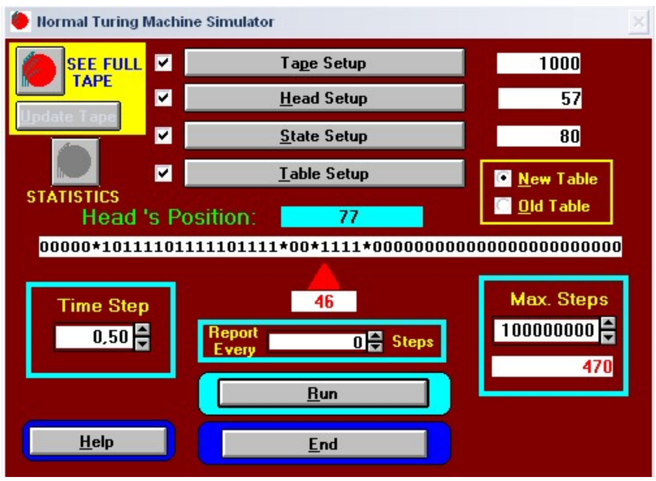
- Presionar el botón HEAD SETUP para colocar la cabeza a la derecha del primer asterisco de la cinta.
- Indicar el número de estados (como se va a llamar a un archivo previo no es necesario indicarlo). Presionar botón STATE SETUP.
- Presionar el botón TABLE SETUP para cargar el archivo MULX_Y.NTM.
- Aumentar el número máximo de pasos (MAX STEPS), por ejemplo 100000000.
- Si se desea observar paso a paso los movimientos en la cinta, aumentar el TIME STEP.
- Para correr la cinta se presiona el botón RUN.

Input=0 Input=1 Input=* Input=#

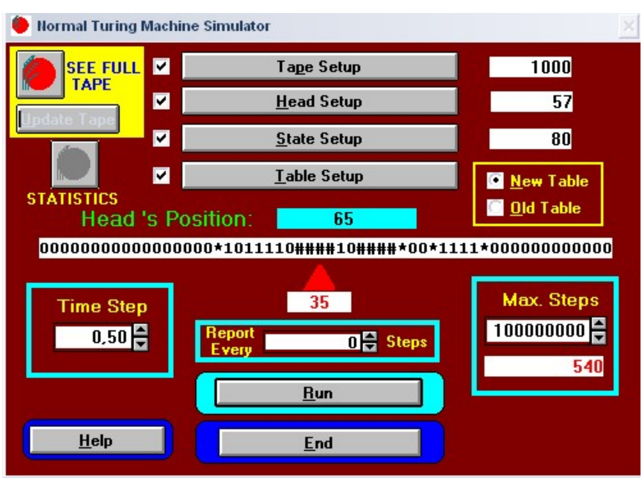
Present State	S1	Mov	N.S.	S2	Mov	N.S.	S3	Mov	N.S.	S4	Mov	N.S.
1	1	L	2	1	L	1	0	L	1	-	-	-
2	*	R	3	-	-	-	-	-	-	-	-	-
3	0	R	3	1	R	3	0	R	4	-	-	-
4	1	R	5	-	-	-	-	-	-	-	-	-
5	*	L	6	-	-	-	-	-	-	-	-	-
6	0	L	7	1	L	6	-	-	-	-	-	-
7	0	R	8	1	L	7	-	-	-	-	-	-
8	0	R	13	#	R	9	-	-	-	#	R	8
9	0	R	10	1	R	9	-	-	-	-	-	-
10	#	R	11	1	R	10	*	R	10	#	R	10
11	0	L	11	-	-	-	*	L	12	#	L	11
12	0	L	12	1	L	12	-	-	-	#	R	8
13	*	R	14	1	R	13	*	R	13	#	R	13
14	1	R	15	-	-	-	-	-	-	-	-	-
15	*	L	15	1	L	15	*	L	16	-	-	-
16	0	L	17	1	L	16	*	L	16	0	L	16
17	0	L	17	1	L	17	*	R	19	1	L	17
18	-	-	-	-	-	-	-	-	-	-	-	-
19	0	R	25	#	R	20	-	-	-	#	R	19
20	0	R	21	1	R	20	-	-	-	-	-	-
21	0	L	22	1	R	21	-	-	-	#	R	21
22	-	-	-	#	L	23	-	-	-	#	L	22
23	0	L	24	1	L	23	-	-	-	-	-	-
24	-	-	-	1	L	24	*	R	19	#	L	24
25	-	-	-	1	R	26	-	-	-	#	R	56
26	0	L	27	1	R	26	-	-	-	#	R	26
27	0	L	28	1	L	27	-	-	-	1	L	27
28	-	-	-	-	-	-	*	R	30	1	L	28
29	-	-	-	-	-	-	-	-	-	-	-	-
30	0	R	30	1	R	30	*	L	31	#	L	31
31	0	L	35	#	L	32	-	-	-	-	-	-
32	0	L	33	1	L	32	-	-	-	-	-	-
33	0	R	34	1	L	33	-	-	-	#	R	34
34	-	-	-	#	R	30	-	-	-	-	-	-
35	-	-	-	1	L	35	-	-	-	#	R	36
36	0	R	37	1	R	39	-	-	-	-	-	-



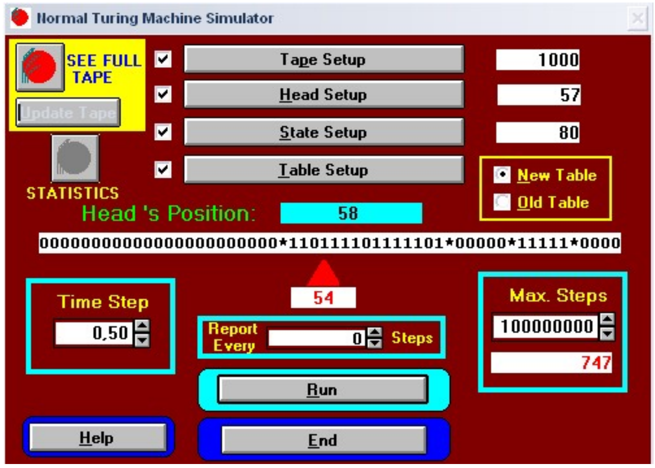
Inicialización de la cinta



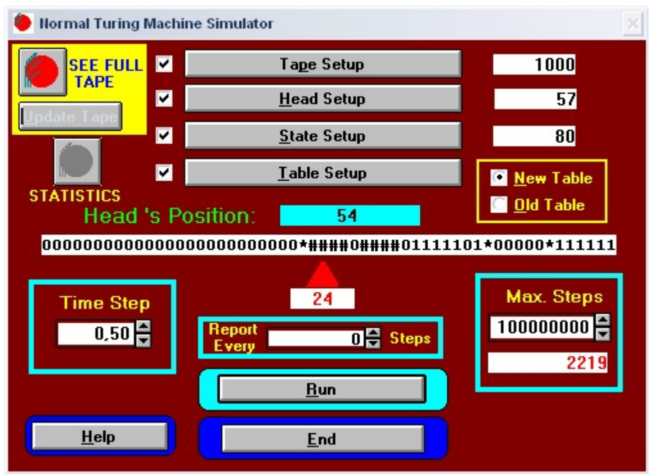
Incremento de contY y resultado



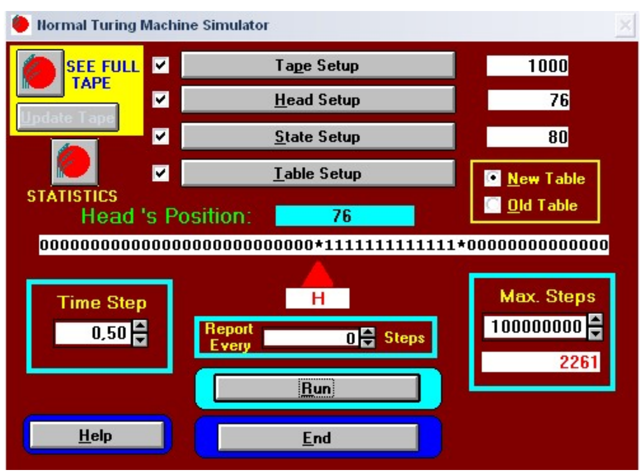
Comparación de "y" y contY



Incremento de contX e inicialización de contY



Comparación entre contX y "x"



Resultado final

Tabla obtenida:

Del estado 1 al 7 se lleva a cabo la inicialización de la cinta.
Del 19 al 28 se lleva a cabo la comparación de contX y “x”.
Del 30 al 40 se lleva a cabo la comparación de “y” y contY.
Del 42 al 47 se incrementa contY, se coloca el espacio de ceros, e incrementa en uno a resultado.
Del 49 al 54 se incrementa contX y se inicializa contY en cero.
Del 56 al 58 se procede a limpiar la cinta para dejar el resultado final.

EDO	SYM0	MOV0	STA0	SYM1	MOV1	STA1	SYM*	MOV*	STA*	SYM#	MOV#	STA#
1	1	L	2	1	L	1	0	L	1	-	-	-
2	*	R	3	-	-	-	-	-	-	-	-	-
3	0	R	3	1	R	3	0	R	4	-	-	-
4	1	R	5	-	-	-	-	-	-	-	-	-
5	*	L	6	-	-	-	-	-	-	-	-	-
6	0	L	7	1	L	6	-	-	-	-	-	-
7	0	R	8	1	L	7	-	-	-	-	-	-
8	0	R	13	#	R	9	-	-	-	#	R	8
9	0	R	10	1	R	9	-	-	-	-	-	-
10	#	R	11	1	R	10	*	R	10	#	R	10
11	0	L	11	-	-	-	*	L	12	#	L	11
12	0	L	12	1	L	12	-	-	-	#	R	8
13	*	R	14	1	R	13	*	R	13	#	R	13
14	1	R	15	-	-	-	-	-	-	-	-	-
15	*	L	15	1	L	15	*	L	16	-	-	-
16	0	L	17	1	L	16	*	L	16	0	L	16
17	0	L	17	1	L	17	*	R	19	1	L	17
18	-	-	-	-	-	-	-	-	-	-	-	-
19	0	R	25	#	R	20	-	-	-	#	R	19
20	0	R	21	1	R	20	-	-	-	-	-	-
21	0	L	22	1	R	21	-	-	-	#	R	21
22	-	-	-	#	L	23	-	-	-	#	L	22
23	0	L	24	1	L	23	-	-	-	-	-	-
24	-	-	-	1	L	24	*	R	19	#	L	24
25	-	-	-	1	R	26	-	-	-	#	R	56
26	0	L	27	1	R	26	-	-	-	#	R	26
27	0	L	28	1	L	27	-	-	-	1	L	27
28	-	-	-	-	-	-	*	R	30	1	L	28
29	-	-	-	-	-	-	-	-	-	-	-	-
30	0	R	30	1	R	30	*	L	31	#	L	31
31	0	L	35	#	L	32	-	-	-	-	-	-
32	0	L	33	1	L	32	-	-	-	-	-	-
33	0	R	34	1	L	33	-	-	-	#	R	34
34	-	-	-	#	R	30	-	-	-	-	-	-
35	-	-	-	1	L	35	-	-	-	#	R	36
36	0	R	37	1	R	39	-	-	-	-	-	-
37	0	R	37	1	R	37	*	L	38	#	R	37
38	0	L	38	1	L	38	*	R	49	1	L	38
39	0	R	39	1	R	39	*	L	40	#	R	39
40	0	L	40	1	L	40	*	R	42	1	L	40
41	-	-	-	-	-	-	-	-	-	-	-	-
42	0	R	42	1	R	42	1	R	43	-	-	-

43	*	R	44	-	-	-	-	-	-	-	-	-
44	0	R	44	1	R	45	*	R	44	-	-	-
45	*	L	46	1	R	45	1	R	45	-	-	-
46	0	L	47	1	L	46	*	L	46	-	-	-
47	0	L	47	-	-	-	*	L	30	-	-	-
48	-	-	-	-	-	-	-	-	-	-	-	-
49	*	R	50	1	L	49	1	L	49	-	-	-
50	0	R	50	1	R	50	0	L	51	-	-	-
51	0	R	52	0	L	51	-	-	-	-	-	-
52	1	R	53	-	-	-	-	-	-	-	-	-
53	*	L	54	-	-	-	-	-	-	-	-	-
54	0	L	54	1	L	54	*	R	19	-	-	-
55	-	-	-	-	-	-	-	-	-	-	-	-
56	0	L	56	1	L	56	0	R	57	#	L	56
57	0	R	57	0	R	57	0	R	58	0	R	57
58	0	R	58	-	-	-	*	R	H	-	-	-
59	-	-	-	-	-	-	-	-	-	-	-	-