

Curso de Lógica Matemática

Teoría de la Computabilidad

Cristo Daniel Alvarado

30 de octubre de 2024

Índice general

3. Conjuntos y Funciones computables	2
3.1. Máquinas de Turing	2
4. Teoremas de Completud	7

Capítulo 3

Conjuntos y Funciones computables

Todo de lo que se va a tratar esta parte es de: ¿Cómo formalizar la noción de *procedimiento mecánico, efectivo o sistemático*? Con esto nos referimos a:

- Tener un número finito de instrucciones.
- Terminar el procedimiento en un número finito de pasos.
- Usar únicamente *papel y lápiz*.
- No requiere razonamiento, solo se siguen reglas.

Básicamente se pretendía que dada una fórmula, encontrar un algoritmo que nos diga si esa fórmula es verdadera o falsa. Básicamente se pretendía formalizar las demostraciones para ver lo que nosotros podemos demostrar únicamente usando los axiomas.

Turing y Alonzo Church eventualmente se hicieron preguntas en la misma dirección. En la Tesis de Church-Turing se probó que estas tres preguntas en realidad se reducen a un mismo problema.

3.1. Máquinas de Turing

Definición 3.1.1

Una **máquina de Turing** consta de:

- Un *alfabeto*, un conjunto finito L .
- Un conjunto S de *estados*.
- Una función parcial $T : L^* \times S \rightarrow L^* \times S \times \{<, -, >\}$ llamada *función de transición*.

donde $L^* = L \cup \{*\}$.

Intuitivamente, uno debe imaginar que esto es una especie de *computadora rudimentaria*. Generalmente esto se conceptualiza como una cinta.

El cabezal c puede moverse a la derecha, izquierda o no moverse, dependiendo del estado en el que esté. En la Figura 3.1 se muestra que el hay al menos 5 diferentes estados, desde el estado inicial (s_i) hasta el final (s_f). Dependiendo de la entrada, la función T nos dirá lo que hará el cabezal, si cambia un elemento de la banda, si se mueve o si cambia de estado (o todas a la vez).

En este ejemplo, el alfabeto sería $L = \{0, 1\}$, el conjunto de estados es $S = \{s_i, s_1, \dots, s_f\}$ y la función sería representada por lo que sea que haga el cabezal.

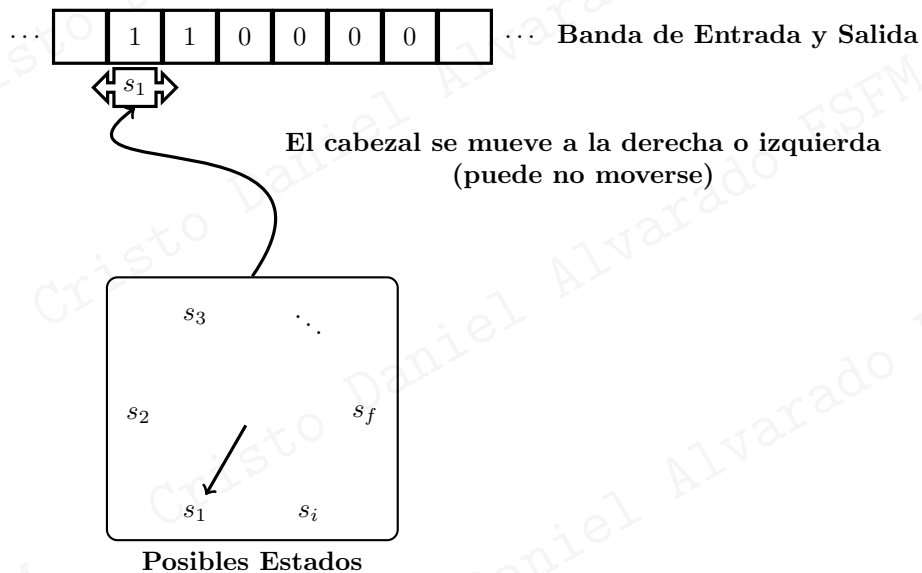


Figura 3.1: Ejemplo de Máquina de Turing

Ejemplo 3.1.1

Considere $L = \{1\}$, $S = \{s_i, s_1, s_2\}$ y,

$$T = \{(s_i, *, s_1, *, >), (s_i, 1, s_1, 1, >), (s_1, 1, s_1, 1, >), (s_1, 1, s_2, 1, -)\}$$

La cinta se ve más o menos así:

Para los siguientes ejercicios, ir a la página: [Simulador Máquina de Turing](#).

Ejercicio 3.1.1

Codifique una máquina de Turing que sume 1 a un número dado en binario.

```

1      name: Sumar uno en unario
2      init: s0
3      accept: sf
4
5
6      // Funciones de Transicion
7
8      s0, _
9      s0, _, >
10
11     s0, 1
12     s1, 1, -
13
14     s0, 0
15     s1, 0, -
16
17     s1, 1
18     s1, 0, >
19
20     s1, 0
21     s1, 1, >

```

```

22
23     s1,_
24     sf,_, -
25
26     // < = left
27     // > = right
28     // - = hold
29     // use _ for blank cells
30
31     // States and symbols are case-sensitive
32
33     // Load your code and click COMPILE.
34     // or load an example (top-right).

```

Ejercicio 3.1.2

Codifique una máquina de Turing que dada un número en binario, invierta su orientación, es decir, si la cadena es (a_1, \dots, a_n) , que la máquina de Turing la convierta en (a_n, \dots, a_1) .

```

1     name: invertirCadena
2     init: s0
3     accept: s1,sf,l,c,u
4
5     //esto para que se empiece a mover
6     s0,_
7     s0,_,>
8
9     s0,0
10    x,0,<
11
12    s0,1
13    x,1,<
14
15    x,_
16    s1,2,>
17
18    s1,0
19    s1,0,>
20
21    s1,1
22    s1,1,>
23
24    //logica cuando encuentre cosas
25
26    s1,_
27    s2,_,<
28
29    s2,_
30    s2,_,<
31
32    s2,0
33    c00,_,>

```

```

34
35     s2,1
36     u00,_,>
37
38     //mueve cosas al inicio
39
40     c00, _
41     m,0,<
42
43     u00, _
44     m,1,<
45
46     //ya en ciclo
47
48     //mueve derecha
49
50     m, _
51     l,_,<
52
53     l, _
54     l,_,<
55
56     l,0
57     c0,_,>
58
59     l,1
60     u0,_,>
61
62     c0, _
63     c0,_,>
64
65     //mueve izquierda
66
67     u0, _
68     u0,_,>
69
70     c0,0
71     c1,0,>
72
73     c0,1
74     c1,1,>
75
76     u0,0
77     u1,0,>
78
79     u0,1
80     u1,1,>
81
82     c1,0
83     c1,0,>
84
85     c1,1

```

```
86         c1,1,>
87
88         u1,0
89         u1,0,>
90
91         u1,1
92         u1,1,>
93
94         c1,_
95         m,0,<
96
97         u1,_
98         m,1,<
99
100        m,0
101        m,0,<
102
103        m,1
104        m,1,<
105
106        l,2
107        sf,_,>
108
109        sf,_
110        sf,_,>
111
112        sf,0
113        sff,0,-
114
115        sf,1
116        sff,1,-
```

Capítulo 4

Teoremas de Completud