



R E C O M E N D A C I O N E S
C O N T R O L



Economía del lenguaje: Las palabras no son gratis, cuestan esfuerzo tanto para el que la escribe como para el que lo lee.

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32.

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.



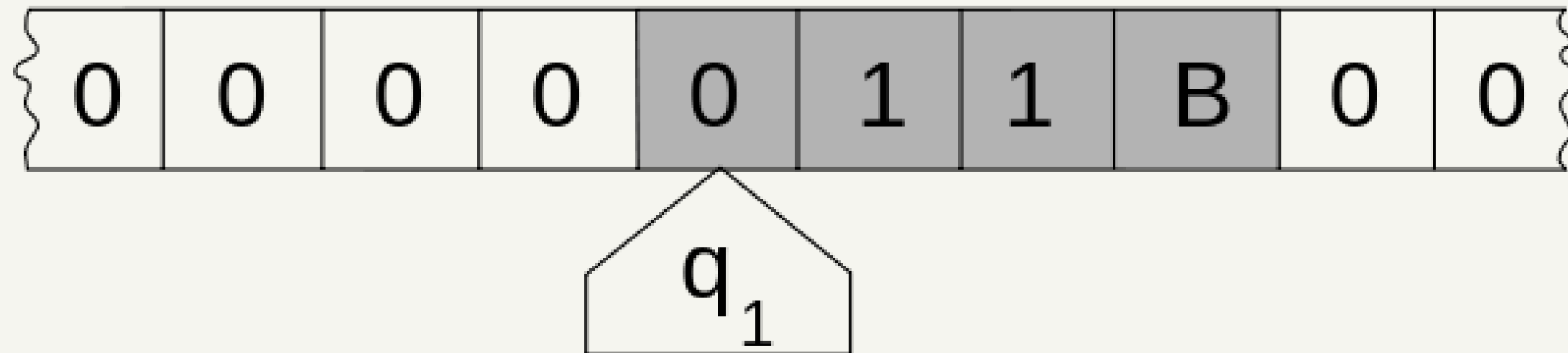
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s

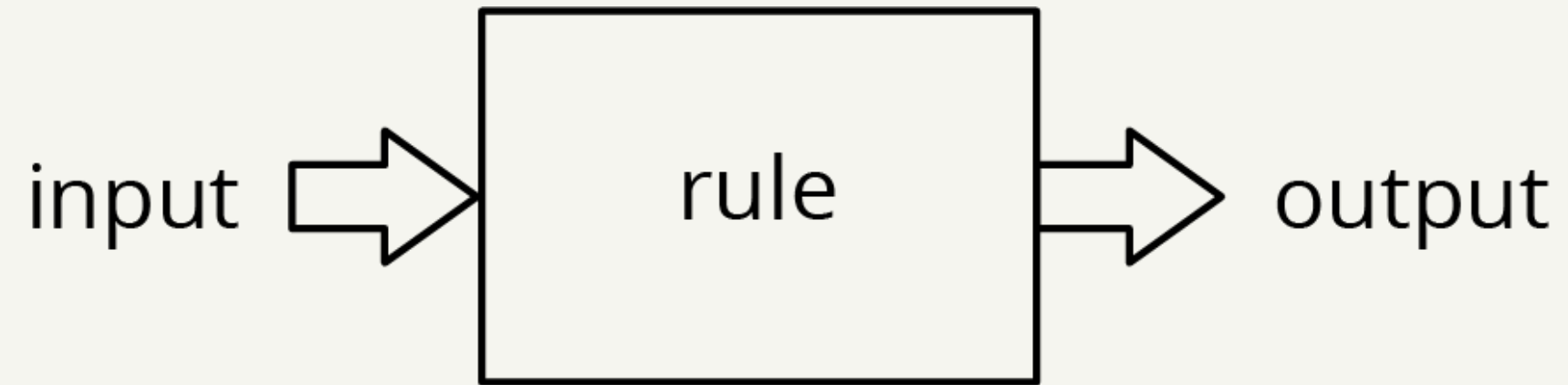
¿CÓMO EVITARLO?

- Preguntas puntuales (Ej: Cual, Donde, Cuando) == Respuesta Corta.
- Lee la pregunta antes, durante y después de responder, se te puede haber ido el hilo y no respondiste lo que te preguntaban.
- Si te estas demorando, deja la pregunta para el final. La inspiración vendrá sola.
- Escribe un párrafo por cada “¿?” o idea de la pregunta que encuentres. Facilita la escritura y revisión.
- Toda pregunta tiene un concepto clave, no tienes porque escribir todas las ideas que se mencionan. No recitar datos del texto si no te los piden.



M Á Q U I N A S D E T U R I N G





¿QUÉ ES UNA MÁQUINA DE TURING?

- Mecanismo de cómputo
- Input - Procesamiento - Respuesta

DEFINICIÓN INFORMAL



- Modelo matemático teórico que define una máquina que opera sobre una cinta infinita en base a un set de instrucciones.
- En cada paso, el cabezal lee un símbolo sobre la cinta (input).
- Según el estado actual de la máquina y el símbolo que se lee, se determinará si se mantiene o reemplaza el símbolo, la dirección en la que se moverá el cabezal y el nuevo estado de la máquina.

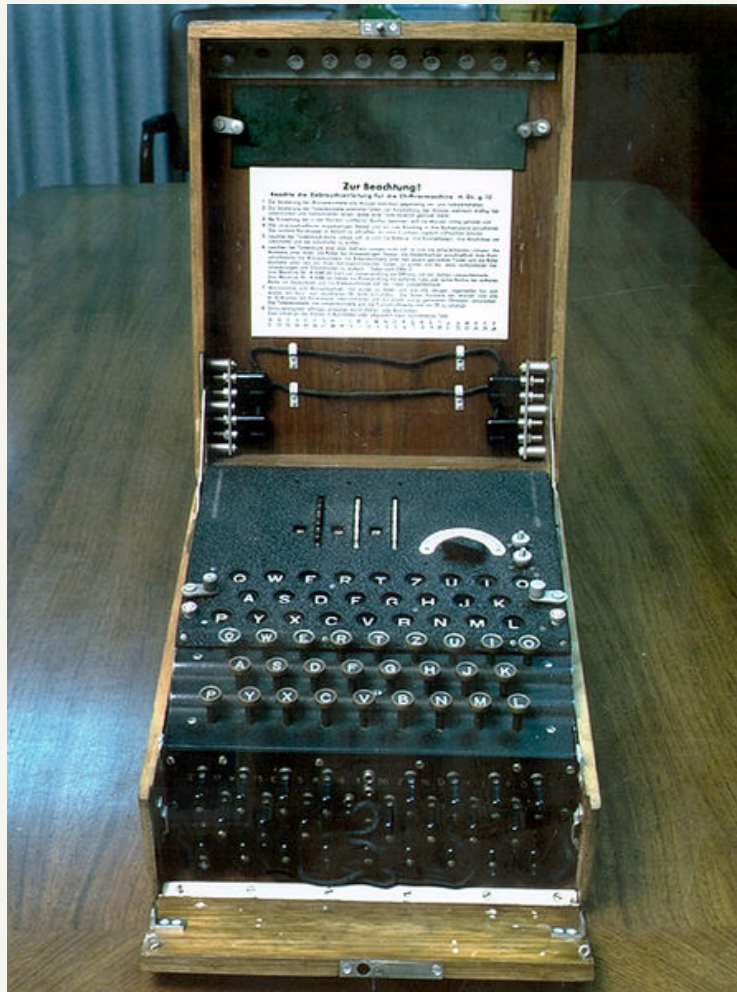
ESTRUCTURA DE LAS INSTRUCCIONES

(TRANSICIONES)

EstadoActual, SímboloQueSeLee
EstadoSiguiente, NuevoSimbolo, Dirección

- Si no hay una instrucción definida para el par símbolo estado, la máquina se detendrá.
- Si el estado en el que se detiene la máquina es un estado de aceptación, retornará TRUE.
- En otro caso, retornará FALSE.

DEFINICIÓN FORMAL

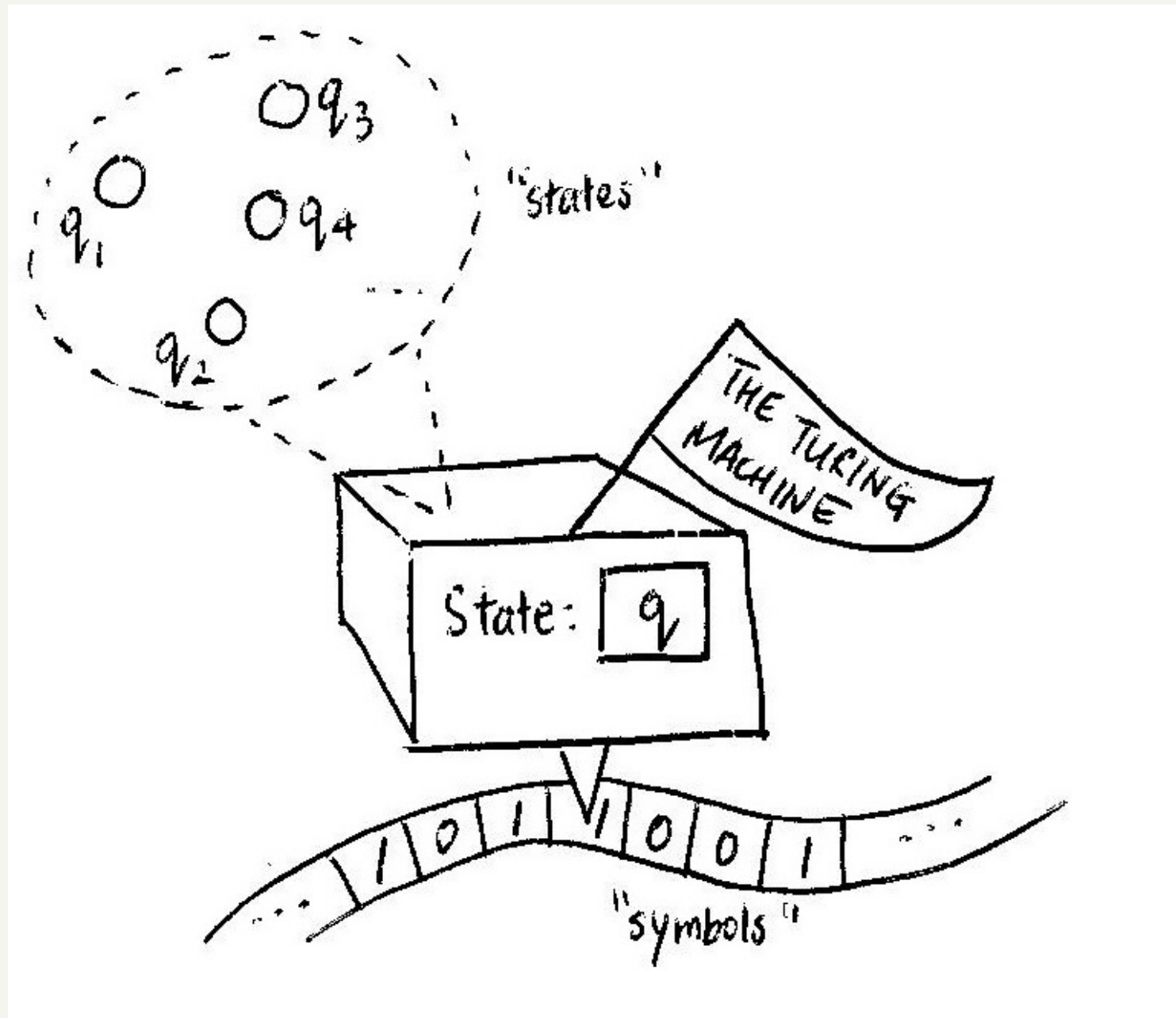


Una máquina de Turing es una
tupla $M = (Q, \Gamma, q_0, \delta, F)$

Donde:

- Q es un conjunto de estados.
- Γ es el alfabeto de la máquina.
- $\Sigma \subsetneq \Gamma$ es el alfabeto de entrada.
- $q_0 \in Q$ es el estado inicial.
- δ es una función de transición
- $F \subseteq Q$ es un conjunto de estados finales.

IMPORTANCIA

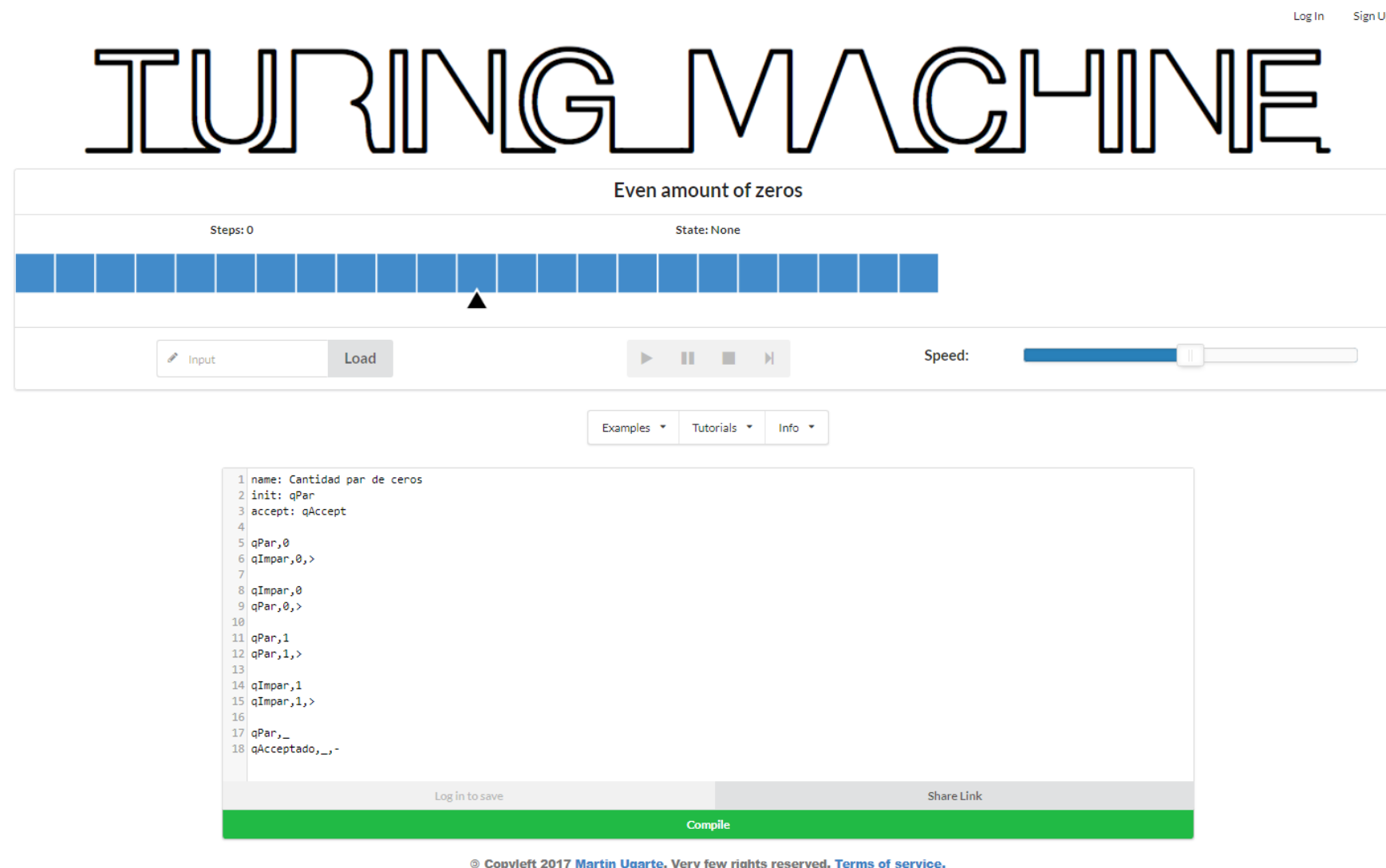


- Al ser sencillas, el análisis de sus propiedades abstractas ha contribuido fuertemente en las ciencias de la computación.
- El concepto de Máquinas de Turing fue fundamental en la categorización de los problemas computacionales en P y NP.
- Las Máquinas de Turing Universales definen los sistemas Turing completos y son la base de la Teoría de la computabilidad.
- Tesis de Church Turing:
“Todo algoritmo es una Máquina de Turing”



[HTTPS://TURINGMACHINESIMULATOR.COM](https://turingmachinesimulator.com)

TURING MACHINE SIMULATOR



- Permite usar más de una cinta.
- No es necesario definir el alfabeto de la máquina.
- El símbolo blanco es “_”
- No permite usar wildcards, se debe definir una función de transición para cada símbolo.
- Permite varios estados finales.


TURING MACHINE



Cinta

Even amount of zeros





Steps: 0


State: None



 Input

Load

Speed: 

Examples ▾

Tutorials ▾

Info ▾

1 name: Cantidad par de ceros

2 init: qPar

3 accept: qAccept

4

5 qPar,0

6 qImpar,0,>

7

8 qImpar,0

9 qPar,0,>

10

11 qPar,1

12 qPar,1,>

13

14 qImpar,1

15 qImpar,1,>

16

17 qPar,_

18 qAcceptado,_,-

Log in to save

Share Link

Compile

TURING MACHINE

Even amount of zeros

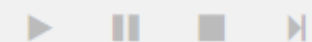
Steps: 0

State: None



 Input

Load



Speed:



Examples ▾

Tutorials ▾

Info ▾

Input

```
1 name: Cantidad par de ceros
2 init: qPar
3 accept: qAccept
4
5 qPar,0
6 qImpar,0,>
7
8 qImpar,0
9 qPar,0,>
10
11 qPar,1
12 qPar,1,>
13
14 qImpar,1
15 qImpar,1,>
16
17 qPar,_
18 qAcceptado,_,-
```

[Log in to save](#)

[Share Link](#)

Compile

TURING MACHINE

Even amount of zeros

Steps: 0 State: None



Input Load ▶ || ◼ ▶▶ Speed:

[Examples](#) [Tutorials](#) [Info](#)

```
1 name: Cantidad par de ceros
2 init: qPar
3 accept: qAccept
4
5 qPar,0
6 qImpar,0,>
7
8 qImpar,0
9 qPar,0,>
10
11 qPar,1
12 qPar,1,>
13
14 qImpar,1
15 qImpar,1,>
16
17 qPar,_
18 qAcceptado,_,-
```

[Log in to save](#)

[Share Link](#)

Compile

© Copyleft 2017 [Martin Ugarte](#). Very few rights reserved. [Terms of service](#).

Cabezal

TURING MACHINE

Even amount of zeros

Steps: 0 State: None

Input Load [Play] [Pause] [Stop] [Fast Forward] Speed: [Slider]

Examples Tutorials Info

Código

```
1 name: Cantidad par de ceros
2 init: qPar
3 accept: qAccept
4
5 qPar,0
6 qImpar,0,>
7
8 qImpar,0
9 qPar,0,>
10
11 qPar,1
12 qPar,1,>
13
14 qImpar,1
15 qImpar,1,>
16
17 qPar,_
18 qAcceptado,_,-
```

Log in to save Share Link

Compile

TURING MACHINE

Estados

Even amount of zeros

Steps: 0

State: None

Input Load

Speed:

Examples Tutorials Info

```
1 name: Cantidad par de ceros
2 init: qPar
3 accept: qAccept
4
5 qPar,0
6 qImpar,0,>
7
8 qImpar,0
9 qPar,0,>
10
11 qPar,1
12 qPar,1,>
13
14 qImpar,1
15 qImpar,1,>
16
17 qPar,_
18 qAcceptado,_,-
```

Log in to save Share Link

Compile


```
1 name: Cantidad par de ceros
2 init: qPar
3 accept: qAccept
4
5 qPar,0
6 qImpar,0,>
7
8 qImpar,0
9 qPar,0,>
10
11 qPar,1
12 qPar,1,>
13
14 qImpar,1
15 qImpar,1,>
16
17 qPar,_
18 qAcceptado,_,-
```

Estado inicial

[Log in to save](#)

[Share Link](#)

Compile

```
1 name: Cantidad par de ceros
2 init: qPar
3 accept: qAccept
4
5 qPar,0
6 qImpar,0,>
7
8 qImpar,0
9 qPar,0,>
10
11 qPar,1
12 qPar,1,>
13
14 qImpar,1
15 qImpar,1,>
16
17 qPar,_
18 qAcceptado,_,-
```



Transiciones

Log in to save

Share Link

Compile

```
1 name: Cantidad par de ceros
2 init: qPar
3 accept: qAccept
4
5 qPar,0
6 qImpar,0,>
7
8 qImpar,0
9 qPar,0,>
10
11 qPar,1
12 qPar,1,>
13
14 qImpar,1
15 qImpar,1,>
16
17 qPar,_
18 qAcceptado,_,-
```



Transiciones:

- Condición**
- Instrucciones**

Log in to save

Share Link

Compile

```
1 name: Cantidad par de ceros
2 init: qPar
3 accept: qAccept
4
5 qPar,0
6 qImpar,0,>
7
8 qImpar,0
9 qPar,0,>
10
11 qPar,1
12 qPar,1,>
13
14 qImpar,1
15 qImpar,1,>
16
17 qPar,_
18 qAcceptado,_,-
```



Estado que se acepta

Log in to save

Share Link

Compile

PROCESAMIENTO DEL INPUT

DEPENDE DE:

- Estado inicial
- Set de transiciones
- Estados aceptados

ENUNCIADO

DE BINARIO A DECIMAL

Binario / base 2

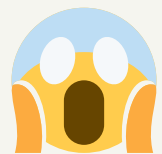
$\{0, 1\}$

Decimal / base 10

$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Hexadecimal / base 16

$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9,$



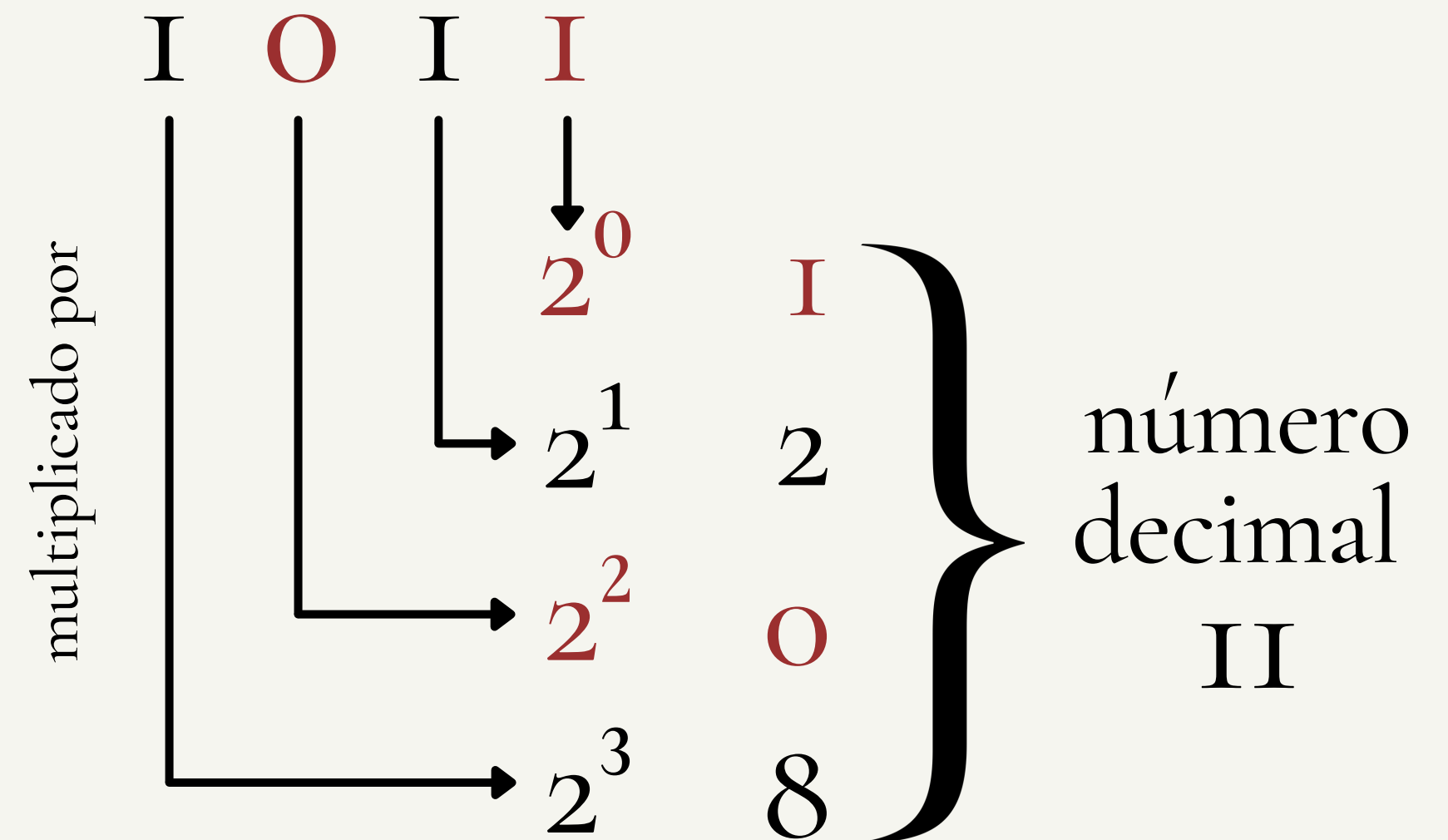
$A, B, C, D, E, F\}$

DE BINARIO A DECIMAL

Binario / base 2
 $\{0, 1\}$

Decimal / base 10
 $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

número binario



ANTECESOR Y SUCESOR



EL ENUNCIADO

- Números binarios ordenados en forma decreciente.
- El primer número es sucesor del segundo, el segundo del tercero, etc.
- Todos los números tendrán el mismo largo.

1101#1100#1011#1010
13#12#11#10

100110#100101#100100#100011
38#37#36#35

EJEMPLO

[HTTPS://TURINGMACHINESIMULATOR.COM](https://turingmachinesimulator.com)

INFORME

[HTTPS://WWW.OVERLEAF.COM](https://www.overleaf.com)