Teoria da Computação

Máquinas de Turing

Thiago Alves

Introdução

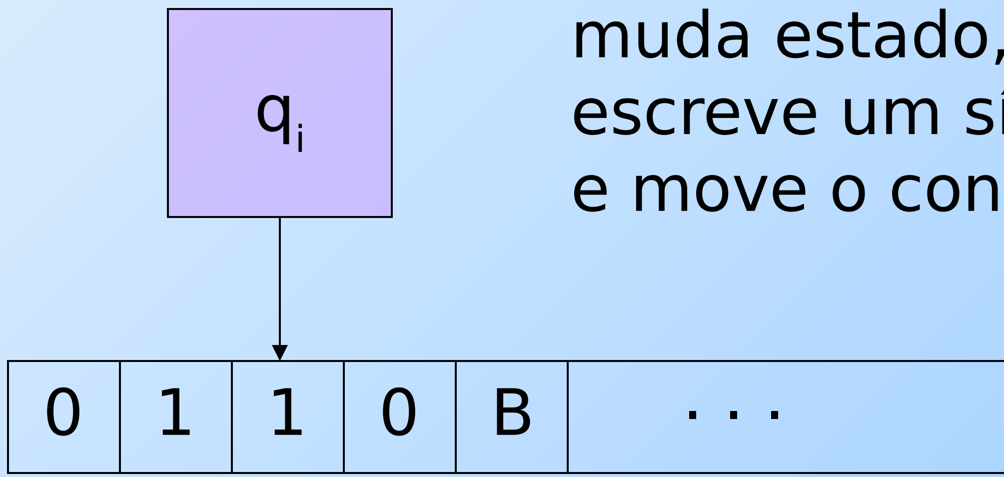
- Queremos um modelo mais poderoso que o autômato de pilha
- Como criar um representação simples para algoritmos?

Introdução

- Algoritmos usam variáveis
- Variáveis podem mudar de valor
- Precisamos de uma forma de armazenar os valores das variáveis

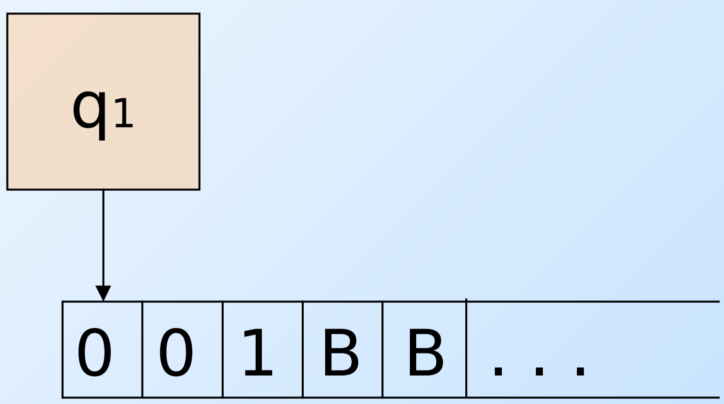
Máquina de Turing

Ação: baseda no estado e no símbolo lido pelo controle: muda estado, escreve um símbolo e move o controle.



Fita infinita com células contendo símbolos de um alfabeto finito

Exemplo



$\delta(q_1, 0) = (q_2, 1, R)$

$\delta(q_1, 1) = (q_1, 0, R)$

$\delta(q_1, B) = (q_a, B, R)$

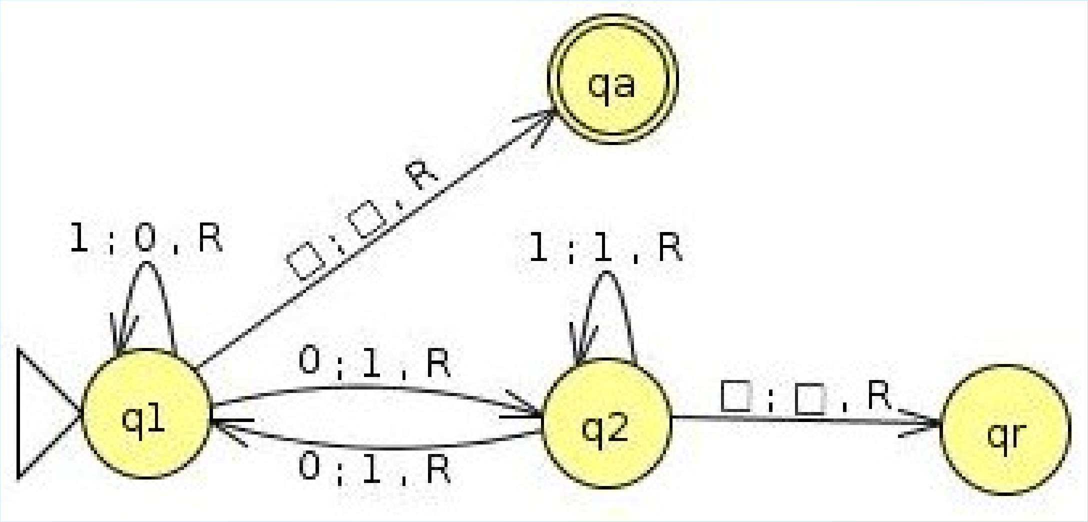
$\delta(q_2, 0) = (q_1, 1, R)$

$\delta(q_2, 1) = (q_2, 1, R)$

$\delta(q_2, B) = (q_r, B, R)$

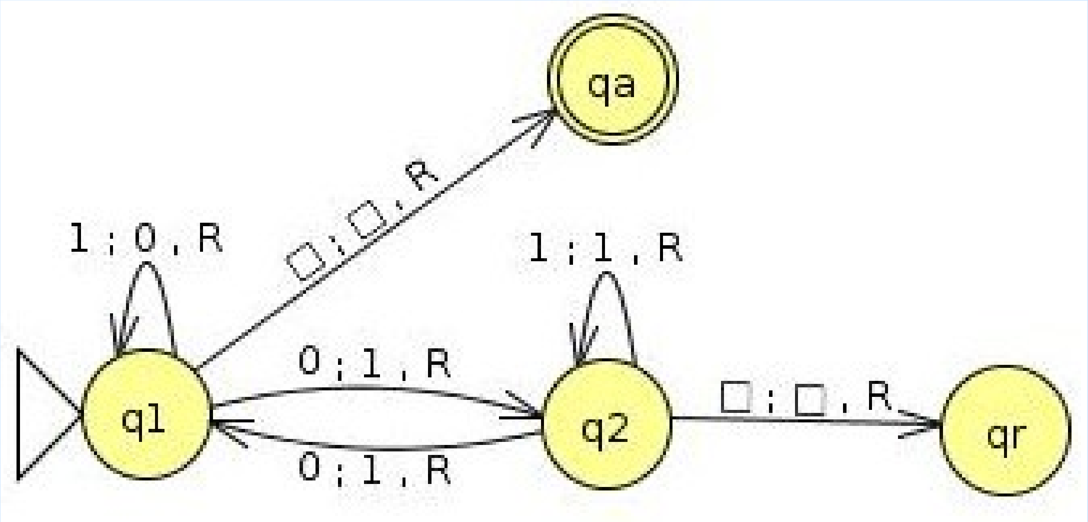
Exemplo

- Que strings aceita?



Exemplo

- Aceita strings com uma quantidade par de 0's



Máquina de Turing - Definição

- Uma Máquina de Turing é descrita por:
  1. Um conjunto finito de *estados* (Q).
  2. Um *alfabeto de entrada* (Σ).
  3. Um *alfabeto da fita* (Γ contendo Σ).
  4. Uma *função de transição* (δ).
  5. Um *estado inicial* (q_0 em Q).
  6. Um *símbolo branco* (B em Γ- Σ).
  7. Um estado de aceitação q_a e um estado de rejeição q_r

  $M=(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

A Função de Transição

- Dois argumentos:
  1. Um estado em Q
  2. Um símbolo da fita em Γ
- $\delta(q_i, a)$ pode ser indefinido ou uma tripla da forma (q_j, b, D)
  - q_j é um estado
  - b é um símbolo de fita
  - D é uma *direção*, L, R ou S.

A Função de Transição

- $\delta(q_i, a)$ pode ser indefinido ou uma tripla da forma (q_j, b, D)
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R,S\}$

A Função de Transição

- Fita tem um início na esquerda
- Se o cabeçote estiver no limite da esquerda e a transição indicar um movimento para a esquerda?
  - A máquina de Turing continua na mesma posição da fita

Exemplo

- Definição formal da máquina de Turing do exemplo anterior
- $Q = \{q_1, q_2, q_a, q_r\}$.
- $\Sigma = \{0, 1\}$.
- $\Gamma = \{0, 1, B\}$.
- $q_0 = q_1$

Exemplo

- Definição formal da máquina de Turing do exemplo anterior
- $\delta(q_1, 0) = (q_2, 0, R)$
- $\delta(q_1, 1) = (q_1, 1, R)$
- $\delta(q_1, B) = (q_a, B, R)$
- $\delta(q_2, 0) = (q_1, 0, R)$
- $\delta(q_2, 1) = (q_2, 1, R)$
- $\delta(q_2, B) = (q_r, B, R)$

Exemplo

- Descreva uma máquina para a linguagem

  $L = \{w \in \{0,1\}^* \mid \text{termina em } 1\}$

Exemplo

$\delta(q_1, 0) = (q_1, 0, R)$ $q_a = q_3$ e $q_r = q_5$

$\delta(q_1, 1) = (q_1, 0, R)$

$\delta(q_1, B) = (q_2, B, L)$

$\delta(q_2, 1) = (q_3, 1, R)$

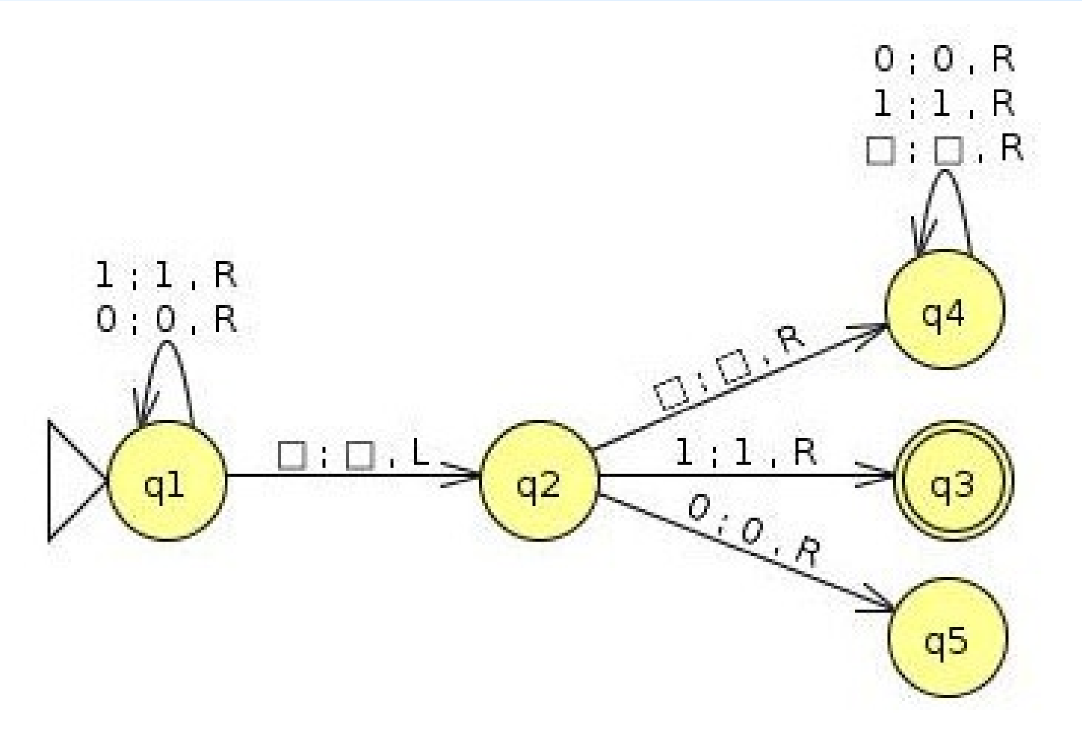
$\delta(q_2, 0) = (q_5, 0, R)$

$\delta(q_2, B) = (q_4, B, R)$

$\delta(q_4, 0) = (q_4, 0, R)$

$\delta(q_4, 1) = (q_4, 1, R)$

$\delta(q_4, B) = (q_4, B, R)$



Máquinas de Turing

- Máquinas de Turing podem
  - Ler e escrever na fita
  - Podem ir para a direita e esquerda da entrada
  - Com uma entrada, pode chegar em um estado de aceitação, um de rejeição ou entrar em loop

Configuração

- Na computação de uma máquina de Turing
  - O estado pode mudar
  - O conteúdo da fita pode mudar
  - A posição do cabeçote pode mudar
- Uma configuração é a situação destes itens

Configuração

- Inicialmente, uma máquina de Turing tem um fita com a string de entrada seguida por brancos infinitos
- A máquina de Turing está no estado inicial
- O cabeçote está no primeiro símbolo da entrada

Configuração

- Configuração inicial
- $q_0 a_1 a_2 \ldots a_n$
- Configuração inicial para a máquina do primeiro exemplo:
- Entrada 001:
  - $q_1 001$
- Entrada 101
  - $q_1 101$

Configuração

- Configuração geral:
- É uma string $\alpha q \beta$, em que $\alpha\beta$ inclui a fita até o não branco mais à direita
- O estado q está imediatamente à esquerda do símbolo da fita lido pelo cabeçote

Configuração

- Configuração geral:
- Exemplos:
  - $1q_2 01$
  - $11q_1 1$
  - $110q_1 B$

Configuração

- Configuração de aceitação
  - O estado da configuração é q_a
  - $10q_a1$
- Configuração de rejeição
  - O estado da configuração é q_r
  - $101q_r$

Movimentos

- Usamos o símbolo $\vdash$ e $\vdash^*$ para representar um movimento e zero ou mais movimentos, respectivamente
- Usamos a função de transição para determinar o movimento
- Exemplo: Os movimentos do primeiro exemplo são
  - $q_1 001 \vdash 1 q_2 01$
  - $1 q_2 01 \vdash 11 q_1 1$
  - $11 q_1 1 \vdash 110 q_1 B$
  - $110 q_1 B \vdash 110B q_a B$

Linguagem de uma MT

- Uma máquina de Turing aceita uma entrada w se $q_0 w \vdash^* C$
  - Em que C é uma configuração com estado de aceitação q_a
- A linguagem de M:
- $L(M) = \{w \mid q_0 w \vdash^* C$, em que C tem $q_a\}$
  - Também dizemos que L(M) é reconhecida por M

Linguagem Recursivamente Enumerável

- Uma linguagem L é Turing-reconhecível se existe uma máquina de Turing M tal que
  - L(M) = L
- Também chamada de Linguagem Recursivamente Enumerável

Linguagem Recursivamente Enumerável

- Uma máquina de Turing M pode ter três comportamentos com uma entrada w
  - Chegar em um estado q_a e parar
    - $w \in L(M)$
  - Chegar em um estado q_r e parar
    - $w \notin L(M)$

Linguagem Recursivamente Enumerável

- Uma máquina de Turing M pode ter três comportamentos com uma entrada w
  - Não chegar em q_a nem em q_r
    - Não vai parar
    - $w \notin L(M)$

Linguagens Recursivas

- Preferimos máquinas de Turing que param para qualquer entrada w
  - Chega em q_a aceitando w
  - Chega em q_r rejeitando w

Linguagens Recursivas

- Uma linguagem L é Turing-decidível se existe uma máquina de Turing M que pára em toda entrada, tal que
  - L = L(M)
- Também chamada de Linguagem Recursiva

Exemplo

- Faça uma máquina de Turing para decidir a linguagem

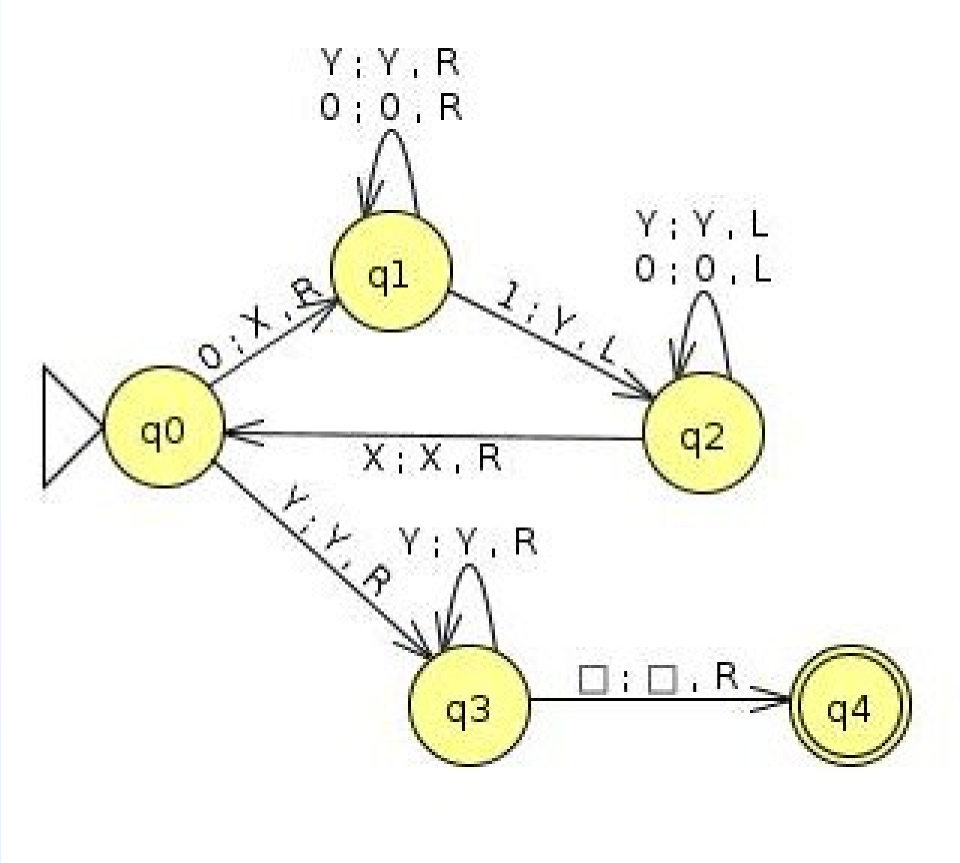
  $\{0^n1^n \mid n>0\}$

Exemplo

- Faça uma máquina de Turing para decidir a linguagem $\{0^n1^n \mid n>0\}$
  - Estado inicial para marcar um zero
  - Um estado para achar o primeiro 1 e marcá-lo
  - Um estado para voltar ao primeiro zero não marcado
  - Um para terminar as marcações
  - Estado de aceitação e um de rejeição

Exemplo

- Faça uma máquina de Turing para decidir $\{0^n 1^n \mid n>0\}$



Transições não descritas vão para q_r

Teorema

- Se A é uma linguagem recursiva, então A é uma linguagem recursivamente enumerável

Teorema

- Se A é uma linguagem recursiva, então A é uma linguagem recursivamente enumerável
- Prova:
  - Suponha A recursiva. Existe máquina de Turing M que decide A.
  - Podemos construir M' a partir de M trocando o estado q_r por um estado q' e forçando esse novo estado entrar em loop.

Exemplo

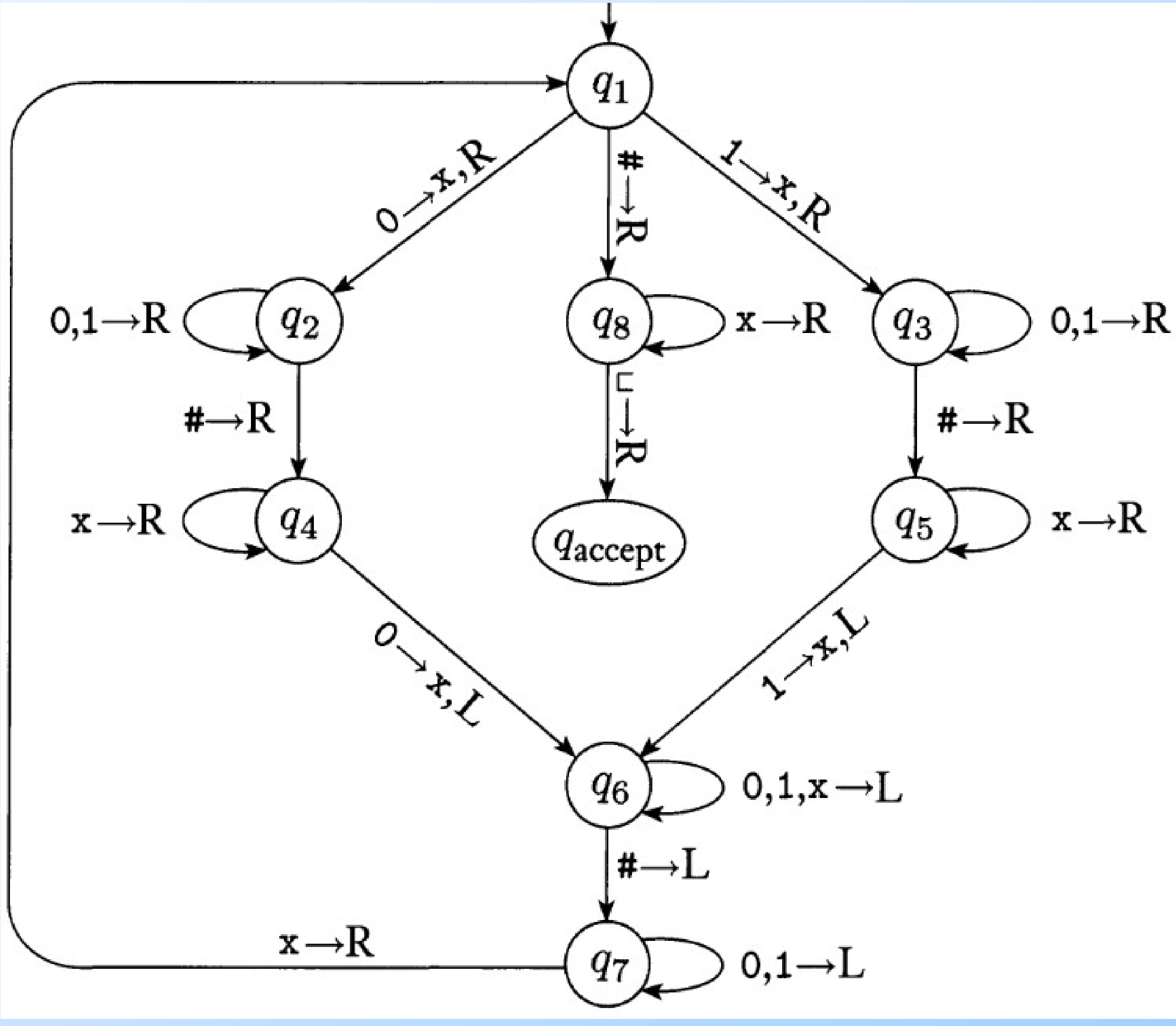
- Descreva uma máquina para decidir a linguagem

  $\{w\#w \mid w \in \{0,1\}^*\}$

Exemplo

- Fazer um zigue-zague nas posições correspondentes em cada lado do # para verificar se são o mesmo símbolo e marcá-los
  - Caso não sejam, rejeitar.
- Quando todos da esquerda tiverem sido marcados verificar se ainda existem símbolos na direita
  - Caso não existam, aceitar.

Exemplo



q_1
$0 \rightarrow x,R$
$\# \rightarrow R$
$1 \rightarrow x,R$
$0,1 \rightarrow R$
q_2
q_8
$x \rightarrow R$
q_3
$0,1 \rightarrow R$
$\# \rightarrow R$
$\sqcup \rightarrow R$
$\# \rightarrow R$
$x \rightarrow R$
q_4
q_{accept}
q_5
$x \rightarrow R$
$0 \rightarrow x,L$
$1 \rightarrow x,L$
q_6
$0,1,x \rightarrow L$
$\# \rightarrow L$
$x \rightarrow R$
q_7
$0,1 \rightarrow L$

Exemplo

- Faça uma máquina de Turing para decidir a linguagem
  $\{ 0^{2^n} \mid n \geq 0\}$

Exemplo

- A ideia é cortar a quantidade de 0's pela metade a cada passo
- Ao percorrer a fita verificar se a quantidade de 0's é ímpar e maior que um
  - Se for, rejeita!
- No final, se a quantidade de 0's for um, aceita!

Exemplo

- Faça uma máquina de Turing para decidir $\{ 0^{2^n} \mid n \geq 0 \}$