

PCS3616

Programação de Sistemas

(Sistemas de Programação)

Máquinas de Turing

Escola Politécnica da Universidade de São Paulo

Objetivos

- Familiarização com o funcionamento global de software de sistema computacional
- Desenvolver experimentos em computador
  - Ambiente (Expressões Regulares, scripts)
  - Máquinas/Algoritmos (Máquina de Turing)
  - Software de Sistema (Montadores, Editores de Ligação. Relocadores)
  - Noções de Compiladores (Front-end)
- Ferramenta para automatização de geração de código, otimização em Compiladores (Back-end).

Organização

- Material no Moodle do Stoa
  - http://disciplinas.stoa.usp.br/course/view.php?id=8061
- Professor
  - Ricardo Luis de Azevedo da Rocha
- Monitora
  - Débora Setton Fernandes

Roteiro desta Aula

1. Introdução às Máquinas de Turing
   - Introdução
   - Ações básicas de uma Máquina de Turing
   - Definição formal de uma Máquina de Turing
   - Linguagens decidíveis
2. Aplicações das Máquinas de Turing
   - Computação de funções
   - Uma Máquina de Turing para Somar em Unário

Alan Mathison Turing

(23/06/1912 – 07/06/1954)

Matemático, lógico, criptógrafo e herói de guerra britânico. Considerado o pai da ciência da computação. Fez previsões acerca da Inteligência Artificial e propôs o Teste de Turing, contribuindo para o debate sobre a consciência das máquinas e suas capacidades de pensar. Formalizou o conceito de algoritmo e computação com a Máquina de Turing, gerando sua versão da Tese de Church-Turing. Responsável pela quebra do código alemão *Enigma* durante a II Guerra Mundial. Depois da guerra, projetou um pioneiro computador digital programável eletronicamente. Foi processado e condenado por ser homossexual (em 1952). Morreu envenenado (provável suicídio). O Turing Award foi criado em sua homenagem.



Máquina de Turing

- **Máquina de Turing**: modelo mais poderoso de computador, proposto pelo inglês Alan M. Turing em 1936.
  - Similar a um autômato finito, porém com uma memória ilimitada e irrestrita, constituindo um modelo mais exato de um computador de propósito geral.
- Uma Máquina de Turing **computa** uma entrada.

Máquina de Turing (1)

- **Máquina de Turing**: modelo de computação proposto pelo inglês Alan M. Turing em 1936.
  - Uma Máquina de Turing pode ser vista como um computador com um único programa fixo: para alterar o programa, é preciso construir outra máquina.
  - O software é representado pelo conjunto de transições, e o hardware, pela fita, pelo cabeçote e pelo mecanismo que controla as atividades de movimento e de leitura/escrita da cabeçote.

Máquina de Turing (2)

- É possível construir uma **Máquina de Turing Universal**, a qual simula a computação de Máquinas de Turing arbitrárias sobre entradas arbitrárias.
- Eliminadas suas limitações de recursos, um **computador moderno** pode ser visto como um dispositivo similar à Máquina de Turing Universal.

Máquina de Turing (3)

- Uma Máquina de Turing compõe-se de:
  - Uma fita infinita, composta de células, cada qual contendo um símbolo de um alfabeto finito disponível (a fita também implementa a memória externa da máquina);
  - Um cursor, que pode efetuar leitura ou escrita em uma célula, ou mover-se para a direita ou para a esquerda;
  - Uma máquina de estados finitos, que controla o cursor;



Computação em uma MT (4)

- Inicialmente a fita contém somente a cadeia de entrada, com o cursor posicionado (por convenção) no início da cadeia (o restante da fita está em branco **b**);
- Para armazenar algo, a máquina o grava na fita;
- Se a máquina tentar mover o cursor para a esquerda, estando o cursor posicionado na primeira célula da fita, este não se moverá;
- As saídas **aceita** e **rejeita** são obtidas ao entrar a máquina nos estados de aceitação e rejeição, respectivamente;
- Se a máquina não entrar em um estado de aceitação ou de rejeição, continuará sua computação para sempre (*loop* infinito).

MT como um Conjunto de Ações

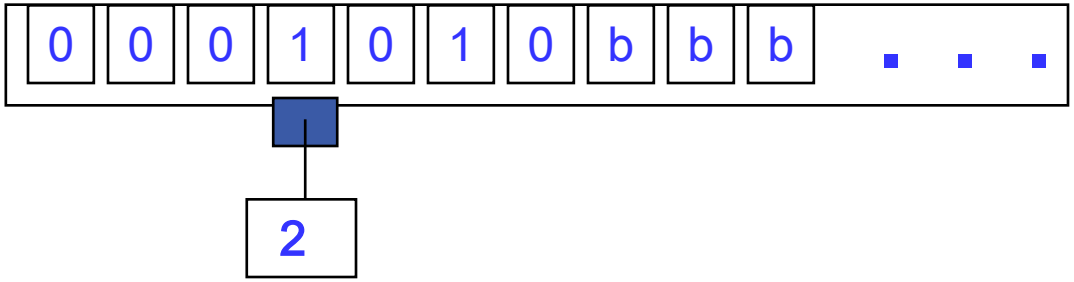
- Uma MT pode ser descrita por um conjunto de ações.
- **Ações**: **(s, i, i', s', d)** sendo:
  - s: estado corrente da MEF
  - i: símbolo que está sendo lido na fita
  - i': símbolo que é gravado na fita, no lugar de i
  - s': próximo estado da MEF
  - d$\in${D,E}, indicando que o cursor pode se mover para a Direita ou para a Esquerda.

Ações da Máquina de Turing (1)

- Exemplo:

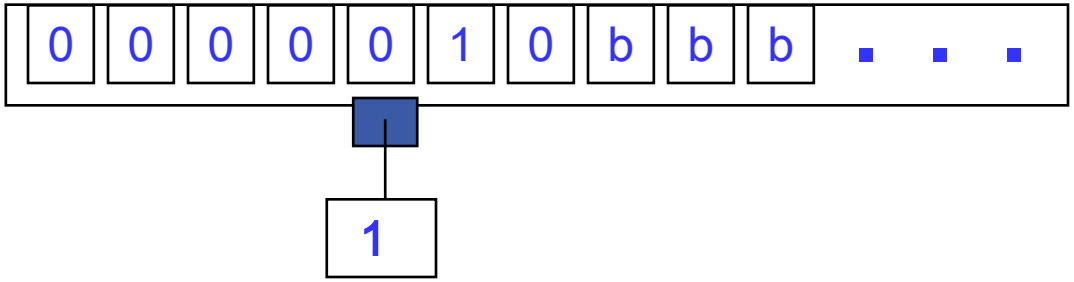
Estando a máquina na situação (a):

(a)



executando a ação (2,1,0,1,D), a nova situação será (b):

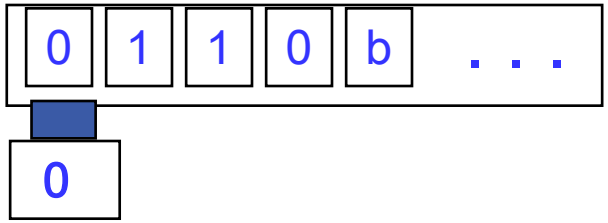
(b)



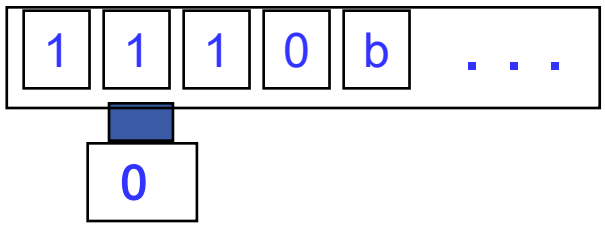
Ações da Máquina de Turing (2)

- Ex: uma MT é definida pelo conjunto de quíntuplas: (0,0,1,0,D), (0,1,0,0,D), (0,b,1,1,E), (1,0,0,1,D), (1,1,0,1,D), (1,b,b,2,D). O estado de aceitação é 2. Verificar sua computação:

Início:

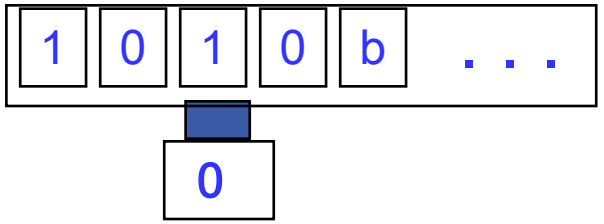


(0,0,1,0,D)

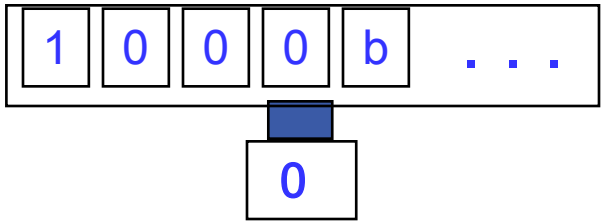


(0,1,0,0,D)

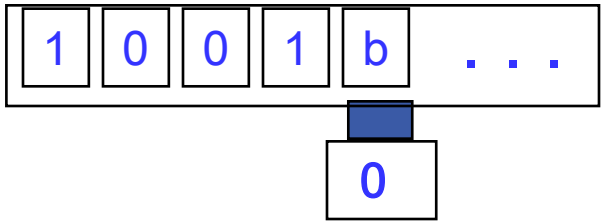
Ações da Máquina de Turing (3)



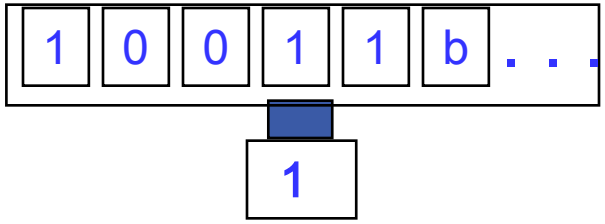
(0,1,0,0,D)



(0,0,1,0,D)

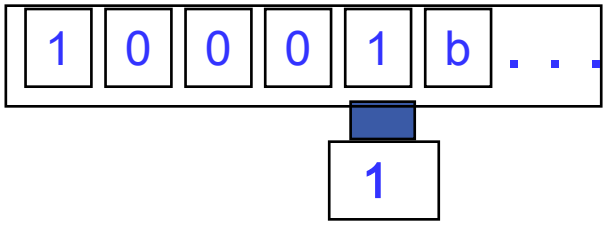


(0,b,1,1,E)

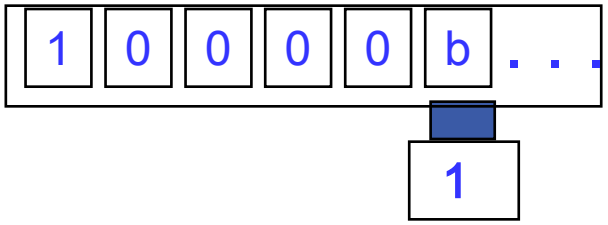


(1,1,0,1,D)

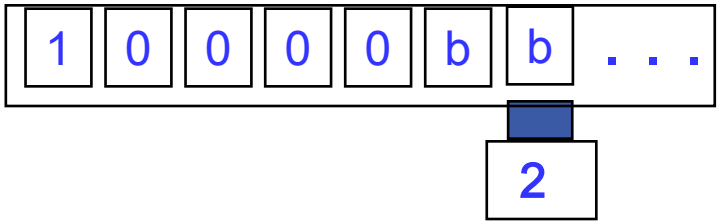
Ações da Máquina de Turing (4)



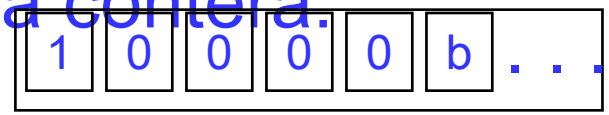
(1,1,0,1,D)



(1,b,b,2,D)



Quando a máquina parar no estado 2 (estado de aceitação), a fita conterá:



Definição Formal da Máquina de Turing

- Uma MT $(S, I, \Gamma, f, \sigma_0, \sigma_A, \sigma_R)$ compõe-se de:
  - Um conjunto finito S de estados;
  - Um conjunto finito I de símbolos de entrada, $b \notin I$;
  - Um conjunto finito Γ de símbolos da fita, com $b \in \Gamma$ e $I \subseteq \Gamma$;
  - Uma função $f: S \times \Gamma \rightarrow S \times \Gamma \times \{E, D\}$, sendo D: direita, E: esquerda;
  - $\sigma_0 \in S$ é o estado inicial;
  - $\sigma_A \in S$ é o estado de aceitação;
  - $\sigma_R \in S$ é o estado de rejeição, com $\sigma_A \neq \sigma_R$.
    - OBS: b: célula em branco da fita

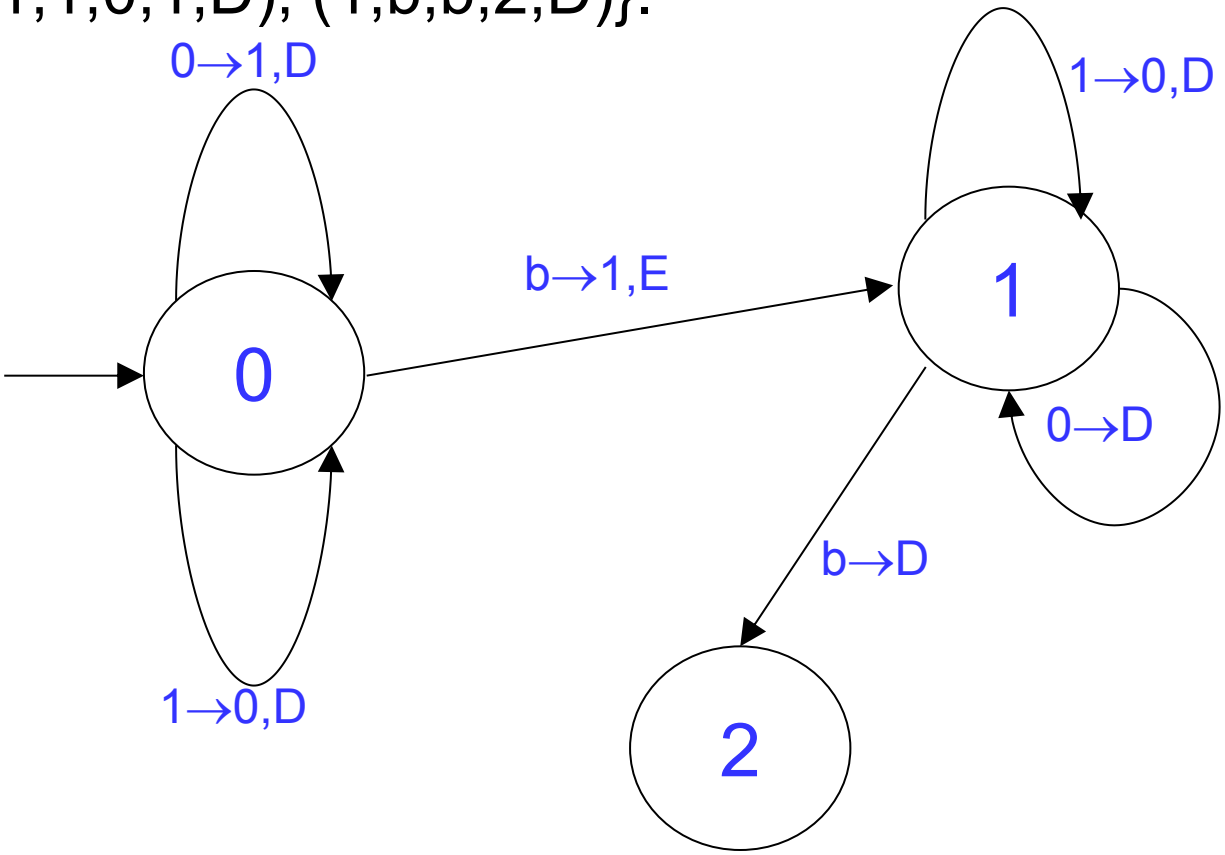
Diagrama de Transições de uma MT (1)

- Seja M = (S, I, Γ, f, σ_0, σ_A, σ_R) uma máquina de Turing. O **diagrama de transições** de M é um grafo orientado G com:
  - vértices membros de S.
  - uma seta indicando o **estado inicial** σ_0.
  - uma aresta orientada (σ_1,σ_2) existe em G se existir uma entrada de fita i$\in\Gamma$ com f(σ_1,i)= (σ_2, j, m), sendo σ_1,$\sigma_2\in$S, j$\in\Gamma$ e m indica o movimento do cursor, m$\in${E, D}.

    Neste caso, a aresta (σ_1,σ_2) é rotulada com **i $\rightarrow$ j, m** para i $\neq$ j, e **i $\rightarrow$ m** para i $=$ j.

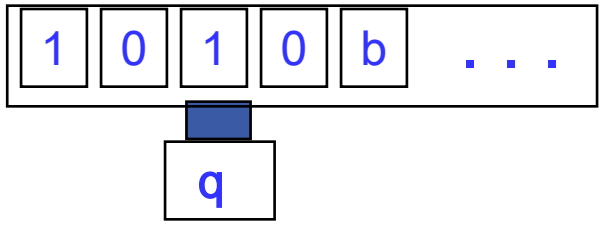
Diagrama de Transições de uma MT (2)

Desenhar o diagrama de transições da MT definida pelo conjunto de ações seguinte: {(0,0,1,0,D), (0,1,0,0,D), (0,b,1,1,E), (1,0,0,1,D), (1,1,0,1,D), (1,b,b,2,D)}.



Configuração de uma MT (1)

- Conforme a MT efetua sua computação, mudanças ocorrem no estado atual, no conteúdo da fita e na posição do cursor.
- Uma situação da computação pode ser representada por uma **configuração**, que descreve de forma abreviada a situação da máquina.
- Por exemplo: a configuração **10q10** representa a seguinte situação:



Configuração de uma MT (2)

- A configuração inicial de uma MT é dada por $\sigma_0 w$, para a cadeia de entrada w.
- O estado, numa configuração de aceitação, deve ser σ_A .
- O estado, numa configuração de rejeição, deve ser σ_R .
- Configurações de aceitação e de rejeição constituem sempre **configurações de parada** da MT.

Aceitação de Cadeias na MT

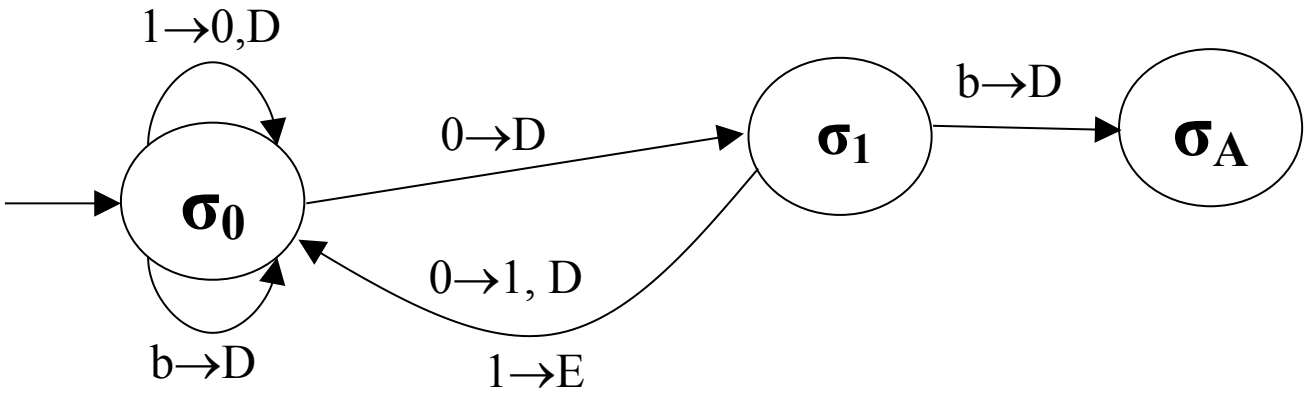
- Uma MT **aceita** a entrada w se existir alguma seqüência de configurações C_1, C_2,... C_k tal que:
  - C_1 é a configuração inicial da MT, $\sigma_0 w$;
  - Aplicando f a C_i , obtém-se C_{i+1};
  - C_k é uma configuração de aceitação.
- A coleção de cadeias que a Máquina de Turing M aceita é a **linguagem** definida pela máquina M, e é denotada por L(M).

Aceitação de Cadeias na MT

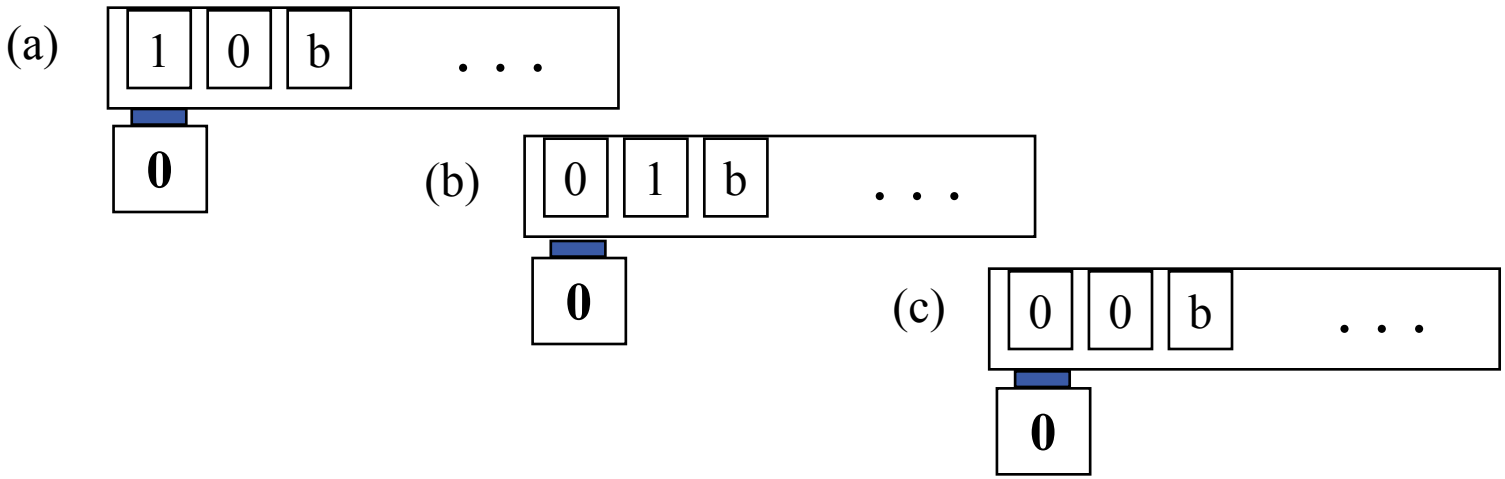
- Ao iniciar uma MT com uma entrada, pode-se: aceitar a entrada ou não aceitar a entrada.
- Uma MT pode não aceitar uma entrada:
  1. ao entrar em σ_R e rejeitar a cadeia ou
  2. ao entrar em *loop* infinito e não parar.
     - Pode ser muito difícil distinguir uma MT que entrou em *loop* infinito de outra que está demorando para computar.

Exemplo de computação

Considere novamente a MT:



Verifique seu funcionamento para:



Linguagem Decidível

Uma MT **decide uma linguagem** quando **pára** para <u>todas</u> as entradas:

- se $w \notin L(M)$, MT pára em σ_R ;
- se $w \in L(M)$, MT pára em σ_A .

Neste caso, a MT é chamada de **decisor**.

Linguagem Decidível

- Ao iniciar uma MT com uma entrada, pode-se aceitar a entrada ou não aceitar a entrada.
- Uma MT pode **não aceitar** uma entrada:
  1. ao entrar em σ_R e **rejeitar** a cadeia ou
  2. ao entrar em *loop* infinito e **não parar nunca**.

     Pode ser muito difícil distinguir uma MT que entrou em *loop* infinito de outra que está demorando para completar a computação.
- Uma MT **decide uma linguagem** quando pára para todas as suas possíveis entradas:
  - se $w \notin L(M)$, MT pára em σ_R ;
  - se $w \in L(M)$, MT pára em σ_A .

Exemplo de MT (1)

Projeto de uma Máquina de Turing M que decide a linguagem $A=\{0^n1^n \mid n>0\}$:

Algoritmo esquematizado de M:

M= "para a cadeia de entrada $w \in \{0,1\}^*$:

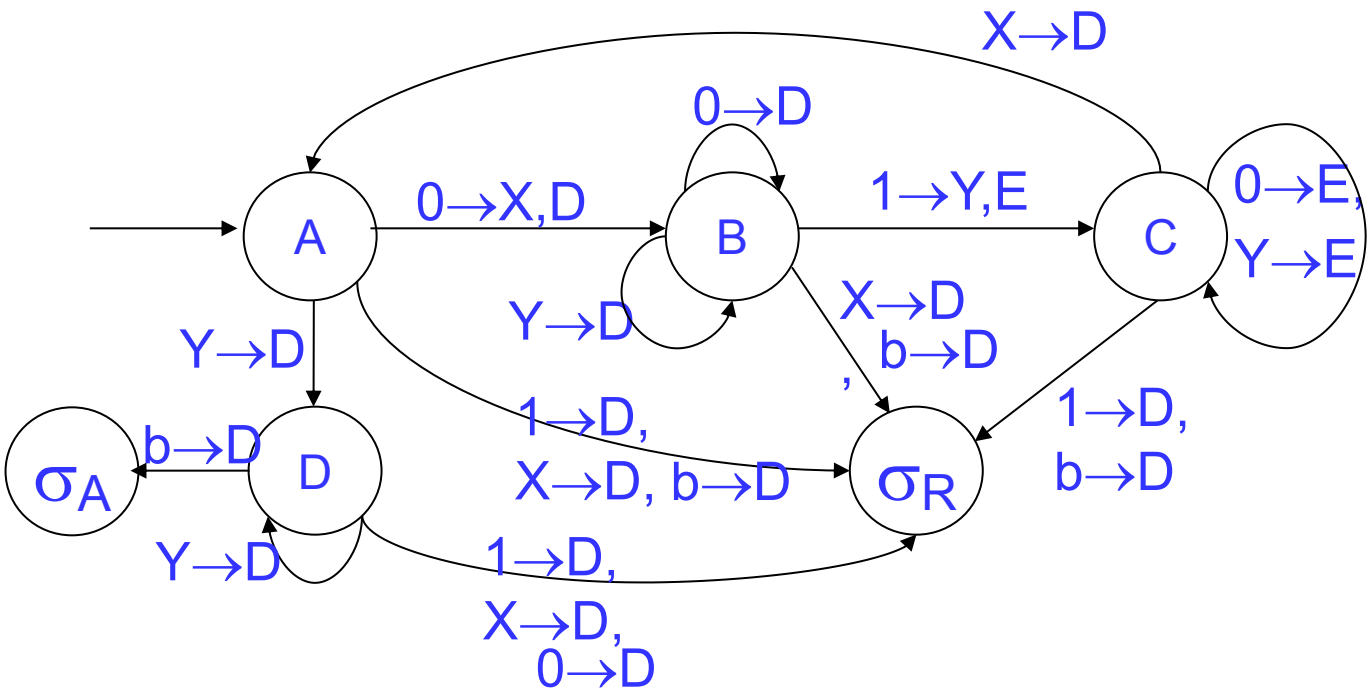
1. Se $w = \lambda$, rejeite.
2. Leia a célula corrente. Se 0, troque por X e vá para o passo 3. Se Y, procure o final da cadeia de Ys e aceite. Se não, rejeite.
3. Vá para a direita da fita, procurando o primeiro 1. Se encontrar, troque-o por Y e vá para o passo 4. Se não, rejeite.
4. Vá para a esquerda na fita, procurando por um X. Se encontrar, volte a avançar na fita e vá para o passo 2. Se não, rejeite."

Exemplo de MT (2)

- Descrição formal de M = (S, I, Γ, f, σ_0, σ_A, σ_R):

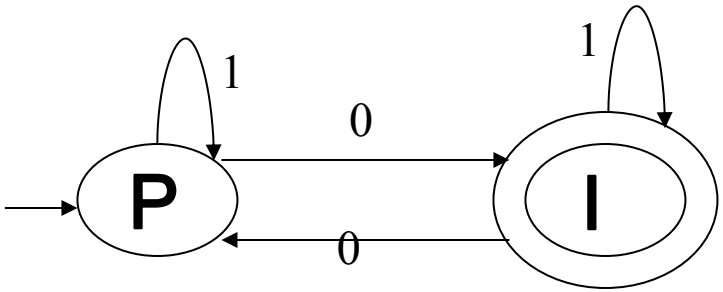
  S = {A,B,C,D, σ_A, σ_R}, σ_0 = A, I={0,1} , Γ= {0,1,X,Y,b},

  f = diagrama de transições abaixo:



Máquina de Turing e Autômato Finito

- Um autômato finito é um caso muito particular de MT, que sempre imprime o símbolo lido na célula, sempre se move para a direita e sempre pára ao ler o símbolo *b* (término da cadeia).
- Exercício: descreva o AF abaixo como uma MT



Variantes da MT

- Existem inúmeros modelos variantes de MT, porém todos se mostram equivalentes em termos de poder de computação (prova-se).
- Muitos outros modelos de computação de propósito geral também foram propostos, alguns bem diferentes de MT. No entanto, é possível provar que todos podem simular uns aos outros e, portanto, também são equivalentes em termos de poder computacional!

➔ Implicação importante: a classe de algoritmos que todos descrevem é única!

Aplicações da Máquina de Turing

- Vimos que uma MT pode decidir uma linguagem. No entanto, uma MT também pode **computar funções**:

  Dada uma MT T e uma cadeia α, começamos com T na configuração inicial padrão em uma fita contendo α. Se T em algum momento pára deixando uma cadeia β na fita, podemos definir β como sendo o valor de uma função avaliada em α. Assim, **$T(\alpha) = \beta$**. Nesse caso, o **domínio da função** T compreende todas as cadeias α para as quais T, em algum momento, pára.

Algoritmos e MTs

- **Noção intuitiva de algoritmo**: é um conjunto finito de instruções que podem ser executadas mecanicamente em tempo finito para resolver algum problema. Com dados de entrada apropriados ao problema, o algoritmo deve obrigatoriamente parar e produzir como saída a resposta correta.
- Assim, qualquer função f, computável por uma MT, será uma função cujos valores podem ser determinados pela execução de um algoritmo ou procedimento computacional.

Alonzo Church (14/06/1903 – 11/08/1995)

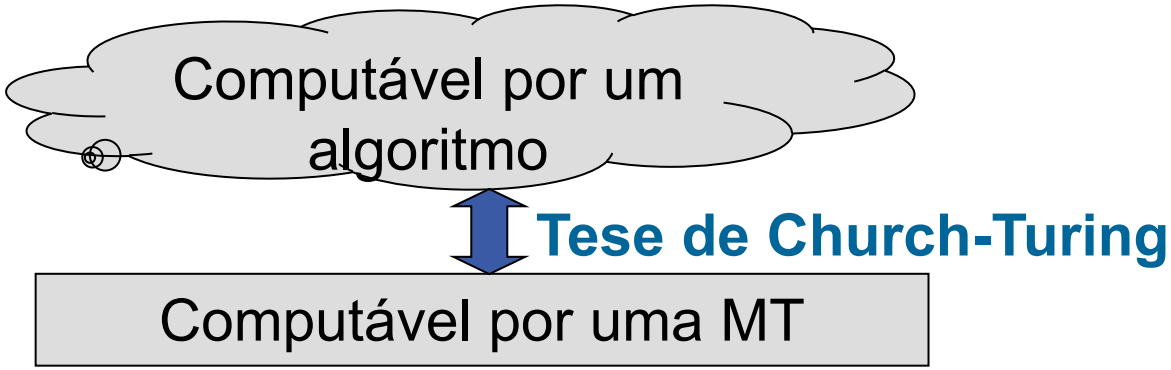
Matemático americano, responsável por alguns dos fundamentos da CC. Nasceu em Washington, DC, recebeu BA e PhD na Universidade de Princeton, USA, em 1924 e 1927, respectivamente. Tornou-se professor em Princeton em 1929. Foi orientador de Stephen C. Kleene (1909-1994), entre outros. Também orientou Alan Turing de 1936 a1938. Seu trabalho mais conhecido é o desenvolvimento do cálculo-λ no seu famoso artigo de 1936, mostrando a existência de problemas indecidíveis. Ele e Turing mostraram que o cálculo-λ e a MT são equivalentes em capacidades, resultando na tese de Church-Turing. Como há disputa sobre "quem foi o primeiro", esta tese também é conhecida como tese de Church e tese de Turing.



Tese de Church-Turing

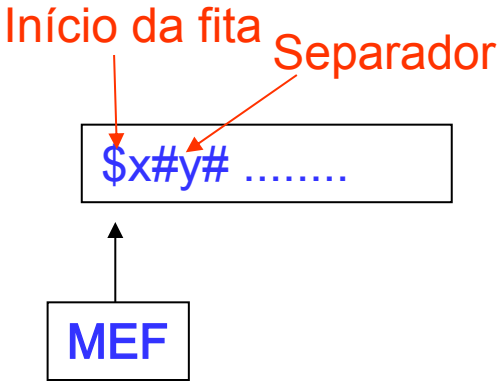
- Em 1936, com os artigos publicados por Turing e Alonzo Church, é que se definiu claramente essa noção de **algoritmo**: É o procedimento representado por alguma **MT que pára em respostas a todas as suas possíveis entradas.**
- Tese de Church-Turing:

"Qualquer função de teoria dos números é computável por um algoritmo se, e somente se, for computável por uma Máquina de Turing."



MT para Computar Funções (1)

- Exemplo 1: Projeto de uma MT que calcula a soma x + y. Considere x e y em representação unária, fornecidos na fita da seguinte forma:



unária	decimal
1	0
11	1
111	2
1111	3
.....	.....

Exemplo: x=3, y=2

fita: $1111#111# ...

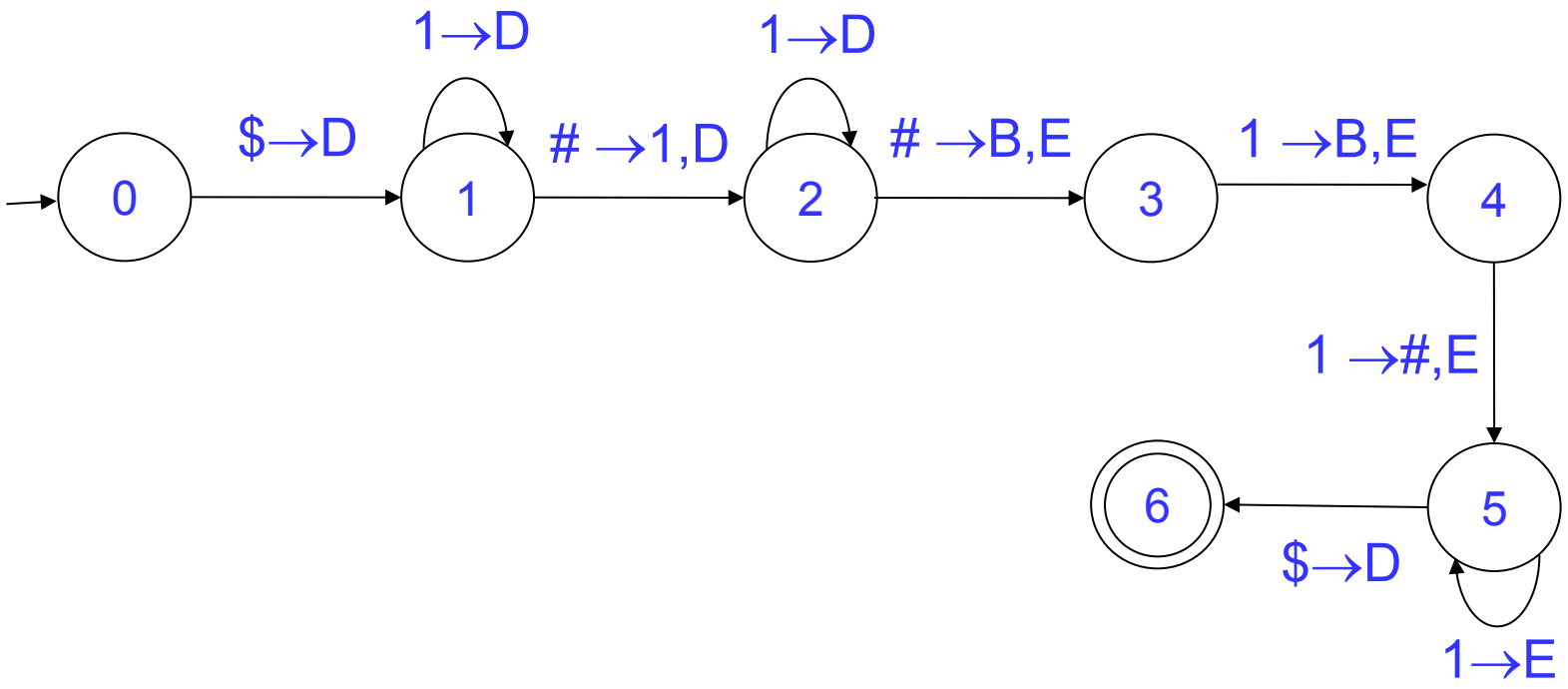
(Formato 1.1) Resposta na fita: $111111#...

(Formato 1.2) Resposta na fita: $1111#111#111111# ...

MT para Computar Funções (2)

- Uma solução para o formato (1.1)

Diagrama de transições:



Linguagens Formais e Modelos Computacionais

- Existem dispositivos computacionais com capacidades intermediárias entre Autômatos Finitos e Máquinas de Turing, que reconhecem linguagens do tipo 2 (livres de contexto) e tipo 1(sensíveis ao contexto).
- Os **Autômatos de Pilha** reconhecem linguagens do tipo 2 (livres de contexto).
- Os **Autômatos Limitados Lineares** reconhecem linguagens do tipo 1(sensíveis ao contexto).

Linguagens Formais e Modelos Computacionais

Linguagem	Gramática	Disp. Computacional
irrestrita	irrestrita	Máquina de Turing
sensível ao contexto	sensível ao contexto	Autômato Limitado Linear
livre de contexto	livre de contexto	Autômato de Pilha
regular	regular	Autômato Finito

Referências Bibliográficas

SAVAGE, J. ***Models of Computation: Exploring the Power of Computing***, 2008.

SIPSER, M. ***Introduction to the Theory of Computation***. PWS Publishing Company, Boston, MA. 1997.

LUCCHESI, C. L. ***Aspectos Teóricos da Computação***, 1979.

Transparências (baseadas em SIPSER) extraídas de:

José Neto, J., Sichman, J. S., Silva, P.S.M., Rocha, R.L.A. ***Material didático da disciplina PCS 2024 – Laboratório de Fundamentos da Engenharia de Computação***, PCS/EPUSP, São Paulo, SP. 2005-2015.

Costa, A.H.R., Sato, L.M., Sichman, J.S., Tori, R. ***Material didático da disciplina PCS 2214 – Fundamentos da Engenharia de Computação I***, PCS/EPUSP, São Paulo, SP. 2004-2005.