Máquinas de Turing

EDUARDO FREIRE NAKAMURA

Instituto de Computação
Universidade Federal do Amazonas
nakamura@icomp.ufam.edu.br

[1]Este material utiliza conteúdo das aulas fornecidas pelo Prof. Vilar da Câmara Neto (disponível em http://http://prof.vilarneto.com).
[2]Permissão de uso fornecida pelos autores.
[3]As figuras utilizadas neste material são de domínio público, disponíveis na Internet sem informações de direitos autorais.

Máquinas de Turing

OBJETIVO

COMPREENDER O FUNCIONAMENTO DE UMA MT

PROJETAR UMA MT

Introdução

- Os dois formalismos vistos até agora (AFs e APs) possuem poder limitado para a representação de linguagens formais

- AFs
  - $\{a^*\}$, $\{(aa)^*\}$, $(a+b)^*$
  - $\{w = (a+b)^* \mid w$ não contém $ab\}$

- APs
  - $\{a^n b^n\}$, $\{a^n b^{3n}\}$, $\{a^m b^n a^n b^m\}$,
  - $\{a^n b^m c^p \mid m + p = n\}$

AFs permitem a criação de repetições que não precisam de contadores de símbolos

APs permitem a criação de contadores (possivelmente aninhados) que podem ser usados uma única vez.

Eduardo Freire Nakamura (nakamura@icomp.ufam.edu.br)

Introdução

- Exemplo 1: A linguagem $L_1=\{a^nb^kc^nd^k \mid n, k>=0\}$ é livre de contexto?

- Exemplo 2: A linguagem $L_2=\{a^nb^nc^n \mid n>=0\}$ é livre de contexto?

- Modelo com poder computacional superior aos AFs e APs?

Não : Apesar de serem linguagens relativamente simples, L_1 e L_2 não podem ser reconhecidas por APs.

Máquinas de Turing

Máquina de Turing

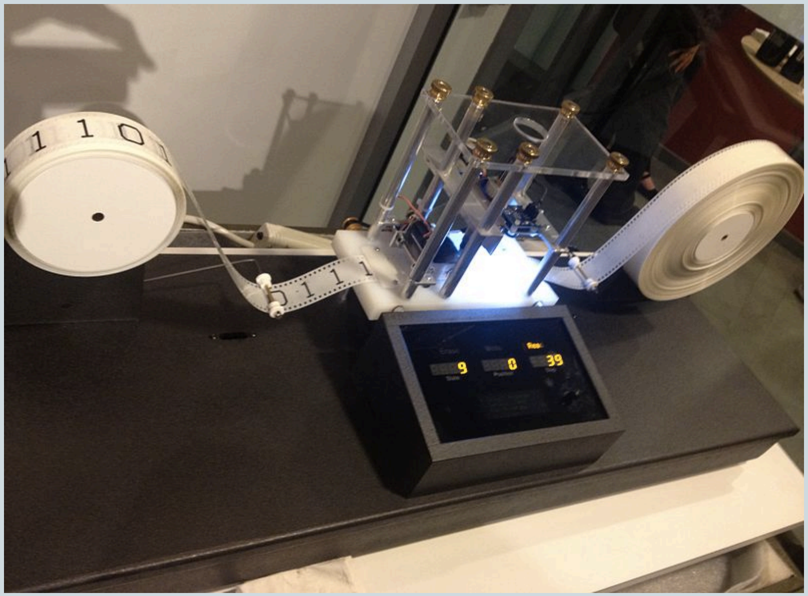
- Uma Máquina de Turing (MT) é uma máquina teórica extremamente simples, composta por:
  - Um conjunto de estados (como AFs e APs)
  - Uma fita de memória com várias células que armazenam um símbolo cada
  - Um cabeçote de leitura e escrita que está sempre posicionado em uma das células
  - Um conjunto de regras de transição
- Proposta em 1936 por Alan Turing[1]

http://en.wikipedia.org/wiki/File:Alan_Turing_photo.jpg

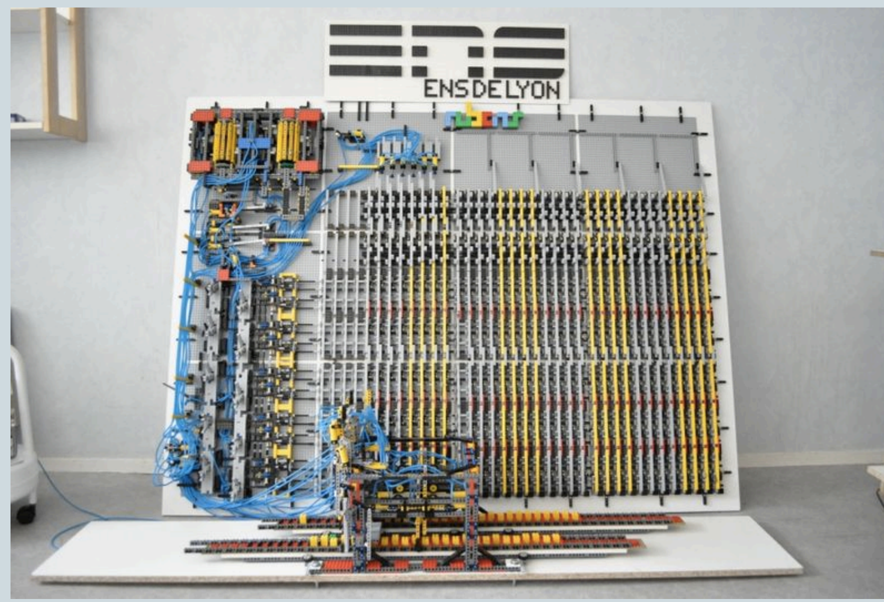
(1912 – 1954)

1 - Turing, A. M. (1936). On computable numbers with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 2, no. 42, pp. 230-265.

Máquina de Turing

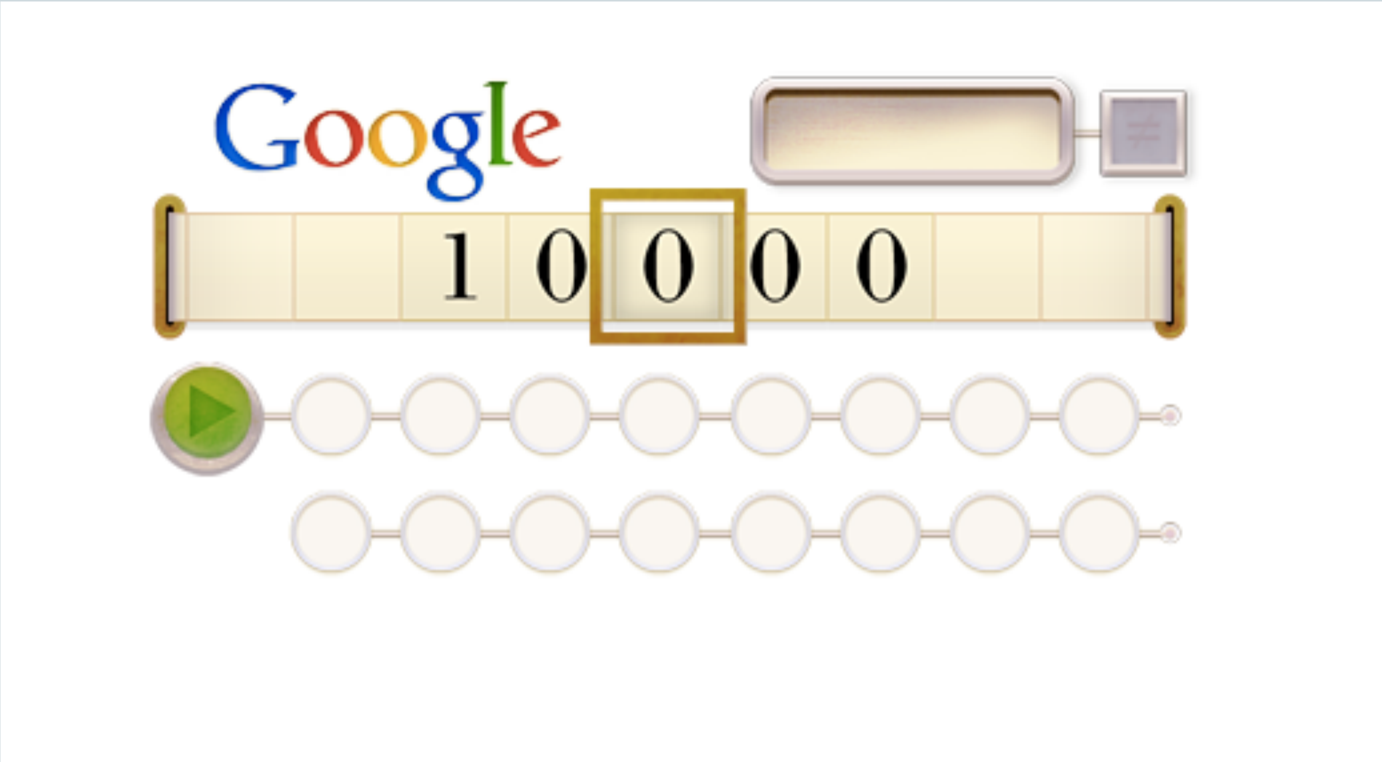


http://en.wikipedia.org/wiki/
File:Model_of_a_Turing_machine.jpg



http://en.wikipedia.org/wiki/File:Lego_Turing_Machine.jpg

Máquina de Turing



https://www.google.com/doodles/alan-turings-100th-birthday

A fita de memória

- A fita de memória de uma MT possui tamanho infinito para a direita.
- A primeira célula contém um símbolo especial para indicar o começo da fita
  - Aqui será adotado o símbolo "〈"
  - Única célula que não pode ser modificada
- A palavra de entrada ocupa as células a partir da segunda
- As células após a palavra de entrada contêm outro símbolo especial para indicar célula vazia
  - Aqui será adotado o símbolo "⊔"

O cabeçote de leitura e escrita

- O cabeçote de leitura sempre está posicionado em uma das células

- Inicialmente, ele está posicionado na segunda célula (no início da palavra quando esta não é nula)

- Exemplo de configuração inicial para a palavra de entrada *abcabc*

〈	$\boxed{a}$	*b*	*c*	*a*	*b*	c	⊔	⊔	⊔	...

- Configuração inicial para entrada vazia (palavra λ)

〈	$\boxed{⊔}$	⊔	⊔	⊔	⊔	...

O cabeçote de leitura e escrita

- O cabeçote pode mover-se para a esquerda ou para a direita, uma posição por vez
- Se o cabeçote estiver na primeira célula, ele não pode mover-se para a esquerda

Configuração inicial	〈	**a**	b	c	a	b	⊔	⊔	⊔	⊔	…
Movimento para esquerda	**〈**	a	b	c	a	b	⊔	⊔	⊔	⊔	…
Movimento para direita	〈	**a**	b	c	a	b	⊔	⊔	⊔	⊔	…
Outro movimento para direita	〈	a	**b**	c	a	b	⊔	⊔	⊔	⊔	…

O cabeçote de leitura e escrita

- Operação básica de uma MT

Escrever um símbolo na célula atual e mover o cabeçote para esquerda ou para direita

Configuração inicial

〈	*a*	*b*	*c*	*a*	*b*	⊔	⊔	⊔	⊔	...
〈	*d*	*b*	*c*	*a*	*b*	⊔	⊔	⊔	⊔	...
〈	*d*	*c*	*c*	*a*	*b*	⊔	⊔	⊔	⊔	...
〈	⊔	*c*	*c*	*a*	*b*	⊔	⊔	⊔	⊔	...
〈	⊔	*c*	*c*	*a*	*b*	⊔	⊔	⊔	⊔	...

A primeira célula que contém "〈" nunca pode ser modificada. Portanto, quando se está sobre ela, só se pode mover após reescrever o mesmo símbolo.

Escrever 〈 e mover para direita

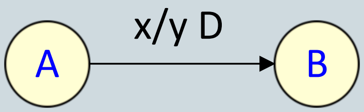
Regras de transição

- Numa MT, uma regra de transição depende de dois pre-requisitos
  1. O estado atual
  2. O símbolo que está na célula atual (a célula que está sob o cabeçote)

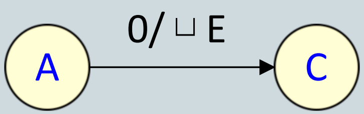
- Uma transição de uma MT possui o seguinte formato
  - $\delta(e, a) = (e', a', E)$ ou $\delta(e, a) = (e', a', D)$
  - Se a MT estiver no estado *e* e o símbolo sob o cabeçote for *a*, então:
    1. Vá para o estado *e'*
    2. Escreva o símbolo *a'* na célula atual
    3. Mova o cabeçote para esquerda (E) ou para a dieita (D)

Regras de transição

$\delta(A, x) = (B, y, D)$

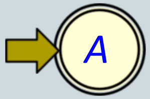


$\delta(A, 0) = (C, \sqcup, E)$



Regras de transição

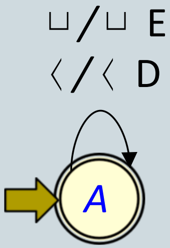
- No modelo de MT padrão, as transições são determinísticas
  - Não há mais de uma transição para a mesma configuração de estado e símbolo lido
- Se não houver transição possível, então a máquina pára no estado atual
- Uma MT aceita uma palavra de entrada se ela pára em um estado final
- Não há o conceito de consumir toda a palavra!
  - A MT sem nenhuma transição aceita qualquer entrada



Regras de transição

- Cuidado com o critério de aceitação
  - Uma MT aceita uma palavra de entrada se ela pára em um estado final
- Isto significa que uma palavra não é aceita se
  - a MT parar em um estado não final; ou
  - a MT não parar

- Qual a linguagem reconhecida pela MT abaixo?



Eduardo Freire Nakamura (nakamura@icomp.ufam.edu.br)

Formalização

- Uma Máquina de Turing (MT) é uma óctupla $(E, \sum, \Gamma, \langle, \sqcup, \delta, i, F)$, onde:
  - E é um conjunto finito de estados;
  - Σ ($\Sigma \subseteq \Gamma$) é o alfabeto de símbolos que podem ocorrer na palavra de entrada;
  - Γ é o alfabeto dos símbolos que podem ser gravados na fita;
  - $\delta: E \times \Gamma \rightarrow E \times \Gamma \times \{E, D\}$ é a função de transição, uma função parcial;
  - $\langle$, um símbolo de Γ, é o símbolo de início da fita, sendo $\langle \notin \sum$;
  - $\sqcup$, um símbolo de Γ, é o símbolo branco, sendo $\sqcup \notin \sum$ e $\sqcup \neq \langle$;
  - i, um estado de E, é o estado inicial;
  - F, um subconjunto de E, é o conjunto de estados finais.

Para que serve uma MT?

- Reconhecer linguagens

  Dada uma palavra de entrada, ela faz parte da linguagem se e somente se a MT parar em um estado final

- Implementar uma função

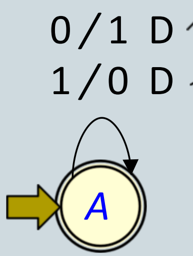
  Dada uma palavra de entrada, transformá-la de acordo com uma regra preestabelecida

  - Neste caso, o conteúdo da fita quando a máquina pára, chama-se saída, e é importante

Eduardo Freire Nakamura (nakamura@icomp.ufam.edu.br)

Exemplo 01: MT como função

- Inverter uma entrada binária
  - Transformar 0s em 1s e vice-versa



Se o valor da célula for 0, a máquina escreve 1 e move para direita

Se o valor da célula for 1, a máquina escreve 0 e move para direita

A MT pára quando encontra o símbolo ⊔ , que indica o fim da palavra de entrada!

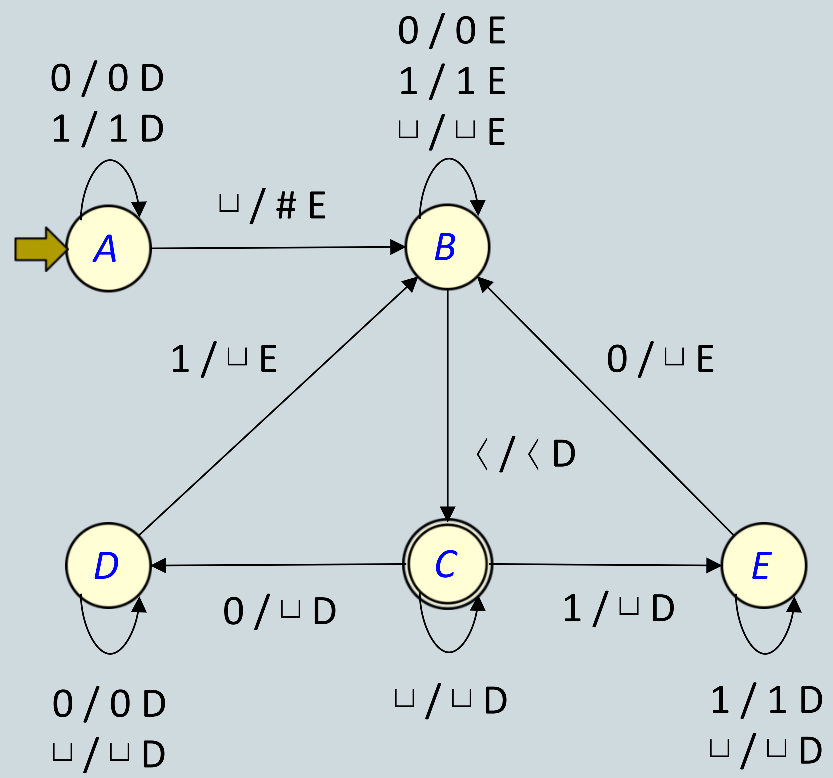
A MT reconhece qualquer palavra em (0+1)*, mas não é esse seu objetivo!

Exemplo 02: MT reconhecedora

{ $w \in \{0,1\}^*$ | o número de 0s é igual ao de 1s }

Exemplo 02: MT reconhecedora

{ $w \in \{0,1\}^*$ | o número de 0s é igual ao de 1s }

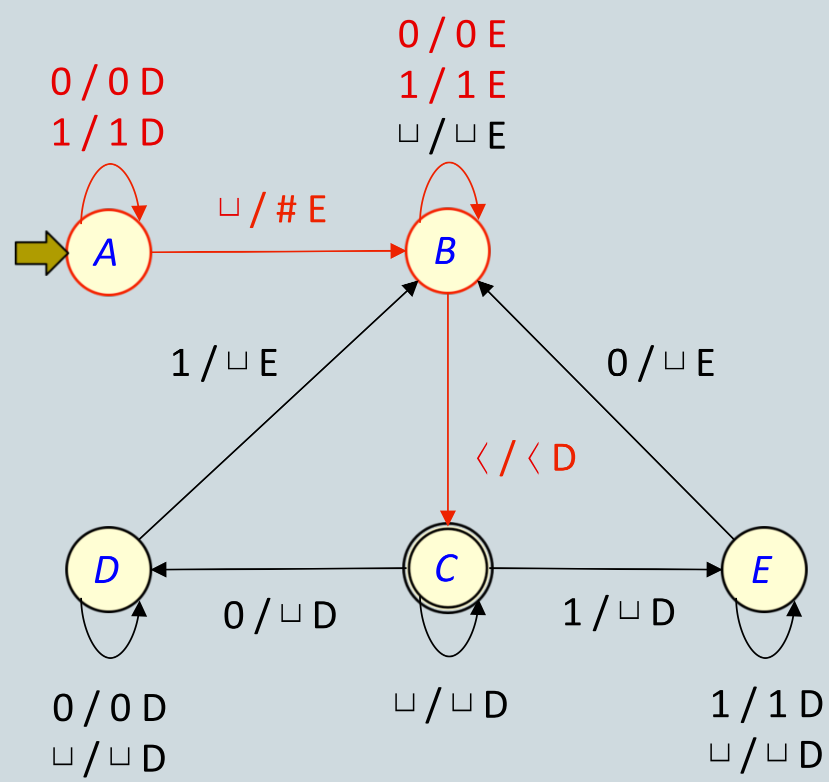


Exemplo 02: MT reconhecedora

{ $w \in \{0,1\}^*$ | o número de 0s é igual ao de 1s }

Parte 1

- Varre a palavra inteira
- Grava o símbolo # na primeira célula com ⊔
- Volta para o início da palavra

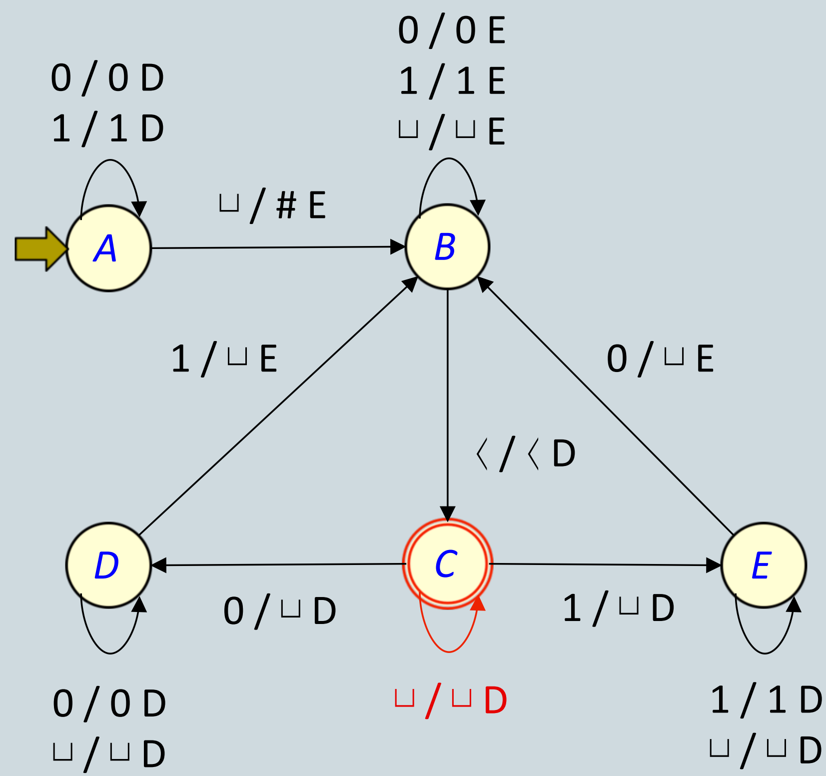


Exemplo 02: MT reconhecedora

{ $w \in \{0,1\}^*$ | o número de 0s é igual ao de 1s }

Parte 2

- Avança para o primeiro símbolo ⊔ da palavra de entrada
- Na primeira iteração essa transição não é usada, pois a palavra não foi sobre escrita com símbolos ⊔
- Se não houver símbolos restantes, a máquina pára no estado C, final

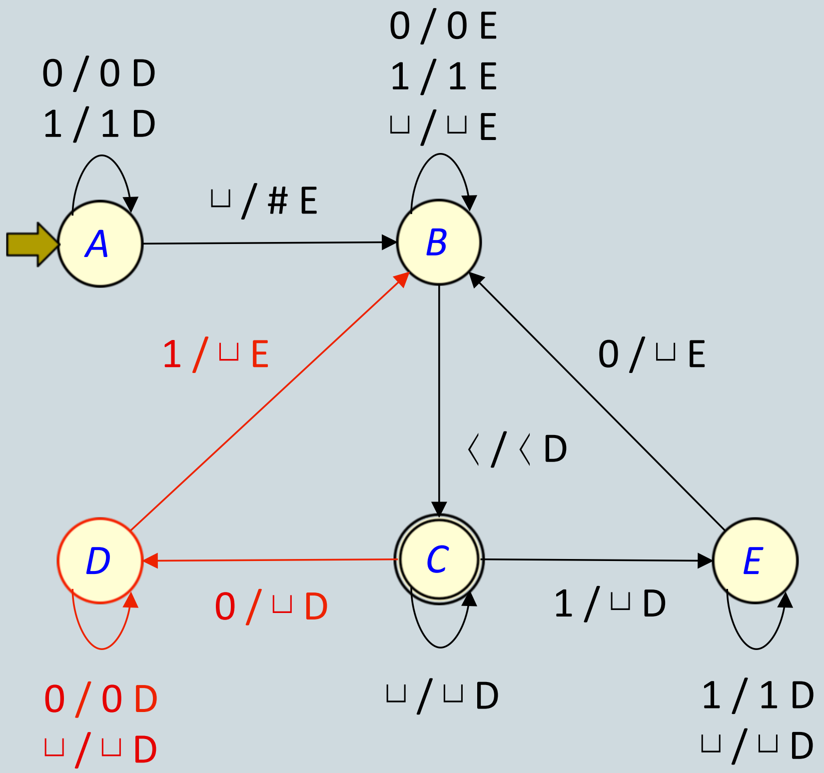


Exemplo 02: MT reconhecedora

{ $w \in \{0,1\}^*$ | o número de 0s é igual ao de 1s }

Parte 3

- Se o primeiro símbolo for 0
  1. Subsitituí-lo por ⊔
  2. Avançar até o primeiro 1
  3. Substituí-lo por ⊔
  4. Se não houver nenhum 1, a MT pára em D, não final, pois chega no símbolo #, sem transição definida



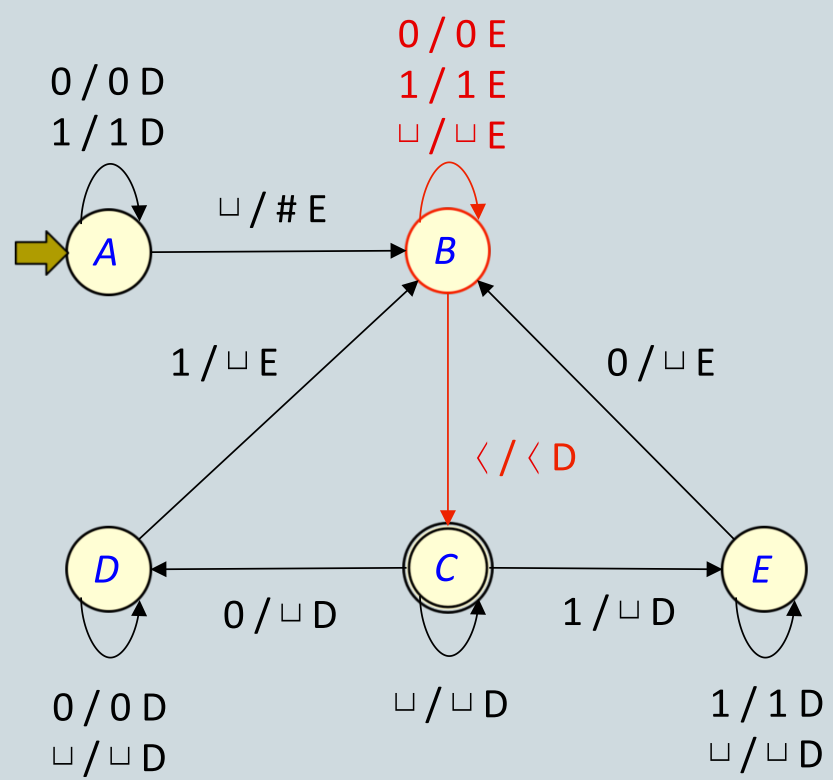
Eduardo Freire Nakamura (nakamura@icomp.ufam.edu.br)

Exemplo 02: MT reconhecedora

{ $w \in \{0,1\}^*$ | o número de 0s é igual ao de 1s }

Parte 3

- Se o primeiro símbolo for 0
  1. Subsitituí-lo por ⊔
  2. Avançar até o primeiro 1
  3. Substituí-lo por ⊔
  4. Se não houver nenhum 1, a MT pára em D, não final, pois chega no símbolo #, sem transição definida
- Retorna para o início da palavra

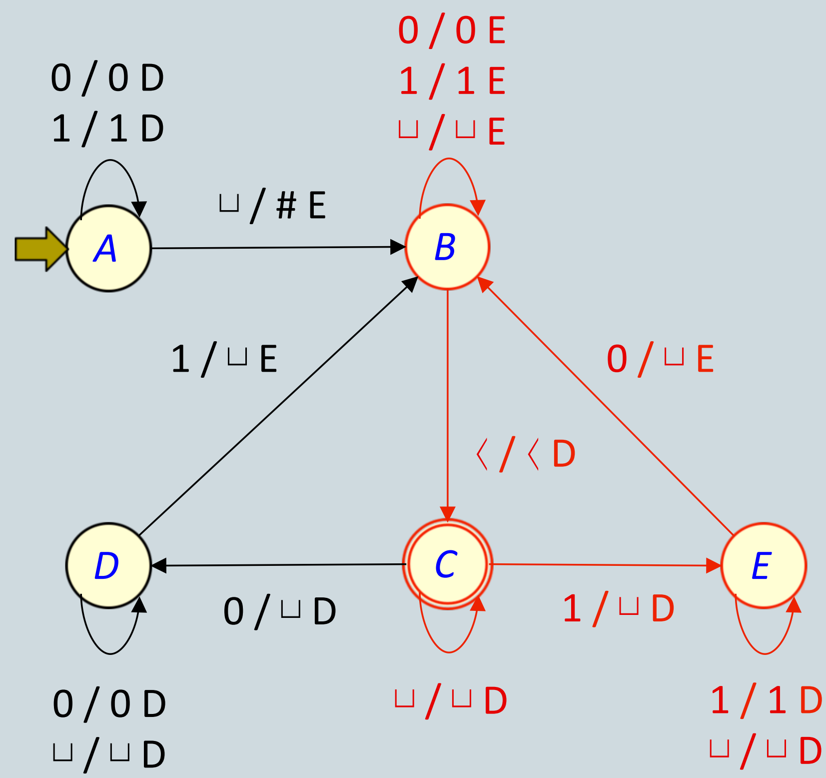


Exemplo 02: MT reconhecedora

{ $w \in \{0,1\}^*$ | o número de 0s é igual ao de 1s }

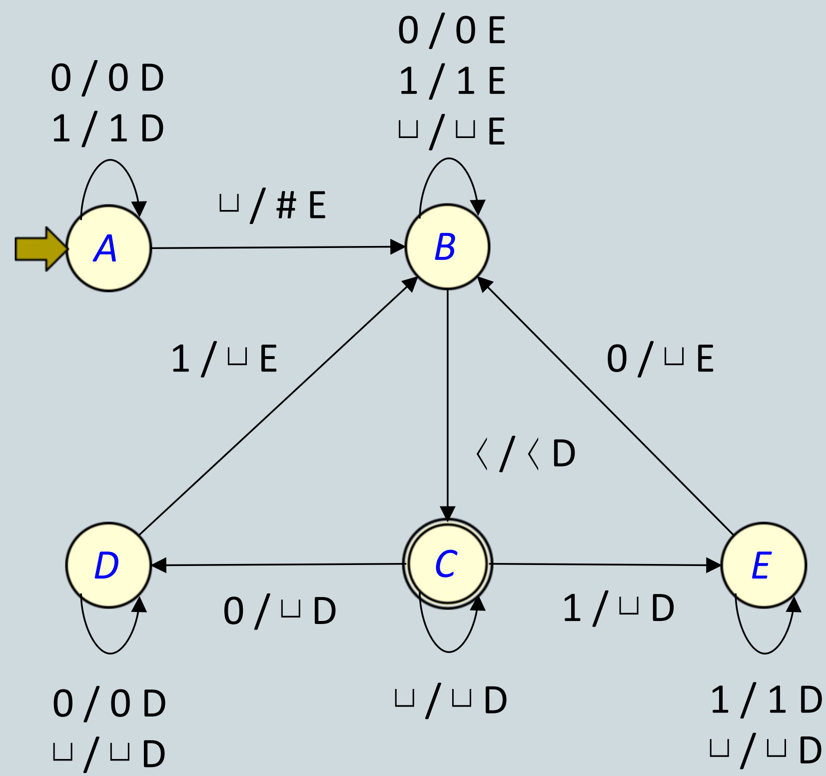
Parte 4

- Se o primeiro símbolo for 1, o processo é simétrico
  1. Apaga o 1
  2. Avança até o primeiro 0
  3. Apaga o 0
  4. Retorna o cabeçote
- A máquina pára no estado E se encontrar # após apagar o 1



Exemplo 02: MT reconhecedora

{ $w \in \{0,1\}^*$ | o número de 0s é igual ao de 1s }



Exercícios

1. Faça uma máquina de Turing que reconheça cada linguagem abaixo:
   a) $\{a^n b^n c^n \mid n \geq 0\}$
   b) $\{w\#w \mid w \in \{0,1\}^*\}$
   c) $\{wcw^R \mid w \in \{a,b\}^*\}$

2. Faça uma máquina de Turing que realize uma cópia da palavra de entrada, com alfabeto {a,b} – a palavra de entrada deve ser w⊔ e saída deve ser w⊔w

3. Construa uma Máquina de Turing que insira um espaço em branco entre cada símbolo de entrada, com alfabeto {a,b}. Por exemplo, se entrada for aba⊔, a saída deve ser a⊔b⊔a⊔.

Modelos Equivalentes à Máquina de Turing

OBJETIVO

COMPREENDER DIFERENTES MODELOS DE MT

Introdução

- Uma razão para considerar a máquina de Turing como o mais geral dispositivo de computação:
  - Todos os demais modelos e máquinas propostos, bem como as diversas modificações da máquina de Turing possuem, no máximo, o mesmo poder computacional da máquina de Turing.

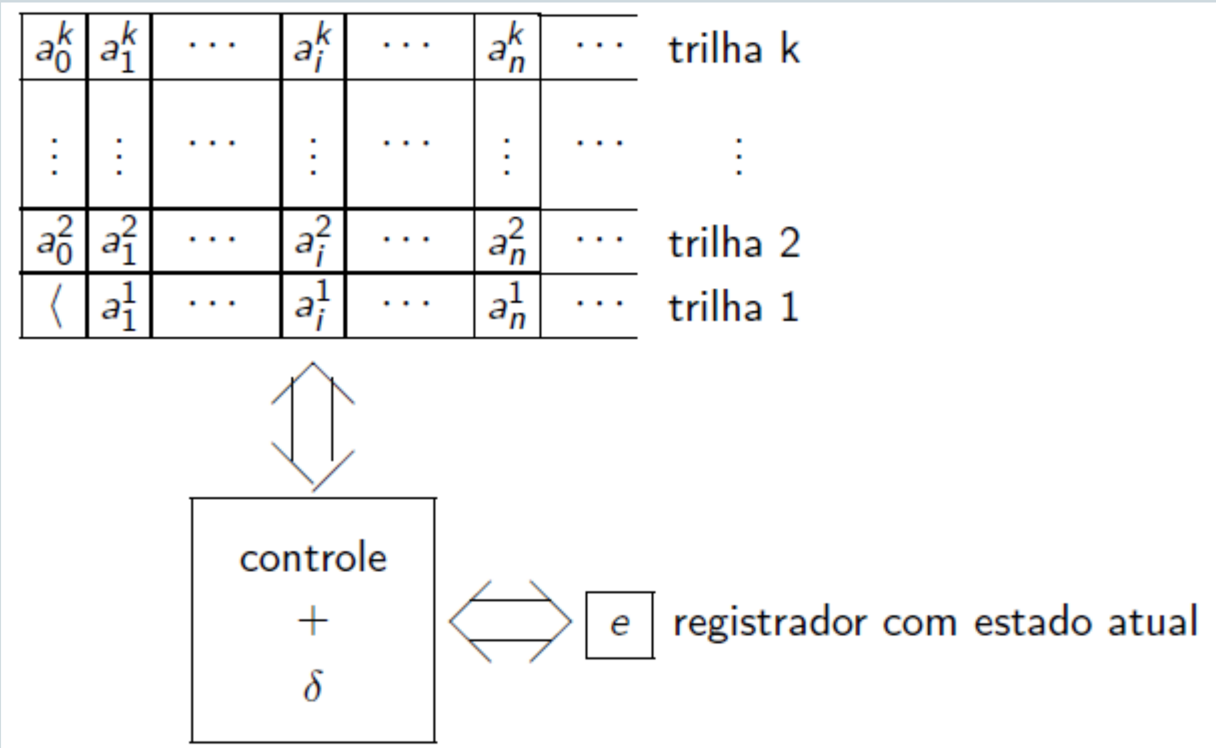
- **Alonzo Church (1936)**
  - **Hipótese de Church**
    - *Qualquer função computável pode ser processada por uma máquina de Turing* → existe um procedimento expresso na forma de uma máquina de Turing capaz de processar a função.

Máquina de Turing com Cabeçote Imóvel

- Essa MT acrescenta, na função de transição, a possibilidade de não mover a cabeça da fita a cada movimento.
- Pode ocorrer a transição (e, a) = [e′, b, I] , em que I indica que o cabeçote deve ficar imóvel.
- Uma transição (e, a) = [e′, b, I] pode ser simulada por transições das formas a seguir, sendo d um novo estado:
  - (e, a) = [d, b,D];
  - (d, c) = [e′, c, E] para cada c $\in$ Γ.
- O poder computacional de uma MT cujo cabeçote pode ficar imóvel não é maior que o de uma MT-padrão.

Máquina de Turing com Múltiplas Trilhas

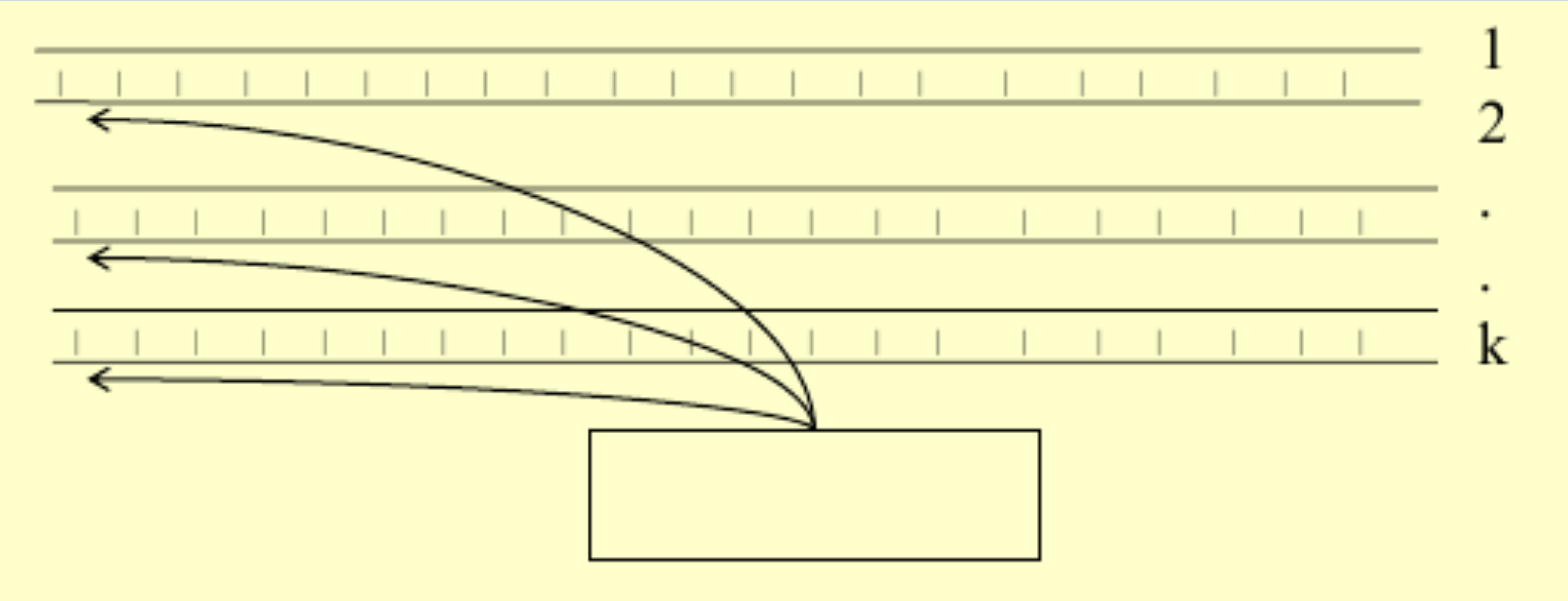
- A fita é composta por múltiplas trilhas: cada célula contém uma k-upla de símbolos.
- No início, a palavra de entrada está na trilha 1.



Máquina de Turing com com Múltiplas Fitas

- Definição: Uma MT com **k** fitas consiste de um controle finito e k fitas de trabalho, cada uma conectada com o controle finito por meio de uma cabeça de fita.
- A função de transição para esta máquina, com base no estado atual, escolhe uma fita para ler, escrever e mover à esquerda ou à direita
- A máquina para quando sua função de transição é indefinida.
- A entrada para a MT é colocada em algumas fitas, designadas de entrada; e a saída, se houver, é obtida de algumas fitas designadas de saída.
- Se a MT for reconhecedora, um subconjunto de estados é designado como final.

Máquina de Turing com com Múltiplas Fitas



Eduardo Freire Nakamura (nakamura@icomp.ufam.edu.br)

Máquinas de Turing com Fita Infinita à Esquerda e à Direita

- Fita infinita dos dois lados não aumenta o poder computacional.
- Pode ser facilmente simulada por uma fita tradicional
  - Células pares: parte direita da fita
  - Células ímpares: parte esquerda da fita

...	a_{-3}	a_{-2}	a_{-1}	a_1	a_2	a_3	...

〈	a_1	a_{-1}	a_2	a_{-2}	a_3	a_{-3}	...

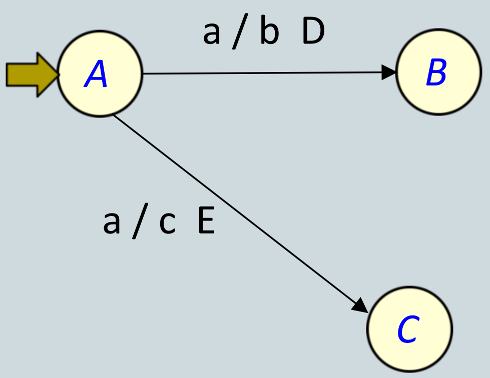
MT Não Determinísticas

- Uma MTND difere da determinística pelo fato de que, para cada estado q e símbolo de fita X, $\delta(q, X)$ é um conjunto de triplas $\{(q_1,Y_1,D_1),\ldots(q_k,Y_k,D_k)\}$

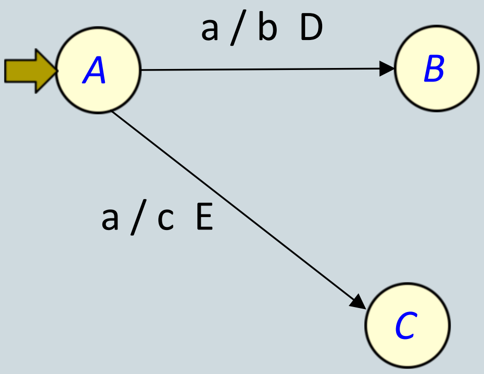
- A linguagem aceita por uma MTND é dada pelo conjunto de cadeias para as quais haja ao menos uma sequência de escolhas de movimentos que a leve de q_1 a um estado final.

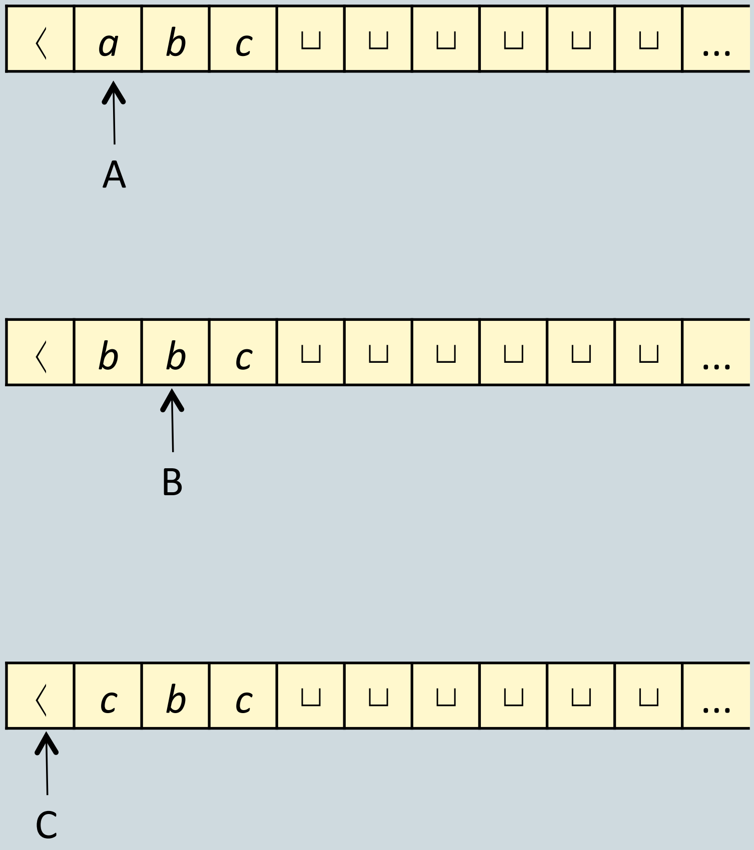
- **Teorema:** Se MN é uma MTND, então existe uma MT determinística MD tal que L(MN) = L(MD).
  - Ou seja, as MTND não aceitam nenhuma linguagem não aceita por uma MTD.

MT Não Determinísticas



MT Não Determinísticas





MT Não Determinísticas

- Exemplo:
  - $\{ ww \mid w \in \{0,1\}^* \}$

Modificações combinadas sobre a Máquina de Turing

- Combinação de algumas ou todas as modificações não aumenta o poder computacional da máquina de Turing

- **MT e os Computadores**
  - Aceitam o mesmo conjunto de linguagens – as linguagens recursivamente enumeráveis;
  - Mostra-se isso simulando-se uma MT por um computador e simulando-se um computador por uma MT (de várias fitas – cada fita um recurso: memória, contador de instruções, arquivo de E/S).

Gramáticas Irrestritas e Linguagens Sensíveis ao Contexto

OBJETIVO

COMPREENDER GRAMÁTICAS IRRESTRITAS E SENSÍVEIS AO CONTEXTO

Gramáticas Irrestritas

- Uma linguagem é recursivamente enumerável se existe uma MT que aceita toda palavra da linguagem, e não aceita palavras que não pertencem à linguagem.
  - "Não aceita" não é o mesmo que "rejeita" – a MT pode entrar em um loop infinito e nunca parar para aceitar ou rejeitar a palavra.
- Uma linguagem L é recursivamente enumerável se, e somente se, L é gerada por uma gramática irrestrita.

Gramáticas Irrestritas

- **Definição:** Uma Gramática Irrestrita é uma quádrupla:

$G = (V, \Sigma, R, P)$

1. V é um conjunto finito de objetos denominados *variáveis*
2. Σ é um conjunto finito, disjunto de V, denominado *terminais*
3. R é um conjunto finito de *regras de produção*, onde toda regra é da forma

$u \rightarrow v$, onde $u \in (\Sigma \cup V)^+$ e $v \in (\Sigma \cup V)^*$

Obs: Qualquer número de variáveis e terminais pode ocorrer, em qualquer ordem, tanto do lado direito quanto do lado esquerdo da produção $\rightarrow$, λ não é permitido no lado esquerdo.

4. $P \in V$ é a *variável inicial*.

Gramáticas Irrestritas

- **Exemplo 1**: Gramática Irrestrita que gera $L = \{a^n b^n c^n \mid n \geq 0\}$

  G = ({ P, C }, { a, b, c }, R, P)

  R =

  $P \rightarrow abc \mid \lambda$

  $ab \rightarrow aabbC$

  $Cb \rightarrow bC$

  $Cc \rightarrow cc$

  - Derivação de aaabbbccc

  $P \Rightarrow abc \Rightarrow aabbCc \Rightarrow aaabbCbCc \Rightarrow aaabbCbcc \Rightarrow aaabbbCcc \Rightarrow aaabbbccc$

Gramáticas Irrestritas

- **Exemplo 2**: Gramática Irrestrita que gere $L = \{a^n b^n a^n b^n \mid n \geq 1\}$

  G = ({ P, A, B C }, { a, b}, R, P)

  R = P → aAbab

  aAb → aabbA | ab

  bAb → bbA

  bAa → baB

  aBa → aaB

  aBb → Caabb

  aCa → Caa

  bC → Cb

  aCb → aAb

  - Derivação de $a^3b^3a^3b^3$

S → aAbab → aabbAab → aabbaBb → aabbCaabb→ aabCbaabb → aaCbbaabb
→ aaAbbaabb → aaabbAbaabb → aaabbbAaabb → aaabbbaBabb
→ aaabbbaaBbb → aaabbbaCaabbb → aaabbbCaaabbb → aaabbCbaaabbb
→ aaabCbbaaabbb → aaaCbbbaaabbb → aaaAbbbaaabbb → aaabbbaaabbb

Gramáticas Irrestritas

- **Teorema:** Toda linguagem gerada por uma gramática irrestrita é recursivamente enumerável.
  - Uma gramática irrestrita pode ser simulada em uma MT.
  - Uma MT pode ser simulada em uma gramática irrestrita.
  - (não será demonstrado)

Gramáticas Sensíveis ao Contexto

- **Definição:** Uma Gramática Sensível ao Contexto é uma quádrupla:

  $G = (V, \Sigma, R, P)$

1. V é um conjunto finito de objetos denominados ***variáveis***.
2. Σ é um conjunto finito, disjunto de V, denominado ***terminais***
3. R é um conjunto finito de ***regras de produção***, onde toda regra é da forma

   $u \rightarrow v$ **, e** $u \in (\Sigma \cup V)^{+}$ **e** $v \in (\Sigma \cup V)^{*}$ **e** $|u| <= |v|$ **,** excetuando-se $P \rightarrow \lambda$. Nesse caso, não pode haver recursão com P.

4. $P \in V$ é a ***variável inicial***.

Gramáticas Sensíveis ao Contexto

- Portanto, em uma gramática sensível ao contexto, a cada etapa de derivação o tamanho da palavra derivada *não* pode diminuir
  - excetuando-se para gerar a palavra vazia λ se esta pertencer à linguagem
- **Sensíveis ao Contexto** – razão?
  - Foi provado que todas as gramáticas sensíveis ao contexto podem ser reescritas em uma forma normal em que todas as regras de produção sejam da forma:

    xAy → xvy

  - Equivalente a dizer que a produção **A → v** pode ser aplicada somente quando A ocorre com x do lado esquerdo e y do lado direito.

Gramáticas Sensíveis ao Contexto

- **Exemplo 1**: Gramática sensível ao contexto que gere $L = \{a^n b^n c^n \mid n \geq 1\}$

  G = ({ P, A,B }, { a, b, c }, R, P)

  R =

  P → abc | aAbc

  Ab → bA

  Ac → Bbcc

  bB → Bb

  aB → aa | aaA

  - Derivação de $a^3b^3c^3$

S ⇒ aAbc ⇒ abAc ⇒ abBbcc ⇒ aBbbcc ⇒ aaAbbcc ⇒ aabAbcc ⇒ aabbAcc ⇒ aabbBbccc ⇒ aabBbbccc ⇒ aaBbbbccc ⇒ aaabbbccc

Gramáticas Sensíveis ao Contexto

- **Exemplo 2**: Gramática sensível ao contexto que gere $L = \{a^n b^{2n} a^n \mid n \geq 1\}$

  $G = (\{ P, A, B, C \}, \{ a, b\}, R, P)$

  R =

  $P \rightarrow abba \mid aAbba$

  $aAb \rightarrow aabbbA$

  $Ba \rightarrow Caa \mid aa$

  $bCa \rightarrow Cba$

  $bC \rightarrow Cb$

  $aCb \rightarrow aAb$

Gramáticas Sensíveis ao Contexto

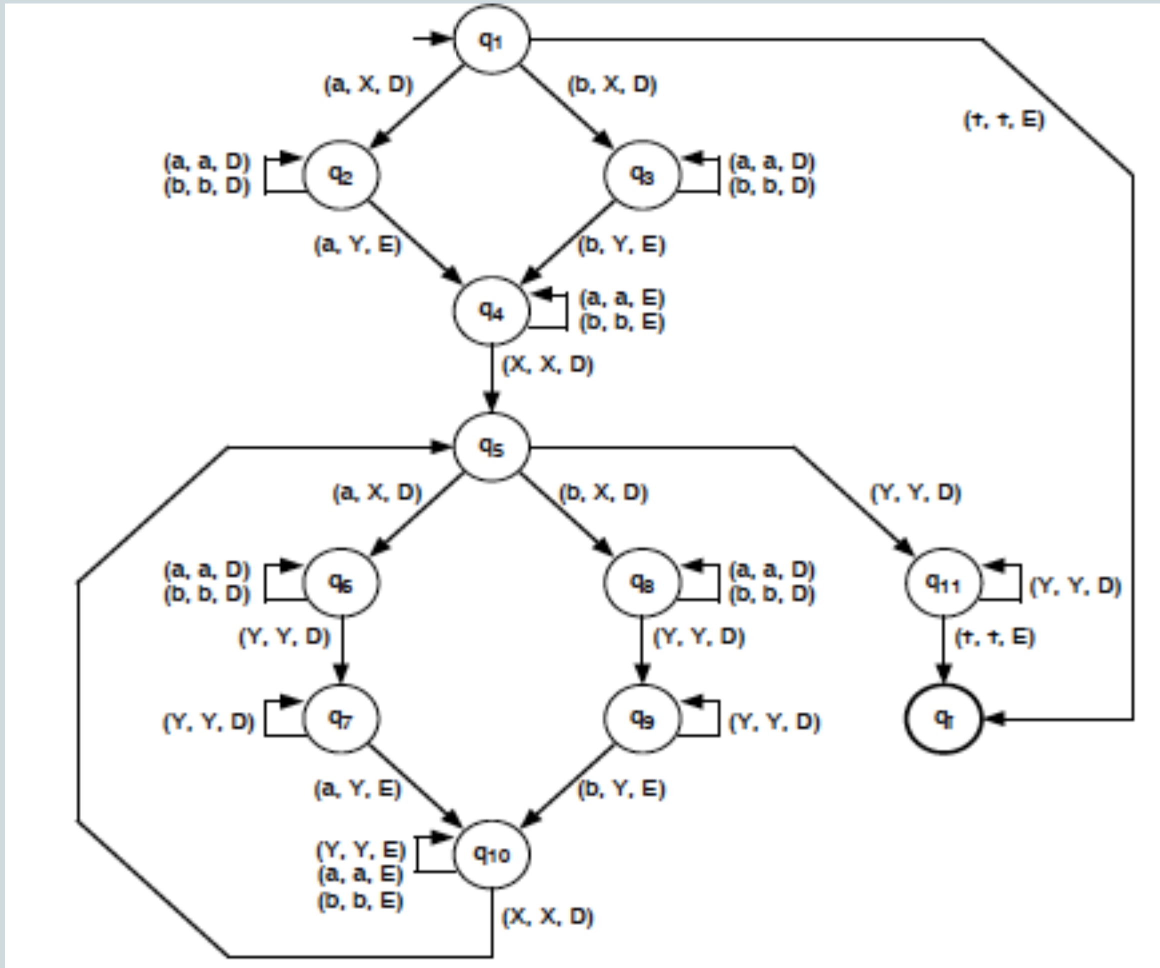
- As gramáticas sensíveis ao contexto geram linguagens sensíveis ao contexto.
- As linguagens sensíveis ao contexto podem ser reconhecidas por Máquinas de Turing com Fita Limitada.
- **MT com Fita Limitada**
  - Fita limitada ao tamanho da entrada
  - Mais duas células: marcadores de início e de fim de fita

M = (E, Σ, Γ, 〈, † , δ, i, F)

- 〈 - símbolo de início ou marcador de início da fita
- † - símbolo de fim ou marcador de fim da fita

Gramáticas Sensíveis ao Contexto

- **Máquina de Turing com Fita Limitada**
  - **Exemplo:** MT com Fita Limitada que reconheça a $L = \{ww \mid w \in \{a,b\}^*\}$



q1
(a, X, D)
(b, X, D)
(†, †, E)
q2
(a, a, D)
(b, b, D)
q3
(a, a, D)
(b, b, D)
(a, Y, E)
(b, Y, E)
q4
(a, a, E)
(b, b, E)
(X, X, D)
q5
(a, X, D)
(b, X, D)
(Y, Y, D)
q6
(a, a, D)
(b, b, D)
q8
(a, a, D)
(b, b, D)
q11
(Y, Y, D)
(Y, Y, D)
(Y, Y, D)
(†, †, E)
q7
(Y, Y, D)
q9
(Y, Y, D)
qf
(a, Y, E)
(b, Y, E)
q10
(Y, Y, E)
(a, a, E)
(b, b, E)
(X, X, D)

Gramáticas Sensíveis ao Contexto

- **Exercício**

1. Faça uma MT que reconheça a seguinte linguagem:

a) $a^*b(a+b)^*$, onde $w \in \{a,b\}^*$

2. Faça a MT com fita limitada que reconheça a seguinte linguagem:

a) $\{a^n b^{2n} a^n \mid n \geq 1\}$

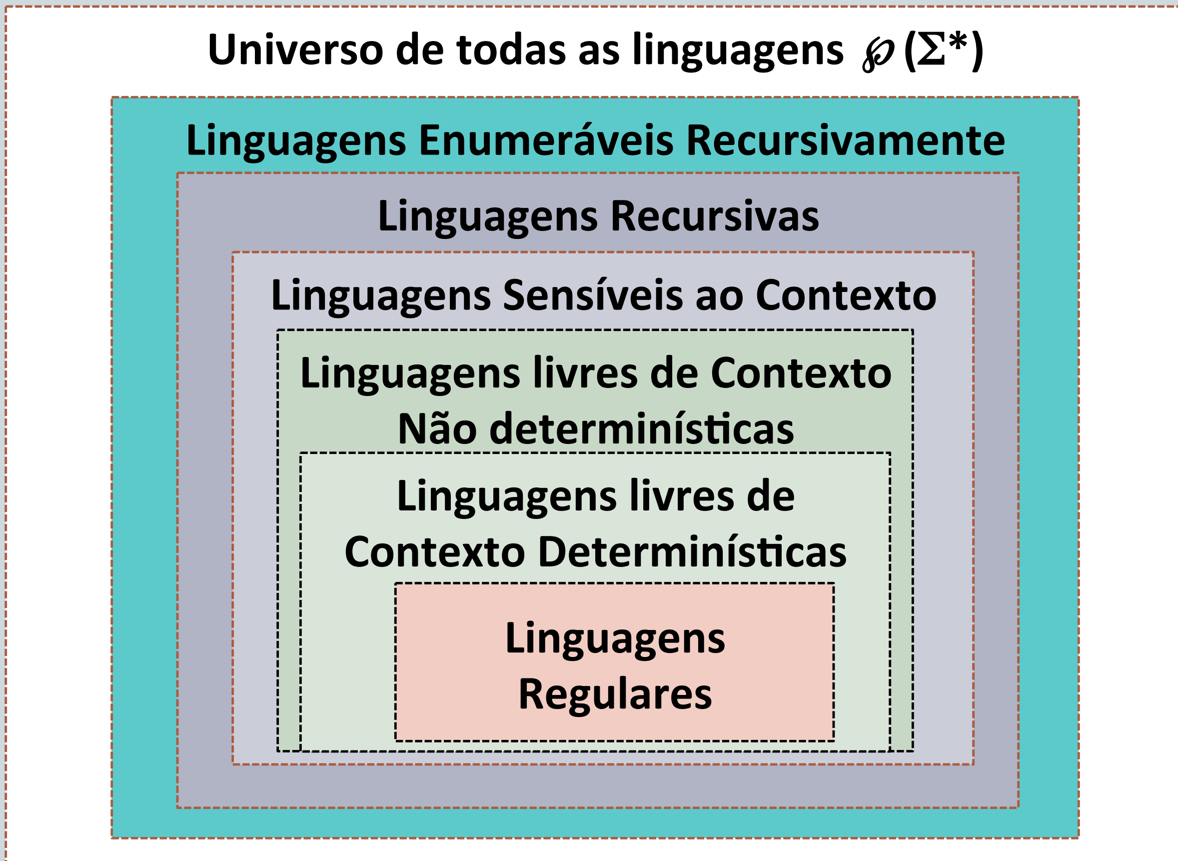
Sumarizando as relações gramática/linguagem

- As gramáticas sensíveis ao contexto produzem uma classe de linguagens localizada entre as linguagens livres de contexto e as linguagens recursivas *.
- As Linguagens Sensíveis ao Contexto são fechadas para operação de união, concatenação e intersecção.
  - A construção do resultado dessas operações é similar às feitas nas Linguagens Regulares e Livres de Contexto.
- A classe das Linguagens Recursivas é fechada em relação às operações de união, concatenação, complementação e intersecção.

* Reconhecidas por Máquinas de Turing que sempre param

Sumarizando as relações gramática/linguagem

Universo das linguagens:



Sumarizando as relações gramática/linguagem

- Linguagens Recursivamente enumeráveis:
  - Gramática: Irrestrita
  - Reconhecedor: Máquinas de Turing
- Linguagens Recursivas:
  - Gramática: Irrestrita
  - Reconhecedor: Máquinas de Turing que sempre param
- Linguagens Sensíveis ao Contexto:
  - Gramática: Sensível ao Contexto
  - Reconhecedor: Máquinas de Turing Limitadas linearmente
- Linguagens Livres de Contexto:
  - Gramática: Livre de Contexto
  - Reconhecedor: Autômatos com Pilha
- Linguagens Regulares:
  - Gramática: Regular
  - Reconhecedor: Autômatos Finito

Sumarizando as relações gramática/linguagem

- Exercício: Resuma o formato das regras de produção das seguintes gramáticas:

  - Gramáticas Irrestritas
  - Gramáticas Sensíveis ao Contexto
  - Gramáticas Livres de Contexto
  - Gramáticas Regulares