ACH2043 INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 10

Cap. 2.3 – Linguagens não livres de contexto
Cap 3 – Máquinas de Turing

Profa. Ariane Machado Lima
ariane.machado@usp.br

Lema do bombeamento para linguagens livres de contexto

Lema do bombeamento para linguagens livres de contexto

- Para cadeias “longas” da linguagem, sua árvore sintática deve conter pelo menos um caminho “longo” da variável inicial (raiz) até um terminal (folha)

Lema do bombeamento para linguagens livres de contexto

TEOREMA 2.34

Lema do bombeamento para linguagens livres-do-contexto Se A é uma linguagem livre-do-contexto, então existe um número p (o comprimento de bombeamento) onde, se s é uma cadeia qualquer em A de comprimento pelo menos p, então s pode ser dividida em cinco partes $s = uvxyz$ satisfazendo as condições

1. para cada $i \geq 0$, $uv^ixy^iz \in A$,
2. $|vy| > 0$, e
3. $|vxy| \leq p$.

Lema do bombeamento para linguagens livres de contexto (prova)

- b: nr máximo de símbolos do lado direito de uma regra
- Árvore sintática:
  - um nó não pode ter mais de b filhos
  - Cada nível n tem no máximo b^n folhas
  - Árvore de altura no máximo h: cadeia de tamanho no máximo b^h
- Se uma cadeia tem tamanho >= b^h +1, então suas árvores têm altura >= h+1
- Comprimento de bombeamento: $b^{|V|}$ +1 (|V| = nr de variáveis de G)
- Se s pertence à LLC A, |s| >= $b^{|V|}$ +1, então as árvores de s têm altura >= |V| +1
- Escolha a árvore com menor número de nós. Nela existe pelo menos um caminho com comprimento >= |V| + 1 (1 nó folha e |V|+1 nós de variáveis) => pelo menos uma variável se repete (escolhemos R, que se repete entre as |V|+1 variáveis mais baixas desse caminho)

Lema do bombeamento para linguagens livres de contexto (prova)

- Dividimos s em uvxyz (figura)
- Podemos substituir cada subárvore de R pela outra, obtendo uv^ixy^iz (i > 1) ou uxz (cond 1 do lema)
- (cond 2 : |vy| > 0): Se vy = ε, então poderíamos trocar a subárvore maior de R pela menor, e a nova árvore (com menos nós) ainda geraria s. Mas tal árvore não existe, pois já tínhamos escolhido a menor.
- (cond 3 : |vxy| <= p): vxy é gerada pela ocorrência superior de R, e R foi escolhida dentre as |V|+1variáveis mais baixas do caminho => subárvore maior de R tem altura <= |V|+1 => sequência gerada por essa subárvore (vxy) tem tamanho <= $b^{|V|+1}$ => $|vxy| <= b^{|V|} + 1$

Exemplos

Prove que $B = \{a^n b^n c^n \mid n >= 0\}$ não é livre de contexto

–

–

Exemplos

Prove que B = $\{a^n b^n c^n \mid n >= 0\}$ não é livre de contexto

Assume B é LLC, então vale Lema do Bombeamento

$S = a^p b^p c^p$

Duas situações:

- v e y contêm o mesmo símbolo (viola proporção)
- v e y contêm diferentes símbolos (viola ordem)

Contradição!

Exemplos

Prove que $C = \{a^i b^j c^k \mid 0 <= i <= j <= k\}$ não é livre de contexto

Exemplos

Prove que $C = \{a^i b^j c^k \mid 0 <= i <= j <= k\}$ não é livre de contexto

Assume C é LLC, então vale Lema do Bombeamento

$S = a^p b^p c^p$

Duas situações:

v e y contêm o mesmo símbolo

- A´s não aparecem: $uv^0xy^0z = uxz$ contém menos b's ou c's
- B's não aparecem:
  - A's aparecem: uv^2xy^2z contém mais a's que b's
  - C's aparecem: uv^0xy^0z contém mais b's que c's
- C's não aparecem:uv^2xy^2z contém mais a's ou mais b's que c's

v e y contêm diferentes símbolos (viola ordem)

Contradição!

Exemplos

Prove que $D = \{ w w \mid w \in \{0,1\}^* \}$ não é livre de contexto

Exemplos

Prove que $D = \{ w w \mid w \in \{0,1\}^* \}$ não é livre de contexto

Assume D é LLC, então vale Lema do Bombeamento

$S = 0^p 1^p 0^p 1^p$

vxy deve passar da metade de s para manter a duplicação ao bombear para cima

Mas nesse caso, quando se bombeia para baixo até uxz, ela tem a forma $0^p 1^i 0^j 1^p$ onde i e j podem não ser p

Contradição!

Cap. 3

A tese de Church-Turing

Cap. 3 - A tese de Church-Turing

3.1 – Máquinas de Turing

3.2 – Variantes da Máquinas de Turing

3.3 – A Definição de Algoritmo

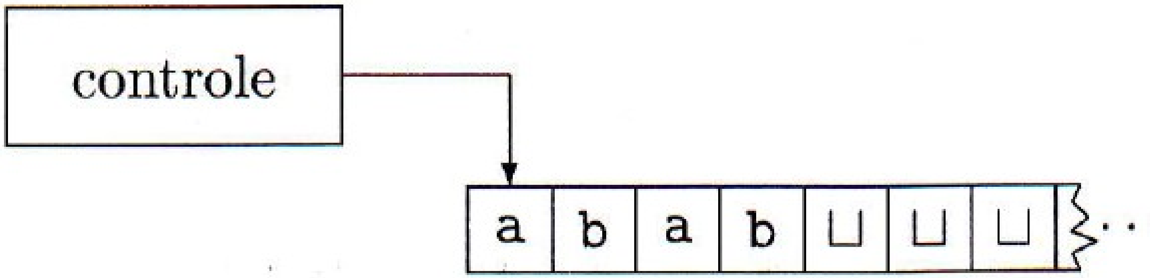
3.1 - Máquinas de Turing

- Autômatos como modelos de computação:
  - AF: memória pequena
  - AP: memória ilimitada mas utilizável apenas em sistema LIFO (last in, first out) de leitura

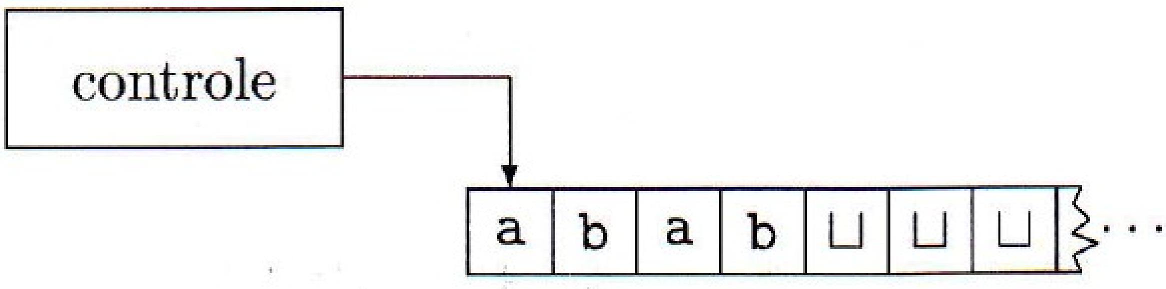
3.1 - Máquinas de Turing

- Propostas por Alan Turing em 1936
  - Memória ilimitada e irrestrita
  - Modelo de um computador real (possibilidades e limitações)

Máquinas de Turing



Máquinas de Turing



1. Uma máquina de Turing pode tanto escrever sobre a fita quanto ler a partir dela.
2. A cabeça de leitura-escrita pode mover-se tanto para a esquerda quanto para a direita.
3. A fita é infinita.
4. Os estados especiais para rejeitar e aceitar fazem efeito imediatamente.

Máquinas de Turing - Exemplo

B= { w # w | w pertence a {0,1}*}

Máquinas de Turing - Exemplo

B= { w # w | w pertence a {0,1}*}

M_1 = "Sobre a cadeia de entrada w:

1. Faça um zigue-zague ao longo da fita checando posições correspondentes de ambos os lados do símbolo # para verificar se elas contêm o mesmo símbolo. Se elas não contêm, ou se nenhum # for encontrado, *rejeite*. Marque os símbolos à medida que eles são verificados para manter registro de quais símbolos têm correspondência.
2. Quando todos os símbolos à esquerda do # tiverem sido marcados, verifique a existência de algum símbolo remanescente à direita do #. Se resta algum símbolo, *rejeite*; caso contrário, *aceite*."

Máquinas de Turing - Exemplo

```
0 1 1 0 0 0 # 0 1 1 0 0 0 ⊔ ···
x 1 1 0 0 0 # 0 1 1 0 0 0 ⊔ ···
x 1 1 0 0 0 # x 1 1 0 0 0 ⊔ ···
x 1 1 0 0 0 # x 1 1 0 0 0 ⊔ ···
x x 1 0 0 0 # x 1 1 0 0 0 ⊔ ···
x x x x x x # x x x x x x ⊔ ···
                          aceita
```

Máquinas de Turing – Definição formal

Uma ***máquina de Turing*** é uma 7-upla, $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{aceita}}, q_{\text{rejeita}})$, onde Q, Σ, Γ são todos conjuntos finitos e

1. Q é o conjunto de estados,
2. Σ é o alfabeto de entrada sem o ***símbolo em branco*** $\sqcup$,
3. Γ é o alfabeto de fita, onde $\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$,
4. $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\mathrm{E}, \mathrm{D}\}$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial,
6. $q_{\text{aceita}} \in Q$ é o estado de aceitação, e
7. $q_{\text{rejeita}} \in Q$ é o estado de rejeição, onde $q_{\text{rejeita}} \neq q_{\text{aceita}}$.