[Aula 15] LLC – Autômato com Pilha

Prof. João F. Mari
joaof.mari@ufv.br

BIBLIOGRAFIA

- MENEZES, P. B. **Linguagens formais e autômatos**, 6. ed., Bookman, 2011.
  - Capítulo 6.
  - + Slides disponibilizados pelo autor do livro.

ROTEIRO

- Autômato com Pilha
- Definição do Autômato com Pilha
- Autômato com Pilha (Não-Determinístico)
- EXEMPLO: $\{a^n b^n \mid n >= 0\}$ Duplo Balanceamento
- EXEMPLO: Palavra e sua Reversa
- EXEMPLO: Autômato com Pilha: $a^n b^m a^{n+m}$
- AP e Linguagens Livres do Contexto
- GLC → AP
- EXEMPLO: GLC → AP: L5 = $\{ a^n b^n \mid n \geq 1 \}$
- Corolários

Autômato com Pilha

- Classe das Linguagens Livres do Contexto
  - Pode ser associada a um formalismo do tipo autômato
  - Autômato com Pilha

- Autômato com pilha
  - Análogo ao autômato finito
  - Incluindo uma pilha como memória auxiliar
  - Não determinismo

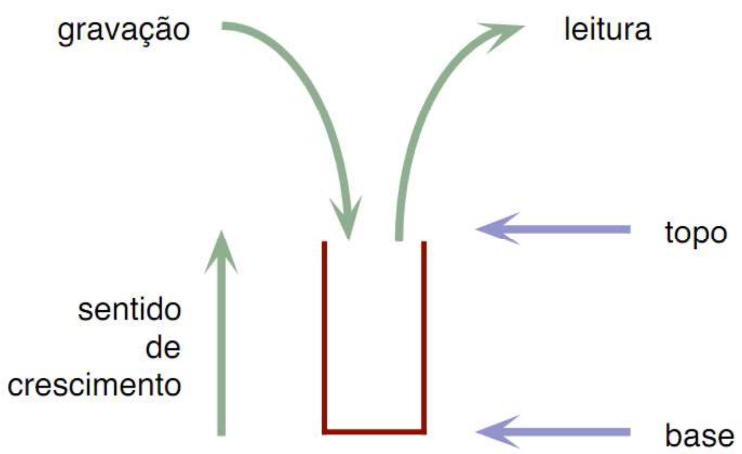
- Pilha
  - Independente da fita de entrada
  - Não possui limite máximo de tamanho
    - “tão grande quanto necessário”
    - Baseada na noção de conjunto infinitamente contável

Autômato com Pilha

- Não-determinismo: importante e necessário
  - Aumenta o poder computacional dos AP
- Exemplo:
  - { ww^r | w é palavra sobre { a, b } }
  - Reconhecimento só é possível por um AP Não-Determinístico
    - * w^r é a palavra w escrita ao contrário.

Autômato com Pilha

- Estrutura de uma pilha
  - Último símbolo gravado é o primeiro a ser lido
  - Base: fixa e define o seu início
  - Topo: variável e define a posição do último símbolo gravado



Autômato com Pilha

- AP × Número de estados
  - Qualquer LLC pode ser reconhecida por um AP
    - Com somente um estado
    - (ou três estados, dependendo da definição)
  - Pilha é suficiente como única memória
    - Os estados não são necessários para "memorizar" informações passadas
      - Ao contrário do que ocorria com os Autômatos Finitos.
  - Estados no AP
    - Poderiam ser excluídos sem reduzir o poder computacional
  - Como a pilha não possui tamanho máximo
    - AP pode assumir tantos estados quanto se queira

Definição do Autômato com Pilha

- Duas definições universalmente aceitas:
  - Estados finais
    - O AP PARA **aceitando** quando atinge um estado final
    - Inicialmente a pilha é vazia
  - Pilha vazia
    - O AP PARA aceitando quando a pilha estiver vazia.
    - Inicialmente, a pilha possui um símbolo inicial da pilha
    - Não existem estados finais
  - São definições equivalentes (possuem o mesmo poder computacional)
    - Adotada a definição que usa estados finais.

Autômato com Pilha (Não-Determinístico)

M = (Σ, Q, δ, q0, F, V)

- Σ - alfabeto de símbolos de entrada
- Q - conjunto de estados possíveis (finito)
- δ - (função) programa ou **função de transição**
  - função parcial

$$\delta: Q \times (\Sigma \cup \{ \varepsilon, ? \}) \times (V \cup \{ \varepsilon, ? \}) \rightarrow 2^{Q \times V^*}$$

$$\delta(p, x, y) = \{ (q1, v1),\ldots,(qn, vn) \} \text{ transição}$$

- q0 - elemento distinguido de Q: estado inicial
- F - subconjunto de Q: conjunto de estados finais
- V - alfabeto auxiliar ou alfabeto da pilha

- ? → Teste de Pilha Vazia ou Teste de Final de Fita.

Autômato com Pilha (Não-Determinístico)

- Características da função de transição (função programa)
  - função parcial
  - "?" indica teste de:
    - pilha vazia
    - toda palavra de entrada lida

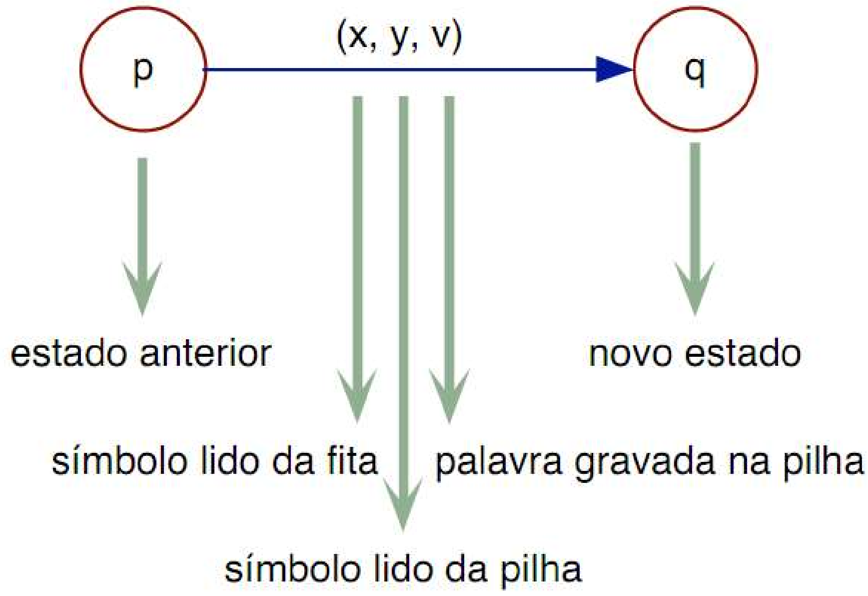
- leitura de ε indica
  - movimento vazio da fita ou pilha (não lê, nem move a cabeça)
  - não-determinístico: basta que o movimento seja vazio na fita

- gravação de ε
  - nenhuma gravação é realizada na pilha (e não move a cabeça)

- Exemplo: **δ(p, ?, ε) = { (q, ε) }**
  - no estado p, se a entrada foi completamente lida, não lê da pilha
  - assume o estado q e não grava na pilha

Autômato com Pilha (Não-Determinístico)

- Programa como diagrama: $\delta(p, x, y) = \{ (q, v) \}$



Autômato com Pilha (Não-Determinístico)

- Computação de um AP
  - Sucessiva aplicação da função programa
    - Para cada símbolo da entrada (da esquerda para a direita)
    - Até ocorrer uma condição de parada
  - É possível que nunca atinja uma condição de parada
    - processa indefinidamente (loop infinito)
    - EXEMPLO: empilha e desempilha um mesmo símbolo indefinidamente, sem ler da fita

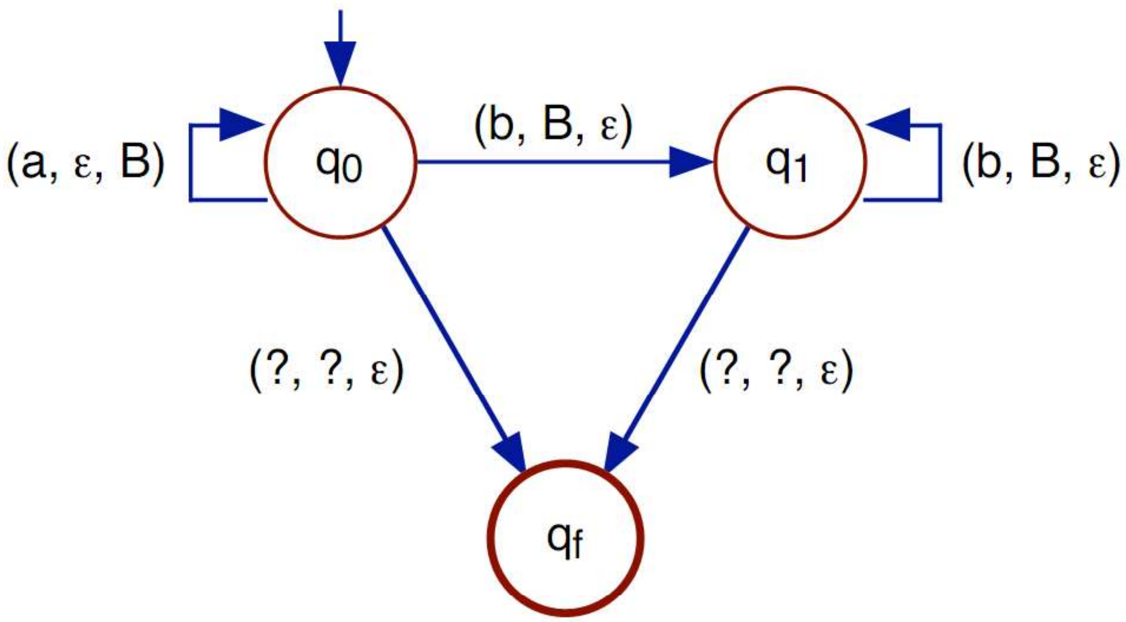
Autômato com Pilha (Não-Determinístico)

- Parada de um AP
  - Aceita
    - Pelo menos um dos caminhos alternativos atinge um estado final
    - não importa se leu ou não toda a entrada
  - Rejeita
    - todos os caminhos alternativos rejeitam a entrada
    - a função programa é indefinida para cada caso
  - Loop
    - pelo menos um caminho alternativo está em loop infinito
    - E os demais rejeitam ou também estão em loop infinito

- Partição de Σ* induzida por um AP M

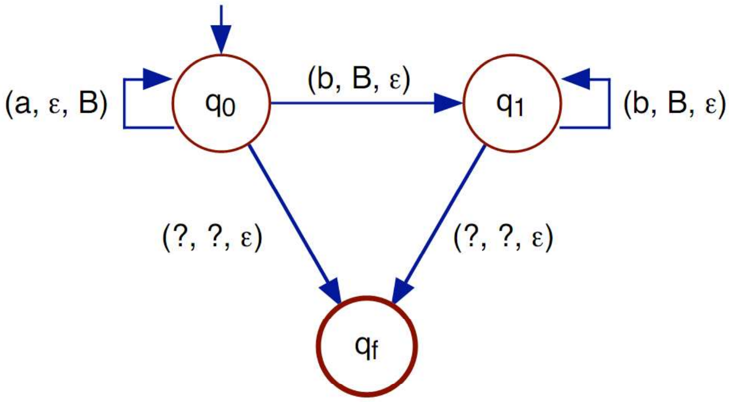
EXEMPLO: $\{a^n b^n \mid n >= 0\}$ Duplo Balanceamento

M1 = ({ a, b }, { q0, q1, qf }, δ1, q0, { qf }, { B })



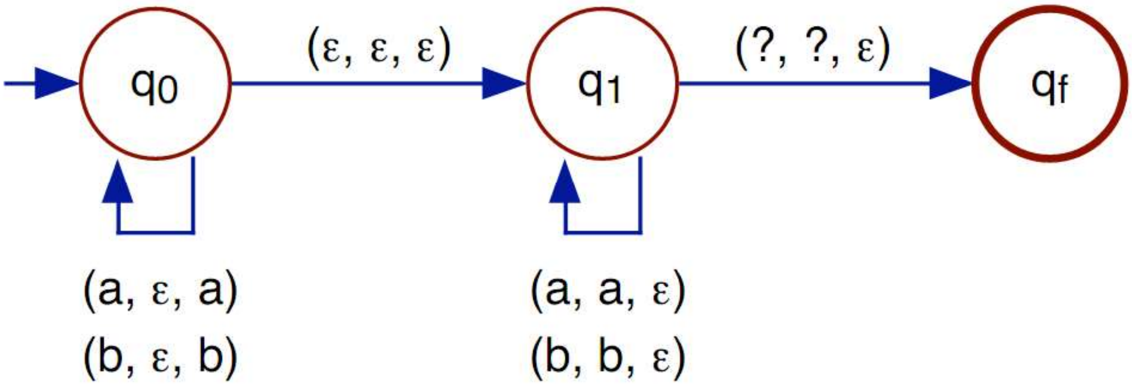
EXEMPLO: $\{a^n b^n \mid n >= 0\}$ Duplo Balanceamento

- M1 = ({ a, b }, { q0, q1, qf }, δ1, q0, { qf }, { B })
  - AP determinístico
  - ACEITA(M1) = L1
    - δ1 (q0, a, ε) = { (q0, B) }
    - δ1 (q0, b, B) = { (q1, ε) }
    - δ1 (q0, ?, ?) = { (qf, ε) }
    - δ1 (q1, b, B) = { (q1, ε) }
    - δ1 (q1, ?, ?) = { (qf, ε) }



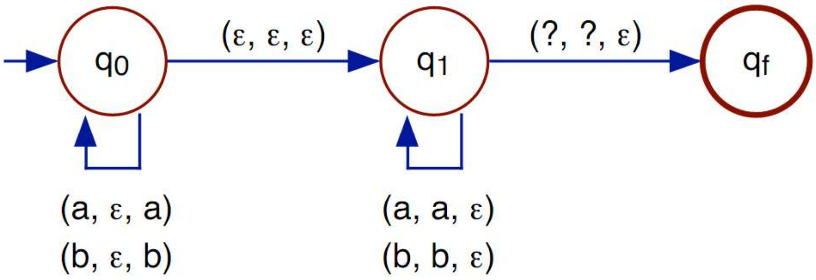
EXEMPLO: Palavra e sua Reversa

- L3 = { ww^r | w pertence a { a, b }* }
  - ACEITA(M3) = L3
  - AP não-determinístico



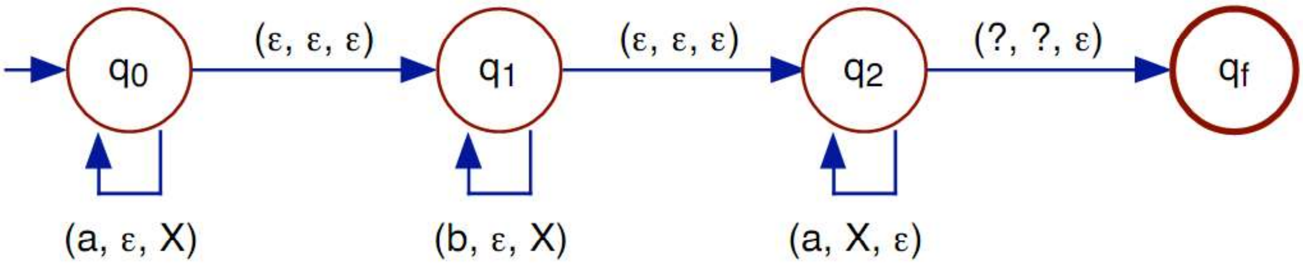
EXEMPLO: Palavra e sua Reversa

- L3 = { ww^r | w pertence a { a, b }* }
  - ACEITA(M3) = L3
  - AP não-determinístico



EXEMPLO: Autômato com Pilha: $a^nb^ma^{n+m}$

- L4 = { $a^nb^ma^{n+m}$ | $n \geq 0$, $m \geq 0$ }
  - ACEITA(M4) = L4
  - AP não-determinístico



AP e Linguagens Livres do Contexto

- A classe de linguagens aceitas por um AP
  - É a classe das LLC
  - A mesma classe das linguagens geradas pelas GLC
- Construção de um AP a partir de uma GLC qualquer, permite concluir
  - Construção de um reconhecedor para uma LLC a partir de sua gramática é simples e imediata.
  - Qualquer LLC pode ser aceita por um AP com somente um estado (ou três, dependendo da definição).
    - Estados não aumentam o poder computacional.

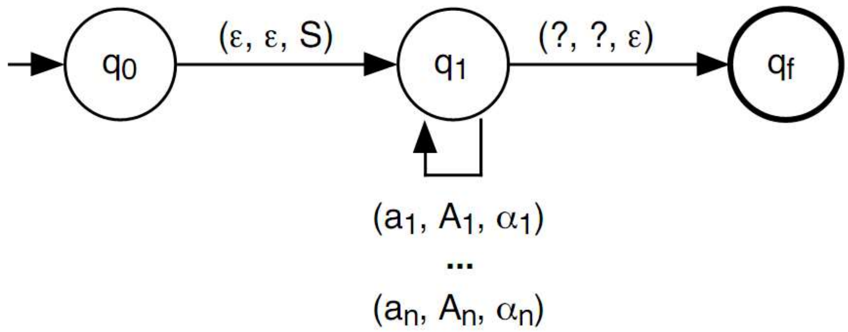
GLC → AP

- Suponha que $\varepsilon \notin L$
- Construção de um AP a partir da gramática na FNG
  - produções da forma **$A \rightarrow a\alpha$**, α palavra de variáveis
- AP resultante simula a derivação mais à esquerda
  - lê o símbolo **a** da fita
  - lê o símbolo **A** da pilha
  - empilha a palavra de variáveis **α**

GLC → AP (Algoritmo)

- AP M a partir da gramática G = (V, T, P, S)
  - GFNG = (V_{FNG}, T_{FNG}, P_{FNG}, S), é G na Forma Normal de Greibach

- M = (T_{FNG}, { q0, q1, qf }, δ, q0, { qf }, V_{FNG})
  - δ(q0, ε, ε) = { (q1, S) }
  - δ(q1, a, A) = { (q1, α) | A → aα ∈ P_{FNG} }
  - δ(q1, ?, ?) = { (qf, ε) }

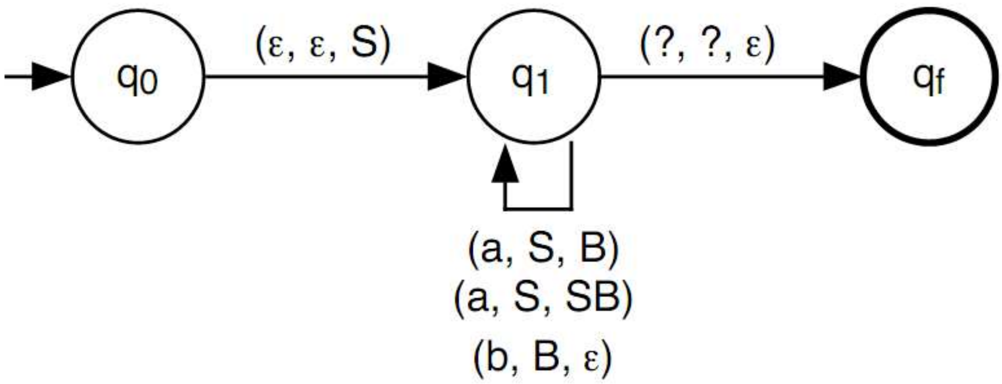


[Aula 15] LLC – Autômato com Pilha SIN 131 – Introdução à Teoria da Computação (PER-3)

EXEMPLO: GLC → AP: L5 = { $a^n b^n$ | n ≥ 1 }

- Gramática na Forma Normal de Greibach
  - G5 = ({ S, B }, { a, b }, P5, S)
  - P5 = { S → aB | aSB, B → b }

- Correspondente AP
  - M5 = ({ a, b }, { q0, q, qf }, δ5, q0, { qf }, { S, B })



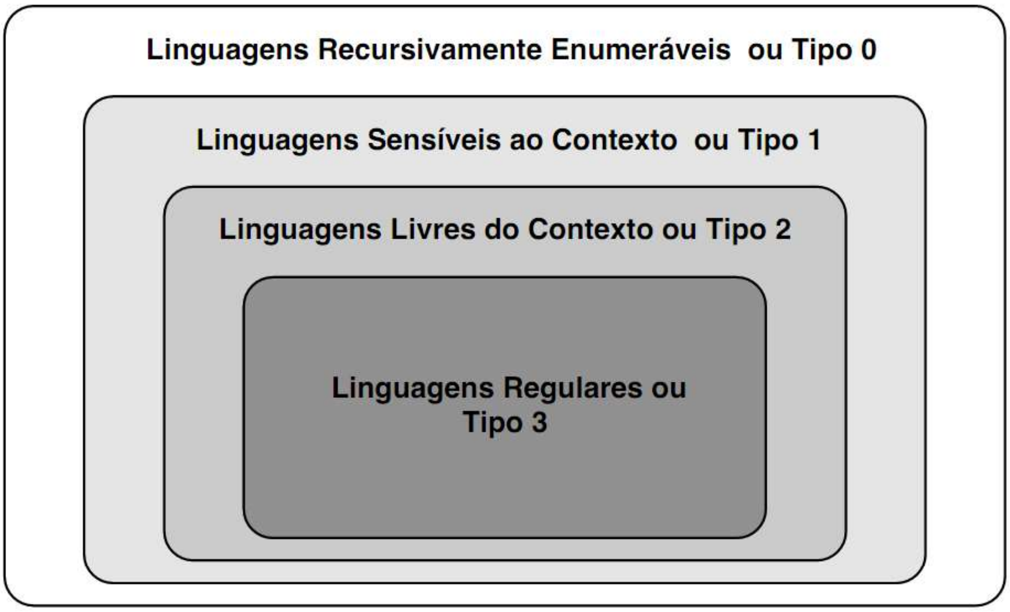
Corolários

- Se L é uma LLC, então existe um AP M:
  - M : Um AP com controle de aceitação por estados finais, com três estados, tal que ACEITA(M) = L
  - M : AP com controle de aceitação por pilha vazia, com um estado tal que ACEITA(M) = L
- Existência de um AP que Sempre PARA:
  - Se L é uma LLC, então existe um AP M, tal que
    - ACEITA(M) = L
    - REJEITA(M) = Σ^* - L
    - LOOP(M) = $\emptyset$ → Ou seja, AP sempre para LLC.

Número de Pilhas e o Poder Computacional

- Autômato com Pilha, sem usar a estrutura de pilha
  - Estados: única forma de memorizar informações passadas
    - Equivale a um Autômato Finito
  - AP, sem usar a pilha, com ou sem não-determinismo
    - Reconhecem a Classe das Linguagens Regulares
- Autômato com Pilha Determinístico
  - Aceita a Classe das Linguagens Livres do Contexto Determinísticas
    - Um importante subconjunto próprio da Classe das LLC
  - Implementação de um AP determinístico é simples e eficiente
    - facilita o desenvolvimento de **analisadores sintáticos**.
- Autômato com (uma) Pilha Não-Determinístico
  - Aceitam exatamente a Classe das LLC

Hierarquia de Chomsky



Número de Pilhas e o Poder Computacional

- Autômato com Duas Pilhas
  - Mesmo poder computacional da Máquina de Turing
    - Considerada o **dispositivo mais geral de computação**
      - **Resolve QUALQUER problema computável!!!!**
  - Se **existe** um algoritmo para resolver um problema
    - Ele pode ser expresso como um autômato com duas pilhas
  - O não determinismo não aumenta o poder computacional

- Autômato com Múltiplas Pilhas
  - Poder computacional de um autômato com mais de duas pilhas
    - Equivalente ao do autômato com duas pilhas
  - Se um problema é solucionado por um autômato com múltiplas pilhas
    - Também pode ser solucionado por um autômato com duas pilhas

[FIM]

- FIM:
  - **[AULA 15]** LLC – Autômato com Pilha
- Próxima aula:
  - **[AULA 16]** Propriedades e reconhecimento das LLC