# Teoria da Computação
# Equivalência entre GLCs e APs

Prof. Jefferson Magalhães de Morais

# Equivalência entre GLCs e APs

- Note que há equivalência da classe de linguagens **aceita** pelos APND com a classe de linguagens **gerada** pelas gramáticas LLC (tipo 2)
- A equivalência será apresentada mostrando que
  - Para qualquer GLC é possível definir um APND que reconhece exatamente a mesma linguagem gerada pela gramática (GLCs $\Rightarrow$ APs)

## GLCs $\Rightarrow$ APs

- **Teorema**: Seja $G$ uma GLC. Então é possível definir um APND $M$, com critério de aceitação baseado em pilha vazia, de modo que $V(M) = L(G)$
- **Algoritmo: GLC $\Rightarrow$ AP (v.1)**

  *Entrada: uma GLC $G = (V, \Sigma, P, S)$ na FNG e $\varepsilon \notin L(G)$*

  *Saída: um AP $M = (Q, \Sigma, \Gamma, \delta, q, S, \emptyset)$ com critério de aceitação de pilha vazia, tal que $V(M) = L(G)$*

  *Método*

  - $Q \leftarrow \{q\}$
  - $\Gamma \leftarrow N$
  - *Função de transição*
    a) $\delta \leftarrow \emptyset$
    b) $\delta(q, \sigma, A) = \{(q, \gamma) \mid A \rightarrow \sigma\gamma \in P\}, \forall A \in N, \sigma \in \Sigma, \gamma \in N^*$
    Note que se $\gamma = \varepsilon$, então $\{(q, \varepsilon)\} \subseteq \delta(q, \sigma, A)$

# GLCs $\Rightarrow$ APs

- O algoritmo produz um autômato com pilha que possui **um único estado**
- Este AP simula a sequência de derivações à esquerda que seria efetuada pela gramática correspondente na geração das mesma sentenças
- Considerando a leitura da pilha no sentido do topo para o fundo
  - Em qualquer instante, a pilha contém símbolos não-terminais que ainda não foram substituídos na correspondente forma sentencial
- As cadeias formadas por símbolos terminais que compõem o prefixo das formas sentenciais obtidas a partir de derivações mais à esquerda, em vez de serem mantidas na pilha, correspondem sempre à sequências de símbolos que formam a porção já lida da fita de entrada

## Exemplo 1

- Considere a GLC na FNG

$$
\begin{aligned}
\{F &\to (EX \mid a,\\
T &\to (EX \mid a \mid (EXY \mid aY,\\
E &\to (EX \mid a \mid (EXY \mid aY \mid\\
&\quad (EXZ \mid aZ \mid (EXYZ \mid aYZ,\\
Y &\to *F \mid *FY,\\
Z &\to +T \mid +TZ,\\
X &\to )\}
\end{aligned}
$$

- Aplicando o algoritmo para converter GLC em AP, $\delta =$

$$
\begin{aligned}
\{&(q, (, F) \to \{(q, EX)\},\\
&(q, a, F) \to \{(q, \varepsilon)\},\\
&(q, (, T) \to \{(q, EX), (q, EXY)\},\\
&(q, a, T) \to \{(q, \varepsilon), (q, Y)\},\\
&(q, (, E) \to \{(q, EX), (q, EXY),\\
&\qquad (q, EXZ), (q, EXYZ)\},\\
&(q, a, E) \to \{(q, \varepsilon), (q, Y), (q, Z), (q, YZ)\}\\
&(q, *, Y) \to \{(q, F), (q, FY)\},\\
&(q, +, Z) \to \{(q, T), (q, TZ)\},\\
&(q, ), X) \to \{(q, \varepsilon)\}\}
\end{aligned}
$$

# Observação

- Para aplicar o algoritmo apresentado, a GLC deve estar na FNG com produções obedecendo exclusivamente ao formato

$$A \rightarrow \sigma\gamma, \text{ com } \sigma \in \Sigma$$

- Portanto, para que o algoritmo apresentado se aplique também a LLC que incluem a cadeia vazia, deve-se introduzir uma pequena modificação
  - Introduzir um novo estado $q_0$ em $M$
  - Renomear o estado $q$ para $q_1$
  - Adicionar duas novas transições

$$\delta(q_0, \varepsilon, S) \rightarrow \{(q_0, \varepsilon), (q_1, S)\}$$

- A transição $\delta(q_0, \varepsilon, S) = \{(q_0, \varepsilon)\}$ permite reconhecer a cadeia vazia
- A transição $\delta(q_0, \varepsilon, S) = \{(q_1, S)\}$ preserva o conteúdo da pilha, conduzindo o autômato ao estado convencional para o reconhecimento de $L - \{\varepsilon\}$

# Exemplo 2 (parte 1)

- Considere a GLC definida pelo seguinte conjunto de produções

$$\{S \to aSBS \mid aB$$
$$B \to b\}$$

- A $L(G)$ definida por essa gramática é aceita por um autômato $M$ que tenha

$$\delta(q, a, S) = \{(q, SBS), (q, B)\}$$
$$\delta(q, b, B) = \{(q, \varepsilon)\}$$

- A menor e a segunda menor sentenças da LLC são, respectivamente
  - $S \Rightarrow aB \Rightarrow ab$
  - $S \Rightarrow aSBS \Rightarrow aaBBS \Rightarrow^* aabbS \Rightarrow^* aabbab$
- Logo, $L(G) = \{ab, aabbab, \ldots\}$

# Exemplo 2 (parte 2)

- Considerando a linguagem $L' = L \cup \{\varepsilon\}$, pode observar que a mera inclusão de $S \rightarrow \varepsilon$ não produziria o efeito desejado (e.g., $L'$ incluiria a sentença $aabb$ que, originalmente, não pertence a $L$). Portanto, a transição

$$\delta(q, \varepsilon, S) = \{(q, \varepsilon)\}$$

  não produziria o efeito desejado
- Por isso, é preciso acrescentar o estado adicional, tornando $\delta$ de $M$ como

$$\begin{aligned}\delta(q_0, \varepsilon, S) &= \{(q_0, \varepsilon), (q_1, S)\}\\ \delta(q_1, a, S) &= \{(q_1, SBS), (q_1, B)\}\\ \delta(q_1, b, B) &= \{q_1, \varepsilon\}\end{aligned}$$

- O correspondente conjunto de produções de $G'$, tal que $L' = L'(G')$ é

$$\begin{aligned}Z &\rightarrow S \mid \varepsilon\\ S &\rightarrow aSBS \mid aB\\ B &\rightarrow b\end{aligned}$$