- Uma linguagem l.c. L é não-ambígua se existe uma glc não-ambígua G tq $L = L(G)$.
- Uma linguagem l.c. L é inerentemente ambígua se toda glc que gera L é ambígua.
  Exemplo: $L = \{a^i b^j c^k : i, j, k \geq 0 \text{ e } (i = j \text{ ou } j = k)\}$

Obs: As palavras $a^n b^n c^n$, para $n \geq 0$, são deriváveis de duas maneiras diferentes.

Exercício: Escreva uma glc que gera L.

Teorema: Não existe um algoritmo para decidir se uma linguagem livre de contexto é inerentemente ambígua ou não.

Teorema 4: A classe das ling. l.c. é fechada para as operações de união, concatenação e estrela.

Prova: Sejam L_1 e L_2 ling. l.c. sobre Σ.
Então, existem glc $G_1 = (V_1, \Sigma, \mathcal{P}_1, \mathcal{S}_1)$, e $G_2 = (V_2, \Sigma, \mathcal{P}_2, \mathcal{S}_2)$, com $V_1 \cap V_2 = \Sigma$ tq $L(G_1) = L_1$ e $L(G_2) = L_2$.

- União:
  Considere a glc $G = (V, \Sigma, \mathcal{P}, \mathcal{S})$, onde
  $V = V_1 \cup V_2 \cup \{\mathcal{S}\}$ (onde $\mathcal{S} \notin V_1 \cup V_2$)
  e $\mathcal{P} = \{\mathcal{S} \to \mathcal{S}_1 | \mathcal{S}_2\} \cup \mathcal{P}_1 \cup \mathcal{P}_2$.
  Prove que $L(G) = L(G_1) \cup L(G_2)$.
- Concatenação:
  Considere a glc $G = (V, \Sigma, \mathcal{P}, \mathcal{S})$, onde
  $V = V_1 \cup V_2 \cup \{\mathcal{S}\}$ (onde $\mathcal{S} \notin V_1 \cup V_2$)
  e $\mathcal{P} = \{\mathcal{S} \to \mathcal{S}_1 \mathcal{S}_2\} \cup \mathcal{P}_1 \cup \mathcal{P}_2$.
  Prove que $L(G) = L(G_1)L(G_2)$.
- Estrela:
  Considere a glc $G = (V, \Sigma, \mathcal{P}, \mathcal{S})$, onde
  $V = V_1 \cup \{\mathcal{S}\}$ (onde $\mathcal{S} \notin V_1$)
  e $\mathcal{P} = \{\mathcal{S} \to \lambda | \mathcal{S}_1 \mathcal{S}\} \cup \mathcal{P}_1$.
  Prove que $L(G) = (L(G_1))^*$.

1 Autômatos com pilha

- Um autômato com pilha não-determinístico é um sistema $\mathcal{A} = (\mathcal{Q}, \Sigma, \Gamma, \delta, s, F)$:
  - $\mathcal{Q}$ é o conjunto finito de estados;
  - Σ é o alfabeto dos símbolos de entrada;
  - Γ é o alfabeto dos símbolos da pilha;
  - $s \in \mathcal{Q}$ é o estado inicial;
  - $F \subseteq \mathcal{Q}$ é o conjunto de estados finais;

- $\delta : \mathcal{Q} \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \longrightarrow 2^{\mathcal{Q} \times (\Gamma \cup \{\lambda\})}$ é a função de transição.

- Uma computação num a.p.n-det. começa no estado inicial, com a entrada na fita e a pilha vazia.

- Seja $\mathcal{A} = (\mathcal{Q}, \Sigma, \Gamma, \delta, s, F)$ um a.p.n-det.
  Se $(q, \mathcal{B}) \in \delta(p, a, \mathcal{A})$ (q = próximo estado, $\mathcal{B}$ = símbolo a ser empilhado ou λ, p = estado atual, a = simbolo da entrada ou λ, A = simbolo no topo da pilha ou λ).

- Uma configuração de $\mathcal{A}$ é um elemento de $\mathcal{Q} \times \Sigma^* \times \Gamma^*$ descrevendo:
  - O estado inicial;
  - A parte não lida da entrada;
  - O conteúdo atual da pilha (topo..base).

- A configuração inicial para uma entrada x deve ser (s, x, λ).

$\vdash_{\mathcal{A}}$ relaciona duas configurações consecutivas de $\mathcal{A}$.
Ex: se $(q, B) \in \delta(p, a, A)$, com $a \in (\Sigma \cup \{\lambda\})$ e $A, B \in (\Gamma \cup \{\lambda\})$, então para qualquer $y \in \Sigma^*$ e $\gamma \in \Gamma^*$, $(p, ay, A\gamma) \vdash_{\mathcal{A}} (q, y, B\gamma)$.

- $\vdash_{\mathcal{A}}^*$ denota o fecho reflexivo e transitivo de $\vdash_{\mathcal{A}}$.

- Uma palavra $x \in \Sigma^*$ é aceita por $\mathcal{A}$ se existe uma computação tq
  $(s, x, \lambda) \vdash_{\mathcal{A}}^* (q, \lambda, \lambda)$ para algum $q \in F$

- A linguagem reconhecida por $\mathcal{A}$ por estado final e pilha vazia é:
  $L(\mathcal{A}) = \{x \in \Sigma^* : x \text{ é aceita por } \mathcal{A}\}$.

Exemplo 1: $L = \{a^n b^n : n \geq 0\}$