# 11) PROPRIEDADES DAS LLCs

## Propriedades de fechamento das LLCs

→ As linguagens livres de contexto são fechadas sob:

- União
- Concatenação
- Estrela
- Reverso
- Interseção com LRs

→ Elas não são fechadas sob:

- Interseção
- Complemento

Diferença:
$A \setminus B = \{x : x \in A \text{ e } x \notin B\}$
$= \bar{B} \cap A$

TEOREMA: A união de duas LLCs é uma LLC.

DEMONSTRAÇÃO: Sejam $A_1$ e $A_2$ duas LLCs.
Seja $G_i = (V_i, \Sigma_i, R_i, S_i)$ uma GLC que gera $A_i$, para $i = 1, 2$.
Renomeie $V_2$ para que $V_1 \cap V_2 = \emptyset$.
Construa $G = (V, \Sigma, R, S)$ em que

- $V = V_1 \cup V_2 \cup \{S\}$
- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $R = R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}$.

Claramente, $L(G) = L(G_1) \cup L(G_2) = A_1 \cup A_2$. CQD

TEOREMA: A concatenação de duas LLCs é uma LLC.

DEMONSTRAÇÃO: Sejam $A_1$ e $A_2$ duas LLCs.
Seja $G_i = (V_i, \Sigma_i, R_i, S_i)$ uma GLC que gera $A_i$, para $i = 1, 2$.
Renomeie $V_2$ para que $V_1 \cap V_2 = \emptyset$.
Construa $G = (V, \Sigma, R, S)$ em que

- $V = V_1 \cup V_2 \cup \{S\}$
- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $R = R_1 \cup R_2 \cup \{S \to S_1 S_2\}$

Claramente, $L(G) = L(G_1)L(G_2) = A_1 A_2$. CQD

**TEOREMA:** A estrela de uma LLC é uma LLC.

**DEMONSTRAÇÃO:** Seja $A$ uma LLC.

Seja $G = (V, \Sigma, R, S)$ uma GLC que gera $A$.

Construa $G' = (V', \Sigma, R', S')$ fazendo:

- $V' = V \cup \{S'\}$
- $R' = R \cup \{S' \to SS', S' \to \varepsilon\}$

Claramente, $L(G') = L(G)^* = A^*$. CQD

**TEOREMA:** LLCs não são fechadas sob interseção.

**DEMONSTRAÇÃO:** Seja $A_1 = \{0^n 1^n 2^i : n, i \geq 0\}$ e $A_2 = \{0^n 1^j 2^n : n, j \geq 0\}$

Note que ambas são LLCs. (por quê?)

Note ainda que $A_1 \cap A_2 = \{0^n 1^n 2^n : n \geq 0\} = C$.

mas $C$ não é LLC. (spoiler) CQD

"A interseção de duas LLCs nem sempre é uma LLC."

$$A_1 = 0^* 1^*$$
$$A_2 = \{ w \in \{0,1\}^* : |w|_0 = |w|_1 \}$$

$$A_1 = \{ 0^i 1^j : i \geq j \}$$
$$A_2 = \{ 0^i 1^j : i \leq j \}$$

$$A_1 \cap A_2 = \{ 0^n 1^n : n \geq 0 \}$$

**TEOREMA:** LLCs não são fechadas sob complemento.

**DEMONSTRAÇÃO:** Suponha que elas sejam fechadas sob complemento.

Sejam $A$ e $B$ duas LLCs quaisquer.

Então $C = \bar{A} \cup \bar{B}$ é uma LLC.

Logo, $\bar{C}$ também é LLC.

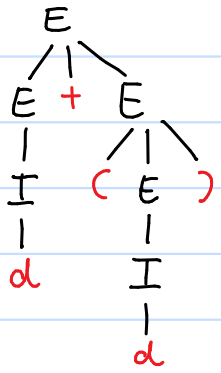
mas $\bar{C} = \overline{\bar{A} \cup \bar{B}} = A \cap B$, uma contradição. CQD

# Mais sobre árvores sintáticas

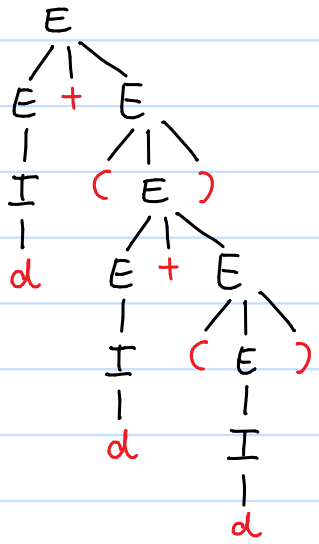
→ Considere novamente: $E \to I \mid E+E \mid E*E \mid (E) \mid d$

$I \to d \mid Id$

$E \overset{*}{\Rightarrow} d + (E) \overset{*}{\Rightarrow} \underbrace{d + (}_{\alpha} \underbrace{d}_{\beta} )$




$E \overset{*}{\Rightarrow} d + (E) \overset{*}{\Rightarrow} \underbrace{d + (}_{\alpha} \underbrace{d + (}_{\alpha} \underbrace{d}_{\beta} \underbrace{))}_{\beta\beta}$




$E \Rightarrow d + (E) \overset{*}{\Rightarrow} \underbrace{d + (}_{\alpha} \underbrace{d + (}_{\alpha} \underbrace{d + (}_{\alpha} \underbrace{d}_{\beta} \underbrace{))}_{\beta\beta}$

$E \overset{*}{\Rightarrow} \alpha^k d \beta^k$

# 13) O LEMA DO BOMBEAMENTO PARA LLCs

## O lema do bombeamento para LLCs

→ É um resultado que afirma que todas as LLCs têm uma propriedade especial.

→ Então se uma dada linguagem não tem essa propriedade, ela não é livre de contexto.

→ Infelizmente, se a linguagem tem a propriedade, não significa que ela é livre de contexto.
↳ Só se prova que uma linguagem é LLC mostrando uma GLC ou um AP para ela!

## O lema do bombeamento para LLCs

LEMA: Se $L$ é uma linguagem livre de contexto, então existe um número $p$ tal que toda cadeia $w \in L$ com $|w| \geq p$ pode ser escrita da forma $w = \alpha\beta\gamma\lambda\mu$ em que

(1) $\beta\lambda \neq \varepsilon$

(2) $|\beta\gamma\lambda| \leq p$

(3) para todo $k \geq 0$, $\alpha\beta^k\gamma\lambda^k\mu \in L$.

# 14) THE ROAD SO FAR: LINGUAGENS LIVRES DE CONTEXTO

## Linguagens Livres de Contexto

→ Por definição: aquelas geradas por GLCs

→ Equivalentemente:
aquelas reconhecidas por APs

→ Representam problemas mais complexos do que LRs:

- $\{ a^n b^m c^{n+m} : n, m \geq 0 \}$ → Decidir a soma de dois números
- $\{ w \in \{a,..,z\}^* : w = w^R \}$ → Decidir se uma palavra é palíndromo
- $\{ w \in \{(,)\}^* : w$ tem parênteses balanceados$\}$ → Decidir se uma expressão é válida
- $\{ w :$ $w$ é um programa sintaticamente correto em C$\}$ → Decidir se o código é válido.

→ É fácil / possível:

- Decidir se uma GLC gera uma dada cadeia
- Decidir se uma GLC gera alguma cadeia
- Decidir se a linguagem de uma GLC é finita