GRAMÁTICA

DEF: UMA GRAMÁTICA É UMA QUÁDRUPLA (T, V, S, R) EM QUE

- T : CONJUNTO FINITO DE SÍMBOLOS - TERMINAIS $\quad T \cap V = \emptyset$
- V : CONJUNTO FINITO DE SÍMBOLOS - VARIÁVEIS
- $S \in V$: SÍMBOLO INICIAL
- R : CONJUNTO DE REGRAS

DEF: UMA REGRA TEM O FORMATO $u \longrightarrow v$ EM QUE

- u E v SÃO PALAVRAS DO ALFABETO : $u, v \in (T \cup V)^*$
- u CONTÉM PELO MENOS UM SÍMBOLO DE V : $u \notin T^*$

DEF: DADA $G=(T,V,S,R)$ E $x,y \in (T \cup V)^*$,

DIZEMOS QUE y PODE SER DERIVADO EM UM PASSO A PARTIR DE x,

DENOTADO POR $x \Rightarrow y$, SE EXISTE REGRA $u \rightarrow v \in R$ T.Q.

$$x = x' u x'' \quad \text{E} \quad y = x' v x''$$

⤳ A DERIVAÇÃO (EM UM PASSO) SUBSTITUI UMA SUBPALAVRA.

DEF: UMA DERIVAÇÃO $x \Rightarrow^* y$ É UMA SEQUÊNCIA DE DERIVAÇÕES EM UM PASSO

↳ POSSIVELMENTE ZERO

$$x \Rightarrow y_1 \Rightarrow y_2 \Rightarrow \cdots \Rightarrow y$$

OBS: $x \Rightarrow^* x$

DEF: A linguagem gerada por G é o conjunto de palavras

$$L(G) = \{ w \in T^* : S \Rightarrow w \}$$

↳ Apenas com símbolos terminais.

DEF: Dizemos que uma gramática $G = (T, V, S, R)$ é regular se cada elemento de R é de um dos formatos a seguir

1. $X \rightarrow aY$ → No máximo uma variável

   ↳ $\in T$

2. $X \rightarrow a$

3. $X \rightarrow \varepsilon$

Sempre uma variável isolada

em que $X, Y \in V$ e $a \in T$

EX: $G = (T, V, S, R)$, $T = \{0, 1\}$, $V = \{X, Y, Z\}$, $S = X$

$$R = \begin{cases} X \to 0X & 1 \\ X \to 1Y & 2 \\ Y \to 0 & 3 \\ Y \to 1X & 4 \\ Y \to 1Z & 5 \\ Z \to \varepsilon & 6 \end{cases}$$

$$\begin{array}{ll} X \to 1Y' & 2' \\ Y' \to 0 & 3' \\ Y' \to 1X & 4' \\ Y' \to 1Z & 5' \end{array}$$

QUAL É A LINGUAGEM GERADA POR G?

$0011 \in L(G)$: $X \overset{1}{\Rightarrow} 0X \overset{1}{\Rightarrow} 00X \overset{2}{\Rightarrow} 001Y \overset{5}{\Rightarrow} 0011Z$

$\overset{6}{\Rightarrow} 0011$

GRAMÁTICAS REGULARES E AUTÔMATOS FINITOS

OBJ: MOSTRAR QUE AS LINGUAGENS GERADAS POR GRAMÁTICAS REGULARES SÃO PRECISAMENTE AS LINGUAGENS REGULARES.

1. DADA GRAM. REG. G, VAMOS CRIAR UM AFND QUE ACEITA $L(G)$

2. DADO AFD A VAMOS CRIAR GRAMÁTICA REGULAR G T.Q. $L(G) = L(A)$.

1) $G = (T, V, S, R)$ REGULAR

VAMOS CONSTRUIR UM AFND $A = (\Sigma, Q, q_0, F, \Delta)$ T.Q.

$L(A) = L(G)$

- $\Sigma = T$
- $Q = V \cup \{q_f\}$
- $q_0 = S$
- $F = \{q_f\}$
- Δ DEFINIDO POR

a) SE $X \to aY \in R$, ADICIONAMOS Y À $\Delta(X, \underset{\downarrow}{\overset{\in T}{a}})$

b) SE $X \to a \in R$, ADICIONAMOS q_f À $\Delta(X, a)$

c) SE $X \to \varepsilon \in R$, ADICIONAMOS q_f À $\Delta(X, \varepsilon)$

- Δ DEFINIDO POR

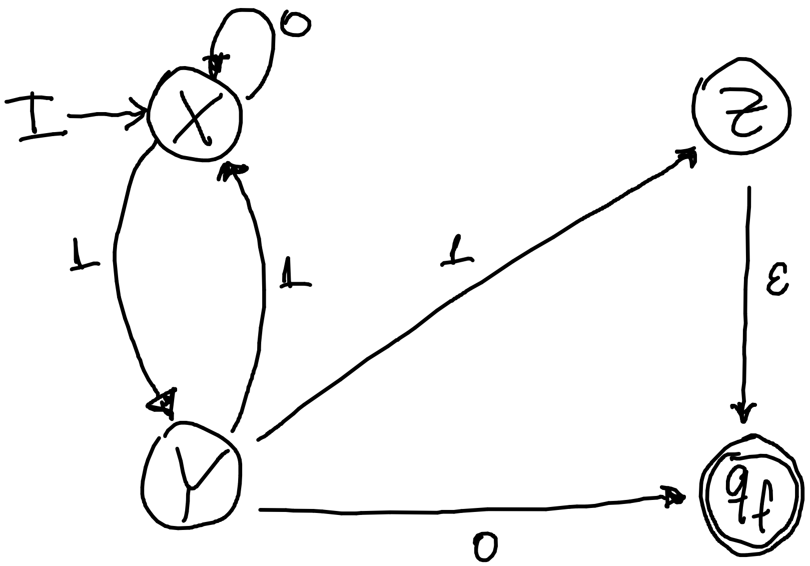
a) SE $X \to aY \in R$, ADICIONAMOS Y À $\Delta(X, a)$

b) SE $X \to a \in R$, ADICIONAMOS q_f À $\Delta(X, a)$

c) SE $X \to \varepsilon \in R$, ADICIONAMOS q_f À $\Delta(X, \varepsilon)$

Ex: $R = \{X \xrightarrow{a} 0X, X \xrightarrow{a} 1Y, Y \xrightarrow{b} 0, Y \xrightarrow{a} 1X, Y \xrightarrow{a} 1Z, Z \xrightarrow{c} \varepsilon\}$

$S = X$



0011

$X \Rightarrow 0X$
$\Rightarrow 00X$
$\Rightarrow 001Y$
$\Rightarrow 0011Z$
$\Rightarrow 0011$

$(X, 0011)$
$\vdash (X, 011)$
$\vdash (X, 11)$
$\vdash (Y, 1)$
$\vdash (Z, \varepsilon)$
$\vdash (q_f, \varepsilon)$

2) DADO AFD $A = (\Sigma, Q, q_0, F, \delta)$

VAMOS CONSTRUIR GRAM. REG. $G = (T, V, S, R)$ T.Q. $L(G) = L(A)$

- $T = \Sigma$
- $V = Q$
- $S = q_0$
- R :

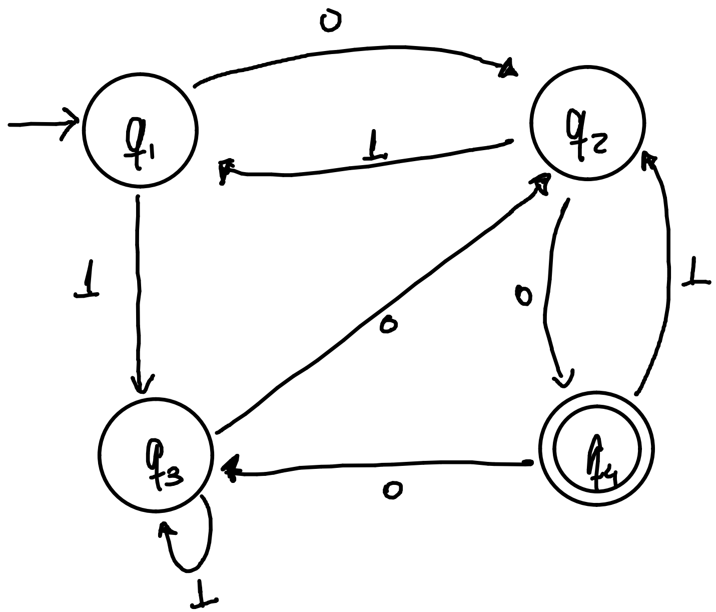
a) SE $q' = \delta(q, a)$, ADICIONAMOS $q \rightarrow a q'$ A R

b) SE $q \in F$, ADICIONAMOS $q \rightarrow \varepsilon$ A R

a) SE $q' = \delta(q, a)$, ADICIONAMOS $q \rightarrow a q'$ A R

b) SE $q \in F$, ADICIONAMOS $q \rightarrow \varepsilon$ A R

EX :



$T = \Sigma = \{0, 1\}$

$V = Q = \{q_1, q_2, q_3, q_4\}$

$S = q_0 = q_1$

$$
R = \begin{array}{c|ll}
 & a & b \\
1 & q_1 \rightarrow 0q_2, & q_1 \rightarrow 1q_3 \\
2 & q_2 \rightarrow 0q_4, & q_2 \rightarrow 1q_1 \\
3 & q_3 \rightarrow 0q_2, & q_3 \rightarrow 1q_3 \\
4 & q_4 \rightarrow 0q_3, & q_4 \rightarrow 1q_2 \\
 & & q_4 \xrightarrow{c} \varepsilon
\end{array}
$$

01100 :

$(q_1, 01100)$	q_1	
$\vdash (q_2, 1100)$	$\Rightarrow 0q_2$	1a
$\vdash (q_1, 100)$	$\Rightarrow 01q_1$	2b
$\vdash (q_3, 00)$	$\Rightarrow 011q_3$	1b
$\vdash (q_2, 0)$	$\Rightarrow 0110q_2$	3a
$\vdash (q_4, \varepsilon)$	$\Rightarrow 01100q_4$	2a

$\overset{c}{\Rightarrow} 01100$

TEOREMA: AS SEGUINTES AFIRMAÇÕES SÃO EQUIVALENTES

1) L É UMA LINGUAGEM REGULAR
2) L É ACEITO POR UM AFD
3) L É ACEITA POR UM AFND
4) L PODE SER GERADA POR UMA EXPRESSÃO REGULAR
5) L PODE SER GERADA POR UMA GRAMÁTICA REGULAR

GRAMÁTICA LIVRE DE CONTEXTO

⤳ REGRAS MAIS FLEXÍVEIS QUE A GRAMÁTICA REGULAR

(GLC)

<u>DEF</u>: DIZEMOS QUE UMA GRAMÁTICA $G = (T, V, S, R)$ É LIVRE DE CONTEXTO SE TODAS AS SUAS REGRAS SÃO DO TIPO

$$X \longrightarrow w$$

LEMBRE-SE QUE GRAM. REG:

$X \rightarrow aY$

$X \rightarrow a$

$X \rightarrow \varepsilon$

COM $X \in V$ E $w \in (T \cup V)^*$

OBS: TODA GRAMÁTICA REGULAR É LIVRE DE CONTEXTO

NEM TODA GRAMÁTICA LIVRE DE CONTEXTO É REGULAR

EX: $R = \{ S \rightarrow 0S, S \rightarrow 1, X \rightarrow XX \}$

EX: G_1 com $T = \{0,1\}$, $V = \{S\}$ e $R = \{S \to 0S1, S \to \varepsilon\}$

$$L(G_1) = \{0^n 1^n : n \in \mathbb{N}\}$$

EX: G_2 com $T = \{0,1\}$, $V = \{S\}$ e $R = \{S \to X1X, X \to 0X, X \to \varepsilon\}$

$$L(G_2) = \{0^n 1 0^m : n, m \in \mathbb{N}\} = 0^* 1 0^*$$

(LLC)

DEF: Uma linguagem L é livre de contexto se existe GLC G t.q. $L(G) = L$.

EX: $L(G_1) = \{0^n 1^n : n \in \mathbb{N}\}$ é uma linguagem livre de contexto que não é regular

$0^{n_0} 1^{n_0}$ $\rightsquigarrow$ 1) $0^{n_0+m} 1^{n_0}$

$w = xyz$ 2) $\cancel{0^{n_0} 1^{n_0+m}}$

$xy^k z \in L$ 3) $\cancel{0^{n_1} 0^{n_2} 1^{n_3} 0^{n_c} 1^{n_3} 1^{n_4}}$

$\notin L(G_1)$

$0000\underbrace{1111}_{w}$

$00001100 11\ 11$

TEO: A CLASSE DAS LINGUAGENS REGULARES É UM SUBCONJUNTO PRÓPRIO DA CLASSE DAS LLCs

EX: G_2 COM $T = \{0, 1\}$, $V = S$ E $R = \{S \to X1X, X \to 0X, X \to \varepsilon\}$ (1) (2) (3)

$$L(G_2) = \{0^n 1 0^m : n, m \in \mathbb{N}\} = 0^* 1 0^*$$

$$0^2 1 \in L(G_2)$$

$$S \Rightarrow \underline{X}1X \overset{(2)}{\Rightarrow} 0\underline{X}1X \overset{(2)}{\Rightarrow} 00\underline{X}1X \overset{(3)}{\Rightarrow} 001\underline{X} \overset{(3)}{\Rightarrow} 001$$

$$\Rightarrow X1\underline{X} \overset{(3)}{\Rightarrow} \underline{X}1 \overset{(2)}{\Rightarrow} 0\underline{X}1 \overset{(2)}{\Rightarrow} 00X1 \overset{(3)}{\Rightarrow} 001$$

↳ INDICA EM QUAL VARIÁVEL APLICAMOS A REGRA

GRAMÁTICAS QUE NÃO SÃO LIVRES DE CONTEXTO

EX: $T = \{a, b, c\}$, $V = \{S, X, Y\}$,

$$R = \left\{ \begin{array}{llll} 1 & S \to abc & 5 & S \to aXbc \\ 2 & Xb \to bX & 6 & Xc \to Ybc^2 \\ 3 & bY \to Yb & 7 & aY \to a^2 \\ 4 & aY \to a^2X & & \end{array} \right\}$$

TEMOS $a^2b^2c^2 \in L(G)$

$$S \overset{5}{\Rightarrow} a\underline{Xb}c \overset{2}{\Rightarrow} ab\underline{Xc} \overset{6}{\Rightarrow} ab\underline{Yb}c^2$$

$$\overset{3}{\Rightarrow} \underline{aY}b^2c^2 \overset{7}{\Rightarrow} a^2b^2c^2$$

TEMOS QUE $a^nb^nc^n \in L(G)$ $\forall n$

CONJ. $L(G) = \{a^nb^nc^n : n \in \mathbb{N}\}$

EX: GRAMÁTICA QUE GERA UMA FÓRMULA LEGÍTIMA,

$G_{exp} = (T, V, S, R)$ COM $T = \{id, +, *, (,)\}$, $V = \{E\}$, $S = E$

REGRAS :

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$E \rightarrow id$

$3 \cdot x + 7$

$3 \cdot y + 7$

$7 \cdot y + 3$

- NÃO PRECISAMOS CONHECER AS VARIÁVEIS, APENAS SUAS POSIÇÕES
- NÃO QUEREMOS EXPRESSÕES DO TIPO $\underline{(+\underline{id})\,\underline{id}\, * \in T^*}$
- O MESMO SÍMBOLO id PARA VARIÁVEIS

REGRAS :

$E \to E + E$

$E \to E * E$

$E \to (E)$

$E \to id$

AMBÍGUA

$7 + 8 \cdot 52$

⇝ DERIVAR $id + id * id$

$E \Rightarrow E + \underline{E} \Rightarrow \underline{E} + E * E \Rightarrow id + \underline{E} * E \Rightarrow id + id * \underline{E} \Rightarrow id + id * d$

$E \Rightarrow \underline{E} * E \Rightarrow \underline{E} + E * E \Rightarrow id + \underline{E} * E \Rightarrow id + id * \underline{E} \Rightarrow id + id * d$

REGRAS :

$$E \to (E + E)$$
$$E \to (E * E)$$
$$E \to id$$

AMBÍGUA

7 + 8 · 52

⇝ DERIVAR ~~id + id * id~~ ⇝ ((id + id) * id) ou (id + (id * id))

$$E \Rightarrow E + \underline{E} \Rightarrow \underline{E} + E * E \Rightarrow id + \underline{E} * E \Rightarrow id + id * \underline{E} \Rightarrow id + id * d$$

$$E \Rightarrow \underline{E} * E \Rightarrow E + E * E \Rightarrow id + \underline{E} * E \Rightarrow id + id * \underline{E} \Rightarrow i \quad d * d$$

DEF: SEJA G UMA GLC E $w \in L(G)$.

UMA DERIVAÇÃO MAIS À ESQUERDA DE w EM G É AQUELA NA QUAL EM CADA PASSO, A VARIÁVEL À QUAL A REGRA FOI APLICADA É A VARIÁVEL MAIS À ESQUERDA DA PALAVRA.

ANALOGAMENTE, PODEMOS DEFINIR DERIVAÇÃO MAIS À DIREITA

EX: PALÍNDROMOS

DEF: UM PALÍNDROMO É UMA PALAVRA w CUJO REFLEXO w^R É IGUAL A w.

EX: 101 , 00 111001001111

OBS:

1) w É UM PALÍNDROMO SE COMEÇA E TERMINA COM O MESMO SÍMBOLO σ, E
2) $w = \sigma w' \sigma$ EM QUE w' É UM PALÍNDROMO

$G_{Pal} = (T, V, S, R)$ COM $T = \{0, 1\}$, $V = \{S\}$, E

$$R = \{ S \rightarrow 0S0,\ S \rightarrow 1S1,\ S \rightarrow \varepsilon,\ S \rightarrow 0,\ S \rightarrow 1 \}$$

COMPACTA : $S \rightarrow 0S0 \mid 1S1 \mid \varepsilon \mid 0 \mid 1$

OBS: COMO GERAR APENAS PALÍNDROMOS DE COMPRIMENTO PAR?

FECHAMENTO LLC

SUPONDO $V_1 \neq V_2$

1. <u>UNIÃO</u>: SEJAM L_1 E L_2 DUAS LLCs.

SEJA $G_i = (T_i, V_i, S_i, R_i)$ A GRAMÁTICA QUE GERA L_i, $i=1,2$.

TOME $G_U = (T_U, V_U, S_U, R_U)$ EM QUE

$T_U = T_1 \cup T_2$

$V_U = V_1 \cup V_2 \cup \{S_U\}$

$R_U = R_1 \cup R_2 \cup \{S_U \to S_1 \mid S_2\}$

SE $w_i \in L_i$. ENTÃO $S_i \overset{R_i}{\Rightarrow^*} w_i$. LOGO $S_U \Rightarrow S_i \overset{R_i}{\Rightarrow} w_i$, $i=1,2$

LOGO $L(G_U) = L_1 \cup L_2$

2.

Concatenação

$G_\bullet = (T_\bullet, V_\bullet, S_\bullet, R_\bullet)$

Supondo $V_1 \neq V_2$

$T_\bullet = T_1 \cup T_2$

$V_\bullet = V_1 \cup V_2 \cup \{S_\bullet\}$

$R_\bullet = R_1 \cup R_2 \cup \{S_\bullet \to S_1 S_2\}$

$w_1 \in L_1$

$w_2 \in L_2$

$w_1 w_2 \in L_\bullet = L_1 L_2$

3.

Estrela de Kleene

$G_* = (T_*, V_*, S_*, R_*)$

$T_* = T_1$, $V_* = V_1 \cup S_*$

$R_* = R_1 \cup \{S_* \to S_1 S_* \mid \varepsilon\}$

$w \in L_1$ $\quad w_1, w_2, \dots, w_k \in L$

$w_* \in L_{1*}$ $\quad w_1 w_2 \cdots w_k \in L_*$

$=$

$\varepsilon, w, ww, w \cdots w$

LEMBRE-SE QUE $\{a^n b^n c^n : n \in \mathbb{N}\}$ NÃO É LLC

4. INTERSEÇÃO : NÃO É VERDADE QUE SE L_1 E L_2 SÃO LLC, ENTÃO $L_1 \cap L_2$ É LLC.

EX: $L_1 = \{a^n b^n c^m : n, m \in \mathbb{N}\}$
$L_2 = \{a^m b^n c^n : n, m \in \mathbb{N}\}$ SÃO LLC

$L_1 \cap L_2 = \{a^n b^n c^n : n \in \mathbb{N}\}$ QUE NÃO É LLC

OBS: JÁ SABEMOS QUE $L_{ab} = \{a^n b^n : n \in \mathbb{N}\}$ E $L_{bc} = \{b^n c^n : n \in \mathbb{N}\}$ SÃO LLC.

LOGO $L_{ab} \cdot \{c\}_*$ E $\{a\}_* \cdot L_{bc}$ SÃO LLC.

COMPLEMENTO : TAMBÉM NÃO VALE.

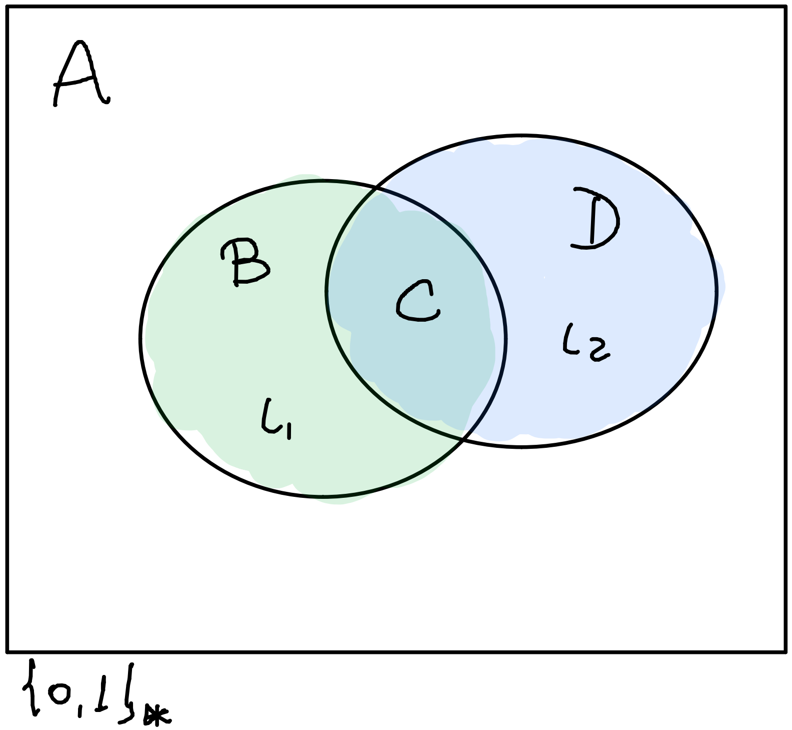
LEMBRE-SE QUE $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

SUPONHA QUE O COMPLEMENTO DE UMA LLC É LLC, ENTÃO

1) $\overline{L_1}$, $\overline{L_2}$ É LLC

2) LOGO, A UNIÃO $\overline{L_1} \cup \overline{L_2}$ É LLC

3) FINALMENTE $\overline{\overline{L_1} \cup \overline{L_2}}$ É LLC



$\overline{L_1} = A \cup D$

$\overline{L_2} = A \cup B$

$\overline{L_1} \cup \overline{L_2} = A \cup B \cup D$

$\overline{\overline{L_1} \cup \overline{L_2}} = C = L_1 \cap L_2$

<u>DIFERENÇA</u>: TAMBÉM NÃO É VERDADE.

SUPONHA QUE PARA TODAS LLCs L_3 E L_4 TEMOS QUE $L_3 \setminus L_4$ É LLC.

TOME $L_3 = \Sigma^*$, QUE É REGULAR E, PORTANTO, É LLC.

E NOTE QUE $\overline{L_4} = L_3 \setminus L_4$

Árvores de Análise Sintática

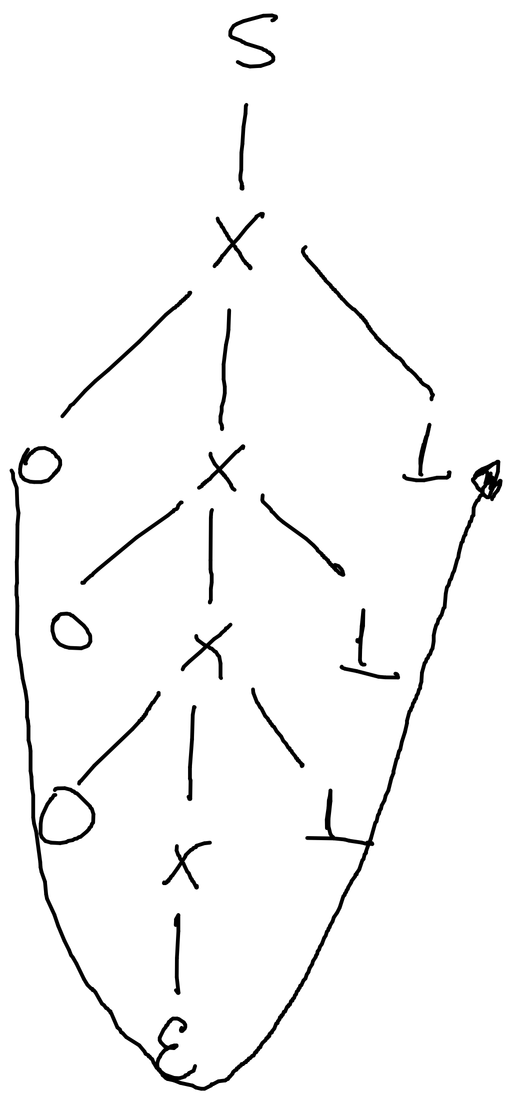
<u>Ex:</u> $S \rightarrow X$

$X \rightarrow 0X1 \mid \varepsilon$

$\omega = 000111$

$S \Rightarrow X \Rightarrow 0X1 \Rightarrow 00X11$

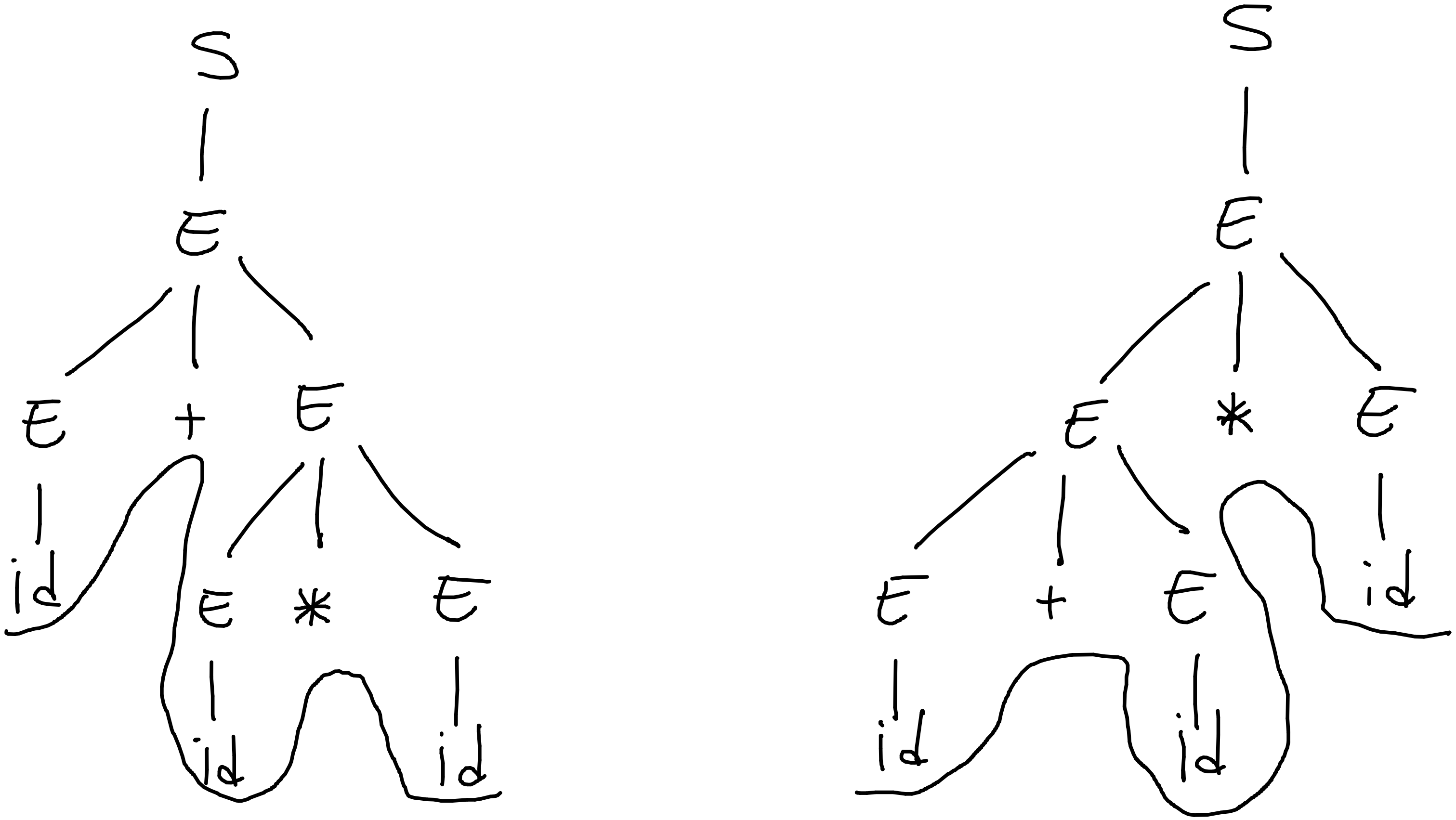
$\Rightarrow 000X111 \Rightarrow 000111$



Ex: $S \rightarrow E$

$E \rightarrow E+E \mid E*E \mid id$

$w = id + id*id$

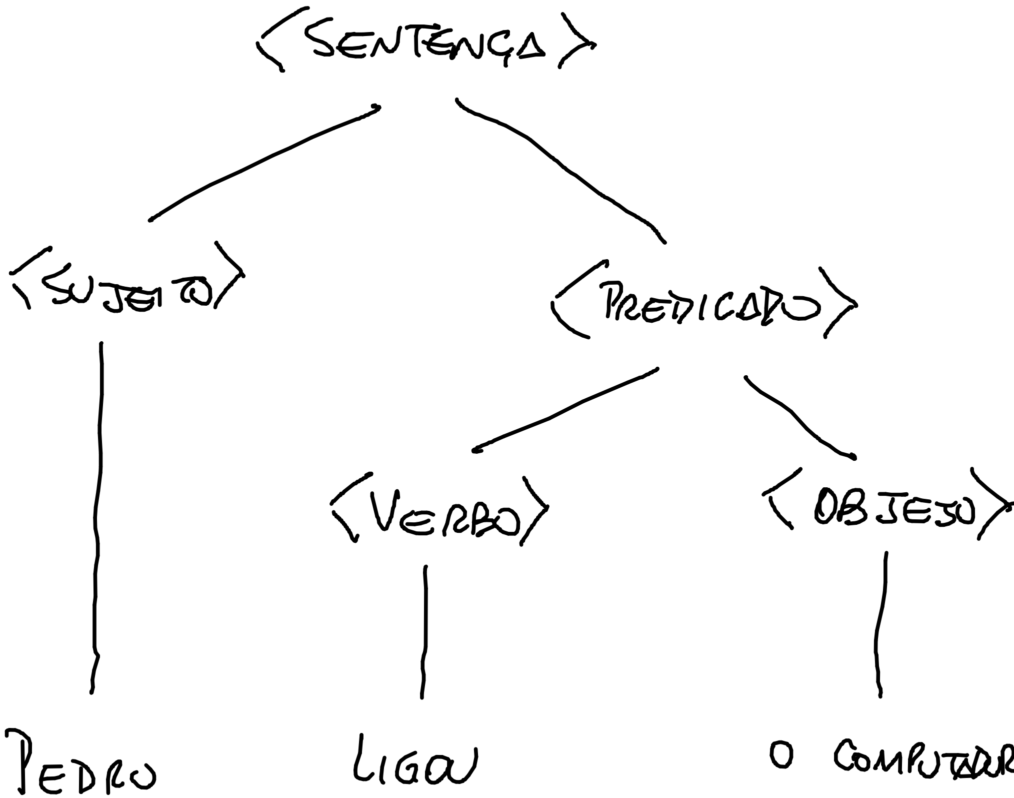


- Diagramar uma linguagem.

- Gramáticas geram frases com sintaxe correta

- Gostaríamos de analisar a estrutura de uma frase e determinar o seu significado

Ex: $\underbrace{\text{Pedro}}_{\text{sujeito}}\ \underbrace{\underbrace{\text{ligou}}_{\text{verbo}}\ \underbrace{\text{o computador}}_{\text{objeto}}}_{\text{predicado}}$



- Cada bifurcação corresponde a uma regra

<sentença> → <sujeito> <predicado>

<sujeito> → Pedro

<predicado> → <verbo> <objeto>

<verbo> → ligou

<objeto> → o computador

DEFINIÇÃO FORMAL

- ÁRVORE : GRAFOS SEM CICLOS
- AQUI : "ORIENTADOS" , SEM SETAS, "DE CIMA PARA BAIXO"

1) HÁ ÚNICO VÉRTICE CHAMADO DE RAIZ

2) OS SUCESSORES DE UM VÉRTICE SÃO TOTALMENTE ORDENADOS "DA ESQUERDA PARA DIREITA"

↳ PORQUE QUEREMOS ESCREVER DA ESQUERDA P/ DIREITA

DEF: SEJA V' UM VÉRTICE DA ÁRVORE. HÁ UM ÚNICO CAMINHO QUE LIGA ELE À RAÍZ.

SEJA V UM VÉRTICE NESTE CAMINHO

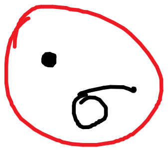
- V É ASCENDENTE DE V', E V' É DESCENTENTE DE V
- SE V E V' ESTÃO SEPARADOS POR UMA ÚNICA ARESTA ENTÃO V É PAI DE V' E V' É FILHO DE V
- FILHOS DO MESMO PAI SÃO IRMÃOS
- VÉRTICE SEM FILHOS É UMA FOLHA
- VÉRTICE QUE NÃO É FOLHA É INTERIOR

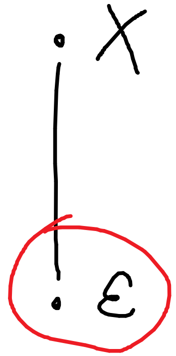
→ A ORDENAÇÃO ENTRE IRMÃOS PODE SER ESTENDIDA PARA UMA ORDENAÇÃO DAS FOLHAS.

DEFINIMOS ÁRVORES DE ANÁLISE SINTÁTICA RECURSIVAMENTE

FIXE GRAMÁTICA $G = (T, V, S, R)$

BÁSICAS: $\sigma \in T$, $X \in V$ e $X \to \varepsilon \in R$

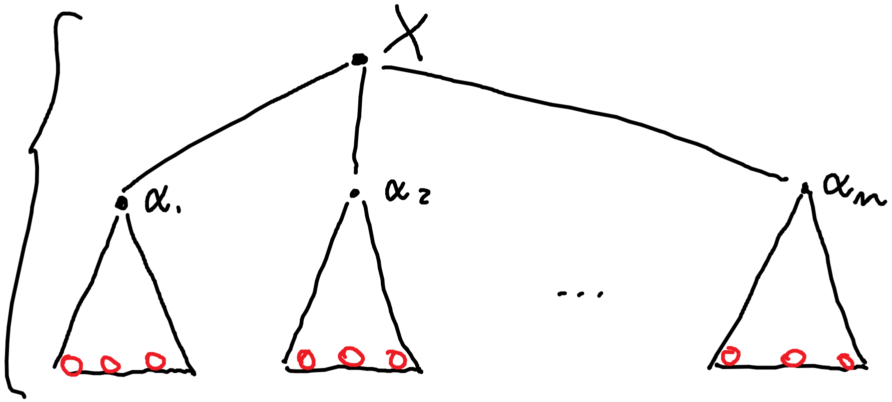




REGRAS DE COMBINAÇÃO: $T_1, \dots, T_n$ ÁRVORES COM RAÍZES $\alpha_1, \dots, \alpha_n$, RESP.

SE $X \to \alpha_1 \alpha_2 \cdots \alpha_n \in R$, ENTÃO

É UMA X-ÁRVORE

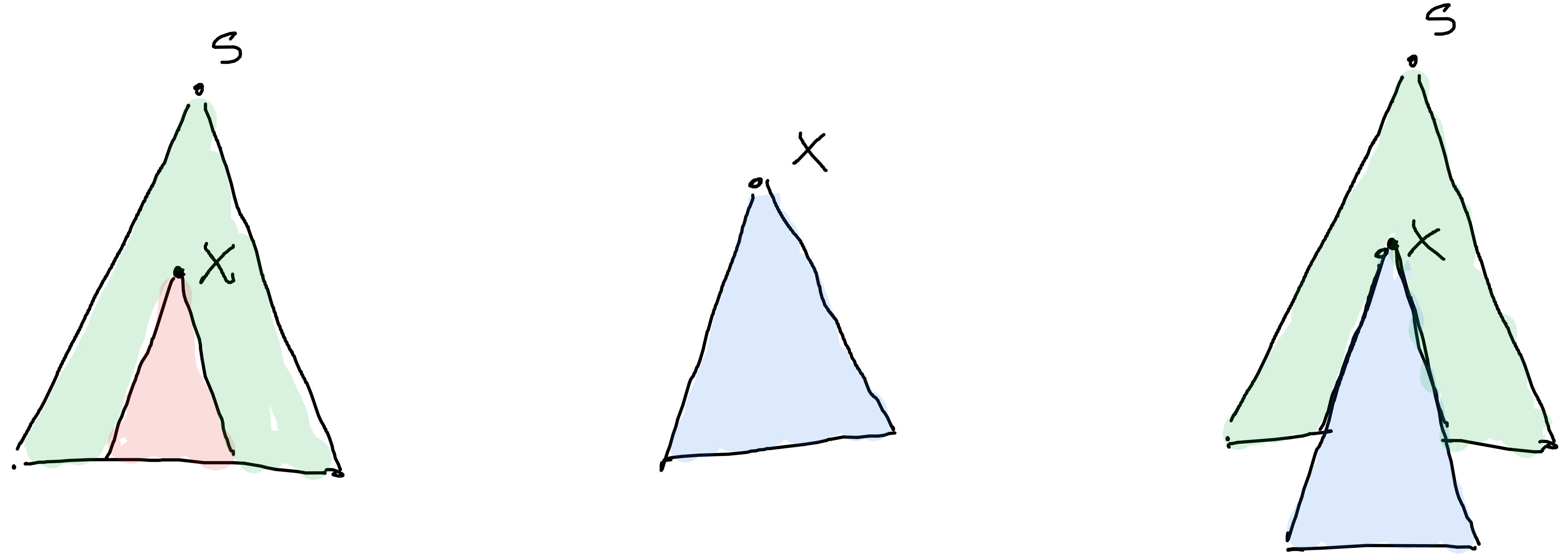


É ÁRVORE, COM REGRA ASSOCIADA

$$X \to \alpha_1 \alpha_2 \cdots \alpha_n$$

- EM PARTICULAR, PELAS REGRAS BÁSICAS, AS FOLHAS DE UMA ÁRVORE DE ANÁLISE SINTÁTICA SÃO APENAS ELEMENTOS EM $T \cup \{\varepsilon\}$

- E OS VÉRTICES INTERNOS SÃO VARIÁVEIS.



- OBS: SE UMA ÁRVORE τ POSSUI UM VÉRTICE ROTULADO POR X, PODEMOS SUBSTITUIR TODA SUBÁRVORE DE τ ENRAIZADA EM X, POR OUTRA ÁRVORE ENRAIZADA EM X.

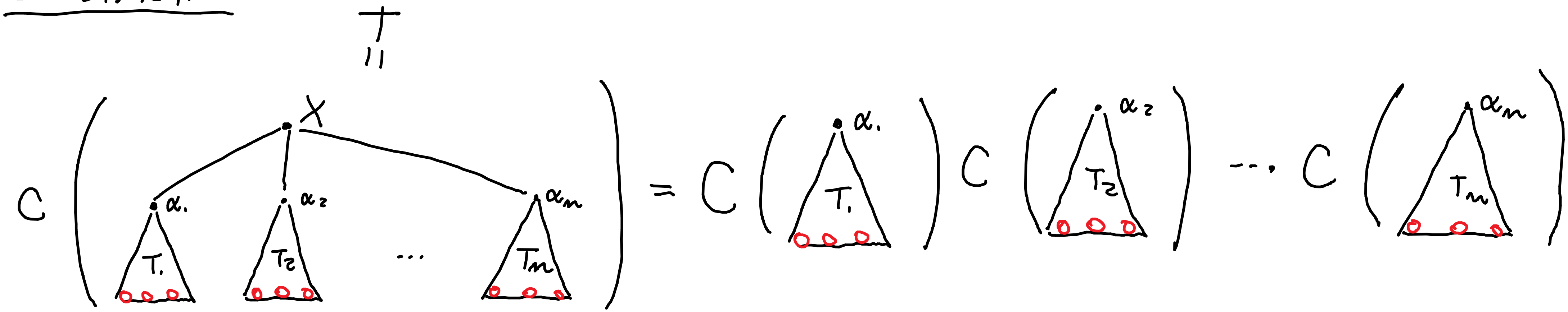
COLHEITA

COLHEITA OU RESULTADO DE UMA ÁRVORE É UMA FUNÇÃO C QUE LEVA ÁRVORES EM PALAVRAS

BÁSICAS

$$C(\bullet_{\sigma}) = \sigma \qquad\qquad C\left(\begin{array}{c}\bullet\, X\\ |\\ \bullet\, \varepsilon\end{array}\right) = \varepsilon$$

COMBINAÇÃO



$$C(T) = C(T_1)\, C(T_2) \cdots C(T_n)$$

- COMO AS FOLHAS ESTÃO ORDENADAS, NÃO HÁ AMBIGUIDADE NA COLHEITA.

DEF: UMA ÁRVORE COM RAIZ X É UMA X-ÁRVORE

DEF: SE $w \in L(G)$, ENTÃO UMA ÁRVORE DE DERIVAÇÃO P/ w É UMA S-ÁRVORE T CUJA COLHEITA É w, i.e., $c(T) = w$

DEF: SEJA $w \in (T \cup V)^*$, $w = \alpha_1 \alpha_2 \cdots \alpha_m$.
UMA w-FLORESTA É UMA SEQUÊNCIA ORDENADA $T_1, \ldots, T_m$, DE ÁRVORES, EM QUE T_i É UMA α_i-ÁRVOR

• COMO OBTER UMA DERIVAÇÃO A PARTIR DE UMA ÁRVORE?

ENTRADA : UMA X-ÁRVORE $\mathcal{T}$, COM $X \in \mathcal{V}$

SAÍDA : UMA DERIVAÇÃO MAIS À ESQUERDA DE $c(\mathcal{T})$

1) INICIALIZE $\mathcal{F} = \mathcal{T}$

2) SE $\mathcal{F}$ É UMA w-FLORESTA, IMPRIMA w.

3) SE $w \in T^*$, ENTÃO PARE

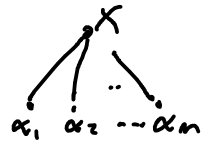
4) CASO CONTRÁRIO, SEJA v A RAIZ DA PRIMEIRA ÁRVORE DE $\mathcal{F}$ QUE É ROTULADA POR UMA VARIÁVEL

FAÇA $\mathcal{F} = \mathcal{F} \setminus v$

Proposição: Seja X uma variável de uma GLC.
Se existe derivação $X \Rightarrow^* w$, então w é a colheita de uma X-árvore.

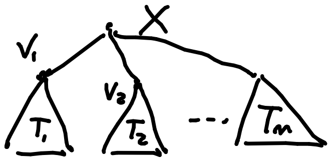
Prova: Indução no número p de passos da derivação.

Base: Um único passo. $X \Rightarrow w$. Então $X \to w \in R$.
Logo, se $w = \alpha_1 \cdots \alpha_m$, então



é uma X-árvore cuja colheita é w,

Passo indutivo: Suponha que toda p-derivação $Y \to w'$ pode ser "transformada" em uma Y-árvore cuja colheita é w'.
Seja $X \Rightarrow V_1 \cdots V_m$ o primeiro passo de $X \Rightarrow^* w$.
Isso quer dizer que $w = w_1 \cdots w_m$ e que existem derivações $V_i \Rightarrow^* w_i$ para todo i.
Mas $V_i \Rightarrow^* w_i$ é uma r-derivação com $r \leq p$. Pela HI, existe V_i-árvore T_i cuja colheita é w_i.
Logo,



é uma X-árvore cuja colheita é w. ∎

TEOREMA: SEJA G UMA GLC E $w \in L(G)$. ENTÃO

1) EXISTE ÁRVORE DE DERIVAÇÃO CUJA COLHEITA É w

2) CADA ÁRVORE DE DERIVAÇÃO CUJA COLHEITA É w CORRESPONDE A UMA ÚNICA DERIVAÇÃO MAIS À ESQUERDA DE w.

AMBIGUIDADE

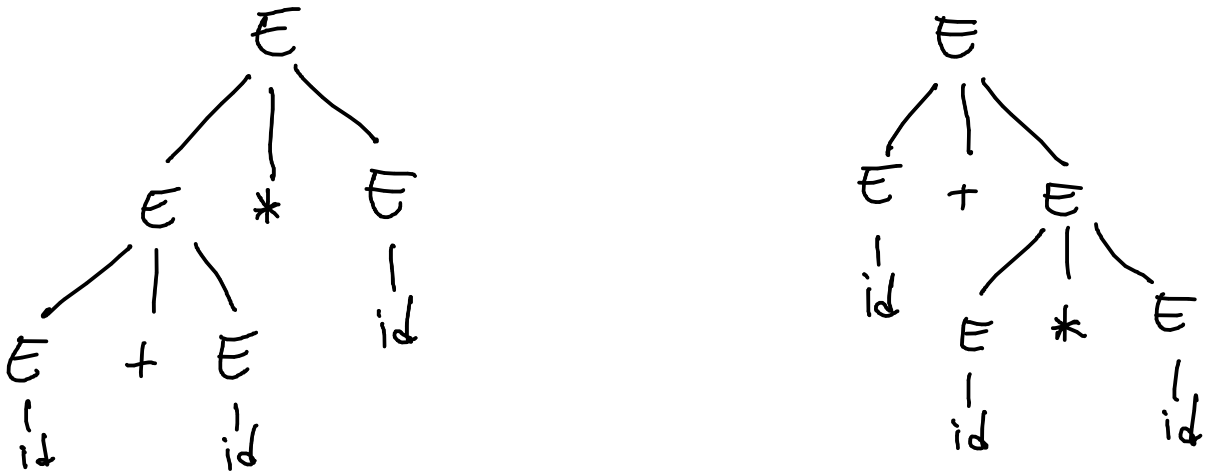
- ÁRVORES NOS AJUDAM A INTERPRETAR UMA FRASE CORRETAMENTE.
- ALGUMAS FRASES POSSUEM INTERPRETAÇÕES DISTINTAS

EX: A SEGUIR VEIO UMA MÃE COM UMA CRIANÇA EMPURRANDO UM CARRINHO.

QUEM ESTÁ EMPURRANDO O CARRINHO?

DEF: A GRAMÁTICA G É AMBÍGUA SE EXISTE UMA PALAVRA $w \in L(G)$ QUE ADMITE DUAS ÁRVORES DE DERIVAÇÃO DISTINTAS.

EX:



EX: $T = \{a, +, -\}$, $V = \{S, A\}$,

$$R = \begin{cases} S \longrightarrow A \\ A \longrightarrow A+A \mid A-A \mid a \end{cases}$$

EX: SEJAM L_1 E L_2 DUAS LINGUAGENS COM GRAMÁTICAS

$$G_1 = (T_1, V_1, S_1, R_1) \quad \text{E} \quad G_2 = (T_2, V_2, S_2, R_2)$$

SEJA $U = L_1 \cup L_2$ A LINGUAGEM COM GRAMÁTICA

$$G_U = (T_1 \cup T_2, V_1 \cup V_2, S_U, R_1 \cup R_2 \cup \{S_U \rightarrow S_1 \mid S_2\})$$

SE $L_1 \cap L_2 \neq \emptyset$, G_U É AMBÍGUA.

SE $w \in L_1 \cap L_2$, COMO $S_1 \Rightarrow^* w$ E $S_2 \Rightarrow^* w$, ENTÃO

$S_U \Rightarrow S_1 \Rightarrow^* w$ E $S_U \Rightarrow S_2 \Rightarrow^* w$ SÃO DERIVAÇÕES

QUE PRODUZEM ÁRVORES DE ANÁLISE SINTÁTICA DIFERENTES.

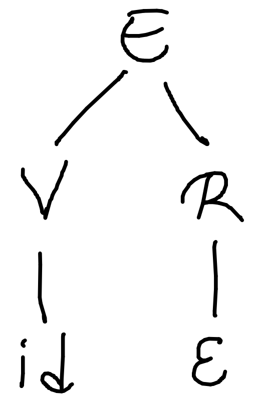
EX: A gramática G''_{EXP} com regras

$$
\begin{cases}
S \longrightarrow E \\
E \longrightarrow (E)R \mid VR \\
R \longrightarrow +E \mid *E \mid \varepsilon \\
V \longrightarrow id
\end{cases}
$$

- Podemos deduzir a derivação a partir da palavra dada.

Prova: Indução no comprimento da palavra w.

Se $|w| = 1$, então $w = id$



$$S \Rightarrow E \Rightarrow VR \Rightarrow idR \Rightarrow id$$
$$\Rightarrow V\varepsilon \Rightarrow id$$

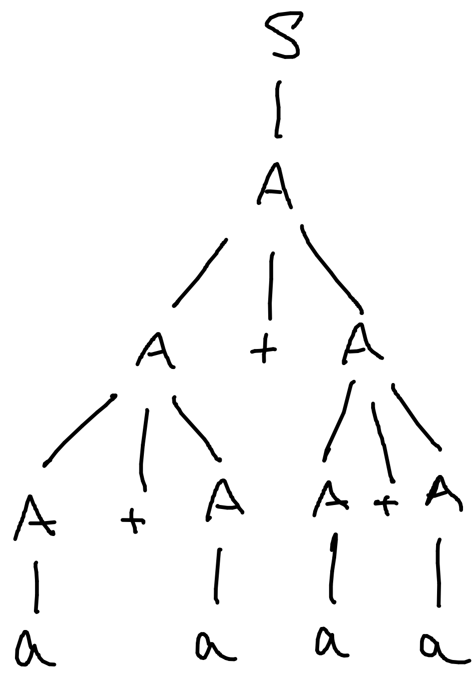
Se $|w| = n$ e $w = idw'$, então $|w'| < n$, e há uma única derivação $R \Rightarrow w'$

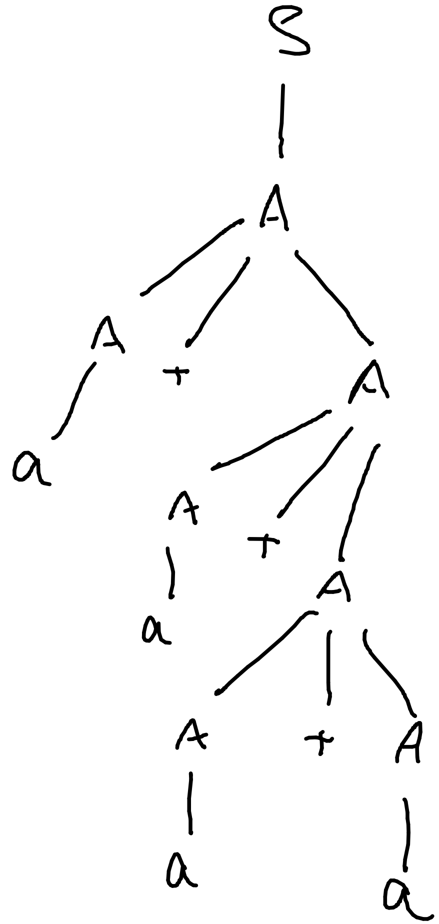
Se $|w| = n$, e $w = (w'$, então $w = (w_1)w_2$ e $|w_1|, |w_2| < n$
e temos $E \Rightarrow^* w_1$ e $R \Rightarrow^* w_2$

EX: $L = \{ a(+a)^n : n \geq 0 \} = \{ a, a+a, a+a+a, \ldots \}$

$$R_1 = \begin{cases} S \to A \\ A \to A+A \mid a \end{cases}$$

É AMBÍGUA





$$R_2 = \begin{cases} S \to A \\ A \to A+a \mid a \end{cases}$$

NÃO É AMBÍGUA.

OBS: O PROBLEMA DE DECIDIR SE UMA GRAMÁTICA É AMBÍGUA É INDECIDÍVEL.

- UMA LINGUAGEM L É INERENTEMENTE AMBÍGUA SE TODA GRAMÁTICA G T.Q. $L = L(G)$ É AMBÍGUA

EX. $\{a^n b^m c^m d^n : n, m > 0\} \cup \{a^n b^n c^m d^m : n, m > 0\}$

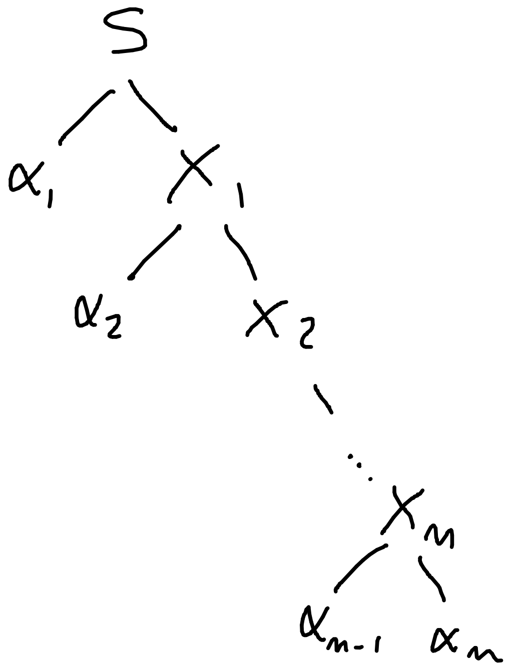
- Sempre é possível obter uma gramática não ambígua para uma linguagem regular.

Alg: Dada uma gram. reg.

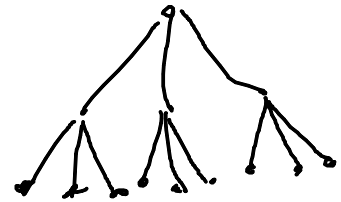
Devolve gramática G' não ambígua t.q. $L(G') = L(G)$

1) Determinar AFND M para $L(G)$
2) Determinar <u>AFD</u> M' para $L(G)$
3) Determinar gramática G' obtida de M'.

ωX



BOMBEAMENTO LLC



ÁRVORE 3-ÁRIA COMPLETA COM ALTURA 2

- COMO PROVAR QUE UMA LINGUAGEM NÃO É LLC?

EX: $L_{abc} = \{a^n b^n c^n : n \geq 0\}$ NÃO É LLC

→ VAMOS PROVAR QUE TODA LLC POSSUI UMA PROPRIEDADE TESTÁVEL

DEF: DIZEMOS QUE UMA ÁRVORE É m-ÁRIA SE CADA VÉRTICE POSSUI NO MÁXIMO m FILHOS.

→ A ALTURA DE UMA ÁRVORE É O COMPRIMENTO DO CAMINHO MAIS LONGO DE SUA RAIZ ATÉ UMA FOLHA

DEF: $f(h)$ = NÚMERO DE FOLHAS DE UMA ÁRVORE (COM ALTURA h) m-ÁRIA COMPLETA

→ TODO NÓ POSSUI m FILHOS

$f(0) = 1$, $f(1) = m$ , $f(2) = m^2$ , ..., $\boxed{f(h) = m^h}$

DEF: A AMPLITUDE $\alpha(G)$ DE UMA GLC É O COMPRIMENTO MÁXIMO DAS PALAVRAS QUE APARECEM À DIREITA DE UMA REGRA

EX: $X \to 0X1 \mid 0 \mid 1 \mid \varepsilon$ TEM AMPLITUDE 3.

OBS: SE G POSSUI AMPLITUDE α, ENTÃO TODAS AS SUAS ÁRVORES SÃO α-ÁRIAS

LEMA 1: SEJA G UMA GLC. SE X É UMA VARIÁVEL DE G, E w É A COLHEITA DE UMA X-ÁRVORE DE ALTURA h, ENTÃO

$$|w| \leq \alpha(G)^h$$

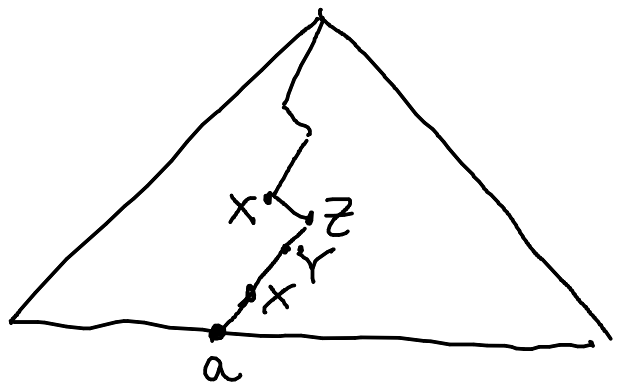
→ ISSO IMPLICA QUE SE A COLHEITA DE UMA ÁRVORE É MUITO GRANDE, ENTÃO A ALTURA DA ÁRVORE TAMBÉM É.

LEMA: SEJA G UMA GLC. SE X É UMA VARIÁVEL DE G, E w É A COLHEITA DE UMA X-ÁRVORE DE ALTURA h, ENTÃO

$$|w| \leqslant \alpha(G)^h$$

- ISSO IMPLICA QUE SE A COLHEITA DE UMA ÁRVORE É MUITO GRANDE, ENTÃO A ALTURA DA ÁRVORE TAMBÉM É.

- SE A ALTURA DA ÁRVORE FOR MAIOR QUE O NÚMERO DE VARIÁVEIS DA GRAMÁTICA, UM CAMINHO MÁXIMO (NA ÁRVORE) CONTÉM DOIS VÉRTICES ROTULADOS COM A MESMA VARIÁVEL.



OU SEJA, AS REGRAS ASSOCIADAS A ESSES VÉRTICES TÊM A MESMA VARIÁVEL À ESQUERDA.

LEMA DO BOMBEAMENTO: SEJA G UMA GRAMÁTICA LIVRE DE CONTEXTO. EXISTE INTEIRO p (QUE DEPENDE DE G) TAL QUE SE $w \in L(G)$ E $|w| \geq p$, ENTÃO EXISTE DECOMPOSIÇÃO DE w COMO $w = uvxyz$, PARA A QUAL

1) $vy \neq \varepsilon$

2) $|vxy| \leq p$

3) $uv^n xy^n z \in L(G)$, PARA TODO $n \in \mathbb{N}$.

$\alpha(G)^h \geq |w| \geq p = \alpha(G)^{k+1}$

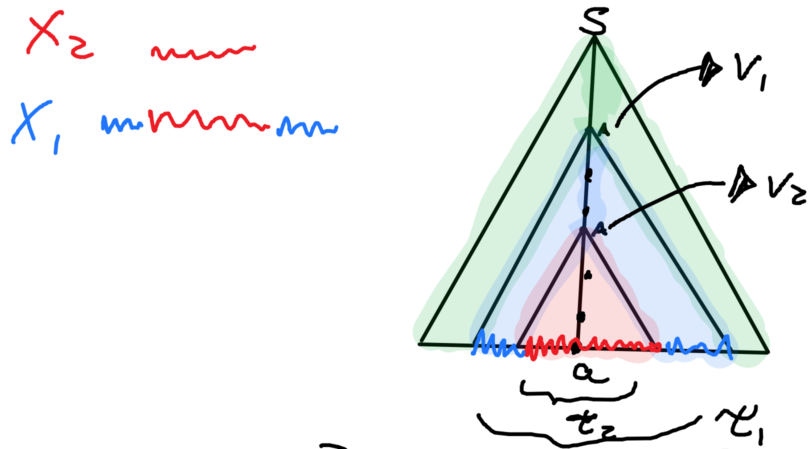
PROVA: TOME $p = \alpha(G)^{k+1}$, EM QUE k É O NÚMERO DE VARS DE G.

SEJA $w \in L(G)$ COM $|w| \geq p$, E SEJA T UMA ÁRVORE DE DERIVAÇÃO P/ w. PELO LEMA 1, A ALTURA DE T É PELO MENOS $k+1$. HÁ UM CAMINHO DA RAIZ DE T ATÉ UMA FOLHA QUE REPETE VARIÁVEL

1) $vy \neq \varepsilon$

2) $|vxy| \leq p$

3) $uv^n x y^n z \in L(G)$, PARA TODO $n \in \mathbb{N}$.



PROVA: TOME $p = \alpha(G)^{k+1}$, EM QUE k É O NÚMERO DE VARS DE G.

SEJA $w \in L(G)$ COM $|w| \geq p$, E SEJA T UMA ÁRVORE DE DERIVAÇÃO P/ w.* PELO LEMA 1, A ALTURA DE T É PELO MENOS $k+1$. HÁ UM CAMINHO DA RAIZ DE T ATÉ UMA FOLHA QUE REPETE VARIÁVEL, DIGAMOS A.

* COM O MENOR NÚMERO DE VÉRTICES POSSÍVEL

SEJAM v_1 E v_2 OS DOIS ÚLTIMOS VÉRTICES QUE REPETEM VARIÁVEL

SEJA t_i A ÁRVORE COM RAIZ v_i, COM COLHEITA X_i

$\Rightarrow$ X_2 É UMA SUBPALAVRA DE X_1

$\Rightarrow$ $X_1 = v X_2 y$

$\Rightarrow$ COMO t_2 NÃO REPETE VARIÁVEL, A ALTURA DE t_2 É NO MÁ

SEJAM v_1 E v_2 OS DOIS ÚLTIMOS VÉRTICES QUE REPETEM VARIÁVEL.

SEJA τ_i A ÁRVORE COM RAIZ v_i, COM COLHEITA X_i

$\Rightarrow$ X_2 É UMA SUBPALAVRA DE X_1 (EXCETO A)

$\Rightarrow$ $X_1 = vX_2y$

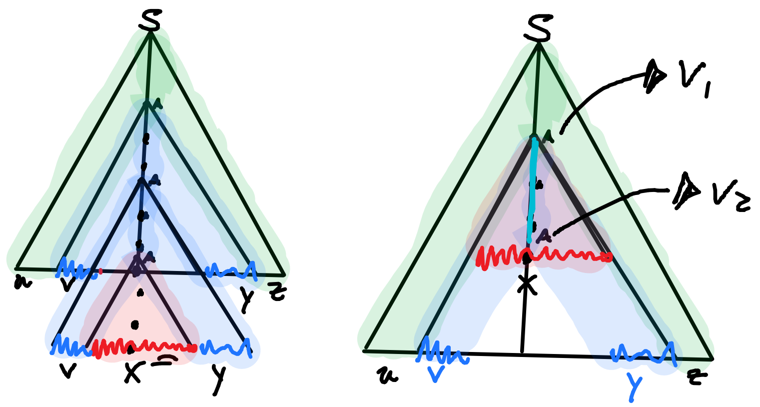
$\Rightarrow$ COMO τ_1 NÃO REPETE VARIÁVEL, A ALTURA DE τ_1 É NO MÁXIMO K+1

ISSO QUER DIZER QUE A COLHEITA DE τ_1 (X_1) TEM COMP. NO MÁXIMO $\underbrace{\alpha(G)^{K+1}}_{=p}$, OU SEJA $|X_1| = |vX_2y| \leq p$

ISSO PROVA O ITEM 2.

$\Rightarrow$ SE $|vy| = 0$, ENTÃO $vy = \varepsilon$, CONTRADIÇÃO COM A MINIMALIDADE DE T.

↳ ISSO PROVA O ITEM 1.



$\Rightarrow$ SUBSTITUINDO τ_1 NO LUGAR DE τ_2 REPETIDAMENTE OBTEMOS UMA ÁRVORE DE DERIVAÇÃO P/ uv^nxy^nz $\forall n$

↳ ISSO PROVA O ITEM 3

- Como aplicar

(*)
1) $vy \neq \varepsilon$
2) $|vxy| \leq p$
3) $uv^n xy^n z \in L(G)$, para todo $n \in \mathbb{N}$.

1) Supõe que L é LLC.
2) Toma qualquer GLC G p/ L
3) Usa o Lema de Bombeamento em G. Obtemos p
4) Escolhe palavra $w \in L$ com $|w| \geq p$
3) Mostrar que não é possível bombear.

<u>EX.</u> $L_{abc} = \{a^n b^n c^n : n \geq 0\}$

Suponha que L_{abc} é LLC, e seja G_{abc} uma GLC p/ L_{abc}.

Pelo L.B. existe p com as prop. (*).

Tome $w = a^p b^p c^p$. Claramente $|w| = 3p > p$

Pelo L.B. $w = uvxyz$ com (*)

EX: $L_{abc} = \{a^n b^n c^n : n \geq 0\}$

Suponha que L_{abc} é LLC, e seja G_{abc} uma GLC p/ L_{abc}.

Pelo L.B. existe p com as prop. (*).

Tome $w = a^p b^p c^p$. Claramente $|w| = 3p > p$

Pelo L.B. $w = uvxyz$ com (*)

$$w = \underbrace{a\cdots a}_{vxy}\overbrace{\underbrace{b\cdots b}_{vxy}\,c}^{vxy}\underbrace{\cdots c}_{vxy}$$

Análise de caso

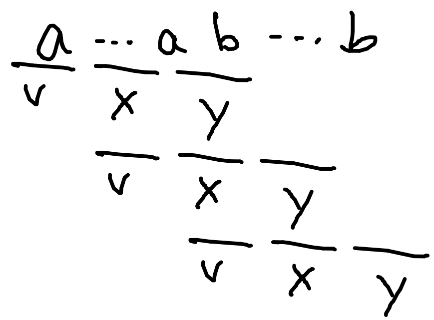
1) Se $vxy = a^i a^j a^k$, então por (3) $w' = uv^2xy^2z \in L$,

$i+k>0$ por (1)

mas $w' = a^{\boxed{p+i+k}} b^p c^p \notin L$, uma contradição.

Analogamente, verificamos o caso $vxy \subseteq b^p$ ou $vxy \subseteq c^p$

2) Se $vxy = a^i b^j$, tome $w' = uv^2xy^2z$



temos $w' = a^{p'} b^{p''} c^p$ com $p' > p$ ou $p'' > p$

Analogamente, verificamos o caso $w' = b^i c^j$

(*) (1) $vy \neq \varepsilon$
(2) $|vxy| \leq p$
(3) $uv^n x y^n z \in L(G)$, para todo $n \in \mathbb{N}$.

Logo, L_{abc} não é LLC.

EX: $L_{PRIMOS} = \{0^q : q \text{ é primos}\}$

Pelo L.B. existe p t.q. $(*)$

Seja m um primo maior que p

$w = 0^m \in L_{PRIMOS}$

Pelo L.B. $w = uvxyz$ com $|vxy| \leq p$, $vy \neq \varepsilon$

Sejam i, j t.q. $\underline{v = 0^i}$ e $\underline{y = 0^j}$

Pelo L.B. $w_n = uv^nxy^nz \in L_{PRIMOS}$ $\forall n$

Tome $n = m + 1$

$$w_n = u\underbrace{v}_{0^i}^{m+1}x\underbrace{y}_{0^j}^{m+1}z = 0^{m + \underline{m \cdot i} + \underline{mj}} = 0^{\underbrace{m(1+\overset{\neq 0}{i}+\overset{\neq 0}{j})}_{\text{não é primo}}}$$

Como $m(1+i+j)$ não é primo, $w_n \notin L_{PRIMOS}$, uma contradição

Logo, L_{PRIMOS} não é LLC. ∎

AUTÔMATOS DE PILHA

- UM AUTÔMATO DE PILHA NÃO-DETERMINÍSTICO É UM AUTÔMATO COM UMA PROPRIEDADE EXTRA: MEMÓRIA INFINITA.

- MEMÓRIA: PILHA ⇝ O ÚLTIMO ELEMENTO A SER COLOCADO É O PRÓXIMO A SER LIDO.

LISTA : $a_1 a_2 \cdots a_n$ POP(LISTA) $= a_1$

PILHA : $a_1 a_2 \cdots a_n$ POP(PILHA) $= a_n$

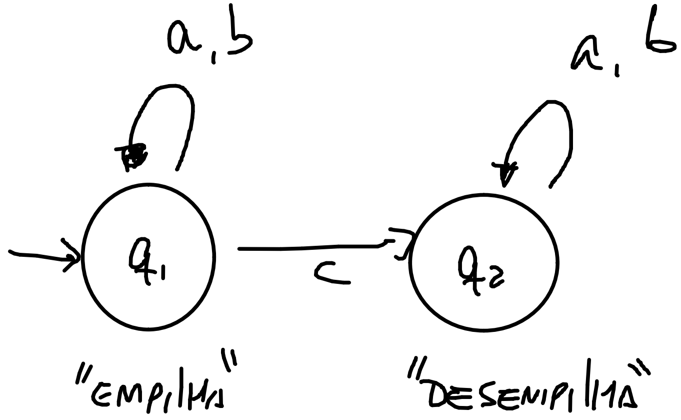
⇝ DISCOS PERFURADOS EM UMA HASTE.

⇝ LEMBRAR É COLOCAR UM DISCO NA HASTE E UM DISCO SÓ PODE SER REMOVIDO SE ESTIVER NO TOPO DA PILHA.

EX: $L_1 = \{ v c v^R : v \in \{a,b\}^* \}$

$abaacaaba \in L_1$

$abcab \notin L_1$



ETAPA 1: SE LÊ a, EMPILHA a

SE LÊ b, EMPILHA b

ETAPA 2: SE LÊ c, MUDA DE ATITUDE

ETAPA 3: COMPARA O QUE ESTÁ NO TOPO DA PILHA COM O SÍMBOLO LIDO.

SE SÃO IGUAIS, OK.

SE SÃO DIFERENTES, PARA A COMPUTAÇÃO

EX: $L_2 = \{ v v^R : v \in \{a,b\}^* \}$

→ NÃO HÁ SÍMBOLO PARA INDICAR A MUDANÇA DE ATITUDE.

→ UM AUTÔMATO P/ L_2 DEVE ADIVINHAR A HORA DE MUDAR DE ATITUDE.

DEFINIÇÃO FORMAL

DEF: UM AUTÔMATO DE PILHA NÃO-DETERMINÍSTICO (AP) A

É UMA 6-TUPLA $(\Sigma, \Gamma, Q, q_0, F, \Delta)$, EM QUE

1) Σ É O ALFABETO DE ENTRADA
2) Γ É O ALFABETO DA PILHA
3) Q É UM CONJUNTO DE ESTADOS
4) $q_0 \in Q$ É O ESTADO INICIAL
5) $F \subseteq Q$ É O CONJUNTO DE ESTADOS FINAIS
6) Δ É A FUNÇÃO DE TRANSIÇÃO

$$\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \longrightarrow P_f(Q \times \Gamma^*)$$

- NOTE QUE O AUTÔMATO PODE FAZER TRASIÇÃO SEM CONSULTAR A ENTRADA OU A PILHA.

$$\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \longrightarrow P_f(Q \times \Gamma^*)$$

COMO INTERPRETAR A FUNÇÃO DE TRANSIÇÃO?

$$(p, u) \in \Delta(q, \sigma, \gamma)$$

SIGNIFICA QUE M ESTÁ NO ESTADO q, LENDO σ NA ENTRADA E γ NA PILHA; E "VAI" PARA O ESTADO p E

TROCA γ POR u NA PILHA

- γ PODE SER TROCADA POR UMA PALAVRA INTEIRA
- γ PODE SER SIMPLESMENTE REMOVIDA $(u = \varepsilon)$

EX: $L_1 = \{ v c v^R \; : \; v \in \{a, b\}^* \}$

$Q = \{q_1, q_2\}$

$\Delta(q_1, a, \varepsilon) = \{(q_1, a)\}$
$\Delta(q_1, b, \varepsilon) = \{(q_1, b)\}$

$\Delta(q_1, c, \varepsilon) = \{(q_2, \varepsilon)\}$

$\Delta(q_2, a, a) = \{(q_2, \varepsilon)\}$
$\Delta(q_2, b, b) = \{(q_2, \varepsilon)\}$

OBS: A PALAVRA SÓ PODE SER ACEITA SE A PILHA TERMINA VAZIA.

$F = q_2$

TABELA DE TRANSIÇÃO

ESTADO	ENTRADA	TOPO DA PILHA	TRANSIÇÕES
q_1	a	ε	(q_1, a)
q_1	b	ε	(q_1, b)
q_1	c	ε	(q_2, ε)
q_2	a	a	(q_2, ε)
q_2	b	b	(q_2, ε)

EX: $L_2 = \{ vv^R : v \in \{a,b\}^* \}$

a a a a a a 0

3

a
a

	ESTADO	ENTRADA	TOPO DA PILHA	TRANSIÇÕES
	q_1	a	ε	(q_1, a)
	q_1	b	ε	(q_1, b)
3	q_1	ε	ε	(q_2, ε)
	q_2	a	a	(q_2, ε)
	q_2	b	b	(q_2, ε)

COMPUTANDO E ACEITANDO

DEF: SEJA $M = (\Sigma, \Gamma, Q, q_0, F, \Delta)$

UMA CONFIGURAÇÃO DE M É UM ELEMENTO

$$(q, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$$

OBS: POR CONVENÇÃO, O TOPO DA PILHA É O ELEMENTO MAIS À ESQUERDA.

DEF: SEJA $C = (q, \sigma w, \gamma \alpha)$ UMA CONFIGURAÇÃO DE M,

EM QUE $\sigma \in \Sigma \cup \{\varepsilon\}$ E $\gamma \in \Gamma \cup \{\varepsilon\}$.

DIZEMOS QUE $C' = (q', w, v\alpha)$ É UMA CONFIGURAÇÃO SEGUINTE A C SE

$$(q', v) \in \Delta(q, \sigma, \gamma)$$

DENOTAMOS POR $C \vdash C'$

DEF: UMA COMPUTAÇÃO DE M É UMA SEQUÊNCIA $C_0, \dots, C_K$ DE CONFIGURAÇÕES T.Q. C_{i+1} É UMA CONF. SEGUINTE DE C_i

DENOTAMOS POR $C_0 \vdash^* C_K$

EX: $L_1, M_1, \quad C_1 = (q_1, abcba^2, a)$

$$(q_1, abcba^2, a) \vdash (q_1, bcba^2, a^2) \vdash (q_1 cba^2, ba^2)$$

$$\vdash (q_2, ba^2, ba^2) \vdash (q_2, a^2, a^2)$$

$$\vdash (q_2, a, a) \vdash (q_2, \varepsilon, \varepsilon)$$

Ex: L_2, M_2 $\quad C = (q_1, a^2b, \varepsilon)$

$$(q_1, a^2b, \varepsilon) \vdash (q_1, ab, a) \vdash (q_1, b, a^2) \vdash (q_2, b, a^2)$$

$$(q_1, a^2b, \varepsilon) \vdash (q_2, a^2b, \varepsilon)$$

$$(q_1, a^2b, \varepsilon) \vdash (q_1, ab, a) \vdash (q_2, ab, a) \vdash (q_2, b, \varepsilon)$$

DEF: SEJA $M = (\Sigma, \Gamma, Q, q_0, F, \Delta)$ UM AP.

UMA PALAVRA $w \in \Sigma^*$ É ACEITA POR M SE EXISTE COMPUTAÇÃO

$$(q_0, w, \varepsilon) \vdash^* (p, \varepsilon, \varepsilon)$$

EM QUE $p \in F$.

- BASTA EXISTIR UMA COMPUTAÇÃO
- A PALAVRA w TEM QUE SER TOTALMENTE CONSUMIDA
- A PILHA TEM QUE ACABAR VAZIA
- O AUTÔMATO TEM QUE CHEGAR A UM ESTADO FINAL.

Ex: $L = \{a^i b^i : i \geq 0\}$

$Q = \{q_1, q_2\} = F$

ESTADO	ENTRADA	TOPO DA PILHA	TRANSIÇÕES
q_1	a	ε	(q_1, X)
q_1	b	X	(q_2, ε)
q_2	b	X	(q_2, ε)

EX: L é a linguagem formada pelas palavras $w \in \{a, b\}^*$ que possuem o mesmo número de a's e b's.

$\Gamma = \{a, b, \beta\}$

ESTADO	ENTRADA	TOPO DA PILHA	TRANSIÇÕES
q_0	ε	ε	(q_1, β)
q_1	a	β	$(q_1, a\beta)$
q_1	b	β	$(q_1, b\beta)$
q_1	a	a	(q_1, aa)
q_1	b	b	(q_1, bb)
q_1	a	b	(q_1, ε)
q_1	b	a	(q_1, ε)
q_1	ε	β	(q_2, ε)

GLC x AP

<u>Teorema</u>: Uma linguagem é livre de contexto se e somente se é aceita por um autômato de pilha não-determinístico.

$\Rightarrow$ Construir um autômato de pilha que aceita a linguagem gerada por uma GLC. dada

$\Leftarrow$ Construir uma GLC que gera a linguagem reconhecida por um AP. dado.

1) GLC $\rightarrow$ AP

- TRANSFORMAR VARIÁVEIS E REGRAS EM ESTADOS E TRANSIÇÕES.
- A PILHA DESEMPENHA PAPEL FUNDAMENTAL
- ESTADOS DESEMPENHAM PAPEL SECUNDÁRIO
- VAMOS CONSTRUIR UM AUTÔMATO M QUE SIMULA AS DERIVAÇÕES MAIS À ESQ. NA PILHA

SEJA $G = (T, V, S, R)$

- CLARAMENTE O ALFABETO DO AUTÔMATO TEM QUE SER T $(\Sigma = T)$
- CADA DERIVAÇÃO DE G CORRESPONDE A UMA TRANSIÇÃO DE M.
- O ALFABETO DA PILHA É $\Gamma = T \cup V$.

⇝ APENAS DOIS ESTADOS: i, f

⇝ COMO TODA DERIVAÇÃO COMEÇA COM S, O AUTÔMATO COMEÇA EMPILHANDO S:

$$\Delta(i, \varepsilon, \varepsilon) = \{(f, S)\}$$

⇝ SE TEMOS A REGRA $X \to \alpha$, CRIAMOS A TRANSIÇÃO

$$(f, \alpha) \in \Delta(f, \varepsilon, X)$$

- A PRINCÍPIO O AUTÔMATO NÃO LÊ A ENTRADA E DERIVA A PALAVRA NA PILHA ENQUANTO POSSÍVEL, E DEPOIS COMPARA COM A ENTRADA.
- O AUTÔMATO É OBRIGADO A COMPARAR QUANDO HÁ TERMINAL NO TOPO DA PILHA.

Ex: $S \longrightarrow Sc \mid aSb \mid \varepsilon$

ESTADO	ENTRADA	PILHA	TRANSIÇÕES
i	ε	ε	(f, S)
f	ε	S	$(f, Sc), (f, aSb), (f, \varepsilon)$

abc^2 É DERIVADA DA SEGUINTE FORMA

$$S \Rightarrow Sc \Rightarrow Sc^2 \Rightarrow aSbc^2 \Rightarrow abc^2$$

A COMPUTAÇÃO FICA

$$(i, abc^2, \varepsilon) \vdash (f, abc^2, S) \vdash (f, abc^2, Sc) \vdash (f, abc^2, Sc^2)$$

$$\vdash (f, abc^2, aSbc^2)$$

abc^2 É DERIVADA DA SEGUINTE FORMA

$$S \Rightarrow Sc \Rightarrow Sc^2 \Rightarrow aSbc^2 \Rightarrow abc^2$$

A COMPUTAÇÃO FICA

$$(i, abc^2, \varepsilon) \vdash (f, abc^2, S) \vdash (f, abc^2, Sc) \vdash (f, abc^2, Sc^2)$$

$$\vdash (f, abc^2, aSbc^2)$$

ENTÃO PRECISAMOS COMPARAR

ESTADO	ENTRADA	PILHA	TRANSIÇÕES
f	a	a	(f, ε)
f	b	b	(f, ε)
f	c	c	(f, ε)

AS DUAS OUTRAS COMPARAÇÕES

O QUE NOS PERMITE CONCLUIR A COMPUTAÇÃO

$$(f, abc^2, aSbc^2) \vdash (f, bc^2, Sbc^2) \vdash (f, bc^2, bc^2) \vdash^* (f, \varepsilon, \varepsilon)$$

MAIS FORMALMENTE

$\Sigma_1 = T$

$\Gamma = T \cup V$

$Q = \{i, f\}$

$q_0 = i$

$F = \{f\}$

$\Delta : \{i, f\} \times \Sigma_1 \cup \{\varepsilon\} \times \Gamma \cup \{\varepsilon\} \longrightarrow \{i, f\} \cup (\Gamma \cup \{\varepsilon\})^*$

$$\Delta(q, \sigma, \gamma) = \begin{cases} \{(f, S)\} & \text{se } q = i \text{ e } \sigma = \gamma = \varepsilon \\ \{(f, u) : X \to u \in R\} & \text{se } q = f, \sigma = \varepsilon, \gamma = X \in V \\ \{(f, \varepsilon)\} & \text{se } q = f, \sigma = \gamma \in T \end{cases}$$

PROPOSIÇÃO: SE G É GLC E M É O AUTÔMATO CONSTRUÍDO COMO MOSTRADO, ENTÃO $L(G) = L(M)$

PROVA: SEJA $w \in L(G)$. VAMOS MOSTRAR QUE $w \in L(M)$.

TEMOS QUE MOSTRAR QUE EXISTE COMPUTAÇÃO

$$(i, w, \varepsilon) \vdash^* (f, \varepsilon, \varepsilon)$$

SABEMOS QUE $S \Rightarrow^* w$ (PODEMOS ESCOLHER UMA DERIVAÇÃO MAIS A ESQ)

SUPONHA QUE APÓS ALGUMAS ETAPAS DA DERIVAÇÃO, TEMOS

$$S \Rightarrow \cdots \Rightarrow \alpha X v \Rightarrow \cdots \Rightarrow w$$

EM QUE $\alpha \in T^*$ E X É A VARIÁVEL MAIS A ESQUERDA

ISSO IMPLICA QUE $w = \alpha\beta$ PARA ALGUM $\beta \in T^*$

- VAMOS MOSTRAR QUE SEMPRE CONSEGUIMOS UMA COMPUTAÇÃO QUE DEIXA UMA VARIÁVEL NO TOPO DA PILHA

- SE CONSTRUÍMOS A COMPUTAÇÃO, TEMOS NESTA ETAPA

$$(i, w, \varepsilon) \vdash (f, w, S) \vdash^* (f, \beta, Xv)$$

- SEJA $X \to u$ A PRÓXIMA REGRA

$$\cdots \Rightarrow \alpha X v \xRightarrow{X \to u} \alpha u v \Rightarrow \cdots$$

- COMO X ESTÁ NO TOPO DA PILHA, PODEMOS FAZER

$$(f, \beta, Xv) \vdash (f, \beta, uv)$$

DIGAMOS QUE $uv = \gamma Y v'$, EM QUE $Y \in V$ É A VAR MAIS A ESQ.

LOGO $\gamma \in T^*$. CONCLUÍMOS QUE γ É UM PREFIXO DE β, I.E.

$$\beta = \gamma \beta'$$

ENTÃO

USANDO COMPARAÇÕES

$$(f, \beta, uv) = (f, \gamma\beta', \gamma Y v') \vdash^* (f, \beta', Yv')$$

POR INDUÇÃO (NO TAMANHO DA ENTRADA), TEMOS

$$(f, \beta', Yv') \vdash^* (f, \varepsilon, \varepsilon)$$

$L(G) \subseteq L(M)$

ISSO IMPLICA QUE $(i, w, \varepsilon) \vdash^* (f, \varepsilon, \varepsilon)$. LOGO $w \in L(M)$.

PROPOSIÇÃO: DADO AUTÔMATO DE PILHA M, EXISTE GLC G T.Q.

$$L(G) = L(M)$$

PROVA: MODIFICAMOS O AUTÔMATO PARA TER AS SEGUINTES PROPRIEDADES

1) POSSUI UM ÚNICO ESTADO FINAL

2) A PILHA SÓ FICA VAZIA NO FINAL

3) CADA TRASIÇÃO OU COLOCA ALGO NA PILHA, OU REMOVE, MAS NÃO AMBOS.

1 E 2 SÃO OBTIDOS AO ADICIONARMOS DOIS NOVOS ESTADOS: INICIAL E FINAL, SUAS TRANSIÇÕES E UM SÍMBOLO QUE INDICA QUE A PILHA ESTÁ VAZIA

⇝ 3 é obtido ao "dividirmos" cada transição em duas com um estado intermediário

$$p \xrightarrow{y \in \Delta^*(a,x)} q \quad \rightsquigarrow \quad p \xrightarrow{\varepsilon \in \Delta^*(a,x)} r \xrightarrow{y \in \Delta^*(\varepsilon,\varepsilon)} q$$

- Para cada par de estados p e q criamos uma variável A_{pq}

⇝ Queremos que A_{pq} gere todas as palavras que levam M de p a q iniciando e acabando com a pilha vazia.

→ A primeira operação é de incluir X

→ A última operação é de remover Y

HÁ DUAS POSSIBILIDADES

1) SE $Y = X$, QUEREMOS A REGRA $A_{pq} \longrightarrow a A_{rs} b$

EM QUE a É O SÍMBOLO LIDO NA ENTRADA AO ADICIONARMOS X NA PILHA, E b É O SÍMBOLO LIDO AO REMOVERMOS X.

2) SE $Y \neq X$. NESTE CASO, X É REMOVIDO "NO MEIO DO CAMINHO". OU SEJA, EXISTE ESTADO r NO QUAL X É REMOVIDO.

$$A_{pq} \longrightarrow A_{pr} A_{rq}$$

Formalmente: $M = (\Sigma, F, Q, q_0, \{q_f\}, \Delta)$

Definimos $G = (\Sigma, V, S, R)$ em que $V = \{A_{pq} : p, q \in Q\}$

A variável inicial é $S = A_{q_0 q_f}$

Regras:

1) $p, q, r, s \in Q$, $u \in \Gamma$, $a, b \in \Sigma \cup \{\varepsilon\}$

se $(r, u) \in \Delta(p, a, \varepsilon)$ e $(q, \varepsilon) \in \Delta(s, b, u)$

adicione

$$A_{pq} \rightarrow a A_{rs} b$$

REGRAS:

1) $p, q, r, s \in Q$, $u \in \Gamma$, $a, b \in \Sigma \cup \{\varepsilon\}$

SE $(r, u) \in \Delta(p, a, \varepsilon)$ E $(q, \varepsilon) \in \Delta(s, b, u)$

ADICIONE

$$A_{pq} \to a A_{rs} b$$

2) $p, q, r \in Q$, ADICIONE $A_{pq} \to A_{pr} A_{rq}$

3) PARA CADA $p \in Q$, ADICIONE $A_{pp} \to \varepsilon$

AF: SE A_{pq} GERA x, ENTÃO x LEVA p A q COM A PILHA VAZIA

PROVA: INDUÇÃO NO NÚMERO DE PASSOS DA DERIVAÇÃO

SE HÁ APENAS UM PASSO, ENTÃO $q = p$ E $A_{pq} = A_{pp}$,

E A REGRA APLICADA É $A_{pp} \to \varepsilon$ COM $x = \varepsilon$

ENTÃO PODEMOS SUPOR QUE HÁ MAIS DE UM PASSO

$$A_{pq} \Rightarrow^* x$$

SÃO DERIVAÇÕES MAIS CURTAS QUE $A_{pq} \Rightarrow^* x$

HÁ DUAS POSSIBILIDADES:

1) $A_{pq} \Rightarrow A_{pr} A_{rq}$: $x = x'y$ EM QUE $A_{pr} \Rightarrow^* x'$ E $A_{rq} \Rightarrow^* y$

2) $A_{pq} \Rightarrow a A_{rs} b$: ENTÃO G POSSUI $(r, u) \in \Delta(p, a, \varepsilon)$ E $(q, \varepsilon) \in \Delta(s, b, u)$

LOGO $x = a x' b$ E $A_{rs} \Rightarrow^* x'$ $\square$

AF: SE x LEVA M DE p A q COM PILHA VAZIA, ENTÃO A_{pq} GERA x

SABEMOS QUE EXISTE COMPUTAÇÃO $(p, x, \varepsilon) \vdash^* (q, \varepsilon, \varepsilon)$

SE A COMPUTAÇÃO TEM ZERO PASSOS, ENTÃO $q = p$ E $x = \varepsilon$,
MAS TEMOS $A_{pp} \to \varepsilon = x$

SUPONHA ENTÃO QUE A COMPUTAÇÃO TENHA $k+1$

DAS DUAS UMA: OU A PILHA SÓ ESVAZIA NO FINAL,
OU A PILHA ESVAZIA NUM ESTADO r.

1) O símbolo u que entra na pilha no começo é o mesmo que sai no final. Seja $x = ayb$ e r é o estado depois de p, s é o estado antes de q.

Então $(r, u) \in \Delta(p, a, \varepsilon)$ e $(q, \varepsilon) \in \Delta(s, b, u)$

Mas $A_{pq} \rightarrow a A_{rs} b$ está em G, e pela HI, com y leva M de r a s com a pilha vazia, temos $A_{rs} \Rightarrow^{*} y$.

Logo $A_{pq} \Rightarrow a A_{rs} b \Rightarrow^{*} ayb = x$

2) As porções da computação entre p e r (que lê y) e entre r e s (que lê z) $x = yz$ possuem no máximo K passos. Pela HI.

$$A_{pr} \Rightarrow y \text{ e } A_{rq} \Rightarrow z.$$

Logo, $A_{pq} \Rightarrow A_{pr} A_{rq} \Rightarrow^{*} y A_{rq} \Rightarrow^{*} yz = x$

→ Vimos que a interseção de duas LLCs não é necessariamente LLC.

→ Vamos ver que $L \cap R$ e $L - R$ é LLC se L é LLC e R é regular.

- A ideia é a mesma que usamos em autômatos regulares,
  - → Nosso autômato vai simular em paralelo as computações das duas linguagens
  - → Como o autômato regular não usa a pilha, precisamos de apenas uma pilha.

Temos $Q_\cap = Q \times Q'$, $q_{0\cap} = (q_0, q_0')$, $F_\cap = F \times F'$

$$((p,p'), u) \in \Delta_\cap((q,q'), \sigma, \gamma) \text{ se } (p,u) \in \Delta(q,\sigma,\gamma)$$

$$p' \in \delta'(q', \sigma)$$

• PARA MOSTRAR A DIFERENÇA, BASTA LEMBRARMOS QUE

1) $L - R = L \cap \overline{R}$; E

2) R É REGULAR SE E SOMENTE SE $\overline{R}$ É REGULAR.