



COMPILADORES

Prof. Me Wanderlan Carvalho de Albuquerque



Objetivo :

Compreender a estrutura da Análise Sintática

Compilador

Fase de Análise

- Léxico
- Sintático
- Semântico

Erros

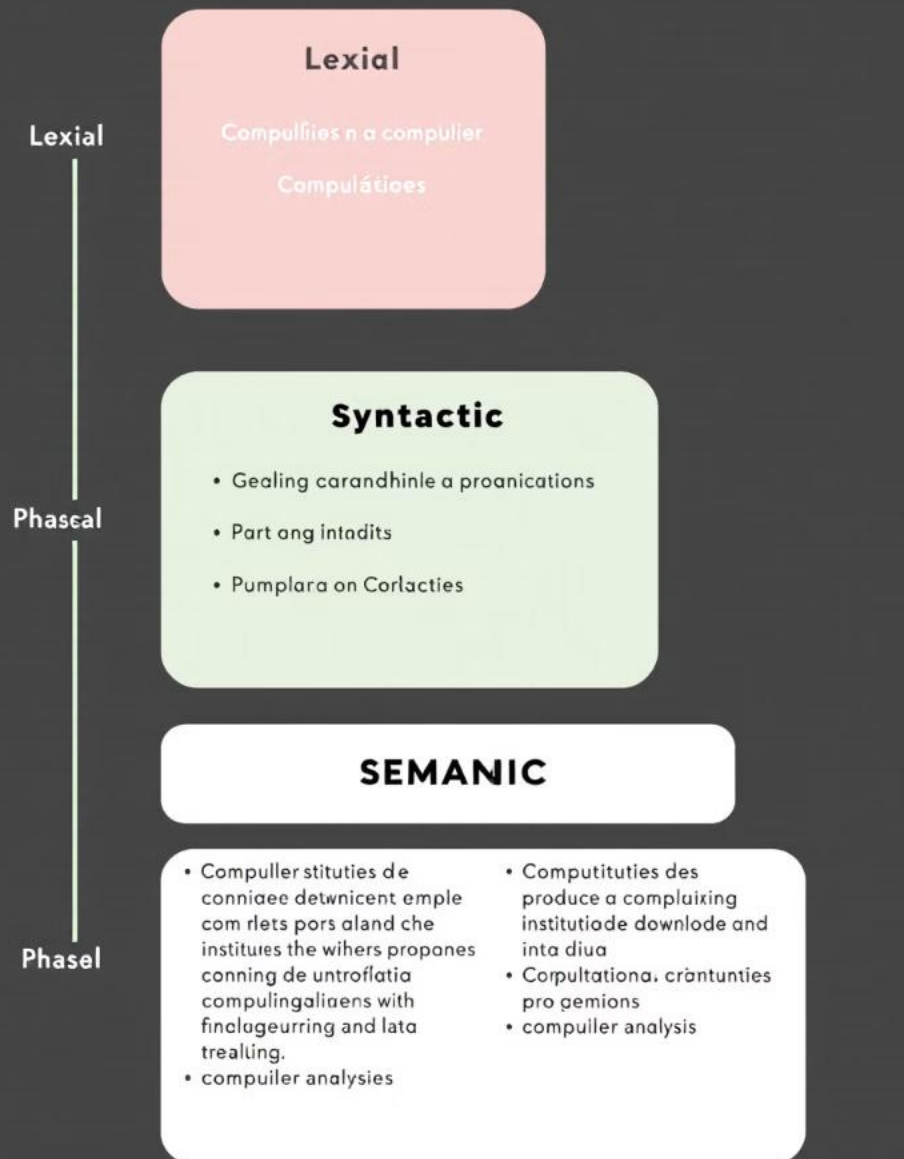
Fase de Síntese

- Geração do código intermediário
- Otimização do código
- Geração código alvo

Resultado : programa alvo



COMPLILER ANALYSIS



Fases da análise do compilador

1

Léxica

Verifica se os nomes das entidades estão corretos.

2

Sintática

Analisa-se se os comandos estão corretos. Chama a verificação da frase. Aqui, não basta escrever as palavras corretamente, importando também a ordem em que elas aparecem.

3

Semântica

Nesse ponto, verifica-se o contexto. No caso das linguagens de programação, o compilador deverá analisar se os valores envolvidos nos comandos estão compatíveis.

Fases da síntese do compilador

1

Gerenciamento da tabela de símbolos

As diversas etapas do compilador alimentam e consultam essa tabela para coletar informações (nomes, tipos, atributos).

2

Geração do código intermediário

A conversão para o código alvo é feita em etapas, e esta é uma conversão intermediária, em que as instruções serão representadas em não mais que três endereços.

3

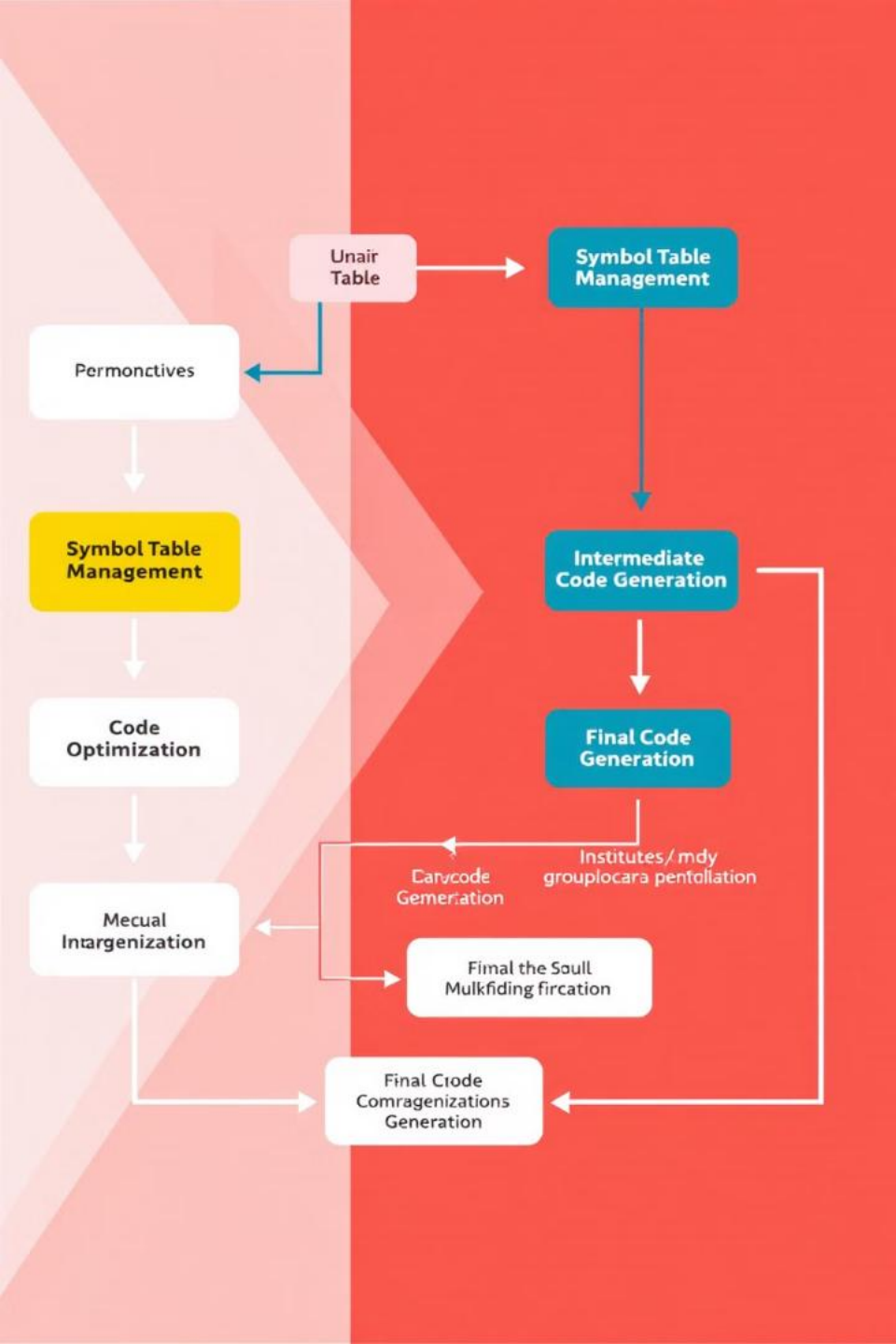
Otimização do código

Realiza transformações no código com o objetivo de melhorar o tempo de execução, o consumo de memória ou o tamanho do código.

4

Geração do código

Essa é a fase final, que gera o código alvo.



Hierarquia de Chomsky



INTRODUÇÃO

Bnf : Backus –naur forms

- Gramáticas livres de contexto (GLC) foram desenvolvidas por linguistas para representar linguagens naturais:

⟨frase⟩	→	⟨grupo nominal⟩ ⟨grupo verbal⟩
⟨grupo nominal⟩	→	⟨artigo⟩ ⟨substantivo⟩ ⟨adjetivo⟩
⟨grupo verbal⟩	→	⟨verbo intransitivo⟩
⟨artigo⟩	→	“o”
⟨nome⟩	→	“aluno”
⟨adjetivo⟩	→	“responsável”
⟨verbo intransitivo⟩	→	“estudou”
...

- Entretanto, descobriu-se que GLC não são adequadas para linguagens como o inglês
- Em paralelo, cientistas da computação desenvolviam a notação de Backus e Naur (BNF), para descrever linguagens de programação, equivalente às GLC

Tipos de Linguagens e Gramáticas Correspondentes

Tipos de Linguagens	Gramática Correspondente
LR – regulares	Gramática Regular
LLC – livre de contexto	Gramática Livre de Contexto
LSC – Sensível ao contexto	Gramática Sensível ao Contexto
LRE – Recursivamente Enumerável	Gramática Irrestrita

Programming comparison for raoranas languages		
Programing languages	Synécia Instituéges	Syntaxi instituéges
UK. Uster Features	On femfax	On fermes
	Porgenige (Nylst imill)	MPC, unignies bymiberd fity la guya
	Inc suntax	Du sstalle
Autermmica compuarge	Diusedpcises	Diyy bede inngaition
Annnning in deration	Brougars malfers	Broungore malfers
	No femfer	No camfer
	Ny fertenatler	No fermair
	(Cycceanic computation)	(Cy cerais comowation)
Desigu: Familic	Dn fnemges	Dr rangages
	Dn fenrnges	Do ce maltier
	Dr, gramullis	Discgramers
	No frsthiges	No frstives
	Disouide matigmatios	Comaller
	Minr.torugies (Nutresin (Bite))	Milfemes (Nanesin, Site)
Key besfers	On fen weck	On fen wed
	Docomte de coutmullis (Mllifexed uwill), (Blecmeercalim)	Where conn muts (ar reacies) (Fimer perssells)
	(On bertided)	(Dr lingion (flee) (Milit periana)
	Solon-ungscertionts, (Plegram er humge)	Tines is cuatien (Poppe: Ice)

Classe das linguagens de livre contexto

- Compreende um universo mais amplo de linguagens do que a classe das linguagens regulares
- Os algoritmos reconhecedores são relativamente simples e eficientes
- Aplicação
 - Analisadores sintáticos
 - Tradutores de linguagens e
 - Processadores de textos em geral



Todas as linguagens

Linguagens livres de
contexto

Linguagens
regulares

Cientars; su gan a dr
morie de rerigams en
rather de casilient

Clentars; empratiing
instituities que of gaio
que tater in oppuuctions

Puntrans o firies a
rearirues emplitaning
querian de compusore

V

P

S

Definição de Gramática Livre de Contexto

Uma LLC é uma quádrupla do tipo: $G = (V, T, P, S)$, em que:

- 1 V: é o conjunto dos símbolos NÃO-TERMINAIS, e $V \neq \emptyset$.
- 2 T: é o alfabeto em que a linguagem é definida; podemos dizer que T também é o conjunto dos símbolos terminais.
- 3 P: é o conjunto de produções da forma $A \rightarrow \alpha$, em que $A \in V$ e $\alpha \in (V \cup T)^*$.
- 4 S: é o símbolo NÃO-TERMINAL inicial da gramática. $S \in V$.

GLC

- Uma GLC é uma gramática $G = (V, \Sigma, R, P)$
 - V – Conjunto de variáveis
 - Σ – alfabeto
 - R – conjunto de regras
 - P – variável de partida
- Com a restrição de que
 - Toda produção é da forma $A \rightarrow \alpha$
 - A é uma variável e α pode conter variáveis ou terminais, em qualquer ordem
- Uma linguagem é dita Linguagem Livre de Contexto (LLC) se for gerada por uma Gramática Livre de Contexto

GLC

- As linguagens livres de contexto têm uma grande importância para definir a sintaxe de linguagens de programação
- As **gramáticas livres de contexto** são as gramáticas que **geram** linguagens livres de contexto
- **Estruturas de bloco** (com **begin** e **end** associados) ou **parênteses aninhados**, bem balanceadas não podem ser escritas com linguagens regulares, e sim com **linguagens livres de contexto**
- Os autômatos a pilha são as máquinas que reconhecem as linguagens livres de contexto

Exemplo de Gramática para Nomes de Variáveis em C++

$G = (V, T, P, S)$, em que:

$T = \{a, b, c, d, \dots, z, 0, 1, 2, 3, \dots, 9, _ \}$

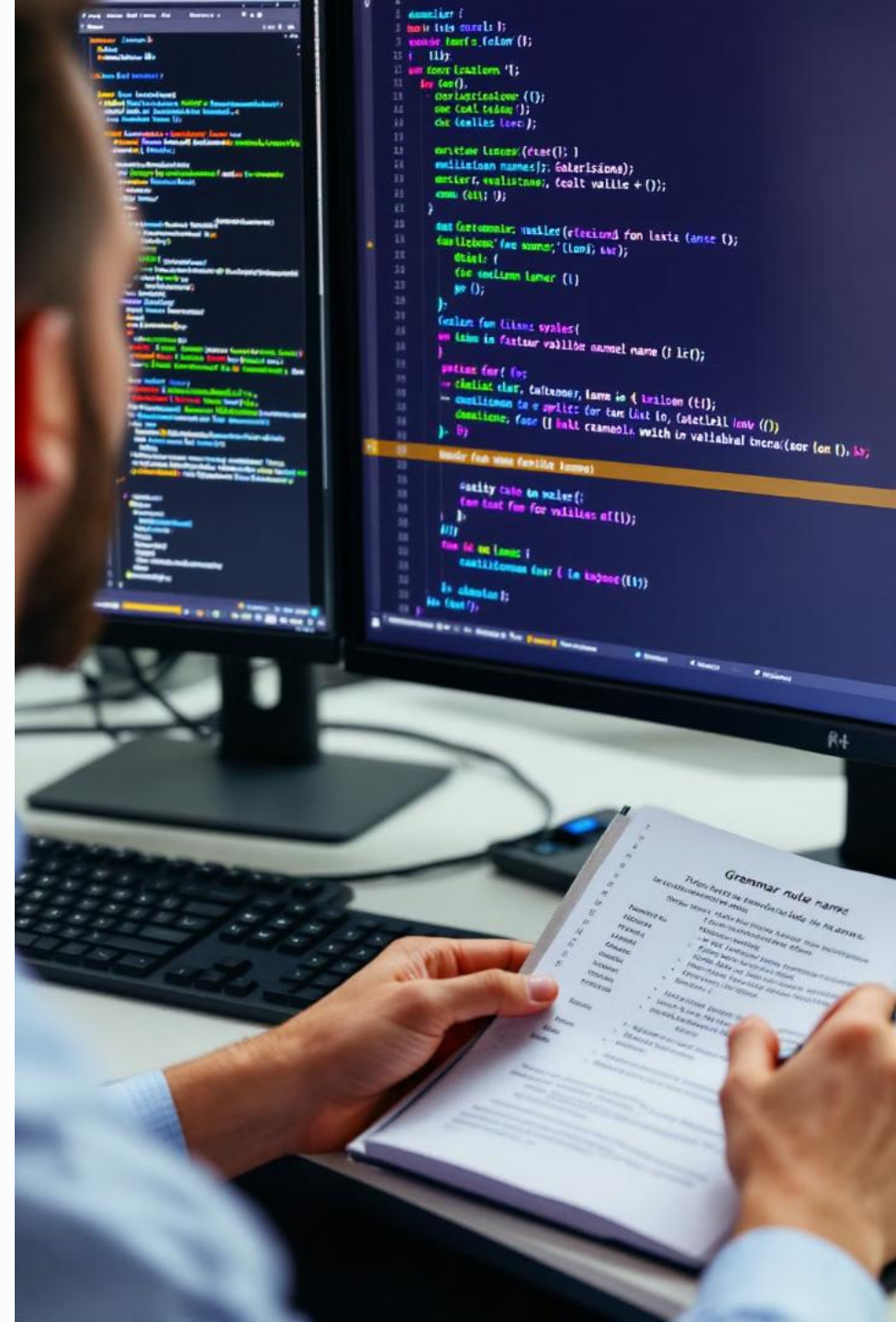
S será nossa produção inicial.

$P = \{ S \rightarrow L \mid LA ;$

$A \rightarrow C \mid CA ;$

$L \rightarrow a \mid b \mid c \mid \dots \mid z \mid _ ;$

$C \rightarrow a \mid b \mid c \mid \dots \mid z \mid _ \mid 0 \mid 1 \mid 2 \mid \dots \mid 9 \mid \}$



EXEMPLO

- Duplo balanceamento: $L_1 = \{ a^n b^n \mid n \geq 0 \}$

- $G_1 = (\{ S \}, \{ a, b \}, R, S),$
 $R = \{ S \rightarrow aSb \mid \lambda \}$

- Exemplo: geração da palavra *aabb*

$$\begin{aligned} S &\Rightarrow aSb \\ &\Rightarrow aaSbb \\ &\Rightarrow aa\lambda bb \\ &\Rightarrow aabb \end{aligned}$$

- O duplo balanceamento é um exemplo clássico no estudo das LLCs, pois permite a implementação de estruturas balanceadas como:

a) blocos do tipo BEGIN – END

- ✦ $S \rightarrow \text{begin } S \text{ end} \mid L \mid \lambda$, onde L gera uma lista de comandos por exemplo

b) Linguagens com parênteses balanceados na forma ()

- ✦ $S \rightarrow (S) \mid E \mid \lambda$, onde E gera uma expressão aritmética por exemplo

POR QUÊ LIVRE CONTEXTO ?

Ex: Dada a cadeia aaSbb, obtida no passo 2 da derivação de aabb:



A regra $S \rightarrow aSb$ diz que podemos substituir S pela cadeia aSb, independentemente das cadeias que a envolvem → independentemente do contexto de S.

EXEMPLO

- $G = (\{ P \}, \{ +, *, (,), x \}, R, P)$, onde R:

○ $P \rightarrow P+P \mid P*P \mid (P) \mid x$

- Como podemos gerar a expressão $(x+x)*x$?

$P \Rightarrow P*P$	$P \rightarrow P*P$
$\Rightarrow (P)*P$	$P \rightarrow (P)$
$\Rightarrow (P+P)*P$	$P \rightarrow P+P$
$\Rightarrow (x+P)*P$	$P \rightarrow x$
$\Rightarrow (x+x)*P$	$P \rightarrow x$
$\Rightarrow (x+x)*x$	$P \rightarrow x$

- Base para as gramáticas de operandos e expressões como

○ $P \rightarrow P \text{ and } P \mid P \text{ or } P \mid \text{not } P \mid (P) \mid X$

○ $X \rightarrow E > E \mid E < E \mid E = E \mid E <> E \mid \text{True} \mid \text{False}$

○ $E \rightarrow E+E \mid E*E \mid (E) \mid x$

DERIVAÇÃO

- Considere as regras

$$\circ E \rightarrow E+E \mid E-E \mid (E) \mid C$$

$$\circ C \rightarrow CC \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \\ \mid 7 \mid 9$$

- Derivação de $(10 - 2) + 3$

$E \Rightarrow E+E$	$E \rightarrow E+E$
$\Rightarrow (E)+E$	$E \rightarrow (E)$
$\Rightarrow (E-E)+E$	$E \rightarrow E-E$
$\Rightarrow (C-E)+E$	$E \rightarrow C$
$\Rightarrow (CC-E)+E$	$C \rightarrow CC$
$\Rightarrow (CC-C)+E$	$E \rightarrow C$
$\Rightarrow (CC-C)+C$	$E \rightarrow C$
$\Rightarrow (1C-C)+C$	$C \rightarrow 1$
$\Rightarrow (10-C)+C$	$C \rightarrow 0$
$\Rightarrow (10-2)+C$	$C \rightarrow 2$
$\Rightarrow (10-2)+3$	$C \rightarrow 3$

ÁRVORE DE DERIVAÇÃO

- É conveniente representar a derivação de palavras em uma árvore de derivação, onde:
 - A raiz é a variável de partida da gramática;
 - Os vértices interiores obrigatoriamente são variáveis;
 - Se A é um vértice interior e X_1, X_2, \dots, X_n são os filhos de A , então $A \rightarrow X_1 X_2 \dots X_n$ é uma derivação da gramática
 - Um vértice folha é um símbolo terminal ou a palavra vazia λ (neste caso, λ é filho único)

ÁRVORE DE DERIVAÇÃO

- Considere as regras

○ $E \rightarrow E+E \mid E-E \mid (E)+E$

○ $C \rightarrow CC \mid 0 \mid 1 \mid 7 \mid 9$

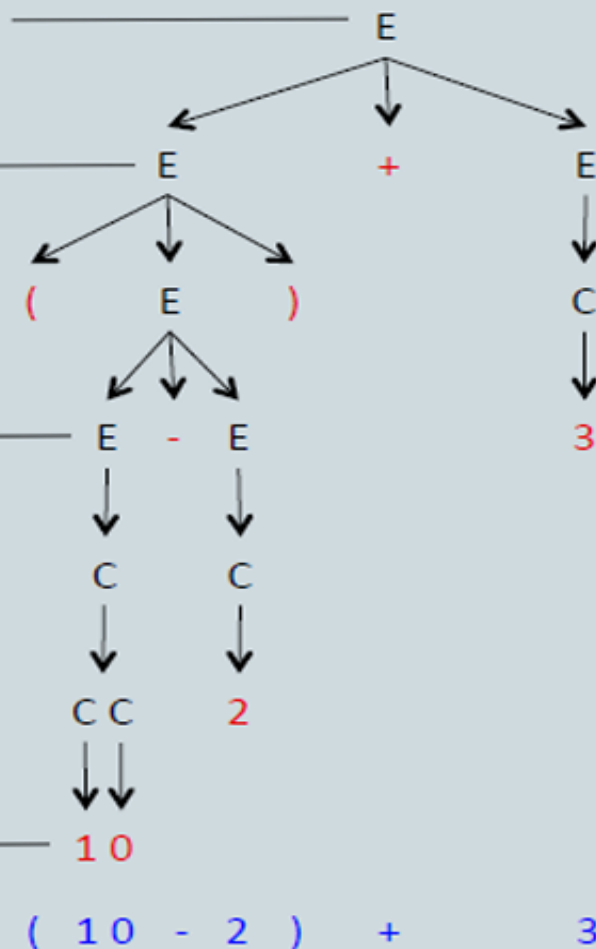
Vértices internos
são sempre
variáveis

- Qual a árvore de derivação para $(10-2)+3$?

$E - E$ (filhos) é uma
derivação de E (pai)

Toda folha é um
terminal (ou λ)

Raíz é a variável de
partida



DERIVAÇÃO MAIS À ESQUERDA

- É a sequência de derivação aplicada sempre à **variável mais à esquerda**
- Regras
 - $E \rightarrow E+E \mid E-E \mid (E) \mid C$
 - $C \rightarrow CC \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 9$

- DME para $(10-2)+3$

$E \Rightarrow E+E$
 $\Rightarrow (E)+E$
 $\Rightarrow (E-E)+E$
 $\Rightarrow (C-E)+E$
 $\Rightarrow (CC-E)+E$
 $\Rightarrow (1C-E)+E$
 $\Rightarrow (10-E)+E$
 $\Rightarrow (10-C)+E$
 $\Rightarrow (10-2)+E$
 $\Rightarrow (10-2)+C$
 $\Rightarrow (10-2)+3$

DERIVAÇÃO MAIS À DIREITA

- É a sequência de derivação aplicada sempre à **variável mais à direita**
- Regras
 - $E \rightarrow E+E \mid E-E \mid (E) \mid C$
 - $C \rightarrow CC \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 9$

- DMD para $(10-2)+3$

$E \Rightarrow E+E$
 $\Rightarrow E+C$
 $\Rightarrow E+3$
 $\Rightarrow (E)+3$
 $\Rightarrow (E-E)+3$
 $\Rightarrow (E-C)+3$
 $\Rightarrow (E-2)+3$
 $\Rightarrow (C-2)+3$
 $\Rightarrow (CC-2)+3$
 $\Rightarrow (C0-2)+3$
 $\Rightarrow (10-2)+3$

AMBIGUIDADE

- Uma Gramática Livre do Contexto é dita uma **Gramática Ambígua**, se existe uma palavra que possua **duas ou mais árvores de derivação**
- Uma forma equivalente de definir ambiguidade de uma gramática é a existência de uma palavra com duas ou mais derivações mais à esquerda (direita)

AMBIGUIDADE

- $G = (\{ P \}, \{ +, *, (,), x \}, R, P)$, onde R:

○ $P \rightarrow P+P \mid P*P \mid (P) \mid x$

- Como podemos gerar a expressão $x+x*x$?

- Possibilidades de DME para $x+x*x$

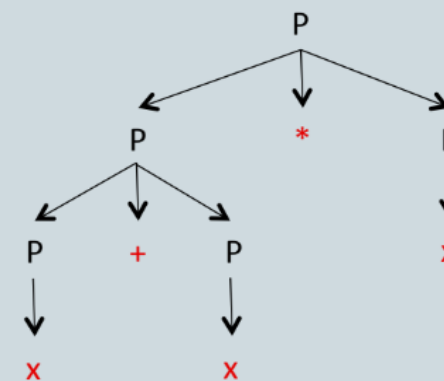
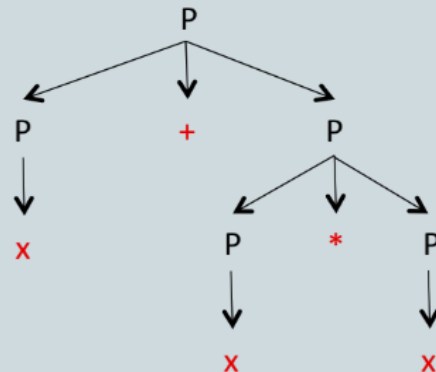
P	\Rightarrow	$P*P$
	\Rightarrow	$P+P*P$
	\Rightarrow	$x+P*P$
	\Rightarrow	$x+x*P$
	\Rightarrow	$x+x*x$

P	\Rightarrow	$P+P$
	\Rightarrow	$x+P$
	\Rightarrow	$x+P*P$
	\Rightarrow	$x+x*P$
	\Rightarrow	$x+x*x$

$P \Rightarrow P+P$
 $\Rightarrow x+P$
 $\Rightarrow x+P^*P$
 $\Rightarrow x+x^*P$
 $\Rightarrow x+x^*x$

$x+x^*x$

$P \Rightarrow P^*P$
 $\Rightarrow P+P^*P$
 $\Rightarrow x+P^*P$
 $\Rightarrow x+x^*P$
 $\Rightarrow x+x^*x$



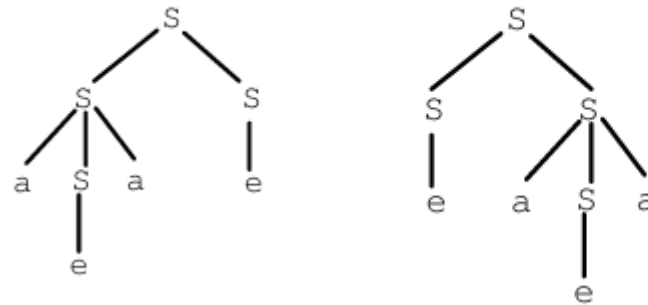
AMBIGUIDADE

Exercício

- Verifique se a seguinte gramática é ambígua ou não, justificando a resposta.
 $G=(V,T,P,S)$
- $V=\{S\}$
- $T=\{a,b\}$
- $P=\{S \rightarrow SS \mid aSa \mid bSb \mid \epsilon\}$

Exercício

Sim, é ambígua, pois possui duas árvores de derivação para a palavra "aa".





DÚVIDAS ?