

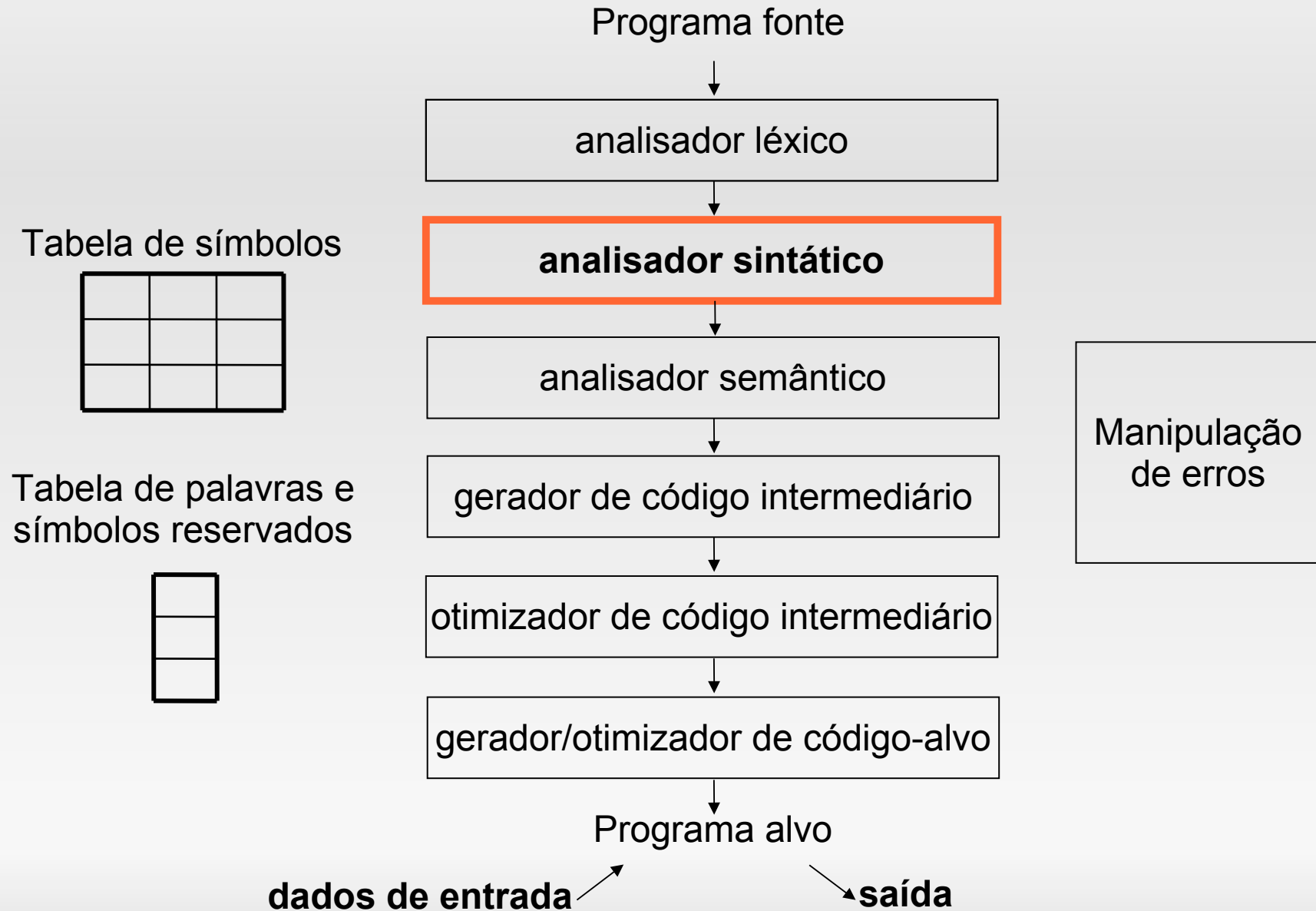
Construção de Compiladores

Análise Sintática – parte1

Introdução

Profa. Helena Caseli
helenacaseli@dc.ufscar.br

Processo de Tradução



Análise Sintática

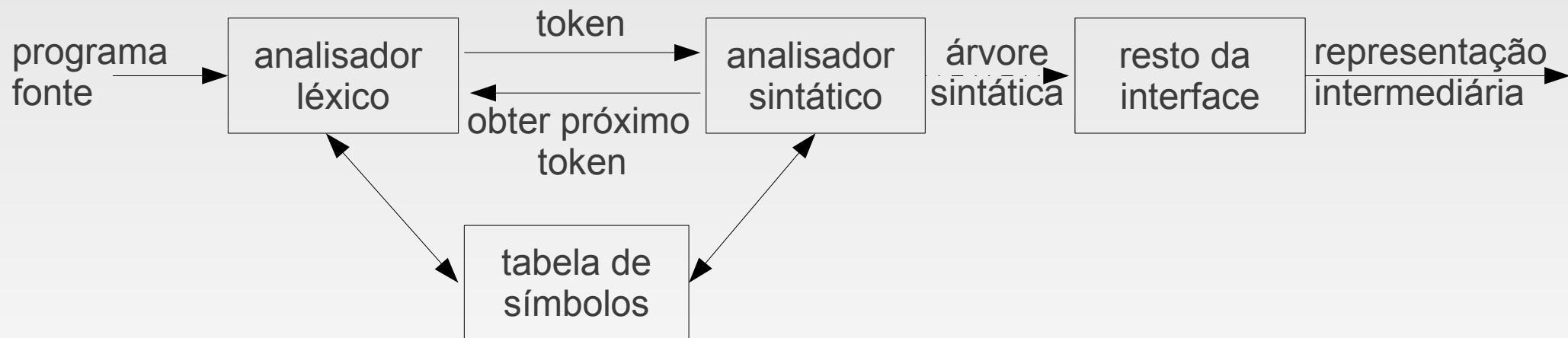
- O que é?
 - Etapa que determina a **sintaxe (estrutura)** de um programa
 - É tarefa do analisador sintático
 - Determinar a estrutura sintática de um programa a partir dos *tokens* produzidos pelo analisador léxico e
 - Construir explícita ou implicitamente uma árvore sintática que represente essa estrutura

Análise Sintática

- O que é?
 - A sintaxe de uma linguagem de programação é normalmente
 - Especificada pelas **regras gramaticais** de uma **gramática livre de contexto**
 - Representada por uma **árvore sintática**
 - Diferentemente da análise léxica
 - Na sintática há **recursividade** nas regras e na estrutura de árvore

Análise Sintática

- Como é feita?
 - O analisador sintático obtém uma cadeia de *tokens* proveniente do analisador léxico e verifica se a mesma pode ser gerada pela gramática da linguagem fonte



Análise Sintática

- Como é feita?

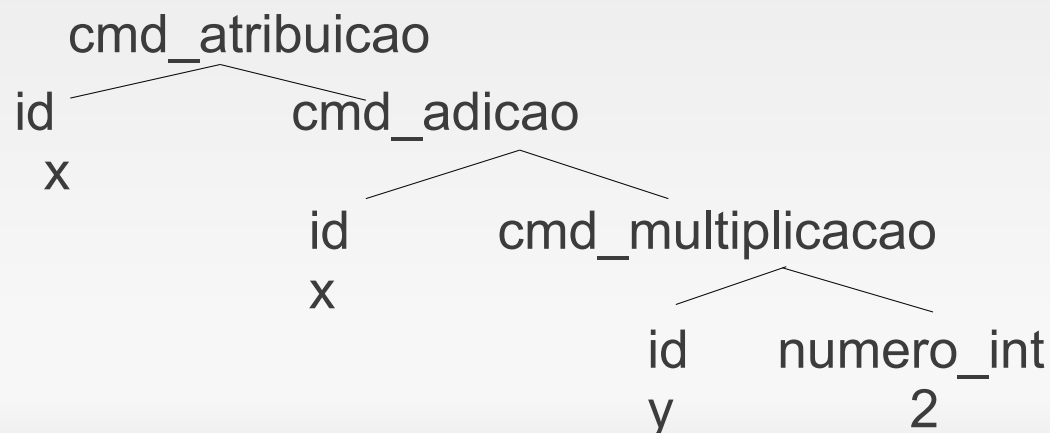
- Exemplo

$\langle x, id \rangle \langle :=, := \rangle \langle x, id \rangle \langle +, op \rangle \langle y, id \rangle \langle *, op \rangle \langle 2, num \rangle$



$cmd_atribuicao \rightarrow id_1 := id_1 op_1 id_2 op_2 num$

Árvore sintática



Análise Sintática

- Como é feita?
 - Há duas abordagens gerais para análise sintática
 - Descendente (ou Top-down)
 - raiz → folhas
 - Ascendente (ou Bottom-up ou redutiva)
 - folhas → raiz
 - Orientadas pelo sentido de construção da árvore sintática correspondente

Análise Sintática

- O que é?
 - A sintaxe de uma linguagem de programação é normalmente
 - Especificada pelas **regras gramaticais** de uma **gramática livre de contexto**
 - Representada por uma **árvore sintática**

Análise Sintática

- Gramática Livre de Contexto (GLC) – recordando
 - Especificação para a estrutura sintática de uma linguagem de programação
 - Uma gramática livre de contexto consiste de:
 1. Um conjunto T de terminais
 2. Um conjunto N de não terminais (disjunto de T)
 3. Um conjunto P de produções, ou regras gramaticais, da forma $A \rightarrow \alpha$ (ou $A ::= \alpha$), em que A é um elemento de N e α é um elemento de $(T \cup N)^*$ (uma sequência de terminais e não terminais que pode ser vazia)
 4. Um símbolo inicial S do conjunto N

Análise Sintática

- Gramática Livre de Contexto (GLC) – recordando

- Exemplo

$\langle \text{declaração} \rangle ::= \langle \text{if-decl} \rangle \mid \text{outra}$

$\langle \text{if-decl} \rangle ::= \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle$

$\mid \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle \text{ else } \langle \text{declaração} \rangle$

$\langle \text{exp} \rangle ::= 0 \mid 1$

- OU (com ϵ -produções)

$\langle \text{declaração} \rangle ::= \langle \text{if-decl} \rangle \mid \text{outra}$

$\langle \text{if-decl} \rangle ::= \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle \langle \text{else-parte} \rangle$

$\langle \text{else-parte} \rangle ::= \text{else } \langle \text{declaração} \rangle \mid \epsilon$

$\langle \text{exp} \rangle ::= 0 \mid 1$

Análise Sintática

- Gramática Livre de Contexto (GLC) – recordando
 - Derivação
 - Uma derivação sobre G é da forma $S \Rightarrow^* w$ em que $w \in T^*$
 - w é uma cadeia apenas de terminais denominada **sentença**
 - S é o símbolo inicial de G
 - Derivação à esquerda e Derivação à direita
 - Linguagem
 - $L(G) = \{w \in T^* \mid \text{existe uma derivação } S \Rightarrow^* w \text{ de } G\}$
 - o conjunto de sentenças deriváveis de G
 - Linguagem livre de contexto
- Cada derivação leva a uma árvore sintática
 - Cada árvore sintática tem uma derivação à esquerda e uma derivação à direita que são únicas

Análise Sintática

- Árvore Sintática – recordando
 - Uma árvore sintática sobre a gramática G é uma árvore com as seguintes propriedades:
 1. Cada nó é rotulado com um terminal, não terminal ou ϵ
 2. O nó-raiz é rotulado com o símbolo inicial S
 3. Cada nó-folha é rotulado com um terminal ou ϵ
 4. Cada nó não-folha é rotulado com um não terminal
 5. Se um nó com rótulo $A \in N$ tiver n filhos com rótulos X_1, X_2, \dots, X_n (terminais ou não terminais), então $A \rightarrow X_1 X_2 \dots X_n \in P$ (uma produção da gramática)

Análise Sintática

- Árvore Sintática – recordando

- Exemplo

Dada a gramática

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle \mid (\langle \text{exp} \rangle) \mid \text{número}$

$\langle \text{op} \rangle ::= + \mid - \mid *$

Derivando a cadeia **número + número**

Análise Sintática

- Árvore Sintática – recordando

- Exemplo

Dada a gramática

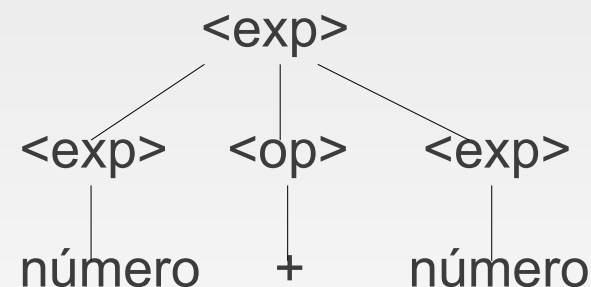
$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle \mid (\langle \text{exp} \rangle) \mid \text{número}$

$\langle \text{op} \rangle ::= + \mid - \mid *$

Derivação

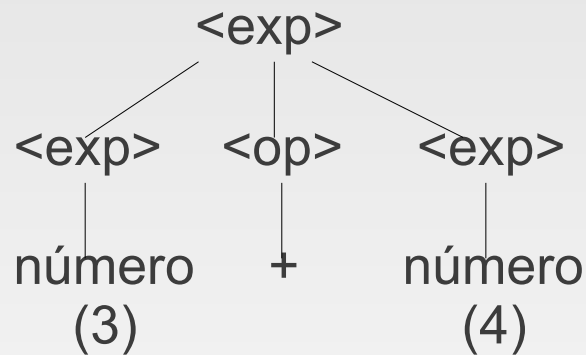
$\begin{aligned} \langle \text{exp} \rangle &\Rightarrow \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle \\ &\Rightarrow \text{número} \langle \text{op} \rangle \langle \text{exp} \rangle \\ &\Rightarrow \text{número} + \langle \text{exp} \rangle \\ &\Rightarrow \text{número} + \text{número} \end{aligned}$

Árvore sintática



Análise Sintática

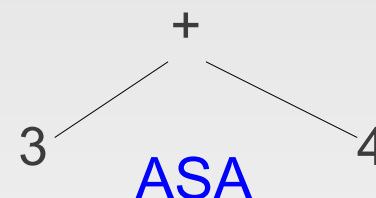
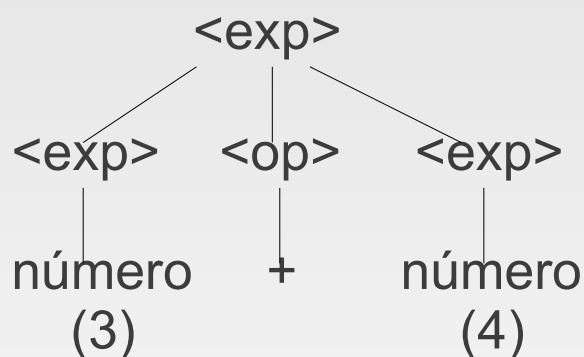
- Árvore Sintática Abstrata (ASA)
 - Forma compacta de representar uma árvore sintática
 - Exemplo
 - Considere a árvore sintática para a expressão 3+4



Qual é a árvore sintática abstrata equivalente?

Análise Sintática

- Árvore Sintática Abstrata (ASA)
 - Forma compacta de representar uma árvore sintática
 - Exemplo
 - A árvore sintática abstrata equivalente é



- Um analisador sintático efetua todos os passos representados na árvore sintática mas, em geral, constrói apenas uma árvore sintática abstrata

Análise Sintática

- Gramática Livre de Contexto e Árvore Sintática

- Exercício

- Dada a gramática

- $\langle \text{declaração} \rangle ::= \langle \text{if-decl} \rangle \mid \text{outra}$

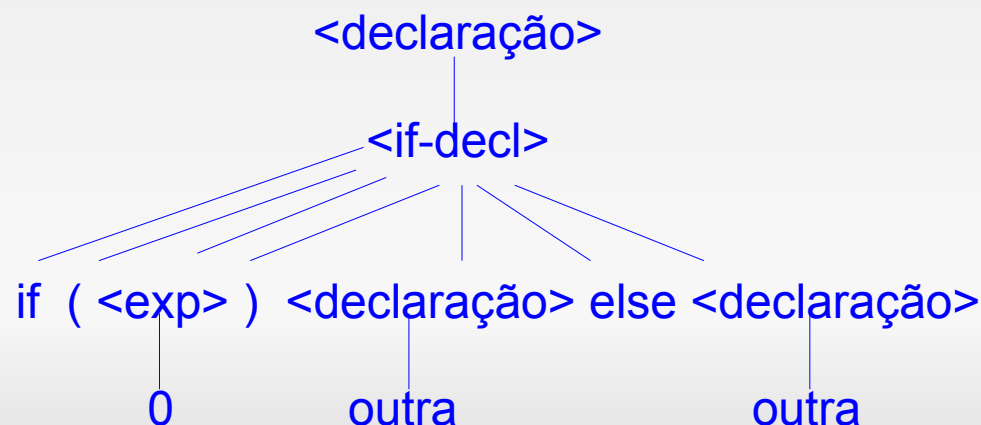
- $\langle \text{if-decl} \rangle ::= \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle$

- $\mid \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle \text{ else } \langle \text{declaração} \rangle$

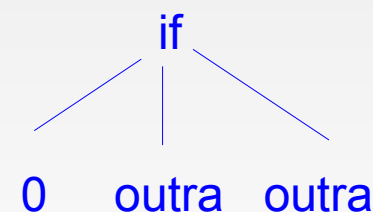
- $\langle \text{exp} \rangle ::= 0 \mid 1$

- E a cadeia: **if (0) outra else outra**

Árvore Sintática



ASA



Análise Sintática

- Ambiguidade
 - Uma gramática ambígua é aquela que permite a geração de mais de uma derivação à esquerda (ou à direita) para uma mesma sentença

Gramática

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $\quad \quad \quad | (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + \mid - \mid *$

Por exemplo, construa 2 árvores de derivação à esquerda para a expressão **3+4-5**

Análise Sintática

- Ambiguidade
 - Uma gramática ambígua é aquela que permite a geração de mais de uma derivação à esquerda (ou à direita) para uma mesma sentença
 - Eliminar a ambiguidade
 - Não é tão simples como eliminar o não determinismo dos autômatos finitos
 - Felizmente, existem técnicas para tratar as ambiguidades típicas
- ➔ A ambiguidade é apenas um dos problemas que precisam ser contornados por meio de alterações na gramática

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Eliminação de recursão à esquerda
 - Fatoração à esquerda

A gramática deve ser não ambígua para que a estrutura sintática de um programa seja determinada com precisão

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Eliminação de recursão à esquerda
 - Fatoração à esquerda

Gramáticas recursivas à esquerda não podem ser processadas por métodos de análise sintática descendente, pois podem entrar em loop infinito

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Eliminação de recursão à esquerda
 - Fatoração à esquerda

A fatoração à esquerda é essencial para analisadores preditivos já que permite que a escolha de qual produção usar seja postergada até que se saiba o suficiente para realizar o processamento sem erros

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Eliminação de recursão à esquerda
 - Fatoração à esquerda

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Regras de eliminação de ambiguidade
 - Corrigem a ambiguidade sem alterar a gramática
 - Porém, a estrutura sintática da linguagem não é mais determinada apenas pela gramática
 - Reescrita da gramática considerando-se regras de
 - Precedência de operadores
 - Associatividade à esquerda

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Reescrita da gramática – precedência de operadores

Gramática ambígua

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle \text{ <op> } \langle \text{exp} \rangle \\ &\quad | (\langle \text{exp} \rangle) | \text{ número} \\ \langle \text{op} \rangle &::= + | - | * \end{aligned}$$

Os operadores com precedência diferente devem ser agrupados em regras distintas

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Reescrita da gramática – precedência de operadores

Gramática ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + | - | *$

A multiplicação é agrupada sob a regra $\langle \text{termo} \rangle$ e a adição e a subtração são agrupadas sob a regra $\langle \text{exp} \rangle$

Gramática em transformação

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{exp} \rangle$
 $| \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{termo} \rangle$
 $| \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Reescrita da gramática – precedência de operadores

Gramática ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + | - | *$

Como o caso base de $\langle \text{exp} \rangle$ é termo, a adição e a subtração aparecerão mais próximas da raiz (menor precedência)

Gramática em transformação

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{exp} \rangle$
 $| \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{termo} \rangle$
 $| \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Reescrita da gramática

Gramática ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + | - | *$

Essa gramática ainda é ambígua? Ainda é possível gerar 2 árvores sintáticas com derivação à esquerda para a expressão **3+4-5**?

Gramática em transformação

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{exp} \rangle$
 $| \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{termo} \rangle$
 $| \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Reescrita da gramática – associatividade à esquerda

Gramática ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + | - | *$

Regras recursivas à esquerda
garantem a associatividade à
esquerda dos operadores

Gramática em transformação

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \textcolor{blue}{\langle \text{exp} \rangle}$
 $| \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \textcolor{blue}{\langle \text{termo} \rangle}$
 $| \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Reescrita da gramática – associatividade à esquerda

Gramática ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + | - | *$

Gramática não ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \text{<termo>}$
 $| \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \text{<fator>}$
 $| \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Agora, com a gramática não-ambígua, só é possível obter 1 árvore de derivação à esquerda para a expressão **3+4-5**

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Reescrita da gramática – associatividade à esquerda

Gramática ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + | - | *$

Gramática não ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle$
 $| \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle$
 $| \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Porém, a gramática resultante
tem um problema para ASD ...

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Eliminação de recursão à esquerda
 - Fatoração à esquerda

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de recursão à esquerda – tipos de recursão
 - Recursão imediata à esquerda simples
$$\langle \text{exp} \rangle ::= \underline{\langle \text{exp} \rangle} \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$$
 - Recursão imediata à esquerda geral
$$\langle \text{exp} \rangle ::= \underline{\langle \text{exp} \rangle} + \langle \text{termo} \rangle \mid \underline{\langle \text{exp} \rangle} - \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$$
 - Recursão imediata à esquerda – a recursão à esquerda ocorre apenas na produção de um único não terminal ($\langle \text{exp} \rangle$)
 - Recursão à esquerda geral
$$\begin{aligned} \langle \mathbf{A} \rangle &::= \underline{\langle \mathbf{B} \rangle} b \mid \dots \\ \underline{\langle \mathbf{B} \rangle} &::= \langle \mathbf{A} \rangle a \mid \dots \end{aligned}$$
 - Quase nunca ocorre em gramáticas de linguagens de programação

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de recursão à esquerda
 - Recursão imediata à esquerda simples
$$\langle \text{exp} \rangle ::= \underline{\langle \text{exp} \rangle} \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$$

Transforma regras do tipo
$$\langle A \rangle ::= \langle A \rangle \alpha \mid \beta$$

em

$$\begin{aligned} \langle A \rangle &::= \beta \langle A' \rangle \\ \langle A' \rangle &::= \alpha \langle A' \rangle \mid \varepsilon \end{aligned}$$

Reescreve-se a regra gramatical como duas regras:

- uma que gera β
- e uma que gera as repetições de α com base em recursão à direita ao invés de recursão à esquerda

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de recursão à esquerda
 - Recursão imediata à esquerda simples
$$\langle \text{exp} \rangle ::= \underline{\langle \text{exp} \rangle} \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$$

Transforma regras do tipo
$$\langle A \rangle ::= \langle A \rangle \alpha \mid \beta$$

em

$$\begin{aligned}\langle A \rangle &::= \beta \langle A' \rangle \\ \langle A' \rangle &::= \alpha \langle A' \rangle \mid \varepsilon\end{aligned}$$

Antes

$$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$$

Após remoção da recursão à esquerda

$$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$$
$$\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{termo} \rangle \langle \text{exp}' \rangle \mid \varepsilon$$

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de recursão à esquerda
 - Recursão imediata à esquerda geral
$$\langle \text{exp} \rangle ::= \underline{\langle \text{exp} \rangle} + \langle \text{termo} \rangle \mid \underline{\langle \text{exp} \rangle} - \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$$

Transforma regras do tipo

$$\langle A \rangle ::= \langle A \rangle \alpha_1 \mid \langle A \rangle \alpha_2 \mid \dots \mid \langle A \rangle \alpha_n \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_m$$

nas quais nenhum dos β_1, \dots, β_m começa com $\langle A \rangle$, em

$$\langle A \rangle ::= \beta_1 \langle A' \rangle \mid \beta_2 \langle A' \rangle \mid \dots \mid \beta_m \langle A' \rangle$$

$$\langle A' \rangle ::= \alpha_1 \langle A' \rangle \mid \alpha_2 \langle A' \rangle \mid \dots \mid \alpha_n \langle A' \rangle \mid \varepsilon$$

Antes

$$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle + \langle \text{termo} \rangle \mid \langle \text{exp} \rangle - \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$$

Após remoção da recursão à esquerda

$$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$$

$$\langle \text{exp}' \rangle ::= + \langle \text{termo} \rangle \langle \text{exp}' \rangle \mid - \langle \text{termo} \rangle \langle \text{exp}' \rangle \mid \varepsilon$$

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de recursão à esquerda

Gramática ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $\quad \quad \quad | (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + | - | *$

Gramática não ambígua porém
recursiva à esquerda

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle | \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle | \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Como seria a gramática final não ambígua nem recursiva à esquerda?

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de recursão à esquerda

Gramática ambígua

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $\quad \quad \quad | (\langle \text{exp} \rangle) | \text{número}$
 $\langle \text{op} \rangle ::= + | - | *$

Gramática não ambígua porém
recursiva à esquerda

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle | \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle | \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Gramática final não ambígua nem recursiva à esquerda

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$
 $\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{termo} \rangle \langle \text{exp}' \rangle | \epsilon$
 $\langle \text{soma} \rangle ::= + | -$
 $\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$
 $\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{fator} \rangle \langle \text{termo}' \rangle | \epsilon$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) | \text{número}$

Análise Sintática

- Alterações nas gramáticas
 - Eliminação de ambiguidade
 - Eliminação de recursão à esquerda
 - Fatoração à esquerda

Análise Sintática

- Alterações nas gramáticas
 - Fatoração à esquerda

O que fazer com regras do tipo:

```
<if-decl> ::= if ( <exp> ) <declaração>  
           | if ( <exp> ) <declaração> else <declaração>
```


Análise Sintática

- Alterações nas gramáticas

- Fatoração à esquerda

$\langle A \rangle ::= \alpha \beta \mid \alpha \gamma$

em que α , β , γ são sequências de terminais e não terminais
(duas ou mais escolhas compartilham um prefixo comum)

→ A solução nesses casos é "fatorar" α e reescrever a regra como duas regras

$\langle A \rangle ::= \alpha \langle A' \rangle$

$\langle A' \rangle ::= \beta \mid \gamma$

α deve ser a cadeia mais longa compartilhada do lado direito

Antes

$\langle \text{if-decl} \rangle ::= \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle$
 $\quad \quad \quad \mid \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle \text{ else } \langle \text{declaração} \rangle$

Após fatoração à esquerda

$\langle \text{if-decl} \rangle ::= \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle \langle \text{else-parte} \rangle$
 $\langle \text{else-parte} \rangle ::= \text{else } \langle \text{declaração} \rangle \mid \epsilon$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Úteis em algoritmos de análise sintática descendente e ascendente

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Conjunto Primeiros
 - Composto por **terminais** e possivelmente ϵ
 - $\text{Primeiro}(A) = \{x\}$, onde A (não terminal) produz x como seu símbolo mais à esquerda

Gramática

$\langle S \rangle ::= \langle A \rangle \langle B \rangle$

$\langle A \rangle ::= a \langle A \rangle \mid a$

$\langle B \rangle ::= b \langle B \rangle \mid b$

Primeiros

$\text{Primeiro}(S) = \{a\}$

$\text{Primeiro}(A) = \{a\}$

$\text{Primeiro}(B) = \{b\}$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Regras para determinar o Conjunto Primeiros
 - Se X é terminal ou ε , então $\text{Primeiro}(X) = \{X\}$
 - Se X é não terminal e $X \rightarrow a\alpha$ é uma produção então $a \in \text{Primeiro}(X)$
 - Se X é não terminal e $X \rightarrow \varepsilon$ é uma produção então $\varepsilon \in \text{Primeiro}(X)$
 - Se $X \rightarrow Y_1 Y_2 Y_3 \dots Y_k$ é uma produção, então para todo i tal que todos $Y_1 \dots Y_{i-1}$ são não terminais e $\varepsilon \in \text{Primeiro}(Y_j)$, para $j = 1, 2, \dots, i-1$, acrescenta-se todo símbolo diferente de ε de $\text{Primeiro}(Y_i)$ em $\text{Primeiro}(X)$. Se $\varepsilon \in \text{Primeiro}(Y_i)$ para todo $i = 1, 2, \dots, k$, então $\varepsilon \in \text{Primeiro}(X)$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Conjunto Primeiros

Gramática

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$
 $\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{termo} \rangle \langle \text{exp}' \rangle \mid \varepsilon$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$
 $\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{fator} \rangle \langle \text{termo}' \rangle \mid \varepsilon$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Primeiros

$\text{Primeiro}(\text{exp}) = \{ (, \text{número} \}$
 $\text{Primeiro}(\text{exp}') = \{ +, -, \varepsilon \}$
 $\text{Primeiro}(\text{soma}) = \{ +, - \}$
 $\text{Primeiro}(\text{termo}) = \{ (, \text{número} \}$
 $\text{Primeiro}(\text{termo}') = \{ *, \varepsilon \}$
 $\text{Primeiro}(\text{mult}) = \{ * \}$
 $\text{Primeiro}(\text{fator}) = \{ (, \text{número} \}$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Conjunto Primeiros

Gramática

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$
 $\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{termo} \rangle \langle \text{exp}' \rangle \mid \varepsilon$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$
 $\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{fator} \rangle \langle \text{termo}' \rangle \mid \varepsilon$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Primeiros

$\text{Primeiro}(\text{exp}) = \{ (, \text{número} \}$
 $\text{Primeiro}(\text{exp}') = \{ +, -, \varepsilon \}$
 $\text{Primeiro}(\text{soma}) = \{ +, - \}$
 $\text{Primeiro}(\text{termo}) = \{ (, \text{número} \}$
 $\text{Primeiro}(\text{termo}') = \{ *, \varepsilon \}$
 $\text{Primeiro}(\text{mult}) = \{ * \}$
 $\text{Primeiro}(\text{fator}) = \{ (, \text{número} \}$

E se nessa gramática tivéssemos a regra

$\langle \text{novo} \rangle ::= \langle \text{exp}' \rangle \langle \text{termo} \rangle$

$\text{Primeiro}(\text{novo}) = \{ \text{Primeiro}(\text{exp}') - \varepsilon \} \cup \text{Primeiro}(\text{termo}) = \{ +, -, (, \text{número} \}$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Conjunto Primeiros

Gramática

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$
 $\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{termo} \rangle \langle \text{exp}' \rangle \mid \varepsilon$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$
 $\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{fator} \rangle \langle \text{termo}' \rangle \mid \varepsilon$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Primeiros

$\text{Primeiro}(\text{exp}) = \{ (, \text{número} \}$
 $\text{Primeiro}(\text{exp}') = \{ +, -, \varepsilon \}$
 $\text{Primeiro}(\text{soma}) = \{ +, - \}$
 $\text{Primeiro}(\text{termo}) = \{ (, \text{número} \}$
 $\text{Primeiro}(\text{termo}') = \{ *, \varepsilon \}$
 $\text{Primeiro}(\text{mult}) = \{ * \}$
 $\text{Primeiro}(\text{fator}) = \{ (, \text{número} \}$

E se essa regra fosse

$\langle \text{novo} \rangle ::= \langle \text{exp}' \rangle \langle \text{termo}' \rangle$

$\text{Primeiro}(\text{novo}) = \{ +, -, *, \varepsilon \}$

→ pois, ε pertence tanto a $\text{Primeiro}(\text{exp}')$ quanto a $\text{Primeiro}(\text{termo}')$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Conjunto Seguidores
 - Composto por **terminais** e possivelmente **\$**
 - Conjunto de terminais que aparecem imediatamente à direita de um símbolo da gramática

Gramática

$\langle S \rangle ::= \langle A \rangle \langle B \rangle$

$\langle A \rangle ::= a \langle A \rangle \mid a$

$\langle B \rangle ::= b \langle B \rangle \mid b$

Seguidores

$\text{Seguidor}(S) = \{\$ \}$

$\text{Seguidor}(A) = \{b\}$

$\text{Seguidor}(B) = \{\$ \}$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Regras para determinar o Conjunto Seguidores
 - Se A for um símbolo inicial então $\$ \in \text{Seguidor}(A)$
 - Se existe uma produção $A \rightarrow \alpha B \beta$, e $\beta \neq \epsilon$ então $\{\text{Primeiro}(\beta) - \epsilon\} \subset \text{Seguidor}(B)$
 - Se existe uma produção $A \rightarrow \alpha B$ ou $A \rightarrow \alpha B \beta$ e $\epsilon \in \text{Primeiro}(\beta)$ então $\text{Seguidor}(A) \subset \text{Seguidor}(B)$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Conjunto Seguidores

Gramática

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$
 $\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{termo} \rangle \langle \text{exp}' \rangle \mid \varepsilon$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$
 $\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{fator} \rangle \langle \text{termo}' \rangle \mid \varepsilon$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Seguidores

$\text{Seguidor}(\text{exp}) = \{ \$,) \}$
 $\text{Seguidor}(\text{exp}') = \{ \$,) \}$
 $\text{Seguidor}(\text{soma}) = \{ (, \text{número} \}$
 $\text{Seguidor}(\text{termo}) = \{ +, -, \$,) \}$
 $\text{Seguidor}(\text{termo}') = \{ +, -, \$,) \}$
 $\text{Seguidor}(\text{mult}) = \{ (, \text{número} \}$
 $\text{Seguidor}(\text{fator}) = \{ *, +, -, \$,) \}$

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Exercício
 - Construa os Conjuntos Primeiros e Seguidores em
$$\begin{aligned}\langle \text{exp} \rangle &::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle \\ \langle \text{soma} \rangle &::= + \mid - \\ \langle \text{termo} \rangle &::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle \\ \langle \text{mult} \rangle &::= * \\ \langle \text{fator} \rangle &::= (\langle \text{exp} \rangle) \mid \text{número}\end{aligned}$$

Primeiro(exp) = { (, número }
Primeiro(termo) = { (, número }
Primeiro(fator) = { (, número }
Primeiro(soma) = { +, - }
Primeiro(mult) = { * }

Seguidor(exp) = { \$, +, -,) }
Seguidor(termo) = { \$, +, -, *,) }
Seguidor(fator) = { \$, +, -, *,) }
Seguidor(soma) = { (, número }
Seguidor(mult) = { (, número }

Análise Sintática

- Conjuntos Primeiros e Seguidores
 - Exercício
 - Construa os Conjuntos Primeiros e Seguidores em
$$\begin{aligned} \langle \text{declaração} \rangle &::= \langle \text{if-decl} \rangle \mid \text{outra} \\ \langle \text{if-decl} \rangle &::= \text{if } (\langle \text{exp} \rangle) \langle \text{declaração} \rangle \langle \text{else-parte} \rangle \\ \langle \text{else-parte} \rangle &::= \text{else } \langle \text{declaração} \rangle \mid \varepsilon \\ \langle \text{exp} \rangle &::= 0 \mid 1 \end{aligned}$$

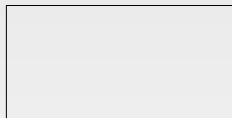
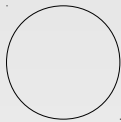
Primeiro(declaração) = { if, outra }
Primeiro(if-decl) = { if }
Primeiro(else-parte) = { else, ε }
Primeiro(exp) = { 0, 1 }

Seguidor(declaração) = { \$, else }
Seguidor(if-decl) = { \$, else }
Seguidor(else-parte) = { \$, else }
Seguidor(exp) = {) }

Análise Sintática

- Grafos (ou diagramas) sintáticos
 - Representações gráficas das regras de produção
 - Notação básica

Símbolo



Significado

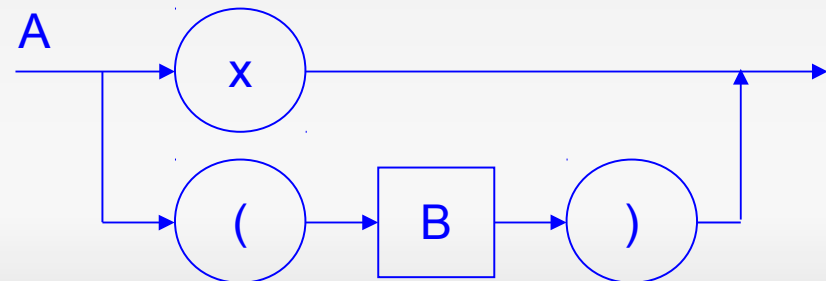
Terminal

Não-terminal

Fluxo

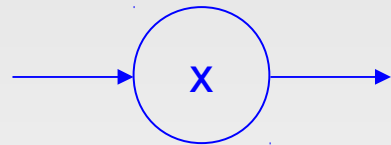
Exemplo

$\langle A \rangle ::= x \mid (\langle B \rangle)$



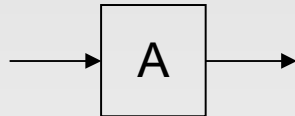
Análise Sintática

- Grafos sintáticos
 - Terminal
 - A ocorrência de um terminal x em uma regra corresponde
 - Ao seu reconhecimento na cadeia de entrada e
 - À leitura do próximo símbolo dessa cadeia

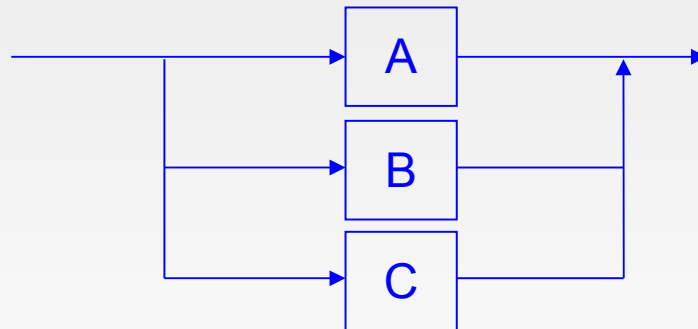


Análise Sintática

- Grafos sintáticos
 - Não terminal
 - A ocorrência de um não terminal A corresponde
 - À análise imediata de A

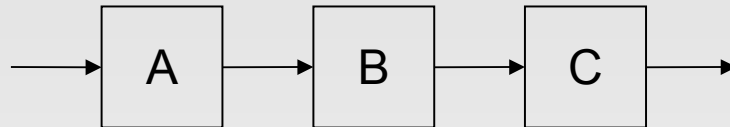


- Alternativas são representadas como

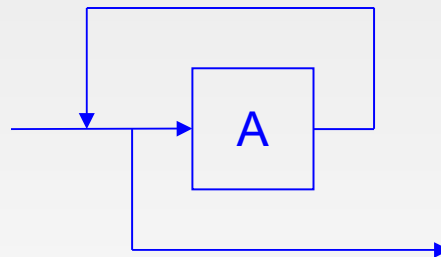


Análise Sintática

- Grafos sintáticos
 - Não terminal
 - Uma sequência A B C é mapeada em



- A forma $\{A\}^*$ ou A^* é representada por



Análise Sintática

- Grafos sintáticos

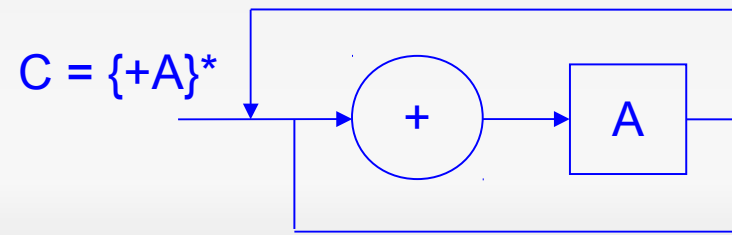
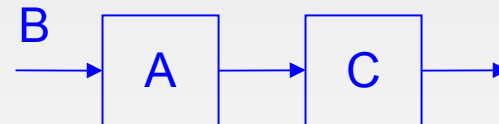
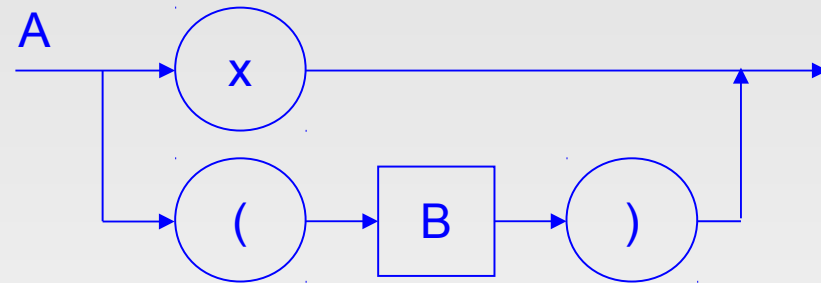
- Exercícios

1. Desenhe os grafos sintáticos para a gramática

$\langle A \rangle ::= x \mid (\langle B \rangle)$

$\langle B \rangle ::= \langle A \rangle \langle C \rangle$

$\langle C \rangle ::= +\langle A \rangle \langle C \rangle \mid \epsilon$

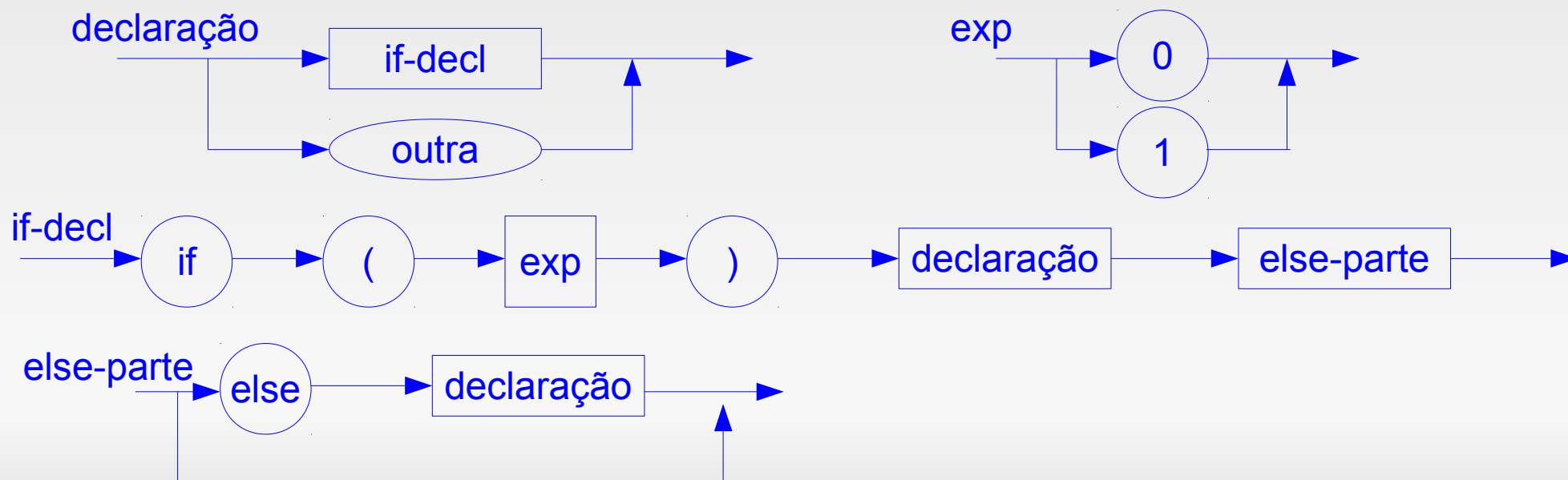


Análise Sintática

- Grafos sintáticos

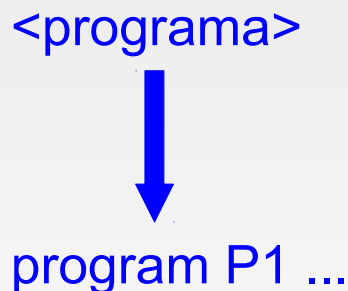
- Exercícios

2. Desenhe os grafos sintáticos para a gramática
 $\langle \text{declaração} \rangle ::= \langle \text{if-decl} \rangle \mid \text{outra}$
 $\langle \text{if-decl} \rangle ::= \text{if} (\langle \text{exp} \rangle) \langle \text{declaração} \rangle \langle \text{else-parte} \rangle$
 $\langle \text{else-parte} \rangle ::= \text{else} \langle \text{declaração} \rangle \mid \epsilon$
 $\langle \text{exp} \rangle ::= 0 \mid 1$



Análise Sintática

- Como é feita?
 - Há duas abordagens gerais para análise sintática
 - Descendente ou top-down
 - A análise é feita da raiz para as folhas
 - Parte-se do símbolo inicial da gramática e, por meio de derivações, chega-se à sequência de *tokens* retornada pelo analisador léxico



Análise Sintática

- Como é feita?
 - Há duas abordagens gerais para análise sintática
 - Ascendente ou bottom-up
 - A análise é feita das folhas para a raiz
 - Analisadores de empilha-reduz (*shift-reduce*)
 - Analisadores desse tipo são, geralmente, gerados por geradores automáticos

