TÉCNICO LISBOA

Compiladores[1]

Introdução à Análise Sintáctica. Análise Sintáctica Descendente

Capítulo 4.2-4.4 - "Compilers: Principles, Techniques and Tools"

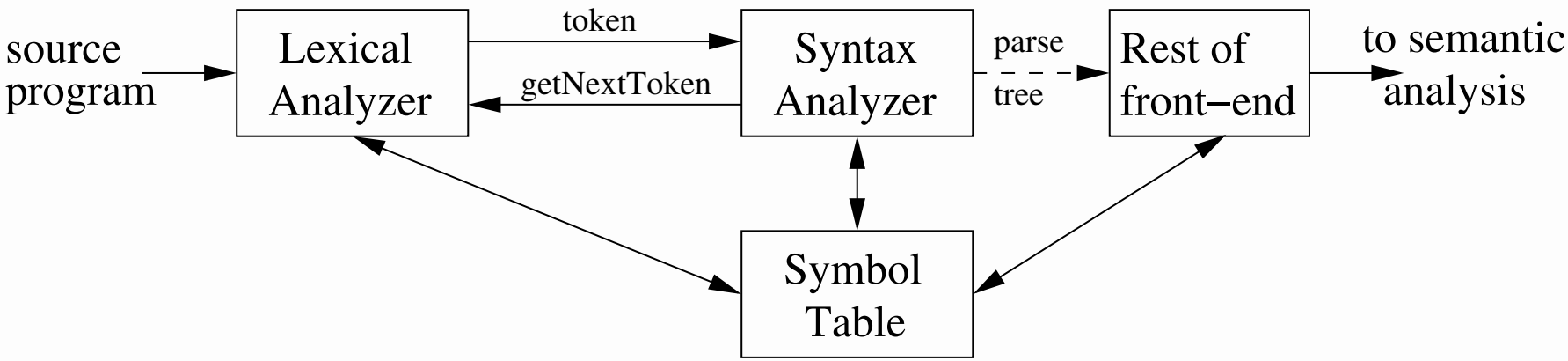
Prof. Alberto Abad

IST - Universidade de Lisboa

2021/2022

[1] Slides adaptados a partir do material do Prof. Pedro T. Monteiro (2017/2018)

Introdução à Análise Sintáctica



Para além da construção da árvore sintáctica, é tarefa do compilador:

- Identificar a presença de erros de forma clara e precisa
- Recuperar rapidamente de um erro para detectar potenciais erros seguintes
- Adicionar overhead mínimo no processamento de programas correctos

Para além da construção da árvore sintáctica, é tarefa do compilador:

- Identificar a presença de erros de forma clara e precisa
- Recuperar rapidamente de um erro para detectar potenciais erros seguintes
- Adicionar overhead mínimo no processamento de programas correctos

Estratégia simples - reportar a linha (e a posição) no programa onde o erro foi detectado.

Outline

- Gramática serve para produzir/derivar palavras a partir de outras palavras, **recursivamente**, começando no símbolo inicial:

$$S \rightarrow 0S1$$
$$S \rightarrow \epsilon$$

- As gramáticas são uma notação formal para representar **definições recursivas** de linguagens

Vantagens:

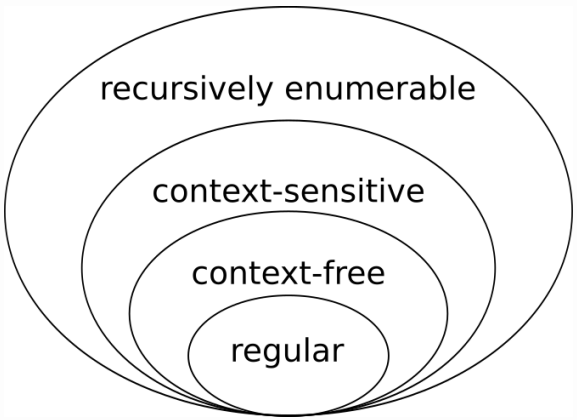
- Descrição precisa e fácil de compreender da sintaxe de uma linguagem
- Extensão e actualização fácil
- Possibilidade de gerar automaticamente um analisador sintáctico eficiente
- Relação directa com a estrutura da linguagem usada
- ...

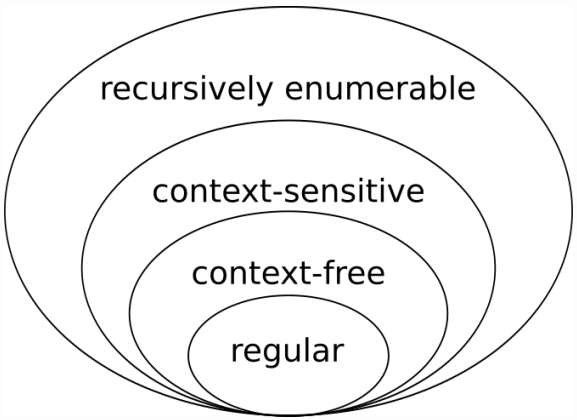
Backus-Naur Form (BNF) é uma notação para gramáticas livres de contexto com regras na forma

$$\text{Símbolo} \rightarrow \text{expressão1} \mid \ldots \mid \text{expressãoN}$$

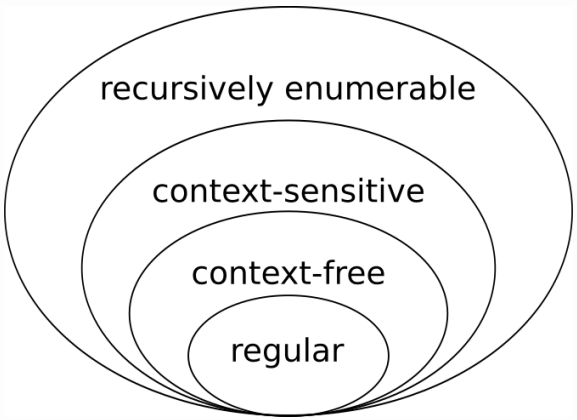
Gramáticas

Hierarquia de Chomsky

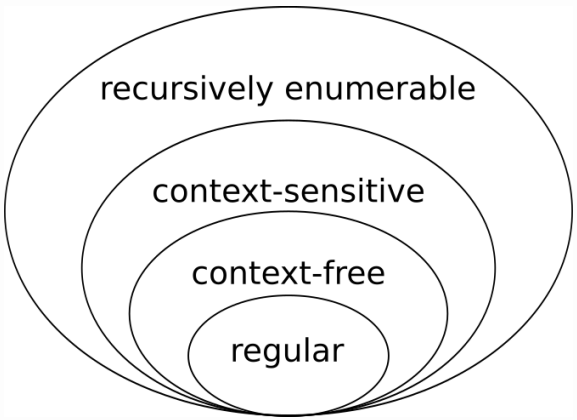




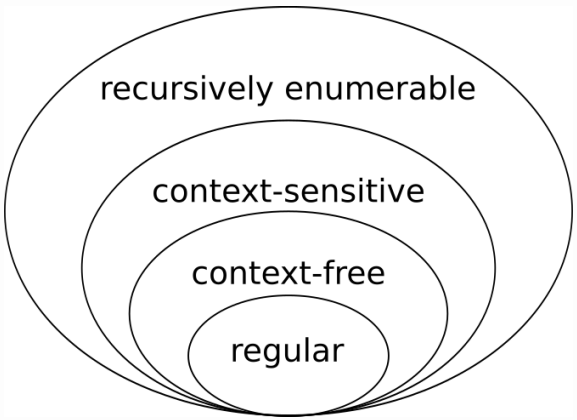
- **regulares** - representam linguagens regulares. Podem ser representada por expressões regulares ou um autómato finito. forma: $A \rightarrow aB$



- **regulares** - representam linguagens regulares. Podem ser representada por expressões regulares ou um autómato finito. forma: $A \rightarrow aB$
- **livres de contexto** - representam linguagens livres de contexto. Podem ser representadas por um autómato de pilha forma: $A \rightarrow \gamma$



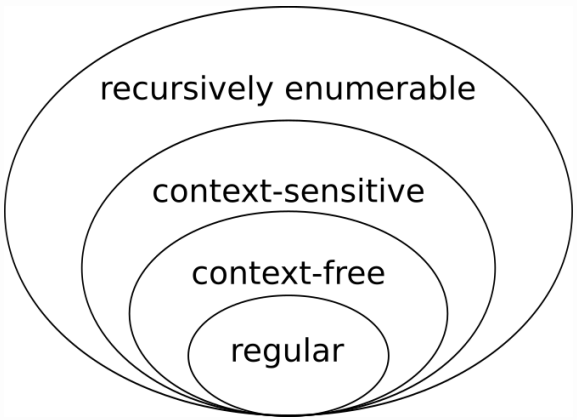
- **regulares** - representam linguagens regulares. Podem ser representada por expressões regulares ou um autómato finito. forma: $A \rightarrow aB$
- **livres de contexto** - representam linguagens livres de contexto. Podem ser representadas por um autómato de pilha forma: $A \rightarrow \gamma$
- outras ...
  - expressividade
  - facilidade fazer parsing



- **regulares** - representam linguagens regulares. Podem ser representada por expressões regulares ou um autómato finito. forma: $A \rightarrow aB$
- **livres de contexto** - representam linguagens livres de contexto. Podem ser representadas por um autómato de pilha forma: $A \rightarrow \gamma$
- outras ...
  - expressividade
  - facilidade fazer parsing

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

- L não é regular. Porquê?



- **regulares** - representam linguagens regulares. Podem ser representada por expressões regulares ou um autómato finito. forma: $A \rightarrow aB$
- **livres de contexto** - representam linguagens livres de contexto. Podem ser representadas por um autómato de pilha forma: $A \rightarrow \gamma$
- outras ...
  - expressividade
  - facilidade fazer parsing

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

- L não é regular. Porquê?
- DFA não permite memorizar o número de 0's

Uma **gramática livre de contexto** $G = (V, \Sigma, R, S)$ é um tuplo onde:

- V é o conjunto de **símbolos não terminais**
- Σ é o conjunto de **símbolos terminais**
- R é o conjunto finito de **produções** ou **regras**, com a forma $X \to Y_1 Y_2 \ldots Y_n$, onde:
  - X é o símbolo não terminal a ser definido
  - $\to$ é o símbolo de produção
  - $Y_1 Y_2 \cdots Y_n$ é o corpo da produção
- $S \in V$ é o **símbolo inicial**

Hopcroft and Ullman, 1979

Gramática para a definição de expressões aritméticas. **Exemplo**:

- $V = \{expr, op\}$
- $\Sigma = \{id, +, -, *, /, (,)\}$
- Regras:

$$
\begin{array}{rcl}
expr & \rightarrow & expr\ op\ expr \\
expr & \rightarrow & (\ expr\ ) \\
expr & \rightarrow & \text{-}\ expr \\
expr & \rightarrow & id \\
op & \rightarrow & + \mid \text{-} \mid * \mid /
\end{array}
$$

- $S = expr$

Considere uma **gramática livre de contexto** $G = (V, \Sigma, R, S)$:

Considere uma **gramática livre de contexto** $G = (V, \Sigma, R, S)$:

- **Passo de Derivação**:
  - Palavra actual: $\alpha A \beta$
  - Seja $A \to \gamma$ uma produção de G
  - $\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$ representa um passo de derivação da palavra $\alpha \gamma \beta$ a partir $\alpha A \beta$

Considere uma **gramática livre de contexto** $G = (V, \Sigma, R, S)$:

- **Passo de Derivação**:
  - Palavra actual: $\alpha A \beta$
  - Seja $A \to \gamma$ uma produção de G
  - $\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$ representa um passo de derivação da palavra $\alpha \gamma \beta$ a partir $\alpha A \beta$
- **Derivações**: * representa 0 ou mais passos de derivação:
  - **Caso base**: Para qualquer palavra α de terminais e não terminais, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$
  - **Indução**: Se $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$ e $\beta \underset{G}{\Rightarrow} \gamma$, então $\alpha \overset{*}{\underset{G}{\Rightarrow}} \gamma$

Considere uma **gramática livre de contexto** $G = (V, \Sigma, R, S)$:

- **Passo de Derivação**:
  - Palavra actual: $\alpha A \beta$
  - Seja $A \to \gamma$ uma produção de G
  - $\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$ representa um passo de derivação da palavra $\alpha \gamma \beta$ a partir $\alpha A \beta$
- **Derivações**: * representa 0 ou mais passos de derivação:
  - **Caso base**: Para qualquer palavra α de terminais e não terminais, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$
  - **Indução**: Se $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$ e $\beta \underset{G}{\Rightarrow} \gamma$, então $\alpha \overset{*}{\underset{G}{\Rightarrow}} \gamma$
- Nota: pode utilizar-se $\Rightarrow$ e $\overset{*}{\Rightarrow}$ como alternativa a $\underset{G}{\Rightarrow}$ e $\overset{*}{\underset{G}{\Rightarrow}}$, respectivamente

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$$S \stackrel{*}{\Rightarrow} 11011 \qquad \text{Sim}$$

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$$S \stackrel{*}{\Rightarrow} 11011 \qquad \text{Sim}$$

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$S \stackrel{*}{\Rightarrow} 11011$ Sim

$S \stackrel{*}{\Rightarrow} 1001$

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$$S \stackrel{*}{\Rightarrow} 11011 \qquad \text{Sim}$$
$$S \stackrel{*}{\Rightarrow} 1001 \qquad \text{Sim}$$

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$S \stackrel{*}{\Rightarrow} 11011$ Sim

$S \stackrel{*}{\Rightarrow} 1001$ Sim

$S \stackrel{*}{\Rightarrow} 1011$

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$S \stackrel{*}{\Rightarrow} 11011$	Sim
$S \stackrel{*}{\Rightarrow} 1001$	Sim
$S \stackrel{*}{\Rightarrow} 1011$	Não

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$S \stackrel{*}{\Rightarrow} 11011$	Sim
$S \stackrel{*}{\Rightarrow} 1001$	Sim
$S \stackrel{*}{\Rightarrow} 1011$	Não
$S \stackrel{*}{\Rightarrow} 010010$	

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$S \stackrel{*}{\Rightarrow} 11011$	Sim
$S \stackrel{*}{\Rightarrow} 1001$	Sim
$S \stackrel{*}{\Rightarrow} 1011$	Não
$S \stackrel{*}{\Rightarrow} 010010$	Sim

Considere a **gramática livre de contexto** G:

$$S \rightarrow 1S1 \,|\, 0S0 \,|\, 1 \,|\, 0 \,|\, \epsilon$$

- Quais as palavras que G gera?
- Exemplos:

$S \stackrel{*}{\Rightarrow} 11011$	Sim
$S \stackrel{*}{\Rightarrow} 1001$	Sim
$S \stackrel{*}{\Rightarrow} 1011$	Não
$S \stackrel{*}{\Rightarrow} 010010$	Sim

- Resposta: G gera palíndromos sobre o alfabeto $\{0, 1\}$

Dada uma **gramática livre de contexto** $G = (N, \Sigma, R, S)$:

- linguagem de G, L(G) é o conjunto de palavras com derivações a partir do símbolo inicial

$$L(G) = \{w \in \Sigma^* \mid S \overset{*}{\underset{G}{\Rightarrow}} w\}$$

- Derivação à esquerda:
  Substituir símbolo não terminal mais à esquerda a cada passo de derivação

- Derivação à esquerda:
  Substituir símbolo não terminal mais à esquerda a cada passo de derivação
  - Notação: $\underset{lm}{\Longrightarrow}$ e $\overset{*}{\underset{lm}{\Longrightarrow}}$

- Derivação à esquerda:
  Substituir símbolo não terminal mais à esquerda a cada passo de derivação
  - Notação: $\underset{lm}{\Longrightarrow}$ e $\overset{*}{\underset{lm}{\Longrightarrow}}$
  - Exemplo. Derivação à esquerda da *string* $id * (-id)$:

$$\begin{aligned} expr &\underset{lm}{\Longrightarrow} expr\ op\ expr \underset{lm}{\Longrightarrow} id\ op\ expr \underset{lm}{\Longrightarrow} id * expr \\ &\underset{lm}{\Longrightarrow} id * (expr) \underset{lm}{\Longrightarrow} id * (-expr) \underset{lm}{\Longrightarrow} id * (-id) \end{aligned}$$

- Derivação à esquerda:
  Substituir símbolo não terminal mais à esquerda a cada passo de derivação
  - Notação: $\underset{lm}{\Longrightarrow}$ e $\overset{*}{\underset{lm}{\Longrightarrow}}$
  - Exemplo. Derivação à esquerda da *string* $id * (-id)$:

$$\begin{aligned} expr &\underset{lm}{\Longrightarrow} expr\ op\ expr \underset{lm}{\Longrightarrow} id\ op\ expr \underset{lm}{\Longrightarrow} id * expr \\ &\underset{lm}{\Longrightarrow} id * (expr) \underset{lm}{\Longrightarrow} id * (-expr) \underset{lm}{\Longrightarrow} id * (-id) \end{aligned}$$

- Derivação à direita:
  Substituir símbolo não terminal mais à direita a cada passo de derivação

- Derivação à esquerda:
  Substituir símbolo não terminal mais à esquerda a cada passo de derivação
  - Notação: $\underset{lm}{\Longrightarrow}$ e $\overset{*}{\underset{lm}{\Longrightarrow}}$
  - Exemplo. Derivação à esquerda da *string* $id * (-id)$:

$$\begin{aligned} expr &\underset{lm}{\Longrightarrow} expr\ op\ expr \underset{lm}{\Longrightarrow} id\ op\ expr \underset{lm}{\Longrightarrow} id * expr \\ &\underset{lm}{\Longrightarrow} id * (expr) \underset{lm}{\Longrightarrow} id * (-expr) \underset{lm}{\Longrightarrow} id * (-id) \end{aligned}$$

- Derivação à direita:
  Substituir símbolo não terminal mais à direita a cada passo de derivação
  - Notação: $\underset{rm}{\Longrightarrow}$ e $\overset{*}{\underset{rm}{\Longrightarrow}}$

- Derivação à esquerda:
  Substituir símbolo não terminal mais à esquerda a cada passo de derivação
  - Notação: $\underset{lm}{\Longrightarrow}$ e $\overset{*}{\underset{lm}{\Longrightarrow}}$
  - Exemplo. Derivação à esquerda da *string* $id * (-id)$:

$$\begin{aligned} expr &\underset{lm}{\Longrightarrow} expr\ op\ expr \underset{lm}{\Longrightarrow} id\ op\ expr \underset{lm}{\Longrightarrow} id * expr \\ &\underset{lm}{\Longrightarrow} id * (expr) \underset{lm}{\Longrightarrow} id * (-expr) \underset{lm}{\Longrightarrow} id * (-id) \end{aligned}$$

- Derivação à direita:
  Substituir símbolo não terminal mais à direita a cada passo de derivação
  - Notação: $\underset{rm}{\Longrightarrow}$ e $\overset{*}{\underset{rm}{\Longrightarrow}}$
  - Exemplo. Derivação à direita da *string* $id * (-id)$:

$$\begin{aligned} expr &\underset{rm}{\Longrightarrow} expr\ op\ expr \underset{rm}{\Longrightarrow} expr\ op\ (expr) \underset{rm}{\Longrightarrow} expr\ op\ (-expr) \\ &\underset{rm}{\Longrightarrow} expr\ op\ (-id) \underset{rm}{\Longrightarrow} expr\ *(-id) \underset{rm}{\Longrightarrow} id * (-id) \end{aligned}$$

Árvore sintáctica

Definição

Uma árvore sintáctica (ou de parsing) é uma representação gráfica de uma derivação.

- A ordem em que as produções são aplicadas não é representada

Árvore sintáctica

Definição

Uma árvore sintáctica (ou de parsing) é uma representação gráfica de uma derivação.

- A ordem em que as produções são aplicadas não é representada

Árvore de sintáctica para $G = (V, \Sigma, R, S)$:

- Cada nó interior etiquetado com um símbolo não terminal de V
- Cada nó folha etiquetado com um símbolo terminal, não terminal, ou ϵ
  - Se um nó folha é etiquetado com ϵ, então é o único filho do nó pai
- Se nó interior é etiquetado com A e os filhos são etiquetados com $X_1, X_2, \ldots, X_k$ (a partir da esquerda), então $A \to X_1 X_2 \ldots X_k$ é uma produção de G

Árvore sintáctica

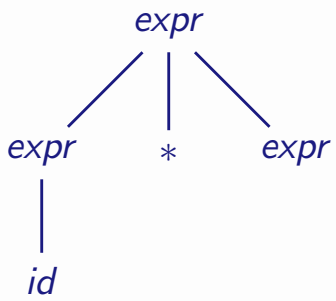
Exemplo

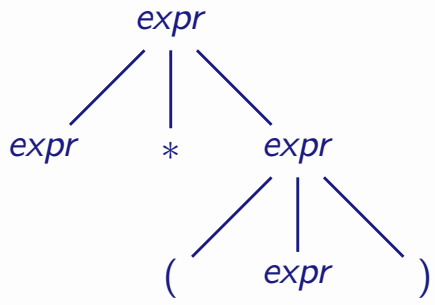
- Derivação de $id * (-id)$ a partir de *expr*

Árvore sintáctica

Exemplo

- Derivação de $id * (-id)$ a partir de *expr*

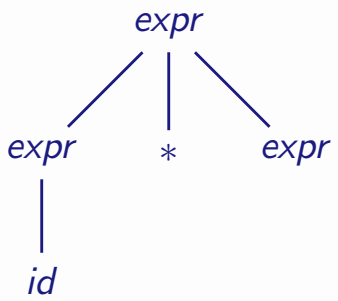


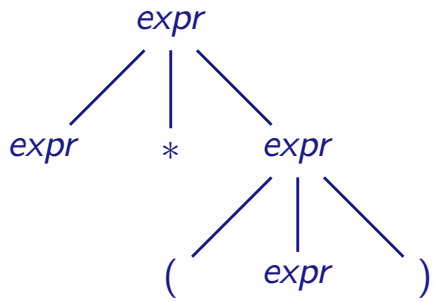


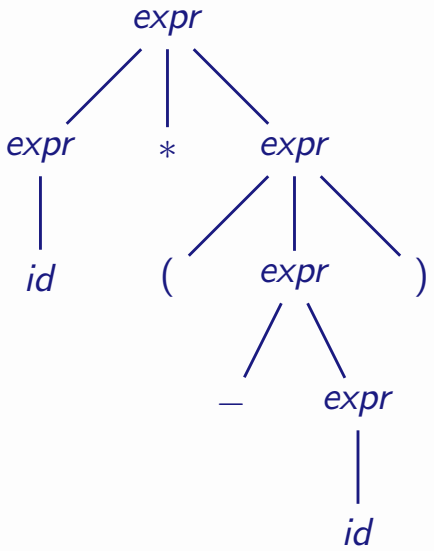
Árvore sintáctica

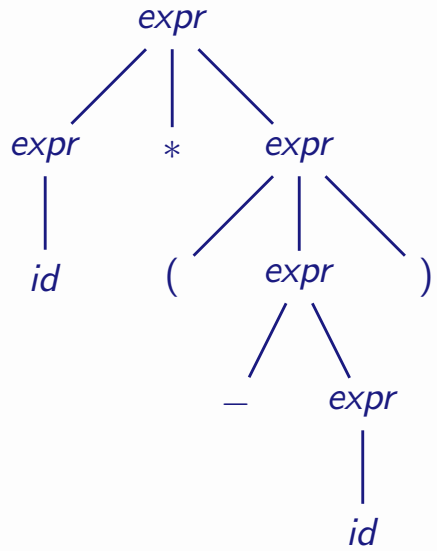
Exemplo

- Derivação de $id * (-id)$ a partir de *expr*





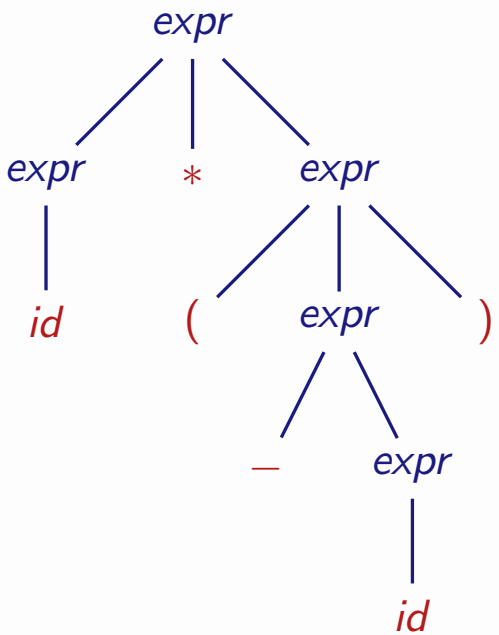




Resultado: nós folha da árvore sintáctica concatenados da esquerda para a direita

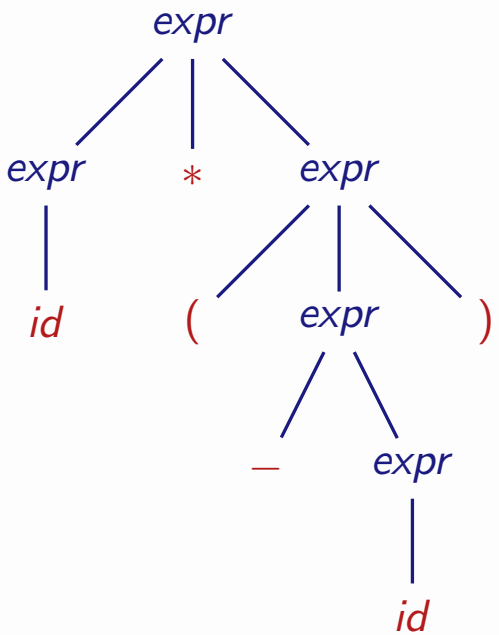
Resultado: nós folha da árvore sintáctica concatenados da esquerda para a direita

Exemplo:



Resultado: nós folha da árvore sintáctica concatenados da esquerda para a direita

Exemplo:



Resultado é: id * (- id)

Propriedades:

- Ambiguidade
- Precedência
- Associatividade

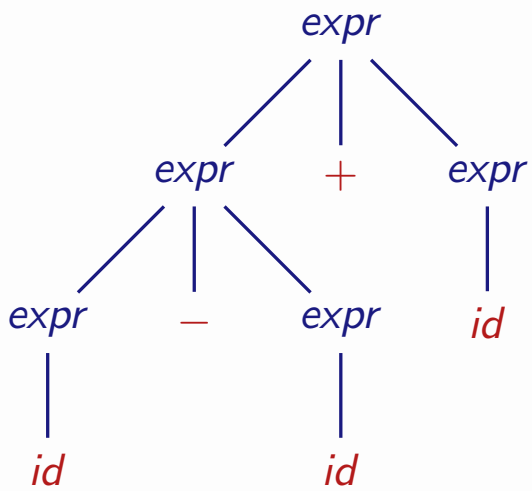
Uma gramática $G = V, \Sigma, R, S)$ diz-se ambígua se existe pelo menos uma palavra em Σ^* que tem duas árvores sintácticas diferentes, ambas com raíz S

Uma gramática $G = V, \Sigma, R, S)$ diz-se ambígua se existe pelo menos uma palavra em Σ^* que tem duas árvores sintácticas diferentes, ambas com raíz S

Exemplo: id - id + id

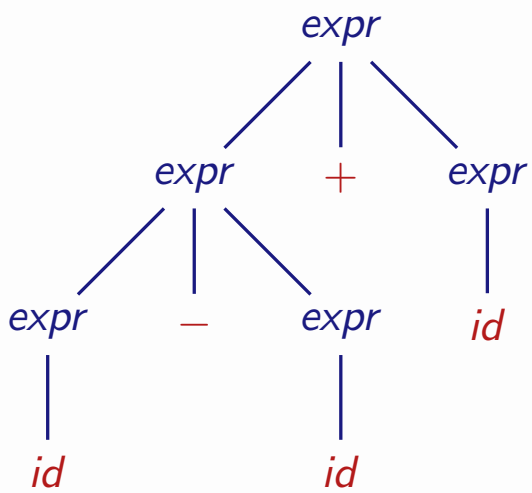
Uma gramática $G = V, \Sigma, R, S)$ diz-se ambígua se existe pelo menos uma palavra em Σ^* que tem duas árvores sintácticas diferentes, ambas com raíz S

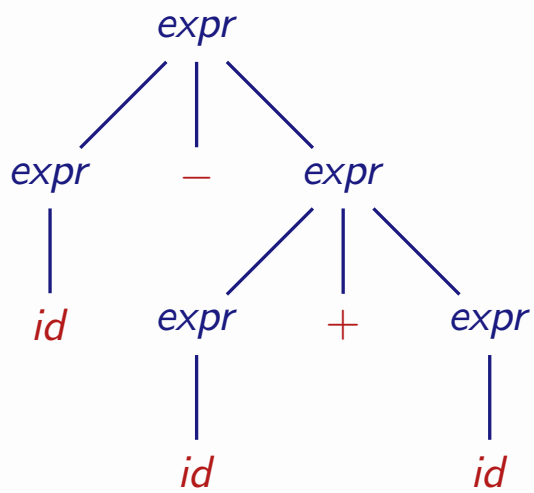
Exemplo: id - id + id



Uma gramática $G = V, \Sigma, R, S)$ diz-se ambígua se existe pelo menos uma palavra em Σ^* que tem duas árvores sintácticas diferentes, ambas com raíz S

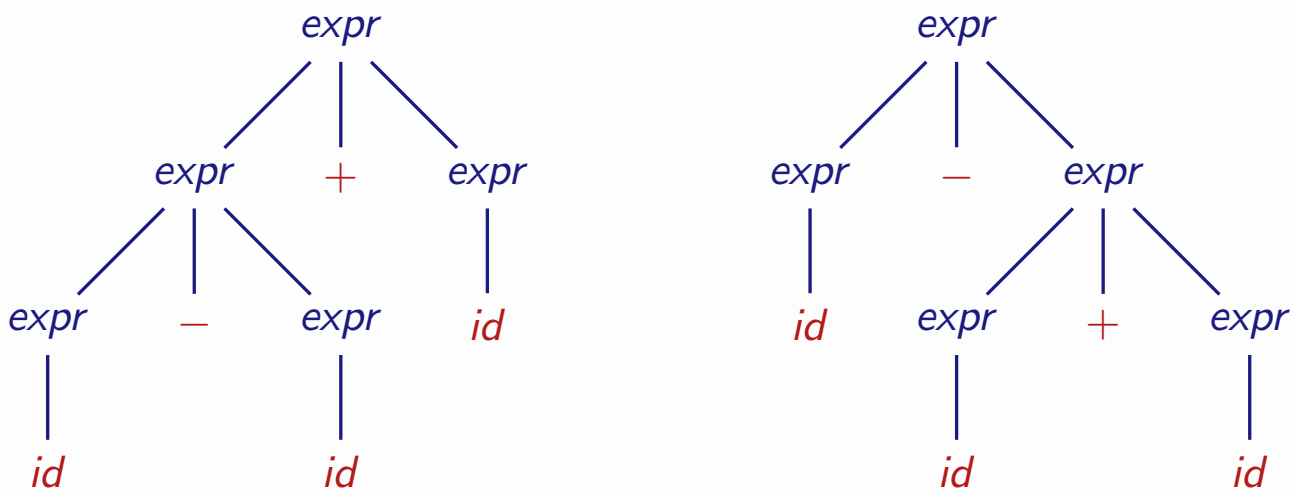
Exemplo: id - id + id





Uma gramática $G = V, \Sigma, R, S)$ diz-se ambígua se existe pelo menos uma palavra em Σ^* que tem duas árvores sintácticas diferentes, ambas com raíz S

Exemplo: id - id + id



Nota: Não é possível processar gramáticas ambíguas de forma eficiente!

Por vezes é necessário transformar a gramática de forma a remover as ambiguidades.

A associatividade é uma propriedade de operadores binários.

$$\forall x, y, z \;\; (x \Delta y) \Delta z = x \Delta (y \Delta z)$$

Mas nem todos os operadores binários são associativos.

- e.g. subtracção não é associativa $(5 - 2) - 1 \neq 5 - (2 - 1)$

A associatividade é uma propriedade de operadores binários.

$$\forall x, y, z \quad (x \Delta y) \Delta z = x \Delta (y \Delta z)$$

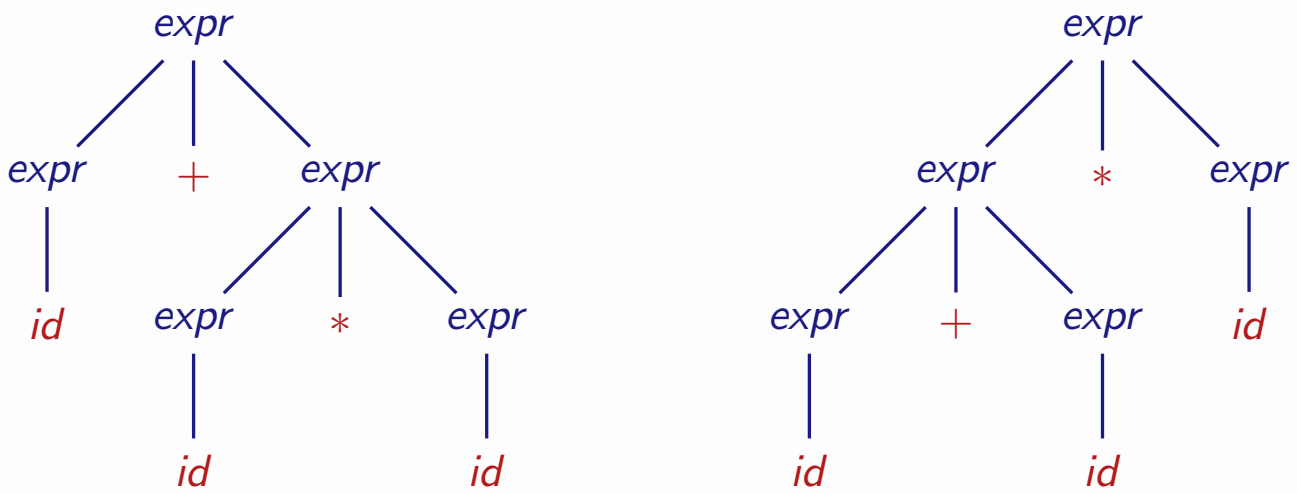
Mas nem todos os operadores binários são associativos.

- e.g. subtracção não é associativa $(5 - 2) - 1 \neq 5 - (2 - 1)$

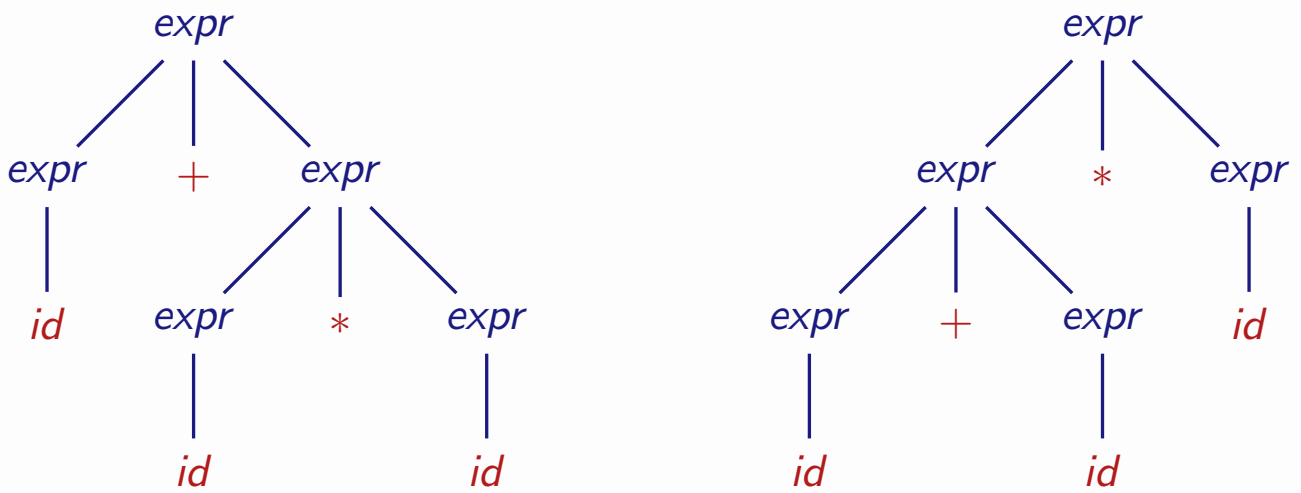
Solução: usar um símbolo diferente à esquerda e à direita do operador

$$\begin{array}{l} E \rightarrow E \ - \ E \\ E \rightarrow E \ + \ E \\ E \rightarrow id \\ E \rightarrow (E) \end{array} \quad \Rightarrow \quad \begin{array}{l} E \rightarrow E \ - \ F \\ E \rightarrow E \ + \ F \\ E \rightarrow F \\ F \rightarrow id \\ F \rightarrow (E) \end{array}$$

Muitas linguagens têm regras de precedência entre operadores.
Exemplo (árvores sintácticas anteriores):



Muitas linguagens têm regras de precedência entre operadores.
Exemplo (árvores sintácticas anteriores):



$E \rightarrow E + T \mid T$
$E \rightarrow E - T \mid T$
$T \rightarrow T * F \mid F$
$T \rightarrow T / F \mid F$
$F \rightarrow id$
$F \rightarrow (E)$

As precedências podem ser ditadas pela estrutura das regras! afectando a profundidade a que os operadores ficam na árvore.
Quanto maior a precedência, mais profundos ficam!

Uma gramática diz-se recursiva à esquerda se tem pelo menos um símbolo não terminal A com uma derivação contendo A como o símbolo mais à esquerda.

- e.g., $E \rightarrow E + T$

Uma gramática diz-se recursiva à direita se tem pelo menos um símbolo não terminal A com uma derivação contendo A como o símbolo mais à direita.

- e.g., $E \rightarrow T + E$

Outline

Conjuntos de análise

Os algoritmos de análise sintáctica usam conjuntos de símbolos terminais que ajudam a determinar as regras da gramática.

Conjuntos de análise

Os algoritmos de análise sintáctica usam conjuntos de símbolos terminais que ajudam a determinar as regras da gramática.

- **FIRST** - conjunto de símbolos terminais que iniciam sequências derivadas a partir de X

Conjuntos de análise

Os algoritmos de análise sintáctica usam conjuntos de símbolos terminais que ajudam a determinar as regras da gramática.

- **FIRST** - conjunto de símbolos terminais que iniciam sequências derivadas a partir de X
- **FOLLOW** - conjunto de símbolos terminais a que podem aparecer imediatamente a seguir a X

Conjuntos de análise

Os algoritmos de análise sintáctica usam conjuntos de símbolos terminais que ajudam a determinar as regras da gramática.

- **FIRST** - conjunto de símbolos terminais que iniciam sequências derivadas a partir de X
- **FOLLOW** - conjunto de símbolos terminais a que podem aparecer imediatamente a seguir a X
- **LOOKAHEAD** - conjunto de antevisão

FIRST(X) - conjunto de símbolos terminais que iniciam sequências derivadas a partir de X

Algoritmo:

- se α é um símbolo terminal, então $FIRST(\alpha) = \{\alpha\}$
- se X é um símbolo não terminal:
  - se $X \rightarrow \epsilon$ é uma produção, então $\epsilon \in FIRST(X)$
  - se $X \rightarrow Y_1 \ldots Y_n$ é uma produção, então:
    - se $\epsilon \notin FIRST(Y_1)$, $FIRST(Y_1) \subseteq FIRST(X)$
    - se $\epsilon \in FIRST(Y_1)$, $FIRST(Y_1) \setminus \{\epsilon\} \cup FIRST(Y_2 \ldots Y_n) \subseteq FIRST(X)$
    - se $\epsilon \in FIRST(Y_1) \wedge \ldots \wedge \epsilon \in FIRST(Y_n)$, $\epsilon \in FIRST(X)$

Conjuntos de análise

FIRST - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

Conjuntos de análise

FIRST - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\forall_{\alpha \in \Sigma}, \mathsf{FIRST}(\alpha) = \{\alpha\}$, $\Sigma = \{(,), +, *, \mathsf{id}\}$

Conjuntos de análise

FIRST - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\forall_{\alpha \in \Sigma}, \mathsf{FIRST}(\alpha) = \{\alpha\}$, $\Sigma = \{(,), +, *, \mathsf{id}\}$
- $\mathsf{FIRST}(F) = \{(, \mathsf{id}\}$

Conjuntos de análise

FIRST - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\forall_{\alpha \in \Sigma}, \mathsf{FIRST}(\alpha) = \{\alpha\}$, $\Sigma = \{(,), +, *, \mathsf{id}\}$
- $\mathsf{FIRST}(F) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(T) = \{(, \mathsf{id}\}$

Conjuntos de análise

FIRST - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\forall_{\alpha \in \Sigma}, \mathsf{FIRST}(\alpha) = \{\alpha\}$, $\Sigma = \{(,), +, *, \mathsf{id}\}$
- $\mathsf{FIRST}(F) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(T) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(E) = \{(, \mathsf{id}\}$

Conjuntos de análise

FIRST - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\forall_{\alpha \in \Sigma}, \mathsf{FIRST}(\alpha) = \{\alpha\}$, $\Sigma = \{(,), +, *, \mathsf{id}\}$
- $\mathsf{FIRST}(F) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(T) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(E) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(E') = \{+, \epsilon\}$

Conjuntos de análise

FIRST - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\forall_{\alpha \in \Sigma}, \mathsf{FIRST}(\alpha) = \{\alpha\}$, $\Sigma = \{(,), +, *, \mathsf{id}\}$
- $\mathsf{FIRST}(F) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(T) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(E) = \{(, \mathsf{id}\}$
- $\mathsf{FIRST}(E') = \{+, \epsilon\}$
- $\mathsf{FIRST}(T') = \{*, \epsilon\}$

FOLLOW - conjunto de símbolos terminais *a* que podem aparecer imediatamente a seguir a X. Calculado apenas para símbolos não terminais.

Algoritmo:

- se X é o símbolo inicial, então $\$ \in FOLLOW(X)$
- se $A \rightarrow \alpha X \beta$ é uma produção, então $FIRST(\beta) \setminus \{\epsilon\} \subseteq FOLLOW(X)$
- se $A \rightarrow \alpha X$ ou $A \rightarrow \alpha X \beta$ ($\beta \stackrel{*}{\Rightarrow} \epsilon$), então $FOLLOW(A) \subseteq FOLLOW(X)$
- Repetir até que os conjuntos FOLLOW não sejam alterados

Conjuntos de análise

FOLLOW - Exemplo

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\mathrm{FOLLOW}(E) = \{\$\}$

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\mathsf{FOLLOW}(E) = \{\$\}$
- $\mathsf{FOLLOW}(T) = \mathsf{FIRST}(E') \setminus \{\epsilon\} = \{+\}$

Conjuntos de análise

FOLLOW - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\mathsf{FOLLOW}(E) = \{\$\}$
- $\mathsf{FOLLOW}(T) = \mathsf{FIRST}(E') \setminus \{\epsilon\} = \{+\}$
- $\mathsf{FOLLOW}(E) = \mathsf{FOLLOW}(E) \cup \mathsf{FIRST}()) = \{), \$\}$

Conjuntos de análise

FOLLOW - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

- $\mathsf{FOLLOW}(E) = \{\$\}$
- $\mathsf{FOLLOW}(T) = \mathsf{FIRST}(E') \setminus \{\epsilon\} = \{+\}$
- $\mathsf{FOLLOW}(E) = \mathsf{FOLLOW}(E) \cup \mathsf{FIRST}()) = \{), \$\}$
- $\mathsf{FOLLOW}(E') = \mathsf{FOLLOW}(E) = \{), \$\}$

Conjuntos de análise

FOLLOW - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

- $\mathrm{FOLLOW}(E) = \{\$\}$
- $\mathrm{FOLLOW}(T) = \mathrm{FIRST}(E') \setminus \{\epsilon\} = \{+\}$
- $\mathrm{FOLLOW}(E) = \mathrm{FOLLOW}(E) \cup \mathrm{FIRST}()) = \{), \$\}$
- $\mathrm{FOLLOW}(E') = \mathrm{FOLLOW}(E) = \{), \$\}$
- $\mathrm{FOLLOW}(T) = \mathrm{FIRST}(E') \setminus \{\epsilon\} \cup \mathrm{FOLLOW}(E) \cup \mathrm{FOLLOW}(E') = \{+,), \$\}$

Conjuntos de análise

FOLLOW - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

- $\mathrm{FOLLOW}(E) = \{\$\}$
- $\mathrm{FOLLOW}(T) = \mathrm{FIRST}(E') \setminus \{\epsilon\} = \{+\}$
- $\mathrm{FOLLOW}(E) = \mathrm{FOLLOW}(E) \cup \mathrm{FIRST}()) = \{), \$\}$
- $\mathrm{FOLLOW}(E') = \mathrm{FOLLOW}(E) = \{), \$\}$
- $\mathrm{FOLLOW}(T) = \mathrm{FIRST}(E') \setminus \{\epsilon\} \cup \mathrm{FOLLOW}(E) \cup \mathrm{FOLLOW}(E') = \{+,), \$\}$
- $\mathrm{FOLLOW}(T') = \mathrm{FOLLOW}(T) = \{+,), \$\}$

Conjuntos de análise

FOLLOW - Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

- $\mathrm{FOLLOW}(E) = \{\$\}$
- $\mathrm{FOLLOW}(T) = \mathrm{FIRST}(E') \setminus \{\epsilon\} = \{+\}$
- $\mathrm{FOLLOW}(E) = \mathrm{FOLLOW}(E) \cup \mathrm{FIRST}()) = \{), \$\}$
- $\mathrm{FOLLOW}(E') = \mathrm{FOLLOW}(E) = \{), \$\}$
- $\mathrm{FOLLOW}(T) = \mathrm{FIRST}(E') \setminus \{\epsilon\} \cup \mathrm{FOLLOW}(E) \cup \mathrm{FOLLOW}(E') = \{+,), \$\}$
- $\mathrm{FOLLOW}(T') = \mathrm{FOLLOW}(T) = \{+,), \$\}$
- $\mathrm{FOLLOW}(F) = \mathrm{FIRST}(T') \setminus \{\epsilon\} \cup \mathrm{FOLLOW}(T) \cup \mathrm{FOLLOW}(T') = \{*, +,), \$\}$

LOOKAHEAD - conjunto de antevisão

Dada uma regra $A \rightarrow \alpha$, o $LOOKAHEAD(A \rightarrow \alpha)$ é:

- $FIRST(\alpha)$, se $\alpha \overset{*}{\not\Rightarrow} \epsilon$
- $FIRST(\alpha) \bigcup FOLLOW(A)$, se $\alpha \overset{*}{\Rightarrow} \epsilon$

Outline

Algoritmos de parsing

- **Objectivo**: decidir se $w \in L(G)$?

Algoritmos de parsing

- **Objectivo**: decidir se $w \in L(G)$?

- Universais
  - Algoritmo de Earley
  - Algoritmo de Cocke-Younger-Kasami (CYK)
    - Gramática representada em CNF (Chomsky Normal Form)
  - Ineficientes, não utilizados em compiladores

Algoritmos de parsing

- **Objectivo**: decidir se $w \in L(G)$?

- Universais
  - Algoritmo de Earley
  - Algoritmo de Cocke-Younger-Kasami (CYK)
    - Gramática representada em CNF (Chomsky Normal Form)
  - Ineficientes, não utilizados em compiladores
- Específicos para gramáticas LL(k)/LR(k)
  - Eficientes

Algoritmos de parsing

Gramáticas livres de contexto divididas em:

- **LL(k)** parsing
  - Top-down - análise descendente
  - Considera sempre derivações mais à esquerda

Algoritmos de parsing

Gramáticas livres de contexto divididas em:

- **LL(k)** parsing
  - Top-down - análise descendente
  - Considera sempre derivações mais à esquerda
- **LR(k)** parsing
  - Bottom-up - análise ascendente
  - Considera sempre derivações mais à direita

Algoritmos de parsing

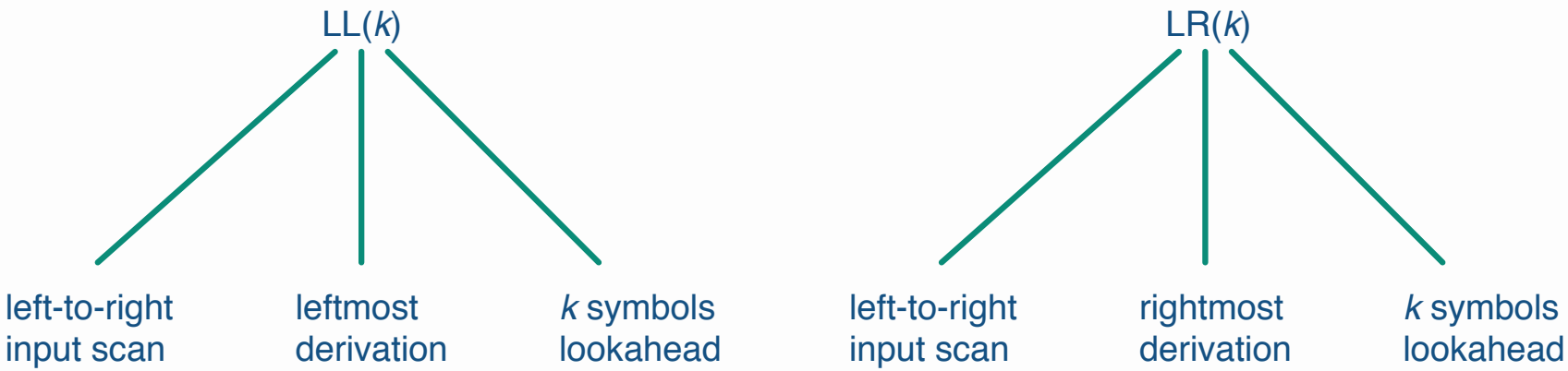
Gramáticas livres de contexto divididas em:

- **LL(k)** parsing
  - Top-down - análise descendente
  - Considera sempre derivações mais à esquerda
- **LR(k)** parsing
  - Bottom-up - análise ascendente
  - Considera sempre derivações mais à direita

k - indica o número de símbolos de LOOKAHEAD

LL vs LR parsing

- Parsers LL vs. parsers LR:



Outline

Re-escrita de gramáticas LL(1)

Uma gramática livre de contexto diz-se **LL(1)** se:

- Não possui recursividade à esquerda
- Para todos os pares de regras do mesmo símbolo não terminal, $A \to \alpha$ e $A \to \beta$:

$$LOOKAHEAD(A \to \alpha) \cap LOOKAHEAD(A \to \beta) = \emptyset$$

Re-escrita de gramáticas LL(1)

Uma gramática livre de contexto diz-se **LL(1)** se:

- Não possui recursividade à esquerda
- Para todos os pares de regras do mesmo símbolo não terminal, $A \to \alpha$ e $A \to \beta$:

$$LOOKAHEAD(A \to \alpha) \cap LOOKAHEAD(A \to \beta) = \emptyset$$

Abordagem - Re-escrever a gramática de forma a:

- eliminar recursividade à esquerda
- factorizar à esquerda

Re-escrita de gramáticas LL(1)

Passos

0. Eliminação de singularidades e símbolos desnecessários
1. Eliminação da recursão mútua
2. Eliminação de cantos à esquerda
3. Eliminação da recusão à esquerda
4. Factorização à esquerda

Símbolos não terminais com uma única regra constituída apenas por um símbolo terminal (ou ϵ), e podem ser substituídos nas regras onde ocorrem.

Símbolos não terminais com uma única regra constituída apenas por um símbolo terminal (ou ϵ), e podem ser substituídos nas regras onde ocorrem.

- Caso geral:

$$A \rightarrow B\,\alpha \mid \ldots$$
$$B \rightarrow b$$

Símbolos não terminais com uma única regra constituída apenas por um símbolo terminal (ou ϵ), e podem ser substituídos nas regras onde ocorrem.

- Caso geral:

$$A \to B\,\alpha \mid \ldots$$
$$B \to b$$

- Transformação:

$$A \to b\,\alpha \mid \ldots$$
$$B \to b$$

Re-escrita de gramáticas

Passo 0 - Eliminação de singularidades e símbolos desnecessários

Símbolos não terminais com uma única regra constituída apenas por um símbolo terminal (ou ϵ), e podem ser substituídos nas regras onde ocorrem.

- Caso geral:

$$A \to B\,\alpha \mid \ldots$$
$$B \to b$$

- Transformação:

$$A \to b\,\alpha \mid \ldots$$
$$B \to b$$

Se um símbolo não terminal B é não atingível então pode ser removido.

- Transformação:

$$A \to b\,\alpha \mid \ldots$$

Nota: Nunca eliminar o símbolo inicial da gramática!

Se um símbolo não terminal A inicia uma regra de um símbolo não terminal B e vice-versa, B pode ser substituído em todas as ocorrências em A, pelas suas regras.

Se um símbolo não terminal A inicia uma regra de um símbolo não terminal B e vice-versa, B pode ser substituído em todas as ocorrências em A, pelas suas regras.

- Caso geral:

$$A \rightarrow \sigma\,\delta \mid B\,\gamma \mid \ldots$$
$$B \rightarrow A\,\alpha \mid \epsilon$$

Se um símbolo não terminal A inicia uma regra de um símbolo não terminal B e vice-versa, B pode ser substituído em todas as ocorrências em A, pelas suas regras.

- Caso geral:

$$A \to \sigma\,\delta \mid B\,\gamma \mid \ldots$$
$$B \to A\,\alpha \mid \epsilon$$

- Transformação:

$$A \to \sigma\,\delta \mid A\,\alpha\,\gamma \mid \gamma \mid \ldots$$

Se um símbolo não terminal A inicia uma regra de um símbolo não terminal B e vice-versa, B pode ser substituído em todas as ocorrências em A, pelas suas regras.

- Caso geral:

$$A \to \sigma\,\delta \mid B\,\gamma \mid \ldots$$
$$B \to A\,\alpha \mid \epsilon$$

- Transformação:

$$A \to \sigma\,\delta \mid A\,\alpha\,\gamma \mid \gamma \mid \ldots$$

Nota: se A for símbolo inicial da gramática, não pode ser eliminado!

Se um símbolo não terminal B inicia uma regra de um símbolo não terminal A, B pode ser substituído em todas as ocorrências em A, pelas suas regras.

Se um símbolo não terminal B inicia uma regra de um símbolo não terminal A, B pode ser substituído em todas as ocorrências em A, pelas suas regras.

- Caso geral:

$$A \rightarrow \sigma\,\delta \mid B\,\gamma \mid \ldots$$
$$B \rightarrow \alpha \mid \epsilon$$

Se um símbolo não terminal B inicia uma regra de um símbolo não terminal A, B pode ser substituído em todas as ocorrências em A, pelas suas regras.

- Caso geral:

$$A \to \sigma\,\delta \mid B\,\gamma \mid \ldots$$
$$B \to \alpha \mid \epsilon$$

- Transformação:

$$A \to \sigma\,\delta \mid \alpha\,\gamma \mid \gamma \mid \ldots$$

Se um símbolo não terminal A iniciar algumas das suas regras, criar um novo símbolo A' da seguinte forma.

Se um símbolo não terminal A iniciar algumas das suas regras, criar um novo símbolo A' da seguinte forma.

- Caso geral:

$$A \to A\,\alpha_1 \mid \ldots \mid A\,\alpha_n \mid \beta_1 \mid \ldots \mid \beta_m$$

Se um símbolo não terminal A iniciar algumas das suas regras, criar um novo símbolo A' da seguinte forma.

- Caso geral:

$$A \to A\,\alpha_1 \mid \ldots \mid A\,\alpha_n \mid \beta_1 \mid \ldots \mid \beta_m$$

- Transformação:

$$A \to \beta_1\,A' \mid \ldots \mid \beta_m\,A'$$
$$A' \to \alpha_1\,A' \mid \ldots \mid \alpha_n\,A' \mid \epsilon$$

Dada uma gramática G sem recursões à esquerda:

- G é ambígua se várias produções tiverem os mesmos conjuntos de LOOKAHEAD

Exemplo:

$$
\begin{array}{lcl}
\text{stmt} & \rightarrow & \textbf{if}\ \text{expr}\ \textbf{then}\ \text{stmt} \\
& | & \textbf{if}\ \text{expr}\ \textbf{then}\ \text{stmt}\ \textbf{else}\ \text{stmt} \\
& | & \textbf{other}
\end{array}
$$

Dada uma gramática G sem recursões à esquerda:

- G é ambígua se várias produções tiverem os mesmos conjuntos de LOOKAHEAD

Exemplo:

$$
\begin{array}{lcl}
\text{stmt} & \rightarrow & \textbf{if}\ \text{expr}\ \textbf{then}\ \text{stmt} \\
& | & \textbf{if}\ \text{expr}\ \textbf{then}\ \text{stmt}\ \textbf{else}\ \text{stmt} \\
& | & \textbf{other}
\end{array}
$$

- Caso geral:

$$A \rightarrow \alpha\,\beta \mid \alpha\,\gamma$$

Dada uma gramática G sem recursões à esquerda:

- G é ambígua se várias produções tiverem os mesmos conjuntos de LOOKAHEAD

Exemplo:

$$
\begin{array}{lcl}
\text{stmt} & \rightarrow & \textbf{if}\ \text{expr}\ \textbf{then}\ \text{stmt} \\
& | & \textbf{if}\ \text{expr}\ \textbf{then}\ \text{stmt}\ \textbf{else}\ \text{stmt} \\
& | & \textbf{other}
\end{array}
$$

- Caso geral:

$$A \rightarrow \alpha\,\beta \mid \alpha\,\gamma$$

- Transformação:

$$
\begin{array}{l}
A \rightarrow \alpha\,A' \\
A' \rightarrow \beta \mid \gamma
\end{array}
$$

Re-escrita de gramáticas

Exemplo

$$
\begin{aligned}
E &\rightarrow E + T \mid E \text{ - } T \mid T \\
T &\rightarrow T * F \mid T / F \mid F \\
F &\rightarrow (E) \mid \text{id}
\end{aligned}
$$

Re-escrita de gramáticas

Exemplo

$$
\begin{array}{lcl}
E & \rightarrow & E + T \mid E \text{ - } T \mid T \\
T & \rightarrow & T * F \mid T / F \mid F \\
F & \rightarrow & (E) \mid \text{id}
\end{array}
$$

Aplicação do passo 3 (Eliminação da recursão à esquerda):

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & + T\ E' \mid \text{ - } T\ E' \mid \epsilon
\end{array}
$$

$$
\begin{array}{lcl}
E & \rightarrow & E + T \mid E \text{ - } T \mid T \\
T & \rightarrow & T * F \mid T / F \mid F \\
F & \rightarrow & (E) \mid \text{id}
\end{array}
$$

Aplicação do passo 3 (Eliminação da recursão à esquerda):

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & + T\ E' \mid \text{ - } T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & * F\ T' \mid / F\ T' \mid \epsilon
\end{array}
$$

Re-escrita de gramáticas

Exemplo

$$
\begin{array}{lcl}
E & \rightarrow & E + T \mid E \text{ - } T \mid T \\
T & \rightarrow & T * F \mid T / F \mid F \\
F & \rightarrow & (E) \mid \text{id}
\end{array}
$$

Aplicação do passo 3 (Eliminação da recursão à esquerda):

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & + T\ E' \mid \text{ - } T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & * F\ T' \mid / F\ T' \mid \epsilon \\
F & \rightarrow & (E) \mid \text{id}
\end{array}
$$

Re-escrita de gramáticas

Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

Re-escrita de gramáticas

Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

Aplicação do passo 2 (Eliminação de cantos à esquerda):

$$
\begin{array}{rcl}
E & \rightarrow & F\ T'\ E'
\end{array}
$$

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

Aplicação do passo 2 (Eliminação de cantos à esquerda):

$$
\begin{array}{lcl}
E & \rightarrow & F\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

Re-escrita de gramáticas

Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & F\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathsf{id}
\end{array}
$$

Re-escrita de gramáticas

Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & F\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \text{id}
\end{array}
$$

Aplicação do passo 2 (Eliminação de cantos à esquerda):

$$
\begin{array}{rcl}
E & \rightarrow & (\ E\ )T'\ E' \mid \text{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon
\end{array}
$$

Re-escrita de gramáticas

Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & F\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \text{id}
\end{array}
$$

Aplicação do passo 2 (Eliminação de cantos à esquerda):

$$
\begin{array}{rcl}
E & \rightarrow & (\ E\ )T'\ E' \mid \text{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \text{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \text{id}
\end{array}
$$

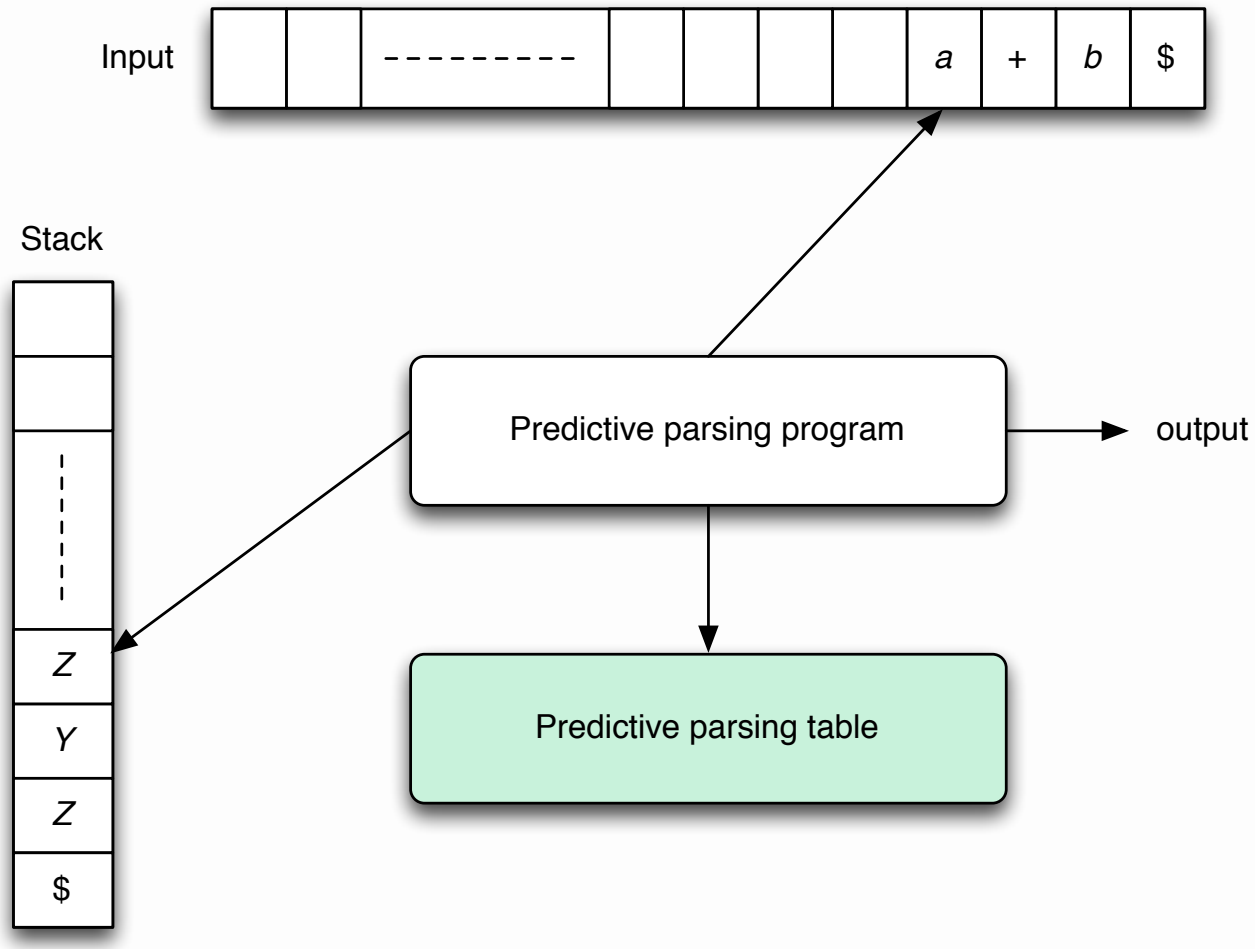
Outline

Analisador descendente

- Começar com símbolo inicial como símbolo alvo
- Se símbolo alvo é terminal:
  - Se entrada condiz, passar ao próximo símbolo na entrada, e passar ao símbolo alvo seguinte
- Caso contrário:
  - **Escolher** regra para mudar símbolo alvo
- Se sem símbolo alvo e sem input, então **backtrack**
- **Nota**: Parsing sem backtrack (e sem escolhas) é denominado parsing preditivo
  - Objectivo: parsing preditivo utilizando tabela de análise

Parsing preditivo

Abordagem



Dada uma gramática $G = (V, \Sigma, R, S)$, construir uma tabela de análise T:

- Colunas - uma por cada símbolo em $\Sigma \cup \{\$\}$
- Linhas - uma por cada símbolo não terminal em V

Parsing preditivo

Tabela de análise – Construção

Dada uma gramática $G = (V, \Sigma, R, S)$, construir uma tabela de análise T:

- Colunas - uma por cada símbolo em $\Sigma \cup \{\$\}$
- Linhas - uma por cada símbolo não terminal em V

Algoritmo

Para cada produção $A \to \alpha$:

1. Para cada $a \in \text{FIRST}(\alpha)$, adicionar $A \to \alpha$ a $T[A, a]$
2. Se $\epsilon \in \text{FIRST}(\alpha)$, para cada $b \in \text{FOLLOW}(A)$ adicionar $A \to \alpha$ a $T[A, b]$
   Se $\epsilon \in \text{FIRST}(A)$ e $\$ \in \text{FOLLOW}(A)$, então adicionar $A \to \alpha$ a $T[A, \$]$

Parsing preditivo

Tabela de análise – Construção

Dada uma gramática $G = (V, \Sigma, R, S)$, construir uma tabela de análise T:

- Colunas - uma por cada símbolo em $\Sigma \cup \{\$\}$
- Linhas - uma por cada símbolo não terminal em V

Algoritmo
Para cada produção $A \to \alpha$:

1. Para cada $a \in \text{FIRST}(\alpha)$, adicionar $A \to \alpha$ a $T[A, a]$
2. Se $\epsilon \in \text{FIRST}(\alpha)$, para cada $b \in \text{FOLLOW}(A)$ adicionar $A \to \alpha$ a $T[A, b]$
   Se $\epsilon \in \text{FIRST}(A)$ e $\$ \in \text{FOLLOW}(A)$, então adicionar $A \to \alpha$ a $T[A, \$]$

- Entradas vazias representam um **erro de parsing**
- Entradas duplicadas significam que gramática é **ambígua**, **não** é LL(1)

Parsing preditivo

Tabela de análise – Construção

Dada uma gramática $G = (V, \Sigma, R, S)$, construir uma tabela de análise T:

- Colunas - uma por cada símbolo em $\Sigma \cup \{\$\}$
- Linhas - uma por cada símbolo não terminal em V

Algoritmo

Para cada produção $A \to \alpha$:

1. Para cada $a \in \text{FIRST}(\alpha)$, adicionar $A \to \alpha$ a $T[A, a]$
2. Se $\epsilon \in \text{FIRST}(\alpha)$, para cada $b \in \text{FOLLOW}(A)$ adicionar $A \to \alpha$ a $T[A, b]$
   Se $\epsilon \in \text{FIRST}(A)$ e $\$ \in \text{FOLLOW}(A)$, então adicionar $A \to \alpha$ a $T[A, \$]$

- Entradas vazias representam um **erro de parsing**
- Entradas duplicadas significam que gramática é **ambígua**, **não** é LL(1)
- Se tabela **sem** entradas duplicadas, então gramática é LL(1)

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & (\; E \;) \, T' \;\; E' \mid \mathrm{id} \;\; T' \;\; E' \\
E' & \rightarrow & + \;\; T \;\; E' \mid - \;\; T \;\; E' \mid \epsilon \\
T & \rightarrow & (\; E \;) \;\; T' \mid \mathrm{id} \;\; T' \\
T' & \rightarrow & * \;\; F \;\; T' \mid / \;\; F \;\; T' \mid \epsilon \\
F & \rightarrow & (\; E \;) \mid \mathrm{id}
\end{array}
$$

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & (\ E\ )T'\ E' \mid \mathrm{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \mathrm{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

	FIRST
E	$\{(, \mathrm{id}\}$
E'	$\{+, \text{-}, \epsilon\}$
T	$\{(, \mathrm{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \mathrm{id}\}$

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & (\ E\ )T'\ E' \mid \mathrm{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \mathrm{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

	FIRST
E	$\{(, \mathrm{id}\}$
E'	$\{+, \text{-}, \epsilon\}$
T	$\{(, \mathrm{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \mathrm{id}\}$

	FOLLOW
E	$\{), \$\}$
E'	$\{), \$\}$
T	$\{+, \text{-},), \$\}$
T'	$\{+, \text{-},), \$\}$
F	$\{*, /, +, \text{-},), \$\}$

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{lcl}
E & \rightarrow & (\ E\ )T'\ E' \mid \text{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \text{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \text{id}
\end{array}
$$

	FIRST
E	$\{(, \text{id}\}$
E'	$\{+, \text{-}, \epsilon\}$
T	$\{(, \text{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \text{id}\}$

	FOLLOW
E	$\{), \$\}$
E'	$\{), \$\}$
T	$\{+, \text{-},), \$\}$
T'	$\{+, \text{-},), \$\}$
F	$\{*, /, +, \text{-},), \$\}$

	+	-	*	/	(	)	id	$
E								

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{lcl}
E & \rightarrow & (\ E\ )T'\ E' \mid \text{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \text{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \text{id}
\end{array}
$$

	FIRST
E	$\{(, \text{id}\}$
E'	$\{+, \text{-}, \epsilon\}$
T	$\{(, \text{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \text{id}\}$

	FOLLOW
E	$\{), \$\}$
E'	$\{), \$\}$
T	$\{+, \text{-},), \$\}$
T'	$\{+, \text{-},), \$\}$
F	$\{*, /, +, \text{-},), \$\}$

	+	-	*	/	(	)	id	$
E					$E \rightarrow (E)T'E'$		$E \rightarrow \text{id}\,T'E'$	
E'								

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & (\ E\ )T'\ E' \mid \mathrm{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \mathrm{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

	FIRST
E	$\{(, \mathrm{id}\}$
E'	$\{+, \text{-}, \epsilon\}$
T	$\{(, \mathrm{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \mathrm{id}\}$

	FOLLOW
E	$\{), \$\}$
E'	$\{), \$\}$
T	$\{+, \text{-},), \$\}$
T'	$\{+, \text{-},), \$\}$
F	$\{*, /, +, \text{-},), \$\}$

	+	-	*	/	(	)	id	$
E					$E \rightarrow (E)T'E'$		$E \rightarrow \mathrm{id}\,T'E'$	
E'	$E' \rightarrow +TE'$	$E' \rightarrow \text{-}TE'$				$E' \rightarrow \epsilon$		$E' \rightarrow \epsilon$
T								

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & (\ E\ )T'\ E' \mid \mathrm{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \mathrm{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

	FIRST
E	$\{(, \mathrm{id}\}$
E'	$\{+, \text{-}, \epsilon\}$
T	$\{(, \mathrm{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \mathrm{id}\}$

	FOLLOW
E	$\{), \$\}$
E'	$\{), \$\}$
T	$\{+, \text{-},), \$\}$
T'	$\{+, \text{-},), \$\}$
F	$\{*, /, +, \text{-},), \$\}$

	+	-	*	/	(	)	id	$
E					$E \rightarrow (E)T'E'$		$E \rightarrow \mathrm{id}\,T'E'$	
E'	$E' \rightarrow +TE'$	$E' \rightarrow \text{-}TE'$				$E' \rightarrow \epsilon$		$E' \rightarrow \epsilon$
T					$T \rightarrow (E)T'$		$T \rightarrow \mathrm{id}\,T'$	
T'								

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & (\ E\ )\,T'\ E' \mid \mathrm{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \mathrm{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

	FIRST
E	$\{(, \mathrm{id}\}$
E'	$\{+, \text{-}, \epsilon\}$
T	$\{(, \mathrm{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \mathrm{id}\}$

	FOLLOW
E	$\{), \$\}$
E'	$\{), \$\}$
T	$\{+, \text{-},), \$\}$
T'	$\{+, \text{-},), \$\}$
F	$\{*, /, +, \text{-},), \$\}$

	+	-	*	/	(	)	id	$
E					$E \rightarrow (E)\,T'\,E'$		$E \rightarrow \mathrm{id}\,T'\,E'$	
E'	$E' \rightarrow +\,T\,E'$	$E' \rightarrow \text{-}\,T\,E'$				$E' \rightarrow \epsilon$		$E' \rightarrow \epsilon$
T					$T \rightarrow (E)\,T'$		$T \rightarrow \mathrm{id}\,T'$	
T'	$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$	$T' \rightarrow *\,F\,T'$	$T' \rightarrow /\,F\,T'$		$T' \rightarrow \epsilon$		$T' \rightarrow \epsilon$
F								

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & (\ E\ )\,T'\ E' \mid \mathrm{id}\ T'\ E' \\
E' & \rightarrow & +\ T\ E' \mid \text{-}\ T\ E' \mid \epsilon \\
T & \rightarrow & (\ E\ )\ T' \mid \mathrm{id}\ T' \\
T' & \rightarrow & *\ F\ T' \mid /\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathrm{id}
\end{array}
$$

	FIRST
E	$\{(, \mathrm{id}\}$
E'	$\{+, \text{-}, \epsilon\}$
T	$\{(, \mathrm{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \mathrm{id}\}$

	FOLLOW
E	$\{), \$\}$
E'	$\{), \$\}$
T	$\{+, \text{-},), \$\}$
T'	$\{+, \text{-},), \$\}$
F	$\{*, /, +, \text{-},), \$\}$

	+	-	*	/	(	)	id	$
E					$E \rightarrow (E)\,T'\,E'$		$E \rightarrow \mathrm{id}\,T'\,E'$	
E'	$E' \rightarrow +\,T\,E'$	$E' \rightarrow \text{-}\,T\,E'$				$E' \rightarrow \epsilon$		$E' \rightarrow \epsilon$
T					$T \rightarrow (E)\,T'$		$T \rightarrow \mathrm{id}\,T'$	
T'	$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$	$T' \rightarrow *\,F\,T'$	$T' \rightarrow /\,F\,T'$		$T' \rightarrow \epsilon$		$T' \rightarrow \epsilon$
F					$F \rightarrow (E)$		$F \rightarrow \mathrm{id}$	

Parsing preditivo

Tabela de análise – Exemplo

$$
\begin{array}{rcl}
E & \rightarrow & (\; E \;) \; T' \;\; E'^{(1)} \mid \text{id} \;\; T' \;\; E'^{(2)} \\
E' & \rightarrow & + \;\; T \;\; E'^{(3)} \mid - \;\; T \;\; E'^{(4)} \mid \epsilon^{(5)} \\
T & \rightarrow & (\; E \;) \;\; T'^{(6)} \mid \text{id} \;\; T'^{(7)} \\
T' & \rightarrow & * \;\; F \;\; T'^{(8)} \mid / \;\; F \;\; T'^{(9)} \mid \epsilon^{(10)} \\
F & \rightarrow & (\; E \;)^{(11)} \mid \text{id}^{(12)}
\end{array}
$$

	FIRST
E	$\{(, \text{id}\}$
E'	$\{+, -, \epsilon\}$
T	$\{(, \text{id}\}$
T'	$\{*, /, \epsilon\}$
F	$\{(, \text{id}\}$

	FOLLOW
E	$\{), \$\}$
E'	$\{), \$\}$
T	$\{+, -,), \$\}$
T'	$\{+, -,), \$\}$
F	$\{*, /, +, -,), \$\}$

	+	-	*	/	(	)	id	$
E					**(1)**		**(2)**	
E'	**(3)**	**(4)**				**(5)**		**(5)**
T					**(6)**		**(7)**	
T'	**(10)**	**(10)**	**(8)**	**(9)**		**(10)**		**(10)**
F					**(11)**		**(12)**	

Algoritmo:

- Seja:
  - M a tabela de parsing de uma gramática G.
  - a o primeiro símbolo da string de entrada w.
  - X o símbolo no topo da pilha (inicializada com o símbolo inicial de G seguido de $).
- Enquanto $X \neq \$$:
  - Se entrada condiz ($X == a$) $\longrightarrow$ Retirar X do topo da pilha e a da entrada.
  - Caso contrário:
    - Se X é um símbolo terminal $\longrightarrow$ Rejeitar (**erro**)
    - Caso contrário, se não existe entrada na tabela de parsing $M[X, a]$ $\longrightarrow$ Rejeitar (**erro**)
    - Caso contrário, se $M[X, a] = X \rightarrow Y_1 Y_2 \ldots Y_k$ $\longrightarrow$ Retirar X to topo da pilha e pushar o corpo da produção (fincando Y_1 como o novo topo da ilha)

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)\,T'\,E'$		$E \to \mathrm{id}\,T'\,E'$	
E'	$E' \to +\,T\,E'$	$E' \to -\,T\,E'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)\,T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *\,F\,T'$	$T' \to /\,F\,T'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)\,T'\,E'$		$E \to \mathrm{id}\,T'\,E'$	
E'	$E' \to +\,T\,E'$	$E' \to -\,T\,E'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)\,T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *\,F\,T'$	$T' \to /\,F\,T'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \mathrm{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	$E \to \mathrm{id}\,T'E'$
2	$\mathrm{id}\,T'E'\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \mathrm{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	$E \to \mathrm{id}\,T'E'$
2	$\mathrm{id}\,T'E'\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	(id)
3	$T'E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \mathrm{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	$E \to \mathrm{id}\,T'E'$
2	$\mathrm{id}\,T'E'\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	(id)
3	$T'E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$E' \to +TE'$
5	$+T\ E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \mathrm{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	$E \to \mathrm{id}\,T'E'$
2	$\mathrm{id}\,T'E'\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	(id)
3	$T'E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$E' \to +TE'$
5	$+\,T\;E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	(+)
6	$T\;E'\$$	$\mathrm{id} * \mathrm{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \mathrm{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	$E \to \mathrm{id}\,T'E'$
2	$\mathrm{id}\,T'E'\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	(id)
3	$T'E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$E' \to +TE'$
5	$+\,T\;E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	(+)
6	$T\;E'\$$	$\mathrm{id} * \mathrm{id}\,\$$	$T \to \mathrm{id}\,T'$
7	$\mathrm{id}\,T'E'\$$	$\mathrm{id} * \mathrm{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \text{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \text{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \text{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\text{id} + \text{id} * \text{id}\,\$$	$E \to \text{id}\,T'E'$
2	$\text{id}\,T'E'\$$	$\text{id} + \text{id} * \text{id}\,\$$	(id)
3	$T'E'\$$	$+\,\text{id} * \text{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\text{id} * \text{id}\,\$$	$E' \to +TE'$
5	$+\,T\;E'\$$	$+\,\text{id} * \text{id}\,\$$	(+)
6	$T\;E'\$$	$\text{id} * \text{id}\,\$$	$T \to \text{id}\,T'$
7	$\text{id}\,T'E'\$$	$\text{id} * \text{id}\,\$$	(id)
8	$T'E'\$$	$*\,\text{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \text{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \text{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \text{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\text{id} + \text{id} * \text{id}\,\$$	$E \to \text{id}\,T'E'$
2	$\text{id}\,T'E'\$$	$\text{id} + \text{id} * \text{id}\,\$$	(id)
3	$T'E'\$$	$+\,\text{id} * \text{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\text{id} * \text{id}\,\$$	$E' \to +TE'$
5	$+T\ E'\$$	$+\,\text{id} * \text{id}\,\$$	(+)
6	$T\ E'\$$	$\text{id} * \text{id}\,\$$	$T \to \text{id}\,T'$
7	$\text{id}\,T'E'\$$	$\text{id} * \text{id}\,\$$	(id)
8	$T'E'\$$	$*\,\text{id}\,\$$	$T' \to *FT'$
9	$*FT'E'\$$	$*\,\text{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \mathrm{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	$E \to \mathrm{id}\,T'E'$
2	$\mathrm{id}\,T'E'\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	(id)
3	$T'E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$E' \to +TE'$
5	$+T\ E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	(+)
6	$T\ E'\$$	$\mathrm{id} * \mathrm{id}\,\$$	$T \to \mathrm{id}\,T'$
7	$\mathrm{id}\,T'E'\$$	$\mathrm{id} * \mathrm{id}\,\$$	(id)
8	$T'E'\$$	$*\,\mathrm{id}\,\$$	$T' \to *FT'$
9	$*FT'E'\$$	$*\,\mathrm{id}\,\$$	(*)
10	$FT'E'\$$	$\mathrm{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \mathrm{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \mathrm{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \mathrm{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	$E \to \mathrm{id}\,T'E'$
2	$\mathrm{id}\,T'E'\$$	$\mathrm{id} + \mathrm{id} * \mathrm{id}\,\$$	(id)
3	$T'E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	$E' \to +TE'$
5	$+T\ E'\$$	$+\,\mathrm{id} * \mathrm{id}\,\$$	(+)
6	$T\ E'\$$	$\mathrm{id} * \mathrm{id}\,\$$	$T \to \mathrm{id}\,T'$
7	$\mathrm{id}\,T'E'\$$	$\mathrm{id} * \mathrm{id}\,\$$	(id)
8	$T'E'\$$	$*\,\mathrm{id}\,\$$	$T' \to *FT'$
9	$*FT'E'\$$	$*\,\mathrm{id}\,\$$	(*)
10	$FT'E'\$$	$\mathrm{id}\,\$$	$F \to \mathrm{id}$
11	$\mathrm{id}\ T'\ E'\ \$$	$\mathrm{id}\,\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \text{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \text{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \text{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\text{id} + \text{id} * \text{id}\,\$$	$E \to \text{id}\,T'E'$
2	$\text{id}\,T'E'\$$	$\text{id} + \text{id} * \text{id}\,\$$	(id)
3	$T'E'\$$	$+\,\text{id} * \text{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\text{id} * \text{id}\,\$$	$E' \to +TE'$
5	$+T\ E'\$$	$+\,\text{id} * \text{id}\,\$$	(+)
6	$T\ E'\$$	$\text{id} * \text{id}\,\$$	$T \to \text{id}\,T'$
7	$\text{id}\,T'E'\$$	$\text{id} * \text{id}\,\$$	(id)
8	$T'E'\$$	$*\,\text{id}\,\$$	$T' \to *FT'$
9	$*FT'E'\$$	$*\,\text{id}\,\$$	(*)
10	$FT'E'\$$	$\text{id}\,\$$	$F \to \text{id}$
11	$\text{id}\ T'\ E'\ \$$	$\text{id}\,\$$	(id)
12	$T'\ E'\ \$$	$\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \text{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \text{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \text{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\text{id} + \text{id} * \text{id}\,\$$	$E \to \text{id}\,T'E'$
2	$\text{id}\,T'E'\$$	$\text{id} + \text{id} * \text{id}\,\$$	(id)
3	$T'E'\$$	$+\,\text{id} * \text{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\text{id} * \text{id}\,\$$	$E' \to +TE'$
5	$+T\ E'\$$	$+\,\text{id} * \text{id}\,\$$	(+)
6	$T\ E'\$$	$\text{id} * \text{id}\,\$$	$T \to \text{id}\,T'$
7	$\text{id}\,T'E'\$$	$\text{id} * \text{id}\,\$$	(id)
8	$T'E'\$$	$*\,\text{id}\,\$$	$T' \to *FT'$
9	$*FT'E'\$$	$*\,\text{id}\,\$$	(*)
10	$FT'E'\$$	$\text{id}\,\$$	$F \to \text{id}$
11	$\text{id}\ T'\ E'\ \$$	$\text{id}\,\$$	(id)
12	$T'\ E'\ \$$	$\$$	$T' \to \epsilon$
13	$E'\ \$$	$\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \text{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \text{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \text{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\text{id} + \text{id} * \text{id}\,\$$	$E \to \text{id}\,T'E'$
2	$\text{id}\,T'E'\$$	$\text{id} + \text{id} * \text{id}\,\$$	(id)
3	$T'E'\$$	$+\,\text{id} * \text{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\text{id} * \text{id}\,\$$	$E' \to +TE'$
5	$+T\ E'\$$	$+\,\text{id} * \text{id}\,\$$	(+)
6	$T\ E'\$$	$\text{id} * \text{id}\,\$$	$T \to \text{id}\,T'$
7	$\text{id}\,T'E'\$$	$\text{id} * \text{id}\,\$$	(id)
8	$T'E'\$$	$*\,\text{id}\,\$$	$T' \to *FT'$
9	$*FT'E'\$$	$*\,\text{id}\,\$$	(*)
10	$FT'E'\$$	$\text{id}\,\$$	$F \to \text{id}$
11	$\text{id}\ T'\ E'\ \$$	$\text{id}\,\$$	(id)
12	$T'\ E'\ \$$	$\$$	$T' \to \epsilon$
13	$E'\ \$$	$\$$	$E' \to \epsilon$
14	$\$$	$\$$	

Parsing preditivo

Processamento da entrada – Exemplo

	+	-	*	/	(	)	id	$
E					$E \to (E)T'E'$		$E \to \text{id}\,T'E'$	
E'	$E' \to +TE'$	$E' \to -TE'$				$E' \to \epsilon$		$E' \to \epsilon$
T					$T \to (E)T'$		$T \to \text{id}\,T'$	
T'	$T' \to \epsilon$	$T' \to \epsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \epsilon$		$T' \to \epsilon$
F					$F \to (E)$		$F \to \text{id}$	

Passo	Pilha	Entrada	Acção
1	$E\,\$$	$\text{id} + \text{id} * \text{id}\,\$$	$E \to \text{id}\,T'E'$
2	$\text{id}\,T'E'\$$	$\text{id} + \text{id} * \text{id}\,\$$	(id)
3	$T'E'\$$	$+\,\text{id} * \text{id}\,\$$	$T' \to \epsilon$
4	$E'\$$	$+\,\text{id} * \text{id}\,\$$	$E' \to +TE'$
5	$+T\ E'\$$	$+\,\text{id} * \text{id}\,\$$	(+)
6	$T\ E'\$$	$\text{id} * \text{id}\,\$$	$T \to \text{id}\,T'$
7	$\text{id}\,T'E'\$$	$\text{id} * \text{id}\,\$$	(id)
8	$T'E'\$$	$*\,\text{id}\,\$$	$T' \to *FT'$
9	$*FT'E'\$$	$*\,\text{id}\,\$$	(*)
10	$FT'E'\$$	$\text{id}\,\$$	$F \to \text{id}$
11	$\text{id}\ T'\ E'\ \$$	$\text{id}\,\$$	(id)
12	$T'\ E'\ \$$	$\$$	$T' \to \epsilon$
13	$E'\ \$$	$\$$	$E' \to \epsilon$
14	$\$$	$\$$	ACCEPT

Parsing preditivo

Exemplos

- Mais exemplos disponíveis na WIKI:
  https://web.tecnico.ulisboa.pt/~david.matos/w/pt/index.php/Top-Down_Parsing
- Abordagem:
  1. Transformar gramática inicial:
     - Substituir produções singulares
     - Eliminar left-recursions, mutual recursions e left-corners
     - Aplicar left-factorization
  2. Obter conjuntos FIRST e FOLLOW
  3. Obter tabela de parsing
  4. Fazer parsing da entrada
     - Casos especial: Gramática com conflitos (não LL(1))

Questões?

Dúvidas?