

Lógica da Computação

Aula 07

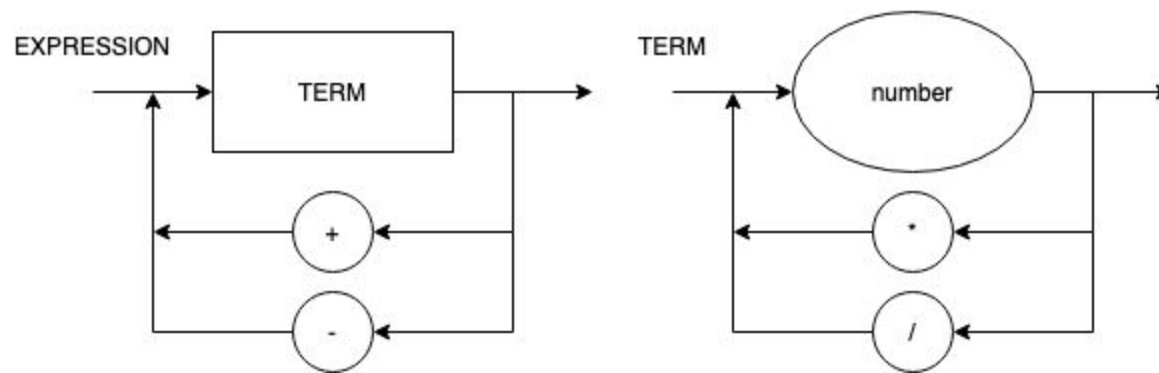
Raul Ikeda

1º semestre de 2020

Esta Aula

- Árvores de Derivação
- Melhorias no Compilador:
 - Operadores Unários
 - Parenteses

Situação atual



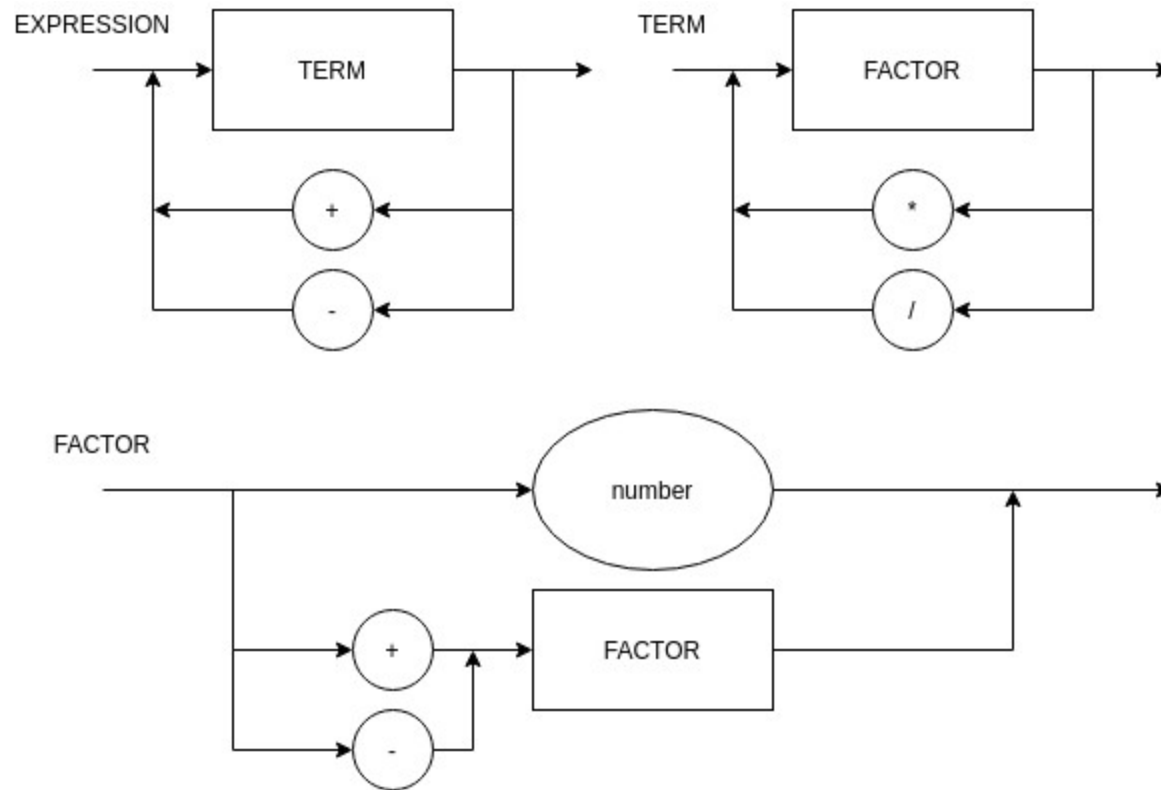
Operadores Unários

- Até o momento trabalhamos com operadores BINÁRIOS (operam dois elementos).
- Exemplos de operações UNÁRIAS (um elemento):

```
>> -2  
>> --2  
>> +2  
>> ++--++2
```

- Vocês conhecem outros operadores unários?

Atualizando o DS



Precedência de novo

- Já foi resolvido o problema de precedência e portanto:

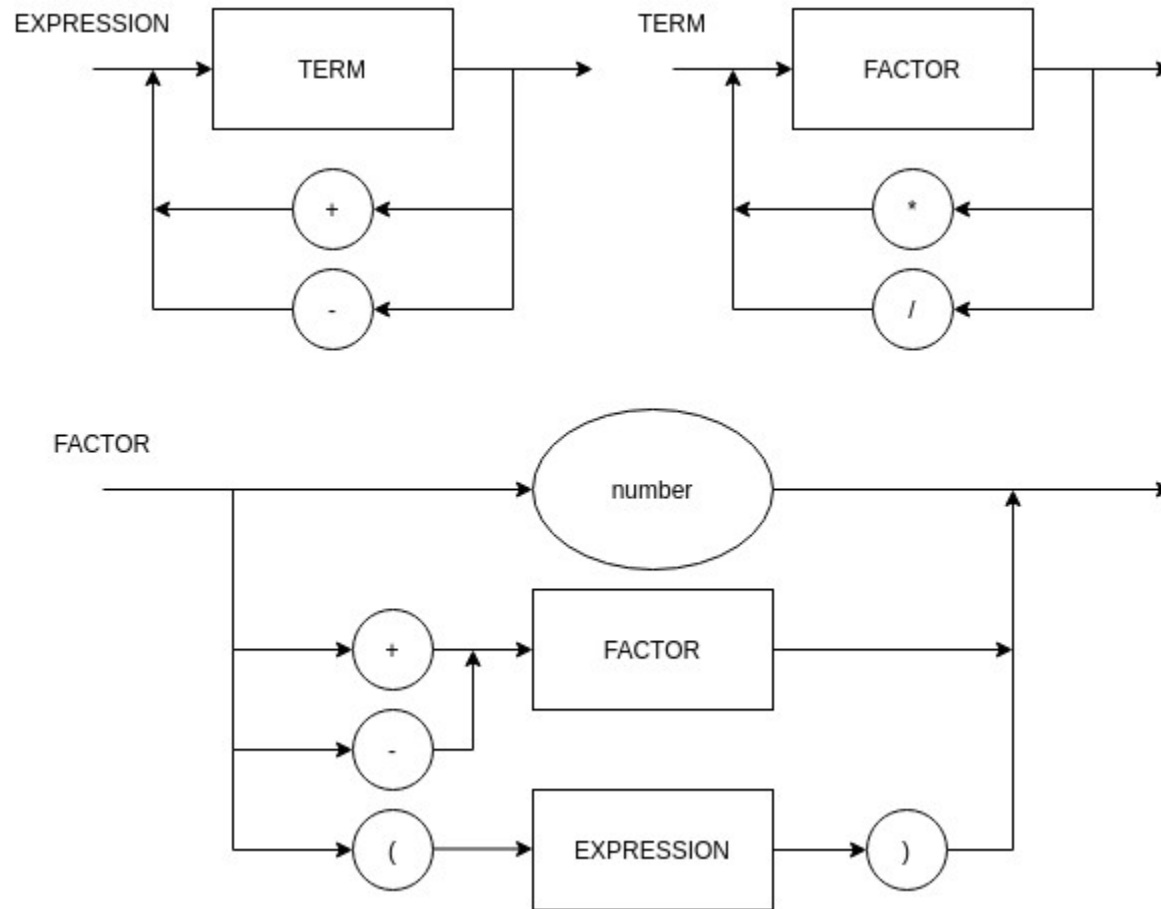
$$2 + 4/2 \rightarrow 4$$

- Mas se quiséssemos realizar a soma primeiro?

$$(2 + 4)/2 \rightarrow 3$$

- Qual o impacto dessa alteração no DS?
- E na gramática?

Atualizando o DS



Tentando uma Gramática Linear

$$G = (\{E, T, F, +, -, *, /, (,), n\}, \{+, -, *, /, (,), n\}, P, E)$$

$$P = \left\{ \begin{array}{l} E \rightarrow T \\ E \rightarrow n + T \\ E \rightarrow n - T \\ T \rightarrow n * E \\ T \rightarrow n / E \\ T \rightarrow F \\ F \rightarrow -E \\ F \rightarrow +E \\ F \rightarrow (E) \\ F \rightarrow n \end{array} \right.$$

Essa gramática é linear?

EBNF e Nova Gramática

```
EXPRESSION = TERM, { ("+" | "-"), TERM } ;  
TERM = FACTOR, { ("*" | "/"), FACTOR } ;  
FACTOR = ("+" | "-") FACTOR | "(" EXPRESSION ")" | number ;
```

$$G = (\{E, T, F, +, -, *, /, (,), n\}, \{+, -, *, /, (,), n\}, P, E)$$

$$P = \left\{ \begin{array}{l} E \rightarrow T \\ E \rightarrow E + E \\ E \rightarrow E - E \\ T \rightarrow F \\ T \rightarrow T * T \\ T \rightarrow T / T \\ F \rightarrow -F \\ F \rightarrow +F \\ F \rightarrow (E) \\ F \rightarrow n \end{array} \right.$$

Abrindo mão de ser linear, podemos ver uma nova ferramenta!

Árvore de Derivação

$$G = (\{E, T, F\} \cup \Sigma, \Sigma, P, E)$$

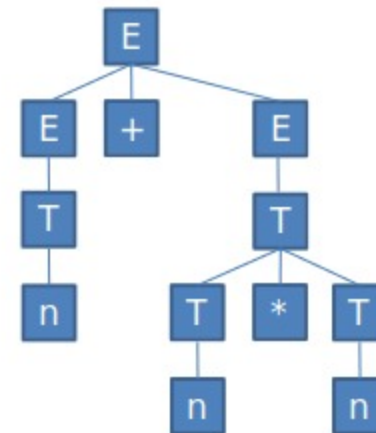
$$\Sigma = \{+, -, *, /, (,), n\}$$

$$P = \left\{ \begin{array}{l} E \rightarrow T \\ E \rightarrow E + E \\ E \rightarrow E - E \\ T \rightarrow F \\ T \rightarrow T * T \\ T \rightarrow T / T \\ F \rightarrow -F \\ F \rightarrow +F \\ F \rightarrow (E) \\ F \rightarrow n \end{array} \right.$$

Exemplo: $n + n * n$

$$\begin{aligned} E &\rightarrow E + E \rightarrow T + E \rightarrow T + T \rightarrow \\ &T + T * T \rightarrow n + n * n \end{aligned}$$

Formatado em Árvore:

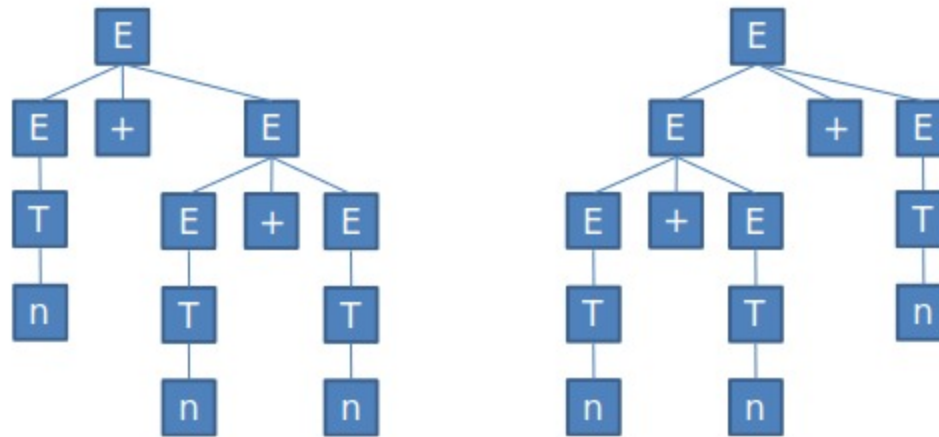


Essa gramática é **AMBÍGUA**!

Ambiguidade

Teste: $n + n + n$

Ambiguidade: É possível derivar duas árvores diferentes para a mesma cadeia:



Ambiguidade

Remova { e } da linguagem C e tente interpretar:

```
if (a==b)
c++;
if (c<=n)
d++; e++;
else // Esse else pertence a quem?
d--;
e++; // isso deveria executar dentro ou fora do if? Qual deles?
```

Corrigindo a Gramática

$$G = (\{E, T, F, +, -, *, /, (,), n\}, \{+, -, *, /, (,), n\}, P, E)$$

$$P = \left\{ \begin{array}{l} E \rightarrow T \\ E \rightarrow E + T \\ E \rightarrow E - T \\ T \rightarrow F \\ T \rightarrow T * F \\ T \rightarrow T / F \\ F \rightarrow -F \\ F \rightarrow +F \\ F \rightarrow (E) \\ F \rightarrow n \end{array} \right.$$

- Derivar: $n + n * n + n$

Atividade: Roteiro 3

- Roteiro Impresso ou PDF no Blackboard.
- Mudança de paradigma: as questões dissertativas são preparatórias para o próximo roteiro.

Próxima Aula

- Gramáticas Livre de Contexto
- Autômatos de Pilha

Referências:

- Hopcroft et al. Cap. 5 e 6