

Lógica da Computação

Aula 07

Raul Ikeda

2º semestre de 2018

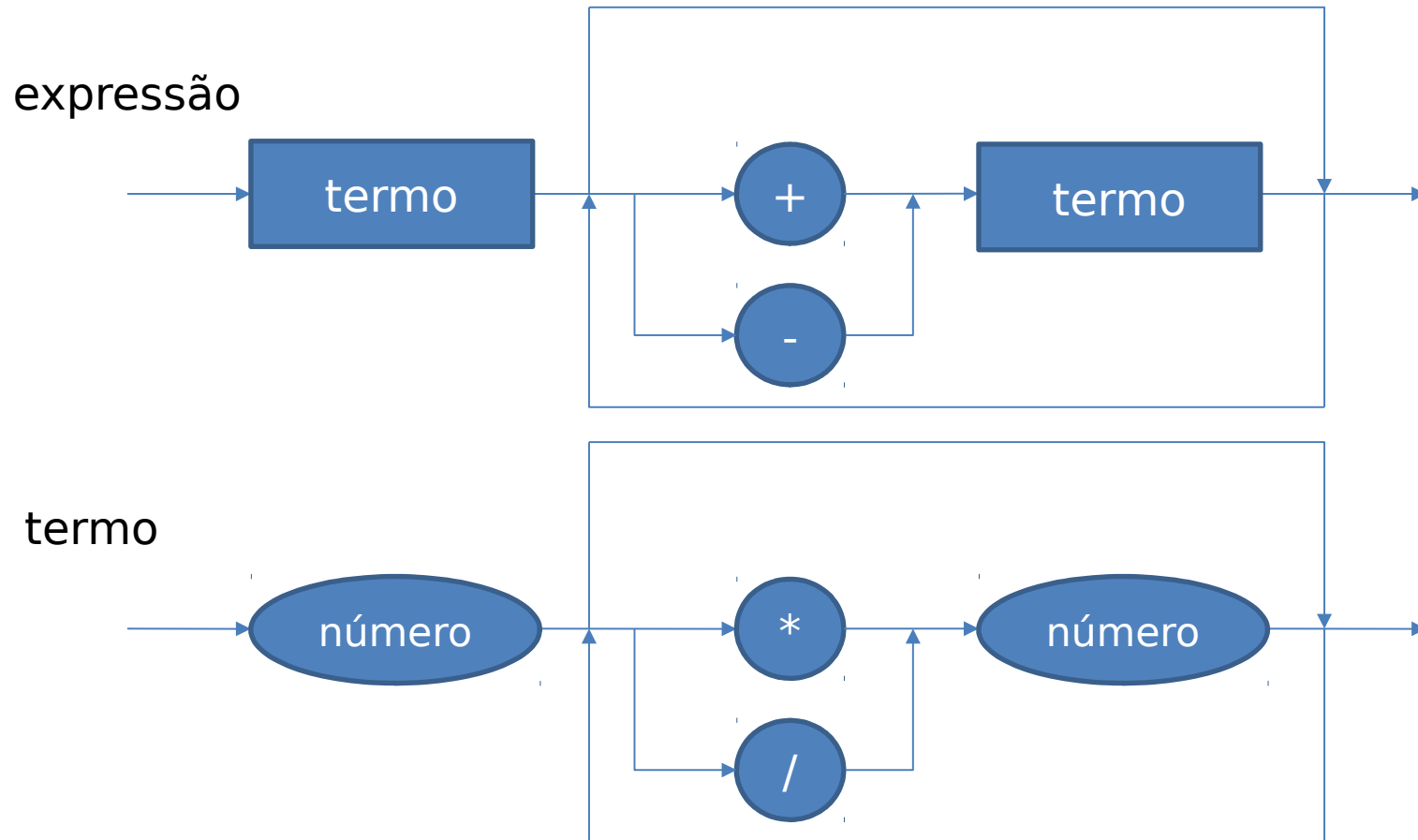
Aula Passada

1. Gramática Regulares
2. Pumping Lemma

Esta Aula

1. Árvores de Derivação
2. Melhorias no Compilador:
 1. Operadores Unários
 2. Parenteses

Situação atual



Operadores Unários

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

```
>> -2
```

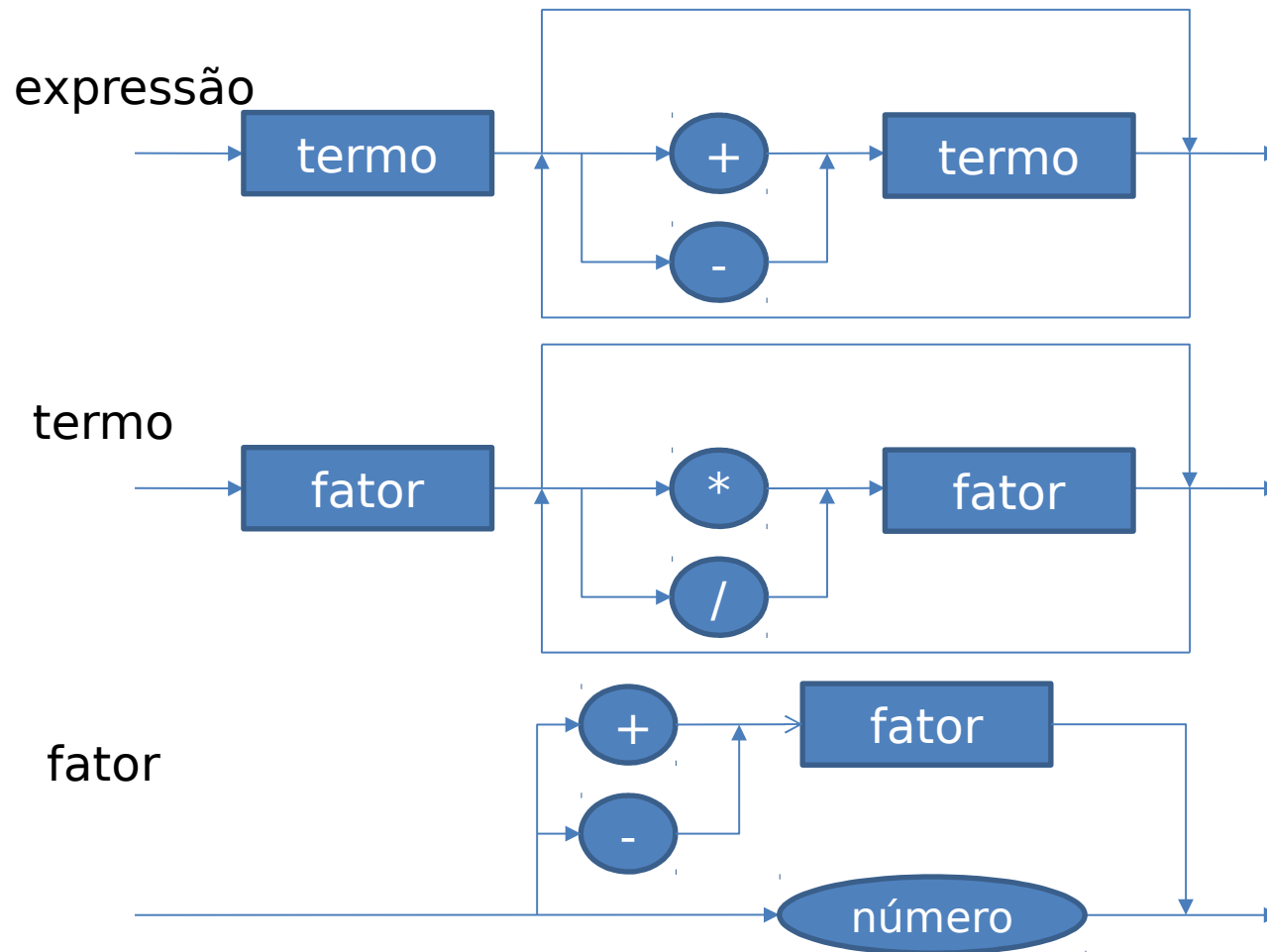
```
>> --2
```

```
>> +2
```

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

- Vocês conhecem outros operadores unários?

Atualizando o DS



Precedência

- Já foi resolvido o problema de precedência e portanto:

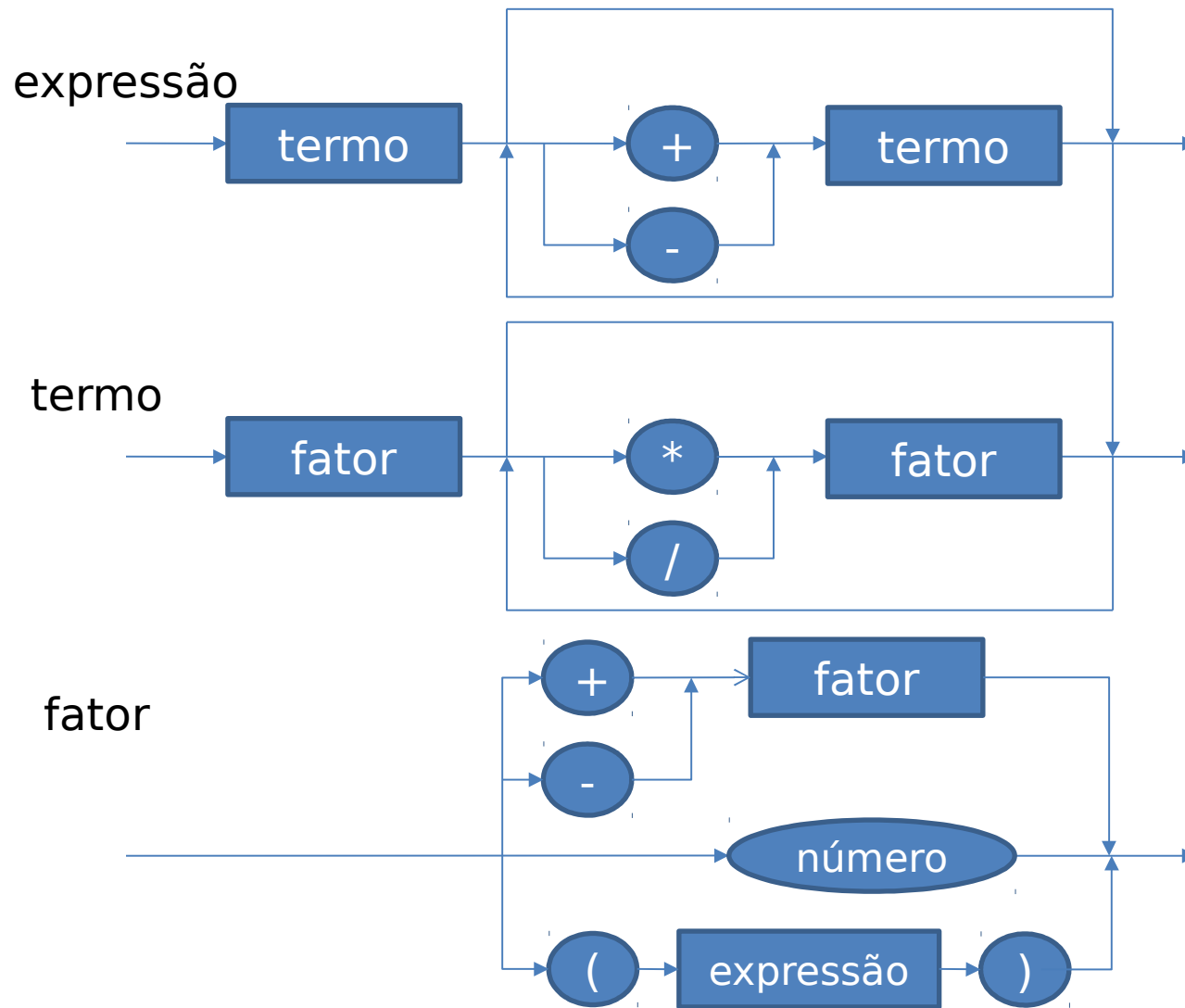
$$3 + 2/5 \rightarrow 3.4$$

- Mas e se quiséssemos executar a soma primeiro?

$$(3 + 2)/5 \rightarrow 1$$

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

Atualizando o DS



Tentando uma Gramática Regular

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

$$P = \begin{cases} E \rightarrow nT \\ E \rightarrow +E \\ E \rightarrow -E \\ E \rightarrow (E)T \\ T \rightarrow +E \\ T \rightarrow -E \\ T \rightarrow *E \\ T \rightarrow /E \\ T \rightarrow \lambda \end{cases} \longrightarrow \text{Opa! Isso não é Regular!}$$

EBNF e Nova Gramática

expressão = termo, { ("+" | "-"), termo } ;
termo = fator, { ("*" | "/"), fator } ;
fator = ("+" | "-") fator, número, "(" expressão ")" ;
número = "-2⁶³" | ... | "2⁶³" ;

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

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

Agora que nos libertamos das **gramáticas regulares**, podemos reescrever a gramática mais parecido com o Diagrama Sintático

Espera um pouco, essa gramática não é **AMBIGUA**?

Árvore de Derivação

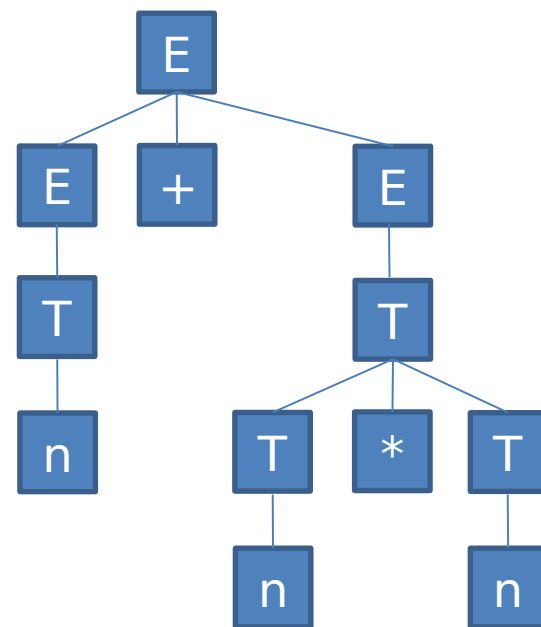
Exemplo: $n + n * n$

$E \rightarrow E + E \rightarrow T + E \rightarrow T + T \rightarrow$
 $T + T * T \rightarrow n + n * n$

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

$$P = \begin{cases} E \rightarrow T \\ E \rightarrow E + E \\ E \rightarrow E - E \\ T \rightarrow F \\ T \rightarrow F * F \\ T \rightarrow F / F \\ F \rightarrow -F \\ F \rightarrow +F \\ F \rightarrow (E) \\ F \rightarrow n \end{cases}$$

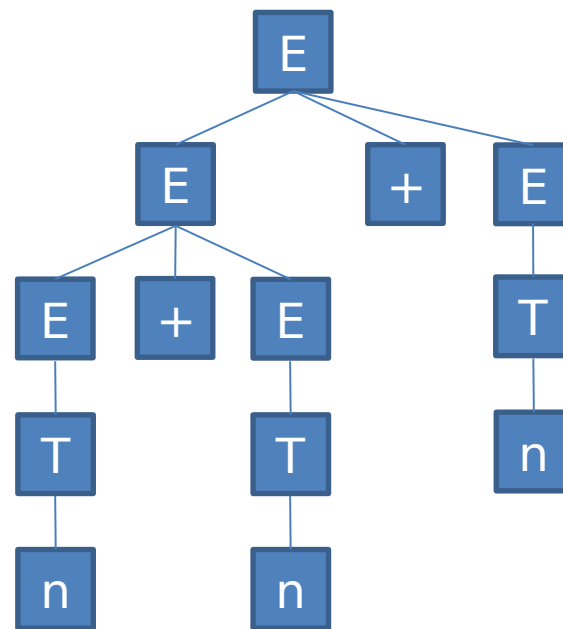
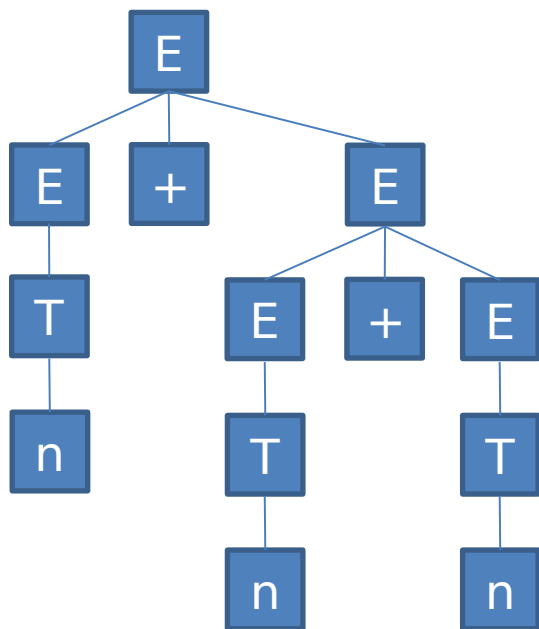
Formatado em Árvore:



Ambiguidade

Teste: $n + n + n$

Ambiguidade: Consigo derivar duas árvores diferentes para a mesma expressão:



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