



Análise Sintáctica

Compiladores, Aula N^o 15
João M. P. Cardoso

1

Aula 15



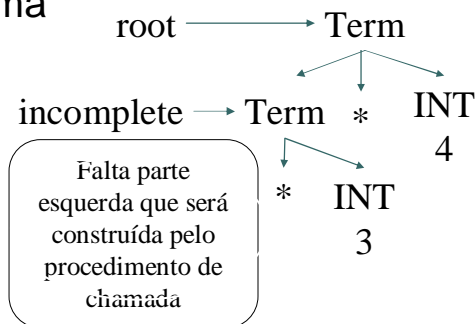
Geração Directa da AST

- TermPrime constrói uma árvore incompleta

- Falta *leftmost child*
- Retorna a raiz e o nó incompleto

- $(\text{root}, \text{incomplete}) = \text{TermPrime}()$

- Chamada com token: *
- Tokens que faltam: 3 * 4



2

Aula 15

Código para Term

```
Term() {  
    if (token == INT) { ←  
        leftmostInt = token;  
        token = NextToken();  
        (root, incomplete) = TermPrime();  
        if (root == NULL) return leftmostInt;  
        incomplete.leftChild = leftmostInt;  
        return root;  
    } else throw SyntaxError;  
}
```

Entrada

2*3*4
↑

token → INT
2

3

Aula 15

Código para Term

```
Term() {  
    if (token == INT) { ←  
        leftmostInt = token;  
        token = NextToken();  
        (root, incomplete) = TermPrime();  
        if (root == NULL) return leftmostInt;  
        incomplete.leftChild = leftmostInt;  
        return root;  
    } else throw SyntaxError;  
}
```

Entrada

2*3*4
↑

token → INT
2

4

Aula 15

Código para Term

```
Term() {
  if (token == INT) {
    leftmostInt = token;
    token = NextToken(); ←
    (root, incomplete) = TermPrime();
    if (root == NULL) return leftmostInt;
    incomplete.leftChild = leftmostInt;
    return root;
  } else throw SyntaxError;
}
```

Entrada

2*3*4
↑



5

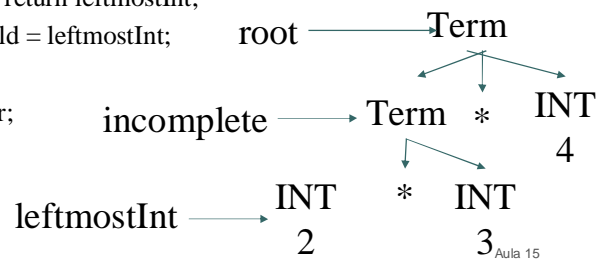
Aula 15

Código para Term

```
Term() {
  if (token == INT) {
    leftmostInt = token;
    token = NextToken();
    (root, incomplete) = TermPrime(); ←
    if (root == NULL) return leftmostInt;
    incomplete.leftChild = leftmostInt;
    return root;
  } else throw SyntaxError;
}
```

Entrada

2*3*4
↑



6

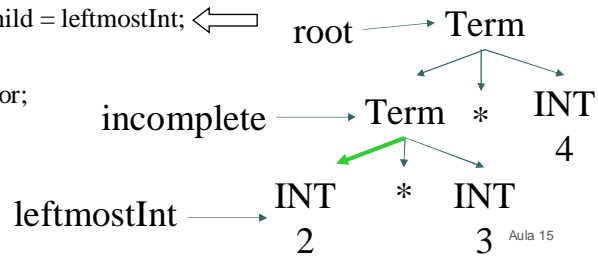
Aula 15

Código para Term

```
Term() {
  if (token == INT) {
    leftmostInt = token;
    token = NextToken();
    (root, incomplete) = TermPrime();
    if (root == NULL) return leftmostInt;
    incomplete.leftChild = leftmostInt;
    return root;
  } else throw SyntaxError;
}
```

Entrada

2*3*4



7

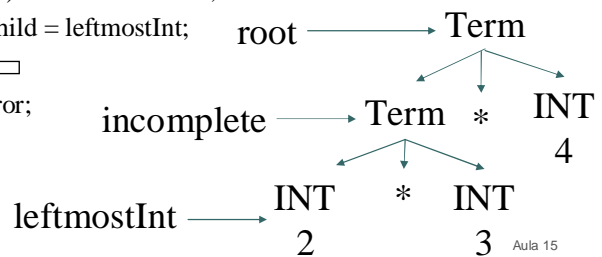
Aula 15

Código para Term

```
Term() {
  if (token == INT) {
    leftmostInt = token;
    token = NextToken();
    (root, incomplete) = TermPrime();
    if (root == NULL) return leftmostInt;
    incomplete.leftChild = leftmostInt;
    return root;
  } else throw SyntaxError;
}
```

Entrada

2*3*4



8

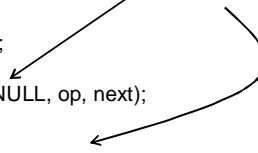
Aula 15



Código para TermPrime

```
TermPrime() {  
    if((token == "*" || (token == '/') {  
        op = token;  
        next = NextToken();  
        if (next == INT) {  
            token = NextToken();  
            (root, incomplete) = TermPrime();  
            if (root == NULL) {  
                root = new ExprNode(NULL, op, next);  
                return(root, root);  
            } else {  
                newChild = new ExprNode(NULL, op, next);  
                incomplete.leftChild = newChild;  
                return(root, newChild);  
            }  
        } else throw SyntaxError;  
    } else return(NULL,NULL);  
}
```

Filho da esquerda a ser colocado pelo procedimento de chamada



9

Aula 15



Sumário

- Analisador sintático descendente (*Top-Down Parser*)
- Utilizar *Lookahead* para evitar *Backtracking*
- O *parser* é um conjunto de procedimentos mutuamente recursivos

10

Aula 15



Terminologia

- Muitas técnicas de análise sintáctica diferentes
 - Cada uma pode manusear algum conjunto de CFGs (gramáticas independentes do contexto)
 - Categorização das técnicas



11

Aula 15



Terminologia

- Muitas técnicas de análise sintáctica diferentes
 - Cada uma pode manusear algum conjunto de CFGs (gramáticas independentes do contexto)
 - Categorização das técnicas
 - **L** – análise da esq. para a drt.
 - **R** - análise da drt. para a esq.



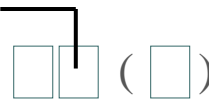
12

Aula 15



Terminologia

- Muitas técnicas de análise sintáctica diferentes
 - Cada uma pode manusear algum conjunto de CFGs (gramáticas independentes do contexto)
 - Categorização das técnicas
 - **L** – derivação pela esquerda
 - **R** – derivação pela direita



13

Aula 15



Terminologia

- Muitas técnicas de análise sintáctica diferentes
 - Cada uma pode manusear algum conjunto de CFGs (gramáticas independentes do contexto)
 - Categorização das técnicas
 - Número de lookahead



14

Aula 15

Terminologia

- Muitas técnicas de análise sintáctica diferentes
 - Cada uma pode manusear algum conjunto de CFGs (gramáticas independentes do contexto)
 - Categorização das técnicas
 - Exemplos: LL(0), LR(1)
 - Até agora: LL(1)
 - Nas próximas aulas
 - Análises LR(k)

LL(k)

15

Aula 15

Terminologia

- LL(k)
 - Descendente (*top-down*), preditiva
 - Constrói derivação pela esquerda (*leftmost*) de cima para baixo
- LR(k)
 - Ascendente (*bottom-up*), *shift-reduce*
 - Constrói derivação pela direita (*rightmost*) de baixo para cima

16

Aula 15