



Análise Sintáctica

Compiladores, Aula N^o 14
João M. P. Cardoso

1

Aula 14



Gramática Modificada

Pedaço da gramática
original

$\text{Term} \rightarrow \text{Term} \text{ "*" } \text{INT}$
 $\text{Term} \rightarrow \text{Term} \text{ "/" } \text{INT}$
 $\text{Term} \rightarrow \text{INT}$

Pedaço da gramática
modificada

$\text{Term} \rightarrow \text{INT Term}'$
 $\text{Term}' \rightarrow \text{"*" } \text{INT Term}'$
 $\text{Term}' \rightarrow \text{"/" } \text{INT Term}'$
 $\text{Term}' \rightarrow \epsilon$

2

Aula 14

● ● ● | Analisador Sintático Construído Manualmente

- Um procedimento por símbolo não-terminal
- Esse procedimento examina o símbolo corrente de entrada
- Chama recursivamente procedimentos para RHS da produção escolhida
- Os procedimentos retornam verdadeiro se a análise sintática foi bem sucedida e retornam falso em caso contrário

3 Aula 14

● ● ● | Exemplo

Produções para o símbolo não-terminal Term:
Term → INT Term'

○ Procedimento para o símbolo não-terminal Term:

```
Term() {  
  if (token == INT) {  
    token = NextToken();  
    return TermPrime();  
  } else return false;  
}
```

A função NextToken() avança um token na sequência de tokens gerada pela análise lexical e retorna o token nessa posição.

4 Aula 14



Exemplo

- o Procedimento para o símbolo não-terminal Term':

```
TermPrime() {  
    if((token == '*' || (token == '/')) {  
        token = NextToken();  
        if (token == INT) {  
            token = NextToken();  
            return TermPrime();  
        }  
    }  
}
```

5

Aula 14



Exemplo

- o Pseudo-código para a parte do programa responsável pela análise sintática:

```
...  
token = NextToken();  
Term();  
...
```

6

Aula 14

Construção da Árvore Sintáctica

- Cada procedimento retorna a secção da árvore sintáctica para a parte da String que analisou
- Utilizar excepções para tornar clara a estrutura do código (outra forma será utilizar uma função de erro)
- Em geral, podemos ajustar o algoritmo de análise sintáctica para satisfazer objectivos diferentes
 - Tipicamente, produz AST em vez de árvore sintáctica concreta

7 Aula 14

Construção da Árvore Sintáctica para o exemplo

○ Com geração de excepções:

```
Term() {  
    if (token == INT) {  
        oldToken = token;  
        token = NextToken();  
        node = TermPrime();  
        if (node == NULL) return oldToken;  
        else return new TermNode(oldToken, node);  
    } else throw SyntaxError;  
}
```

8 Aula 14

Construção da Árvore Sintáctica para o exemplo

- o Com geração de excepções:

```

TermPrime() {
  if ((token == '*') || (token == '/')) {
    first = token;
    next = NextToken();
    if (next == INT) {
      token = NextToken();
      return new TermPrimeNode(first, next, node);
    } else throw SyntaxError;
  } else return NULL;
}

```

9 Aula 14

Construção da Árvore Sintáctica para o exemplo

- o Sem geração de excepções

```

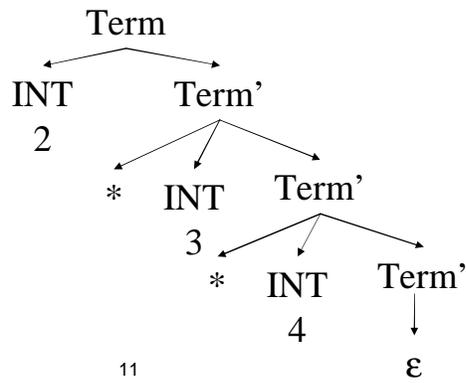
Term() {
  if (token == INT) {
    oldToken = token;
    token = NextToken();
    node = TermPrime();
    if (node == NULL) return oldToken;
    else return new TermNode(oldToken, node);
  } else error();
}
TermPrime() {
  if ((token == '*') || (token == '/')) {
    first = token; next = NextToken();
    if (next == INT) {
      token = NextToken();
      return new TermPrimeNode(first, next, TermPrime());
    } else error();
  } else return NULL;
}

```

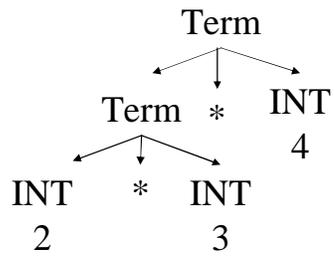
10 Aula 14

Árvore Sintáctica para 2*3*4

Árvore Sintáctica
Concreta



AST desejada



Aula 14