

Aula 08: Tópico 06 - Análise Sintática Conclusão

Melhor tipo de ASA: LALR

Melhor tipo de ASD: ALL(*) – Usando Antlr

```
2 token prox() {  
3  
4 S -> c A d  
5 A -> a b A | c  
6  
7 void S() {  
8     match("c");  
9     A();  
10    match("d");  
11 }  
12  
13 void A() {  
14     if(prox() == "a") {  
15         match("a");  
16         match("b");  
17         A();  
18     } else if(prox() == "c") {  
19         match("c");  
20     } else {  
21         // erro sintático  
22     }  
23 }
```

Criado projeto Java com o Ant

Alguns métodos importantes:

```
private void lerToken() {  
    if (bufferTokens.size() > 0) {  
        bufferTokens.remove(0);  
    } while (bufferTokens.size() < TAMANHO_BUFFER && !chegouNoFim) {  
        Token proximo = lex.proximoToken();  
        bufferTokens.add(proximo);  
        if (proximo.nome == TipoToken.Fim) {  
            chegouNoFim = true;  
        } } System.out.println("Lido: " + lookahead(1));  
}
```

```
void match(TipoToken tipo) {  
    if (lookahead(1).nome == tipo) {  
        System.out.println("Match: " + lookahead(1));  
        lerToken();  
    } else {  
        erroSintatico(tipo.toString());  
    } }  
}
```

```
Token lookahead(int k) {  
    if (bufferTokens.isEmpty()) {  
        return null;  
    } if (k - 1 >= bufferTokens.size()) {  
        return bufferTokens.get(bufferTokens.size() - 1);  
    } return bufferTokens.get(k - 1); }  
}
```

Método usado para decidir se o quarto símbolo é variável ou se terminou a declaração, da regra

```
programa : ':' 'DECLARACOES' listaDeclaracoes ':' 'ALGORITMO' listaComandos;  
programa : ':' 'DECLARACOES' listaDeclaracoes ':' 'ALGORITMO'  
listaComandos; void listaDeclaracoes() {  
    if (lookahead(4).nome == TipoToken.Delim) {  
        declaracao();  
    }  
}
```

```
} else if (lookahead(4).nome == TipoToken.Var) {  
    declaracao();  
    listaDeclaracoes();  
} else {  
    erroSintatico(TipoToken.Delim.toString(), TipoToken.Var.toString());  
} }
```

É apresentado como é feito as regras de análise sintática pela linguagem Java.

Feito testes no final do vídeo 01:03:00 ~