

# **20222 CET058-CMP**

## **Proj2b**

Entrega: 17 de outubro

# **Sumário**

- 1. Analisador Descendente Recursivo**
- 2. Árvore n-aria em vetor**
- 3. GLC LL[1] para o projeto**
- 4. Relatório**

# 1. Analisador Descendente Recursivo

- Implementar, na linguagem C, um analisador descendente recursivo (veja cap.4 do livro de Sebesta) para a gramática de nosso projeto. O analisador deve emitir uma mensagem dizendo se o programa foi aceito ou se aconteceu um erro sintático e, em qualquer caso, deve ser impressa a lista de produções envolvidas na análise e a árvore de análise n-aria, codificada num vetor de inteiros.

## 2. Árvore n-aria em vetor

- Binária:  $\text{esq}(i) = 2i+1$ ,  $\text{dir}(i) = 2i+2$ ,

- Ternária :

$$\text{esq}(i) = 3i+1, \text{esq}(i) = 3i+2, \text{dir}(i) = 3i+3,$$

- Quaternária:

$$d1(i) = 4i+1, d2(i) = 4i+2, d3(i) = 4i+3, d4(i) = 4i+4,$$

- Penta-ária

$$d1(i) = 5i+1, d2(i) = 5i+2, d3(i) = 5i+3, d4(i) = 5i+4, \\ d5(i) = 5i+5$$

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

### 3. GLC LL[1] para o projeto

- $p_1: S \rightarrow M \mid G M \mid F G M$
- $p_6: M \rightarrow m() \{ C; r(E); \}$
- $p_7: E \rightarrow 0 \mid 1 \mid x \mid y \mid (EXE)$
- $p_{12}: X \rightarrow + \mid - \mid * \mid /$
- $p_{16}: C \rightarrow h=E \mid i=E \mid j=E \mid k=E$   
 $\mid z=E \mid (EXE) \mid w(E) \{ C; \} \mid$   
 $f(E) \{ C; \} \mid o(E; E; E) \{ C; \}$

# **F G M**

- $p_4: \mathbf{F} \rightarrow \mathbf{f}() \{ \mathbf{C}; \mathbf{r}(\mathbf{E}); \}$
- $p_5: \mathbf{G} \rightarrow \mathbf{g}() \{ \mathbf{C}; \mathbf{r}(\mathbf{E}); \}$
- $p_6: \mathbf{M} \rightarrow \mathbf{m}() \{ \mathbf{C}; \mathbf{r}(\mathbf{E}); \}$

$p_6: \mathbf{M} \rightarrow \mathbf{m} ( ) \{ \mathbf{C}; \mathbf{r} ( \mathbf{E} ) ; \}$

$p_6: \mathbf{M} \rightarrow \mathbf{main} ( ) \{$   
                     $\mathbf{COMMAND};$   
                     $\mathbf{return} ( \mathbf{E} ) ;$   
                     $\}$

**w (E) { C; }**

**while (EXPR) {  
    COMMAND;  
}**



**f (E) { C; }**

**if (EXPR) {  
    COMMAND;  
}**

○ (E ; E ; E) { C ; }

```
for (EXPR; EXPR; EXPR) {  
    COMMAND;  
}
```

# 4. Relatório

- O relatório deve conter:
  - Folha de rosto
  - Sumário
  - Link para download da implementação.
  - Comando para compilação
  - Comando para a execução.
  - Saída para programas exemplo
  - Referências.