

Analizador sintático e léxico para a linguagem C- - Demonstrando o software

Victor E. Almeida Marco A. G. Pedroso

UNIOESTE

11 de abril de 2022

Conteúdo

- 1 Introdução
- 2 Bison
- 3 Analizador sintático
- 4 Conclusão

Atividades realizadas

- **Analisador sintático:** Victor E.;
- **Analisador léxico:** Victor E.;
- **funções auxiliares como log:** Victor E.;
- **Geração da árvore sintática:** Marco G.;
- **Geração dos slides:** Marco G.;
- **Geração do README:** Marco G.;
- **Geração dos arquivos fonte para entrada:** Victor E.;

Softwares utilizados

- **Sistema para compilar:** GNU make 4.3 e gcc 11.2,
- **Linguagem de programação:** C11,
- **Gerador de analisador léxico:** GNU flex 2.6.4;
- **Gerador de analisador sintático:** GNU bison 3.8.2.

Exemplo mínimo do bison

```
1      %{
2          definicoes C
3      %}
4      %token algum_token
5      %start simbolo_inicial
6
7      %%
8      simbolo_inicial: algum_token ';'
9      %%
10
11      int main() {
12          return yyparse();
13      }
```

Exemplo de regra do bison

```
1 keyword:
2     TOKEN_KEYWORD_IF
3     | TOKEN_KEYWORD_ELSE
4     | TOKEN_KEYWORD_CONST
5     | TOKEN_KEYWORD_FOR
6     | TOKEN_KEYWORD_WHILE
7     | TOKEN_KEYWORD_RETURN
8     | TOKEN_KEYWORD_STRUCT
9     ;
```

Integrando Flex e bison

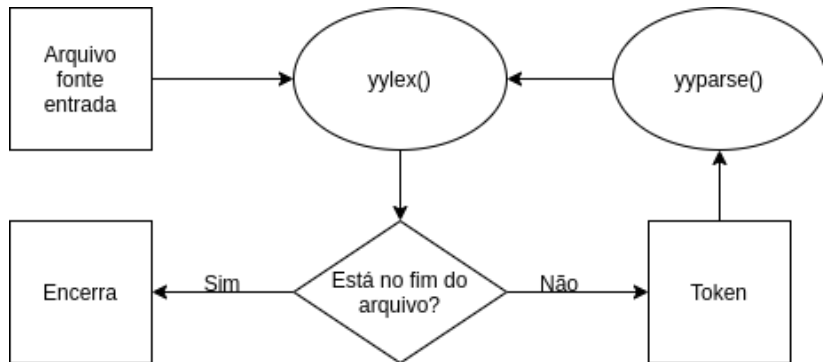


Figura 1: Fluxograma do compilador

Lista de algumas regras gramaticais I

- $\langle \text{programa} \rangle ::=$
 $\text{TOKEN_PREPROCESSOR_COMMAND } \langle \text{programa} \rangle \mid$
 $\langle \text{definition} \rangle \langle \text{programa} \rangle \mid \lambda$
- $\langle \text{keyword} \rangle ::=$ $\text{TOKEN_KEYWORD_IF} \mid$
 $\text{TOKEN_KEYWORD_ELSE} \mid$
 $\text{TOKEN_KEYWORD_CONST} \mid$
 $\text{TOKEN_KEYWORD_FOR} \mid$
 $\text{TOKEN_KEYWORD_WHILE} \mid$
 $\text{TOKEN_KEYWORD_RETURN} \mid$
 $\text{TOKEN_KEYWORD_STRUCT}$

Lista de algumas regras gramaticais II

- $\langle \text{literal} \rangle ::= \text{TOKEN_INTEGER_LITERAL} \mid \text{TOKEN_FLOAT_LITERAL} \mid \text{TOKEN_CHAR_LITERAL} \mid \text{TOKEN_STRING_LITERAL}$
- $\langle \text{value} \rangle ::= \text{TOKEN_ID} \mid \text{TOKEN_INTEGER_LITERAL} \mid \text{TOKEN_FLOAT_LITERAL}$
- $\langle \text{exp} \rangle ::= \langle \text{complete_variable} \rangle \langle \text{exp} \rangle \mid \langle \text{ifel} \rangle \langle \text{exp} \rangle \mid \langle \text{atribuition} \rangle \langle \text{exp} \rangle \mid \langle \text{for} \rangle \langle \text{exp} \rangle \mid \langle \text{while} \rangle \langle \text{exp} \rangle \mid \lambda$

Mão na massa!!



Agradecimentos

Perguntas?



Obrigado pela atenção