

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 lex/Flex
- 3 Analisador léxico
- 4 Conclusão

Softwares utilizados

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

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     }
```

Funcionamento do lex/flex

```
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     ;
```

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

- `<literal> ::= TOKEN_INTEGER_LITERAL |
TOKEN_FLOAT_LITERAL | TOKEN_CHAR_LITERAL |
TOKEN_STRING_LITERAL`
- `conditional: value | value
TOKEN_RELATIONAL_OPERATOR value`

Mão na massa!!



Agradecimentos

Perguntas?



Obrigado pela atenção