

Compiladores

Análise Léxica

Flex – Tiny

Profa. Dra. Ana Carolina Lorena
UNIFESP

Flex – Gerador de Analisador Léxico

- Gera sistema de varredura a partir de expressões regulares
 - Implementação de AFD baseada em tabelas

Flex – Gerador de Analisador Léxico

- Convenções de metacaracteres em *Lex*

Tabela 2.2 Convenções de metacaracteres em Lex

Padrão	Significado
<i>a</i>	caractere <i>a</i>
" <i>a</i> "	caractere <i>a</i> , mesmo se <i>a</i> for um metacaractere
\ <i>a</i>	caractere <i>a</i> se <i>a</i> for um metacaractere (pela interpretação ANSI –C)
<i>a</i> *	zero ou mais repetições de <i>a</i>
<i>a</i> +	uma ou mais repetições de <i>a</i>
<i>a</i> ?	um <i>a</i> opcional
<i>a</i> <i>b</i>	<i>a</i> ou <i>b</i>
(<i>a</i>)	<i>a</i> propriamente dito
[<i>abc</i>]	qualquer caractere entre <i>a</i> , <i>b</i> e <i>c</i>
[<i>a–d</i>]	qualquer caractere entre <i>a</i> , <i>b</i> , <i>c</i> e <i>d</i>
[^ <i>ab</i>]	qualquer caractere, exceto <i>a</i> ou <i>b</i>
.	qualquer caractere, exceto mudança de linha
{ <i>xxx</i> }	a expressão regular representada pelo nome <i>xxx</i>

Flex – Gerador de Analisador Léxico

- Formato do arquivo de entrada *Lex*

%{ DEFINIÇÕES
qualquer código C a ser inserido externamente a
qualquer função opcional

%}
definições regulares opcional

%% REGRAS
expressões regulares seguidas do código C a ser executado quando
houver
casamento com a expressão regular correspondente

%% ROTINAS AUXILIARES
código C para rotinas auxiliares ativadas pela segunda seção, pode
também
conter um programa principal opcional

Flex – Gerador de Analisador Léxico

- Nomes internos utilizados por *Flex*

Tabela 2.3 Alguns nomes internos Lex

Nome Interno Lex	Significado/Utilização
<code>lex.yy.c</code> or <code>lexyy.c</code>	Arquivo de saída Lex
<code>yylex</code>	Rotina de varredura Lex Funciona como <code>getToken</code>
<code>yytext</code>	Cadeia casou com ação corrente
<code>yyin</code>	Entrada Lex (padrão: <code>stdin</code>)
<code>yyout</code>	Saída Lex (padrão: <code>stdout</code>)
<code>input</code>	Rotina de entrada com reservatório Lex
<code>ECHO</code>	Ação básica Lex (imprime <code>yytext</code> em <code>yyout</code>)

A documentação completa de *FLEX* está disponível em <http://westes.github.io/flex/manual/>

`yytext` costuma ser usada para guardar o lexema

Flex

- Resolvendo ambiguidade:
 - ▮ Flex sempre casará a subcadeia mais longa com a regra
 - ▮ Se duas ou mais regras casarem com subcadeias de mesmo comprimento, Flex selecionará a regra citada antes
 - ▮ Se nenhuma regra casa, Flex copia o caractere na saída e segue em frente

A Linguagem *TINY*

- Estrutura léxica da linguagem *TINY*

Tabela 2.1 Marcas da linguagem TINY

Palavras Reservadas	Símbolos Especiais	Outras
<i>if</i>	+	<i>número</i>
<i>then</i>	-	(1 ou mais
<i>else</i>	*	dígitos)
<i>end</i>	/	
<i>repeat</i>	=	
<i>until</i>	<	<i>identificador</i>
<i>read</i>	((1 ou mais
<i>write</i>)	letras)
	;	
	::=	

Flex – Linguagem Tiny

```
%{
char tokenString[MAXTOKENLEN+1]; // lexema identificador
%}
digit [0-9]
number {digit}+
letter [a-zA-Z]
id {letter}+
newline \n
space [ \t]+
%%
"if" {return IF;}
"then" {return THEN;}
"else" {return ELSE;}
"end" {return END;}
"repeat" {return REPEAT;}
"until" {return UNTIL;}
"read" {return READ;}
"write" {return WRITE;}
":=" {return ASSIGN;}
"=" {return EQ;}
"<" {return LT;}
"+" {return PLUS;}
```


Flex – Linguagem Tiny

```
"-" {return MINUS;}
"*" {return TIMES;}
"/" {return OVER;}
"(" {return LPAREN;}
")" {return RPAREN;}
";" {return SEMI;}
{number} {return NUM;}
{id} {return ID;}
{newline} {lineno++;}
{space} {/* skip */}
"{\" { char c;
    do{
        c = input();
        if(c == '\\n') lineno++;
    }while(c!='}');
}
{return ERROR}
%%
```

Flex – Linguagem Tiny

```
TokenType getToken(void) {
    static int firstTime = TRUE;
    TokenType currentToken;
    if(firstTime){
        FirstTime = FALSE;
        Lineno++;
        yyin = source;
        yyout = listing;
    }
    CurrentToken = yylex();
    strncpy(tokenString, yytext, MAXTOKENLEN);
    if(TraceScan){
        fprintf(listing, "\t%d: ", lineno);
        printToken(currentToken, tokenString);
    }
    return currentToken;
}
```

Flex – Gerador de Analisador Léxico

- Bibliografia consultada

Capítulo 2 e Apêndice B - LOUDEN, K. C. **Compiladores: princípios e práticas**. São Paulo: Pioneira Thompson Learning, 2004

RICARTE, I. **Introdução à Compilação**. Rio de Janeiro: Editora Campus/Elsevier, 2008.