UNISINOS - **UNIVERSIDADE DO VALE DO RIO DOS SINOS**
CEN CIÊNCIAS EXATAS E TECNOLÓGICAS – Curso de Informática

COMPILADORES I

Disciplina: Compiladores I
Professor responsável: ***Fernando Santos Osório***
Semestre: 2004/1
Horário: 63

E-mail: ***osorio@exatas.unisinos.br***
Web: ***http://inf.unisinos.br/~osorio/compil.html***
Xerox : ***Pasta 68*** **– (Xerox ao lado doLAPRO)**

GERADORES DE ANALISADORES LÉXICOS – LEX / FLEX

O **LEX / FLEX** servem para gerar automaticamente programas (usualmente em "C") fazendo a leitura de uma entrada, de modo a varrer um texto e/ou programa ("scanners") a fim de obter uma seqüência de unidades léxicas ("tokens"). Os tokens gerados pelos programas criados pelo LEX/FLEX serão usualmente processados posteriormente por um programa que realizará a analise sintática.

Lex => Gerador de analisadores léxicos (UNIX – Ex.: Lex AT&T, Berkeley BSD)
Flex => Gerador de analisadores léxicos (LINUX / Windows-DOS – GNU Lex)

Entrada: Arquivo de descrição do analisador léxico
Saída: Programa na linguagem "C" que realiza a análise léxica (default: lexyy.c)

Outros geradores de analisadores:
TPly – TP Lex / Yacc => Gera um programa em PASCAL (scanner em Pascal)
JavaCC => Para linguagem Java
Flex++ ou Flexx => Para linguagem C++ (orientado a objetos)

Usando o FLEX:

1. FLEX -o<arq_saida.c> <arq_def>.l
   *.l => Arquivos que contêm as definições das unidades léxicas a serem identificadas
   Contém um conjunto de especificações de ***expressões regulares*** que serão usadas para reconhecer os tokens.
   *.c => Arquivo do programa "C" que implementa o analisador léxico especificado.
   Principais opções do FLEX:
   -i => Case Insensitive (ignora diferença entre maiúscula/minúsculas)
   --version => Exibe a versão atual do programa flex em uso
   -+ => Geração de código de saída em C++
2. GCC <arq_saida.c> -o <arq_executavel> -lfl
   Observações importantes sobre a geração do programa de análise léxica:
   - É necessário linkar (-l) uma bibliotec ("lib") do analisador léxico na compilação do código gerado. O FLEX usa a lib "fl" e o LEX usa a lib "l".
   - O programa gerado pode ser executado, onde usualmente a entrada do texto a ser analisado é feita pela "stdin" (teclado).
3. Executar o programa gerado...

EXPRESSÕES REGULARES – Usada pelo LEX / FLEX:

[0-9] => Reconhece um dígito
[a-zA-Z] => Reconhece uma letra (comum = sem acentos)
[\ \t\n] => Reconhece um espaço em branco ou um tab ou uma nova linha
xxxxx => Reconhece a seqüência de caracteres "xxxxx"

Símbolos especiais:		Exemplo:	
+	=> 1 ou mais ocorrências	[0-9]+	=> Um número
*	=> 0 (nenhuma) ou mais ocorrências	[0-9][0-9]*	=> Um número
?	=> 0 (nenhuma) ou apenas 1 ocorrência	-?[0-9]+	=> Um número com/sem sinal

\n => Reconhece a marca de fim de linha / nova linha
. => Aceita um caracter qualquer de entrada
xxx$ => Reconhece xxx se for seguido de um fim de linha
^xxx => Reconhece xxx se este estiver imediatamente após o início de uma linha
[^x] => Reconhece qualquer caracter menos "x"
[xyz] => Reconhece um dos caracteres "xyz" indicados
[a-z] => Reconhece um caracter pertencente ao intervalo de "a-z"
x{n} => Reconhece um número exato "n" de ocorrência de "x"
x{n,} => Reconhece a ocorrência de no mínimo "n" vezes de "x"
x{n,m}=> Reconhece a ocorrência de "x" entre no mínimo "n" e no máximo "m" vezes
xx|yy => Reconhece a ocorrência de "xx" ou de "yy"
(x|y) => Agrupa (sub)expressões regulares
"x" => Reconhece exatamente o caracter "x" (usado com caracteres especiais). Ex.: "+"

Exemplos de expressões regulares simples:

DIGITO	[0-9]	
LETRA	[a-zA-Z]	
ESPACO	[\ \t\n]	
INTEIRO	[0-9]+	
INTSIGNED	-?[0-9]+	
DECIMAL	[0-9]*\.[0-9]+	=> aceita .33 / não aceita números sem casas decimais
INTOUDEC	([0-9]+)|([0-9]*\.[0-9]+)	
IOUDSIGNED	-?(([0-9]+)|([0-9]*\.[0-9]+))	
NOMEVAR	[a-z][a-z0-9_]*	=> usando opção "case insensitive"...

Exercícios: Defina as expressões regulares capazes de reconhecer...

1) Nros. de Telefones no Brasil
2) Placas de Carros Brasileiros
3) ISBN de um livro
4) Endereços IP válidos
5) Prefixos de estações de rádio (e.g. 102.3 MHz)
6) Números romanos
7) Número de matrícula da Unisinos
8) Números reais (qualquer notação, incluindo científica)
9) Tags HTML (padrão)
10) URL de páginas Web
11) Palavras da Língua Portuguesa
12) Strings de um programa em linguagem "C"