Regular Expressions and Lex

EECS 665 Compiler Construction

Dr. Prasad Kulkarni
Marianne Jantz

Regular Expressions

- Given an alphabet Σ
  - ε is a regular expression that denotes {ε}, the set containing the empty string
  - For each a ϵ Σ, a is a regular expression denoting {a}, the set containing the string a
  - r and s are regular expressions denoting languages L(r) and L(s). Then,

(r)|(s)	denotes L(r) υ L(s)
(r)(s)	denotes L(r) L(s)
(r)*	denotes (L(r))*

Regular Expressions (cont.)

- * has the highest precedence, left associative
- concatentation has the second highest precedence, left associative
- | has the lowest precedence, left associative
- Example:

  a | bc* = a|(b(c*))

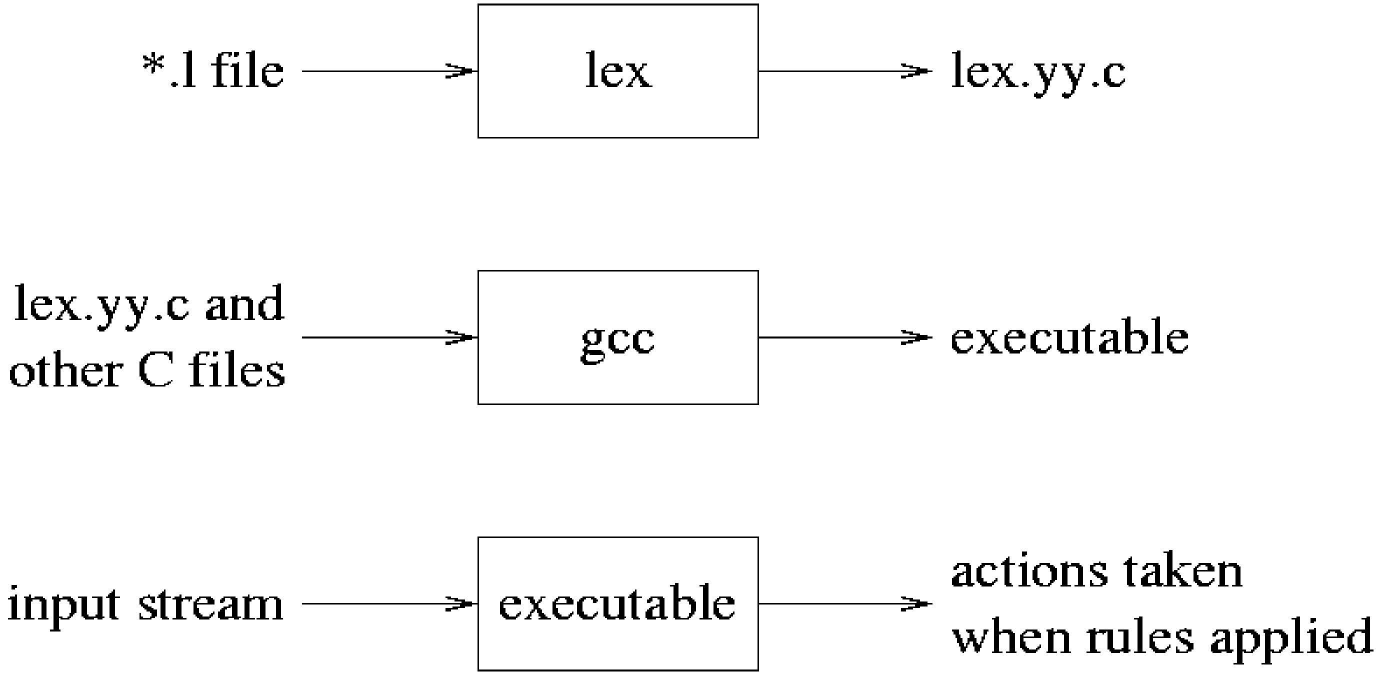
Examples of Regular Expressions

- Let Σ = {a, b}

a | b	=> {a, b}
(a | b) (a | b)	=> {aa, ab, ba, bb}
a*	=> {ε, a, aa, aaa, ...}
(a | b)*	=> all strings containing zero or more instances of a's and b's
a | a * b	=> {a, b, ab, aab, aaab, ...}

Lex – A Lexical Analyzer Generator

- Can link with a lex library to get a main routine
- Can use as a function called yylex()
- Easy to interface with yacc



Lex – A Lexical Analyzer Generator (cont.)

Lex Source

```
{ definitions }
%%
{ rules }
%%
{ user subroutines }
```

Lex – A Lexical Analyzer Generator (cont.)

Definitions

- Declarations of variables, constants, and regular definitions

Rules

- regular expression   action

Regular Expressions

- operators   “\[]^-?.*+|()$/{}

Actions

- C code

Lex Regular Expression Operators

“s”	string s literally
\c	character c literally (used when c would normally be used as a lex operator)
[s]	for defining s as a character class
^	to indicate the beginning of a line
[^s]	means to match characters not in the s character class
[a-b]	used for defining a range of characters, a to b, in a character class
r?	means that r is optional

Lex Regular Expression Operators (cont.)

.	means any character but a newline
r*	means zero or more occurrences of r
r+	means one or more occurrences of r
r1|r2	r1 or r2
(r)	r (used for grouping)
$	means the end of the line
r1/r2	means r1 when followed by r2
r{m,n}	means m to n occurrences of r

Example Regular Expressions in Lex

• a*	zero or more a's
• a+	one or more a's
• [abc]	a, b, or c
• [a-z]	lower case letter
• [a-zA-Z]	any letter
• a.b	a followed by any character followed by b
• ab|cd	ab or cd
• a(b|c)d	abd or acd
• ^B	B at the beginning of line
• E$	E at the end of the line

More on Lex

Actions

Actions are C source fragments. If it is a compound or takes more than one line, then it should enclosed in braces

Example Rules

```
[a-z]+              printf(“found word\n”);
[A-Z][a-z]*         {   printf(“found capitalized word\n”);
                        printf(“ %s\n”, yytext);
                    }
```

Definitions

name	translation

Example Definition

```
digits              [0-9]
```

Example Lex Program

```
digits              [0-9]
ltr                 [a-zA-Z]
alpha               [a-zA-Z0-9]
%%
[-+]{digits}+   |
{digits}+           printf(“number: %s\n”, yytext);
{ltr}(_|{alpha})*   printf(“identifier: %s\n”, yytext);
"‘"."’"             printf(“character: %s\n”, yytext);
.                   printf(“?: %s\n”, yytext);
```

Prefers longest match and earlier of equals.

Lex References

- Your Compilers Textbook
- The Lex man page

  http://plan9.bell-labs.com/magic/man2html/1/lex
- A Lex Online Manual

  http://dinosaur.compilertools.net/lex/index.html
- Linux Documentation Project Lex and Yacc Tutorial

  http://tldp.org/HOWTO/Lex-YACC-HOWTO.html

References for using Regular Expressions with Text Editors

- Vim

  http://vim.wikia.com/wiki/Search_and_replace

- Emacs

  GNU Manual

  http://www.emacswiki.org/emacs/RegularExpression

- Google