FLEX : A lexical Analyser generator.
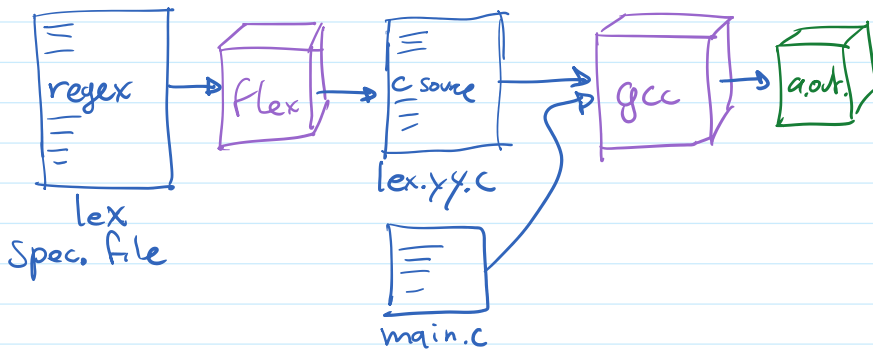
regex → automata → code.
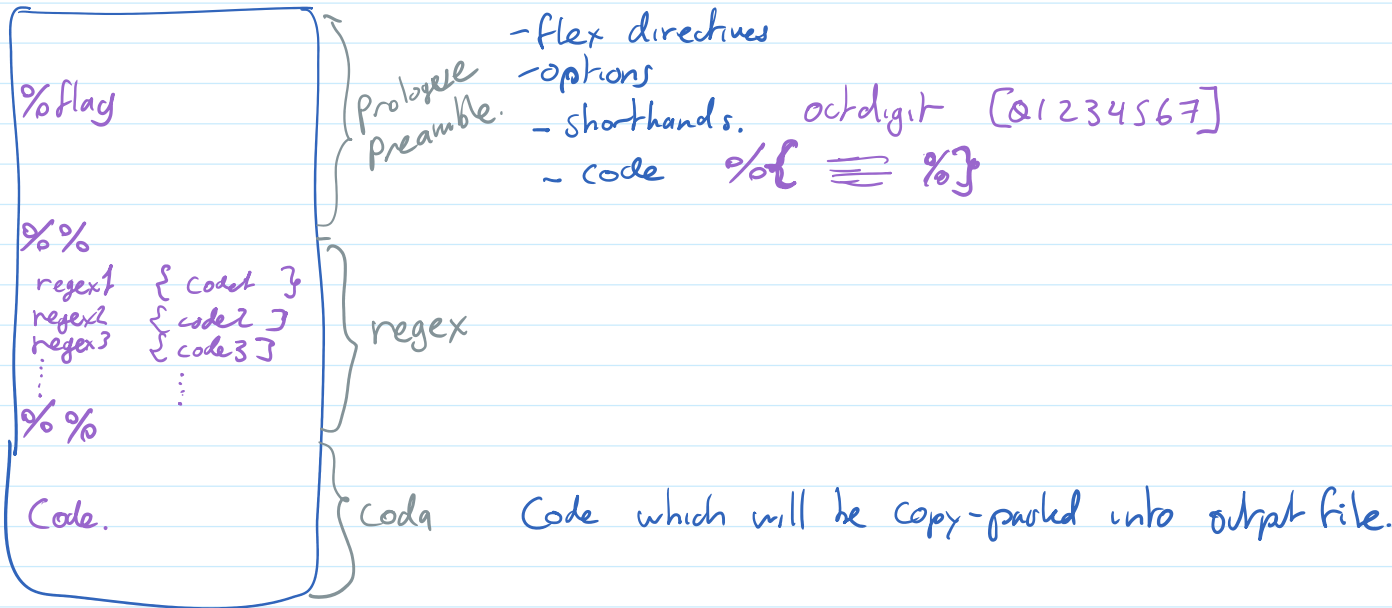
`70 by Mike Lesk
based on Lex by Eric Schmidt. for AT&T UNIX

- uses POSIX standard.

- Flex is a "generator"
  by Vern Paxson for B.S.D.

regex
lex
Spec. file

flex

C Source
lex.yy.c

main.c

gcc

a.out.

- THE FLEX SPECIFICATION FILE:

%flag

%%
regex1   { code1 }
regex2   { code2 }
regex3   { code3 }
...   ...
%%

Code.

Prologue
Preamble.
- flex directives
- options
- shorthands.   octdigit [01234567]
- code   %{ ≡ %}

regex

coda   Code which will be copy-pasted into output file.

- THE CODE GENERATED BY FLEX:

- Variables:
  yytext - ptca the last string matched.

- Variables:
  - yytext :- ntca. the last string matched.
  - yylen :- the length of yytext
  - yylval :- A value attached to yytext.
  - yyin :- The input file.

- Function
  - yylex() :- the lexical analyser.
    implements the automata
  - yywrap() :- called when input is exhausted.
  - yyinput() :- reads one character from the input

- FLEX WORKFLOW:

  $ flex   spec-file   → lex.yy.c

  $ gcc -lfl   lex.yy.c *.c   → a.out

  $ ./a.out.

- FLEX TIE-BRAKERS:

Suppose:

- [0-9]+
- [0-9]+\.[0-9]+

123.27

integer 123
decimal .27

- flex prefers longest
  string over shorter string

- (W|w)hale(s)?
- [a-z A-Z]+

whales

whale
string

- flex prefers the first
  regular expression on file.

—o— EOF