

How to invoke the program

1. Extract the zip file.
 - a. Nishimura.Yohei.P2/
 - i. P2.java
 - ii. b.jlex
 - iii. Makefile
 - iv. ErrMsg.java
 - v. sym.java
 - vi. *.in files (to be used in tests. see Note section.)
 - vii. deps/
 1. JLex/
 - a. *.class files
 2. java_cup/
 - a. (files)
 - b. Nishimura.Yohei.P2.pdf

2. Execute the following

- a. make

3. Expected outputs are below:

- a. javac -g -cp ./deps:. sym.java
- b. java -cp ./deps:. JLex.Main b.jlex
- c. Processing first section -- user code.
- d. Processing second section -- JLex declarations.
- e. Processing third section -- lexical rules.
- f. Creating NFA machine representation.
- g. NFA comprised of 373 states.
- h. Working on character
classes:.....
.....
- i. NFA has 46 distinct character classes.
- j. Creating DFA transition table.
- k. Working on DFA
states.....
- l. Minimizing DFA transition table.
- m. 99 states after removal of redundant states.
- n. Outputting lexical analyzer code.
- o. javac -g -cp ./deps:. ErrMsg.java
- p. javac -g -cp ./deps:. b.jlex.java
- q. javac -g -cp ./deps:. P2.java

4. Execute the following

- a. make test

5. Expected outputs are below:

- a. java -cp ./deps:. P2

b. 1:1 ***WARNING*** integer literal too large; using max value
 c. 1:1 ***ERROR*** unterminated String literal is ignored: "unterminated
 d. 2:1 ***ERROR*** unterminated String literal is ignored: "also \
 e. 3:1 ***ERROR*** improper char with bad escape is ignored: "backslash \
 f. 3:1 ***ERROR*** unterminated String literal is ignored: "
 g. 4:1 ***ERROR*** improper char with bad escape is ignored: "bad escape \
 h. 4:1 ***ERROR*** unterminated String literal is ignored: "
 i. 5:1 ***ERROR*** improper char with bad escape is ignored: "bad escape \
 j. 6:1 ***ERROR*** unterminated String literal is ignored: "unterminated string (eof)
 k. 1:1 ***ERROR*** improper ident starting: 1_
 l. 2:1 ***ERROR*** improper ident starting: 1x
 m. 3:1 ***ERROR*** improper ident starting: 1x_
 n. 4:1 ***ERROR*** improper ident starting: 123x_
 o. 1:1 ***ERROR*** ignoring illegal character: #
 p. 1:2 ***ERROR*** ignoring illegal character: #
 q. 1:1 ***ERROR*** improper symbol; <==
 r. 2:1 ***ERROR*** improper symbol; >==
 s. 3:1 ***ERROR*** improper symbol; !=
 t. 4:1 ***ERROR*** improper symbol; <=====
 u. 5:1 ***ERROR*** improper symbol; !======
 v. 6:1 ***ERROR*** improper symbol; >=====
 w. 7:1 ***ERROR*** improper symbol; +++++
 x. 8:1 ***ERROR*** improper symbol; -----
 y. 9:1 ***ERROR*** improper symbol; ***
 z. 10:1 ***ERROR*** improper symbol; //
 aa. 11:1 ***ERROR*** improper symbol; &&&&
 bb. 12:1 ***ERROR*** improper symbol; ||||
 cc. 13:1 ***ERROR*** improper symbol; <<<<
 dd. 14:1 ***ERROR*** improper symbol; >>>>
 ee. 1:1 ***ERROR*** ignoring illegal character: ~
 ff. 2:1 ***ERROR*** ignoring illegal character: \$
 gg. 3:1 ***ERROR*** ignoring illegal character: %
 hh. 4:1 ***ERROR*** ignoring illegal character: ^
 ii. 5:1 ***ERROR*** ignoring illegal character: &
 jj. 6:1 ***ERROR*** ignoring illegal character: |
 kk. 7:1 ***ERROR*** ignoring illegal character: ~
 ll. 8:1 ***ERROR*** ignoring illegal character: \$
 mm. 9:1 ***ERROR*** ignoring illegal character: %
 nn. 10:1 ***ERROR*** ignoring illegal character: ^
 oo. 11:1 ***ERROR*** ignoring illegal character: &
 pp. 12:1 ***ERROR*** ignoring illegal character: |
 qq. diff validReserved.in validReserved.out

rr. diff validIntegerLiteral.in validIntegerLiteral.out
ss. diff validIdent.in validIdent.out
tt. diff validStringLiteral.in validStringLiteral.out
uu. diff validSymbol.in validSymbol.out
vv. diff validComment_expect.in validComment.out
ww. diff validWhitespace_expect.in validWhitespace.out
xx. diff invalidIntegerLiteral_expect.in invalidIntegerLiteral.out
yy. diff invalidStringLiteral_expect.in invalidStringLiteral.out
zz. diff invalidIdent_expect.in invalidIdent.out
aaa. diff invalidComment_expect.in invalidComment.out
bbb. diff invalidSymbol_expect.in invalidSymbol.out
ccc. diff invalidReserved_expect.in invalidReserved.out

Note

1. *.in files are below. The file name represents the test.
 - a. validReserved.in, invalidReserved.in, invalidReserved_expect.in,
 - b. validIdent.in, invalidIdent.in, invalidIdent_expect.in,
 - c. validIntegerLiteral.in, invalidIntegerLiteral.in, invalidIntegerLiteral_expect.in
 - d. validStringLiteral.in, invalidStringLiteral.in, invalidStringLiteral_expect.in
 - e. validSymbol.in, invalidSymbol.in, invalidSymbol_expect.in
 - f. validComment.in, validComment_expect.in, invalidComment.in, invalidComment_expect.in
 - g. validWhitespace.in, validWhitespace_expect.in