How to invoke the program

1. Extract the zip file.
    a. P3/
        i. ast.java
        ii. b.cup
        iii. b.grammar
        iv. b.jlex
        v. ErrMsg.java
        vi. Makefile
        vii. P3.java
        viii. test.b
        ix. deps/
            1. JLex/
                a. *.class files
            2. java_cup/
                a. (files)
    b. Nishimura.Yohei.Nishimura.Yohei.P3.pdf
2. Execute the following
    a. make
3. Expected outputs are below:

```
------------
java -cp ./deps:. java_cup.Main < b.cup
------- CUP v0.11b ${cupversion} (SVN rev ) Parser Generation Summary -------
  0 errors and 0 warnings
  43 terminals, 22 non-terminals, and 69 productions declared,
  producing 144 unique parse states.
  0 terminals declared but not used.
  0 non-terminals declared but not used.
  0 productions never reduced.
  0 conflicts detected (0 expected).
  Code written to "parser.java", and "sym.java".
---------------------------------------------------- (CUP v0.11b ${cupversion} (SVN rev ))
javac -g   -cp ./deps:. ast.java
javac -g   -cp ./deps:. sym.java
java -cp ./deps:. JLex.Main b.jlex
Processing first section -- user code.
Processing second section -- JLex declarations.
Processing third section -- lexical rules.
Creating NFA machine representation.
NFA comprised of 284 states.
Working on character
classes.::.:::::..:.:::.:.....:..:.....:..........::.......:....::.:..::::.......................:::..:...:::::::.:.:.:....:.::.:.:...........:.
```

NFA has 47 distinct character classes.
Creating DFA transition table.
Working on DFA states.....
Minimizing DFA transition table.
99 states after removal of redundant states.
Outputting lexical analyzer code.
javac -g   -cp ./deps:. ErrMsg.java
javac -g   -cp ./deps:. b.jlex.java
javac -g   -cp ./deps:. parser.java
Note: parser.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
javac -g   -cp ./deps:. P3.java
---

4. Execute the following
    a. make test
5. Expected outputs are below:

---
java -cp ./deps:. P3 test.b test.out
program parsed correctly.

Note

1. Nothing special.