Compiler Construction Project

BCH version

Choi Jung In: xchoij00@stud.fit.vutbr.cz

Viggiano Franco Nicola: xviggi00@stud.fit.vutbr.cz

xchoij00 : 50

xvigg00: 50

Lex

It divides the strings such as 'main', 'int', 'string', and a sign such as '(', '{', '+' into tokens.

It distinguishes block comments by recognizing "/*", and line comments by recognizing "//".

```
id  [a - zA - Z_][0 - 9a - zA - Z_] *
integerliteral  [0 - 9] +
stringliteral    \"x(0-9A-F)(0-9A-F)(0-9A-F)(0-9A-F)(0-9A-F)(0-9A-F)\"
linecomment      "//".*
blockcommentopen    ("/*")
```

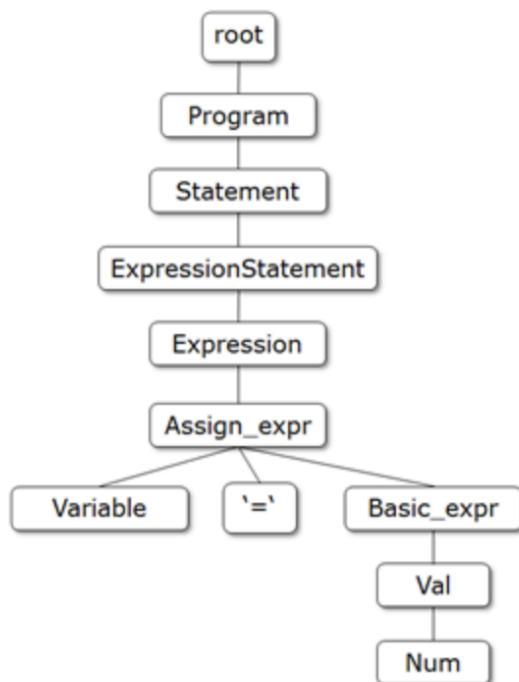| id	recognize variables in characters
integerliteral	recognize integers
stringliteral	recognize strings
linecomment	recognize line comment
blockcommentopen	recognize block comment open

YACC

Create a token of the value recognized and returned by Lex with %token.

Example)

When compiler recognize variable:

(ex. a=1;)

(ex. if(a<5))

```
                          ┌──────┐
                          │ root │
                          └──────┘
                             │
                        ┌─────────┐
                        │ Program │
                        └─────────┘
                             │
                        ┌───────────┐
                        │ Statement │
                        └───────────┘
                             │
                    ┌────────────────────┐
                    │ SelectionStatement │
                    └────────────────────┘
          ┌─────┬──────┬─────────┴──┬────────┬───────────┐
        ┌────┐ ┌───┐ ┌────────────┐ ┌───┐ ┌───────────┐
        │ if │ │ '('│ │ Expression │ │ ')'│ │ Statement │
        └────┘ └───┘ └────────────┘ └───┘ └───────────┘
                            │
                     ┌────────────┐
                     │ Basic_expr │
                     └────────────┘
                 ┌────────┴───┬──────────┐
          ┌────────────┐ ┌─────┐ ┌────────────┐
          │ Basic_expr │ │ '<' │ │ Basic_expr │
          └────────────┘ └─────┘ └────────────┘
                 │                      │
              ┌────┐                 ┌─────┐
              │ ID │                 │ Val │
              └────┘                 └─────┘
```

4