

Sonoma State University
Computer Science Department
CS460 – Fall 2008 - Watts

Project - Part 2 - Parser

For this part of the project you are to write a recursive descent parser that performs syntactical analysis on the source code using the grammar distributed in class.

The functions developed for this parser should be in a separate .cpp file. Necessary types and prototypes should be in an associated .h file.

Input: A source code file. The file name should be accepted as a command line argument.

Output: A listing of the original source code with lexical and syntactical error messages.
A debugging file (filename + .dbg) containing a list of terminal and non-terminal symbols encountered while parsing the program (and other useful debugging information).

Date Due: 4 November 2008; 11:59 pm.

To turn in: Tared and zipped directory containing source files (headers and implementations) and a makefile. Your makefile should create an executable called “proj2”. Your directory should be called yourlastnameP2 and your tarred and zipped file (tar falgz cfvz) file should be called yourlastnameP2.tgz. Submit your .tgz file by copying or moving it to ~tiawatts/cs460drop

CFG for Project

```
T = {
    ERROR, IDENT, NUMLIT, PLUS, MINUS, MULT, DIV, MOD, EXP, NOT,
    AND, OR, XOR, ASSIGN, LT, GT, SHIFTL, SHIFTR, PLUSPLUS, PLUSEQ,
    MINUSMINUS, MINUSEQ, MULTEQ, DIVEQ, MODEQ, EXPEQ, NOTEQ,
    LOGAND, ANDEQ, LOGOR, OREQ, XOREQ, EQUALTO, SHIFTEQ, LTE,
    SHIFTEQ, GTE, TILDA, QUEST, COLON, COMMA, RPAREN, LPAREN,
    SEMI, SIZEOF, EOFT
}

N = {
    <program>, <more_stmts>, <stmt>, <term>, <pre>, <uoppre>,
    <var>, <post>, <stmt_tail>, <binop>
}

S = <program>

P = {
    1. <program> -> <stmt> SEMI <more_stmts> EOFT
    2. <more_stmts> -> <stmt> SEMI <more_stmts>
    3. <more_stmts> -> {}
    4. <stmt> -> <term> <stmt_tail>
    5. <term> -> <pre> <var> <post>
    6. <term> -> LPAREN <stmt> RPAREN
    7. <pre> -> <uoppre> <pre>
    8. <pre> -> {}
    9. <uoppre> -> PLUS
    10. <uoppre> -> MINUS
    11. <uoppre> -> MULT
    12. <uoppre> -> NOT
    13. <uoppre> -> AND
    14. <uoppre> -> TILDA
    15. <uoppre> -> PLUSPLUS
    16. <uoppre> -> MINUSMINUS
    17. <var> -> IDENT
    18. <var> -> NUMLIT
    19. <post> -> PLUSPLUS
    20. <post> -> MINUSMINUS
    21. <post> -> {}
    22. <stmt_tail> -> <binop> <stmt>
    23. <stmt_tail> -> QUEST <stmt> COLON <stmt>
    24. <stmt_tail> -> {}
    25. <binop> -> PLUS
    26. <binop> -> MINUS
    27. <binop> -> MULT
    28. <binop> -> DIV
    29. <binop> -> MOD
    30. <binop> -> AND
    31. <binop> -> OR
    32. <binop> -> XOR
    33. <binop> -> ASSIGN
    34. <binop> -> LT
    35. <binop> -> GT
    36. <binop> -> SHIFTL
    37. <binop> -> SHIFTR
    38. <binop> -> PLUSEQ
    39. <binop> -> MINUSEQ
    40. <binop> -> MULTEQ
    41. <binop> -> DIVEQ
    42. <binop> -> MODEQ
    43. <binop> -> EXPEQ
    44. <binop> -> NOTEQ
    45. <binop> -> LOGAND
    46. <binop> -> ANDEQ
    47. <binop> -> LOGOR
    48. <binop> -> OREQ
    49. <binop> -> XOREQ
    50. <binop> -> EQUALTO
    51. <binop> -> SHIFTEQ
    52. <binop> -> LTE
    53. <binop> -> SHIFTEQ
    54. <binop> -> GTE
    55. <binop> -> COMMA
}
```