

## Program Assignment 2

- Bitwise Manipulation of Hexidecimal Numbers

- CFG
  - E → E "|" A     bitwise OR
  - E → A
  - A → A "^" B     bitwise XOR
  - A → B
  - B → B "&" C     bitwise AND
  - B → C
  - C → "<" C     bitwise shift left 1
  - C → ">" C     bitwise shift right 1
  - C → "~" C     bitwise NOT
  - C → "(" E ")"
  - C → hex

N. Meng, S. Arthur

37

## LL(1) Attribute Grammar

```

E → A EE
EE.st = A.val    E.val = EE.val
EE1 → | A EE2
EE1.st = EE2.st | A.val    ("|" bitwise OR)
EE1.val = EE2.val

EE → ε
EE.val = EE.st

A → B AA
AA.st = B.val    A.val = AA.val
AA1 → ^ B AA2
AA1.st = AA2.st ^ B.val    ("^" bitwise XOR)
AA1.val = AA2.val

AA → ε
AA.val = AA.st

B → C BB
BB.st = C.val    B.val = BB.val
BB1 → & C BB2
BB1.st = BB2.st & C.val    ("&" bitwise AND)
BB1.val = BB2.val

BB → ε
BB.val = BB.st

C → < C1
C1.val = C1.val << 1    ("<<" bitwise shift left one)

C → > C1
C1.val = C1.val >> 1    (">>" bitwise shift right one)

C → ~ C1
C1.val = ~C1.val    ("~" bitwise NOT)

C → ( E )
C.val = E.val

C → hex
C.val = hex.val

```

38

## Program Requirement

- Write a C program using recursive descent parser with lexical analyzer to implement the designated inherited and synthesized attributes. The program evaluates the expressions in a file input.txt, and outputs the results to console
- E.g., input: f&a  
output: f&a = a

N. Meng, S. Arthur

39

## Program Requirements

- You cannot use more than 2 global/non-local variables, and they should be to hold the Operator and HexNumber as detected by the lexical analyzer

N. Meng, S. Arthur

40

## Hints

- To solve the problems, you should take the following steps:
  - Write a lexical analyzer
  - Write a recursive-descent parser
  - Attributes are processed as either pass-in parameters or return value of functions

N. Meng, S. Arthur

41

## Hints

- Write a lexical analyzer
  - You may need to define an enum type for all possible tokens your scanner can generate
  - E.g., when reading hexadecimal numbers 0-9 or a-f, the recognized token is HEX, and the value is saved in HexNumber

N. Meng, S. Arthur

42

## Hints

- Write a recursive-descent parser
  - Parse the program by defining and invoking functions
  - E.g.,  $E \rightarrow A EE$ 

```

EE.st = A.val
E.val = EE.val
int E() {
    int val = A();
    return EE(val);
}

```

N. Meng, S. Arthur

43

## Hints

- There are parameters passed in or returned when invoking functions. When invoking a function, the synthesized attribute is the return value, while the inherited attribute is the passing-in parameter

N. Meng, S. Arthur

44

## Hints

- Sample code of main()
 

```

int main() {
    int val;
    symbol = getNextToken();
    while (symbol != EOF_) {
        if (symbol != NEW_LINE) {
            val = E();
            printf("%x\n", val & 0xf);
        }
        if (symbol == EOF_) break;
        symbol = getNextToken();
    }
    return 1;
}

```

N. Meng, S. Arthur

45

## Submission Requirements

- Pack the following files into a .tar file:
  - Source file: parser.c
  - Executable file: parser
  - Input file: input.txt
  - Output file: output.txt (copy all your console outputs to this file)
  - README file (optional, used if you have any additional comments/explanations about the files)

N. Meng, S. Arthur

46