Compiler Construction

BPDC

(Lab - 06)

1 C Mini-compiler

As before, start with the solution for Lab-05.

1. Modify your compiler to incorporate pointer arithmetic.
   (a) The compiler should successfully parse declaration statements of the form $int\ x, **a$;
   (b) Correspondingly, we have to extend the symbol table structure to encode this additional information. For example, you could have an additional variable $pointerDepth$ that would be k if the variable has reference depth k (declaration of the form $int\ **p$ implies $pointerDepth(p) = 2$) and 0 otherwise.
   (c) Further, every program statement should have all it's participating variables including arrays to be of same depth. For example, the following sequence of statements should parse successfully.

```
int main()
{
    int **p,*q,A[10][20],k;
    //p and A are of pointer depth 2, q of 1 and 0 for k
    p=A; //reference depth 2
    p=&q;////reference depth 2
    q=*(A+2);//reference depth 1, needs special treatment
    **p=*q+k//reference depth 0
 }
```

   Here we have to bring in a new operator & and extend the interpretation of the operator *, both would be complementing each other in program statements. Further, as in the example above, your compiler have to deal with pointer arithmetic involving numbers.

   If in case you encounter a program statement having variables with incompatible reference depths, still your parser should successfully pass through with a warning message saying "incompatible pointer types".