Compiler Construction

1   C Mini-compiler

As before, start with the solution for Lab-04.

1. Update the compiler to let the user to include other header files too. Either from standard library, say $\#include < something.h >$, or from a user defined directory(the filename in double quotes, say, $\#include\,"/home/user/something.h"$). The compiler doesn't have to verify the existence of the so included header file, just have to check the template structure.
2. Extend your compiler to let the programmer to use arrays in programs.
   (a) The compiler should successfully parse declaration statements of the form $int\ x, a[10];$
   (b) Correspondingly, we have to extend the symbol table structure to encode this additional information. For example, you could have an additional variable $isArray$ that would be k if the variable is a k-dimesional array and 0 otherwise. Correspondingly, you could incorporate an additional 1-dimensional array in your symbol table whose first k entries will be positive if the variable under consideration is a k-dimensional array ($k+1^{st}$ entry could be -1).
   (c) The compiler has to ensure that those variables declared as arrays will be accessed only using index variables that are integers. Further, every access to a k-dimensional array should be through a list of k indices, referring to a unique element in the array (say, given a declaration statement of the form $int\ a[20][30], c;$ an expression $a[i][i] = a[j][j]+c;$ should compile successfully, while $a[i][j] = a[j] + c;$ should throw an error).