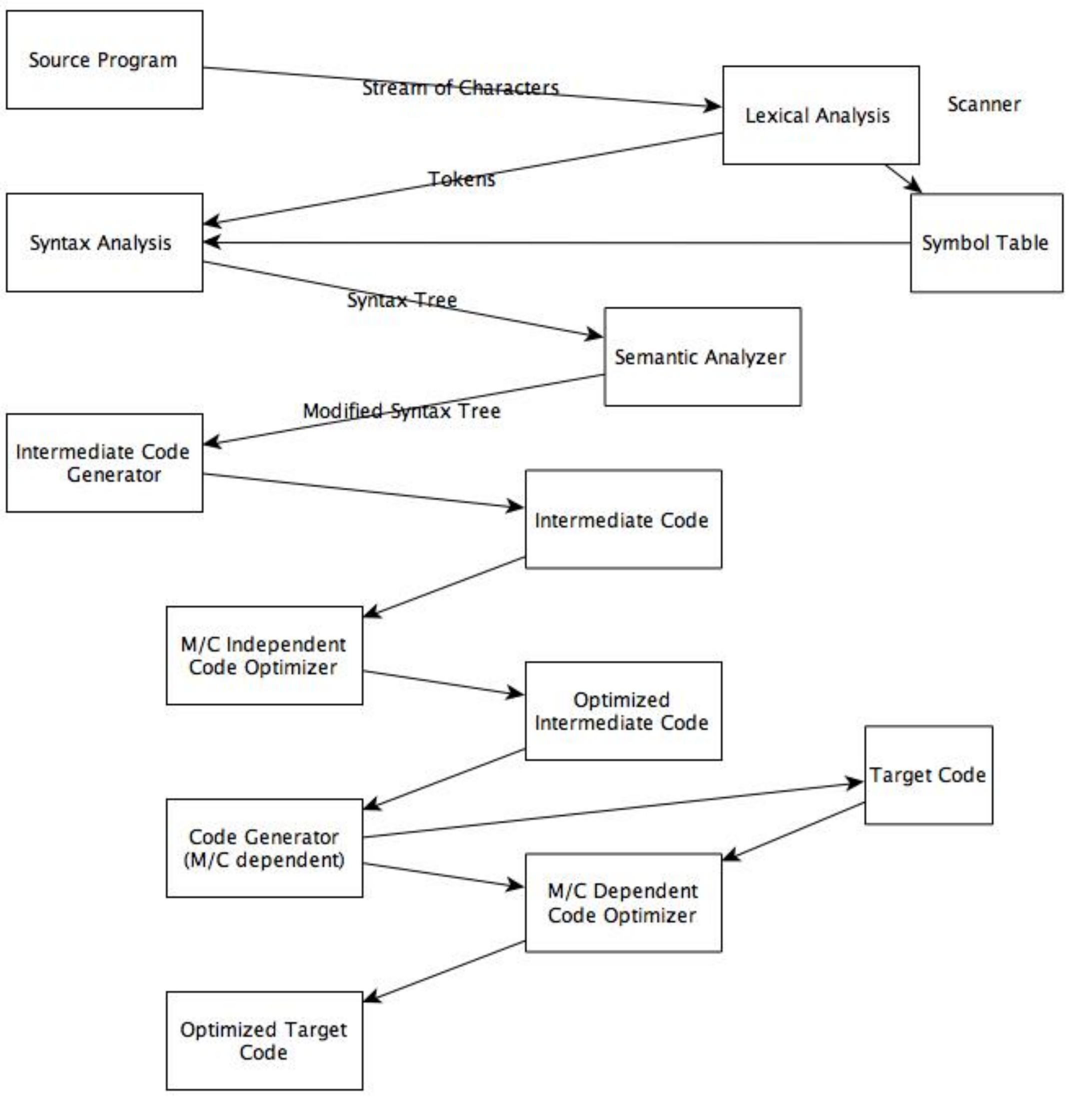
Date: 29th July 2013 Compilers Report

Overview of a Compiler

A compiler converts the source code into a target code by the following steps. The important role of the compiler is to report errors in the source code. First the source program as a stream of characters is analyzed and tokens are generated and also the instructions are stored in the symbol table in order to verify the code in the next steps.



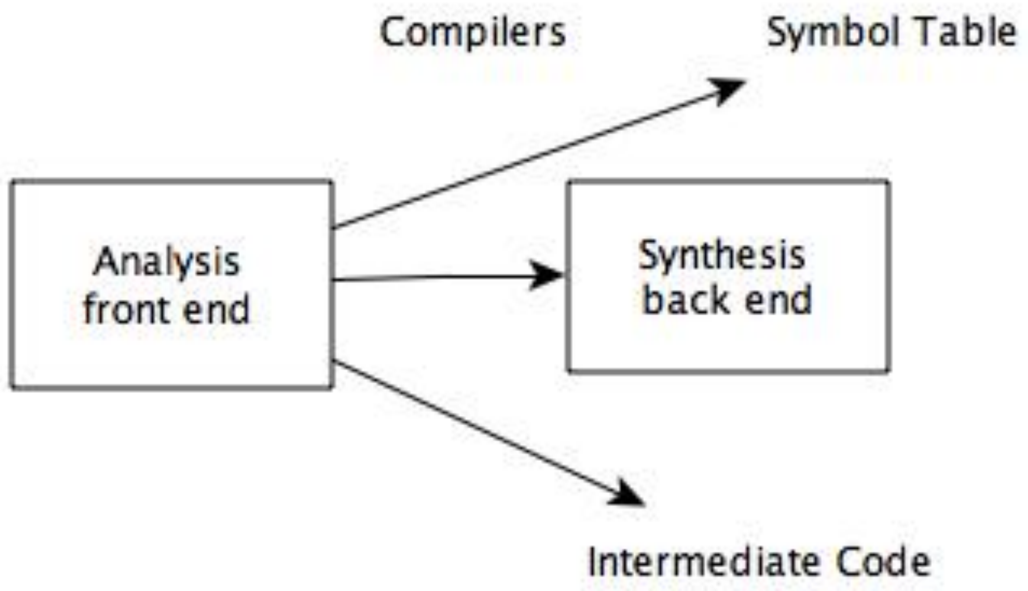
Lexically analyzed code then undergoes syntax analysis and syntax tree is generated if all the syntax is correct else an error is reported. Then the code as a syntax tree is semantically analyzed, for ex: type casting is done wherever required. After this intermediate code is generated, then from the intermediate code machine independent optimized code is generator by the optimizer. An optimized code is generated and then machine dependent code optimizer then Optimized Target Code is generated.

To finalize the overview:

- A compiler splits the source program into constitutional pieces.
- Fit these pieces based on the grammatical rules.
- Generate intermediate representation based on data, Symbol table.

A code optimizer, optimizes the source code and the effect is seen in these terms

- Size of the machine code.
- Minimize the execution time.
- Power
- Memory



Symbol Table

This is a very essential function of a compiler. A symbol Table is used to store the variable names used in the source program. These attributes provide information like the storage allocated for the variable and in case of procedure names, such things as the number and types of its arguments, the method of passing each argument and type returned.

Symbol Table is a data structure containing a record of each variable name, with fields for the attributes of the name. This data structure must be allowed for the compiler to find and record each name quickly and to store or retrieve data from that record quickly.