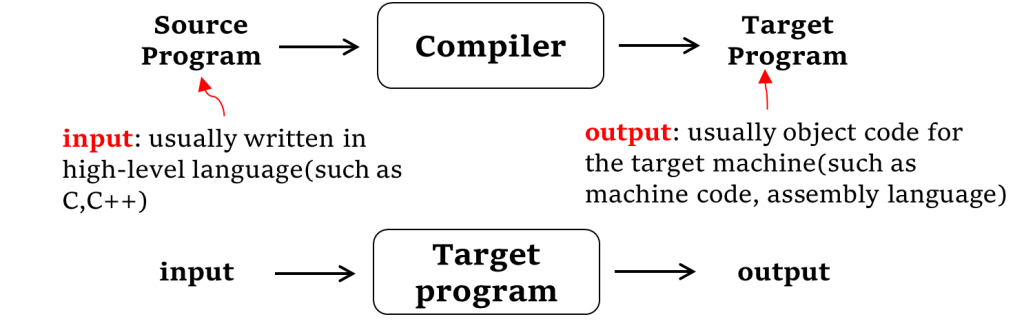
编译原理复习 #1 Compiler

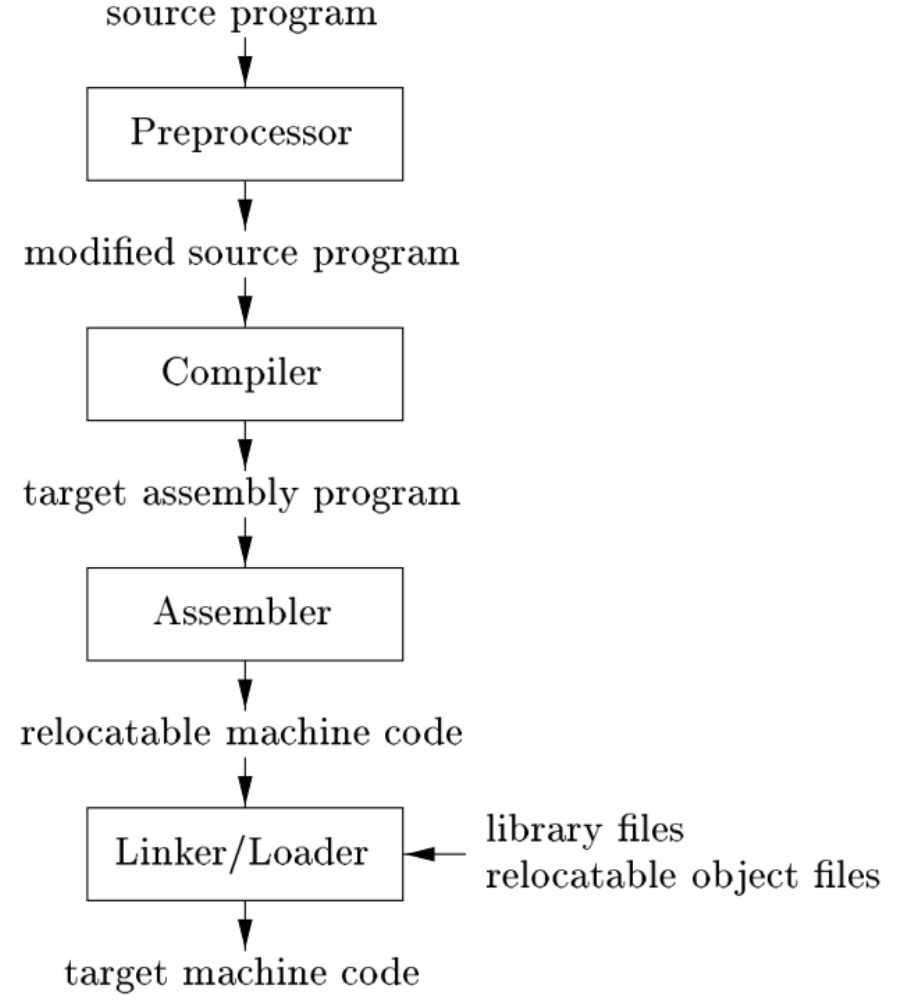
**编译器：**将程序从一个语言翻译另外一个语言的程序



**编译器和解释器：**

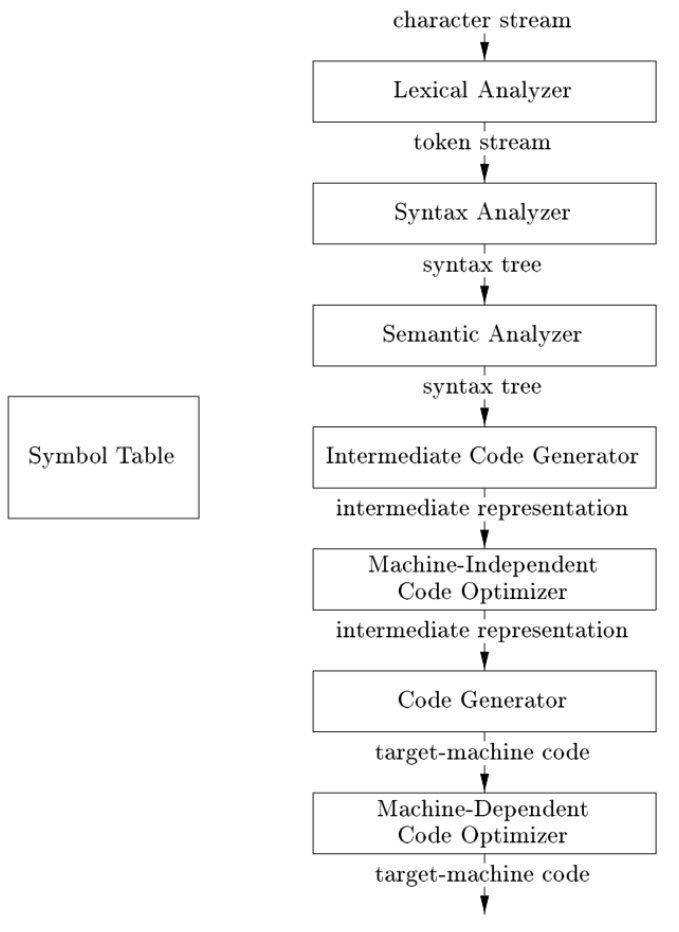
* **The same point:**They are all language implementing system
* **Differences:**Interpreter executes the source program during translation Compiler generates object code that is executed after translation completes

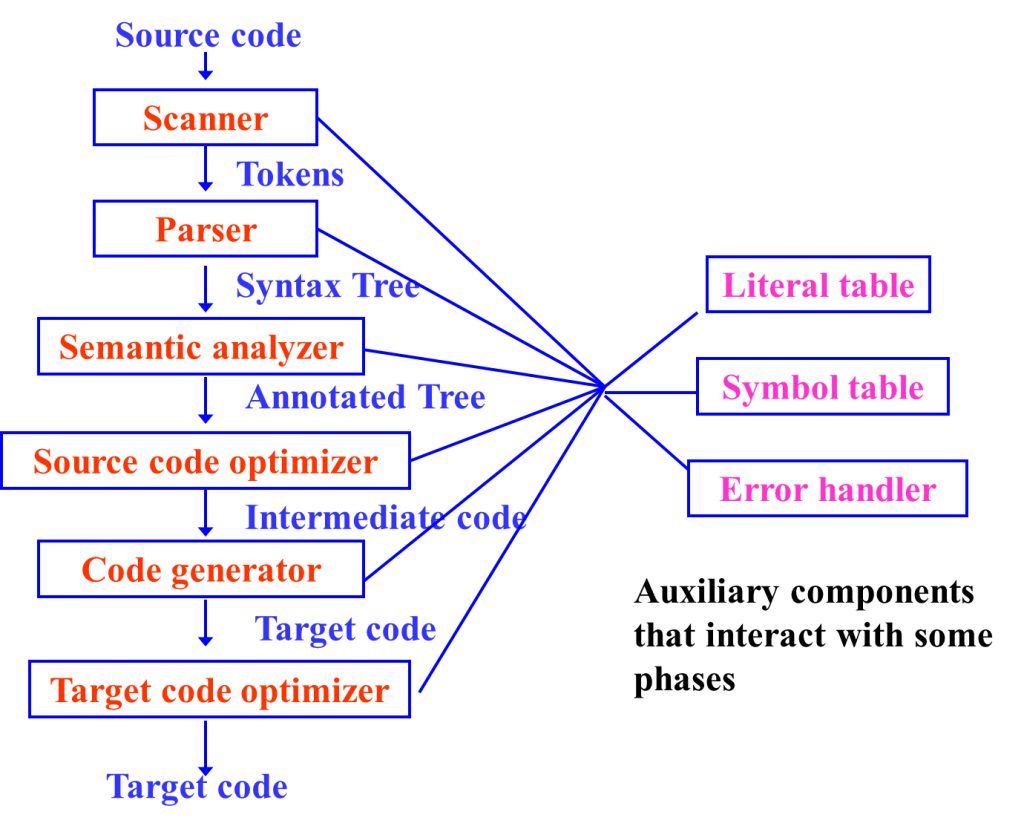
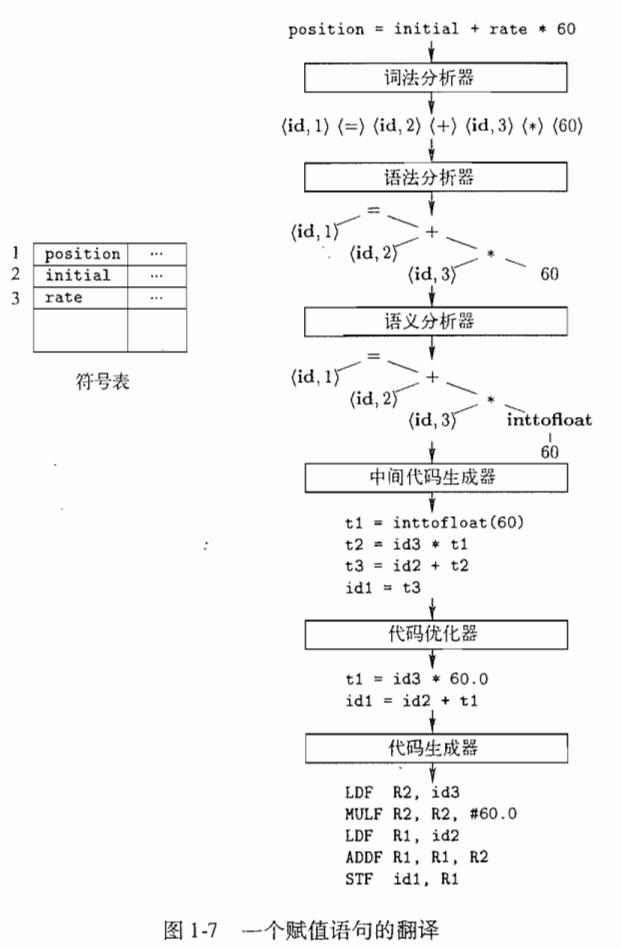
**编译的各个阶段\***



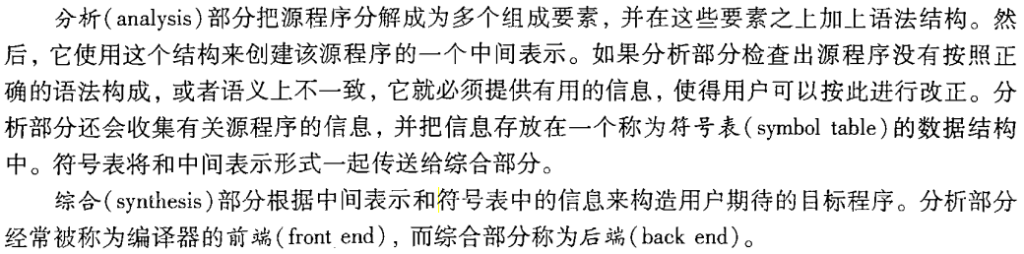
**编译器内部\***

* **Lexical analysis (Scanning):** Identify logical pieces of the description
* **Syntax analysis (Parsing):** Identify how those pieces relate to each other
* **Semantic analysis:** Identify the meaning of the overall structure
* **IR Generation:** Design one possible structure
* **IR Optimization:** Simplify the intended structure
* **Generation:** Fabricate the structure
* **Optimization:** Improve the resulting structure





**编译前端**主要包括词法分析、语法分析、语义分析、中间代码生成这几个部分，**后端**则包含代码优化和目标代码生成部分。前端的特点是仅与编译的源语言有关，而后端则仅与编译的目标语言及运行环境有关。将编译过程划分成前端和后端，主要目的是在多种源语言和多种目标语言的开发过程中，可以灵活搭配组合，消除重复开发的工作量，提高编译系统的开发效率



**符号表（symbol table）**是一种供编译器用于保存有关源程序构造的各种信息的数据结构。**这些信息在编译器的分析阶段被逐步收集并放入符号表，它们在综合阶段用于生成目标代码**。符号表的每个条目中包含与一个标识符相关的信息，比如它的字符串（或者词素）、它的类型、它的存储位置和其他相关信息。符号表通常需要支持同一标识符在一个程序中的多重声明

