**第一章**

1. 编译器的基本结构。（编译器前端是重点）

编译器分为：分析部分和综合部分2个部分。分析部分把源程序分解成为多个组成元素，并在这些要素之上加上语法结构，然后，它使用使用这个结构来创建该源程序的一个中间表示。分析部分被成为编译器的前端，包含词法分析器，语法分析器，语义分析器和中间代码生成器。

1. 如何得到中间表示？书P3的图1-6

编译器的前3个阶段对源程序分别进行不同的分析，以揭示源程序的基本数据和结构，决定它们的定义，建立源程序的中间表示。

1. 用自己的语言描述编译器的各部分结构。

词法分析：逐个读构成源程序的字符，把它们组成词法记号流。

语法分析：把词法记号流依照语言的语法结构按层次分组，以形成语法短语。

语义分析：检查程序的语义正确性，以保证程序各部分能有意义地结合在一起，并为以后的代码生成阶段收集类型信息。

中间代码生成器：用于生成中间代码，中间代码位于高级编程语言和机器语言之间。

代码优化器：改进代码，以产生执行较快的机器代码。

代码生成器：生成可重定位的机器代码或汇编码

**第三章**

1．词法单元，属性，词素，模式

2. 根据给出的要求或者字符串，写出正则表达式，以及正则定义。如书上的digits，id等

3. NFA与DFA的基本元素，画NFA的状态转换图，状态集合中的开始状态，接受状态，状态转移等。

NFA包括

1. 有限的状态集合S
2. 输入符号集合∑
3. 转换函数move：Sx（∑∪{ε}）→P(s)
4. 状态S0是唯一的开始状态
5. FS是接受状态集合

DFA包括：

1. 有限的状态集合S
2. 输入字母集合∑
3. 转换函数move：Sx∑→S，且可以是部分函数
4. 唯一的开始状态S0
5. 接受状态集合FS

4. 有限自动机的分类：不确定有限自动机(NFA)，确定有限自动机(DFA)。

5. 课上讲过的例子，从正则表达式到NFA到DNF到最简DFA（参考(a|b)\*abb）的例子

6. 直接根据正则到DFA，详细参考那几个函数。参考书上例题。P1113同上

7. lex工具的使用方法，以及lex的语法

**第四章**

1. 上下文无关法的四元组：

VT : 终结符集合

VN : 非终结符集合

S : 开始符号，非终结符中的一个

P : 产生式集合， 产生式形式 : A  

2．文法的二义性

3. 看一下上课讲的例子如何消除直接左递归和非直接做递归

**名词解释：**

**源语言：**源语言一般指的是，编写源程序所用的语言，它必须翻译成机器语言才能在计算机中使用。

**目标语言**：用另一种计算机语言写成的文件将被翻译成的计算机语言，常为一种机器语言。

**翻译器：**能够完成从一种语言到另一种语言变换的软件

**编译器：**编译器是一种翻译器，特点是目标语言比源语言低级。

**解释器：**是一种电脑程序，能够把高级编程语言逐行直接转译运行。

**词法单元：**词法单元由一个词法单元名（记号名）和一个可选的属性值组成。

**词法单元名（记号名）：**词法单元名是一个表示某种词法单位的抽象符号。比如一个特定的关键字或者代表一个标识符的输入字符序列。

**模式：**模式描述了一个词法单元的词素可能具有的形式。

**词素：**是源程序中的一个字符序列，它和某个词法单元的模式匹配，并被词法分析器识别为该词法单元的一个实例。

**二义性：**如果一个文法可以为某个句子生成多棵语法树，那么它就是二义性的。二义性文法就是对同一个句子由多个最左推导或最右推导的文法。

**死状态：**对所有输入符号都转换到本身

**有穷自动机：**是状态转换图的形式化表示

**确定有穷自动机：**是一个特殊的有穷自动机，它的任何一个状态对于任意一个输入符号有且只有一个装换。

**不确定有穷自动机：**不是确定有穷自动机的自动机称为不确定有穷自动机。

**终结符号：**是组成串的基本符号，是词法单元名的同义词。

**非终结符号：**是表示串的集合的语法变量。

**上下文无关语言：**可以由文法生成的语言成为被称为上下文无关语言

**文法等价：**如果两个文法生成相同的语言，则称这两个文法是等价的。

**句型：**如果S=>\*α，其中S是文法G的开始符号，那么α是G的一个句型。

句柄：

**可行前缀：**可以出现在移入-规约语法分析器的栈中的最右句型前缀被称为可行前缀。

**简答题：**

**1. 词法分析和语法分析的分离有如下好处：**

简化设计

编译器的效率会改进

编译器的可移植性加强

便于编译器前端的模块划分

**2. 能否把词法分析并入到语法分析中，直接从字符流进行语法分析？**

若把词法分析和语法分析合在一起，则必须将语言的注解和空白的规则反映在文法中，文法将大大复杂。

注解和空白由自己来处理的分析器，比注解和空格已由词法分析器删除的分析器要复杂得多。

**为什么要用正规式定义词法 ?**

词法规则非常简单，不必用上下文无关文法

对于词法记号，正规式描述简洁且易于理解

从正规式构造出的词法分析器效率高

**文法的优点**

文法给出了精确的，易于理解的语法说明

自动产生高效的分析器

可以给语言定义出层次结构

以文法为基础的语言的实现便于语言的修改

**文法的问题**

文法只能描述编程语言的大部分语法，不能描述语言中上下文有关的语法特征

**指令的选择**

目标机器指令系统的性质决定了指令选择的难易程度，指令系统的统一性和完备性是重要的因素

指令的速度和机器特点是另一些重要的因素