

## 大题

- 1 【第三章】

给一个语言描述，看书上例子，给一段字符串的描述，字符串的特征，怎么设计它。给字母表，用字母表构建特定的串。画有穷自动机，化等价的正规式，给出NFA，把NFA化成DFA，DFA最小化，构建正规文法。整章内容都要掌握，共22分。【DFA】

- 2 【第四章】

给一个语句，可能是循环句或者条件句。给文法，先判断是不是LL1型文法，如果不是则把文法改造成LL1型，改造完再判断是不是，再构造分析表。（提取公因子，消除左递归，可以改造成LL1）。给出一个句子，根据自己构造的分析表，写出自顶向下语法分析的过程，画出所有栈每一步的变化情况。共15分。【求SELECT集，画预测分析表，画栈变化】

- 3 【第六章】

给一个文法。6.2.2节活前缀，LR0项目集，会构建自动机。构造LR0分析表，判断是否LR0如果不是LR0文法再构造SLR1分析表判断是否SLR1文法。6.2和6.3节。共15分。【SLR1文法，求项目集】

- 4 【第二章、第七章、第八章】

【小综合题】给一个文法，给出翻译方式，相应语义动作。是哪种类型的文法（第二章）0型1型2型，句子句型句柄等概念。

给一个输入符号串。用语法分析句子，同时执行语义动作，输出是什么。如果没有写，则自己规定是自上而下还是自下而上。【翻译模式，执行语义动作】

- 5 【第十章】

给基本块四元式，DAG优化。哪些变量在基本块之后不活跃，再优化。【DAG优化】

- 【第二套卷子的不同之处】

给一个句子翻译四元式，外面是循环里面有判断，循环条件里面有循环。【写四元式序列】

- 30道小题

大部分小题都是基本概念，比如编译包括哪几个阶段。DFA和NFA基本概念。词法分析器文法符号，语法分析常见方法，文法符号，属性（集成属性等）。中间代码。正规式等价。给一个文法，判断语言。表达式求值，按照给的问答计算，注意优先级。LR0的分析表，归约怎么填表。给一段程序可以做什么优化。自动机。消除左递归。什么是局部优化。

## 客观题归纳：

- 【选择】通常程序设计语言的单词符号都能用**正规式、正规文法**描述。

上下文无关文法和上下文有关文法通常用于语法分析阶段。

- 【选择】算符优先文法从左向右扫描输入串，当栈顶出现**最左素短语**时进行归约。

素短语：是一个短语，至少包含一个终结符，除自身外不包含其他素短语。

- 编译过程通常分为六个阶段：**词法分析、语法分析、语义分析、中间代码生成、代码优化、目标代码生成**（有时去除语义分析也是正确的）

- 文法四元组中( $V_n, v_t, P, S$ ) 非终结符集，终结符集，规则集，开始符。

- 波兰式：前缀表达式。逆波兰式：后缀表达式。

- 程序设计语言的单词符号一般分为5种：**关键字(if,else,for), 标识符, 常量, 运算符, 界符({,})**

- 基本块的常见优化方法：**合并常量、删除公共子表达式、删除无用赋值。**

- 语法制导的语义计算中，含有**S-综合属性, L-继承属性**。

- 布尔表达式翻译的语义规则中使用了拉链和**代码回填技术**。

- 二义文法的定义：**存在某个句子对应两棵不同的语法树。**

- 语法制导的编译程序能同时进行**语法分析**和**语义分析**。
- 常见的中间代码形式**逆波兰式**、**三元式**、**四元式**。
- LR(0)项目集规范族的项目类型可分为**移进项目**（.后是终结符）、**归约项目**（.在最后）、**待约项目**（.后是非终结符）、**接收项目**。
- LR0项目集中，圆点在最后的为**句柄识别态**，第一个产生式的句柄识别态为**句子识别态**。
- 一个有穷自动机可以通过**消除无用状态**、**合并等价状态**而转换成一个与之等价的最小状态的有穷自动机。
  - 两个状态s,t等价的条件是：
    - 一致性条件**：同时为可接受或不可接受状态。
    - 蔓延性条件**：对所有输入符号，s,t转换到等价的状态里。
  - 分割法用来划分不同状态子集。
- 自底向上分析最常用的技术是**移进归约分析**，边移入边分析，一旦栈顶符号串形成某个句型的句柄或其他可归约串时就进行归约。
- 算符优先文法：算符优先级是确定的，不会出现同两个算符，既大于又小于的优先级。
- LR分析器由3个部分组成：总控程序（驱动程序）、分析表（分析函数）、分析栈。
- ACTION动作有4种可能：移进、归约、接受、报错。
- 任何一个二义性文法绝不是LR类文法，也不是算符优先文法或LL(k)文法，也不存在与其相对应的确定的语法分析器。
- 子集法**：将NFA转为DFA。
  - 【判断】**一个二义文法所生成的语言一定也是二义的语言。 **【错】**
  - 【判断】**由开始符号出发，每步进行归约的方法称为自上而下语法分析方法。
  - 【判断】**变量的待用信息指出了该变量值的后续引用位置。 **【对】**
  - 【判断】**一个二义性文法绝对不是LR(K)文法。 **【对】**
  - 【判断】**文法是识别一个语言的集合。 **【错】**是定义不是识别。
  - 【判断】**NFA包含DFA。 **【对】**
  - 【判断】**程序流图中的每条回边对应于一个循环。 **【对】**
  - 【判断】**LR分析自左向右扫描输入串时就能发现错误，但不能准确指出位置。 **【错】**
  - 【判断】**LR分析法是一种规范归约分析法。 **【对】**
  - 【判断】**符号表可以辅助语法错误、上下文语义正确性检查。 **【对】**
  - 【判断】**算符优先分析方法属于自上而下的分析方法。 **【错】**是自下而上的一种方法
  - 【判断】**PLO采用多遍扫描的编译过程。 **【错】**单遍
  - 【判断】**基本块的入口语句可以是以下三类语句：程序的首条语句；跳转语句的转移目标；跳转语句后一条语句。 **【错】**是条件跳转语句的后一条语句。
  - $\varepsilon$ 和 $\emptyset$ 都是  $\Sigma$  上的正规式，它们所表示的正规集分别是  $\{\varepsilon\}$  和  $\emptyset$

# 课本梳理：

## 第一章 引论

- 编译过程：
  - 前端：特点是与源程序相关，与目标机器无关。
    - 词法分析：逐个字符读入源程序，识别出单词。
    - 语法分析：在词法分析基础上，将单词序列分解成各类短语。
    - 语义分析：审查语义错误，为代码生成阶段收集类型信息。工作之一是进行类型审查。
    - 中间代码生成：通常使用三地址指令和四元式：（运算符，运算对象1，运算对象2，结果）
  - 后端包括：中间代码生成和目标代码优化不是编译程序必须的。
    - 中间代码优化
    - 目标代码生成：把中间代码转换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。
    - 目标代码优化。
    - 特点是与目标机器相关，与源语言无关。

## 第二章 文法和语言

- 字母表：元素的非空有穷集合。
- 规则又叫重写规则、产生式、生成式。
- 句型：文法中推导出来的；句子：仅由终结符组成。
- 文法描述的语言是该文法一切句子的集合。
- DAG是用于基本块优化的重要工具，可以识别以下几种优化机会：
  - 常量合并

```
1 t1 = 3 * 5
2 t2 = t1 * 2
3 // 优化为
4 t1 = 8
5 t2 = 16
```

- 删减公共子表达式

```
1 t1 = a + b
2 t2 = c + d
3 t2 = a + b
4 // 优化为
5 t1 = a + b
6 t2 = c + d
7 t3 = t1
```

- 删减无用赋值

```
1 | x = 10
2 | y = 20
3 | x = y + z
4 | w = x + y
5 | // 优化为
6 | y = 20
7 | x = y + z
8 | w = x + y
```