

编译原理复习

复习

1、→ 期末考试

简答题、应用题、综合题

开卷，120 分钟

占最终成绩的 50%。

2、→ 知识点

a. → 编译器的基本概念

编译器的作用，编译的主要过程

b. → 形式语言

字母表、符号串、符号串的运算、产生式、文法的定义 $G=(V_N, V_T, P, S)$ 、句型、句子、语言、短语、直接短语、句柄、最左推导、最右推导、语法树、二义性、文法分类

c. → 词法分析

正规式、有穷自动机、 $NFA \rightarrow DFA$ ， DFA 的化简

d. → 语法分析

自顶向下：消除左递归、提取左因子，递归下降法， $LL(1)$

自底向上：算符优先文法，LR 分析法 ($LR(0)$ ，SLR， $LR(1)$)

e. → 语法制导的翻译

变量声明语句，简单赋值语句，布尔表达式，if 语句，循环语句

f. → 中间代码生成、目标代码生成

感觉可能会大题的：

写一个实现描述功能的文法

NFA DFA转化、化简

$LL(1)$ ，LR，SLR一套流程从first follow 分析表，从DFA action和goto表

写一个给定文法/要求的语法制导定义

目前还有困惑的：

$LR(1)$ 中展望符号怎么确定？

下次引用信息的算法这个不懂流程

第一章编译概述

1.1 概念

1. 高级程序的优点

2. 翻译可分为编译和解释，编译：源程序→目标程序，源程序在编译阶段处理，数据在运行阶段处理。解释：源程序→结果，同时处理源程序和数据

3. 编译程序可分为汇编程序和编译程序
4. Java语言先编译再解释

1.2 编译各阶段

1. 分为分析和综合两阶段，前者分析结构，是跟源语言有关；后者根据分析结构构造程序，跟目标语言有关。
2. 分析包括结构（词法，语法），语义分析
3. 综合：对分析产生的中间形式综合加工，得到目标程序，可分为中间代码生成、代码优化、目标代码生成
4. 符号表管理，收集使用的标识符，在不同阶段不断添入
5. 错误处理

1.3 其他概念

1. 前端和后端
2. 遍，一遍和多遍，优缺点

第二章 形式语言和自动机

2.1 语言和文法

1. 字母表和符号串，符号串基于字母表，长度，前缀，后缀，子串；连接，幂
2. 语言，一堆符号串的一个集合。并，连接，闭包
3. 文法，终结符号、非终结符、开始符、产生式。可分类为0、1、2、3型文法。一般2型上下文无关
4. 推导和短语，一步（直接）推导，推导反义为归约，最左推导，最右（规范）推导。左右句型，句子（纯终结符），句子组成语言。**短语**，最左直接短语是**句柄**。
5. 分析树：推导树，有序有向树，不能完全反映中间推导过程。子树：分析树的一部分。**子树和短语**。
6. **二义性**：一个句子有不只一个分析树，有时可以转成一个等价的无二义性的文法。
7. 文法变换。消除左递归，左递归：非终结符A可以推出自己，直接左递归：一步推出自己，自顶向下的语法分析没法处理左递归。消除直接左递归，固定公式；间接左递归，先全部带入再消除直接左递归。

消除左递归：首先保证无环路，无 ϵ 产生式

消除空产生式

提取左公因子

2.2 有限自动机

1. 离散输入输出的系统模型，系统可以处于任一内部状态。

分为确定，下个状态唯一确定。非确定，两个以上的后继状态

2. 确定的有限自动机DFA，状态转换图、矩阵

3. 非确定的有限自动机NFA，转换成DFA（考虑所有可能状态 2^Q 次方）
4. 含有 ϵ 转移的NFA，会转换p39例2.13
5. DFA化简，不能到终态的状态是死状态可以去掉。化简的话首先划分为非终态和终态，然后对于每个转移符号，看子集里有跨子集的状态吗，如果有则划分。

2.3 正规文法和有限自动机

1. 右线性文法，左线性文法变为自动机
2. 自动机变为文法

2.4 正规表达式和有限自动机

有一定的规则

2.5 正规表达式和正规文法

第三章词法分析

3.1 词法分析和语法分析

1. 词法分析作用：123
2. 和语法分析的关系：3种，独立一遍、子程序、协同程序
3. 独立词法分析程序的好处：123

3.2 词法分析的输入和输出

1. 输入缓冲区：
2. 输出：记号、模式、单词，记号影响语法，记号的属性影响语义。

3.3 记号的描述和识别

1. 词法的定义，正规文法——词法，上下文无关文法——语法
2. 状态转换图和记号识别

3.4 词法分析程序设计

第四章 语法分析

4.1 简介

1. 输入：记号序列；依据：语法规则；功能：识别语法成分语法检查；输出：分析树
2. 错误处理，目标，错误恢复的策略

4.2 自顶向下分析方法

1. 递归下降分析，试探的过程，不确定的方法。为了寻找一个最左推导序列
2. 递归调用预测分析。确定的不用回溯的方法。构建状态转换图，递归调用。
3. **非递归预测分析**。一个分析表和栈，根据栈顶和输入串的字符匹配来决定下一个步骤，如果终结符匹配了，则前移。如果非终结符，则去预测分析表中寻找要使用的产生式。

关键在于如何构建这样一个预测分析表，要先求**first集**和**follow集**，终结符first为自身，无follow，follow里无epsilon。根据上面的集合构建预测分析表。

P91

4. LL (1) 文法

如果一个文法是二义性的，分析表中至少有一个表项重定义。

LL (1) 文法有两个条件。不是所有文法都可以改写为LL (1) 文法

5. 错误处理（带同步信息）

如果在follow集里，但是这个表项空白，则添加一个synch。

遇到错误三种可能，不同处理方法。

4.3 自底向上分析方法

1. 优先分析法：简单优先分析、算符优先分析。简单优先文法，**算符文法**，算符优先文法
2. “移进-归约”方法
3. 规范归约，最右推导-规范推导，逆过程就是规范归约，需满足两个条件
4. “移进=归约方法”实现，移进、归约、接受、错误处理的概念，“移进-归约”冲突和“归约-归约”冲突

4.4 LR分析方法

1. 概念定义：L、R、k的含义，基本思想（历史、预测信息），评价
2. 工作原理：栈包括状态栈和符号栈，分析表包括action（终结符）和goto表（非终结符）。**活前缀概念**，LR分析程序算法，关键在于构建LR分析表
3. SLR (1) 分析表构造：

构造识别所有文法活前缀的DFA，有两种方法，一种项目->NFA->DFA一种closure->go->DFA。后面的涉及了一些概念，项目，有效项目，项目集规范族，闭包，转移函数，脱光文法。

在得到DFA之后就是分析表的构建，**LR (0) 分析表**归约的选择是，只要有 $A \rightarrow a\cdot$ ，不管下一个为什么符号，都归约

当有一个项目集里同时含有移进-归约（**移进-归约冲突**）或者多个归约（**归约-归约冲突**）的选择时，LR（0）就没法解决了

为了解决归约我们有了**SLR（1）**，主要有：判断条件（follow集合与移进的并集为空），构建算法

SLR（1）也有弊端，因为本质上Follow集范围大于了实际可能出现的字符集

4. LR（1）分析

LR（1）分析也有和LR（0）类似的概念，只是有一定的往前看，构造方法、**展望信息**、判断是否为LR（1）文法

第五章 语法制导翻译技术

5.1 语法制导定义和翻译

1. 整体思路，产生式->语义规则->属性计算
2. 定义，**语法制导**，**继承**、**综合属性**
3. 综合属性：来自于子节点。仅使用综合属性：S属性定义，自底向上
继承属性：来自父兄节点。自顶向下
依赖图
S属性和L属性定义，X的继承属性仅来自于其左边的属性
4. 翻译方案，
为了保证每个属性可用，对S属性定义来说语义规则放在末尾即可
L属性定义文法的翻译方案的三个原则

5.2 S属性的自底向上翻译

5.3 5.4 L属性的自顶向下、自底向上翻译

第六章 语义分析

6.1 概述

1. 任务，收集上下文有关信息，**类型检查**，**动态检查**，**静态检查**
2. 位置，输入，输出
3. 错误处理

6.2 符号表

1. 建立的位置
2. 内容（名字、类型、存储地址、维数、声明行、引用行、链域）
3. 操作（插入、检索、显式隐式声明，定位，**重定位**）
4. 组织（具体以什么形式存储，分非块结构语言和块结构语言）

第八章 中间代码生成

8.1 中间代码形式

1. 图形表示，dag和语法树
2. **三地址代码**，实现三地址语句表示有**四元式**、三元式、简介三元式

8.2 赋值语句翻译

布尔表达式翻译、控制翻译（回填技术）

第九章 目标代码生成

9.1 概述

1. 存储管理、指令选择、寄存器分配，计算次序的选择

9.2 基本块和流图

1. 入口语句，流图的结点是基本块

9.3 下次引用信息

1. 根据三地址语句序列输出基本块中各变量的下次引用信息的算法

代码优化

10.1 代码优化概述

1. 功能，位置
2. 主要种类：中间代码优化（基本块优化、循环优化、全局优化）

10.2 基本块优化

1. 常数合并常数传播
2. 删除公共表达式
3. 复制传播
4. 删除死代码（求值后却不引用）
5. 消弱计算强度，避免函数、无意义的计算
6. 改变计算次序

10.3 dag和基本块优化

1. ?

10.4 循环优化

1. 循环展开，空间换时间
2. 代码外提、频度消弱，与循环无关的代码放到循环外面
3. 消弱计算强度
4. 删除归纳变量

10.5 窥孔优化

1. 定义，局部改进，特点
2. 方法：删除冗余传送指令，删除死代码（删除不可达），控制流优化，强度削弱和代数化简