2018-2023 编译原理填空汇总

Garone Lombard

2023 年 12 月 17 日

摘要

编译原理 2018-2023 试卷填空选择自行整理

目录

1 填空

1.1 2018-2019

1. 0 型文法又称为 短语结构 文法。

2. 规范推导又称 最右 推导，它等价于 最左 归约。

3. 自顶向下语法分析的主要处理方法为 递归子程序法 和 LL 分析法 。

4. 给定正则文法：$A \to bB, A \to b, B \to dB, B \to e$，求等价的正则表达式 A= b(d*e)? or b|bd*e 。

5. 给定文法：$S \to aAb|c, A \to B|d, B \to BS|S$，则句型 aBaAbacbSb 的句柄为 。

6. 设数组维度为 4，则数组模板所需空间大小为 14 。

7. 给定 $arrayA(1:3,-2:1,0:3)$，设数字元素大小为 4 字节，求在计算数组元素地址时的不变部分 RC= -32 。

1.2 2019-2020

1. 在编译过程的五个阶段中，词法分析 的输出是 token 序列，语法分析 的输出是抽象语法树。

2. 根据乔姆斯基对文法的分类，正则文法是 3 型文法，它可以被 有穷自动机 接受

3. 算符优先分析过程每次归约的是 最左素短语

4. 向输入文法插入动作符号后得到的文法是 翻译 文法，这个文法推导所产生的终结符号串称为 活动 序列。

5. 根据能否在编译阶段确定所需数据空间大小，可将运行时的存储分配方式分为 静态存储分配 和 动态存储分配

1.3 2020-2021

1. 2 型文法就是指上下文 无关 文法，若用 $G = (V_N, V_T, P, S)$ 表示，则它要求 G 中的所有规则 $\alpha \to \beta$ 都满足：α 是一个非终结符，β 属于 $(V_N \cup V_T)^*$。

2. 词法分析器的输出结果是 token 序列

3. 如果文法无二义性，则与最右推导互为逆过程的是 最左归约

4. 在符号表上最常执行的操作是登录符号表 (就是填表的意思) 和 查询符号表，这些操作根据所编译的语言是否具有显式声明而稍有不同

5. 某个 C 语言程序中有语句 a=f(5)，编译器报告改语句中含有错误，错误信息是函数 f 没有定义，编译程序是在 语义分析 阶段发现此类错误的。

6. 翻译文法中的符号，包括非终结符、终结符和 动作符号，都是有穷集合中的符号，都没有值的概念

7. 与及其相关的代码优化技术，一旦 机器架构 产生变化，相应的优化方法也要做出调整

8. 以下中间代码含有 3 个基本块，每个基本块分别含有的语句是 1-5,6-7,8。

```
1 a=123456*
2 b=567890
3 c=100000007
4 d=a*b
5 if(d<c) goto 8
6 d=d-c*
7 goto 5
8 return d*
```

9. 已知行优先存储的数组 x 的各维度长度依次为 2、5、10，各维度的下标都从 0 开始计算，则元素 x[1][3][5] 的地址和 x 的首地址之差是 85 个元素长度。

1.4 2021-2022

1. 编译过程本质上是 程序转换/翻译 过程，将用 高级语言 书写的源程序加工为与其等价的目标程序。

2. 在编译过程的 5 个基本阶段都要做 符号表管理 和 错误处理 两件事，因此典型的编译程序常划分为 7 个逻辑组成部分

3. 对源程序 (包括源程序中间形式) 从头到尾扫描一遍，并做有关的加工处理，生成新的源程序中间形式或目标程序，通常称之为 一遍，完成编译工作最少需要对源程序做 1 次扫描

4. 生产中间代码的目的是便于做 代码优化 和 编译程序移植

5. 有文法规则 $S \to if\ E\ S \mid if\ E\ S\ else\ S$，用扩充的 BNF 范式表示为 $if\ E\ S\ [else\ S]$

6. 常见的程序设计语言按乔姆斯基的分类是 1 型文法，也称为上下文无关文法。如果采用属性翻译文法处理声明语句 $int\ a$; 时，通常可以得到变量类型和名字这样的 继承 属性，并填入到 符号表 中，以便在使用变量 a 时，能够查找到变量的有关信息。没有声明就使用变量，这属于 语义错误 ，在语法分析只能进行句子的结构分析时并不能发现这个问题

7. 对文法 $G[T]:\ T \Rightarrow T-T|T/T|(T)|i$，规范句型 $T-T/i$ 的句柄为 i 和 T-T ，由此判断该文法 有 (有/无) 二义性。

8. 规范归约每次归约的是句型的 句柄 ，算符优先分析法每次归约的是当前句型的 最左素短语

9. 活动记录中 Display 区存放的是 各外层模块活动记录的基地址

10. 文法 $G=(V_n,V_t,P,Z)$，其中 V_t 代表 文法中的终结符集合

2 判断题

2.1 2018-2019

false 静态存储分配是在编译阶段由编译程序实现对存储空间的管理,并为源程序中的变量分配存储的方法,~~所有数据空间大小都能在编译过程中确定~~

true 对于某个文法，该文法接受的一个句子必定是该文法的句型

true 2 型语言是上下文无关语言，这种语言可以由下推自动机接受。3 型语言又称正则语言，这种语言可以有有穷自动机接受。2 型文法可以产生 3 型文法。

true 用 3 型文法所定义的语言都可以用正则表达式描述，而一个正则表达式则对应一个 DFA M

false 在付出同等代价的情况下，循环优化一般比局部优化效果更好。

false ~~算符文法允许两个非终结符相邻~~，而算符优先文法则不允许两个非终结符相邻

false 素短语一定是简单短语

true 每个 SLR(1) 都是 LR(1) 文法，但反之不成立

true LR(1) 文法合并同心集后只可能出现归约-归约冲突，而没有移入-归约冲突

true 从编译角度，将错误分为语法错误和语义错误，数据溢出错误属于语义错误

2.2 2019-2020

true 整个编译过程中只对源代码做一次从头到尾扫描的编译器，就是“一遍扫描的编译器”

false 文法 G 所描述的语言，就是文法 G 的终结符集合 Vt 的闭包 Vt*

false NFA 的接受状态可以多于一个，但 DFA 只能有一个

false 算符优先分析过程中，栈顶运算符优先级小于栈外输入运算符时，执行入栈操作；栈顶运算符优先级大于栈外输入运算符时，执行出栈规约操作；其他情况说明遇到了错误

true 属性翻译文法中综合属性的求值是自下向上的；而继承属性的求值是自上向下的

true First 集可以包含 ϵ，Follow 集不可以包含 ϵ

flase 规范句型的活前缀不一定是唯一的

true LL(1) 文法和 SLR(1) 文法一定都无二义性

false 与机器有关的优化一般是在中间代码上进行的

true 对于右侧的代码块：语句 return j+1 等价于 return 1.....

2.3 2020-2021

true 对给定的文法 G[S]，若至少有一个句型存在两个或两个以上的不同的最左 (或最右) 推导，这是判定是二义文法的充分必要条件

true 素短语不含其它素短语，且至少含有一个终结符

false 动态数组的存储空间在编译时就可完全确定

true 在 C 语言程序执行过程中，静态变量的存储空间不在过程的活动记录中

true 对于源程序中的声明语句，编译程序通常不产生可执行代码

true LL(1) 分析方法是非递归预测语法分析方法

false LR(1) 文法是 3 型文法

3 简答题

3.1 2019-2020

1. 简述什么是错误的局部化处理，主要作用是什么?

2. 分别简述语法分析的任务和语义分析的任务

3. 分别简述静态存储分配和动态存储，二者的使用场景有什么不同

4. 分别说明什么是局部优化、全局优化和循环优化

5. 说明什么是交叉编译，什么时候需要使用交叉编译

4 知识点

4.1 概论

源程序 用汇编语言或高级语言编写的程序称为源程序

目标程序 用目标语言所表示的程序

翻译程序 将源程序转换为目标程序的程序称为翻译程序

乔姆斯基文法体系

- 0 型文法：短语结构文法，可以使用图灵机接受
- 1 型文法：上下文有关文法，可以使用线性有界自动机接受
- 2 型文法：上下文无关文法，可以使用下推自动机接受
- 3 型文法：正则文法，可以使用有穷自动机接受

推导/归约 最右推导和最左归约互为逆过程

自顶向下分析 主要问题在于左递归问题和回溯问题; 主要方法为递归子程序法和 LL 分析法

自底向上分析 主要问题在于句柄的识别问题; 主要方法为算符优先分析法和 LR 分析法

数组模板大小 取决于数组维数，为固定的 $3n+2$，所以无论是常界或是变界数组，在编译时都能确定数组模板的大小

静态存储分配 在编译阶段由编译程序实现对存储空间管理和源程序中的变量分配存储的方法，但是并不是所有数据空间大小都能在编译过程中确定

动态存储分配 在目标程序运行阶段由目标程序实现对存储空间的组织和管理，和为源程序中的变量分配存储的方法，需要在编译时生成进行动态分配的目标指令

算符文法 设有文法 G，如果 G 中没有形如 $A \to ...BC...$ 的产生式，其中 B,C 为**非终结符**，则称 G 为算符文法

算符优先文法 设 G 是一个不含 ϵ 产生式的算符文法，a 和 b 是任意两个终结符，A,B,C 是非终结符，算法优先关系 $\doteq$ $\lessdot$ $\gtrdot$ 定义如下

1. $a \doteq b$ 当且仅当 G 中含有形如 $A \to ...ab...$ 或 $A \to ...aBb...$ 的产生式
2. $a \lessdot b$ 当且仅当 G 中含有形如 $A \to ...aB...$ 的产生式且 $B \Rightarrow^+ b...$ 或 $B \Rightarrow^+ Cb...$，也就是说 $a \lessdot FIRSTVT(B)$

3. $a \gtrdot b$ 当且仅当 G 中含有形如 $A \to ...Bb...$ 的产生式且 $B \Rightarrow^+ ...a$ 或 $B \Rightarrow^+ ...aC$，也就是说 $LASTVT(B) \gtrdot b$

设 G 是一个不含 ϵ 产生式的算符文法，如果任一终结符对 (a,b) 之间至多有 $\doteq$ $\lessdot$ $\gtrdot$ 三种关系中的一种成立，则称 G 是一个**算符优先文法**

素短语 素短语是一个短语，它至少包含一个终结符号，并且除它自身以外不在包含其他素短语 (算符优先分析归约的就是最左素短语)

LR(0) 文法 如果 LR(0) 分析表中没有语法分析动作冲突，则称该文法为 LR(0) 文法

SLR 文法 解决了部分移入-归约冲突和归约-归约冲突，每个 SLR 文法都是 LR(1) 文法（因为 SLR 文法的要求要比 LR(1) 文法高，需要处理的冲突更多）

翻译文法 插入动作符号的文法，由翻译文法通过推导产生活动序列

活动序列 由翻译文法推导出的符号串，由终结符和动作符号组成

一遍扫描编译程序 一遍扫描即可完成整个编译工作的称为一遍扫描编译程序

活前缀 规范句型的活前缀是唯一的