

《编译原理》复习

考试题型

选择 2'x15

判断 1'x10

大题 10'x6

1. 句子一定是句型，句型不一定是句子。
句子只有终结符；句型既有终结符，也有非终结符。
2. 一个文法所有句子的集合构成该文法定义的语言
3. 所有句子都是规范句型
4. 将编译程序分成若干个“遍”是为了使程序的结构更加清晰
5. 编译程序的功能是将高级语言源程序编译成目标程序
6. 编译程序完成高级语言程序到低级语言程序的等价翻译
7. 编译程序绝大多数时间花在表格管理上
8. 与高级语言相比，汇编语言编写的程序通常执行效率更高
9. 正则表达式和自动机在接受语言的能力上是等价的,并且一个正则表达式有可能等价于多个自动机
10. 由文法定义可知,文法所定义的语言是由该文法的开始符推导出的所有的终极字符串的集合
11. 句柄：最左直接短语
12. 一个句型的直接短语不是唯一的
13. 一个文法的句型的句柄可能是不唯一的
14. 一个无二义文法的句型的句柄是唯一的
15. 如果文法 G 是无二义性的，则它的任何句子 α ：最左推导和最右推导对应的语法树必定相同
16. 四种文法的四种语言之间的关系是： $L_3 \subseteq L_2 \subseteq L_1 \subseteq L_0$
17. 如果一个文法存在某个句子对应两棵不同的语法树，则文法是二义性的
18. 不存在一个算法，它能在有限步骤内确切判定任给的一个文法是否为二义的
19. 文法的二义性是不可判定的

20. 将识别各类单词的有限自动机合并后得到的有限自动机可能是 DFA, 也可能是 NFA
21. 与 DFA 相比, NFA 的非确定性体现在允许有多个开始状态、在没有任何输入的情况下允许进行状态转换
22. NFA 等价的 DFA 可以有多个,但最简 DFA 就一个
23. 不是 DFA 的构成成分的是初始状态集合, 根据 DFA 的状态可知, DFA 只能**唯一**确定的起始状态
24. 一个 DFA 识别的语言是一个无限集合,则该 DFA 的状态图一定含有回路
25. 有限自动机 M 和 N 等价是指 M 和 N 识别的字符串集合相同
26. 一个句型的直接短语可能不只一个, 但句柄是唯一的
27. 句柄对应的是文法某个产生式的右部
28. 由文法的开始符号出发经过若干步(包括 0 步)推导产生的文法符号序列称为句型
29. 词法分析器的输入是源程序
30. 词法分析器的输出结果是单词的种别编码和自身值
31. 词法分析器不可以发现括号不匹配、操作数类型不匹配、标识符重复声明、除法溢出, 可以识别出数值常量、过滤源程序中的注释、扫描源程序并识别单词、出现非法符号错误 (语法错误)
32. 编译程序中的语法分析器接受以**单词**为单位的输入,并产生有关信息供以后各阶段使用
33. 在语法分析处理中, FIRST 集合、 FOLLOW 集合、 SELECT 集合均是终极符集
34. 采用自上而下分析, 必须**消除回溯**
35. 简单优先分析法 (自底向上) 是一种规范归约, 但效率较低, 需要考虑文法的所有符号, 包括终结符与非终结符的优先关系
36. 算法优先分析法不是规范归约方法, 但效率较高, 只考虑终结符之间的优先关系
37. 一个 LL(1)文法一定是无二义的
38. 每个基本块都可以用一个 DAG (有向无环图) 表示

39. 每个过程的活动记录的体积在编译时可静态确定
40. 采用三元式实现三地址码时，不利于对中间代码进行优化
41. 一个优先表不一定存在相应的优先函数
42. 正规文法产生的语言都可以用上下文无关文法来描述
43. 3 型文法（正规）一定是 2 型文法（上下文无关）
44. 如果一个语言的句子是无穷的，则定义该语言的文法一定是递归的
45. 正规式的等价性：若正规式 R_1 与 R_2 描述的正规集相同，则 R_1 与 R_2 等价
46. 若 $L(R_1) = L(R_2)$ ，则 R_1 与 R_2 等价
47. 设 r 和 s 分别是正规式，则有 $L(rs) = L(r)L(s)$
48. 设 r 和 s 分别是正规式，则有 $L(r|s) = L(r) \cup L(s)$
49. NFA 相较于 DFA，允许初态不唯一，多值映射，允许空移
50. DFA 用一条确定的路径来识别某一字符串
51. DFA 与 NFA 的特例
52. 两个状态 s 和 t 等价的条件：一致性条件、蔓延性条件
53. 规范规约（最左）和规范推导（最右）是互逆的两个过程
54. 在规范规约过程中，若符号栈中出现了句柄，则栈顶一定出现了某个规则式的右部（一个句型的句柄一定是某产生式的右部）
55. 反之，若一个句型中出现了某产生式的右部，则此右部不一定是该句型的句柄
56. 3 型文法： $A \rightarrow aB$ $B \rightarrow bA$ (右线性) 或者 $A \rightarrow Ba$ $B \rightarrow Ab$ (左线性)，两者不可以混合)
57. 对任意一个右线性文法 G ，都存在一个 NFA/DFA M ，满足 $L(G) = L(M)$
58. 一个正规语言可以对应**多个**正规文法
59. 一个正规语言可以对应**多个**DFA
60. 一个状态转换图可用于识别一定的字符串
61. 一个有限状态自动机中，不只有一个的初态
62. 下推自动机识别的语言是 2 型语言
63. 两个正规集相等的必要条件是它们产生的符号串是相同的
64. 并不是每个文法都能改写成 LL(1)文法

65. 含有公共左因子的不是 LL(1)文法，递归、右递归、2 型文法是 LL(1)文法
66. 算符优先文法：任何两个终结符之间至多只有一种优先关系
67. 一个优先关系表对应的优先函数 f 和 g 的值不是唯一的
68. 如果重复过程中有一个值大于 $2n$ ，则表明该算符优先文法不存在优先函数(n 为终结符的个数)
69. 当用产生式 $A \rightarrow \alpha$ 归约时，LR(0)无论面临什么输入符号都进行归约；SLR(1)则仅当面临的输入符号 $a \in \text{FOLLOW}(A)$ 时进行归约；LR(1)则在把 α 归约为 A 的规范句型的前缀 $\beta A \alpha$ 的前提下，当 α 后跟终结符 a 时，才进行归约。
70. 在 SLR (1) 分析法的名称中，S 的含义是简单的
71. LR 分析法在自左至右扫描输入串时就能发现错误，但不能准确地指出出错地点
72. LR 文法肯定是无二义的，一个二义文法决不会是 LR 文法
73. 判断文法 G 是否是 LALR(1)文法，则文法 G 必须是 LR(1)文法
74. 语法分析时不是必须先消除文法中的左递归
75. LR 分析法每次规约的是当前句型的句柄
76. LR 分析中的活前缀不一定包含某句型的句柄的一部分或者全部，一定要是规范句型
77. LR(1)和 SLR(1)中的“1”无区别
78. 一个 SLR(1)文法一定也是 LALR(1)文法
79. 自底向上语法分析能力：LR(1) > LALR(1) > SLR(1) > LR(0)
80. 自底向上语法分析包含关系：LR(0) \subseteq SLR(1) \subseteq LR(1)
81. 一个 LALR(1)文法一定也是 LR(1)文法
82. LR 分析法（规范归约）采用移进规约技术，其规约过程是规范的
83. 在规范归约中，用句柄来刻画可归约串。规范归约就是句柄归约
84. 在算符优先分析法中，用最左素短语来刻画可归约串
85. 同心集的合并不会产生新的“移进-归约”冲突
86. 同心集合并可能会产生的新冲突为归约-归约冲突
87. 使用语法制导翻译方法的编译程序能同时进行语法分析和语义分析
88. 每个过程的活动记录的体积在编译时可静态确定

- 89. 动态数组的存储空间在编译时就可以完全确定
- 90. 中间代码生成所依据的是语义规则
- 91. 删除多余运算的优化方法不是针对循环优化进行的
- 92. 代码优化的原则：等价原则、有效原则、合算原则
- 93. 代码优化时所依据的是等价变换原则
- 94. 与机器无关的优化：局部优化、循环优化、全局优化
- 95. 与机器有关的优化：寄存器优化、多处理机优化、特殊指令优化
- 96. 局部优化：合并已知量、删除公共子表达式（删除多余运算）、删除无用赋值
- 97. 循环优化：代码外提、强度削弱、删除归纳变量
- 98. 全局优化：合并已知量、删除全局公共子表达式、复写传播、代码外提
- 99. 编译程序使用说明标识符的过程或函数的静态层次区别标识符的作用域
- 100. 一个文法 G 别是 $LL(1)$ 文法的充要条件是：
 - (1) $FIRST(\alpha) \cap FIRST(\beta) = \Phi$
 - (2) 如果 $\beta \Rightarrow^* \epsilon$, $FIRST(\alpha) \cap FOLLOW(A) = \Phi$
- 101. 中间代码形式包括三元式、四元式、逆波兰式
- 102. 代码优化的目的是节省时间和空间
- 103. 代码生成阶段的主要任务是把中间代码变换成依赖具体机器的目标代码
- 104. 通常把编译过程分为分析前端与综合后端两大阶段。词法、语法和语义分析是对源程序的(分析)，中间代码生成、代码优化与目标代码的生成则是对源程序的(综合)
- 105. 四元式之间的联系是通过临时变量实现
- 106. 堆式动态分配申请和释放存储空间遵守任意原则

短语：每棵子树的叶子

直接短语：每棵直接子树的叶子

句柄：最左直接短语

素短语：至少包含一个终结符且不包含更小的素短语

有穷自动机的确定化(NFA -> DFA)

构造状态集合 I 的闭包

求状态集合的 a 弧转换

合并状态子集，并重命名

证明两个正规式等价

构造它们的最小 DFA

LL(1)文法

文法改写，消除左递归，提取公共子表达式

FIRST FOLLOW SELECT 集，判断是否冲突

分析字符串：步骤、分析栈(#S)、剩余字符串、推导

（逆序入栈）

算符优先文法

计算 FIRSTVT 和 LASTVT 集合

构造算符优先关系表判断任何两个终结符之间是否只有一种优先关系来确定是否是算符优先文法

（上表中每一项无重复值，即任一终结符对之间至多只有 \leq 3 种关系中的任一种，所以是算符优先文法）

算符优先分析表头：步骤、栈、优先关系、当前符号、剩余符号串、移进\归约

优先关系表、优先函数(f, g 初值为 1，小值向大值看齐)

判断 LR(0)文法：

构造 LR(0)项目集，如果存在移进-归约冲突，则该文法不是 LR(0)文法

判断 SLR(1)文法：

基于 LR(0)文法，找移进-归约项目，然后对应符号与 FOLLOW 集是否存在交集，

若存在则不是 SLR(1)文法

判断 LR(1)文法:

当状态 I 发生移进-归约或归约-归约冲突时, 若 $\{a_1, a_2, \dots, a_m\}$ 、 $\{u_1\}$ 、 $\{u_2\}$ 、...、 $\{u_n\}$ 两两相交为空, 则冲突可用 LR(1)文法解决

判断 LALR(1)文法:

合并同心集后无 (归约-归约) 冲突 (核相同, 向前搜索符不同), 则是 LALR(1) 文法

LALR(1): 同心集合并 $\left[\begin{array}{l} A \rightarrow d \cdot, a/c \\ B \rightarrow d \cdot, a/c \end{array} \right]$ 有归约-归约冲突

逆波兰表达式

注意运算符的优先级即可

算术运算符 > 关系运算符 > 逻辑运算符

SLR(1)自底向上文法解题步骤

- 1) 拓广文法
- 2) 构造它的 LR(0)项目集规范族
- 3) 构造分析表, 状态、ACTION(V_T)、GOTO(V_N)
- 4) 对输入串给出分析过程, 步骤、状态栈、符号栈、输入串、ACTION、GOTO

已知文法 $G[S]$

$S \rightarrow S * aF \mid aF \mid *aF$

$F \rightarrow +aF \mid +a$

消除文法左递归和提公共左因子。

消除左递归

$S \rightarrow aFS' \mid *aFS'$

$S' \rightarrow *aFS' \mid \epsilon$

$F \rightarrow +aF \mid +a$

提公共左因子, 文法 $G'(S)$

$S \rightarrow aFS' \mid *aFS'$

$S' \rightarrow *aFS' \mid \epsilon$

$F \rightarrow +aF'$

$F' \rightarrow F \mid \epsilon$

2) 计算FIRST集

• 如何求First (X)集 ($X \in V$) (P73)

1. 若 $X \rightarrow a$, 则 $a \in \text{First}(X)$ ($a \in V_T$)
若 $X \rightarrow \varepsilon$, 则 $\varepsilon \in \text{First}(X)$
2. 若 $X \rightarrow Y$, 且 $Y \in V_N$, 则 $\text{First}(Y) \in \text{First}(X)$
3. 若 $X \rightarrow Y_1 Y_2 \dots Y_n$, 且 $Y_1, Y_2, \dots, Y_n \in V_N$ ($1 \leq i \leq n$)
 - 1) 若 $Y_1 Y_2 \dots Y_{i-1} \xRightarrow{*} \varepsilon$, 则 $\text{First}(Y_i) - \{\varepsilon\}$ 加入 $\text{First}(X)$ 中
 - 2) 若 $Y_1 Y_2 \dots Y_n \xRightarrow{*} \varepsilon$, 则 $\varepsilon \in \text{First}(X)$

反复使用1、2、3, 直到 $\text{First}(X)$ 不再增大为止。

3) 计算FOLLOW集

• 如何求Follow (U) ($U \in V_N$)

1. 对文法的开始符号U, $\# \in \text{Follow}(U)$
2. 若 $A \rightarrow \alpha U \beta$, 则 $\text{First}(\beta) \setminus \varepsilon$ 加入到 $\text{Follow}(U)$ 中
3. 若 $A \rightarrow \alpha U$, 或 $A \rightarrow \alpha U \beta$, 但 $\beta \xRightarrow{*} \varepsilon$, 则 $\text{Follow}(A)$ 加入到 $\text{Follow}(U)$ 中。

计算 FIRST FOLLOW 集中, 如果集中出现大量空时要格外注意, 要把情况考虑全

4) 计算SELECT集

• 如何求SELECT集

对文法G的产生式 $A \rightarrow \alpha$:

若 $\alpha \not\xRightarrow{*} \varepsilon$, 则 $\text{SELECT}(A \rightarrow \alpha) = \text{FIRST}(\alpha)$

若 $\alpha \xRightarrow{*} \varepsilon$, 则 $\text{SELECT}(A \rightarrow \alpha) = \text{FIRST}(\alpha) \setminus \varepsilon \cup \text{FOLLOW}(A)$

构造 FIRSTVT 集合 和 LASTVT 集合

■ 构造FIRSTVT(P)集合的规则:

(1) 若有产生式 $P \rightarrow a \dots$ 或 $P \rightarrow Qa \dots$, 则 $a \in \text{FIRSTVT}(P)$

直接性

(2) 若 $a \in \text{FIRSTVT}(Q)$, 且有产生式 $P \rightarrow \dots Q \dots$,

传递性

则 $a \in \text{FIRSTVT}(P)$

■ 构造LASTVT(P)集合的规则:

(1) 若有产生式 $P \rightarrow \dots a$ 或 $P \rightarrow \dots aQ$, 则 $a \in \text{LASTVT}(P)$

直接性

(2) 若 $a \in \text{LASTVT}(Q)$, 且有产生式 $P \rightarrow \dots Q \dots$,

传递性

则 $a \in \text{LASTVT}(P)$

终结符可以隔一位, 但非终结符不行

试构造与下面 $G(S)$ 等价的无左递归的文法。

$G(S): S \rightarrow Sa | Nb | c$

$N \rightarrow Sd | Ne | f$

解: $S \rightarrow fN'bS'|cS', S' \rightarrow aS'|dN'bS'|\epsilon, N' \rightarrow eN'|\epsilon$

划分基本块的原则

入口语句

1. 四元式序列的第一个语句
2. 由条件语句转移后的目标语句
3. 转移语句的后继语句

出口语句

1. 下一个入口语句的前导语句
2. 转移语句（包括其本身）
3. 停语句（包括其本身）

删除没有被纳入某一基本块中的语句

划分基本块(n_1, n_2, \dots)后, 进而画出控制流图

求控制流图中的循环:

1. 求流图中各结点的必经结点集 $D(n)$, 要算上本身
2. 求回边 (有向边 $a \rightarrow b$ DOM $a = a \rightarrow b$ 是回边), 查看 a 的 $D(n)$ 中是否存在 b
3. 根据回边求循环, b 是唯一的入口, a 是出口。从出口往前找, 直到找到入口为止的所有结点

$n_i \text{ DOM } n_j$: n_i 是 n_j 的必经结点, 即到达 n_j 必须要经过 n_i