编译原理考试题及答案汇总

一、选择

1．将编译程序分成若干个“遍”是为了_B__。

A ． 提高程序的执行效率

B.使程序的结构更加清晰

C. 利用有限的机器内存并提高机器的执行效率

D.利用有限的机器内存但降低了机器的执行效率

2．正规式 MI 和 M2 等价是指__C__。

A ． MI 和 M2 的状态数相等　B.M1 和 M2 的有向弧条数相等。

C .M1 和 M2 所识别的语言集相等 D. M1 和 M2 状态数和有向弧条数相等

3．中间代码生成时所依据的是 _C_。

A．语法规则　B．词法规则　C．语义规则　D．等价变换规则

4．后缀式 ab+cd+/可用表达式__B_来表示。

A．　a+b/c+d　B．(a+b)/(c+d) C．　a+b/(c+d) D．　a+b+c/d

6．　一个编译程序中，不仅包含词法分析，_A____，中间代码生成，代码优化，　目标代码生成等五个部分。

A．( ) 语法分析　B．( )文法分析　C．( )语言分析 D．( )解释分析

7．　词法分析器用于识别__C___。

A．( ) 字符串 B．( )语句 C．( )单词　D．( )标识符

8．　语法分析器则可以发现源程序中的___D__。

A．( ) 语义错误　　B．( ) 语法和语义错误

C．( ) 错误并校正　　D．( ) 语法错误

9．　下面关于解释程序的描述正确的是__B___。

(1) 解释程序的特点是处理程序时不产生目标代码

(2) 解释程序适用于 COBOL 和 FORTRAN 语言

(3) 解释程序是为打开编译程序技术的僵局而开发的

A．( ) (1)(2)　　B．( ) (1)　　C．( ) (1)(2)(3)　　D．( ) (2)(3)

10．　解释程序处理语言时 ， 大多数采用的是__B___方法。

A．( ) 源程序命令被逐个直接解释执行

B．( ) 先将源程序转化为中间代码 ， 再解释执行

C．( ) 先将源程序解释转化为目标程序 ， 再执行

D．( ) 以上方法都可以

11．　编译过程中 ， 语法分析器的任务就是__B___。

(1) 分析单词是怎样构成的　　(2)　分析单词串是如何构成语句和说明的

(3) 分析语句和说明是如何构成程序的　(4) 分析程序的结构

A．( ) (2)(3) B．( ) (2)(3)(4)C．( ) (1)(2)(3) D．( ) (1)(2)(3)(4)

12．　编译程序是一种___C__。

A．( ) 汇编程序 B．( ) 翻译程序 C．( ) 解释程序 D．( ) 目标程序

13．　文法 G 所描述的语言是_C____的集合。

A．( ) 文法 G 的字母表 V 中所有符号组成的符号串

B．( ) 文法 G 的字母表 V 的闭包 V* 中的所有符号串

C．( ) 由文法的开始符号推出的所有终极符串

D．( ) 由文法的开始符号推出的所有符号串

14．　文法分为四种类型，即 0 型、1 型、2 型、3 型。其中 3 型文法是___B__。

A．( ) 短语文法 B．( ) 正则文法 C．( ) 上下文有关文法 D．( ) 上下文无关文法

15．　一个上下文无关文法 G 包括四个组成部分，它们是：一组非终结符号，一 组终结符号，一个开始符号，以及一组 __D___。

A．( ) 句子 B．( ) 句型 C．( ) 单词 D．( ) 产生式

16．　通常一个编译程序中，不仅包含词法分析，语法分析，中间代码生成，代码优化，目 标代码生成等五个部分，还应包括_C____。

A.（）模拟执行器　　B．（）解释器
C.（）表格处理和出错处理　　D.（）符号执行器

17. 文法 G[N]=（{b}，{N，B}，N，{N→b｜bB，B→bN}），该文法所描述 的语言是 C

A.（）L(G[N])={bi｜ i ≥ 0}　　B.（）L(G[N])={b2i｜ i≥ 0}
C.（）L(G[N])={b2i+1｜ i ≥ 0} D.（）L(G[N])={b2i+1｜ i ≥ 1}

18. 一个句型中的最左_B____称为该句型的句柄。

A.（）短语　　B.（）简单短语　　C.（）素短语　　D.（）终结符号

19. 设 G 是一个给定的文法，S 是文法的开始符号，如果 S->x(其中 x∈V*)，则称 x 是文法 G 的一个___B__。

A.（）候选式　　B．（）句型　　C.（）单词　　D.（）产生式

20. 文法 G[E]：

E →T｜E ＋ T
T →F｜T * F
F →a｜（E）

该文法句型 E ＋ F * (E ＋ T) 的简单短语是下列符号串中的_____。

①（E ＋ T）　②E ＋ T　③F　④ F * (E ＋ T)

A.（）① 和 ③　　B.（）② 和 ③　　C.（）③ 和 ④ D.（）③

21. 若一个文法是递归的，则它所产生的语言的句子__A___。

A.（）是无穷多个　　B．（）是有穷多个
C.（）是可枚举的　　D.（）个数是常量

22. 词法分析器用于识别___C__。

A.（）句子　　B．（）句型　　C.（）单词　　D.（）产生式

23. 在语法分析处理中， FIRST 集合、 FOLLOW 集合、 SELECT 集合均是_B____。

A．（）非终极符集　　B．（）终极符集　　C.（）字母表　　D．（）状态集

24. 在自底向上的语法分析方法中，分析的关键是__A___。

A．（）寻找句柄　　B．（）寻找句型　　C．（）消除递归　　D．（）选择候选式

25. 在 LR 分析法中，分析栈中存放的状态是识别规范句型___C__的 DFA 状态。

A．（）句柄　　B．（）前缀　　C．（）活前缀　　D．（）LR(0) 项目

26. 文法 G 产生的__D___的全体是该文法描述的语言。

A.（）句型 B.（）终结符集 C.（）非终结符集 D.（）句子

27. 若文法 G 定义的语言是无限集，则文法必然是 ___A__

A.（）递归的　　B．（）前后文无关的
C．（）二义性的　　D.（）无二义性的

28. 四种形式语言文法中，1 型文法又称为 __A___文法。

A.（）短语结构文法　　B．（）前后文无关文法
C.（）前后文有关文法　　D.（）正规文法

29. 一个文法所描述的语言是__A___。

A.（）唯一的　　B.（）不唯一的
C.（）可能唯一，好可能不唯一　　D.（）都不对

30. __B___和代码优化部分不是每个编译程序都必需的。

A.（）语法分析 B．（）中间代码生成
C.（）词法分析　　D.（）目标代码生成

31. _B____是两类程序语言处理程序。

A.（）高级语言程序和低级语言程序　　B．（）解释程序和编译程序
C.（）编译程序和操作系统　　D.（）系统程序和应用程序

32. 数组的内情向量中肯定不含有数组的_A____的信息。

A．（）维数　　B.（）类型　　C.（）维上下界　　D.（）各维的界差

33. 一个上下文无关文法 G 包括四个组成部分，它们是：一组非终结符号，一组终结符号，一个开始符号，以及一组 __D___。

A.（）句子　　B.（）句型

C.（ ）单词　　D.（ ）产生式

34. 文法分为四种类型，即 0 型、1 型、2 型、3 型。其中 2 型文法是___D__。

A．（ ）短语文法　　B．（ ）正则文法

C.（ ）上下文有关文法　　D.（ ）上下文无关文法

35. 一个上下文无关文法 G 包括四个组成部分，它们是：一组非终结符号，一组终结符号，一个开始符号，以及一组 __D___。

A.（ ）句子　　B.（ ）句型　C.（ ）单词　　D.（ ）产生式

36. __A___是一种典型的解释型语言。

A.（ ）BASIC B.（ ）C C.（ ）FORTRAN D.（ ）PASCAL

37. 与编译系统相比，解释系统___D__。

A.（ ）比较简单 ， 可移植性好 ， 执行速度快

B.（ ）比较复杂 ， 可移植性好 ， 执行速度快

C ．（ ）比较简单 ， 可移植性差 ， 执行速度慢

D.（ ）比较简单 ， 可移植性好 ， 执行速度慢

38. 用高级语言编写的程序经编译后产生的程序叫__B___。

A.（ ）源程序 B ．（ ）目标程序 C.（ ）连接程序 D.（ ）解释程序

39. 编写一个计算机高级语言的源程序后 ， 到正式上机运行之前，一般要经过___B__这几步:

(1) 编辑　(2) 编译　(3) 连接　(4) 运行

A ．（ ）(1)(2)(3)(4) B.（ ）(1)(2)(3) C.（ ）(1)(3) D.（ ）(1)(4)

40. 把汇编语言程序翻译成机器可执行的目标程序的工作是由__A___完成的。

A.（ ）编译器　　B.（ ）汇编器

C.（ ）解释器 D.（ ）预处理器

41. 词法分析器的输出结果是__C___。

A.（ ）单词的种别编码　B.（ ）单词在符号表中的位置

C.（ ）单词的种别编码和自身值　　D.（ ）单词自身值

42. 文法 G ：S→xSx|y 所识别的语言是__C___。

A.（ ）xyx B.（ ）(xyx)* C ．（ ）xnyxn(n≥0) D.（ ）x*yx*

43. 如果文法 G 是无二义的，则它的任何句子α___A__。

A.（ ）最左推导和最右推导对应的语法树必定相同

B.（ ）最左推导和最右推导对应的语法树可能不同

C.（ ）最左推导和最右推导必定相同

D.（ ）可能存在两个不同的最左推导，但它们对应的语法树相同

44. 构造编译程序应掌握___D___。

A.（ ）源程序　　B ．（ ）目标语言

C.（ ）编译方法 D.（ ）以上三项都是

45. 四元式之间的联系是通过__B___实现的。

A.（ ）指示器　　B ．（ ）临时变量

C.（ ）符号表　　D.（ ）程序变量

46. 表达式(┐A ∨B)∧(C∨D)的逆波兰表示为___B__。

A ．（ ）┐AB∨∧CD∨　　B ．（ ）A┐B∨CD∨∧

C.（ ）AB∨┐CD∨∧　　D.（ ）A┐B ∨∧CD∨

47. 优化可生成__D___的目标代码。

A.（ ）运行时间较短 B.（ ）占用存储空间较小

C.（ ）运行时间短但占用内存空间大　　D.（ ）运行时间短且占用存储空间小

48. 下列__C____优化方法不是针对循环优化进行的。

A ．（ ）强度削弱　　B ．（ ）删除归纳变量

C.（ ）删除多余运算　　D.（ ）代码外提

49. 编译程序使用__B___区别标识符的作用域。

A ．（ ）说明标识符的过程或函数名

B.（ ）说明标识符的过程或函数的静态层次

C.（ ）说明标识符的过程或函数的动态层次
D．（ ）标识符的行号

50. 编译程序绝大多数时间花在___D__ 上。
A.（ ）出错处理 B.（ ）词法分析 C.（ ）目标代码生成 D.（ ）表格管理

51. 编译程序是对__D___。
A.（ ）汇编程序的翻译 B．（ ）高级语言程序的解释执行
C.（ ）机器语言的执行 D.（ ）高级语言的翻译

52. 采用自上而下分析，必须__C___。
A.（ ）消除左递归 B．（ ）消除右递归
C.（ ）消除回溯 D.（ ）提取公共左因子

53. 在规范归约中，用__B___来刻画可归约串。
A.（ ）直接短语 B.（ ）句柄
C.（ ）最左素短语 D.（ ）素短语

54. 若 a 为终结符，则 A ->α・aβ 为__B___项目。
A.（ ）归约 B．（ ）移进 C.（ ）接受 D.（ ）待约

55. 间接三元式表示法的优点为__A___。
A.（ ）采用间接码表，便于优化处理 B．（ ）节省存储空间，不便于表的修改
C.（ ）便于优化处理，节省存储空间 D.（ ）节省存储空间，不便于优化处理

56. 基本块内的优化为___B__。
A．（ ）代码外提，删除归纳变量 B.（ ）删除多余运算，删除无用赋值
C.（ ）强度削弱，代码外提 D.（ ）循环展开，循环合并

57. 在目标代码生成阶段，符号表用___D__。
A.（ ）目标代码生成 B.（ ）语义检查 C.（ ）语法检查 D.（ ）地址分配

58. 若项目集 Ik 含有 A ->α・，则在状态 k 时，仅当面临的输入符号 a∈FOLLOW(A) 时，才采取“A ->α・”动作的一定是__D___。
A．（ ）LALR 文法 B.（ ）LR(0)文法
C.（ ）LR(1)文法 D.（ ）SLR(1)文法

59. 堆式动态分配申请和释放存储空间遵守__D___原则。
A．（ ）先请先放 B.（ ）先请后放
C.（ ）后请先放 D．（ ）任意

二、判断

1．计算机高级语言翻译成低级语言只有解释一种方式。(×)
2．在编译中进行语法检查的目的是为了发现程序中所有错误。(×)
3．甲机上的某编译程序在乙机上能直接使用的必要条件是甲机和乙机的操作系统功能完全相同。（√ ）
4．正则文法其产生式为 A->a ， A->Bb, A,B∈VN ， a 、 b∈VT 。（×）
5．每个文法都能改写为 LL(1) 文法。（√）
6．递归下降法不允许任一非终极符是直接左递归的。（√）
7．算符优先关系表不一定存在对应的优先函数。（×）
8．自底而上语法分析方法的主要问题是候选式的选择。（×）
9．LR 法是自顶向下语法分析方法。（×）
10．简单优先文法允许任意两个产生式具有相同右部。（×）
11．“ 用高级语言书写的源程序都必须通过编译，产生目标代码后才能投入运行 ”这种说法。（ × ）
12．若一个句型中出现了某产生式的右部，则此右部一定是该句型的句柄。（ × ）
13．一个句型的句柄一定是文法某产生式的右部。（ √）
14．在程序中标识符的出现仅为使用性的。（ × ）
15．仅考虑一个基本块，不能确定一个赋值是否真是无用的。（ √ ）
16．削减运算强度破坏了临时变量在一基本块内仅被定义一次的特性。（ √ ）

17．在中间代码优化中循环上的优化主要有不变表达式外提和削减运算强度。（ × ）
18．数组元素的地址计算与数组的存储方式有关。（ × ）
19．编译程序与具体的机器有关,与具体的语言无关。（ × ）
20．递归下降分析法是自顶向上分析方法。（ √ ）
21．产生式是用于定义词法成分 的一种书写规则。（ × ）
22．LR 法是自顶向下语法分析方法。（×）
23．在 SLR（ 1 ）分析法的名称中，S 的含义是简单的。（ √）
24．综合属性是用于 “ 自上而下 ” 传递信息。（ × ）
25．符号表中的信息栏中登记了每个名字的 属性和特征等有关信息 ，如类型、种属、所占单元大小、地址等等。（ × ）
26．程序语言的语言处理程序是一种应用软件。（ × ）
27．一个 LL(1)文法一定是无二义的。（ × ）
28．正规文法产生的语言都可以用上下文无关文法来描述。（ × ）
29．一张转换图只包含有限个状态，其中有一个被认为是初态，最多只有一个终态。（ √）
30．目标代码生成时，应考虑如何充分利用计算机的寄存器的问题。（ × ）
31．逆波兰法表示的表达式亦称后缀式 。（ √ ）
32．如果一个文法存在某个句子对应两棵不同的语法树，则称这个文法是二义的。（ √ ）
33．数组元素的地址计算与数组的存储方式有关。（ × ）
34．对于数据空间的存贮分配， FORTRAN 采用动态贮存分配策略。（ × ）
35．编译程序是对高级语言程序的解释执行。（ × ）
36．一个有限状态自动机中，有且仅有一个唯一的终态。（ × ）
37．语法分析时必须先消除文法中的左递归 。（ × ）
38．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。（ √ ）
39．逆波兰表示法表示表达式时无须使用括号。（ √ ）
40．静态数组的存储空间可以在编译时确定。（ × ）
41．进行代码优化时应着重考虑循环的代码优化，这对提高目标代码的效率将起更大作用。（ × ）
42．两个正规集相等的必要条件是他们对应的正规式等价。（ × ）
43．一个语义子程序描述了一个文法所对应的翻译工作。（ × ）
44．r 和 s 分别是正规式，则有 L(r|s)=L(r)L(s)。（ × ）
45．确定的的自动机以及不确定的自动机都能正确地识别正集（√）
46．分析作为单独的一遍来处理较好。（ × ）
47． LR 分析器的任务就是产生 LR 分析表。（ √ ）
48．归约和规范推导是互逆的两个过程。（ √ ）
49．同心集的合并有可能产生新的“移进”/ “归约” 冲突（×）
50．1R 分析技术无法适用二义文法。（ × ）
51树形表示和四元式不便于优化，而三元式和间接三元式则便于优化。（ × ）
52序中的表达式语句在语义翻译时不需要回填技术。（ √）

三、填空

1．编译程序的工作过程一般可以划分为词法分析,语法分析,语义分析,中间代码生成,代码优化等几个基本阶段,同时还会伴有__**表格处理**___和 ___**出错处理**__。
2．若源程序是用高级语言编写的,____**目标程序**__是机器语言程序或汇编程序,则其翻译程序称为 ____**编译程序**__ 。
3．编译方式与解释方式的根本区别在于__**是否生成目标代码**___。
4．对编译程序而言,输入数据是____**源程序**__, 输出结果是__目标程序____。

5. 产生式是用于定义__语法成分__的一种书写规则。
6. 语法分析最常用的两类方法是___自上而下__和___自下而上__分析法
7. 设 G 是一个给定的文法，S 是文法的开始符号，如果 S->x(其中 x∈VT*)，则称 x 是文 法的一个__句子___。
8. 递归下降法不允许任一非终极符是直接_左___递归的。
9. 自顶向下的语法分析方法的基本思想是:从文法的__开始符号_____开始，根据给定的输入串并按照文法的产生式一步一步的向下进行__直接推导_____，试图推导出文法的__句子_____，使之与给定的输入串__匹配___。
10. 自底向上的语法分析方法的基本思想是:从输入串入手，利用文法的产生式一步一步地向上进行__直接归约__ ，力求归约到文法的__开始符号_____。
11常用的参数传递方式有___传地址__，传值和传名。
12. 在使用高级语言编程时,首先可通过编译程序发现源程序的全部__语法___错误和语义的部分错误。
13. 一个句型中的最左简单短语称为该句型的___句柄_。
14. 对于文法的每个产生式都配备了一组属性的计算规则，称为 __语义规则___ 。
15. 一个典型的编译程序中，不仅包括__词法分析___、__语法分析___、__中间代码生成___、 代码优化、目标代码生成等五个部分，还应包括表格处理和出错处理。
16. 从功能上说，程序语言的语句大体可分为__执行性_____语句和__说明性____语句两大类。
17. 产生式是用于定义__语法范畴___的一种书写规则。
18. 语法分析是依据语言的__语法_____规则进行的，中间代码产生是依据语言的__语义___规 进行的。
19. 语法分析器的输入是__单词符号串___，其输出是__语法单位___。
20. 产生式是用于定义___语法成分__的一种书写规则。
21. 逆波兰式 ab+c+ d*e- 所表达的表达式为__(a+b+c)*d-e___ 。
22. 语法分析最常用的两类方法是__自上而下___和__自下而上___分析法。
23. 计算机执行用高级语言编写的程序主要有两种途径:___解释__和__编译___。
24. 扫描器是__词法分析器___，它接受输入的__源程序___，对源程序进行___词法分析__ 并识别出一个个单词符号，其输出结果是单词符号，供语法分析器使用。
25. 自上而下分析法采用___移进__、归约、错误处理、__接受__等四种操作。
26. 一个 LR 分析器包括两部分:一个总控程序和___一张分析表_。
27. 后缀式 abc-/所代表的表达式是___a/(b-c)__。
28. 局部优化是在__基本块___范围内进行的一种优化。
29. 词法分析基于__正则_____文法进行，即识别的单词是该类文法的句子。
30. 语法分析基于__上下文无关___文法进行，即识别的是该类文法的句子。语法分析的有效 工具是__语法树___。
31. 分析句型时，应用算符优先分析技术时，每步被直接归约的是__最左素短语___，而应用 LR 分析技术时，每步被直接归约的是___句柄__。
32. 语义分析阶段所生成的与源程序等价的中间表示形式可以有__逆波兰___、___三元式表示__与___四元式表示__等。
33. 按 Chomsky 分类法，文法按照___规则定义的形式__进行分类。
34. 一个文法能用有穷多个规则描述无穷的符号串集合(语言)是因为文法中存在有___递归__定义的规则。

35. 一个名字的属性包括__**类型**___和__**作用域**___。

四、综合题

1. 已知文法 G(E)

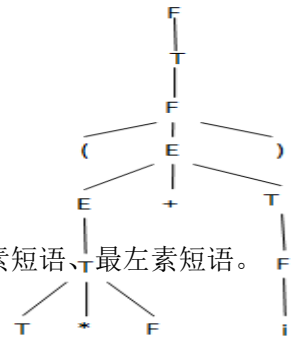
E →T|E＋T

T→F|T *F

F →(E)|i

(1)给出句型(T *F＋i)的最右推导；

(2)给出句型(T *F＋i)的短语、简单短语、句柄、素短语、最左素短语。



解：

(1) 最右推导：E ->T->F->(E)->(E ＋ T)->(E ＋ F)->(E ＋ i)->(T＋i)->(T*F＋i)

(2) 短语：(T*F＋i) ， T*F＋i ， T*F， i

简单短语：T*F,i

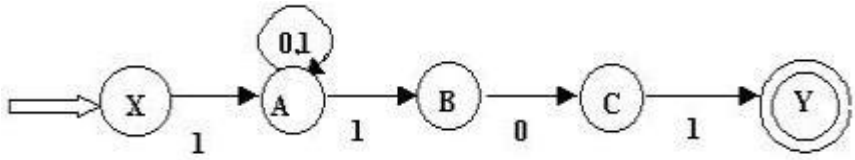
句柄：T*F

素短语：T*F,i

最左素短语：T*F

2. 构造正规式 1(0|1)*101 相应的 DFA 。

解：先构造 NFA ：



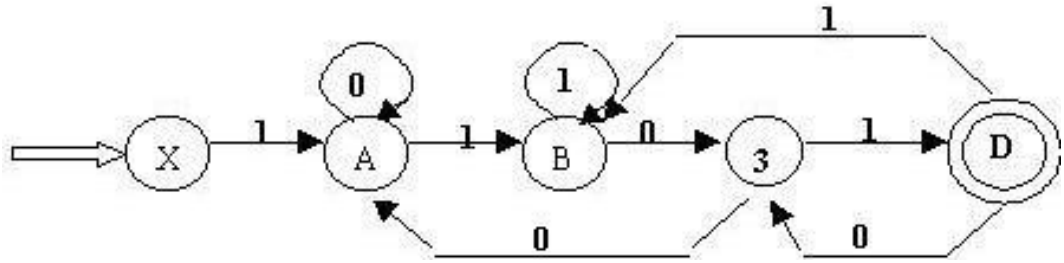
确定化：

	0	1
X		A
A	A	AB
AB	AC	AB
AC	A	ABY
ABY	AC	AB

重新命名，令 AB 为 B、AC 为 C、ABY 为 D 得：

	0	1
X		A
A	A	B
B	C	B
C	A	D
D	C	B

所以，可得 DFA 为：



3. 文法：

S->MH|a
H ->LSo| ε
K ->dML| ε
L ->eHf
M->K|bLM

判断 G 是否为 LL(1) 文法，如果是，构造 LL(1) 分析表。

解：各符号的 FIRST 集和 FOLLOW集为：

	FIRST	FOLLOW
S	{a,d,b, ε ,e}	{#,o}
M	{d, ε ,b}	{e,#,o}
H	{ ε ,e}	{#,f,o}
L	{e}	{a,d,b,e,o,#}
K	{d, ε }	{e,#,o}

各产生式SELECT集为：

	SELECT
S->MH	{d,b,e,#,o}
S->a	{a}
H ->LSo	{e}
H ->ε	{#,f,o}
K ->dML	{d}
K ->ε	{e,#,o}
L ->eHf	{e}
M->K	{d,e,#,o}
M-> bLM	{b}

预测分析表

	a	o	d	e	f	b	#
S	->a	->MH	->MH	->MH		->MH	->MH
M		->K	->K	->K		->bLM	->K
H		-> ε		->LSo	-> ε		-> ε
L				->eHf			
K		-> ε	->dML	-> ε			-> ε

由于预测分析表中无多重入口，所以可判定文法是 LL(1)的。

4. 对下面的文法 G ：

E ->TE'

E'->+E| ε

T ->FT'

T' ->T| ε

F-> PF'

F'-> *F'| ε

P->(E)|a|b|^

(1)计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。 (4 分)

(2) 证明这个方法是 LL(1) 的。(4 分)

(3) 构造它的预测分析表。(2 分)

解：（1）计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。

FIRST 集合有：

FIRST(E)=FIRST(T)=FIRST(F)=FIRST(P)={(,a,b,^};

FIRST(E')={+, ε }

FIRST(T)=FIRST(F)=FIRST(P)={(,a,b,^};

FIRST(T')=FIRST(T)U{ε}={(,a,b,^, ε};

FIRST(F)=FIRST(P)={(,a,b,^};

FIRST(F')=FIRST(P)={*, ε};

FIRST(P)={(,a,b,^};

FOLLOW 集合有：

FOLLOW(E)={),#};

FOLLOW(E')=FOLLOW(E)={),#}; FOLLOW(T)=FIRST(E')UFOLLOW(E)={+,),#};//不包含 ε

FOLLOW(T')=FOLLOW(T)=FIRST(E')UFOLLOW(E)={+,),#};

FOLLOW(F)=FIRST(T')UFOLLOW(T)={(,a,b,^,+,),#};//不包含 ε

FOLLOW(F')=FOLLOW(F)=FIRST(T')UFOLLOW(T)={(,a,b,^,+,),#};

FOLLOW(P)=FIRST(F')UFOLLOW(F)={*,(,a,b,^,+,),#};//不包含 ε

(2)证明这个方法是 LL(1)的。

各产生式的 SELECT 集合有：

SELECT(E->TE')=FIRST(T)={(,a,b,^};

SELECT(E'->+E)={+};

SELECT(E'->ε)=FOLLOW(E/)={),#}

SELECT(T->FT')=FIRST(F)={(,a,b,^};

SELECT(T'->T)=FIRST(T)={(,a,b,^};

SELECT(T'->ε)=FOLLOW(T/)={+,),#};

SELECT(F->PF')=FIRST(P)={(,a,b,^};

SELECT(F'->*F')={*};

SELECT(F'->ε)=FOLLOW(F')={(,a,b,^,+,),#};

SELECT(P->(E))={(}

SELECT(P->a)={a}

SELECT(P->b)={b}

SELECT(P->^)={^}

可见，相同左部产生式的 SELECT 集的交集均为空，所以文法 G[E]是 LL(1)文法。

(3)构造它的预测分析表。 文
法 G[E]的预测分析表如下：

	+	*	(	)	a	b	^	#
E			→TE′		→TE′	→TE′	→TE′	
E′	→+E			→ε				→ε
T			→FT′		→FT′	→FT′	→FT′	
T′	→ε		→T	→ε	→T	→T	→T	→ε
F			→PF′		→PF′	→PF′	→PF′	
F′	→ε	→*F′	→ε	→ε	→ε	→ε	→ε	→ε
P			→(E)		→a	→b	→^	

5.已知文法 G[S] 为：

S->a|^|(T)

T-> T,S|S

(1) 计算 G[S] 的 FIRSTVT 和 LASTVT 。

(2) 构造 G[S] 的算符优先关系表并说明 G[S] 是否未算符优先文法。

(3) 给出输入串 (a,a)# 的算符优先分析过程。

解：（1）各符号的 FIRSTVT 和 LASTVT：

	FIRSTVT	LASTVT
S	a、^、(	a、^、)
T	,、a、^、(	,、a、^、)

（2）算符优先关系表：

	a	(	)	,	^	#
a			⋗	⋗		⋗
(	⋖	⋖	≡		⋖	
)			⋗	⋗		⋗
,	⋖	⋖	⋗	⋗	⋖	
^			⋗	⋗		⋗
#	⋖	⋖			⋖	

（3）句子(a,a)#分析过程如下：

步骤	栈	优先关系	当前符号	剩余输入串	移进或归约
1	#	#⋖(	(	a,a)#	移进
2	#(	(⋖a	a	,a)#	移进
3	#(a	a⋗,	,	a)#	归约
4	#(F	(⋖,	,	a)#	移进
5	#(F,	,⋖a	A	)#	移进
6	#(F,a	A⋗)	)	#	归约
7	#(F,F	,⋗)	)	#	归约
8	#(F	(≡)	)	#	移进
9	#(F)	)⋗#	#		归约
10	#F	#≡#	#		接受

6. 已知文法为：

S->a|ˆ|(T)

T->T,S|S

构造它的 LR(0)分析表。

解：加入非终结符 S’，方法的增广文法为：

S’->S

S->a

S->ˆ

S->(T)

T ->T,S

T ->S 下面构造它的 LR(0)项目集规范族为：

状态 \ 当前符号	a	ˆ	(	)	,	#	S	T
I_0: S'→·S S→·a S→·ˆ S→·(T)	I_2: S→a·	I_3: S→ˆ·	I : S→(·T) T→·T,S T→·S S→·a S→·ˆ S→·(T)				I_1: S'→S·	
I_1						acc		
I_2								
I_3								
I : S→(·T) T→·T,S T→·S S→·a S→·ˆ S→·(T)	I_7	I_3	I				I_4: T→S·	I_5: S→(T·) T→T·,S
I_6: S→(T·) T→T·,S				I_7: S→(T)·	I_8: T→T,·S S→·a S→·ˆ S→·(T)			
I_4								
I_5:								
I_8: T→T,·S S→·a S→·ˆ S→·(T)	I_2	I_3	I				I_9 T→T,S·	
I_9								

从上表可看出,不存在移进- 归约冲突以及归约归约冲突，该文法是 LR(0)文法。 从而有下面的 LR(0)分析表：

状态	ACTION						GOTO	
	a	ˆ	(	)	,	#	S	T
0	S_2	S_3	S				1	
1						acc		
2	r_1	r_1	r_1	r_1	r_1	r_1		
3	r_2	r_2	r_2	r_2	r_2	r_2		
4	S_2	S_3	S				6	5
5				S_7	S_8			
6	r_5	r_5	r_5	r_5	r_5	r_5		
7	r_3	r_3	r_3	r_3	r_3	r_3		
8	S_2	S_3	S				9	
9	r	r	r	r	r	r		

7. 已知文法为：

A ->aAd|aAb| ε

判断该文法是否是 SLR(1) 文法, 若是构造相应分析表, 并对输入串 ab# 给出分析过程。

解:增加一个非终结符 S/后，产生原文法的增广文法有： S'->A A ->aAd|aAb|ε 下面构造它的 LR(0)项目集规范族为:

状态 \ 当前符号	a	b	d	#	A
I_0: S'→•A A→•aAd A→•aAb A→•	I_2: A→a•Ad A→a•Ab A→•aAd A→•aAb A→•				I_1: S'→A•
I_1: S'→A•				acc	
I_2: A→a•Ad A→a•Ab A→•aAd A→•aAb A→•	I_2				I_3: A→aA•d A→aA•b
I_3: A→aA•d A→aA•b		I : A→aAb•	I_5: A→aAd•		
I : A→aAb•					
I_5: A→aAd•					

从上表可看出，状态 I0 和 I2 存在移进- 归约冲突，该文法不是 LR(0)文法。对于 I0 来说有：FOLLOW(A)∩{a}={b,d,#}∩{a}=Φ ，所以在 I0 状态下面临输入符号为 a 时移进，为 b,d,# 时

归约，为其他时报错。对于 I2 来说有也有与 I0 完全相同的结论。这就是说，以上的移进 - 归约冲突是可以解决的，因此该文法是 SLR(1)文法。

其 SLR(1)分析表为：

状态	ACTION				GOTO
	a	b	d	#	A
0	S_2	r_4	r_4	r_4	1
1				acc	
2	S_2	r_4	r_4	r_4	3
3		S	S_5		
4	r_2	r_2	r_2	r_2	
5	r_1	r_1	r_1	r_1	

对输入串 ab#给出分析过程为:

步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	#	ab#	S_2	
2	02	#a	b#	r_3	3
3	023	#aA	b#	S	
4	0234	#aAb	#	r_2	1
5	01	#A	#	acc	