# 第一讲 绪论

1. 编译是**(高级语言的翻译)**

**2.** 用高级语言编写的程序经编译后产生的程序叫**(目标程序)**

**3. (设备管理程序)**不是编译程序的组成部分

4.  源程序是句子的集合，**(树)**可以较好地反映句子的结构。

5. 编译程序是一种**(翻译程序)**

6. 按逻辑上划分，编译程序第三步工作是**(语义分析)**

7. 编译程序中语法分析器接收以**(单词)**为单位的输入

8. 编译过程中，语法分析器的任务就是**(分析单词串是如何构成语句和声明的)**

9. 语法分析时所依据的是**(语法规则)**

10. 通常一个编译程序中，不仅包含词法分析，语法分析，语义分析，中间代码生成，代码优化，目标代码生成等六个部分，还应包括**(表格处理和出错处理)**

11. 编译程序绝大多数时间花在**(表格管理)**上。

# 第二讲 程序设计语言及其文法

12. 文法：G：S→xSx | y所识别的语言是x^nyx^n(n>=0)。

13. 给定文法A→bA|ca，为该文法句子的是(**bca** )

14. 设有文法G[S]：S->S1|S0|Sa|Sc|a|b|c，下列符号串中是该文法的句子有(**bc10** )。

15. 文法G产生的(**句子**)的全体是该文法描述的语言

16. 若文法G定义的语言是无限集，则文法必然是(**递归的**)。

17. 乔姆斯基把文法分为四种类型，即0型,1型,2型,3型。其中3型文法是(**正则文法**)。

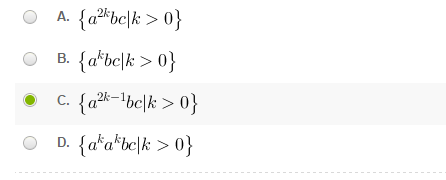
18. 一个上下文无关文法G包括四个组成部分，它们是一组非终结符号，一组终结符号，一个开始符号，以及一组(**产生式**)

19. 若一个文法是递归的，则它所产生的语言的句子(**是无穷多个**)。

20. 给定文法A→bA|cc，则符号串①cc ②bcbc ③bcbcc ④bccbcc ⑤bbbcc中，是该文法句子的是(**①⑤**)。

21. 文法E→E+E|E\*E|i的句子i\*i+i\*i有(**5** )棵不同的语法树

22. 文法 S→aaS|abc 定义的语言是( )。



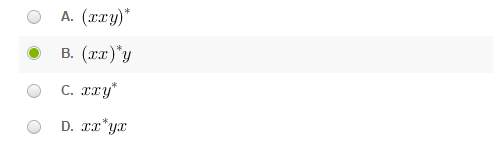
23. 文法G：S→xSx| xS|y所识别的语言是()。



24. 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**)

25. 下列符号串不可以由符号集S＝{a,b}上的正闭包运算产生的是(**ε** )。

26. 文法G：S → x xS | y 所识别的语言是( )



27. 文法G：S → xS | y 所识别的语言是( )



# 第三讲 词法分析

28. 词法分析器的输出结果是(**单词的种别编码和自身值**)

29. 词法分析器不能(**发现括号不匹配**)

30. (**不存在**)这样一些语言，它们能被确定的有穷自动机识别，但不能用正则表达式表示

31. 两个有穷自动机等价是指它们的(**所识别的语言相等**)

32. 词法分析器用于识别(**单词**)

33. 正则表达式 R1和R2 等价是指(**R1和R2 代表同一正则集**)

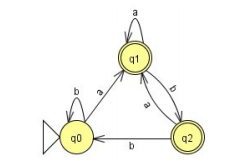
34.  已知文法G[S]:S→A1， A→A1|S0|0，与G等价的正规式是( )



35. 与(a|b)\*(a|b) 等价的正规式是( )



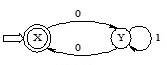
36.  如图所示自动机M，请问下列哪个字符串不是M所能识别的(**aabb**)



37. 有限状态自动机能识别(**正规语言**)

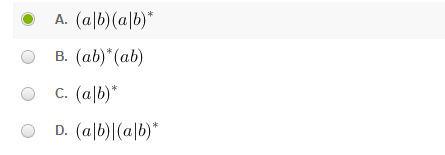
38.  (**多个初始状态的集合** )不是DFA的成分

39. DFA M(见图)接受的字集为(**含偶数个0的二进制数组成的集合**)



40. 有穷自动机M1和M2等价是指(**M1和M2所识别的语言集相等**)

41. 同正规式(a|b)+ 等价的正规式是()



42. 称有限自动机 A1和A2 等价是指(**A1和A2 所能识别的字符串集合相等**)

43. 两个DFA等价是指(**这两个DFA接受的语言相同**)

44. 如果一个正规式所代表的集合是无穷的，则它必含有的运算是(**闭包运算“\* ”** )

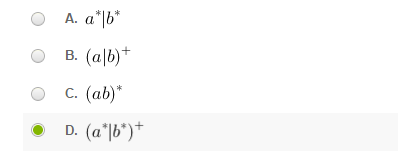
45. 词法分析器的加工对象是(**源程序**)

46. 同正规式a\*b\* 等价的文法是(**G3：S→ aS|Sb|ε**)

47. 一个正规式只能对应一个确定的有限状态自动机(**错**)

48. 一个正规语言可能对应多个正规文法(**对**)

49. 同正规式(a|b)\* 等价的正规式为( )



# 第四章 语法分析

50. 如果文法G是无二义的,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**)

51. 采用自上而下分析，不必(**消除右递归**)

52. 识别上下文无关语言的自动机是(**下推自动机**)

53. (**含有公共左因子的** )文法不是LL(1)的

54. 文法G是无二义的,则对G的任意句型α(**最左推导和最右推导对应的语法树必定相同**)

55. 在自上而下的语法分析中，应从(**文法开始符号**)开始分析

56. 一个文法G，若(**G的LL（1）分析表中不含多重定义的条目** )，则称它是LL（1）文法

57. 语法分析器的输入是（**Token序列**）

58. 在递归子程序方法中，若文法存在左递归，则会使分析过程产生(**无限循环** )

59.  LL（1）分析法中“1”的含义是在输入串中查看一个输入符号，其目的是(**确定使用哪一个产生式进行展开**)

60. 在语法分析处理中，FIRST集合、FOLLOW集合均是(**终结符集**)

61. 在编译过程中，如果遇到错误应该(**当发现错误时，跳过错误所在的语法单位继续分析下去**)

62. 已知文法G[S]:

S→eT|RT T→DR|ε R→dR|ε D→a|bd

求FIRST(S)=( **{e，d，a，b，ε}** )

63. 已知文法G[S]:

S→eT|RT T→DR|ε R→dR|ε D→a|bd

求FOLLOW(D)=（**{d,$}**）

64. FIRST集中可以含有ε(**对**)

65. FOLLOW集中可以含有ε(**错**)

66. SELECT集中可以含有ε(**错**)

67. 若a为终结符，则A→α · aβ为(**移进**)项目

68. 一个句型中的(**最左直接短语**)称为该句型的句柄

69. 在自底向上的语法分析方法中，分析的关键是(**寻找句柄**)

70. 在自顶向下的语法分析方法中，分析的关键是(**选择候选式**)

71. 若B为非终结符，则 A→a · Bb 为(**待约项目**)

72. 在规范归约中，用(**句柄**)来刻画可归约串

73.  若B为非终结符，则A→α·Bβ为(**待约**)项目

74. 下列动作中，不是自下而上分析动作的是(**展开**)

75. 下列动作中，不是自上而下分析动作的是(**移进**)

76. 若B为非终结符，则A→α· 为(**归约**)项目

77. 设有文法G[T]:

T→T\*F|F

F→F↑P|P

P→(T)|a

该文法句型T\*P↑(T\*F)的句柄是下列符号串( **P** )

78. LR分析表中的转移表（goto）是以（**非终结符**）作为列标题的

79. LR分析表中的动作表（action）是以（**终结符和结束符$**）作为列标题的

80. 下列项目中为可归约项目的是（**L→·**）

81. LR分析器的核心部分是一张分析表，该表由(**ACTION表和GOTO表**)组成

82. 一个LR(1)文法合并同心集后若不是LALR(1)文法(**则可能存在归约/归约冲突**)

83. 若状态k含有项目“A→α· ”,且仅当输入符号a∈FOLLOW(A)时，才用规则“A →α”归约的语法分析方法是(**SLR(1)分析法**)

84.  LR(1)文法都是(**无二义性但可能是左递归**)

85. 同心集合并可能会产生新的(**归约/归约**)冲突

86. 就文法的描述能力来说，有(**SLR(1) ⊂ LR(1)**)

87. 在LR(0)的Action表中，如果某行中存在标记为“rj”的栏，则(**该行必定填满“rj”**)

88. 若状态k含有项目“A→α·”，对任意非终结符a，都用规则“A →α”归约的语法分析方法是(**LR(0)分析法**)

89. 在SLR( 1)的Action表中，如果某行中存在标记为“rj”的栏，则(**该行未必填满“rj”**)

90. 编译程序的语法分析器必须输出的信息是(**语法错误信息**)

# 第八章 语法制导翻译

91. 文法G[S]及其语法制导翻译定义如下：

产生式 语义动作

S’ → S print( S.num)

S → ( L) S.num = L.num +1

S → a S.num = 0

L →L( 1), S L.num = L( 1).num + S.num

L →S L.num = S.num

若输入为( a,( a))，且采用自底向上的分析方法，则输出为(**2** )

92. 有文法G及其语法制导翻译如下所示( 语义规则中的\*和+分别是常规意义下的算术运算符)：

 E→E( 1) ∧ T {E.val = E( 1).val \* T.val}

 E→T {E.val = T.val}

 T→T( 1)# n {T.val = T( 1).val + n.val }

T→ n {T.val = n.val}

则分析句子3 ∧ 3 # 4其值为(**21**)

93. 有一语法指导定义如下：

S→bAb print “1”

A→( B print “2”

A→a print “3”

B→aA) print “4”

若输入序列为b( a( a( aa)))b，且采用自底向上的分析方法，则输出序列为(**34242421**)

94. 有一语法指导定义如下，其中+表示符号连接运算：

S→B print B.vers

B→a B.vers=a

B→b B.vers=b

B→Ba B.vers=a+B.vers

B→Bb B.vers=b+B.vers

若输入序列为abab，且采用自底向上的分析方法，则输出序列为(**Baba** )

95. 使用(**语义规则**)可以定义一个程序的意义

96. 终结符具有(**综合**)属性

97. 以下说法正确的是( )。

**A. 语义规则中的属性有两种：综合属性与继承属性**

B. 终结符只有继承属性，它由词法分析器提供

C. 非终结符可以有综合属性，但不能有继承属性

D. 属性值在分析过程中可以进行计算，但不能传递

98. 关于将L-SDD转换为SDT的规则，以下选项中，正确的是( )。

A. 将计算某个非终结符号A的继承属性的动作放在产生式的最后

B. 将计算一个产生式左部符号的继承属性的动作放置在产生式的最后

**C. 将计算某个非终结符号A的继承属性的动作插入到产生式右部中紧靠在A的本次出现之前的位置上**

D. 将每个语义动作都放在产生式的最后

99. 以下说法不正确的是( )。

**A. 如果一个S-SDD的基本文法可以使用LR分析技术，那么它的SDT可以在LL语法分析过程中实现**

B. 如果一个S-SDD的基本文法可以使用LR分析技术，那么它的SDT可以在LR语法分析过程中实现

C. 如果一个L-SDD的基本文法可以使用LL分析技术，那么它的SDT可以在LL语法分析过程中实现

D. 如果一个L-SDD的基本文法可以使用LL分析技术，那么它的SDT可以在LR语法分析过程中实现

100. 以下说法不正确的是( )。

A. 使用语法制导翻译方案的编译程序能同时进行语法分析和语义分析

B. 语法制导翻译方案( SDT )是在产生式右部中嵌入了程序片段( 称为语义动作)的CFG

C. SDD可以看作是SDT的具体实施方案

D. 将一个S-SDD转换为SDT的方法是：将每个语义动作都放在产生式的最后

101. 在非递归的预测分析过程中进行翻译，以下说法不正确的是( )。

A. 要想在非递归的预测分析过程中进行翻译，需要扩展语法分析栈

B. 非终结符A的继承属性和综合属性的计算时机不同

C. 将非终结符A的继承属性和综合属性存放在不同的记录中

**D. 综合属性在A出现之前就可以计算**

102. 在非递归的预测分析过程中进行翻译，以下说法不正确的是( )。

A. 要想在非递归的预测分析过程中进行翻译，需要扩展语法分析栈

B. 综合记录用于存放非终结符综合属性值

C. 动作记录，用来存放指向将被执行的语义动作代码的指针

**D. 综合属性存放在A本身的记录中**

103. 在非递归的预测分析过程中进行翻译，以下说法不正确的是( )。

A. 分析栈中的每一个记录都对应着一段执行代码

B. 综合记录出栈时，要将综合属性值复制给后面特定的语义动作

C. 变量展开时( 即变量本身的记录出栈时)，如果其含有继承属性，则要将继承属性值复制给后面特定的语义动作

**D. 继承属性在A的儿子们都分析完毕之后才能计算**

104. 在递归的预测分析过程中进行翻译，以下说法不正确的是( )。

A. 可以将一个递归的预测分析器扩展为一个翻译器

B. 在语法分析器中，每个非终结符A对应一个过程，在做语义分析时，要将过程扩展成

一个函数

C. 以继承属性作为函数的参数，以综合属性作为函数的返回值

D. 以综合属性作为函数的参数，以继承属性作为函数的返回值

105. 在递归的预测分析过程中进行翻译，以下说法不正确的是( )。

A. 在语法分析器中，每个非终结符A对应一个过程，在做语义分析时，要将过程扩展成

一个函数

B. 对出现在A产生式右部中的每个文法符号的每个属性都设置一个局部变量

C. 如果非终结符含有继承属性，需要将函数调用的返回值赋给相应的局部变量

D. 对于产生式右部的每个动作，将其代码复制到语法分析器，并把对属性的引用改为对

相应变量的引用

106. 以下说法不正确的是( )。

A. 语法制导翻译方案只限自底向上的分析方法

B. 给定一个以LL文法为基础的L-SDD，可以修改这个文法，并在LR语法分析过程中计

算这个新文法之上的SDD

C. 对于这个内嵌的语义动作，向文法中引入一个标记非终结符M来替换它

D. 每个标记非终结符M对应着一个空产生式M→ ε，该产生式对应着一段语义子程序，

它的任务就是完成M所替换的那个语义动作要完成的工作

107.  给定一个以LL文法为基础的L-属性定义，可以修改这个文法，并在LR语法分析过程中计算这个新文法之上的SDD(**对**)

108. 在各个非终结符之前放置语义动作来计算它的继承属性， 并在产生式后端放置语义动作计算综合属性(**对**)

109. 在各个非终结符之前放置语义动作来计算它的综合属性， 并在产生式后端放置语义动作计算继承属性(**错**)

110. 对每个内嵌的语义动作，向文法中引入一个标记非终结符来替换它。每个这样的位置都有一个不同的标记，并且对于任意一个标记M都有一个产生式M→ε(**对**)