

一、填空题：

- 1-01. 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。
- 1-02. 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。
- 1-03. 编译方式与解释方式的根本区别在于 是否生成目标代码。
- 1-04. 翻译程序是这样一种程序，它能够将 用甲语言书写的程序 转换成与其等价的 用乙语言书写的程序。
- 1-05. 对编译程序而言，输入数据是 源程序，输出结果是 目标程序。
- 1-06. 如果编译程序生成的目标程序是机器代码程序，则源程序的执行分为两大阶段：编译阶段 和 运行阶段。如果编译程序生成的目标程序是汇编语言程序，则源程序的执行分为三个阶段：编译阶段，汇编阶段 和 运行阶段。
- 1-07. 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。
- 1-08. 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。其中，词法分析器用于识别 单词。
- 1-09. 编译方式与解释方式的根本区别为是否生成目标代码。
- 2-01. 所谓最右推导是指：任何一步 \Rightarrow 都是对 中最右非终结符进行替换的。
- 2-02. 一个上下文无关文法所含四个组成部分是 一组终结符号、一组非终结符号、一个开始符号、一组产生式。
- 2-03. 产生式是用于定义 语法成分 的一种书写规则。
- 2-04. 设 $G[S]$ 是给定文法，则由文法 G 所定义的语言 $L(G)$ 可描述为： $L(G) = \{x \mid S \xRightarrow{*} x, x \in V_T^+\}$ 。
- 2-05. 设 G 是一个给定的文法， S 是文法的开始符号，如果 $S \xRightarrow{*} x$ (其中 $x \in V_T^+$)，则称 x 是文法的一个句型。
- 2-06. 设 G 是一个给定的文法， S 是文法的开始符号，如果 $S \xRightarrow{*} x$ (其中 $x \in V_T^+$)，则称 x 是文法的一个句子。
- 3-01. 扫描器的任务是从源程序中识别出一个个 单词符号。
- 4-01. 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。
- 4-02. 语法分析的任务是识别给定的终极字符串是否为给定文法的句子。
- 4-03. 递归下降法不允许任一非终极符是直接 左 递归的。
- 4-04. 自顶向下的语法分析方法的关键是 如何选择候选式 的问题。
- 4-05. 递归下降分析法是自 顶向上 分析方法。
- 4-06. 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。
- 5-01. 自底向上的语法分析方法的基本思想是：从给定的终极字符串开始，根据文法的规则一步一步的向上进行直接归约，试图归约到文法的 开始符号。
- 5-02. 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。
- 5-03. 简单优先方法每次归约当前句型的 句柄，算符优先方法每次归约当前句型的 最左素短语，二者都是不断移进输入符号，直到符号栈顶出现 可归约串 的尾，再向前找到 可归约串 的头，然后归约。
- 5-04. 在 $LR(0)$ 分析法的名称中， L 的含义是 自左向右的扫描输入串， R 的含义是 最左归约， 0 的含义是 向貌似句柄的符号串后查看 0 个输入符号。

- 5-05. 在 SLR(1) 分析法的名称中, S 的含义是 简单的。
- 6-01. 所谓属性文法是 一个属性文法是一个三元组: $A = (G, V, F)$, 一个上下文无关文法 G ; 一个属性的有穷集 V 和关于属性的断言或谓词的有穷集 F 。每个断言与文法的某产生式相联。
- 6-02. 综合属性是用于 “自下而上” 传递信息。
- 6-03. 继承属性是用于 “自上而下” 传递信息。
- 6-04. 终结符只有 综合属性, 它们由词法分析器提供。
- 7-01. 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 A 错误和 B 部分错误。
a. 语法 b. 语义 c. 语用 d. 运行
- 8-01. 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。
- 8-02. 一个过程相应的 DISPLAY 表的内容为 现行活动记录地址和所有外层最新活动记录的地址。
- 9-01. 一个过程相应的 DISPLAY 表的内容为 现行活动记录地址和所有外层最新活动记录的地址。
- 9-02. 常用的两种动态存储分配办法是 栈式 动态分配和 堆式 动态分配。
- 9-03. 常用的参数传递方式有 传地址, 传值和传名。
- 10-01. 局部优化是局限于一个 基本块 范围内的一种优化。
- 10-02. 代码优化的主要目标是如何提高 目标程序的运行速度 和如何减少 目标程序运行时所需的空间。

二、单选题:

- 1-10. 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 (1)c. 其中, (2)b 和代码优化部分不是每个编译程序都必需的。
词法分析器用于识别 (3)c, 语法分析器则可以发现源程序中的 (4)d.
(1) 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
(2) a. 语法分析 b. 中间代码生成 c. 词法分析 d. 目标代码生成
(3) a. 字符串 b. 语句 c. 单词 d. 标识符
(4) a. 语义错误 b. 语法和语义错误 c. 错误并校正 d. 语法错误
- 1-11. 程序语言的语言处理程序是一种 (1)a . (2)b 是两类程序语言处理程序, 他们的主要区别在于 (3)d .
(1) a. 系统软件 b. 应用软件 c. 实时系统 d. 分布式系统
(2) 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
(3) a. 单用户与多用户的差别 b. 对用户程序的查错能力
c. 机器执行效率 d. 是否生成目标代码
- 1-12. 汇编程序是将 a 翻译成 b, 编译程序是将 c 翻译成 d。
a. 汇编语言程序 b. 机器语言程序 c. 高级语言程序
d. a 或者 b e. a 或者 c f. b 或者 c
- 1-13. 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)

1-14. 高级语言的语言处理程序分为解释程序和编译程序两种。编译程序有五个阶段，而解释程序通常缺少

(1)e 和 (1)b . 其中 , (1)e 的目的是使最后阶段产生的目标代码更为高效 .

与编译系统相比，解释系统 (2)d 解释程序处理语言时，大多数采用的是 (3)b 方法。(4)a 就是一种典型的解释型语言。

(1): a. 中间代码生成 b. 目标代码生成 c. 词法分析 d. 语法分析 e. 代码优化

(2): 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

(3): 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，在执行 d. 以上方法都可以

(4) : a. BASIC b. C c. FORTRAN d. PASCAL

1-15. 用高级语言编写的程序经编译后产生的程序叫 b . 用不同语言编写的程序产生 b 后, 可用 g 连接在一起生成机器可执行的程序 . 在机器中真正执行的是 e .

a. 源程序 b. 目标程序 c. 函数 d. 过程

e. 机器指令代码 f. 模块 g. 连接程序 h. 程序库

1-16. 要在某一机器上为某种语言构造一个编译程序，必须掌握下述三方面的内容：c，d，f。

a. 汇编语言 b. 高级语言 c. 源语言 d. 目标语言

e. 程序设计方法 f. 编译方法 g. 测试方法 h. 机器语言

1-17. 由于受到具体机器主存容量的限制，编译程序几个不同阶段的工作往往被组合成 (1)d，诸阶段的工作往往是 (2)d 进行的。

(1) a. 过程 b. 程序 c. 批量 d. 遍

(2) a. 顺序 b. 并行 c. 成批 d. 穿插

1-18. 编译程序与具体的机器 a , 与具体的语言 a .

a. 有关 b. 无关

1-19. 使用解释程序时，在程序未执行完的情况下， a 重新执行已执行过的部分。

a. 也能 b. 不可能

1-20. 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)

1-21. 编译程序是一种常用的 b 软件.

a. 应用 b. 系统

1-22. 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)

1-23. 编译程序必须完成的工作有 a .

(1) 词法分析 (2) 语法分析 (3) 语义分析

(4) 代码生成 (5) 中间代码生成 (6) 代码优化

- a. (1)(2)(3)(4) b. (1)(2)(3)(4)(5) c. (1)(2)(3)(4)(5)(6)
d. (1)(2)(3)(4)(6) e. (1)(2)(3)(5)(6)

1-24. “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 a .

- a. 不正确 b. 正确

1-25. 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 b 完成的 .

- a. 编译器 b. 汇编器 c. 解释器 d. 预处理器

1-26. 编译程序生成的目标程序 b 是机器语言的程序 .

- a. 一定 b. 不一定

1-27. 编译程序生成的目标程序 b 是可执行的程序 .

- a. 一定 b. 不一定

1-28. 编译程序是一种 B .

- A. 汇编程序 B. 翻译程序 C. 解释程序 D. 目标程序

1-29. 按逻辑上划分，编译程序第二步工作是 C .

- A. 语义分析 B. 词法分析 C. 语法分析 D. 代码优化

1-30. 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

2-07. 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号串
D. 由文法的开始符号推出的所有符号串

2-08. 乔姆斯基 (Chomsky) 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 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

2-09. 文法 $G[N] = (\{b\}, \{N, B\}, N, \{N \rightarrow bN, B \rightarrow bN\})$ ，该文法所描述的语言是 C .

- A. $L(G[N]) = \{b^i \mid i \geq 0\}$ B. $L(G[N]) = \{b^{2i} \mid i \geq 0\}$
C. $L(G[N]) = \{b^{2i+1} \mid i \geq 0\}$ D. $L(G[N]) = \{b^{2i+1} \mid i \geq 1\}$

2-10. 一个句型中的最左 B 称为该句型的句柄。

可选项有：

- A. 短语 B. 简单短语 C. 素短语 D. 终结符号

2-11. 设 G 是一个给定的文法， S 是文法的开始符号，如果 $S \xRightarrow{*} x$ (其中 $x \in V^+$)，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

2-12. 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

2-13. 文法 $G[E]$ ：

$E \rightarrow T \mid E + T$
 $T \rightarrow F \mid T * F$

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 B。

(E + T) E + T F F * (E + T)

可选项有：

A) 和 B) 和 C) 和 D)

2-14 . 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

3-02 . 词法分析器用于识别 C。

A. 句子 B. 句型 C. 单词 D. 产生式

4-07. 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

4-08. 编译程序中语法分析器接收以 A 为单位的输入。

A. 单词 B. 表达式 C. 产生式 D. 句子

5-06 . 在自底向上的语法分析方法中，分析的关键是 D。

A. 寻找句柄 B. 寻找句型 C. 消除递归 D. 选择候选式

5-07. 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

三、是非题（下列各题，你认为正确的，请在题干的括号内打“ ”，错的打“×”。

1-31. 计算机高级语言翻译成低级语言只有解释一种方式。 (×)

1-32. 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。 (×)

1-34. 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。 (×)

2-15. 正则文法其产生式为 $A \rightarrow a, A \rightarrow Bb, A \rightarrow B, A \in V_N, a, b \in V_T$ 。 ()

4-09. 每个文法都能改写为 LL(1) 文法。 (×)

4-10. 递归下降法允许任一非终极符是直接左递归的。 ()

5-08. 算符优先关系表不一定存在对应的优先函数。 ()

5-09. 自底而上语法分析方法的主要问题是候选式的选择。 (×)

5-10. LR 法是自顶向下语法分析方法。 (×)

5-11. 简单优先文法允许任意两个产生式具有相同右部。 (×)

5-12. 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。 (×)

5-13. 一个句型的句柄一定是文法某产生式的右部。 ()

7-02. 数组元素的地址计算与数组的存储方式有关。 ()

8-03. 在程序中标识符的出现仅为使用性的。 (×)

9-04. 对于数据空间的存贮分配，FORTRAN 采用动态贮存分配策略。 (×)

9-05. 在程序中标识符的出现仅为使用性的。 (×)

四、名词解释

1-35. 扫描遍 _____ 指编译程序对源程序或中间代码程序从头到尾扫描一次。

2-16. 短语——设 $G[Z]$ 是给定文法， $w=xuy \in V^+$ ，为该文法的句型，如果满足下面两个条件：

$$Z \xRightarrow{*} xUy ;$$

$$U \xRightarrow{+} u ;$$

则称句型 xuy 中的子串 u 是句型 xuy 的短语。

2-17. 简单短语——设 $G[Z]$ 是给定文法， $w=xuy \in V^+$ ，为该文法的句型，如果满足下面两个条件：

$$Z \xRightarrow{*} xUy ;$$

$$U \Rightarrow u ;$$

则称句型 xuy 中的子串 u 是句型 xuy 的简单短语（或直接短语）。

2-18. 句柄——一个句型中的最左简单短语称为该句型的句柄。

4-11. 语法分析——按文法的产生式识别输入的符号串是否为一个句子的分析过程。

4-12. 选择符集合 SELECT——给定上下文无关文法的产生式 $A \rightarrow \alpha$ ， $A \in V_N$ ， $\alpha \in V^+$ ，若 $\alpha \xRightarrow{*}$ ，则

$SELECT(A) = FIRST(\alpha)$ ，其中如果 $\alpha \xRightarrow{*}$ ，则 $SELECT(A) = FIRST(\alpha)$

$FOLLOW(A)$ ， $FIRST(\alpha)$ 表示 $FIRST(\alpha)$ 的非 $\{\epsilon\}$ 元素。

5-14. 活前缀——若 $S \xRightarrow{*}_R A \xRightarrow{*}_R$ 是文法 G 中的一个规范推导， G 是 G 的拓广文法，符号串 α 是 A 的前缀，则称 α 是 G 的，也是 G 的一个活前缀。其中 S' 为文法开始符号。或：可归前缀的任意首部。

5-15. 可归前缀——是指规范句型的一个前缀，这种前缀不含句柄之后的任何符号。

5-16. LR(0) 项目——把产生式右部某位置上标有圆点的产生式称为相应文法的一个 LR(0) 项目。

5-17. 最左素短语——设有文法 $G[S]$ ，其句型的素短语是一个短语，它至少包含一个终结符，并除自身外不包含其它素短语，最左边的素短语称最左素短语。

6-05. 语义规则——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。

6-06. 翻译方案——将属性文法中的语义规则用花括号 $\{\}$ 括起来，插在产生式右部的合适地方，指明语义规则的计算次序，陈述一些细节，得到一种语义动作与语法分析交错的表示方法，以表述语义动作在语法分析过程中的执行时刻，称之为翻译方案。

7-03. 后缀式——一种把运算量（操作数）写在前面把算符写在后面（后缀）的表示法。即

一个表达式 E 的后缀形式可以如下定义：

(1) 如果 E 是一个变量或常量，则 E 的后缀式是 E 自身。

(2) 如果 E 是 $E_1 \text{ op } E_2$ 形式的表达式，这里 op 是任何二元操作符，则 E 的后缀式为 $E_1' E_2' \text{ op}$ ，这里 E_1' 和 E_2' 分别为 E_1 和 E_2 的后缀式。

(3) 如果 E 是 (E_1) 形式的表达式，则 E 的后缀式就是 E_1 的后缀式。

答：一个过程的活动指的是该过程的一次执行。就是说，每次执行一个过程体，产生该过程体的一个活动。

9-07. 活动记录

答：为了管理过程在一次执行中所需要的信息，使用一个连续的存储块，这样一个连续的存储块称为活动记录。

9-08. 活动的生存期

答：指的是从执行某过程体第一步操作到最后一步操作之间的操作序，包括执行过程时调用其它过程花费的时间。

10-06. 基本块的 DAG

答：一个基本块的 DAG 是一种其结点带有下列标记或附加信息的 DAG

(1) 图的叶结点（没有后继的结点）以一标识符（变量名）或常数作为标记，表示该结点代表该变量或常数的值。如果叶结点用来代表某变量 A 的地址，则用 $\text{addr}(A)$ 作为该结点的标记。通常把叶结点上作为标记的标识符加上下标 0，以表示它是该变量的初值。

(2) 图的内部结点（有后继的结点）以一运算符作为标记，表示该结点代表应用该运算符对其后继结点所代表的值进行运算的结果。

(3) 图中各个结点上可能附加一个或多个标识符，表示这些变量具有该结点所代表的值。

五、简答题：

2-19 什么是句子？什么是语言？

答：设 G 是一个给定的文法，S 是文法的开始符号，如果 $S \xRightarrow{*} x$ (其中 $x \in V_T^+$)，则称 x 是文法的一个句子。

设 G[S] 是给定文法，则由文法 G 所定义的语言 L(G) 可描述为： $L(G) = \{x \mid S \xRightarrow{*} x, x \in V_T^+\}$ 。

2-20. 已知文法 G[E] 为：

$E \rightarrow T \mid E+T \mid E-T$

$T \rightarrow F \mid T * F \mid T / F$

$F \rightarrow (E) \mid i$

该文法的开始符号（识别符号）是什么？

请给出该文法的终结符号集合 V_T 和非终结符号集合 V_N 。

找出句型 $T+T * F+i$ 的所有短语、简单短语和句柄。

解：该文法的开始符号（识别符号）是 E。

该文法的终结符号集合 $V_T = \{+, -, *, /, (,), i\}$ 。

非终结符号集合 $V_N = \{E, T, F\}$ 。

句型 $T+T * F+i$ 的短语为 i、 $T * F$ 、第一个 T、 $T+T * F+i$ ；

简单短语为 i、 $T * F$ 、第一个 T；句柄为第一个 T。

2-21. 已知文法 G[S] 为：

$S \rightarrow dAB$

$A \rightarrow aA \mid a$

$B \rightarrow Bb \mid b$

G[S] 产生的语言是什么？

G[S] 能否改写为等价的正规文法？

解：G[S] 产生的语言是 $L(G[S]) = \{da^n b^m \mid n \geq 1, m \geq 0\}$ 。

G[S] 能改写为等价的正规文法，其改写后的等价的正规文法 G[S'] 为：

$S' \rightarrow dA$

$A' \rightarrow aA \mid aB \mid a$

$B' \rightarrow bB \mid b$

2-22. 设有语言 $L(G) = \{ada^R \mid a \in (a,b)^*, a^R \text{ 为 } a \text{ 之逆}\}$ ，试构造产生此语言的上下文无关文法 G

解：根据题义，可知 a^R 为 a 之逆的含义就是句子中的符号 a、b 以 d 为中心呈左右对称出现；由于 $a \in (a,b)^*$ ，所以 a、b 的个数可以为零。所以可构造产生此语言的上下文无关文法 G[S] 为： $S \rightarrow aSa \mid bSb \mid d$

3-03 . 简述 DFA 与 NFA 有何区别？

答：DFA与 NFA的区别表现为两个方面：一是 NFA可以若干个开始状态，而 DFA仅只有一个开始状态。另一方面，DFA的映象 M 是从 $K \times$ 到 K ，而 NFA的映象 M 是从 $K \times$ 到 K 的子集，即映象 M 将产生一个状态集合（可能为空集），而不是单个状态。

3-04. 试给出非确定自动机的定义。

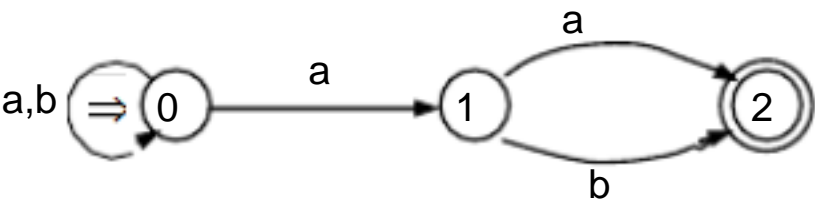
答：一个非确定的有穷自动机（NFA） M 是一个五元组： $M=(K, \Sigma, f, S, Z)$ 。

其中：

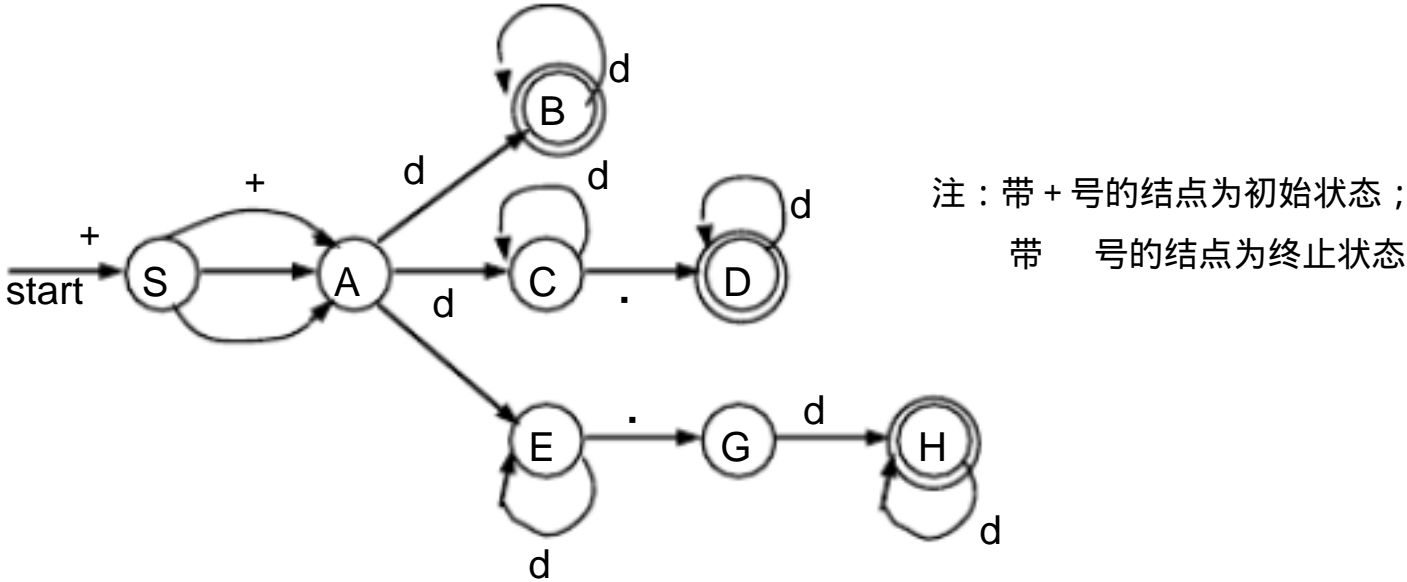
1. K 是一个有穷集，它的每个元素称为一个状态；
2. Σ 是一个有穷字母表，它的每个元素称为一个输入符号，所以也称 Σ 为输入符号表；
3. f 是状态转换函数，是在 $K \times \Sigma^+ K$ 的子集的映射，即， $f: K \times \Sigma^+ \rightarrow 2^K$ ；表明在某状态下对于某输入符号可能有多个后继状态；
4. $S \subseteq K$ 是一个非空初态集；
5. $Z \subseteq K$ 是一个终态集（可空）。

3-05. 为正规式 $(a|b)^+ a(a|b)^*$ 构造一个等价的确定的有限自动机。

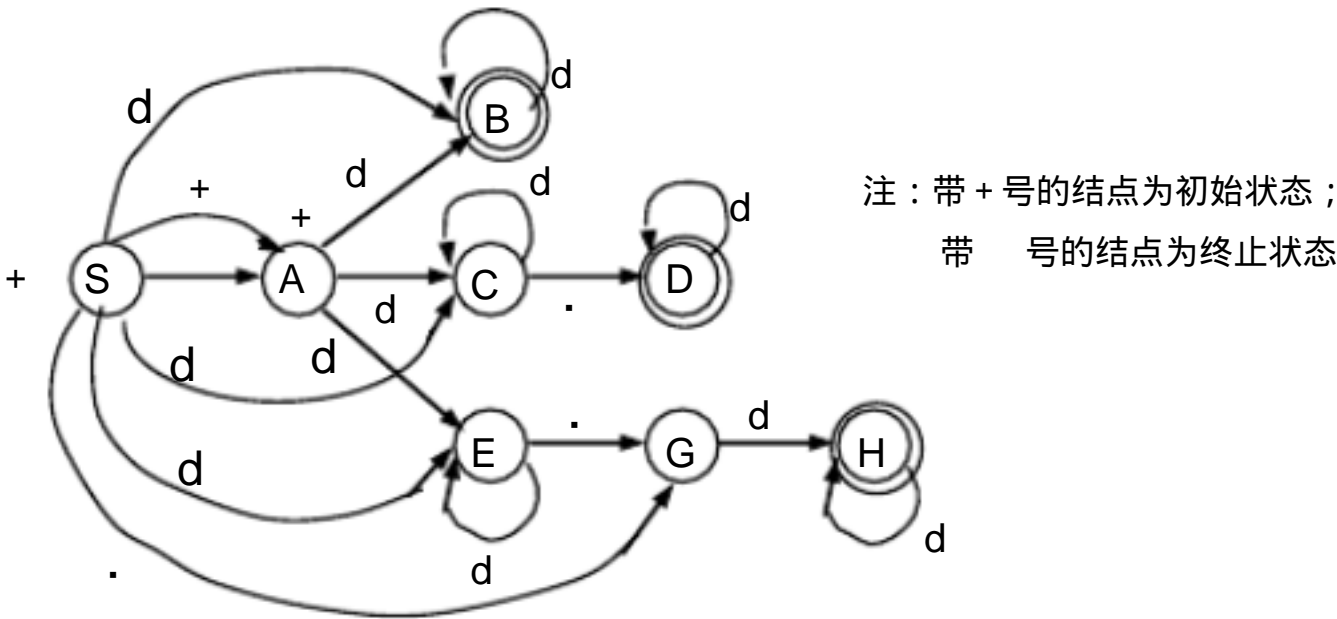
解答：



3-06. 给定下列自动机，将其转换为确定的自动机。

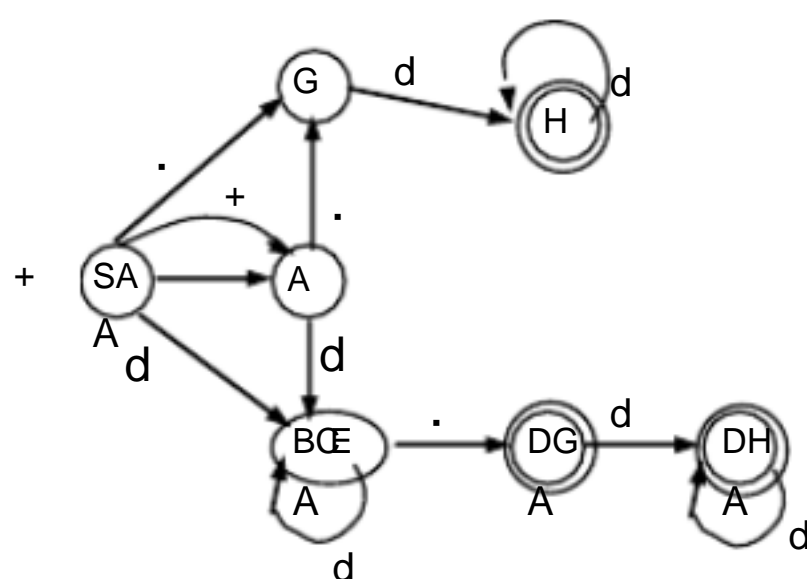


解答：(1) 消除 边，得到 NFA:



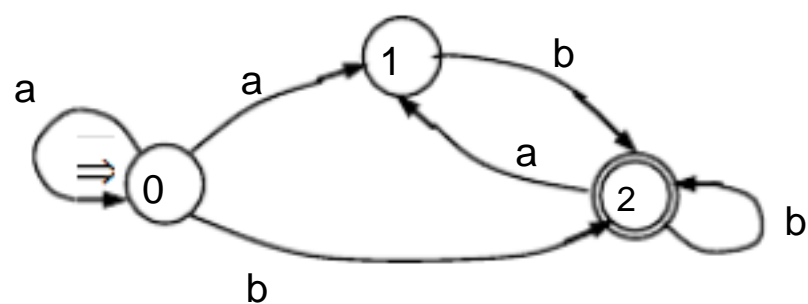
(2) 确定化，得到 DFA:

	+		d	.			+		d	.	
S	A	A	BCE	G			+[SA]	[A]	[A]	[BCE]	[G]
A			BCE				[A]			[BCE]	[G]
B			B				[BCE]			[BCE]	[DG]
C			C	D			[G]			[H]	
D			D				[DG]			[DH]	
E			E	G			[H]			[H]	
G			H				[DH]			[DH]	
H			H								



注：带 + 号的结点为初始状态；
带 号的结点为终止状态

3-07. 给定下列自动机：



其中：开始状态： 0
终止状态： 2

(1) 把此自动机转换为确定自动机 DFA

(2) 给出此 DFA的正则表达式。

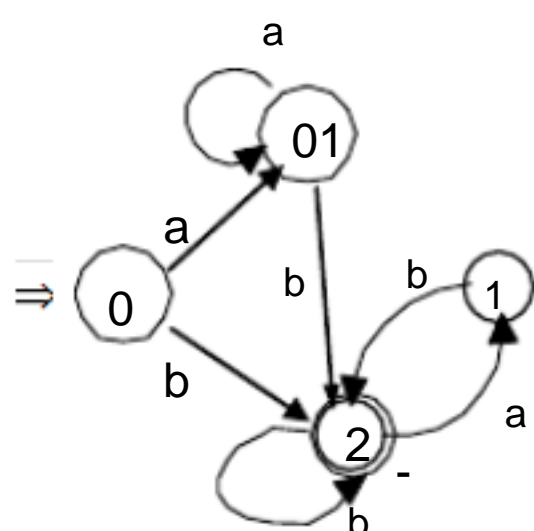
解答：(1)： 有状态矩阵如图：

	a	b
⇒ 0	0,1	2
1		2
- 2	1	2

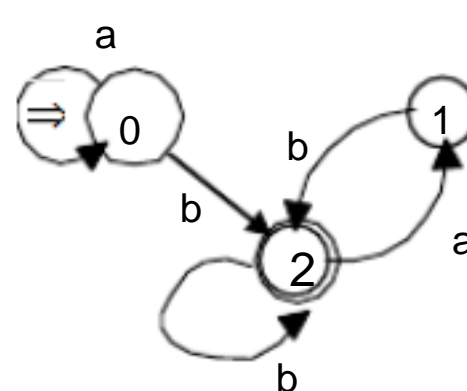
⇒

	a	b
⇒ 0	01	2
01	01	2
- 2	1	2
1		2

从而可得 DFA如图：



极小化后：



(2) 此 DFA 的正则表达式为： $(aa^*b|b)(b|ab)^*$ 或 $a^*b(b|ab)^*$ 。

4-13. 消除下列文法 $G[E]$ 的左递归。

$E \rightarrow E-T \mid T$
 $T \rightarrow T/F \mid F$
 $F \rightarrow (E) \mid i$

解答：

消除文法 $G[E]$ 的左递归后得到：

$E \rightarrow TE'$
 $E' \rightarrow -TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow /FT' \mid \epsilon$
 $F \rightarrow (E) \mid i$

4-14. 在 LL(1) 分析法中, LL 分别代表什么含义？

答: 第一个 L 代表从左到右的扫描, 第二个 L 代表每次进行最左推导。

4-15. 自顶向下分析思想是什么？

答：从开始符出发导出句型并一个符号一个符号地与给定终结字符串进行匹配。如果全部匹配成功, 则表示开始符号可推导出给定的终结字符串。因此判定给定终结字符串是正确句子。

4-16. 自顶向下的缺点是什么？

答：在推导过程中, 如果对文法不做限制。那么产生式的选择成为无根据的, 只好一一去试所有可能的产生式, 直至成功为止。这种方法的致命弱点是不断地回溯, 大大影响速度。

4-17. LL (1) 文法的定义是什么？

答：一个上下文无关文法是 LL(1) 文法的充分必要条件是每个非终结符 A 的两个不同产生式, $A \rightarrow \alpha_1 \mid \alpha_2$; 满足 $SELECT(A, \alpha_1) \cap SELECT(A, \alpha_2) = \emptyset$ 。其中, α_1, α_2 不能同时 \Rightarrow^* 。

4-18. 什么是文法的左递归？

答：一个文法含有下列形式的产生式之一时：

1) $A \rightarrow A \alpha, A \in VN, \alpha \in V^*$
 2) $A \rightarrow B \alpha, B \rightarrow A, A, B \in VN, \alpha \in V^*$

则称该文法是左递归的。

4-19. 递归下降法的主要思想是什么？

答：对每个非终结符按其产生式结构写出相应语法分析子程序。因为文法递归相应子程序也递归, 子程序的结构与产生式结构几乎一致。所以称此种方法称为递归子程序法或递归下降法。

5-19. 自底向上分析法的原理是什么？

答：在采用自左向右扫描, 自底向上分析的前提下, 该类分析方法是从输入符号串入手, 通过反复查找当前句型的句柄 (最左简单短语), 并使用文法的产生式把句柄归约成相应的非终极符来一步步地进行分析的。最终把输入串归约成文法的开始符号, 表明分析成功。

5-23. 给定文法 $G[Z]$ ：

1. $Z \rightarrow CS$
 2. $C \rightarrow \text{if } E \text{ then}$
 3. $S \rightarrow A = E$
 4. $E \rightarrow E \mid A$
 5. $E \rightarrow A$
 6. $A \rightarrow i$

其中： $Z, C, S, A, E \in V_N$;
 $\text{if, then, =, } \mid, i \in V_T$

- a) 构造此文法的 LR(0) 项目集规范族，并给出识别活前缀的 DFA
b) 构造其 SLR(1) 分析表。

解答：1. 首先拓广文法：在 G 中加入产生式 $0. Z \rightarrow Z$ ，然后得到新的文法 G ，再求 G 的识别全部活前缀的 DFA:

$I_0: Z \rightarrow \cdot Z$

$Z \rightarrow \cdot CS$

$C \rightarrow \cdot \text{if } E \text{ then}$

$I_1: Z \rightarrow Z \cdot$

$I_2: Z \rightarrow C \cdot S$

$S \rightarrow \cdot A = E$

$A \rightarrow \cdot i$

$I_3: C \rightarrow \text{if} \cdot E \text{ then}$

$E \rightarrow \cdot E A$

$E \rightarrow \cdot A$

$A \rightarrow \cdot i$

$I_4: Z \rightarrow CS \cdot$

$I_5: S \rightarrow A \cdot = E$

$I_6: A \rightarrow i \cdot$

$I_7: C \rightarrow \text{if } E \cdot \text{ then}$

$E \rightarrow E \cdot A$

$I_9: S \rightarrow A = \cdot E$

$E \rightarrow \cdot E A$

$E \rightarrow \cdot A$

$A \rightarrow \cdot i$

$I_{10}: C \rightarrow \text{if } E \text{ then} \cdot$

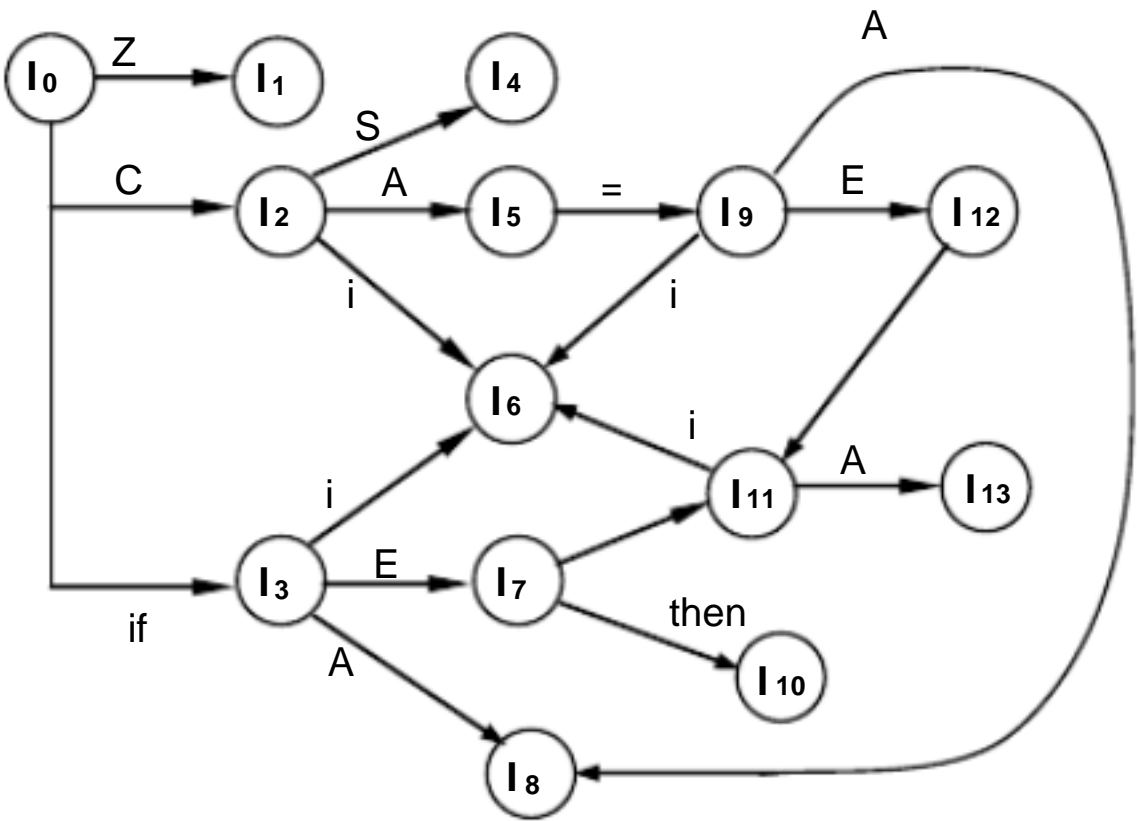
$I_{11}: E \rightarrow E \cdot A$

$A \rightarrow \cdot i$

$I_{12}: S \rightarrow A = E \cdot$

$E \rightarrow E \cdot A$

$I_{13}: E \rightarrow E A \cdot$



2. $\text{Follow}(Z) = \{\#\}$
 $\text{Follow}(C) = \{i\}$
 $\text{Follow}(S) = \{\#\}$
 $\text{Follow}(E) = \{\#, \text{ then}\}$
 $\text{Follow}(A) = \{=, \#, \text{ then}\}$
则可构造 SLR(1) 分析表为：

	ACTION						GOTO				
0	if	then	=		i	#	Z	C	S	E	A
0	S_3						1	2			
1						OK					
2					S_6				4		5
3					S_6					7	8
4						r_1					
5			S_9								
6		r_6	r_6	r_6		r_6					
7		S_{10}		S_{11}							
8		r_5		r_5		r_5					
9					S_6					12	8
10					r_2						

11					S_6						13
12				S_{11}		r_3					
13		r_4		r_4		r_4					

5-24. 设有文法 $G[S]$:

$S \rightarrow aA$
 $A \rightarrow Ab$
 $A \rightarrow b$

求识别该文法所有活前缀的 DFA

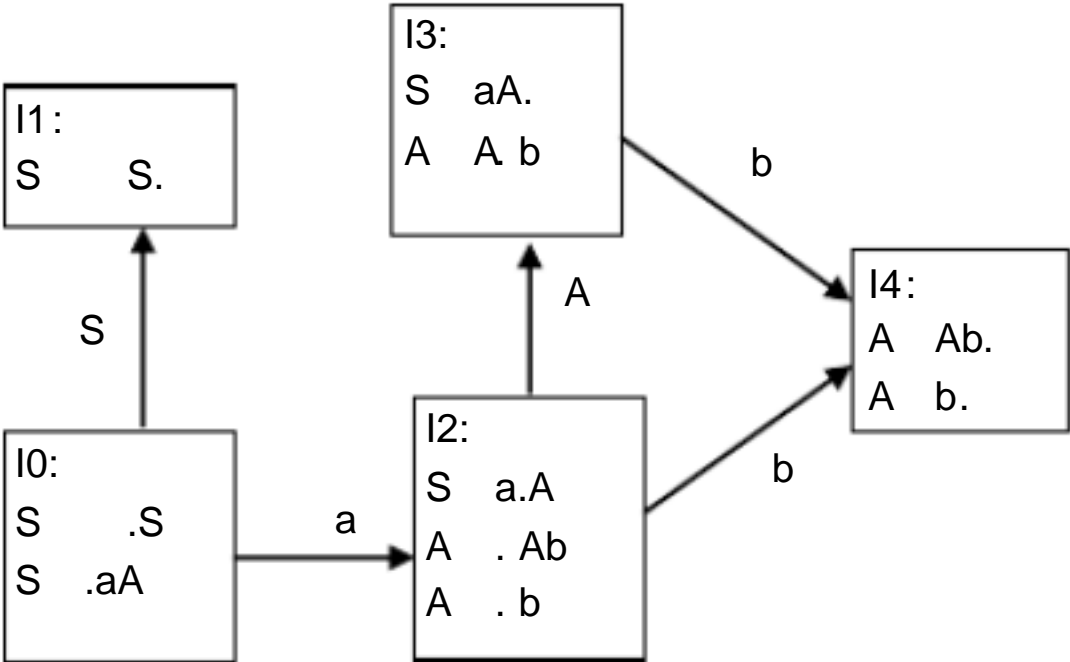
解答 :

(1). 首先拓广文法 :

在 G 中加入产生式 $0.S \rightarrow S$, 然后得到新的文法 G :

$0.S \rightarrow S$
 $1.S \rightarrow aA$
 $2.A \rightarrow Ab$
 $3.A \rightarrow b$

(2). 再求 G 的识别全部活前缀的 DFA:



6-07. 语法制导翻译方法的基本思想是什么 ?

答 : 在语法分析过程中, 每当使用一条产生式进行推导或归约时, 就执行该产生式所对应的语义动作进行属性计算, 完成对输入符号串的翻译。

6-08. 何谓“语法制导翻译” ?

答 : 在语法分析过程中, 随着分析的步步进展, 根据每个产生式所对应的语义子程序 (或语义规则描述的语义动作) 进行翻译的办法称作语法制导翻译。

6-09. 在一个属性文法中, 对应于每个产生式 $A \rightarrow a$ 都有一套与之相关联的语义规则, 每条规则的形式为 $b := f(c_1, c_2, \dots, c_k)$, 其中对于 b 的要求是什么?

答 : 语义规则中的左部属性变量 b 被规定为只能是下述两种变量 :

- 对应产生式左部符号的综合属性变量 ;
- 对应产生式右部符号的继承属性变量。

7-05. 常用的中间语言种类有哪几种 ?

答 : 有逆波兰式、三地址代码、抽象语法树和 DAG

8-06. 符号表的作用有哪些 ?

答 : 在编译程序中符号表用来存放语言程序中出现的有关标识符的属性信息, 这些信息集中反映了标识符的语义特征属性。起主要作用是 :

- 收集符号属性 ;
- 上下文语义的合法性检查的依据 ;

作为目标代码生成阶段地址分配的依据。

9-09 . 运行时存储器的划分是怎样的？

答：运行时存储器的划分如下图所示。

目标代码
静态数据
栈
堆

10-07. 简述优化的原则是什么？

答：编译程序提供的对代码优化必须遵循的原则是：

- (1) 等价原则。经过优化后不应改变程序运行的结果。
- (2) 有效原则。使优化后所产生的目标代码运行时间较短，占用的存储空间较小。
- (3) 合算原则。应尽可能以较低的代价取得较好的优化效果。

10-08 . 简述常用的优化技术有哪些？

答：编译程序中常用的优化技术有：

- (1) 删除公共子表示式；
- (2) 复写传播；
- (3) 删除无用代码；
- (4) 代码外提；
- (5) 强度削弱；
- (6) 删除归纳变量；
- (7) 合并常量。

10-09. 设有基本块：

- (1) $a := b - c$
- (2) $d := a + 4$
- (3) $e := a - b$
- (4) $f := a + 4$
- (5) $b := b + c$
- (6) $c := b - f$
- (7) $b := b - c$
- (8) $f := b + f$
- (9) $a := a - f$

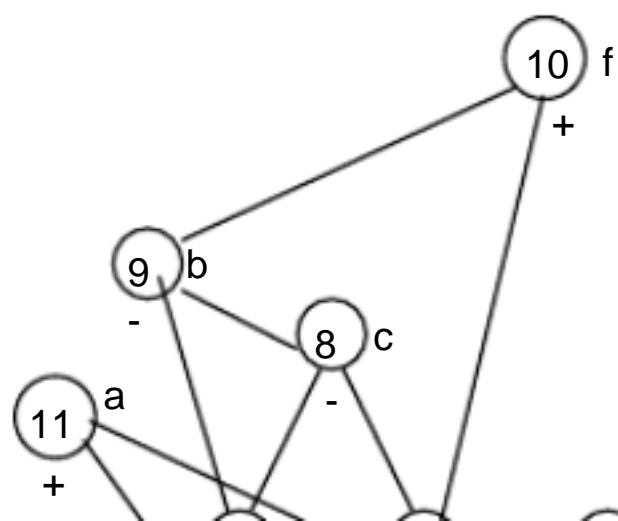
(1) 画出 DAG图；

(2) 假设基本块出口时只有 a, b 还被引用，请写出优化后的三地址代码序列。

解答：

(1) 给出 DAG图如右：

(2) 重写三地址代码如下：



```

a:=b-c
d:=a+4
f:=d
e:=a-b
b:=b+c
c:=c+d
b:=b-c
f:=b-d
a:=a+d

```

10-10. 何谓优化？按所涉及的程序范围可分为哪几级优化？

答：优化：对程序进行各种等价变换，使得从变换后的程序出发，能产生更有效的目标代码。

三种级别：局部优化、循环优化、全局优化。

10-11. 设有基本块

```

T1 := 2
T2 := 10/T1
T3 := S - R
T4 := S + R
A := T2 * T4
B := A
T5 := S + R
T6 := T3 * T5
B := T6

```

(1) 画出 DAG图；

(2) 假设基本块出口时只有 A, B 还被引用，请写出优化后的三地址代码序列。

解：(1)DAG：见右图

(2) 优化后的四元式

```

T3 := S - R
T4 := S + R
A := 5 * T4
B := T3 + T4

```

