

计算机学院、网安学院 2019—2020 学年第一学期

本科生编译系统原理期末考试试卷(A 卷)

专业：_____ 年级：_____ 学号：_____

姓名：_____ 成绩：_____

得分

一、 单项选择题（每空 2 分，共 24 分）

1. C++ 编译器在词法分析阶段 B，在语法分析阶段 D，在语义分析阶段 C。
A. 进行数据流分析 B. 过滤空格、制表符
C. 发现变量未定义就使用 D. 获得并记录变量类型信息
2. 华为今年开源了方舟编译器，安卓系统在运行 Java 程序时大部分代码还需逐条动态转换为机器代码再执行，而方舟则静态将完整 Java 程序转换为完整机器代码，直接执行无需动态转换，因此，其实现是一种 B 方式，传统安卓系统是一种 A 方式。
A. 解释器 B. 编译器
C. 多前端 D. 多后端
3. 关于正则表达式，以下说法错误的是 D。
A. 某些上下文无关文法生成的语言可用正则表达式描述
B. 正则表达式能描述的语言集合是上下文无关文法的真子集
C. 正则表达式转换 NFA 的算法用到了语法分析和语法制导翻译
D. SLR 不能分析的文法生成的语言必然不可能是正规集
4. n 个状态的 NFA 转换为 DFA，最多有 C 个状态。
A. n B. n^2
C. 2^n D. $n!$
5. 对下面 CFG，错误的说法是 C。
 $S \rightarrow aSa \mid aa$
A. 它接受所有偶数个 a 的串 B. 存在等价的正则表达式
C. 是 LL(1)文法 D. 是算符文法

6. 对 LR(0)项目 $A \rightarrow X \cdot YZ$, 错误的说法是__B__。
- A. • 对应活前缀末尾
 - B. • 对应句柄末尾
 - C. • 对应状态栈栈顶
 - D. 接下来 Y 的处理用一次移进不一定能完成
7. 如果一个表达式的语法树是一棵高度为 h 的满二叉树, 则转换为三地址码时采用临时变量重用的话, 最终需要使用__A__个临时变量。
- A. $h-1$
 - B. h
 - C. h^2
 - D. 2^{h-1}
8. 对于单词串 `id +)`, 表达式语法分析程序可给出错误信息__A__、__D__。
- A. 缺少运算符
 - B. 缺少运算符
 - C. 缺少标点
 - D. 未匹配右括号

得分

二、设计题（每题 6 分，共 24 分）

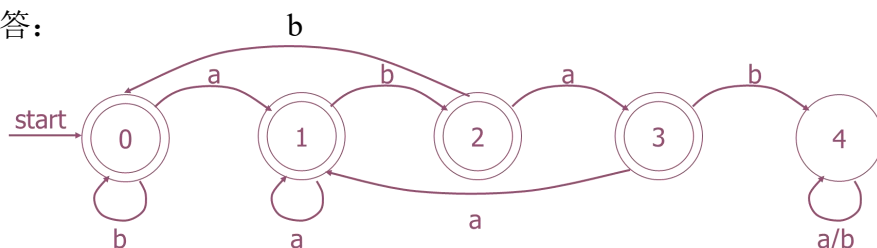
1. 描述所有首尾符号不同的 a 、 b 串的正则表达式。

答: $(a(a^*b)^+)|(b(b^*a)^+)$

或 $(a(a|b)^*b)|(b(a|b)^*a)$ 或其他等价正则表达式

2. 接受所有不包含 $abab$ 的 a 、 b 串的 DFA。

答:



3. 接受语言 $\{a^i b^j a^k \mid j=i+k, i \geq 0, k \geq 0\}$ 的上下文无关文法。

答:

$S \rightarrow AB$

$A \rightarrow aAb \mid \epsilon$

$B \rightarrow bBa \mid \epsilon$

4. 新 C++ 标准中引入了 `lambda` 表达式, 它采用捕获列表将所在作用域中的变量传递到自己的函数体中。其形式是 `[]` 包围的一个逗号分隔的列表, 列表中元素可以是: 空, 什么也不传递; `a`, 传值方式传递变量 `a`; `&b`, 传引用方式传递变量 `b`; `=`, 传值方式传递所有未提及变量; `&`, 传引用方式传递所有未提及变量。设计上下文无关文法接受全部捕获列表。

答:

$clist \rightarrow [] \mid [=] \mid [\&] \mid [=, rlist] \mid [\&, vlist] \mid [rvlist]$

$rlist \rightarrow rlist, \& id \mid \& id$

$vlist \rightarrow vlist, id \mid id$

$rvlist \rightarrow rvlist, \& id \mid rvlist, id \mid \& id \mid id$

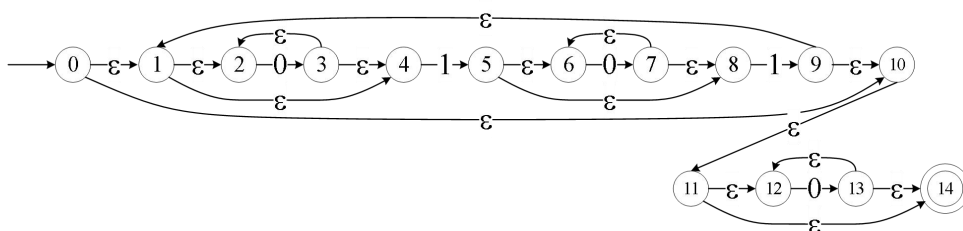
得分

三、（25 分）对下面的正则表达式。

$(0^*10^*1)^*0^*$

1. 用 Thompson 构造法将其转换为 NFA，识别 01001。（10 分）

答:



识别 01001: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 14$

2. 用子集构造法将得到的 NFA 转换为 DFA，画出最终的状态转换图，识别 01001。（10 分）

答:

$\epsilon_closure(\{0\}) = \{0, 1, 2, 4, 10, 11, 12, 14\} = A$

$\epsilon_closure(\delta(A, 0)) = \{2, 3, 4, 12, 13, 14\} = B$

$\epsilon_closure(\delta(A, 1)) = \{5, 6, 8\} = C$

$\epsilon_closure(\delta(B, 0)) = \{2, 3, 4, 12, 13, 14\} = B$

$\epsilon_closure(\delta(B, 1)) = \{5, 6, 8\} = C$

$\epsilon_closure(\delta(C, 0)) = \{6, 7, 8\} = D$

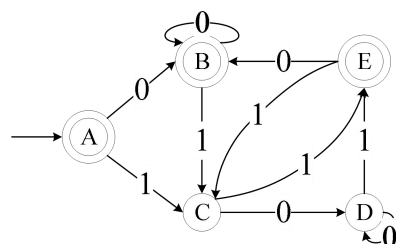
$\epsilon_closure(\delta(C, 1)) = \{1, 2, 4, 9, 10, 11, 12, 14\} = E$

$\epsilon_closure(\delta(D, 0)) = \{6, 7, 8\} = D$

$\epsilon_closure(\delta(D, 1)) = \{1, 2, 4, 9, 10, 11, 12, 14\} = E$

$\epsilon_closure(\delta(E, 0)) = B$

$\epsilon_closure(\delta(E, 1)) = C$



识别 01001: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow D \rightarrow E$

3. 将 DFA 最小化，画出最终的状态转换图。（5 分）

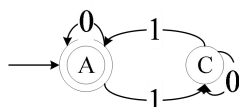
答：

初始划分 $\{A, B, E\}, \{C, D\}$

$\{A, B, E\} \rightarrow 0 \{B\}, \{A, B, E\} \rightarrow 1 \{B\}$

$\{C, D\} \rightarrow 0 \{D\}, \{C, D\} \rightarrow 1 \{E\}$

两个分组都不可再分，最终状态转换图为



得分

四、（17 分）对下面文法：

1. 指出其终结符集合、非终结符集合、开始符号（5 分）

2. 构造基于 LR(0)项目的识别活前缀的 NFA（8 分）

3. 用此 NFA 对句子 220 进行语法分析（4 分）。

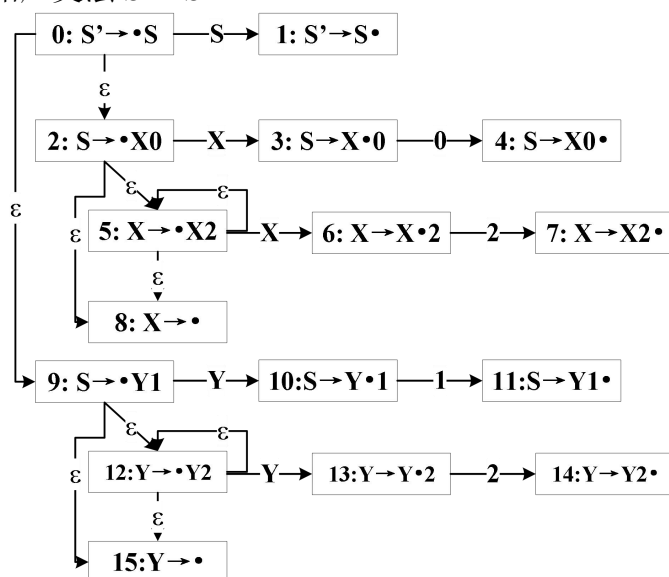
$S \rightarrow X0 \mid Y1$

$X \rightarrow X2 \mid \varepsilon$

$Y \rightarrow Y2 \mid \varepsilon$

1. 终结符集合 $\{0, 1, 2\}$ ，非终结符集合 $\{S, X, Y\}$ ，开始符号 S

2. 拓广文法 $S' \rightarrow S$



3. 分析 220

初态 $\{0, 2, 5, 8, 9, 12, 15\}$ ，8、15 可归约，15 归约错误路线，8 归约 $X \rightarrow \varepsilon$

$\rightarrow \{3, 6\}$ ，移进 2， $\rightarrow \{7\}$ ，归约 $X \rightarrow X2$

$\rightarrow \{3, 6\}$ ，移进 2， $\rightarrow \{7\}$ ，归约 $X \rightarrow X2$

$\rightarrow \{3, 6\}$ ，移进 0， $\rightarrow \{4\}$ ，归约 $S \rightarrow X0$

$\rightarrow \{1\}$ ，accept

得分

五、（10 分）将下面 C 语言程序片段翻译为三地址码程序，画出流图，进行优化。

```

m = 0;
for (i = 1; i < n; i = i + 1)
    if (a[i] > a[m]) m = i;

```

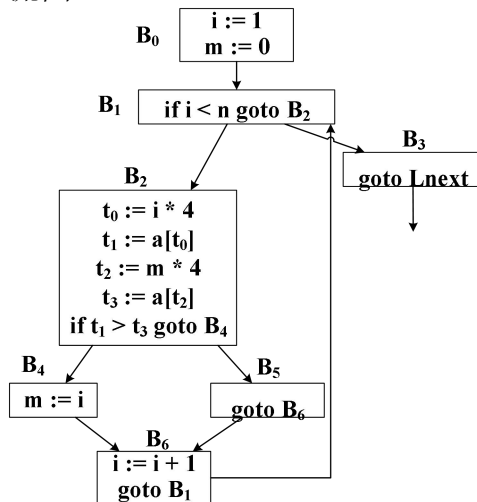
答：三地址码程序为

```

m := 0
i := 1
L0:
    if i < n goto L1
    goto Lnext
L1:
    t0 := i * 4
    t1 := a[t0]
    t2 := m * 4
    t3 := a[t2]
    if t1 > t3 goto L2
    goto L3
L2:
    m := i
L3:
    i := i + 1
    goto L0
Lnext:

```

流图：



优化：

