

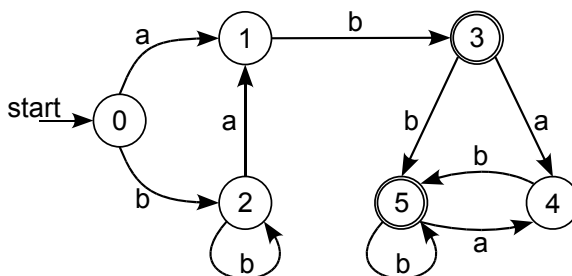
武汉大学计算机学院2006-2007学年第二学期 2004级《编译原理》考试试题

学号：_____ 姓名：_____ 成绩：_____

注意：请将答案全部写在答题纸上，并请写明题目的序号！

一、 (20分, 5+5+5+5)

- (1) 设有语言 $L = \{\alpha \mid \alpha \in \{0,1\}^+, \text{且}\alpha\text{以}0\text{开头, 并以}11\text{结尾}\}$, 试写出描述 L 的正规表达式 r , 使得 $L(r) = L$;
- (2) 请构造与正规表达式 $r = (0|1)^*(00|11)$ 等价的不确定有限自动机 N (请画出 N 的状态转换图), 使得 $L(N) = L(r)$;
- (3) 请直接构造与题(1)中 L 等价的确定的有限自动机 M , 使得 $L(M) = L$;
- (4) 请构造与下图所示 DFA 等价的最小化 DFA (请画出 DFA 的状态转换图)。



二、 (15分, 5+5+5)

- (1) 试设计上下文无关文法 $G(S)$, 使得 $L(S) = \{a^n b^{2n} \mid n \in \mathbb{N} \wedge n > 0\}$;
- (2) 试描述下述文法 $G(T)$ 所生成的语言:

$$T \rightarrow Tab \mid Tb \mid ab \mid b$$

- (3) 试写出文法 $G(T)$ 的一条语句“abbbab”一个最左推导和对应的语法树。

三、 设文法 $G(S)$ 定义如下: (15分, 5+5+5)

$$\begin{aligned}
 S &\rightarrow AB \mid CDa \\
 A &\rightarrow ab \\
 B &\rightarrow bc \mid \varepsilon \\
 C &\rightarrow cC \mid \varepsilon \\
 D &\rightarrow dD \mid \varepsilon
 \end{aligned}$$

- (1) 试对文法 $G(S)$ 的每个非终结符求 First 集合;

- (2) 试对文法 $G(S)$ 的每个非终结符求Follow集合；
 (3) 不用构造LL(1)分析表，说明该文法不是LL(1)文法。

四、设有集合运算文法 $G(S)$ 定义如下： (10分，5+5)

$$S \rightarrow S \cap S \mid S \cup S \mid S - S \mid -S \mid (S) \mid a$$

其中：-在一元运算时为集合的求补运算；在二元运算时为两个集合的差集。

- (1) 试举例说明文法 $G(S)$ 是二义文法；
 (2) 试设计一个与 $G(S)$ 等价的无二义文法，要求该文法对交运算 \cap 和并运算 \cup 均是**左结合的**，对差运算-是**右结合的**；并且该文法对运算优先级别为：一元求补运算-优先级最高；交运算 \cap 次之；并运算 \cup 再次之；差运算-优先级别最低；括号运算可以覆盖原优先级别。

五、设有集合运算文法 $G(S)$ 的拓广文法 $G(S')$ 定义如下： (15分，5+5+5)

$$\begin{aligned} S' &\rightarrow S & (0) \\ S &\rightarrow S \cap S & (1) \\ S &\rightarrow S \cup S & (2) \\ S &\rightarrow S - S & (3) \\ S &\rightarrow -S & (4) \\ S &\rightarrow (S) & (5) \\ S &\rightarrow a & (6) \end{aligned}$$

其中：First(S) = { -, (, a }; Follow(S) = { \$, \cap , \cup , -,) }

- (1) 下表给出了文法 $G(S)$ 的LR(0)项目集规范簇(规范项目集)，其中缺状态 I_5 和 I_{10} 的项目集，设：Dtrans(I_{13}, \cup) = I_5 , Dtrans(I_9, S) = I_{10} ，试求状态 I_5 和 I_{10} 对应的LR(0)项目集规范簇：

$$\begin{aligned} I_0 &= \{ S' \rightarrow \bullet S, S \rightarrow \bullet S \cap S, S \rightarrow \bullet S \cup S, S \rightarrow \bullet S - S, \\ &\quad S \rightarrow \bullet - S, S \rightarrow \bullet (S), S \rightarrow \bullet a \} \\ I_1 &= \{ S \rightarrow - \bullet S, S \rightarrow \bullet S \cap S, S \rightarrow \bullet S \cup S, S \rightarrow \bullet S - S, \\ &\quad S \rightarrow \bullet - S, S \rightarrow \bullet (S), S \rightarrow \bullet a \} \\ I_2 &= \{ S \rightarrow - S \bullet, S \rightarrow S \bullet \cap S, S \rightarrow S \bullet \cup S, S \rightarrow S \bullet - S \} \\ I_3 &= \{ S \rightarrow S \cap \bullet S, S \rightarrow \bullet S \cap S, S \rightarrow \bullet S \cup S, S \rightarrow \bullet S - S, \\ &\quad S \rightarrow \bullet - S, S \rightarrow \bullet (S), S \rightarrow \bullet a \} \\ I_4 &= \{ S \rightarrow S \cap S \bullet, S \rightarrow S \bullet \cap S, S \rightarrow S \bullet \cup S, S \rightarrow S \bullet - S \} \\ I_6 &= \{ S \rightarrow S \cup \bullet S, S \rightarrow S \bullet \cap S, S \rightarrow S \bullet \cup S, S \rightarrow S \bullet - S \} \\ I_7 &= \{ S \rightarrow S - \bullet S, S \rightarrow \bullet S \cap S, S \rightarrow \bullet S \cup S, S \rightarrow \bullet S - S, \\ &\quad S \rightarrow \bullet - S, S \rightarrow \bullet (S), S \rightarrow \bullet a \} \\ I_8 &= \{ S \rightarrow S - S \bullet, S \rightarrow S \bullet \cap S, S \rightarrow S \bullet \cup S, S \rightarrow S \bullet - S \} \\ I_9 &= \{ S \rightarrow (\bullet S), S \rightarrow \bullet S \cap S, S \rightarrow \bullet S \cup S, S \rightarrow \bullet S - S, \\ &\quad S \rightarrow \bullet - S, S \rightarrow \bullet (S), S \rightarrow \bullet a \} \\ I_{11} &= \{ S \rightarrow (S) \bullet \} \\ I_{12} &= \{ S \rightarrow a \bullet \} \\ I_{13} &= \{ S' \rightarrow S \bullet, S \rightarrow S \bullet \cap S, S \rightarrow S \bullet \cup S, S \rightarrow S \bullet - S \} \end{aligned}$$

- (2) 文法 $G(S)$ 的SLR(1)分析表如下(见下页)所示, 请在该表的空白之处填入正确的移进归约操作和goto状态, 使得该分析表能够按**题四(2)**所规定命题公式的优先级别和结合次序正确地分析集合运算:
(注意: 本小题请在答题纸上画表, 仅列出状态0和1的goto项, 状态2、4、6和8的action项即可!)

state	action								goto
	\cap	\cup	$-$	$($	$)$	a	$\$$	S	
0	/	/	s1	s9	/	s12	/	?	
1	/	/	s1	s9	/	s12	/	?	
2	?	?	?	?	?	?	?	/	
3	/	/	s1	s9	/	s12	/	4	
4	?	?	?	?	?	?	?	/	
5	/	/	s1	s9	/	s12	/	6	
6	?	?	?	?	?	?	?	/	
7	/	/	s1	s9	/	s12	/	8	
8	?	?	?	?	?	?	?	/	
9	/	/	s1	s9	/	s12	/	10	
10	s3	s5	s7	/	s11	/	/	/	
11	r5	r5	r5	/	r5	/	r5	/	
12	r6	r6	r6	/	r6	/	r6	/	
13	s3	s5	s7	/	/	/	acc	/	

- (3) 试写出输入串“ $-a \cup a - a$ ”的分析过程。

- 六、设程序中有判断元素是否属于一个集合运算后所得到集合的逻辑表达式, 现需用短路法设计将该表达式翻译为三地址码的语法制导定义。设该如下:

$$E \rightarrow id \in S$$

$$S \rightarrow S \cap S \mid S \cup S \mid S - S \mid -S \mid (S) \mid a$$

设 E 具有来自上下文的继承属性 $E.true$ 和 $E.false$, 该属性对应于该逻辑表达式为真和为假时转移到相关三地址码的标号; 设 E 和 S 都具有code属性, 它对应于相应的三地址代码链; S 还具有来自 \in 运算第一分量 id 对应元素名(name)继承属性 $S.element$, S 也同时具有 $S.true$ 和 $S.false$ 属性。设 $gen()$ 为生成三地址码函数, \parallel 为三地址码链接运算, 设 $newlabel()$ 生成一个新的标号, 则该文法的语法制导定义可定于如下:

语法	语义规则
$E \rightarrow id \in S$	$S.true := E.true; S.false := E.false;$ $S.element := id.name; E.code := S.code;$
$S \rightarrow a$	$S.code := gen("if", S.element, "in", a.name, "goto", S.true) \parallel$ $gen("goto", S.false);$
$S \rightarrow S_1 \cap S_2$	$S_1.true := newlabel(); S_1.false := S.false;$ $S_2.true := S.true; S_2.false := S.false;$ $S.code := S_1.code \parallel gen(S_1.true, ":") \parallel S_2.code;$ $S_1.element := S.element; S_2.element := S.element;$

(10分, 5+5)

- (1) 试写出完成剩余产生式 $S \rightarrow S \cup S \mid S - S \mid -S \mid (S)$ 对应的语法制导定义;
- (2) 设逻辑表达式“ $x \in -A \cup B - C$ ”的true和false属性分别是标号“Ltrue”和“Lfalse”, 试利用上述语法制导定义计算该表达式的code属性。

七、下面是一段示意性源程序, 请给出下述程序分别在换名、传值、传地址和传结果四种参数传递方式下的输出结果。(8分)

调用者有如下片段:

```
.....
B: array [1..4] of integer;
i: integer;
.....
for i := 1 to 4 do
    B[i] := i;
i := 1;
P(B[i]);
writeln(B[1], B[2]);
.....
```

被调用者定义为:

```
procedure P(v: integer);
begin
    v := B[i] + 2;
    i := 2;
    v := B[i] + 2;
    writeln(B[i], B[1])
end;
```

八、设有如两个C语言程序:

程序1

```
#include <stdio.h>
#include <string.h>
int main() {
    char t[] = "1234567";
    char s[] = "abc";
    strcpy(s, t);
    printf("s=%s; t=%s\n", s, t);
    return 0;
}
```

程序2

```
#include <stdio.h>
#include <string.h>
int main() {
    char s[] = "abc";
    char t[] = "1234567";
    strcpy(s, t);
    printf("s=%s; t=%s\n", s, t);
    return 0;
}
```

在Intel X86/Linux操作系统下程序1用gcc编译运行, 输出

s=1234567; t=567

后正常退出; 而将程序1中s和t的申明次序颠倒所得到的程序2后用gcc编译运行, 输出

s=1234567; t=1234567
segmentation fault

试简要分析程序1中t发生变化和程序2不能正常退出的原因。(7分)