

# 广东工业大学考试试卷 ( A )

课程名称: 编译原理 试卷满分 100 分

考试时间: 2018 年 6 月 26 日 (第 17 周 星期二)

考试形式: 闭卷 (开闭卷)

题号	一	二	三	四	五	六	七	八	九	总分
评卷得分										
评卷签名										
复核得分										
复核签名										

一 (6 分) 构造一个 2 型文法  $G$ ; 使其语言为

$$L(G) = \{ a^n b^n \mid n \geq 0 \}$$

二 (14 分)

已知文法  $G[E]: E \rightarrow T \mid E+T$

$T \rightarrow F \mid T * F$

$F \rightarrow (E) \mid i$

- (1) 写出句型  $(E+T)*i$  的最右推导;
- (2) 画出句型  $(E+T)*i$  的语法树;
- (3) 写出句型  $(E+T)*i$  的全部短语、简单短语和句柄。

三 (14 分) 已知文法  $G$ :

$S \rightarrow iESB \mid a$

$B \rightarrow eS \mid \varepsilon$

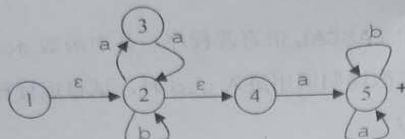
$E \rightarrow b$

- (1) 计算每个产生式的 Select 集
- (2) 判断它是否是 LL(1) 文法
- (3) 填写它的预测分析表。

	i	e	a	b	#
S					
B					
E					

四 (14 分)

已知如下图所示的 NFA: 其中, “-” 表示开始状态; “+” 表示终止状态:



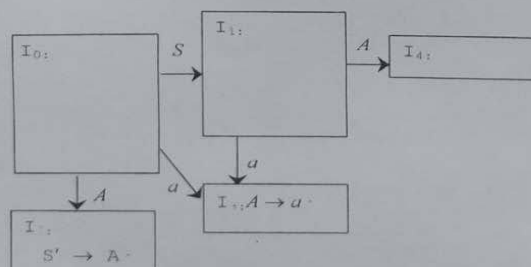
将 NFA 确定化为 DFA, 写出下表中 A、B、C、D 所对应的状态集, 并完成表格内容。I =  $\epsilon$ -CLOSURE (原状态)。

I	Move (I, a)	Ia	Move (I, b)	Ib
A {1, 2, 4}				
B { }				
C { }				
D { }				
E { }				

五 (14 分) 已知拓广文法  $G[S']$ :

$S' \rightarrow S$        $S \rightarrow SA \mid A$        $A \rightarrow a$

(1) 试构造以 LR(0) 项目集为状态的识别活前缀的 DFA, 写出下图 DFA 中  $I$ 、 $I_1$ 、 $I_4$  所对应的项目:



(2) 试判断文法是否是 LR(0) 文法, 并说明理由。

(3) 试判断文法是否是 SLR(1) 文法, 并说明理由

六 (8 分) 将布尔表达式  $a > b$  and  $c < d$  or  $e > f$  翻译成三地址码序列 (从 100 开始编号), 并将去往真假出口的三地址码分别拉成真链、假链, 同时指出真链、假链的链首 (三地址码编号)。

七 (10 分) 下面是一个 PASCAL 语言源程序, 其中函数  $\text{gcd}(m, n)$  的功能是返回  $m$  和  $n$  的最大公约数, 当第二次递归调用进入  $\text{gcd}$  时, 试将运行栈的内容补充完整。

```
PROGRAM gmn( );
```

```
VAR m, n, g: integer;
```

```
FUNCTION gcd(m, n: integer): integer;
```

```
BEGIN
```

```
IF n=0 THEN
```

```
  g:=m
```

```
ELSE
```

```
  g:=gcd(n, m MOD n)
```

```
END;
```

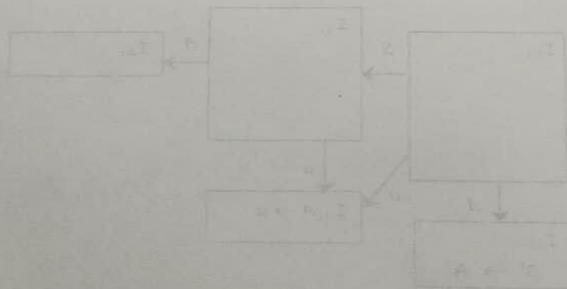
```
BEGIN
```

```
  m:=24;
```

```
  n:=16;
```

```
  g:=gcd(m, n)
```

```
END
```



(A) 卷

17		参数 n 值
16		参数 m 值
15		参数个数
14	RA	
13		DL
12		SL
11		参数 n 值
10		参数 m 值
9		参数个数
8	RA	
7		DL
6		SL
5	g	
4	n	
3	m	
2	RA	
1	0	DL
0	0	SL

八 (10 分) 请把以下程序划分为基本块并作出其程序流图。

(1) i:=m

(2) j:=n

(3) a:=u1

(4) L1:i:=i+1

(5) j:=j-1

(6) if i>j goto L2

(7) a:=u2

(8) L2:i:=u3

(9) goto L1

九 (10 分) 请在空缺处填空, 完成 PL/O 编译器的 WHILE 循环语句的编译算法:  
 <循环语句> ::= WHILE <条件> DO <语句>

switch (SYM) {

**case** WHILESYM:

\_\_\_\_\_ ① \_\_\_\_\_; GetSym();

CONDITION(SymSetAdd(DOSYM,FSYS),LEV, TX);

\_\_\_\_\_ ② \_\_\_\_\_; GEN(JPC, 0, 0);

**if** (\_\_\_\_\_ ③ \_\_\_\_\_) GetSym();

**else** Error(18);

STATEMENT(FSYS,LEV, TX);

GEN(\_\_\_\_\_ ④ \_\_\_\_\_);

\_\_\_\_\_ ⑤ \_\_\_\_\_;

**break**;

}

}