

北京航空航天大学

2019—2020 学年 第一学期期末

《编译技术》

B 卷

班 级 _____ 学 号 _____

姓 名 _____ 成 绩 _____

2020 年 5 月 25 日

班号_____ 学号_____ 姓名_____ 成绩_____

《 编译技术 》期末考试卷

注意事项:

1. 试卷共 6 页 (不含封面和目录), 请仔细检查。
2. 在监考老师统计完试卷后, 再离开考场;

题目:

- 一、简答题..... (20 分)
- 二、正则文法与自动机..... (20 分)
- 三、算符优先分析法..... (15 分)
- 四、SLR 分析法..... (15 分)
- 五、符号表构造与运行时存储分析..... (15 分)
- 六、代码优化..... (15 分)

题号	得分	教师签字
1		
2		
3		
4		
5		
6		
总分		

一. 简答题: (20 分)

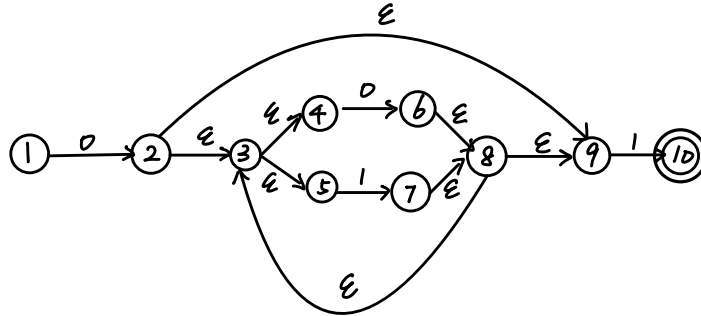
1. 简述什么是错误的局部化处理, 主要作用是什么?
2. 分别简述语法分析的任务和语义分析的任务。
3. 分别简述静态存储分配和动态存储, 二者的使用场景有什么不同。
4. 分别说明什么是局部优化、全局优化和循环优化。
5. 说明什么是交叉编译, 什么时候需要使用交叉编译?

二. 在输入字母表 $\Sigma=\{0, 1\}$ 上定义语言 $L = \{x|x \in \Sigma^*, \text{ 且 } x \text{ 以 } 0 \text{ 开头, 以 } 1 \text{ 结尾}\}$ 。(20 分)

1. 给出识别该语言 L 的正则表达式。

$0(0|1)^*1$

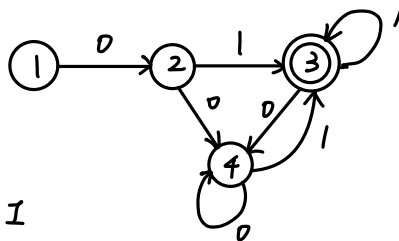
2. 根据正则表达式构造 NFA。



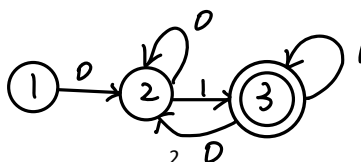
3. 设计识别该语言的极小化 DFA。(给出求解过程：①对应的)

DFA 求解:

	I	I_1	I_0
1	$\{1\}$	\varnothing	$\{2, 3, 4, 5, 9\}$
2	$\{2, 3, 4, 5, 9\}$	$\{7, 8, 9, 3, 5, 4, 10\}$	$\{6, 8, 9, 3, 4, 5\}$
3	$\{7, 8, 9, 3, 5, 4, 10\}$	$\{7, 8, 9, 3, 5, 4, 10\}$	$\{6, 8, 9, 3, 4, 5\}$
4	$\{6, 8, 9, 3, 4, 5\}$	$\{7, 8, 9, 3, 5, 4, 10\}$	$\{6, 8, 9, 3, 4, 5\}$



	0	1	
1	2	\varnothing	} III
2	4	3	
4	4	3	
3	4	3	} II.



三. 有如下文法 $G[S]$: (15 分)

$$S \rightarrow V \quad \#S\#$$

$$V \rightarrow T \mid ViT$$

$$T \rightarrow F \mid T+F$$

$$F \rightarrow bV^* \mid a$$

1. 判断该文法是否是算符文法, 并说明理由。

是, 不存在 $\dots VW\dots$

2. 求每个非终结符的 FIRSTVT 和 LASTVT 集合。

$$\text{FIRSTVT}(S) = \{i, +, b, a\}$$

$$\text{FIRSTVT}(V) = \{i, +, b, a\}$$

$$\text{FIRSTVT}(F) = \{b, a\}$$

$$\text{FIRSTVT}(T) = \{+, b, a\}$$

$$\text{LASTVT}(S) = \{*, a, +, i\}$$

$$\text{LASTVT}(V) = \{*, a, +, i\}$$

$$\text{LASTVT}(F) = \{*, a\}$$

$$\text{LASTVT}(T) = \{*, a, +\}$$

3. 构造算法优先关系矩阵。

	i	+	b	*	a	#
i	>	<	<	>	<	>
+	>	>	<	>	<	>
b	<	<	<	=	<	
*	>	>		>		>
a	>	>		>		>
#	<	<	<		<	

4. 判断该文法是否为算符优先文法。

是

四. 有如下文法 $G[S]$: (15 分)

$S \rightarrow SaBc \mid aBc$

$B \rightarrow bAc$

$A \rightarrow a \mid bS$

$FIRST(S) = \{a\}$

$FIRST(B) = \{b\}$

$FIRST(A) = \{a, b\}$

$FOLLOW(S) = \{\#, a, c\}$

$FOLLOW(B) = \{c\}$

$FOLLOW(A) = \{c\}$

1. SLR 分析表 (Action 表和 GOTO 表)。

	a	b	c	#	S	A	B
0	S_2				1		
1	S_3			acc			
2		S_5					4
3		S_5					6
4			S_7				
5	S_9	S_{10}				8	
6			S_{12}				
7	r_3		r_3	r_3			
8		S_{14}					
9			r_5				
10	S_2				11		
11	S_3		r_6				
12	r_2		r_2	r_2			
13			r_4				

2. 识别句型 $SabAcc$ 活前缀 $SabAc$ 的有效项目集。

$SabAc : B \rightarrow bAc.$

五. 有如下程序段: (15 分)

```

program paser;
  var filename:string;
      curchar:char;
  procedure getsym;
    var symsize:integer;
  procedure getchar;
    var errorinfo:string;
    begin
      ... ----- (1)
    end;
  begin
    ...
    call getchar;
    ...
  end;
  procedure lexer;
    var linelength:integer;
  begin
    ...
    call getsym;----- (2)
    ...
  end;
begin
  ...
  call lexer;
  ...
end.

```

errorinfo
 prev abp: abp3
 ret addr
 abp3 (DISPLAY)
 abp1 (DISPLAY)
 symsize
 prev abp: abp2
 ret addr
 getsym: abp3 → abp1 (DISPLAY)
 linelength
 prev abp: abp1
 ret addr
 lexer = abp2 → abp1 (DISPLAY)
 curchar
 paser: abp1 → filename

1. 请分别画出当编译到位置 (1) 和 (2) 时的符号表。

变量名	类型	维数 / 层号
filename	string	1
curchar	char	1
getsym	procedure	1
symsize	Integer	2
getchar	procedure	2
errorinfo	string	2

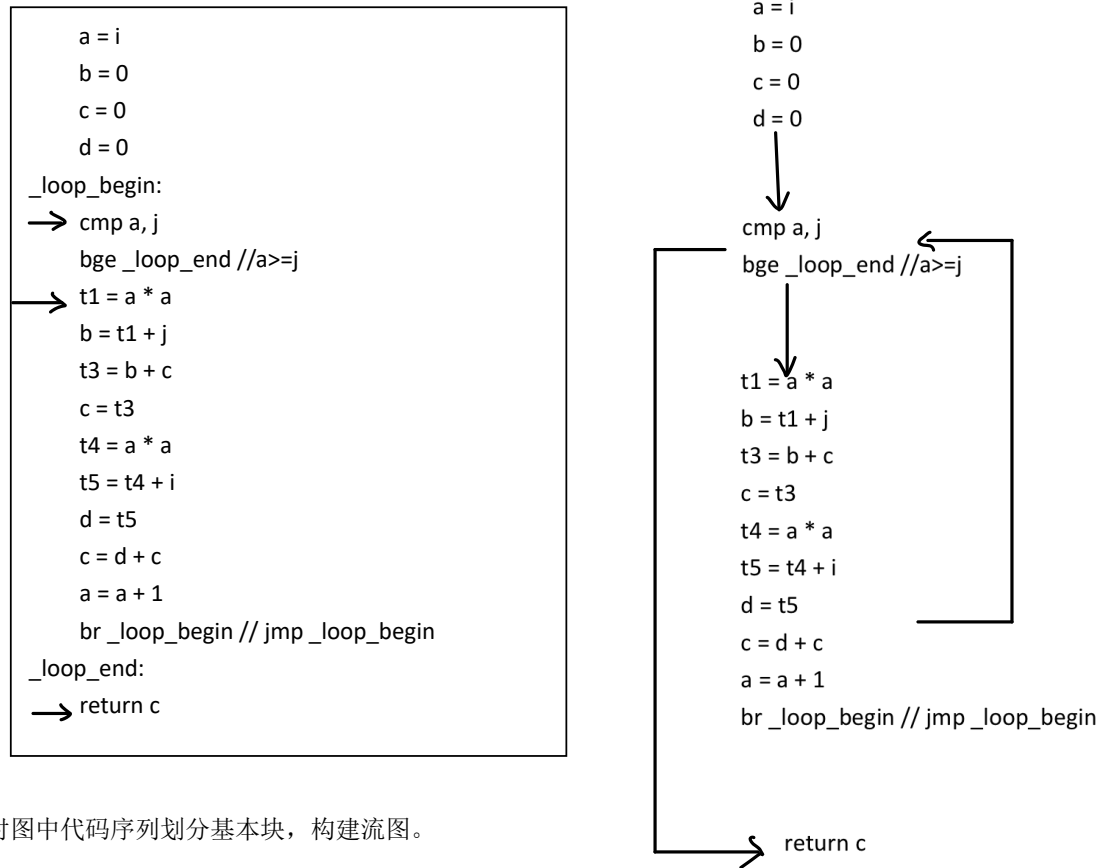
变量名	类型	维数
filename	string	1
curchar	char	1
getsym	procedure	1

lexer	procedure	1
line length	Integer	3

2. 运行到位置 (1) 时的运行栈。

	errorinfo	
	prev abp: abp3	
	ret addr	
	abp3 (DISPLAY)	
	abp1 (DISPLAY)	getchar: abp4
	symsize	
	prev abp: abp2	
	ret addr	
	abp1 (DISPLAY)	getsym: abp3
	linelength	
	prev abp: abp1	
	ret addr	
	abp1 (DISPLAY)	lexer = abp2
	curchar	
	filename	parser: abp1

六. 有如下中间代码序列, 其中{a,b,c,d}为局部变量, {i,j}为形参, {t1,t2,t3,t4,t5,t6}为临时变量 (15 分):



1. 对图中代码序列划分基本块，构建流图。

- 对循环体所在基本块利用 DAG 图做“局部公共子表达式删除”优化，并根据启发式算法给出优化后的中间代码序列。

