



A 卷

2021-2022 学年第 1 学期

(2021 秋季)

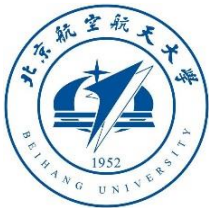
《编译原理与技术》

期末考试卷

班级_____学号_____

姓名_____成绩_____

2022 年 1 月 12 日



《编译原理与技术》

期末考试卷

注意事项：1. 所有答案请直接写在题目中，另附纸无效。

2. 交卷时请以班为单位交卷。

题号	一	二	三	四							总分
				1	2	3	4	5	6	7	
成绩											
阅卷人 签字											
任课教 师签字											

- 题目：
- 一、填空题..... (11 分)
- 二、判断题..... (7 分)
- 三、单选题..... (8 分)
- 四、综合题
1. (15 分)
2. (10 分)
3. (10 分)
4. (14 分)
5. (10 分)
6. (5 分)
7. (10 分)

一、填空题（每空 1 分，共 11 分）

1. 如果一个文法所定义的句子中有某个句子，它存在两棵不同的语法树，则该文法是 二义性文法
2. 3 型文法又被称为 正则 文法，可被 有限自动机 接受。
3. 由翻译文法确定的语言中的符号串称为活动序列。活动序列由 动作符号 和 终结符 组成。
4. 分析工作要部分地或全部地退回去重做，这种情况被称为 回溯。
5. 循环展开 是一种将循环体内代码依序拓展成顺序执行指令的优化方法。
6. 有文法 $G[S]$:

$S \rightarrow (L) \mid aS \mid a$

$L \rightarrow L, S \mid S$

句型 $(S, (a))$ 的简单短语是 S 和 a ，句柄是 S 。

7. $(A+B)*(C+D)-E$ 的后缀表示是 $AB+CD+*E-$ 。

8. 观察以下函数：

```
int get(int a[][5][10], int i, int j, int k){
    return a[i][j][k];
}
```

则每次调用时，访问到的 a 中的元素的首地址与 $a[0][0][0]$ 的首地址相差 $50*i + 10*j + k$ 个 int 类型的距离（用含 i, j, k 的表达式表示）。

二、判断题（每题 1 分，共 7 分）

1. 语言和文法的关系是多对一。 ✓
2. 文法 G 所描述的语言是文法 G 的 终结符号 集中的 所有符号 组成的符号串集合。 ✗
3. 词法分析程序可以编成一个子程序，由语法分析程序调用！ ✓
4. 递归下降分析法中，不允许任意一个非终结符的规则是直接左递归的。 ✓
5. 任一句型的最左素短语称为该句型的句柄。 ✗
6. 属性翻译文法是上下文无关的文法。 ✓
7. 根据程序在 运行时 发现的错误，就能够找出错误在源程序中的 确切位置。 ✗

$$P(3) = 1$$

$$P(2) = 10$$

$$P(1) = 5 \cdot P(2) = 50$$

三、单选题（每题 1 分，共 8 分）

1. 下面哪个过程不属于编译过程的五个基本阶段？

- ① 词法分析 ② 语法分析 ③ 代码优化 ④ 出错处理

2. 文法规则 $E ::= E + T \mid T$ 的 EBNF 表示为：

- ① $E ::= \{+T \mid T\}^*$ ② $E ::= T \{+T\}$

- ③ $E ::= \{T+T\}$ ④ $E ::= \{+T\}T$

3. 正则文法 $G(S)$ 如下。

$S \rightarrow A$

$A \rightarrow Ba \mid Bb$

$B \rightarrow Ca$

$C \rightarrow a \mid b \mid Ca \mid Cb$

则对应的正则表达式可能是：

- ① $(a|b)^*a(a|b)$

- ② $(a|b)^*a(a|b)(a|b)$

- ③ $(a|b)(a|b)a(a|b)^*$

- ④ $(a|b)^*(a|b)a(a|b)$

4. 下列属于自顶向下分析法的是：

- ① 递归子程序法

- ② 算符优先分析法

- ③ LR(0)分析法

- ④ LALR(1)分析法

5. 下面关于基本块的描述，不正确的是：

- ① 控制流只能从一个基本块的开始进入

- ② 控制流只能从一个基本块的末尾离开

- ③ 控制流从一个基本块末尾离开后，只能进入固定的下一个块

- ④ 基本块是一个连续的语句序列

6. 下面的代码片段有几个基本块？

1 a = 123

2 b = 56

3 c = 0

4 if (a < 100) goto 9

5 c = 4

6 if (c == 2 * 2) goto 9

7 c = a + b

8 goto 6

9 return c

Ⓐ 2

Ⓑ 3

Ⓒ 4

Ⓓ 5

7. 若一个优化将 $x = x * 32$; 优化成 $x = x << 5$; 则该优化是：

Ⓐ 强度削弱

Ⓑ 常数合并

Ⓒ 常数传播

Ⓓ 代码外提

8. 下面关于优化的说法，错误的是：

Ⓐ 局部优化指在基本块内进行的优化

Ⓑ 数据流分析属于局部优化

Ⓒ 消除公共子表达式属于基本块内的优化

Ⓓ 循环展开和代码外提都可用于循环优化

四、综合题（共 74 分）

1. (15 分) 已知正则表达式 $(01)^*1^*0(0|1)$

(1) 表达式中括号 $()$ ，星号 $*$ ，分隔符 $|$ 三个符号各有什么作用？

(2) 构造与之等价的右线性文法。

(3) 构造与之等价的 NFA。

(4) 将构造出的 NFA 转化为 DFA 并将其最小化。

11) (ab)* 整体

a^* 1 次 0-n 次

$a|b$ a 或 b 其中之一出现一次

第 3 页 共 12 页

12) $G[S]: S \rightarrow ABC$

$A \rightarrow \varepsilon \mid 0 \mid A$
 $B \rightarrow \varepsilon \mid 1 \mid B$

$A \rightarrow 0 \mid A$
 $A \rightarrow \varepsilon$
 $B \rightarrow 1 \mid B$
 $B \rightarrow \varepsilon$
 $C \rightarrow 0$
 $C \rightarrow 1$



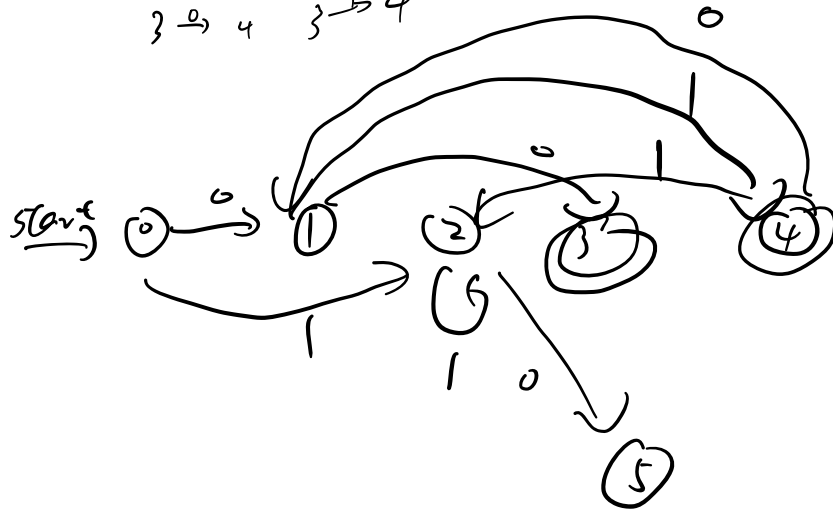
(4) 确定化

$2\text{-closure}(\{0\}) = \{0, 2\}$

2	2_0	2_1
$\{0, 2\}$	$\{1, 3\}$	$\{2\}$
$\{1, 3\}$	$\{4\}$	$\{0, 2, 4\}$
$\{2\}$	$\{3\}$	$\{2\}$
$\{4\}$	ϕ	ϕ
$\{0, 2, 4\}$	$\{1, 3\}$	$\{2\}$
$\{3\}$	$\{4\}$	$\{4\}$

$0 \xrightarrow{0} 1$ $0 \xrightarrow{1} 2$
 $2 \xrightarrow{0} 3$
 $1 \xrightarrow{0} \phi$ $1 \xrightarrow{1} 0 \xrightarrow{0} 2$
 $3 \xrightarrow{0} 4$ $3 \xrightarrow{1} 4$
 $2 \xrightarrow{0} 3$ $2 \xrightarrow{1} 2$
 $0 \xrightarrow{0} 1$
 $2 \xrightarrow{0} 3$
 $3 \xrightarrow{0} 4$ $3 \xrightarrow{1} 4$

状态 \ 输入	0	1
0	1	2
1	3	4
2	5	2
3	-	-
4	1	2
5	3	3



0	1	2
1	3	4
2	5	2
3	-	-
4	1	2

2 确定化

2. (共 10 分) 对于文法 $G[A]$:

$$A \rightarrow iB^*e$$

$$B \rightarrow SB|\epsilon$$

$$S \rightarrow [eC]|.i$$

$$C \rightarrow eC|\epsilon$$

(1) 计算每个产生式右端字符串的 First 和每个非终结符的 Follow 集。

(用 # 代表输入结束)。

(2) 画出该 LL(1) 文法的分析表。

(3) 请写出利用该文法分析句子 $i.i*e$ 的识别过程。

(1)

$$\begin{aligned} \text{FIRST}(A) &= i \\ \text{FIRST}(B) &= \text{FIRST}(S) \cup \{\epsilon\} = \{e, [, .\} \\ \text{FIRST}(S) &= \{[, .\} \\ \text{FIRST}(C) &= \{e, \epsilon\} \\ \text{FIRST}(iB^*e) &= \{i\} \\ \text{FIRST}(SB) &= \{e, ., [\} \\ \text{FIRST}(C) &= \{e\} \\ \text{FIRST}([eC]) &= \{[\} \\ \text{FIRST}(.i) &= \{i\} \\ \text{FIRST}(eC) &= \{e\} \end{aligned}$$

$$\begin{aligned} \text{FOLLOW}(A) &= \{\#\} \\ \text{FOLLOW}(B) &= \{*\} \\ \text{FOLLOW}(S) &= \text{FOLLOW}(B) \cup \text{FIRST}(B) - \{\epsilon\} \\ &= \{*, [, .\} \\ \text{FOLLOW}(C) &= \{]\} \end{aligned}$$

(2)

	i	*	e	[]	.	#
A	$A \rightarrow iB^*e$						
B		$B \rightarrow \epsilon$		$B \rightarrow SB$		$B \rightarrow SB$	
S				$B \rightarrow [eC]$		$B \rightarrow .i$	
C			$C \rightarrow eC$		$C \rightarrow \epsilon$		

(3)

步骤	当前字符串	当前非终结符	当前非终结符
1	# A	i . i * e #	$A \rightarrow iB^*e$
2	# e * B i	i . i * e #	
3	# e * B	. i * e #	$B \rightarrow SB$
4	# e * BS	. i * e #	$S \rightarrow .i$
5	# e * B i .	. i * e #	
6	# e * B i	i * e #	
7	# e * B	* e #	$B \rightarrow \epsilon$
8	# e *	* e #	

3. (共 10 分) 已知文法 G[T]:

$T \rightarrow T - F \mid F$

$F \rightarrow F * P \mid P$

$P \rightarrow (T) \mid i$

(1) 求各非终结符的 FIRSTVT 和 LASTVT 集合。

$FIRSTVT(T) = \{-\} \cup FIRSTVT(F) = \{-, *, (, i\}$

$LASTVT(F) = \{*\} \cup LASTVT(P) = \{*,), i\}$

$LASTVT(P) = \{), i\}$

$LASTVT(T) = \{-\} \cup LASTVT(F) = \{-, *,), i\}$

$LASTVT(T) = \{*\} \cup LASTVT(P) = \{*,), i\}$

$LASTVT(P) = \{), i\}$

$[\# T \#]$

$[T - F]$

$[F * P]$

(2) 构造文法 G 的优先关系矩阵，并判断该文法是否是算符优先文法。

$[(T)]$

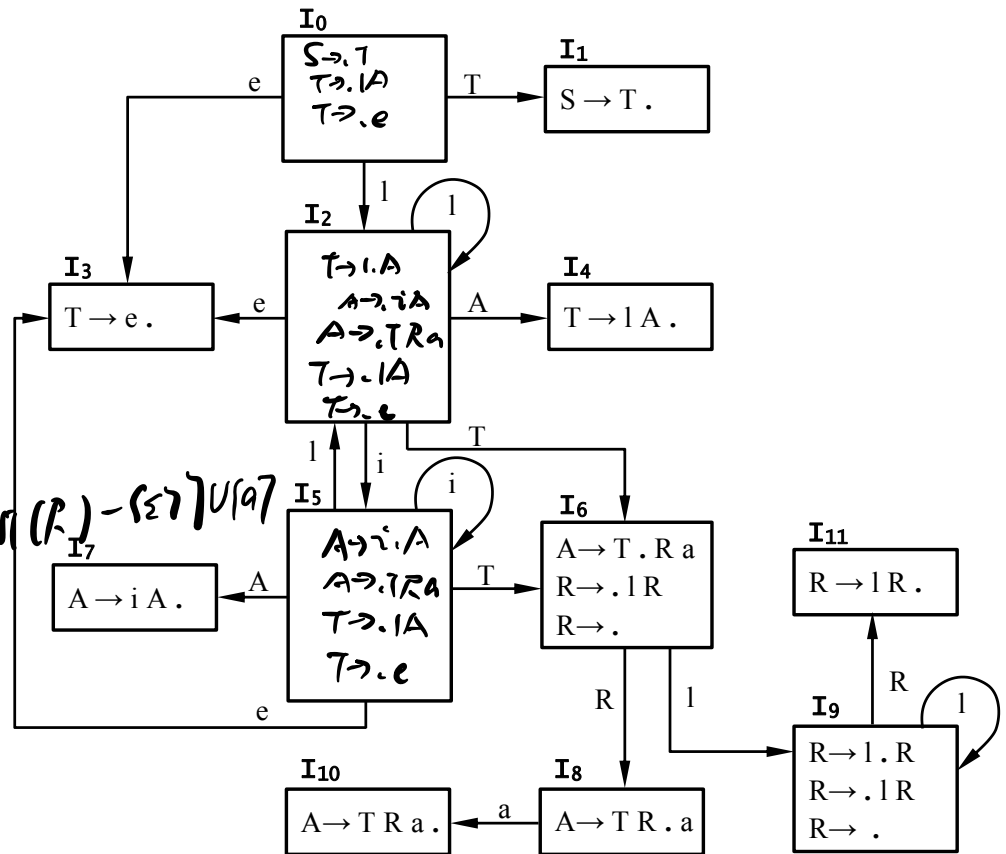
右终结符 (栈外)		-	*	()	i	#
左终结符 (栈内)							
-		>	<	<	>	<	>
*		>	>	<	>	<	>
(<	<	<	=	<	
)		>	>		>		>
i		>	>		>		>
#		<	<	<		<	

是

4. (共 14 分) 已知文法 $G[S]$ 对应的项目集如下图所示。

- ① $S \rightarrow T$
- ② $T \rightarrow lA$
- ③ $T \rightarrow e$
- ④ $A \rightarrow iA$
- ⑤ $A \rightarrow TRa$
- ⑥ $R \rightarrow lR$

$First(R) = \{l, \epsilon\}$
 $Follow(S) = \{\#\}$
 $Follow(T) = Follow(S) \cup First(R) - \{\epsilon\} = \{\#, l, a\}$
 $Follow(A) = Follow(T) = \{\#, l, a\}$
 $Follow(R) = \{a\}$



(1) 试补全上图中项目集 I_0 、 I_2 、 I_5 ，并填写以下的 SLR(1) 分析表。

状态	ACTION					GOTO			
	l	a	i	e	#	S	T	A	R
I ₀	S ₂			S ₃			1		
I ₁					Accept				
I ₂	S ₂		S ₅	S ₃			6	4	
I ₃	V ₃	V ₅			V ₃				
I ₄	V ₂	V ₆			V ₂				
I ₅			S ₅	S ₃			6	7	
I ₆	S ₁	V ₇							8
I ₇	V ₄	V ₄			V ₄				
I ₈		S ₁₀							
I ₉	S ₉	V ₇							11
I ₁₀	V ₅	V ₅			V ₅				
I ₁₁		V ₆							

(2) 判断这个文法是否为 SLR(1)文法, 说明理由。

✓ 并非 LR

(3) 如果是, 利用 SLR(1)分析表, 分析输入串 liella。

步骤	状态栈 (栈底在左)	已识别符号	待输入串	动作
1	# 0	#	liella#	S
2	# 0 1 2	# 1	ella#	S
3	# 0 1 2 5	# 1 5	lla#	S
4	# 0 1 2 5 e 3	# 1 5 e	la#	R: - - -
5	# 0 1 2 5 T 6	# 1 5 T	la#	S
6	# 0 1 2 5 T 6 1 9	# 1 5 T 1	la#	S
7	# 0 1 2 5 T 6 1 9 1 9	# 1 5 T 1 1	a#	R: - - -
8	# 0 1 2 5 T 6 1 9 1 9 R 1 1	# 1 5 T 1 1 R	a#	R: - - -
9	# 0 1 2 5 T 6 1 9 1 1 R 1 1	# 1 5 T 1 1 R	a#	R: - - -
10	# 0 1 2 5 T 6 2 8	# 1 5 T 2	a#	S
11	# 0 1 2 5 T 6 2 8 a 1 0	# 1 5 T 2 a	#	R: - - -
12	# 0 1 2 5 A 7	# 1 5 A	#	R: - - -
13	# 0 1 2 A 6	# 1 A	#	R: - - -
14	# 0 T 1	# 1 T	#	accept
15				
16				
17				
18				
19				
20				
21				
22				
23				

5. (共 10 分) 有如下 C 语言程序段:

```

1 void foo(int val) {
2     int a=1;
3     int b=2;
4 }
5 void bar() {
6     int c,d;
7     int e[10];
8     {
9         int f;
10        {
11            int g;
12            int h[10];
13            foo(g);
14        }
15    }
16 }

```

← 往外跳!

(1) 按照下面的表头格式, 画出**编译到**第 13 行结尾时栈式符号表的内容。

(表格顶端为栈底。不需要写出全局符号) *bar 开始吗?*

层次	名字	种类	类型
0	bar	proc	void
1	c	var	int
1	d	var	int
1	e	array	int
2	f	var	int
3	g	var	int
3	h	array	int

c
d
e 栈帧
e
f
g
h 栈帧
h
ret addr.
prev ebp : 1
val
↑
5

← abp1

(2) 在右表中画出当**运行时**程序控制流从函数 bar 进入, 通过调用 foo 第一次运行到第 3 行结尾时, 程序运行栈上各活动记录的状态和内容。(表格顶端为栈底)

6. (共 5 分) 给定如下四元式代码, 构建 DAG 图, 消除局部公共子表达式, 并使用课本中的启发式算法从 DAG 图中重新导出中间代码。

$$a = b_0 * c_0$$

$$d = b_0$$

$$e = d * c_0$$

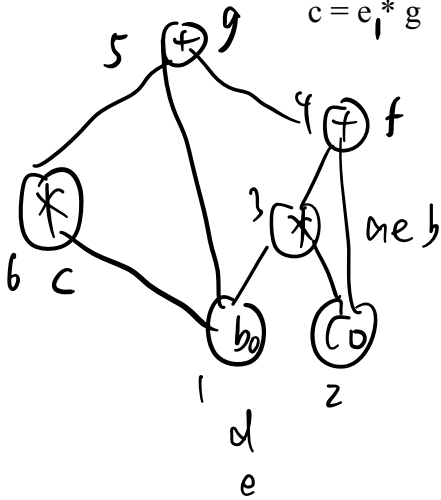
$$b = e$$

$$e_1 = d$$

$$f = b + c_0$$

$$g = d + f$$

$$c = e_1 * g$$



node(x)

b	1	3
c	2	6
a	3	
d	1	
e	2	1
f	4	
g	5	

6 c

5 g

4 f

3 a

c g f a

逆序

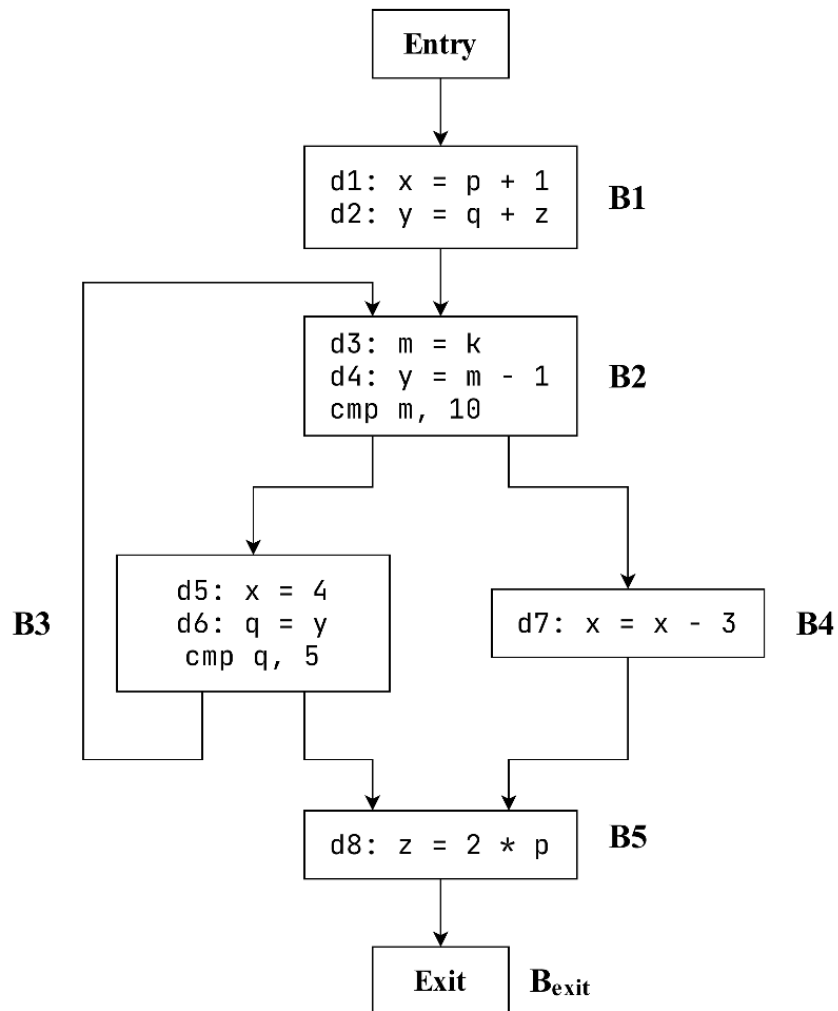
$$a = b_0 * c_0$$

$$f = a * c_0$$

$$g = d + f$$

$$c = b_0 * f$$

7. (共 10 分) 数据流分析。



- (1) 对图中变量，求出每个基本块的 def 和 use 集合；
- (2) 做活跃变量分析，计算每个基本块最终的 in 集合。

(中间各列可用于写明中间过程，最后一行已给出)

	use	def	in	out	in	out	in	out
2 B1	{p,q,z}	{x,y}	{p,q,z,k}	{x,p,k}	{p,q,z,k}	{x,p,k}	{p,q,z,k}	{x,p,k}
3 4 B2	{k}	{m,y}	{x,p,k}	{x,y,p}	{x,p,k}	{x,y,p,k}	{x,p,k}	{x,y,p,k}
4 5 B3	{y}	{x,q}	{y,p}	{p}	{y,p,k}	{x,p,k}	{y,p,k}	{x,p,k}
5 B4	{x}	{}	{x,p}	{p}	{x,p}	{p}	{x,p}	{p}
exit B5	{p}	{z}	{p}	{}	{p}	{}	{p}	{}
B _{exit}			{}	{}	{}	{}	{}	{}

- (3) 假设只有跨越基本块仍活跃的变量才能分配到全局寄存器，且活跃范围重合的变量之间无法共享全局寄存器，根据活跃变量分析结果给出变量之间的冲突图。

x, y, p, q, k, z

