第 11 章　模拟试卷

11.1 模拟试卷 1 及参考答案

11.1.1 模拟试卷 1

一、选择题（10 分，每小题 1 分）

1．编译程序的任务是_____。

A．对机器语言程序解释执行　　B．对汇编程序进行翻译
C．对高级语言程序进行解释执行　　D．对高级语言程序进行翻译

2．有文法 G[E]：

$E \rightarrow T \mid E+T$

$T \rightarrow F \mid T*F$

$F \rightarrow a \mid (E)$

句型　E+(E+T)*F 的句柄是 _____。

A．E　　B．F　　C．E+T　　D．(E+T)

3．在对 C 语言程序进行词法分析时，下面哪个单词符号的确定不需要超前扫描？_____

A．for　　B．+　　C．=　　D．!=

4．LL(1)分析方法是下面哪一种语法分析方法？_____

A．递归下降　　B．非递归预测　　C．规范归约　　D．移进归约

5．某学生编写的 C 语言程序中有声明语句：　int　a, 2c[2];
编译器报告该语句中含有错误，该错误是在编译的哪个阶段被检测出来的？_____

A．词法分析　　B．语法分析　　C．语义分析　　D．代码生成

6．下述文法中，哪一个与正则表达式 a^*b^* 等价？_____

A．$S \rightarrow aSb \mid \varepsilon$　　B．$S \rightarrow aS \mid bS \mid \varepsilon$　　C．$S \rightarrow abS \mid \varepsilon$　　D．$S \rightarrow aS \mid Sb \mid \varepsilon$

7．某学生编写的 C 语言程序中有语句：　a=f(5);
编译器报告该语句中含有错误，错误信息是：函数 f 没有定义，编译程序是在哪个阶段发现此类错误的？_____

A．词法分析　　B．语法分析　　C．语义分析　　D．代码生成

8．有如下翻译方案，若输入串是 aacbb，则翻译结果是_____。

$A \rightarrow aB$ { print(′1′) }

$A \rightarrow c$ { print(′2′) }

$B \rightarrow Ab$ { print(′3′) }

A．13132　　B．23131　　C．12313　　D．13123

9．在 C 语言程序执行过程中，下述哪种对象的存储空间不在过程的活动记录中？_____

A．静态变量　　B．自动变量　　C．形式参数　　D．临时变量

10．源程序中通常包括下述语句，对于其中的哪种语句，编译程序通常不产生可执行代码？_____

A．声明语句　　B．赋值语句　　C．控制语句　　D．输入输出语句

二、（20 分）有如下文法 G[S]：

$E \rightarrow E$+bT | +bT | bT

$T \rightarrow$*cT | *c

(1) 判断该文法是否为 LL(1)文法，说明原因。若不是，做 (2)，若是，做(3)。
(2) 将文法改造为 LL(1)文法，继续做(3)。
(3) 构造文法中非终结符号的 FIRST 和 FOLLOW 集合。
(4) 构造文法的 LL(1)分析表。

三、（30 分）有如下两个文法

（1） $A \rightarrow$aBc　　（2）$A \rightarrow$aBc

$B \rightarrow B$bb | b　　$B \rightarrow$bBb | b

(1) 判断哪一个文法不是 LR(1)文法，要求给出判断过程。
(2) 对于 LR(1)文法，构造其 LR(1)项目集规范族及识别所有活前缀的 DFA。
(3) 构造 LR(1)文法的 LR(1)分析表。
(4) 根据上述分析表，分析输入符号串 abbc，要求给出分析过程，说明分析栈的变化过程及采取的分析动作。

四、（10 分）有如下文法：

S→(L) | a

L→L, S | S

设计一个翻译方案，使其打印出每个 a 在输入符号串中的位置。

比如，对于输入符号串(a, (a, a))，打印输出：2　5　7。

五、（10 分 ）有如下 C 语言代码：

```
typedef    int A[10];
typedef    int B[20];
A a;
typedef    struct {int i;} S;
typedef    struct {int i;} T;
S s;
B *f() { return(&a); }
T    g() { return(s); }
```

(1) 写出这段代码中所定义名字的类型表达式。
(2) C 语言对结构类型采用名字等价，对其它类型采用结构等价。据此判断上述代码中是否存在类型错误。若存在，请指明哪个语句中存在类型错误，并说明产生类型错误的原因。

六、（10 分）有如下 C 语言程序：

```
int m, n;
void f (int x, int y) {
    int m;
    m=30;
    x=x+y;
    n=40;
    y=x+y;
    }
void main( )
    { m=10; n=20;
      f(m, n);
      printf('m=%d, n=%d', m, n);
    }
```

假定采用下面的参数传递机制，该程序的执行结果分别是什么？

（1）传值调用

（2）引用调用

（3）复制恢复

要求：描述程序执行过程的主要步骤。

七、（10 分）有如下三地址代码

(1) i:=1
(2) if i>10 goto (29)
(3) j:=1
(4) if j>20 goto (27)
(5) k:=1
(6) if k>5 goto (25)
(7) t_1:=i*20
(8) t_1:=t_1+j
(9) t_2:=a-84
(10) t_3:=4*t_1
(11) t_4:=i*5
(12) t_4:=t_4+k
(13) t_5:=b-24
(14) t_6:=4*t_4
(15) t_7:=t_5[t_6]
(16) t_8:=k*20
(17) t_8:=t_8+j
(18) t_9:=c-84
(19) t_{10}:=4*t_8
(20) t_{11}:=t_9[t_{10}]
(21) t_{12}:=t_7+t_{11}
(22) t_2[t_3]:=t_{12}

(23) k:=k+1
(24) goto (6)
(25) j:=j+1
(26) goto (4)
(27) i:=i+1
(28) goto (2)
(29) halt

（1）确定其入口语句有哪些，划分基本块，并画出流图。
（2）在该段代码上进行内循环优化，给出优化过程中采用的优化技术及相应的优化结果。

11.1.2 参考答案

一、D C D B B　　D C B A A

二、参考答案：

(1) 该文法不是 LL(1)文法。因为对于产生式 $T\rightarrow$*cT | *c，有

First(*cT)∩ First(*c)={ * }$\neq\Phi$

(2) 将文法改造为 LL(1)文法。

首先消除产生式 $E\rightarrow E$+bT | +bT | bT 中的直接左递归，得到：

$E\rightarrow$+bTE' ｜ bTE'

$E'\rightarrow$+bTE' ｜ ε

其次，对产生式 $T\rightarrow$*cT | *c 进行提取左公因子的改写，得到：

$T\rightarrow$*cT'

$T'\rightarrow T$ ｜ ε

所以，改写后的文法是 $G[E]$：

$E\rightarrow$+bTE' ｜ bTE'

$E'\rightarrow$+bTE' ｜ ε

$T\rightarrow$*cT'

$T'\rightarrow T$ ｜ ε

(3) 构造文法中非终结符号的 FIRST 和 FOLLOW 集合。

各非终结符号的 FIRST 集合和 FOLLOW 集合见表 11-1 所示。

表 11-1 各非终结符号的 FIRST 集合和 FOLLOW 集合

	E	E'	T	T'
FIRST	+, b	+, ε	*	*, ε
FOLLOW	$	$	$, +	$, +

(4) 文法的 LL(1)分析表如表 11-2 所示。

表 11-2 文法的 LL(1)分析表

	+	*	b	c	$
E	$E\rightarrow$+bTE'		$E\rightarrow$bTE'		
E'	$E'\rightarrow$+bTE'				$E'\rightarrow$ε
T		$T\rightarrow$*cT'			
T'	$T'\rightarrow$ε	$T'\rightarrow T$			$T'\rightarrow$ε

三、参考答案：

(1) 文法(2)不是 LR(1)文法。判断过程如下。

首先拓广文法如下：

(0) $S\rightarrow A$　　(1) $A\rightarrow$aBc　　(2) $B\rightarrow$bBb　　(3) $B\rightarrow$b

其次，构造其 LR(1)项目集规范族及识别其所有活前缀的 DFA，见图 11-1。

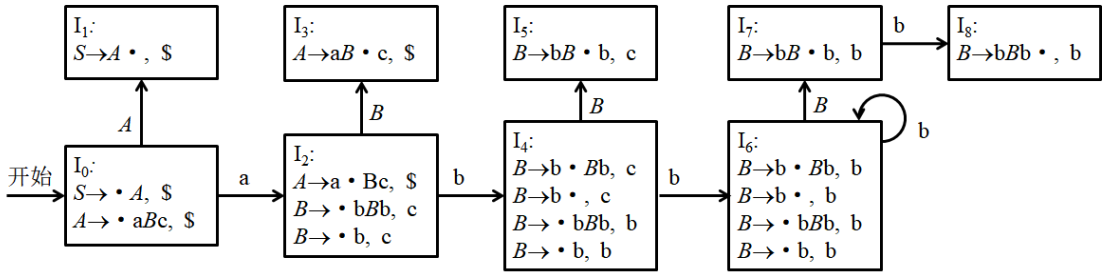


图 11-1　文法(2)的 LR(1)项目集规范族及识别其所有活前缀的 DFA

构造出项目集 I_6 之后，就可以发现其存在移进-归约冲突。因为项目[B→b • , b]是归约项目，当前状态面临下一个输入符号 b 时，要求把当前栈顶的 b 归约为 B。而项目[B→ •bBb, b]和[B→ • b, b]是移进项目，即当前状态面临下一个输入符号 b 时，要求把输入符号移进栈顶。所以，该文法不是 LR(1)文法。

(2) 构造文法(1)的 LR(1)项目集规范族及识别所有活前缀的 DFA。

首先拓广文法(1)为如下文法。

(0) S→A　　(1) A→aBc　　(2) B→Bbb　　(3) B→b

然后构造其 LR(1)项目集规范族及识别所有活前缀的 DFA，构造结果如图 11-2 所示。

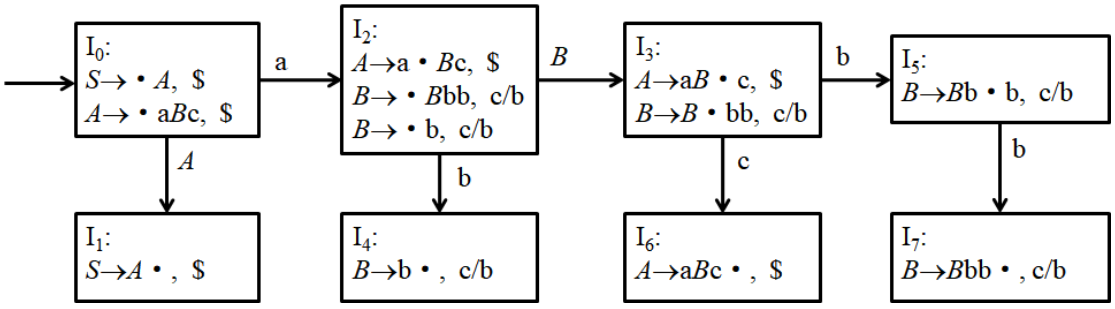


图 11-2　文法(1)的 LR(1)项目集规范族及识别其所有活前缀的 DFA

(3) 构造文法(1)的 LR(1)分析表，结果如表 11-3 所示。

表 11-3　文法(1)的 LR(1)分析表

状态	action				goto	
	a	b	c	$	A	B
0	S2				1	
1				ACC		
2		S4				3
3		S5	S6			
4		R3	R3			
5		S7				
6				R1		
7		R2		R2		

(4) 根据上述分析表，分析输入符号串 abbc，分析栈的变化过程及采取的分析动作见表 11-4 所示。

表 11-4　符号串 abbc 的分析过程

步骤	分析栈	输入	分析动作
(1)	State: 0 Symbol: -	abbc$	S2 移进 a
(2)	State: 0 2 Symbol: - a	bbc$	S4 移进 b
(3)	State: 0 2 4 Symbol: - a b	bc$	R3 用产生式 $B\rightarrow$b 归约
(4)	State: 0 2 3 Symbol: - a B	bc$	S5 移进 b
(5)	State: 0 2 3 5 Symbol: - a B b	c$	报错 处理：弹出栈顶状态
(6)	State: 0 2 3 Symbol: - a B	c$	S6 移进 c
(7)	State: 0 2 3 6 Symbol: - a B c	$	R1 用产生式 $A\rightarrow$aBc 归约
(8)	State: 0 1 Symbol: - A	$	Accept 分析结束

四、参考答案：

定义继承属性 position，记录当前符号的位置信息

定义综合属性 length，记录相应非终结符号推导出的字符串的长度

S'→{ S.position:=1} S

S→({L.position:=S.position+1}

L) { S.length:=L.length+2 }

S→a { S.length:=1; print(S.position) }

L→ { L_1.position:=L.position } L_1,

{ S.position:=L.position+L_1.length+1 } S

{ L.length:=L_1.length+S.length+1 }

L→{ S.position:=L.position } S

{ L.length:=S.length }

五、参考答案：

（1）这段代码中所定义名字及其类型表达式如下：

A：array(0..9, integer)

B：array(0..19, integer)

a: A

S: record(i*integer)

T: record(i*integer)

s:　S

f: void→pointer(B)

g: void→T

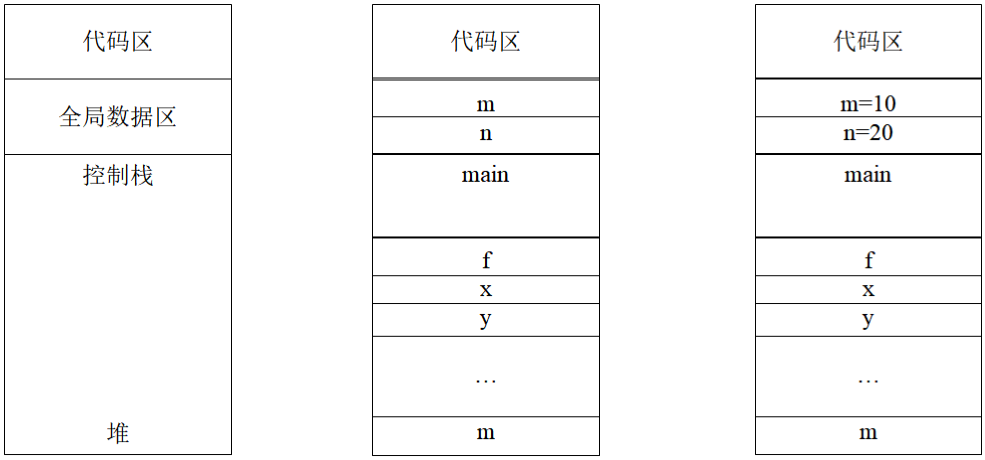
（2）代码中存在类型错误。

函数 g 声明语句中存在类型错误。

因为 g 返回值类型要求是 T，而函数返回的 s 的类型是 S，在 C 语言中，对结构采取名字等价，所以，S 与 T 类型不等价。

六、参考答案：

对于 C 语言程序，运行时存储空间组织如图 11-3(a)所示，图 11-3(b)是在该程序执行过程中，函数 f 执行时的存储空间分布示意图。图 11-3(c)是发生函数调用 f(m, n)时的存储状态。



(a)　　(b)　　(c)

图 11-3　程序运行空间组织示意图

下面针对每种参数传递机制，给出语句及其执行结果。为说明方便，直接用变量名表示其相应的存储空间，用 fm 表示函数 f 中局部变量 m 的存储空间。

（1）传值调用

main 函数将全局变量 m 和 n 的右值分别传递给参数 x 和 y，函数 f 对参数 x 和 y 的操作，均在参数空间进行。

表 11-5　传值调用下，函数 f 中语句及其执行结果

语句	执行结果
m=30;	fm=30
x=x+y;	x=30
n=40;	n=40
y=x+y;	y=50

控制从函数 f 返回后，m=10,　n=40

语句 printf(‘m=%d, n=%d’, m, n)的执行结果是输出：m=10,　n=40

（2）引用调用

main 函数将全局变量 m 和 n 的左值（即存储地址）分别传递给参数 x 和 y。函数 f 对参数 x 和 y 的操作，均在实参 m 和 n 的空间上进行。

表 11-6　引用调用下，函数 f 中语句及其执行结果

语句	执行结果
m=30;	fm=30
x=x+y;	m=30
n=40;	n=40
y=x+y;	n=70

控制从函数 f 返回后，m=30,　n=70

语句 printf(‘m=%d, n=%d’, m, n)的执行结果是输出：m=30,　n=70

（3）复制恢复

main 函数将全局变量 m 和 n 的右值分别传递给参数 x 和 y。函数 f 对参数 x 和 y 的操作，在形参的空间上进行，控制从 f 返回时，将参数空间的内容复制回相应实参的空间。

表 11-7　复制恢复下，函数 f 中语句及其执行结果

语句	执行结果
m=30;	fm=30
x=x+y;	x=30
n=40;	n=40
y=x+y;	y=50

控制从函数 f 返回时，将形参 x 和 y 的当前值复制到相应的实参 m 和 n 的存储空间，故有 m=30,　n=50。

语句 printf(‘m=%d, n=%d’, m, n)的执行结果是输出：m=30,　n=50。

七、参考答案：

（1）这段代码中有 10 个入口语句，分别是语句(1)、(2)、(3)、(4)、(5)、(6)、(7)、(25)、(27)和(29)。

根据入口语句，可以将代码划分为 10 个基本块，各基本块组成如下：

B_1={ (1) }

B_2={ (2) }

B_3={ (3) }

B_4={ (4) }

B_5={ (5) }

B_6={ (6) }

B_7={ (7)~(25) }

B_8={ (25) (26) }

B_9={ (27) (28) }

B_{10}={ (29) }

流图如图 11-4 所示。

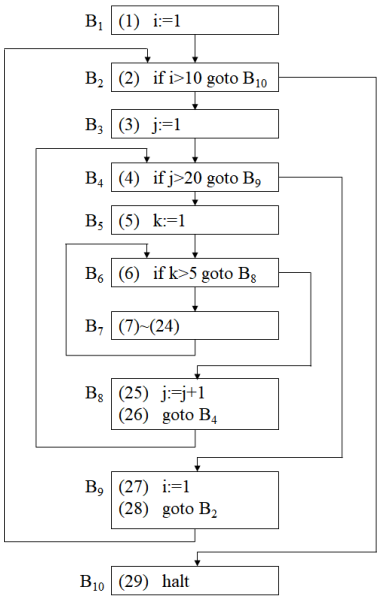


图 11-4 基本块及流图

（2）在该段代码上进行内循环优化，给出优化过程中采用的优化技术及相应的优化结果。

从图 11-14 可以看出，这段代码中有 3 层循环，有基本块 B_6 和 B_7 构成内循环，基本块 B_4、B_5、B_6、B_7 和 B_8 构成中间层循环，基本块 B_2、B_3、B_4、B_5、B_6、B_7、B_8 和 B_9 构成外循环。这里仅对内循环进行优化。

外码外提。对于这段代码而言，a、b 和 c 都是常量，对于内循环而言，i、j 都是循环不变量，故语句(7)、(8)、(9)、(10)、(13)、以及(18)都是循环不变量，可以提到循环之外，放在基本块 B_6 之前，这些语句组成基本块 B_5'，实际上，可以并入基本块 B_5 中。外码外提的结果如图 11-5(a)所示。

削弱计算强度。内循环的循环控制变量 k 每次加 1。由语句(11)和(12)可知，t_4 的初值是 i*5，每次循环其值增加 1。有语句(14) $t_6=4*t_4$ 可知，t_6 每次循环增加 4。故可以将原有的语句(11)、(12)和(14)提到循环外面，放在基本块 B_5'中，增加语句(12′) $t_4=t_4+1$，(14′) $t_6=t_6+4$，并把这两条语句放在语句(23)之后。由语句(16)和(17)可知，$t_8=k*20+j$，每次循环其值增加 20，有语句(19) $t_{10}=4*t_8$ 可知，t_{10} 每次循环增加 80。故可以将原有的语句(16)、(17)和(19)提到循环外面，放在基本块 B_5'中，增加语句(16′)$t_8=t_8+20$，(19′) $t_{10}=t_{10}+80$，并把这两条语句放在语句(24)之前。结果如图 11-5(b)所示。

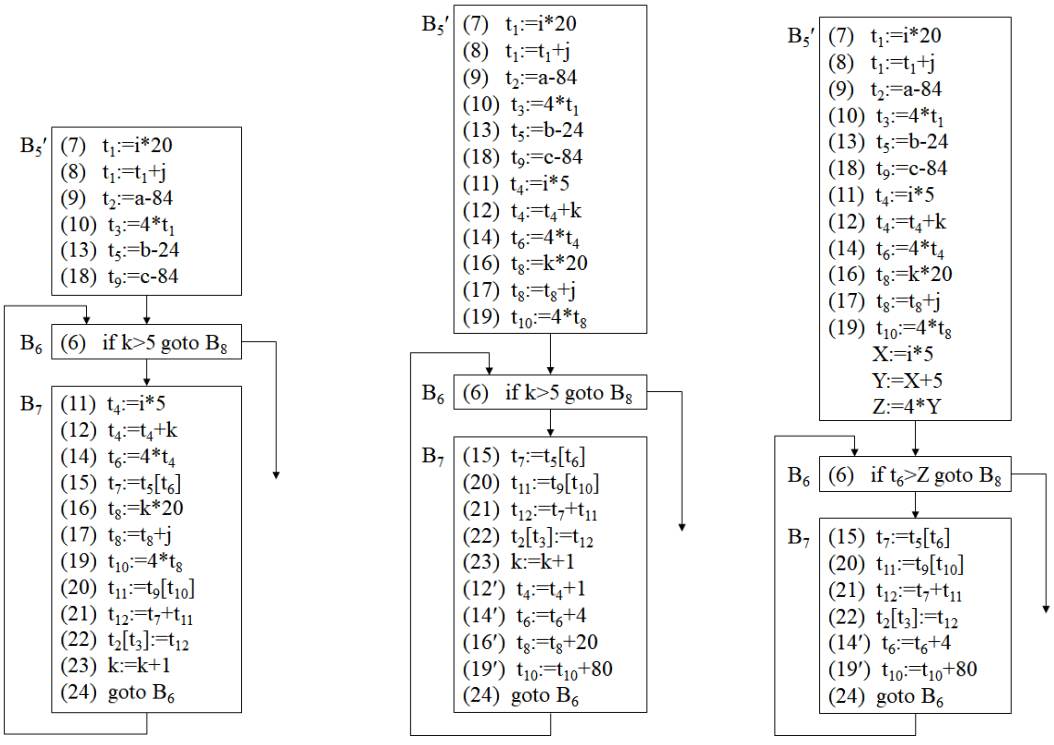
删除归纳变量。内循环的基本归纳变量是 k，$t_4=i*5+k$，$t_6=4*t_4$，$t_8=k*20+j$，$t_{10}=4*t_8$，故 t_4、t_6、t_8、t_{10} 和 k 是同族归纳变量。如果选 t_6 作为循环控制变量，则 k>5 等价于 $t_6>4*(i*5+5)$，由于 4*(i*5)+20 是循环不变量，故在进入循环之前，即在 B_5'中增加相应的计算语句：

X:=i*5

Y:=X+5

Z:=4*Y

这样，就可以用条件 $t_6>Z$ 替换语句(6)中的条件 k>5 了。由于在循环中，k、t_4 和 t_8 没有引用，故可以删除语句(23)、(12′)和(16′)。结果如图 11-5(c)所示。



(a) 代码外提后　　(b) 削弱计算强度后　　(c) 删除归纳变量后

图 11-5　内循环优化过程及结果

11.2 模拟试卷 2 及参考答案

11.2.1 模拟试卷 2

一、（10 分）设字母表 Σ={a, b}，Σ 上的正规表达式 R= b(a|b)*bab

（1）请构造与之等价的 DFA，要求给出 DFA 的状态转换图。
（2）根据上述 DFA，构造与之等价的右线性文法。要求给出产生式集合。

二、（20 分）有如下文法 G[S]：

$E \rightarrow E+T \mid T$

$T \rightarrow A \mid A[E]$

$A \rightarrow i$

(1) 判断该文法是否为 LL(1)文法，说明原因。若不是，做 (2)，若是，做(3)。
(2) 将文法改造为 LL(1)文法 G′，继续做(3)。
(3) 构造文法中非终结符号的 FIRST 和 FOLLOW 集合。
(4) 构造文法的 LL(1)分析表。

三、（30 分）有如下文法 G[S]：

S→mSn | A

A→nA | n

（1）判断该文法是否为 SLR(1)文法，简述理由，若是，进一步构造其 SLR(1)分析表。
（2）判断该文法是否为 LR(1)文法，简述理由，若是，进一步构造其 LR(1)分析表。
（3）判断该文法是否为 LALR(1)文法，简述理由，若是，进一步构造其 LALR(1)分析表。

四、（15 分 ）有如下 C 语言程序：

```
#include<stdio.h>
int i, b[4];
void fp(int x, int y) {
   i=0;
   x+=2;
   b[i]=10;
   y+=3;
   b[i+1]=20;
}
void main() {
   for(i=0; i<4; i++) b[i]=i;
   i=1;
   fp(b[i], b[i+1]);
   for(i=0; i<4; i++) printf(“b[%d]=%d\n”, i, b[i]);
}
```

假定采用下面的参数传递机制，该程序的执行结果分别是什么？要求：给出程序执行过程的主要步骤。（1）传值调用（2）引用调用（3）复制恢复

五、（10 分）有如下文法：

$S \rightarrow L.R \mid L$

$L \rightarrow LB \mid B$

$R \rightarrow BR \mid B$

$B \rightarrow 0 \mid 1$

设计一个语法制导定义，将输入的二进制数转换为等价的十进制数，并输出结果。

六、（15 分）有如下三地址代码:

(1) i:=1
(2) if i<=10 goto (4)
(3) goto (17)
(4) t_1:=a-4
(5) t_2:=4*i
(6) t_3:=a-4
(7) t_4:=4*i
(8) t_5:=t_3[t_4]
(9) t_6:=b-4
(10) t_7:=4*i
(11) t_8:=t_6[t_7]
(12) t_9:=t_5+t_8
(13) t_1[t_2]:=t_9
(14) t_{10}:=i+1
(15) i:=t_{10}
(16) goto (2)
(17) halt

（1）将它划分基本块，并画出控制流图；

（2）在该段代码上进行基本块优化和循环优化，给出优化后的三地址代码。

11.2.2 参考答案

一、参考答案：

（1）首先构造与正规表达式 R= b(a|b)*bab 等价的 NFA，如图 11-6 所示。

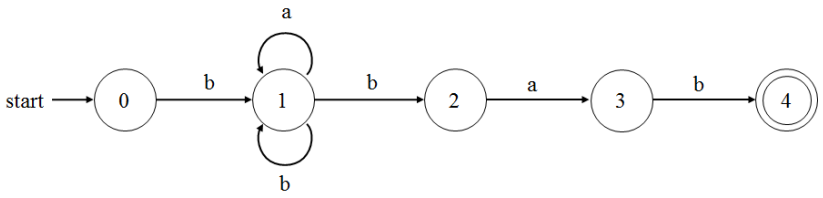


图 11-6　与正规表达式 R= b(a|b)*bab 等价的 NFA

对图 11-6 中的 NFA 确定化，得到如图 11-7 所示的 DFA。

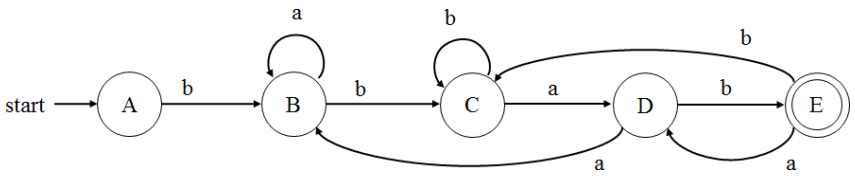


图 11-7　与图 11-6 中 NFA 等价的 DFA

（2）与图 11-7 中 DFA 等价的正规文法如下。

$A \rightarrow bB$

$B \rightarrow aB \mid bC$

$C \rightarrow aD \mid bC$

$D \rightarrow aB \mid bE \mid b$

$E \rightarrow aD \mid bC$

或者：

$A \rightarrow bB$

$B \rightarrow aB \mid bC$

$C \rightarrow aD \mid bC$

$D \rightarrow aB \mid bE$

$E \rightarrow aD \mid bC \mid \varepsilon$

二、参考答案

(1)　该文法不是 LL(1)文法。因为：

由产生式 $E \rightarrow E+T \mid T$ 可知，文法中含有左递归。

由 $T \rightarrow A \mid A[E]$ 可知，FIRST(A)∩FIRST($A[E]$)={i}≠Φ，所以该文法不是 LL(1 文法)。

(2)　首先消除产生式中的左递归。$E \rightarrow E+T \mid T$ 变换为：

$E \rightarrow T\,E'$

$E' \rightarrow +\,T\,E' \mid \varepsilon$

其次，提取左公因子。$T \rightarrow A \mid A[E]$　变换为：

$T \rightarrow A\,T'$

$T' \rightarrow \varepsilon \mid [\,E\,]$

改写后的文法 G′如下：

$E \rightarrow T\,E'$

$E' \rightarrow + \,T\,E' \mid \varepsilon$

$T \rightarrow A\,T'$

$T' \rightarrow \varepsilon \mid [\,E\,]$

$A \rightarrow$ i

(3) 文法中非终结符号的 FIRST 和 FOLLOW 集合见表 11-8 所示。

表 11-8 文法中非终结符号的 FIRST 和 FOLLOW 集合

	First	Follow
E	i	\$,]
E'	+, ε	\$,]
T	i	\$, +,]
T'	[, ε	\$, +,]
A	i	\$, +, [,]

(4) 文法的 LL(1)分析表见表 11-9 所示。

表 11-9 文法的 LL(1)分析表

	i	+	[	]	\$
E	$E \rightarrow T\,E'$				
E'		$E' \rightarrow + \,T\,E'$		$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow A\,T'$				
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow [\,E\,]$	$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
A	$A \rightarrow$ i				

三、参考答案：

拓广文法如下：

(0) S′→S (1) S→mSn (2) S→A (3) A→nA (4) A→n

（1）该文法不是 SLR(1)文法。判断过程和理由如下：

构造该文法的 LR(0)项目集及识别其所有活前缀的 DFA，如图 11-8 所示。

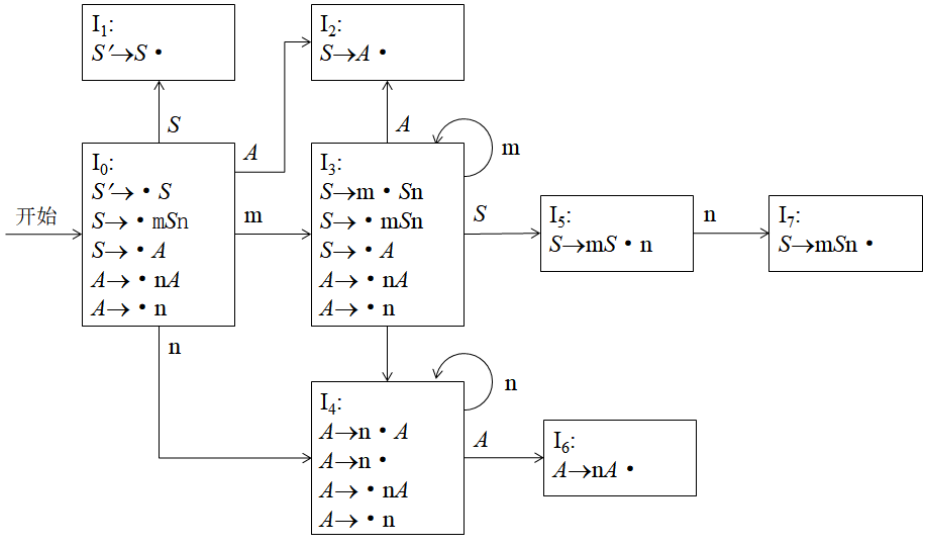


图 11-8 文法的 LR(0)项目集及识别其所有活前缀的 DFA

由图 11-8 可知，项目集 I_4 中含有移进-归约冲突，并且，该冲突无法通过向前看符号解决。因为其中有移进项目 $A \rightarrow \bullet nA$ 和 $A \rightarrow \bullet n$，归约项目 $A \rightarrow n \bullet$，并且 FOLLOW(A)={\$, n}。

（2）该文法不是 LR(1)文法。判断过程和理由如下：

构造该文法的 LR(1)项目集及识别其所有活前缀的 DFA，如图 11-9 所示。

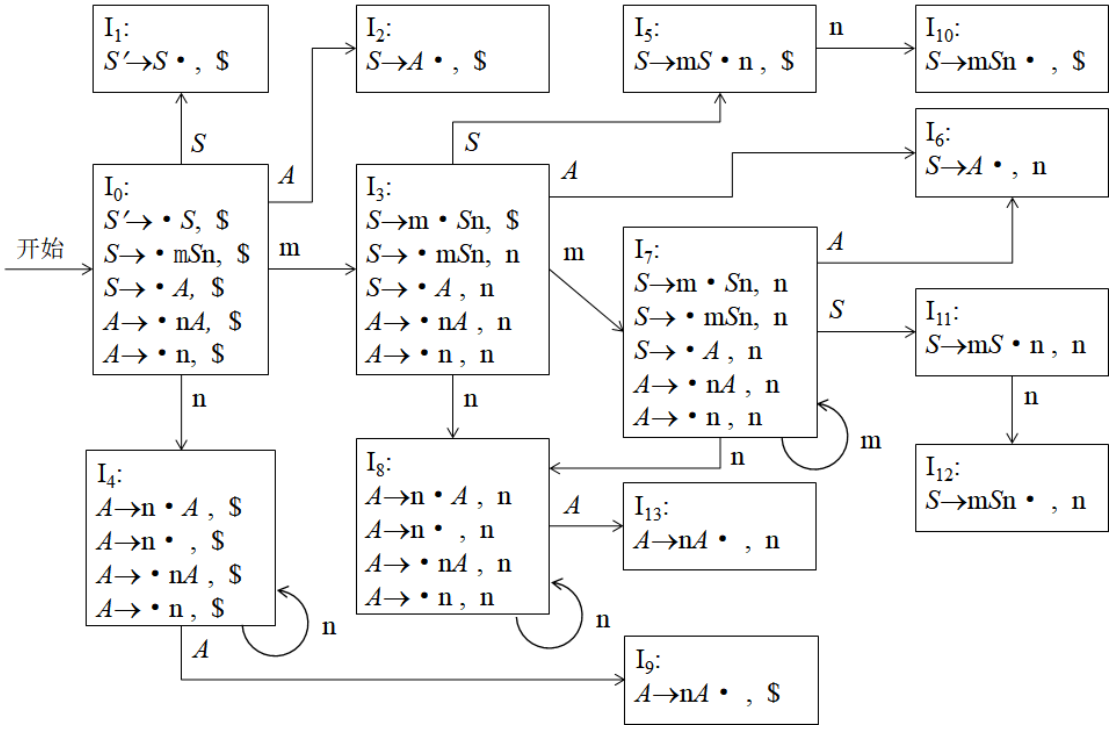


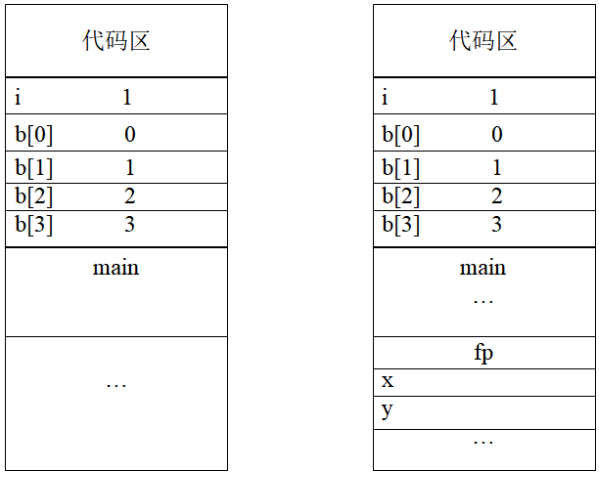
图 11-9 文法的 LR(0)项目集及识别其所有活前缀的 DFA

由图 11-9 可知，项目集 I_8 中含有移进-归约冲突，因为其中有移进项目[$A\rightarrow$ • nA, n]和[$A\rightarrow$ • n, n]，同时含有归约项目[$A\rightarrow$n • , n]。

（3）该文法不是 LALR(1)文法。因为非 LR(1)文法，都不是 LALR(1)文法。

四、参考答案：

主程序执行完 i=1 之后，调用函数 fp 之前的数据对象空间及其状态如图 11-10(a)所示。控制进入函数 fp 之后，程序运行空间组织及其状态如图 11-10(b)所示。



(a) (b)

图 11-10 程序运行空间组织示意图

（1）传值调用

main 函数将全局变量 b[1]和 b[2]的右值分别传递给参数 x 和 y，函数 fp 对参数 x 和 y 的操作，均在参数空间进行。

表 11-10　传值调用下，函数 fp 中语句及其执行结果

语句	执行结果
i=0;	i=0
x+=2;	x=3
b[i]=10;	b[0]=10
y+=3;	y=5
b[i+1]=20	b[1]=20

函数 fp 对全局变量的影响：i=0，b[0]=10，b[1]=20

语句 for(i=0; i<4; i++) printf(“b[%d]=%d\n”, i, b[i])的执行结果是输出：

b[0]=10　　b[1]=20　　b[2]=2　　b[3]=3

（2）引用调用

main 函数将全局变量 b[1]和 b[2]的左值（即存储地址）分别传递给参数 x 和 y。函数 fp 对参数 x 和 y 的操作，均在实参 b[1]和 b[2]的空间上进行。

表 11-11　引用调用下，函数 fp 中语句及其执行结果

语句	执行结果
i=0;	i=0
x+=2;	b[1]=3
b[i]=10;	b[0]=10
y+=3;	b[2]=5
b[i+1]=20	b[1]=20

函数 fp 对全局变量的影响：i=0，b[0]=10，b[1]=20，b[2]=5

语句 for(i=0; i<4; i++) printf(“b[%d]=%d\n”, i, b[i])的执行结果是输出：

b[0]=10　　b[1]=20　　b[2]=5　　b[3]=3

（3）复制恢复

main 函数将全局变量 b[1]和 b[2]的右值分别传递给参数 x 和 y。函数 fp 对参数 x 和 y 的操作，在形参的空间上进行，控制从 fp 返回时，将参数空间的内容复制回相应实参的空间。

表 11-12　复制恢复下，函数 fp 中语句及其执行结果

语句	执行结果
i=0;	i=0
x+=2;	x=3
b[i]=10;	b[0]=10
y+=3;	y=5
b[i+1]=20	b[1]=20

控制从函数 fp 返回时，将形参 x 和 y 的当前值复制到相应的实参 b[1]和 b[2]的存储空间，函数 fp 对全局变量的影响：i=0，b[0]=10，b[1]=3，b[2]=5

语句 for(i=0; i<4; i++) printf(“b[%d]=%d\n”, i, b[i])的执行结果是输出：

b[0]=10　　b[1]=3　　b[2]=5　　b[3]=3

五、参考答案：

为符号 S、L、R、B 定义综合属性 val，记录与之相应的二进制数对应的十进制数值。

语法制导定义如下：

$S'\rightarrow S$ { print(S.val) }

$S\rightarrow L.R$ { S.val=L.val+R.val }

$S\rightarrow L$ { S.val=L.val }

$L\rightarrow L_1B$ { L.val=L_1.val*2+B.val }

$L\rightarrow B$ { L.val=B.val }

$R\rightarrow BR_1$ { R.val=(B.val+R_1.val)/2 }

$R\rightarrow B$ { R.val=B.val/2 }

$B\rightarrow 0$ { B.val=0 }

$B\rightarrow 1$ { B.val=1 }

六、参考答案：

（1）这段代码有 5 个入口语句，分别是语句(1)、(2)、(3)、(4)和(17)，可以划分为 5 个基本块，分别是：

B_1={ (1) }

B_2={ (2) }

B_3={ (3) }

B_4={ (4)~(16) }

B_5={ (17) }

在基本块之间加入控制信息，得到如图 11-11 所示的流图。

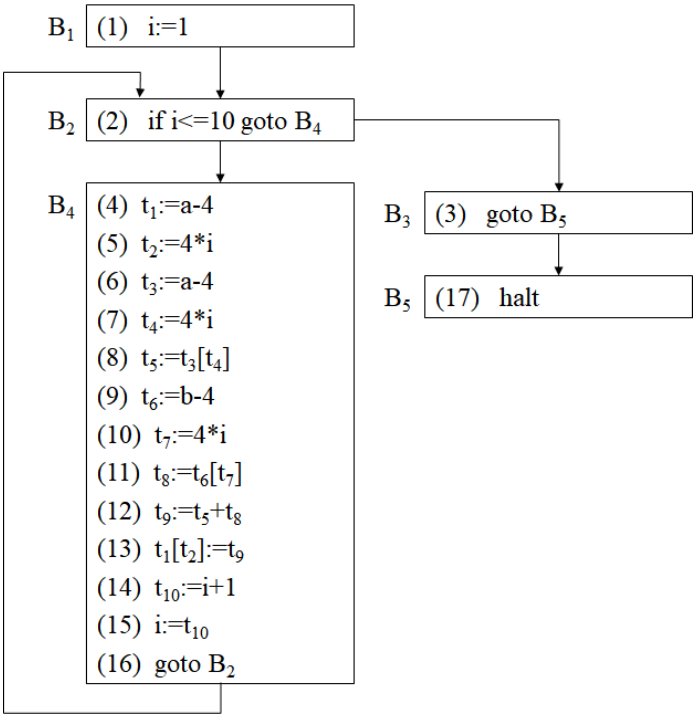


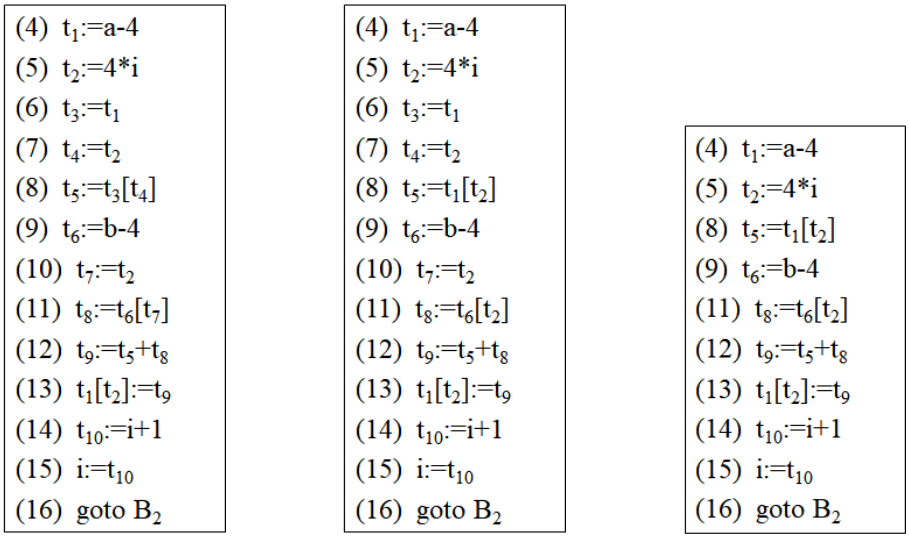
图 11-11 流图

（2）在该段代码上进行基本块优化和循环优化。

首先对基本块 B_4 进行基本块优化。

删除公共子表达式。结果如图 11-12(a)所示，其次，进行复制传播，结果如图 11-12(b)

所示，然后，删除死代码，结果如图 11-12(c)所示。



(a) 删除公共子表达式后　　(b) 复制传播后　　(c) 删除死代码后

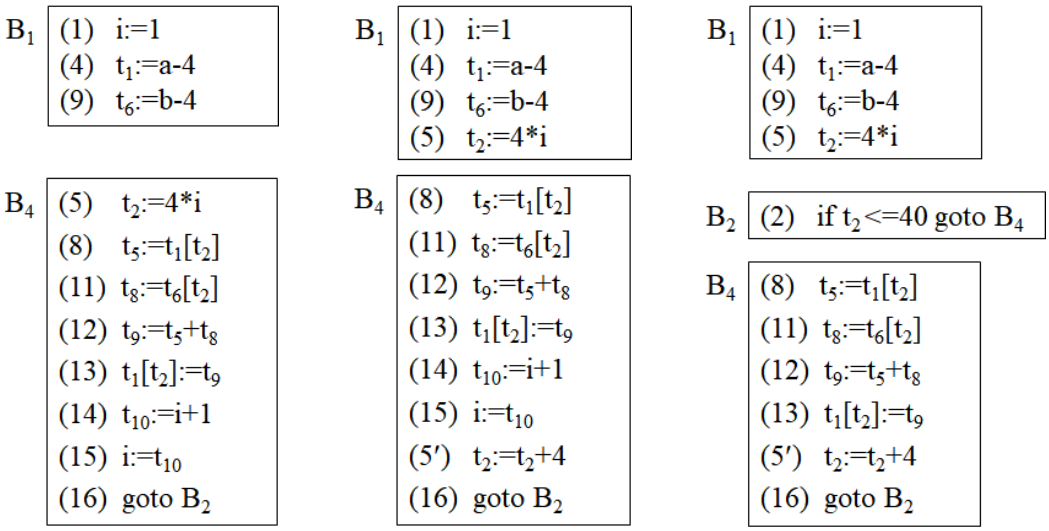
图 11-12 对 B_4 进行基本块优化过程和结果

然后进行循环优化。

代码外提。结果如图 11-13(a)所示。

削弱计算强度。结果如图 11-13(b)所示。

删除归纳变量。结果如图 11-13(c)所示。



(a) 代码外提后　　(b) 削弱计算强度后　　(c) 删除归纳变量后

图 11-13 循环优化过程和结果

最后，再对基本块 B_1 进行常数传播与常数合并、删除死代码优化，结果如图 11-14 所示。

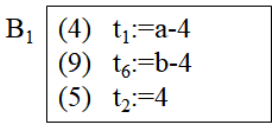


图 11-14 基本块 B_1 优化结果

经过基本块优化和循环优化后，结果如图 11-15 所示。

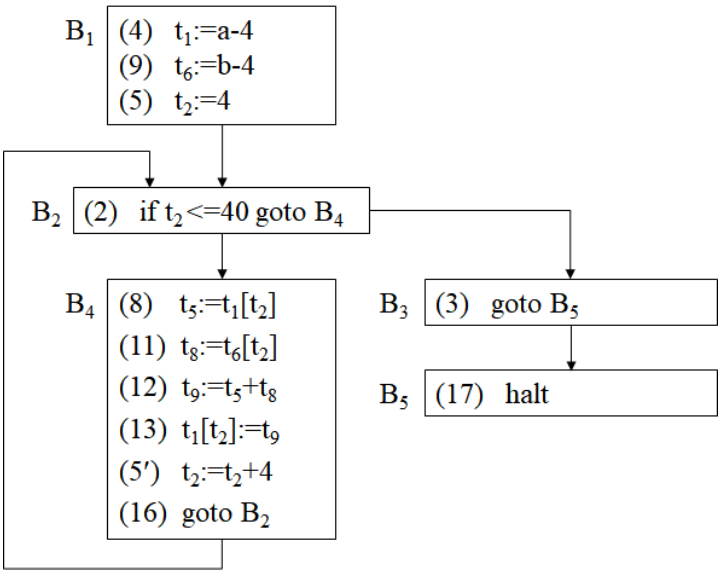

B1
(4) t1:=a-4
(9) t6:=b-4
(5) t2:=4
B2
(2) if t2<=40 goto B4
B4
(8) t5:=t1[t2]
(11) t8:=t6[t2]
(12) t9:=t5+t8
(13) t1[t2]:=t9
(5′) t2:=t2+4
(16) goto B2
B3
(3) goto B5
B5
(17) halt

图 11-15 优化结果