

北京航空航天大学

2021 - 2022 学年 第一学期期末

《编译技术》

考 试 A 卷

班 级_____学 号 _____

姓 名_____成 绩 _____

考试地点_____

2021 年 12 月 21 日

班号_____ 学号_____ 姓名_____ 成绩_____

《 编译技术 》 期末考试卷

题目：

- 一、填空题.....(20 分)
- 二、正则文法与自动机..... (15 分)
- 三、LL(1)和算符优先分析法 (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.....(20 分)

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

注： 1. 试卷共 11 页（不含封面和目录），请仔细检查。

2. 请在封面和答卷上都写上学号和姓名， 试卷和答卷不能拆卸，一起上交，缺一无效。

3. 在监考老师统计完试卷后，再离开考场；

4. 选择题答案写在试卷纸上。

一. 填空题（共 20 分，每空 1 分）

1. 编译过程本质上是一种 翻译 过程，将用 高级程序语言 书写的源程序加工为与其等价的目标程序。

2. 在编译过程的五个基本阶段中都要做 错误处理 和 符号表管理 两件事，因此典型的编译程序常划分为七个逻辑组成部分。

3. 对源程序（包括源程序中间形式）从头到尾扫描一次，并做有关的加工处理，生成新的源程序中间形式或目标程序，通常称之为 遍，完成编译工作最少需要对源程序做 一 次扫描。

4. 生成中间代码的目的是便于做 代码优化 和 不同目标语言移植

5. 有文法规则 $S \rightarrow \text{if } E \text{ } S \mid \text{if } E \text{ } S \text{ else } S$ ，用扩充的 BNF 范式表示为 $S \rightarrow \text{if } E \text{ } S [\text{else } S]$ 。

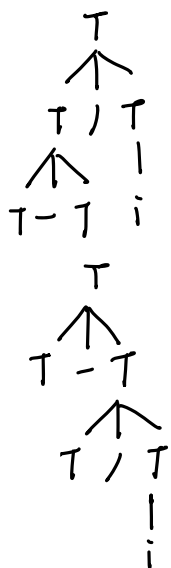
6. 常见的程序设计语言的文法按乔姆斯基的分类是 2 型文法，也称为上下文无关文法。如果采用属性翻译文法处理声明语句 `int a;` 时，通常可以得到变量的类型和名字这样的 综合 属性，并填入到 符号表 中，以便在使用变量 `a` 时，能够查找到变量的有关信息。没有声明就使用变量，属于 语义 错误，在语法分析只进行句子的结构分析时并不能发现这个问题。

7. 对文法 $G[T]: T ::= T-T \mid T/T \mid (T) \mid i$ ，规范句型 $T-T/i$ 的句柄为 i 和 $T-T$ ，由此判断该文法 有（有/无）二义性。

8. 规范归约每次归约的是句型的 最左简单短语，算符优先分析法每次归约的是当前句型的 最左素短语。

9. 活动记录中 Display 区存放的是 外层活动记录的基地址。

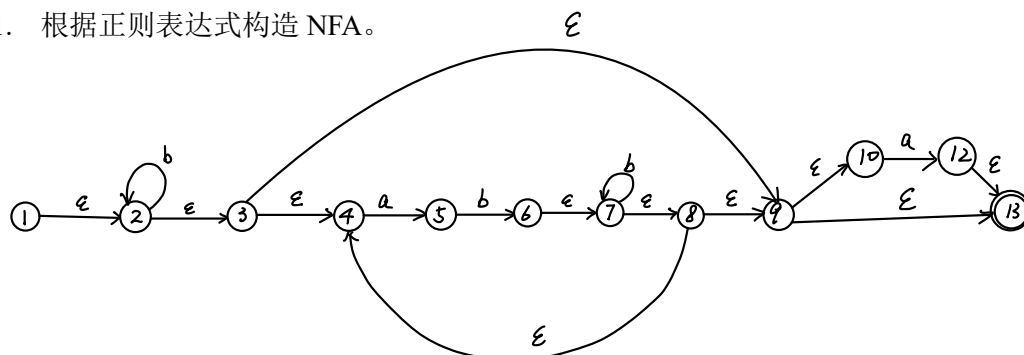
10. 文法 $G = (V_n, V_t, P, Z)$ ，其中 V_t 代表 终结符集。



二. 有如下正则表达式 (15 分)

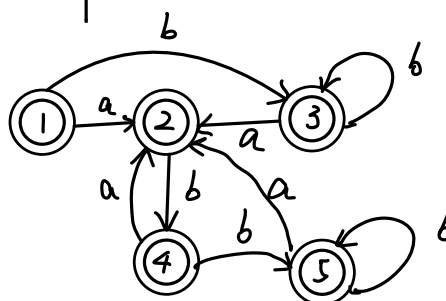
$b^* (abb^*)^* (a|\epsilon)$

1. 根据正则表达式构造 NFA。



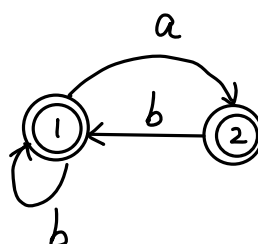
2. 将所得到的 NFA 确定化。

	I	I_a	I_b
1	{1, 2, 3, 4, 9, 10, 13}	{5, 12, 13}	{2, 3, 4, 9, 10, 13}
2	{5, 12, 13}	\varnothing	{6, 7, 8, 9, 4, 10, 13}
3	{2, 3, 4, 9, 10, 13}	{5, 12, 13}	{2, 3, 4, 9, 10, 13}
4	{6, 7, 8, 9, 4, 10, 13}	{5, 12, 13}	{7, 8, 9, 10, 4, 13}
5	{7, 8, 9, 10, 4, 13}	{5, 12, 13}	{7, 8, 9, 10, 4, 13}



3. 将所得到的 DFA 最小化。

	a	b	
1	2	3	I
2	\varnothing	4	
3	2	3	I
4	2	5	
5	2	5	



三. (共 15 分)

1. 试证明所有二义性文法都不是 LL (1) 文法。(6 分)

二义性文法 对于文法所包法后句子有
两棵语法树 即 两个不同的句柄
则会有多重入口, 不是 LL(1) 文法

2. 已知文法 G[T]: (共 9 分)

$$\begin{aligned} T &\rightarrow T - F \mid F && \#T\# \\ F &\rightarrow F / P \mid P \\ P &\rightarrow (T) \mid i \end{aligned}$$

(1) 求各非终结符的 FIRSTVT 和 LASTVT 集合。(3 分)

$$\begin{aligned} \text{FIRSTVT}(T) &= \{ -, /, (, i \} & \text{LASTVT}(T) &= \{ -, /,), i \} \\ \text{FIRSTVT}(F) &= \{ /, (, i \} & \text{LASTVT}(F) &= \{ /,), i \} \\ \text{FIRSTVT}(P) &= \{ (, i \} & \text{LASTVT}(P) &= \{), i \} \end{aligned}$$

(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。(6 分)

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

是

四. 有如下文法 $G[S]$: (共 15 分)

$$S \rightarrow CD \mid DC$$

$$C \rightarrow aCb \mid ab$$

$$D \rightarrow Db \mid b$$

拓展: $S' \rightarrow S$

(1) $S \rightarrow CD$ $FIRST(S) = \{a, b\}$ $FOLLOW(S) = \{\#\}$

(2) $S \rightarrow DC$ $FIRST(C) = \{a\}$ $FOLLOW(C) = \{\#, b\}$

(3) $C \rightarrow aCb$ $FIRST(D) = \{b\}$ $FOLLOW(D) = \{b, \#, a\}$

(4) $C \rightarrow ab$

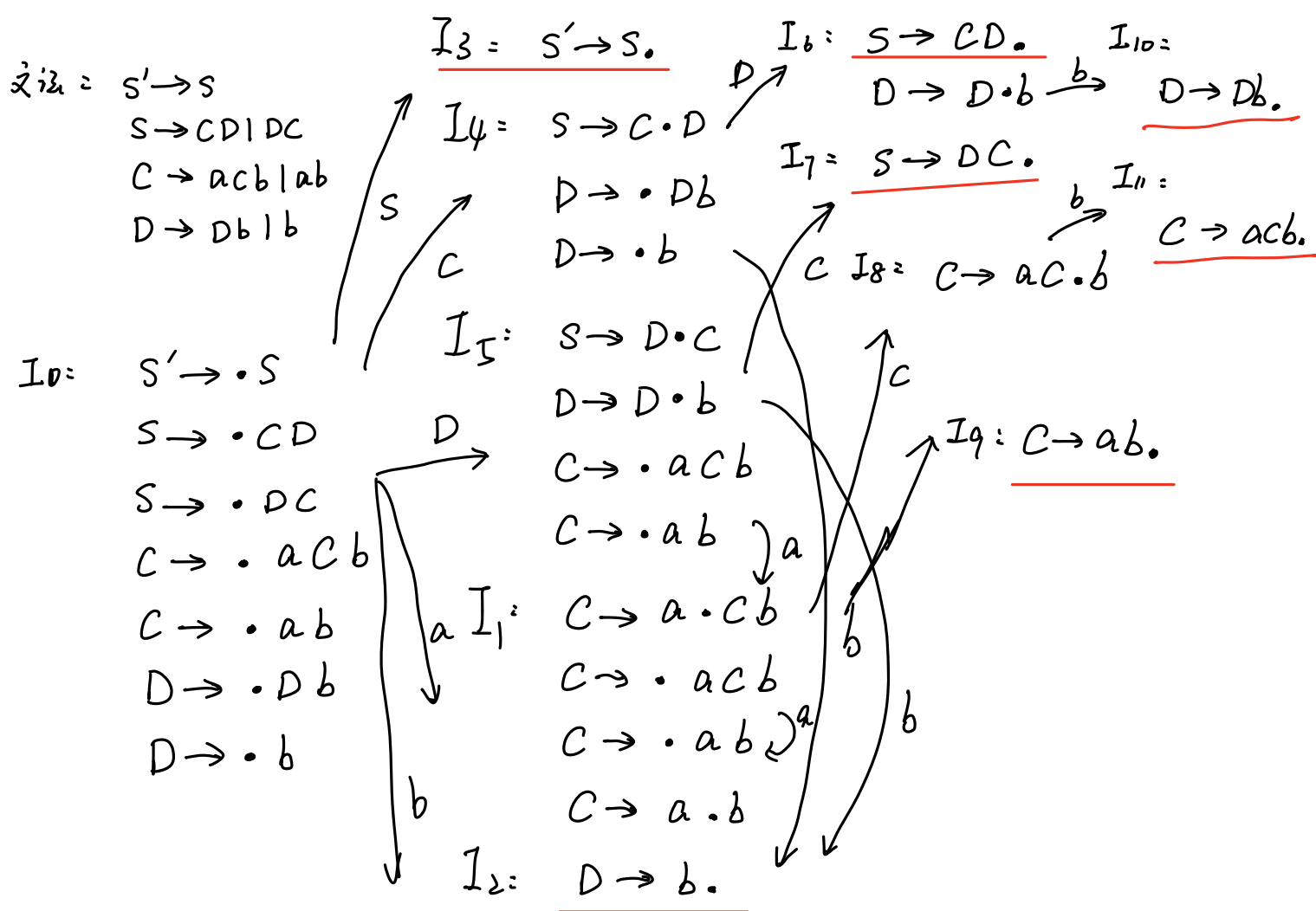
(5) $D \rightarrow Db$

(6) $D \rightarrow b$

1. 拓展文法, 使得文法的开始符号仅出现在一个产生式的左侧; 求原文法所有非终结符的 FOLLOW 集。

2. 求拓展后文法的 SLR 分析表, 包括 GOTO 表和 ACTION 表, 表头如下 (LR(0)项目集直接填在表格中):

ACTION 表				GOTO 表		
状态	a	b	#	S	C	D
0	S1	S2		3	4	5
1	S1	S9			8	
2	r1	r7	r7			
3			acc			
4		S2				b
5		S2			7	
6		S10	r2			
7			r3			
8		S11				
9		r5	r5			
10	r6	r6	r6			
11		r4	r4			



3. 求能识别规范句型 aabbbb 活前缀的有效项目集。

句柄: ab

活前缀: a, aab, aa

$a:$

- $C \rightarrow a \cdot Cb$
- $C \rightarrow \cdot acb$
- $C \rightarrow \cdot ab$
- $C \rightarrow a \cdot b$

$aab:$

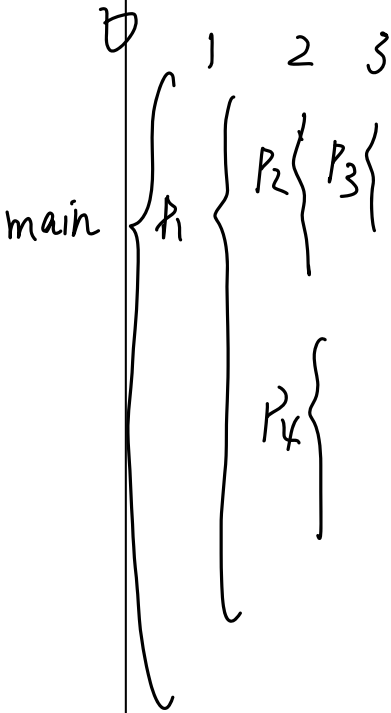
- $C \rightarrow ab \cdot$

$aa:$

- $C \rightarrow a \cdot Cb$
- $C \rightarrow \cdot acb$
- $C \rightarrow \cdot ab$
- $C \rightarrow a \cdot b$

五. 有如下程序段, (共 15 分)

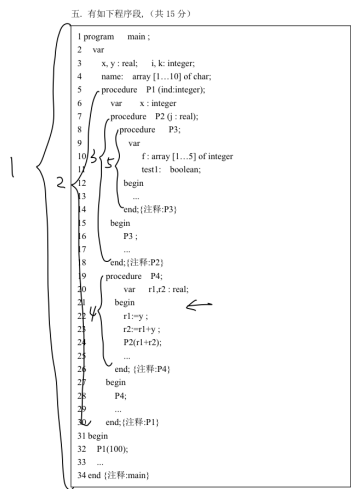
```
1 program    main ;
2  var
3    x, y : real;    i, k: integer;
4    name:  array [1...10] of char;
5    procedure P1 (ind:integer);
6      var    x : integer
7      procedure P2 (j : real);
8        procedure P3;
9          var
10             f : array [1...5] of integer
11             test1:  boolean;
12         begin
13             ...
14             end;{注释:P3}
15         begin
16             P3 ;
17             ...
18         end;{注释:P2}
19     procedure P4;
20       var    r1,r2 : real;
21       begin
22         r1:=y ;
23         r2:=r1+y ;
24         P2(r1+r2);
25         ...
26       end; {注释:P4}
27     begin
28       P4;
29       ...
30     end;{注释:P1}
31 begin
32   P1(100);
33   ...
34 end {注释:main}
```



1. 按照以下格式，画出递归下降编译到第 21 行时，栈式符号表的内容：

序号	名字	种类	类型	层号
1	x	var	real	1
2	y	var	real	1
3	i	var	integer	1
4	k	var	integer	1
5	name	var	array	1
6	p1	procedure.		1
7	ind	param	integer	2
8	x	var	integer	2
9	p2	procedure.		2
10	p4	procedure.		2
11	r1	var	real	2
12	r2	var	real	2

2. 运行到第 12 行时，运行栈的内容如下所示，将空白处填满：

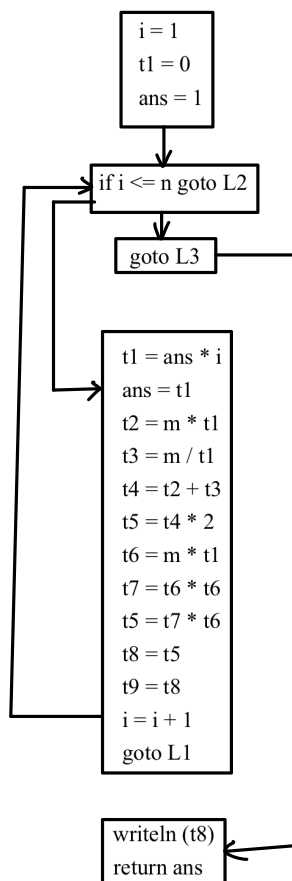


test1	
f	
f 模板	
prev abp:abp4	
ret addr	
abp4(DISPLAY)	
abp2(DISPLAY)	
abp1(DISPLAY)	abp5: P3
j	
prev abp: abp3	
ret Addr	
abp2(DISPLAY)	
abp1(DISPLAY)	abp4: P2
r2	
r1	
prev abp: abp1	
ret addr	
abp2(DISPLAY)	
abp1(DISPLAY)	abp3: P4
x	
ind	
prev abp:abp1	
ret addr	
abp1(DISPLAY)	abp2: P1
name	
name 模板	
k	
i	
y	
x	abp1 : main

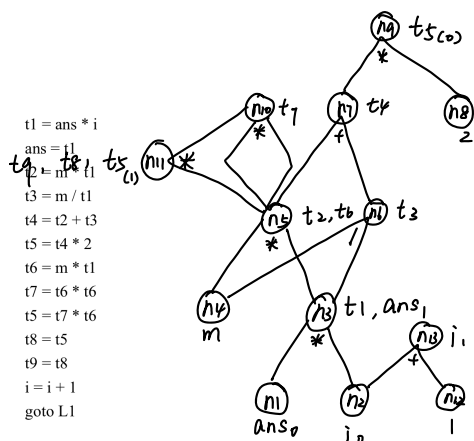
六. (共 20 分) 有如下程序, 其中 n 、 m 是形参, i 、 ans 、 $t1$, $t2$, $t3$, $t4$, $t5$, $t6$, $t7$, $t8$, $t9$ 都是局部变量。

```
i = 1
t1 = 0
ans = 1
L1: if i <= n goto L2
    goto L3
L2: t1 = ans * i
    ans = t1
    t2 = m * t1
    t3 = m / t1
    t4 = t2 + t3
    t5 = t4 * 2
    t6 = m * t1
    t7 = t6 * t6
    t5 = t7 * t6
    t8 = t5
    t9 = t8
    i = i + 1
    goto L1
L3: writeln (t8)
    return ans
```

1. 将该代码划分基本块, 构造相应的控制流图。



2. 试对 L2 所在的基本块用 DAG 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。



node	id
ans	3
i	13
t1	3
m	4
t2	5
t3	6
t4	7
2	8
t5	11
t6	5
t7	10
t8	11
t9	11

中间代码=

按: $t_{5(1)}, t_{5(0)}, t_7, t_4, t_2, t_3, t_1, i,$

$$i_1 = i_0 + 1$$

$$t_1 = i_0 + ans_0$$

$$t_3 = m / t_1$$

$$t_2 = m * t_1$$

$$t_4 = t_2 + t_3$$

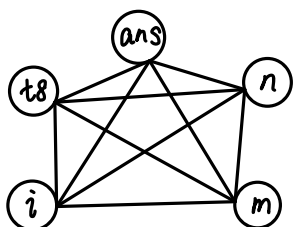
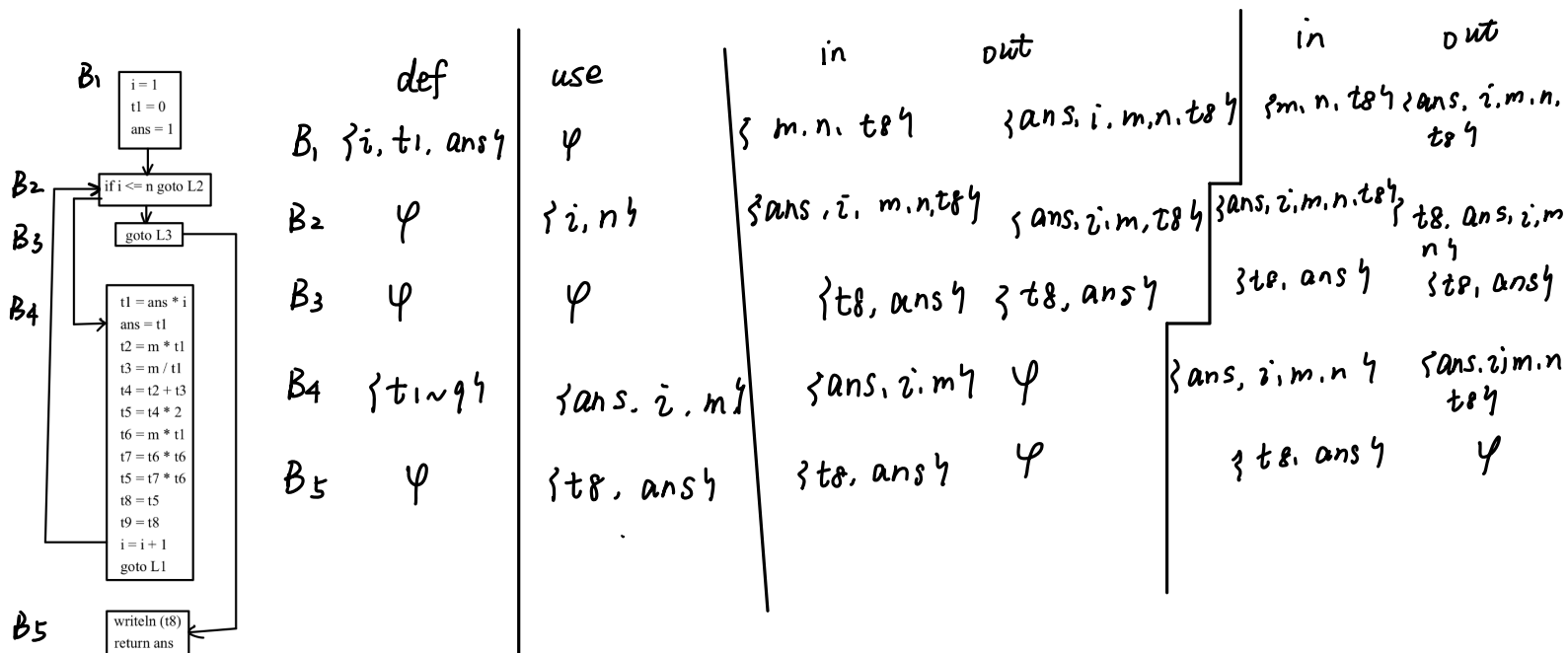
$$t_7 = t_2 * t_2$$

$$t_{5(0)} = t_4 * 2$$

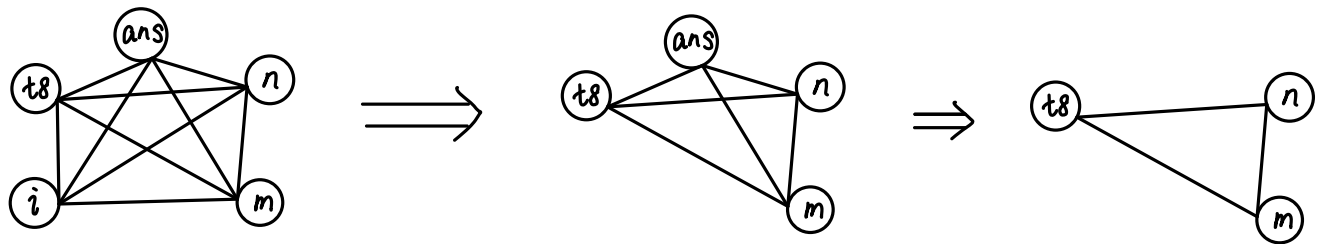
$$t_{5(1)} = t_7 * t_2 \quad [\quad t_9 = t_{5(1)} \quad t_8 = t_{5(1)} \quad]$$

3. 给出每个基本块的 def 和 use 集合，做活跃变量分析，并给出变量的冲突图。注意：变量

A, B 冲突的标准为，变量 B 的定义点处变量 A 活跃，反之亦然。



4. 假设全局寄存器为{X1,X2}, 采用图着色方法做全局寄存器分配, 并写明过程。



首先选择 i 入栈, 再选择 ans 入栈.

所以分配结果为, ans 和 i 用全局寄存器, 其它用内存