

2003-2004 学年第二学期《编译原理》期末考试

一：填充题（如无特殊说明，每空两分）

1. 在标准的编译程序的组成部分中，属于前端部分的有：\_\_\_\_\_，  
\_\_\_\_\_，和\_\_\_\_\_。
2. 给出语言为  $\{a^n b^j c^i \mid n \geq 0, j \geq 0, 0 \leq i \leq n\}$  对应的文法  $G[Z]$ :  
\_\_\_\_\_。这个  
语言可以使用正则文法来描述吗？\_\_\_\_\_，如果能，给出正则  
文法，如果不能，简单说明理由：\_\_\_\_\_。
3. 给定文法  $G[E]$ :  $E \rightarrow T+E \mid T$      $T \rightarrow F^*T \mid F$      $F \rightarrow '(' E ') \mid i$ ; 给出句子  
(i\*i)的最左推导过程：\_\_\_\_\_，并  
给出语法树：

4. 给定文法:  $G[Z]$ :  $Z \rightarrow WV$      $W \rightarrow aB \mid aW \mid a$      $B \rightarrow b \mid bB$      $V \rightarrow bV \mid cC$   
 $C \rightarrow c \mid cC$ 。说明文法  $G[Z]$  为二义性的:

\_\_\_\_\_。(本空 4 分)

给出文法的语言：\_\_\_\_\_。

并给出等价的无二义性的文法：

\_\_\_\_\_。(本空 4 分)

5. 给出和正则文法  $W \rightarrow A0$      $A \rightarrow A0 \mid W1 \mid 0$  等价的有穷状态自动机：

\_\_\_\_\_。(本空 4 分)

6. 对于赋值语句序列  $c = (a+b)*(c+d)$ ;  $x = (a+b)-(c+d)$ 。给出对应的四元式序列：

\_\_\_\_\_。  
找出公共子表达式并进行优化后得到如下四元式序列：

\_\_\_\_\_。(本空 4 分)

从上面的赋值语句序列中得到等价的逆波兰表达式：

7. 对于文法  $G[S]$ :  $S \rightarrow a \mid b \mid (T)$   $T \rightarrow T, S \mid S$  ; 给出  $LR(0)$  项集  $\{T \rightarrow \cdot T, S\}$  的闭包:

8. 给定下列文法  $G[E]$  以及翻译方案

$E \rightarrow T + E_1$   $\{E.val = T.val + E_1.val\}$

$E \rightarrow T - E_1$   $\{E.val = T.val - E_1.val\}$

$E \rightarrow T$   $\{E.val = T.val\}$

$T \rightarrow i$   $\{T.val = i.lexVal\}$

按照这个翻译方案，求解句子 9-3-2 的值：

9. 给出  $(a+b)*(c-d)+e$  的抽象语法树：

10. 简单解释为什么  $SLR(1)$  分析技术的处理范围要比  $LR(1)$  技术的处理范围小？

11. 说明在程序运行的时候下列变量/值的内存分配方式：(1) 局部变量 `vlocal` (2) 全局变量 `p` 指向的整数值，该值由 `malloc` 申请空间。 (静态分配填写 `a`，栈式分配填写 `b`，堆式分配填写 `c`)。

二：使用递归子程序方法扫描文法  $G[P]$  的句子，并作语义处理。(20 分，其中语法扫描 12 分，语义处理 8 分)

$G[P]$   $P \rightarrow D \text{ ';' } S \text{ ';'}$   $D \rightarrow T \text{ id ';' } D \mid \text{空}$   $T \rightarrow \underline{\text{int}} \mid \underline{\text{bool}}$

$S \rightarrow \text{Assign ';' } S \mid \text{空}$   $\text{Assign} \rightarrow \text{id ':' '=' } E$

$E \rightarrow \text{id} + \text{id} \mid \text{id} \text{ and } \text{id}$

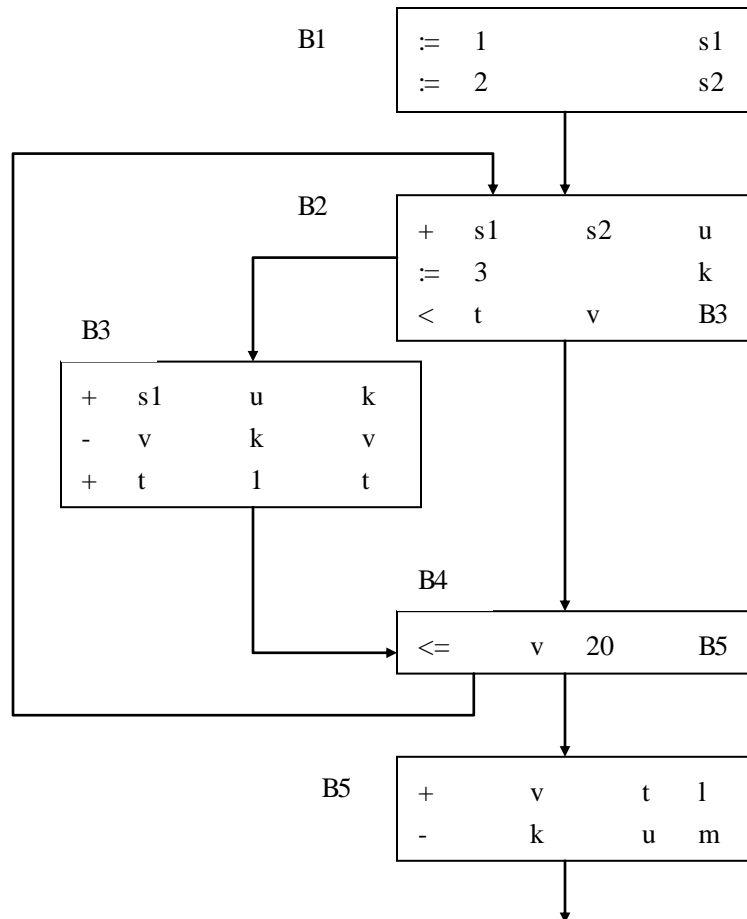
语义处理的要求如下，在句子的申明部分(D)，说明了一些变量，变量的类型可以是 `int` 或者 `bool` 型。语义处理时，必须将变量记录到标识符表中。在处理语句部分的时候进行类型检查：首先+号的两端必须是整数型变量，`and` 的两端必须是 `bool` 型；其次，赋值的时候，两端的类型必须相同。

(使用 `AddIdEntry(id.lexVal, type)` 将 `id` 的类型信息存放到标识符表，使用 `GetIdType(id.lexVal)` 获取标识符的类型。在扫描程序碰到语法/语义错误的时候，可以立刻中止运行，不必作错误恢复。)

三：为下列正则表达式构造 NFA，并进行确定化和最小化。

$1\{10\}^*\{10 \mid 01\}$  (15 分, 构造 NFA 5 分, 确定化 5 分, 最小化 5 分)

四：下面的流图中，循环为 B2,B3,B4 组成了一个自然循环。找出该循环中所有不变四元式 (5 分，不必写出数据流方程)；确定哪些不变四元式不可以外提并简单说明理由 (5 分)；将不变表达式外提，形成优化后的流图(5 分)



一、填空题（如无特殊说明，每空 2 分）

1、编译程序的前端部分包括：词法分析，\_\_\_\_\_，\_\_\_\_\_三个部分。

2、请列出 2 种在基本块内部的优化方法：\_\_\_\_\_，\_\_\_\_\_。

3、在 C 语言程序中，局部变量 `int p` 存放的位置是\_\_\_\_\_；语句 `p=malloc(sizeof(int)10)` 申请得到的空间位于\_\_\_\_\_；全局变量 `int globalIndex` 存放的位置是\_\_\_\_\_；局部变量 `static int si` 的存放位置是\_\_\_\_\_。（以上四个空请填写静态区，栈区，和堆区）

4、给出表达式 `a+b*(c+d)` 的逆波兰表示：\_\_\_\_\_。

5、说明为什么文法  $G[Z]: Z \rightarrow AC \quad A \rightarrow aAc \mid Ac \mid cc \mid ac \quad C \rightarrow cB \mid cC \quad B \rightarrow bB \mid b$  是二义性的\_\_\_\_\_。

\_\_\_\_\_。并给出等价的无二义性的文法：

\_\_\_\_\_（本空 4 分）。

6、给定文法  $G[Z]: Z \rightarrow aZb \mid AB \quad A \rightarrow a \mid aA \quad B \rightarrow bB \mid b$ ，该文法的语言是：\_\_\_\_\_。

\_\_\_\_\_。该语言能否使用正则文法描述？\_\_\_\_\_。

如果能，给出相应的正则文法，如不能，给出理由\_\_\_\_\_。

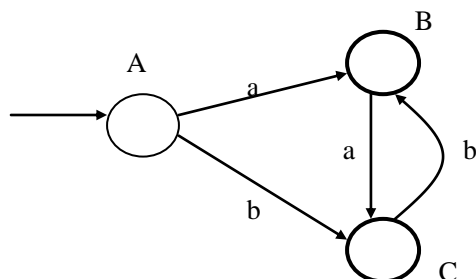
\_\_\_\_\_。

7、给定文法  $G[E]: E \rightarrow T+E \mid T-E \mid T, T \rightarrow \underline{\text{number}}$ ，请给出句子 `2+3-4` 的最左推导过程（假设 `2,3,4` 都是 number）\_\_\_\_\_。

最右推导过程\_\_\_\_\_。

给出语法树(本空 4 分)：

9、给定下图中表示的有穷自动机，请给出相应的正则文法\_\_\_\_\_。  
和正则表达式\_\_\_\_\_。



初始状态 A, 接受状态为 B,C

10、对于下面的算法四元式序列，指出公共子表达式的四元式序号：\_\_\_\_\_。

并在右边给出针对公共子表达式的优化之后的四元式序列(假设 `t1,t2,t3,t4` 不在基本块外使用)。

- (1) + x y t1  
 (2) \* t1 z t2  
 (3) + x y t3  
 (4) + t2 t3 y  
 (5) + x y t4  
 (6) \* t3 t4 m \_\_\_\_\_。

11、给定文法  $G[S]$  的部分规则:  $S \rightarrow \underline{v} \mid \underline{E} \mid \underline{\text{if } B \text{ then } S \text{ else fi}} \mid \underline{\text{if } B \text{ then } S \text{ fi}} \mid \underline{\text{while } B \text{ then } S \text{ end}}$   
 $B \rightarrow B \text{ or } \underline{v} \mid B \text{ and } \underline{v} \mid \underline{v}$  请给出 LR(1) 项集  $\{[S \rightarrow \text{if then } \underline{S} \text{ else fi}, \#]\}$  的闭包 \_\_\_\_\_。

其中大写字母为非终结符号，带下划线的小写字母符号串代表终结符号。该闭包的 if 后继项集闭包是 \_\_\_\_\_。

12、已知文法  $G[Z]$ :  $Z \rightarrow APZ \mid AMZ \mid AMB$   $A \rightarrow a \mid aA$   $P \rightarrow +P \mid -P \mid - \mid +$   $M \rightarrow *M$   
 $B \rightarrow b \mid bB$ , 请对文法进行压缩变换, 得到新的压缩了的文法如下: \_\_\_\_\_。(本空 4 分)

13、给定正则表达式  $a\{b\}c$ , 给出相应的 DFA (本空 6 分)

二、已知用逆波兰方法表示的四则运算表达式的文法如下

$G[E]$   $E \rightarrow EE B \mid EU \mid \underline{\text{number}}$   $B \rightarrow < \mid > \mid + \mid - \mid \underline{\text{and}} \mid \underline{\text{or}}$   
 $U \rightarrow \underline{\text{minus}} \mid \underline{\text{ord}}$

其语义解释如下:

- $<, >$  是比较操作, 其操作数必须是整数型, 结果为 boolean 型,
- $+, -$  为加法和减法操作, 其操作数为整数型, 结果为整数型。
- $\text{and}, \text{or}$  为 boolean 型, 其操作数和结果都是 boolean 型。
- $\text{minus}$  为求负操作, 其操作数和结果都是 boolean 型
- $\text{ord}$  为序数操作, 其操作数为 boolean 型, 结果为 int 型。

请给出进行类型检查的翻译方案。(15 分)

三、请给出 do-while 语句的代码生成的模板, 以及相应的翻译方案。已知其语法如下:

$S \rightarrow \underline{\text{do } S \text{ while } E}$

其中假设其他类型的语句, 和表达式的翻译方案已经生成。(15 分, 如果不使用回填技术扣 5 分)

四、给出下列文法的 LL(1) 分析表。(小写字母串表示终结符号) (10 分)

$E \rightarrow -E \mid (E) \mid \underline{\text{Var}} \underline{\text{Etail}}$   $\underline{\text{Etail}} \rightarrow \underline{\text{lambda}} \mid -E$   $\underline{\text{Var}} \rightarrow \underline{\text{id}} \underline{\text{Vtail}}$   
 $\underline{\text{Vtail}} \rightarrow (E) \mid \underline{\text{lambda}}$

五、填空题（如无特殊说明，每空 2 分）

- 1、编译程序可以分成以下几个部分：\_\_\_\_\_，语法分析，\_\_\_\_\_，代码生成，和优化。
- 2、针对循环的优化技术包括：\_\_\_\_\_，\_\_\_\_\_。（写出两种即可）
- 3、说明四则运算表达式的文法  $E \rightarrow E+E$ ， $E \rightarrow E * E$ ， $E \rightarrow \text{number}$  是二义性的文法（3 分）。  
 \_\_\_\_\_。  
 \_\_\_\_\_。并给出等价的无二义性的文法，其中的+和\*的优先级为**先加后乘**（3 分）：\_\_\_\_\_。
- 4、已知正则表达式  $a\{b|c\}d$ ，给出相应的 DFA：（4 分）  
 \_\_\_\_\_。

以及相应的正则文法：（2 分）  
 \_\_\_\_\_。

- 5、给定下列左递归文法： $T \rightarrow T * F | F$   $F \rightarrow i | (E)$ 。请给出等价的无左递归的文法（4 分）：  
 \_\_\_\_\_。

- a) 设有文法  $G[S]: S \rightarrow a | b | (T)$   $T \rightarrow T, S | S$ （其中(和)为终结符号）。给出对应于规则  $T \rightarrow T, S$  的全部 LR(0)项：\_\_\_\_\_。  
 给出本文法的 LR(1)项集  $\{[T \rightarrow T, \bullet S, ', ']\}$  的闭包：\_\_\_\_\_。

- b) 给定如下的四元式序列，请在右边给出消除公共子表达式之后的四元式序列(4 分)：

```
*   x   y   t1
+   t1  z   t2
*   x   y   t3
*   t2  t3  t4
*   x   y   t5
+   t5  z   x
*   x   y   m
*   m   t4  x
```

- 8、给出语言为  $\{a^n b^j c^i | n \geq 0, j \geq 0, 0 \leq i \leq j\}$  的文法  $G[Z]$ ：\_\_\_\_\_。  
 \_\_\_\_\_。这个语言可以使用正则文法来描述吗？\_\_\_\_\_。

- 9、给出四则运算表达式  $(a+b) * (c+d) / a - b$  的抽象语法树（4 分）：

并给出这个表达式的逆波兰表示方式:\_\_\_\_\_。

10、是否存在能用 SLR(1)技术的处理,但是不能被 LALR(1)技术处理的文法?

11、已知文法  $G[Z]: Z \rightarrow APZ \mid AMZ \mid AMB \mid AP \quad A \rightarrow a \mid aA \quad P \rightarrow +P \mid -P \mid - \mid + \quad M \rightarrow *M \mid B \rightarrow b \mid bB$ , 进行压缩变换后的文法如下: (4分)\_\_\_\_\_。

12、在 C++语言中, static 的类变量的存放位置是:\_\_\_\_\_。假设有类 ClassA, 而 memVar 是 Class 的普通成员变量。假设在程序中有代码 `Class * obj=new Class`, 那么 `obj->memVar` 的存放位置是:\_\_\_\_\_; 假设程序中使用 `ClassA gblobj` 的方式声明全局变量, 那么 `gblobj.memVar` 的存放位置是:\_\_\_\_\_。  
(以上三个空格填写堆区, 栈区和静态区)

13、写出赋值表达式  $x = a*b - c + d*e$  的四元式序列 (4分)

和相应的三元式序列:

二、设有文法  $G[E]: E \rightarrow T + E \mid T - E \mid T \quad T \rightarrow F * T \mid F \quad F \rightarrow (E) \mid i$

1、请使用递归下降分析技术给出扫描该文法句子的程序。必要时先修改文法。(10分)

2、修改上面的程序, 使之能够统计扫描的句子中总共出现了多少对括号。(5分)

三、已知有选择表达式的文法如下:  $Exp \rightarrow (BoolExp ? Exp : Exp)$ ; 其语义如下: 首先对 BoolExp 求值, 如果值为真则对第一个 Exp 进行求值, 其结果为第一个 Exp 的值, 如果 BoolExp 的值为假, 则对第二个 Exp 求值, 表达式的最终结果为第二个 Exp 的值。请给出生成相应代码的翻译方案。(10分, 如果不使用 BackPatch 扣3分)

四、已知源程序如下：

```
int x,y,a,b,c;  
x = a+b*c;  
while(a < b)  
{  
    x = b*c;  
    y = x + a;  
    a = a + 1;  
}
```

- 1、给出相应的四元式序列（5 分）。
- 2、给出相应的流图（5 分）。
- 3、指出其中的循环不变表达式，并外提（5 分）。



## 考试科目名称 编译原理 (A 卷)

2007——2008 学年第 二 学期 教师 赵建华, 戴新宇 考试方式: 开卷: 闭卷

一、填空题 (如无特别指出, 每空 2 分, 本题满分 60 分, **本题直接在卷面上答题**)

1. 编译程序的前端部分包括: 词法分析, \_\_\_\_\_, \_\_\_\_\_ 三个部分。
2. 写出函数定义 `int f(char* a, float f)` 对应的类型表达式: \_\_\_\_\_。
3. 已知上下文无关文法  $G[S]: S \rightarrow aSa \mid aa$ , 这个文法的语言是 \_\_\_\_\_。给出等价的正则文法 \_\_\_\_\_。
4. 已知文法  $S \rightarrow SaB \mid bB$      $A \rightarrow S \mid a$      $B \rightarrow Ac$ 。  $FIRST(SaB) =$  \_\_\_\_\_。  $FOLLOW(A) =$  \_\_\_\_\_。
5. 在 C 语言程序中, 局部变量 `int p` 存放的位置是 \_\_\_\_\_; 语句 `p = malloc(sizeof(int)10)` 申请得到的空间位于 \_\_\_\_\_; 全局变量 `int globalIndex` 存放的位置是 \_\_\_\_\_; 局部变量 `static int si` 的存放位置是 \_\_\_\_\_。(以上四个空请填写静态区, 栈区, 和堆区) (每空 1 分)
6. S-属性的语法制导定义中所有文法符号的属性都是 \_\_\_\_\_ 属性。
7. 对基本块的优化有: \_\_\_\_\_ 和 \_\_\_\_\_ (任写两种), 通常通过 \_\_\_\_\_ 实现基本块的优化。对于下面的算法四元式序列, 指出公共子表达式的四元式序号: \_\_\_\_\_。并在右边给出针对公共子表达式的优化之后的四元式序列(假设 `t1, t2, t3, t4` 不在基本块外使用)。  

(1)	+	x	y	t1
(2)	*	t1	z	t2
(3)	+	x	y	t3
(4)	+	t2	t3	y
(5)	+	x	y	t4
(6)	*	t3	t4	m

\_\_\_\_\_ (本空 4 分)。
8. 将中缀表达式 `a+b*(c-d)` 改写为逆波兰表达式: \_\_\_\_\_。并将表达式计算过程翻译成相应的四元式序列, 结果存放在 `t` 中: \_\_\_\_\_。  
\_\_\_\_\_ (本空 4 分)。  
给出完成同样功能的三元式序列 \_\_\_\_\_。
9. 已知简化后的算术表达式文法  $G[E]: E \rightarrow E * E \mid E + E \mid i$ , 说明这个文法是二义性的 \_\_\_\_\_。

\_\_\_\_\_。按照通常的四则运算规则消除这个文法的二义性，得到文法\_\_\_\_\_。(本空 4 分)。

10. 说明正则表达式  $a\{a|0\}$  的语言：\_\_\_\_\_。  
 \_\_\_\_\_。给出接受这个语言的确定有穷状态自动机：(用图形方式表示，注意指明开始状态和接受状态，本空 4 分)

\_\_\_\_\_。

11. 已知文法的  $E \rightarrow E+T \mid T$   $T \rightarrow T * F \mid F$   $F \rightarrow (E) \mid id$  的 LR(1)项集  $\{[E \rightarrow \cdot T, +]\}$ ，这个项集的闭包  $CLOSURE(\{[E \rightarrow \cdot T, +]\})$  是 \_\_\_\_\_。  
 \_\_\_\_\_。这个项集闭包的  $F$ \_后继项集是：\_\_\_\_\_。  
 \_\_\_\_\_ (不需要再次求闭包)。如果有一个 LR(1)状态对应于这个  $F$ \_后继项集，那么当这个状态位于栈顶，且向前看符号为\_\_\_\_\_时，相应的分析动作为按照规则\_\_\_\_\_进行归约。

得分  二、考虑下面的属性文法 G(D): (8 分)

产生式	语义规则
$D \rightarrow id\ L$	$enter(id.name, L.type)$
$L \rightarrow ,id\ L^{(1)}$	$enter(id.name, L^{(1)}.type)$ $L.type := L^{(1)}.type$
$L \rightarrow :T$	$L.type := T.type$
$T \rightarrow integer$	$T.type := integer$
$T \rightarrow real$	$T.type := real$

过程  $enter(name, type)$  用来把名字  $name$  填入到符号表中，并给出此名字的类型  $type$ 。

- (1) 画出语句  $id_1, id_2, id_3:integer$  的带注释语法树；  
 (2) 文字说明该说明语句的语义。

得分  三、已知选择表达式的文法规则如下： $E \rightarrow E? E:E$ 。这个表达式的含义是：首先求条件表达式（第一个子表达式）的值。如果条件表达式的值为真，那么就求出第二个子表达式的值作为整个表达式的值；如果条件表达式的值为假，就求出第三个子表达式的值作为整个表达式的值。(1) 给出求这个表达式的类型，并进行类型检查的语法制导定义(5 分)。(2) 给出为这个表达式生成代码的翻译方案。要求使用 `backpatch` 技术(不使用 `backpatch`，扣 3 分)。假设 B 有属性 `truelist`, `falselist`。(10 分)

得分	
----	--

四、已知源程序如下：

```
int x,y,a,b,c;  
x = a+b*c;  
repeat  
{   x = b*c;  
    y = x + a;  
    b = b - 1;  
}until (a>= b);
```

4、给出相应的四元式序列（5 分）。

5、给出相应的流图和其中的自然循环（5 分）。

6、指出其中的不变四元式，如果可以外提则外提（5 分）

(A 卷)

考试方式：闭卷

得分

一、填空题：满分 50，每空 2 分（特殊说明除外）

- 和语法分析树: (4 分)

6. 三地址代码序列如下:  $T1 := A + B$ ;  $T2 := 5$ ;  $M := T2 * 4$ ;  $T3 := C - D$ ;  $T4 := M + T3$ ;  $L := T1 * T3$ ;  $T4 := A + B$ ;  $N := T4$  画出该序列的 DAG 图(4 分)

设在该基本块的出口处只有 L、M 和 N 活跃，给出优化后的三地址代码序列(4 分)

7. 已知文法  $G(M): M \rightarrow TB; T \rightarrow Ba \mid \varepsilon; B \rightarrow Db \mid eT \mid \varepsilon; D \rightarrow d \mid \varepsilon$ , 求  $FOLLOW(T)=$ \_\_\_\_\_ ;  $FOLLOW(B)=$ \_\_\_\_\_。
8. 简单说明一个 LR(1)文法的 LALR 分析表中不会存在移入/归约冲突: \_\_\_\_\_。
9. 正则表达式  $(0 \mid 1)^*$  的语言是什么? \_\_\_\_\_  
这个正则表达式和  $((\varepsilon \mid 0)1^*)^*$  等价吗? \_\_\_\_\_ (是/否)。
10. 已知文法  $E \rightarrow E+T \mid T \quad T \rightarrow T * F \mid F \quad F \rightarrow id \mid ( E )$ 。那么句型  $E+T*id$  的句柄是: \_\_\_\_\_。该文法的 LR(0) 项集  $\{[E \rightarrow .E+T]\}$  的闭包是: \_\_\_\_\_。这个闭包的 T 后继的闭包项集是: \_\_\_\_\_。给出这个文法的可行前缀  $E+T*$  对应的任一 LR(0)有效项\_\_\_\_\_。

得分  二: 已知表达式的文法  $E \rightarrow E+T \mid T \quad T \rightarrow T * F \mid F \quad F \rightarrow (E) \mid digit$ , 且 digit 的具体数值可以使用属性 `lexval` 表示; 请给出计算表达式值的语法制导定义 (10 分)

得分  三: (本题满分 10 分) 用正规表达式表示字母表  $\{a, b\}$  上由 a、b 组成, 且不连续出现两个 a 的所有句子的集合, 并给出接受该语言的 DFA。

得分  四、(本题满分 15 分)

首先将  $x=a/(b+c)-d*(e+f)$  转换成三地址指令 (8 分), 然后使用三个寄存器 R1、R2、R3 给出计算这个三地址代码序列的机器代码, 并给出每一步的寄存器描述符和地址描述符。(7 分)  
(在目标机器模型中使用 LD 把数据从内存加载到寄存器, ST 把寄存器的内容保存到内存, OP dst, src1, src2 对 src1 和 src2 中的值进行运算, 并保留到 dst 中; 这些位置都是寄存器, 可以是相同的寄存器。我们假设所有的中间结果都不需要保存)

得分  五、(本题满分 15 分)

下面的翻译方案计算 0 和 1 的串的值 (解释为二进制的正整数)。

$B \rightarrow B_1 0 \{ B.val = B_1.val \times 2 \}$

$B \rightarrow B_1 1 \{ B.val = B_1.val \times 2 + 1 \}$

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

- (1) 消除基础文法中的左递归;
- (2) 为消左递归后的文法写一个计算  $B.val$  的翻译方案。(10 分)

考试科目名称 编译原理 (A 卷)

2010——2011 学年第 二 学期 教师 许畅、陈林 考试方式：开卷：闭卷

得分		一、填空题（本题满分 32 分）
----	--	------------------

1. (本题 4 分) 一个典型的编译程序包括: 词法分析、\_\_\_\_、\_\_\_\_、中间代码生成、机器无关代码优化、\_\_\_\_、机器相关代码优化; 其中“三地址代码生成”属于\_\_\_\_\_。

2.  $L(r) = \{a^n b^{2m+1} \mid n \geq 0, m \geq 0\}$  或 由 0 个或多个 a 连接奇数个 b 组成的串

3. (本题 6 分) 已知文法:

$$E \rightarrow E + T \mid E - T \mid T$$

$$T \rightarrow T * F \mid T / F \mid F$$

$$F \rightarrow (E) \mid \text{id}$$

请给出句子  $id / (id - id)$  的最左推导\_\_\_\_\_

请给出句子  $id / (id - id)$  的语法树:

请给出句型  $T / (F - id)$  的句柄\_\_\_\_\_，以及对该句柄归约所使用的产生式\_\_\_\_\_。

4. (本题 2 分) 语言  $\{a^n cb^m \mid n, m \geq 1\}$  \_\_\_\_\_ 用正则表达式描述, \_\_\_\_\_ 用上下文无关文法描述 (填“可以”或“不可以”)。

5. (本题 8 分) 为文法:

$$S \rightarrow S ; A \mid A$$

$$A \rightarrow i := E$$

$$E \rightarrow E + F \mid F$$

$$F \rightarrow (E) | i$$

设计预测分析表, 则可以求得:

FIRST(A) = {\_\_\_\_\_}; FIRST(E) = {\_\_\_\_\_};

$\text{FOLLOW}(S) = \{ \text{ } \}; \text{FOLLOW}(E) = \{ \text{ } \}.$

6. 3024

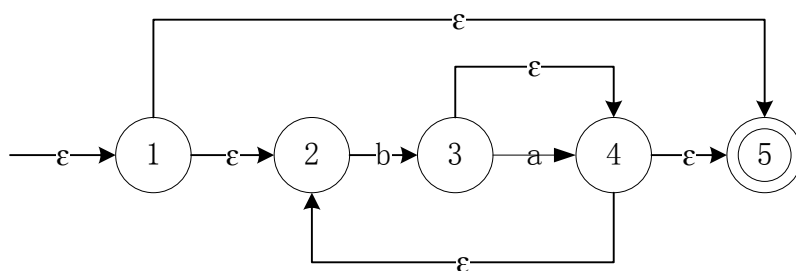
7. (本题 6 分) 在全局机器无关的优化中有哪些常用的手段 (列出三个):

\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_；在数据流分析中有哪些基于数据流方程的具体分析技术（列出三个）：\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。

得分    二、简答题（本题满分 68 分）

1. （本题 9 分）按照课本介绍的方法，为正则表达式 $(ba)^*$ 构造对应的 NFA（只需给出 NFA 图）；并使用子集构造法，将该 NFA 转换为对应的 DFA（需给出 DFA 图以及转换表）。

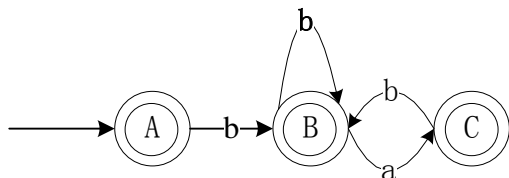
NFA:



状态转化表为:

NFA 状态	DFA 状态	a	b
{1, 2, 5}	A	--	B
{2, 3, 4, 5}	B	C	B
{2, 4, 5}	C	--	B

DFA:



2. （本题 4 分）已知文法:

$$S \rightarrow aA \mid Ab$$

$$A \rightarrow Sc \mid d$$

此文法是否存在左递归？若存在，请消除其左递归（给出结果文法）；否则，说明理由。

3. （本题 5 分）原本无冲突的 LR(1)分析表在合并同心集后得到 LALR(1)分析表，新的表中不可能存在哪种冲突，为什么？

4. （本题 9 分）已知文法

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T F | F$$

$$F \rightarrow F * | (E) | a | b$$

- ① 给出 LR(1)项集  $\{ [F \rightarrow (\cdot E), +] \}$  的闭包;
- ② 给出①中项集闭包相对于文法符号 E 的后继 LR(1)项集闭包。

5. (本题 5 分) 已知文法:

$$E \rightarrow E - T | T$$

$$T \rightarrow \text{num} | \text{num} . \text{num}$$

请给出一个 SDD, 来确定减法表达式的类型。

注: T 有综合属性 type (属性值分为 integer 和 float 两种), E 有综合属性 type。可以使用函数  $\text{getType}(\text{type}_1, \text{type}_2)$  来为减法表达式获取类型。

产生式	语义规则
$E \rightarrow E_1 - T$	$E.\text{type} = \text{getType}(E_1.\text{type}, T.\text{type})$
$E \rightarrow T$	$E.\text{type} = T.\text{type}$
$T \rightarrow \text{num}$	$T.\text{type} = \text{integer}$
$T \rightarrow \text{num} . \text{num}$	$T.\text{type} = \text{float}$

(本题 10 分) 请给出下面 repeat 语句的中间代码翻译方案。已知其语法如下:

$$S \rightarrow \text{repeat } S_1 \text{ while } B$$

假设其他类型的语句和表达式的翻译方案已经生成。

注: S 具有综合属性 nextlist 用来表示跳转指令列表, B 具有综合属性 truelist 和 falselist 来管理布尔表达式跳转代码中的标号, (如果需要) 所插入的额外标记非终结符号具有属性 instr 表示回填的目标标号。必须使用回填方法, 否则不给分。

$S \rightarrow \text{repeat } M S_1 \text{ while } N B$	$\{ \text{backpatch}(B.\text{truelist}, M.\text{instr});$ $\text{backpatch}(S_1.\text{nextlist}, N.\text{instr});$ $S.\text{nextlist} = B.\text{falselist};$ $\text{gen}(\text{'goto' } M.\text{instr}); \}$
---	---

7. (本题 6 分) 下图中给出了各个对象的引用情况, 当从 B 指向 D 的指针被删除后, 各对象的引用计数会发生怎样的变化, 请在表中填写。如果我们采用标记-清扫的方式进行垃圾回收, 哪些对象将会被清扫, 也请在表中标出。

```

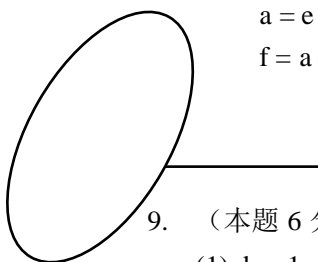
graph TD
    X[X] --> A((A))
    A --> B((B))
    A --> C((C))
    B --> C
    B --> D((D))
    C --> E((E))
    D --> E
    E --> F((F))
    F --> D
  
```

对象	指针删除前的引用计数	指针删除后的引用计数	对象是否被清扫 (清扫的请打√)
A	1	1	
B	1	1	
C	2	2	
D	1	0	√
E	3	1	
F	1	0	√



8. (本题 4 分) 给出下面基本块的 DAG 图 (标出所有被 Killed 的节点)。

$a = b + c$   
 $d[i] = e$   
 $a = e * g$   
 $f = a + e$



9. (本题 6 分) 已知三地址代码片段如下:

- (1)  $b = 1$
- (2)  $n = 10$
- (3)  $d = 1 + n$
- (4)  $c = 2$
- (5)  $t1 = 4 * a$
- (6)  $t2 = 1 + n$
- (7)  $c = c + b$
- (8) if  $c < n$  goto (6)
- (9)  $a = a + b$
- (10) if  $a < n$  goto (4)
- (11)  $c = a - b$

- ① 给出相应的流图 (使用三地址代码前的序号来表示对应的三地址代码), 并指出其中所有的循环。
- ② 指出全局公共子表达式, 并给出其所在的三地址代码的序号。
- ③ 指出每个循环中的可进行强度消减的归纳变量, 并说明如何进行强度消减。

10. (本题 10 分) 将下列三地址语句组成的基本块翻译成机器代码, 并给出每条机器代码执行后的寄存器描述符和地址描述符 (目标指令从 100 开始编号; 假设有三个寄存器可用 R1、R2、R3, 使用课本介绍的 getReg() 分配寄存器; 其中 a、b、c、d 是全局变量; t、u 是临时变量, 不需要保存)。

$t = a * b$   
 $u = a - c$   
 $d = t + u$

附机器代码格式:

取数指令: LD R, var // 将内存空间中变量 var 的值存入寄存器 R 中。

存数指令: ST var, R // 将寄存器 R 中的值存入变量 var 的内存空间中。

双目运算指令: OP dst, src1, src2 // 使用 OP 对寄存器 src1 和 src2 中的值进行计算, 结果存入寄存器 dst 中 (OP 包括 ADD、SUB、MUL、DIV)。