

计算机科学与技术学院 2019-2020 学年第 2 学期 考试试卷

编译原理 A 卷 开卷

姓名_____ 班级_____ 学号_____ 考试日期 2020-07-05 19:00—21:30

题号	一	二	三	四	五	六	七	八	九	总分	核对
题分	10	10	10	14	15	8	9	10	14	100	
得分											

一、请简述编译过程，并说明哪些过程可以组合到一起，有什么好处？(共 10 分)

二、对于文法 $G(S)$:

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

$L \rightarrow L, S \mid S$

请给出句型 $(S, (a))$ 的语法树、最左推导、最右推导和句柄。(共 10 分)

三、请为正规式 $0((0|1)^*|01^*0)^*1$ 构造 NFA 并确定化和最小化(共 10 分)

四、自顶向下的语法分析综合题(共 14 分)

1. 自上而下的分析的第一步如何开始？预测分析法是如何实现的？在选用推导用的产生式前，如何避免回溯现象发生。(4 分)

2. 对于给定文法 $G[V]$: $V \rightarrow TL \mid T \mid L$, $L \rightarrow L, d \mid d$

(1) 文法 G 是 LL(1) 文法吗？如果不是请将其改写为 LL(1) 文法 (4 分)；

(2) 给出改写后 LL(1) 文法的预测分析表 (4 分)。

(3) 该文法会产生什么类型的句子？试举例说明 (2 分)。

五、自底向上的 LR 分析 (共 15 分)

1. LR(k) 分析称为状态分析法，根据分析的态势确定下一步动作，请简述可能的动作类型 (2 分)。

2. 分析过程中，符号栈和输入栈的栈顶连接后会形成一个串，如果输入串是文法的句子，那么这个串是什么？请证明你的结论成立。(3 分)

3. 有如下文法 $G[P]$:

$P \rightarrow DA$

$D \rightarrow d; D \mid d$

$A \rightarrow s; A \mid s \mid e$

(1) 请构造识别该文法活前缀的 DFA；(4 分)

(2) 判断该文法的类型；(2 分)

(3) 构造该文法的 LR 分析表。(2 分)

(4) 给出串 “d;se;es” 的分析过程。(2 分)

六、语法符号的语义计算(共 8 分)

1. 在属性文法定义中，文法的符号的属性是客观固有的，还是主观人为设计的？试举例说明。(3)

2. 属性是如何分类的？S 属性文法和 L 属性文法在不同的语法分析中如何计算属性？(3)

3. 你在编译原理实验中是如何实现属性计算功能的。(2)

七、静态语义计算(9 分)

写出条件赋值语句 $i := \text{if } B \text{ then } E1 \text{ else } E2$ 的语义子程序。其中 B 是布尔表达式， $E1$ 和 $E2$ 是算术表达式， i 代表与 $E1$ 、 $E2$ 类型相同的左部变量。按写出的语义子程序生成赋值语句

Z:=if A>C then x+y else x-y+0.5

八、运行时存储组织(10分)

阅读下列代码，并回答下列问题：

```
#include <stdio.h>

void Sort(int array[],int begin,int end);
int Level;
//main AR
int main()
{
    int i;
    int testArray[10] = {9,5,6,10,4,7,3,2,8,1};
    Level = 0;
    printf("Level=");
    Sort(testArray,0,9);
    printf("\n");
    printf("testArray=");
    for(i=0;i<=9;i++)
    {
        printf("%d ",testArray[i]);
    }
    return 1;
}

//main AR
void Sort(int array[],int begin,int end)
{
    int* pArray=array;
    void exchange(int i,int j)
    {
        int temp;
        temp = pArray[i];
        pArray[i]=pArray[j];
        pArray[j] = temp;
    }
    void quickSort(int begin,int end)
    {
        int pivotIndex;
        void partition(int begin,int end)
        {
            int pivot;
            int i;
            pivot= pArray[begin];
            pivotIndex = begin;
            for(i=begin+1;i<=end;i++)
            {
```

```

        if(pArray[i]<pivot)
        {
            pivotIndex++;
            if(i!=pivotIndex)
            {
                exchange(i,pivotIndex);
            }
        }
        exchange(pivotIndex,begin);
    }
    Level++;
    if(begin>=end)
    {
        printf("%d ",Level);
        Level--;
        return;
    }
    partition(begin,end);
    if(pivotIndex>begin)
        quickSort(begin,pivotIndex-1);
    if(pivotIndex<end)
        quickSort(pivotIndex+1,end);
    printf("%d ",Level);
    Level--;
    return;
}
quickSort(begin,end);
}

```

(1) 完成屏幕输出(2 分)

Level= 6 _____
 testArray= _____

(2) 假设在两处注释（//main AR）之间的代码所对应的汇编代码(IA32 X86 指令体系)如下，根据该汇编代码写出 main 栈帧的内容(4 分)

00401460 <_main>:

```

    push    %ebp
    mov     %esp,%ebp
    and     $0xffffffff0,%esp
    sub     $0x40,%esp
    movl    $0x9,0x14(%esp) //将 0x9 存入地址 0x14(%esp)，4 字节
    movl    $0x5,0x18(%esp)
    movl    $0x6,0x1c(%esp)
    movl    $0xa,0x20(%esp)

```

```

movl    $0x4,0x24(%esp)
movl    $0x7,0x28(%esp)
movl    $0x3,0x2c(%esp)
movl    $0x2,0x30(%esp)
movl    $0x8,0x34(%esp)
movl    $0x1,0x38(%esp)
movl    $0x0,0x407070    //全局变量 Level=0
movl    $0x405064,(%esp)//字符串"Level="的偏址压栈
call    403d08 <_printf>
movl    $0x9,0x8(%esp)
movl    $0x0,0x4(%esp)
lea     0x14(%esp),%eax //将%esp+0x14 处的变量的地址存入寄存器%eax 中
mov     %eax,(%esp)
call    401592 <_Sort>
movl    $0xa,(%esp)      //0xa 换行符的 ASCII 码
call    403d00 <_putchar>
movl    $0x40506b,(%esp) //字符串"testArray="的偏址压栈
call    403d08 <_printf>
movl    $0x0,0x3c(%esp)
jmp     40152f <_main+0xc1>
401512: mov     0x3c(%esp),%eax
mov     0x14(%esp,%eax,4),%eax//0x14(%esp,%eax,4)比例变址寻址方式
//地址是 0x14+%esp+4*%eax
mov     %eax,0x4(%esp)
movl    $0x405076,(%esp) //字符串"%d"的偏址压栈
call    403d08 <_printf>
addl    $0x1,0x3c(%esp)
40152f: cmpl    $0x9,0x3c(%esp)
jle     401512 <_main+0xb2>
mov     $0x1,%eax
leave
ret

```

(3) 当屏幕输出“Level= 6 ”时，写出栈内的内容（以 main 栈帧的栈基址为起始点）(4 分)

九、中间代码和优化(14 分)

3 阅读下列代码，并回答下列问题:

```

#include <stdio.h>
#include <math.h>
void main()
{
    int N;
    int A[21];
    int k;
    int i;
    int j;

```

```

N=20;
k=sqrt(N);
for( i=0;i<=N;i++)
{
    A[i]=1;
}
for( i=2;i<=k;i++)
{
    if(A[i])
    {
        for(j=2*i;j<N;j+=i)
        {
            A[j]=0;
        }
    }
}
for(i=2 ;i<N;i++)
{
    if(A[i])
        printf("%d ",i); //中间代码用 write i 表示
}
}

```

- (1) 试求出上述程序在屏幕的输出结果。(2分)
- (2) 按照如下形式，写出其四元式中间代码,其中 100,101,102 是语句标号；(4分)


```

100: N:=20;
101: K:=sqrt(N);
102: .....
            
```
- (4) 做出流图并求出其中的循环；(4分)
- (5) 进行代码外提；(2分)
- (6) 进行强度削弱和删除归纳变量。(2分)