Southeast University Examination Paper (A)

Course Name Principles of Compiling Examination Term 16-17-2 Score

Related Major Software Engineering Examination Form Close test Test Duration 150 Mins

There are 8 problems in this paper. You can write the answers in English or Chinese on the attached paper sheets.

1. Please construct context-free grammars **with ε-free productions** for the following language (10%).

   L={ω| ω is a non-negative palindrome integer which is read the same backward as forward, and ω is an even number}

2. Please construct a **DFA with minimum states** for the following regular expression. (15%)

   ((a|b)*|(ab)*)*(ab)(a|b)

3. Please **eliminate the left recursions (if there are)** and **extract maximum common left factors (if there are)** from the following context free grammar, and then decide **the resulted grammar** is whether a LL(1) grammar by **constructing the related LL(1) parsing table.**(15%)

   A→Ba|Aa|c

Surname_____

Student No._____

Please obey the rules of examination. If you violate the rules, your answer sheets will be invalid

B→Bb|Ab|d

4. Please **construct a LR(1) parsing table for the following ambiguous grammar with the additional conditions(You determine the required additional conditions by yourself)**.(15%)

S→if C then S| if C then S else S|A

C→ b

A→ d=E

E→ E+E|E*E|i

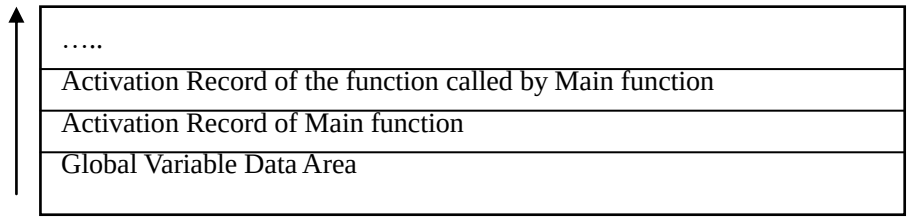
5. Please construct **an annotated parse tree** for the input string 4+@(5*@6+7)+8*9 where the syntax-directed definition is as following (10%):

Productions	Semantic Rules
$E \rightarrow E_1 * T$	$E.val = E_1.val * T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T_1 + F$	$T.val = T_1.val + F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow (E)$	$F.val = E.val$
$F \rightarrow @F_1$	$F.val = 0 - F_1.val$

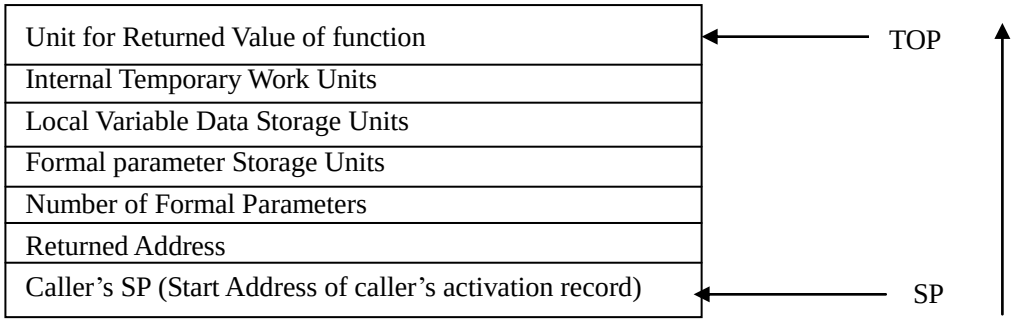
F→i F.val=i.**lexval**

6.We assume that the storage organization and the form of activation record used in C language program run-time stack storage allocation are as following. Please **construct the run-time stack map when it gets the maximum size** for the following C program (10%).

Storage Organization of C Language



The Form of Activation Record of any a function in C




```
#include <stdio.h>
int x,y;

int main()
{
   x=10;
   y=f(x);
}
```

```
int f(int n)
{
   if (n<=1)
         return 1;
   else if(n==2)
          return 2;
         if(n==3)
          return 3;
        else
         {
          int t1,t2,t3,t4,t5,t6,t;
          t1=f(n-1);
          t2=f(n-2);
          t3=f(n-3);
          t4=f(n-4);
           t5=t1+t2;
           t6=t3+t4;
           t=t5+t6;
          return t
         }
}
```

Notes: 1) Here we assume that the caller's sp of Main function is the start address of global variable data area, and the returned address in the activation record of a function (including Main function) is filled by the operating system automatically, you might not care it.

2) The initial value of variable X is 10, the start address of stack used in the program is K.

3) The stack map may get its maximum size for several times, here we ask you draw the stack map at maximum size for the second time.

7. Please translate the following program fragment into **three-address code sequence using short circuit code and back-patching techniques**. (15%)

```
i=1;
while (i<=10) {
   j=1;
   while (j<=10) {
```

```
        c[i,j]=0;
        j=j+1
    }
    i=i+1;
}
i=1;
while (i<=10) {
    j=1;
    while (j<=10) {
        k=1;
        while (k<=10) {
            if (a[i,k]!=0 && b[k,j]!=0)
                c[i,j]=c[i,j]+a[i,k]*b[k,j];
            k=k+1;
      }
      j=j+1;
  }
i=i+1;
}
```

Notes: Here we assume that the declarations of array A,B,C are array [1..10,1..10], each data element of array A,B,C would **use 4 storage unit,** and the start address of array A's storage area is addrA, the start address of array B's storage area is addrB, the start address of array C's storage area is addrC.

8. Please **construct the DAG** for the following basic block, and we assume that only variable "M" is live on exit, please optimize the block and **rewrite the block** in optimized code form.(10%)

C=A+B

D=A-B

E=A*C

D=D*E

F=A+B

F=F*E

G=D*E

M=F*G