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

Name_____

Student No._____

Southeast University Examination Paper (A)

Course Name Principles of Compiling Examination Term 22-23-2 Score _____

Related Major Computer Science & Engineering Examination Form Closed Test Duration 120 Mins

There are 10 problems in this exam. You can write the answers in English or Chinese (Must answer in English for the English class).

1. Consider the following context free grammar:

$S \rightarrow S\ ;\ D \mid D$

$D \rightarrow b := E$

$E \rightarrow E \oplus F \mid F$

$F \rightarrow (\ E\ ) \mid b$

where S is the start symbol. Construct a parsing tree for the string b:=b⊕b;b:=b. (5%)

2. The following grammar generates regular expression over symbols a and b only, using + in place of | for union, to avoid conflict with the use of vertical bar as a metasymbol in grammars.

$R \rightarrow R+R \mid RR \mid R^* \mid (R) \mid a \mid b$

Is the grammar ambiguous? Show your justification. If it is ambiguous, rewrite an unambiguous one. (5%)

3. Give the following regular expression:

$(a|\varepsilon)(b|ab)^*$. (15%)

(a) Construct the NFA for the regular expression.

(b) Convert the NFA to a DFA with minimum states.

(c) Show the context free grammar using the generated minimum DFA.

4. Consider the following grammar:

$S \rightarrow \{L\} \mid i$

$L \rightarrow L, L \mid S$

where S, L are nonterminals, and S is the start symbol.(15%)

(a) Eliminate left recursion.

(b) Compute FIRST and FOLLOW sets for the resulting grammar.

(c) Is the resulting grammar LL(1)? If it is, construct LL(1) parsing table. Otherwise, show your justification.

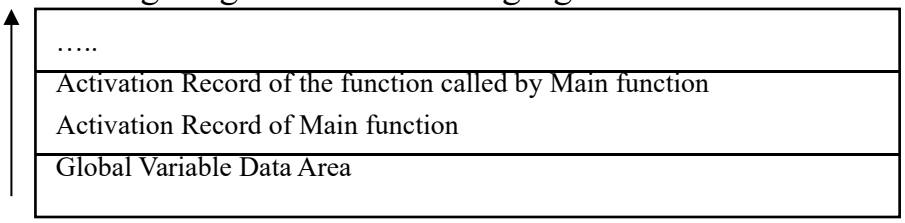
5. Construct an LR(1) parsing table for the following grammar. Is the grammar LR(1)? (10%)

L→MLb | a

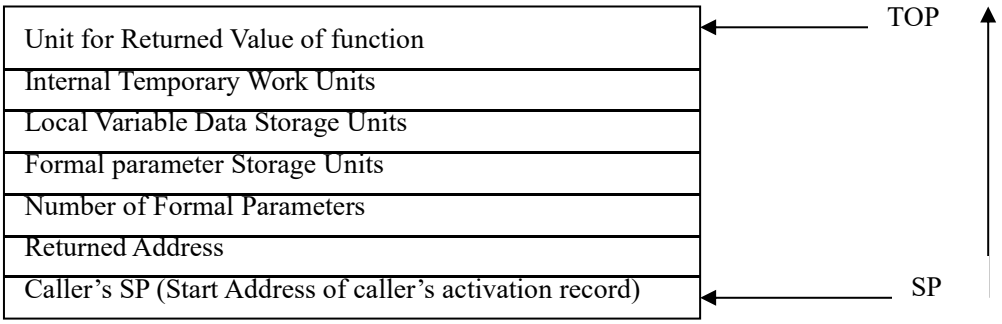
M→ε

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 activation tree, and the run-time stack map when it calls f(1) for the third time for the following C program (10%).

Storage Organization of C Language



The Form of Activation Record of any function in C



#include <stdio.h>

```
int x;

int main()
{
   x=5;
   y=f(x);
}

int f(int n){
    int t1, t2, t3;
    if(n<2) return 1;
    t1 = f(n-1);
    t2= f(n-2);
    t3=f(n-3)
    return t1+t2+t3;
}
```

Notes: 1) Here it is assumed 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, so you might not care about it.

2) The start address of stack used in the program is K.

7. Please construct an annotated parse tree for the input string (1+@2*3+@4)+@5*6 where the syntax-directed definition is as following. (10%)

Productions	Semantic Rules
E→E1+F	{E.val=E1.val+F.val}
E→F	{E.val=F.val}
F→F1*G	{F.val=F1.val*G.val}

F→G	{F.val=G.val}
G→(E)	{G.val=E.val}
G→i	{G.val=i.lexval}
G→@i	{G.val=0-i.lexval}

8. Consider the following grammar

S′→S

S →(L)| b

L →L, S | S

where S′ is the start symbol. Design an SDD to compute the nesting depth of each b in parentheses. For example, the depth of each b in (b, (b, b)) is 1, 2 and 2 respectively. (5%)

9. Translate the following program fragment into three-address-code (TAC) sequence using short circuit code and back-patching techniques. Assume that *a* is a 10*10 array of integers, and the start address of array a is addr_a. (15%)

```
i=1;
while (i<=10)
    { if (x>0&&x<100)
            x=x+a[i][i];
        else
            y=y+1;
      i=i+1;
    }
```

10. Generate a quadruple sequence for the following expression,

and optimize and rewrite it using DAG techniques. Assume only Variable M would be used later (10%)

B=3

D=A+C

E=A*C

F=D+E

G=B*F

H=A+C

I=A*C

J=H+I

K=B*5

L=K+J

M=L