

Southeast University Examination Paper (A)

Course Name Principles of Compiling Examination Term 08-09-2 Score _____
Related Major Software Engineering Examination Form Close test Test Duration 150 Mins

There are 9 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%).

$\{\omega \mid \omega \in (a,b,c,d)^* \text{ and the numbers of } a\text{'s and } b\text{'s and } c\text{'s occurred in } \omega \text{ are even, and } \omega \text{ starts with } b \text{ or } c, \text{ ends with } a\}$

2. Please construct a **DFA with minimum states** for the following regular expression. (10%)

$(a|(a|(a|b^*))^*)^*(a|b^*)$

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%)

$P \rightarrow b S d$

$S \rightarrow S ; A | A$

$A \rightarrow B | C$

$B \rightarrow a$

$C \rightarrow D | D \text{ e } A$

$D \rightarrow E \text{ B}$

$E \rightarrow i \text{ F t}$

$F \rightarrow F \text{ o } G | G$

$G \rightarrow b$

4. Please show that the following operator grammar is whether an operator precedence grammar by **constructing the related parsing table**. (10%)

$E \rightarrow E \text{ a } F | F$

$F \rightarrow F \text{ o } T | T$

$T \rightarrow (E) | n \text{ T} | b$

5. Please **construct a LR(1) parsing table for the following ambiguous grammar with the additional conditions** that $*$, \otimes and \oplus have the properties of left associative law, and $*$ has higher precedence than \otimes , \otimes has higher precedence than

\oplus .(15%)

$E \rightarrow E \oplus E | E \otimes E | E^* | (E) | a | b$

6. Please construct **an annotated parse tree** for the input string 123.123 where the syntax-directed definition is as following (10%):

Productions

Semantic Rules

$S \rightarrow L^{(1)}.L^{(2)}$

$S.val = L^{(1)}.val + L^{(2)}.val / 4^{L^{(2)}.len}$

$S \rightarrow L$

$S.val = L.val$

$L \rightarrow L^{(1)}B$

$L.val = L^{(1)}.val * 4 + B.val, L.len = L^{(1)}.len + 1$

$L \rightarrow B$

$L.val = B.val, L.len = 1$

$B \rightarrow 0$

$B.val = 0$

$B \rightarrow 1$

$B.val = 1$

$B \rightarrow 2$

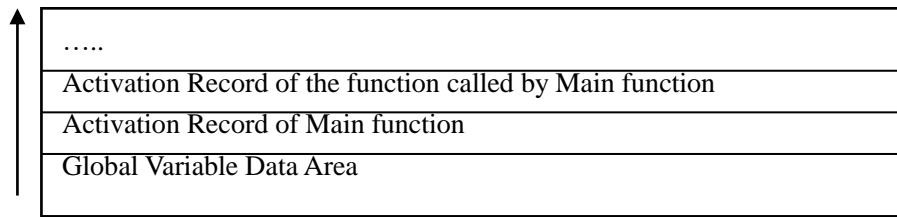
$B.val = 2$

$B \rightarrow 3$

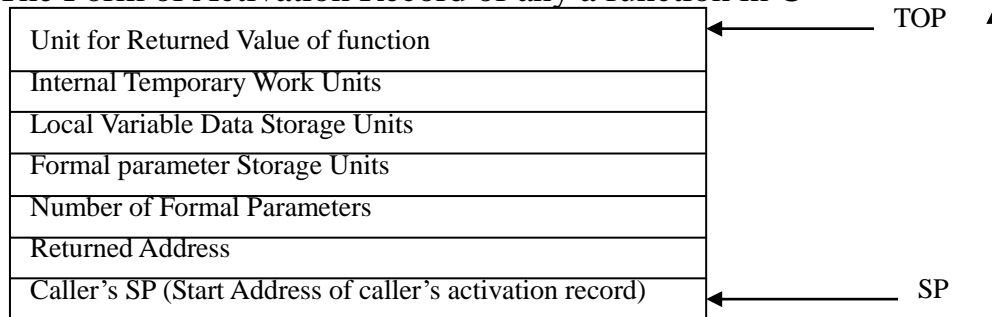
$B.val = 3$

7. 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 at the third time** for the following C program (10%).

Storage Organization of C Language



The Form of Activation Record of any a function in C



The C program is as the following:

```
#include <stdio.h>
```

```
int x,y;
```

```
int main()
{
    x=6;
    y=f(x);
}
```

```
int f(int n)
{
    if (n<=1)
        return 1;
    else if(n==2)
        return 2;
    else
    {
        int t1,t2,t3,t;
        t1=f(n-1);
        t2=f(n-2);
```

```

        t3=f(n-3);
        t=t1+t2;
        t=t+t3;
        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 6, 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 third time.

8. Please translate the following program fragment into **quadruple sequence**. (10%)

```

i=1;
m=0;
loop=0;
n=0;
while (loop==0 && i<=10) {
    j=1;
    while (loop ==0 && j<=i)
        if (a[i,j] != a[j,i])
        {
            loop=1;
            m=i;
            n=j;
        }
    else j=j+1;
    if (loop==0) i=i+1;
}

```

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

9. Please **construct the DAG** for the following basic block. We assume that only **variable P be used later**, please optimize the block and **rewrite the block** in optimized code form.(10%)

$E = A + B$

$F = E - C$

$G = F * D$

$H = A + B$

$I = H - C$

$L = I + G$

$M = I * I$

$M = 2 * M$

$N = L + M$

$P = N + M$