浙江大学 2023－2024 学年春夏学期

《 编译原理》课程期末考试试卷

课程号： 21120471，开课学院：计算机科学与技术学院

考试试卷：√A 卷、B 卷（请在选定项上打√）

考试形式：闭、√开卷（请在选定项上打√），允许带三张 A4 纸入场

考试日期： 2024 年 6 月 25 日,考试时间： 120 分钟

诚信考试，沉着应考，杜绝违纪。

考生姓名：______________学号：______________所属院系：

任课教师：_____________

题序	一	二	三 (1)	三 (2)	三 (3)	三 (4)	三 (5)	三 (6)	三 (7)	三 (8)	总 分
得分											
评卷人											

一、 **Mark each statement *true* or *false***

1.[T] [F]	**8.[T] [F]**	**15. [T] [F]**
2.[T] [F]	**9.[T] [F]**	**16. [T] [F]**
3.[T] [F]	**10.[T] [F]**	**17. [T] [F]**
4.[T] [F]	**11.[T] [F]**	**18. [T] [F]**
5.[T] [F]	**12.[T] [F]**	**19. [T] [F]**
6.[T] [F]	**13.[T] [F]**	**20. [T] [F]**
7.[T] [F]	**14.[T] [F]**	

二、 **Single Choice**

1．[A] [B] [C] [D]	**8．[A] [B] [C] [D]**	**15．[A] [B] [C] [D]**
2．[A] [B] [C] [D]	**9．[A] [B] [C] [D]**	**16．[A] [B] [C] [D]**
3．[A] [B] [C] [D]	**10．[A] [B] [C] [D]**	**17．[A] [B] [C] [D]**
4．[A] [B] [C] [D]	**11．[A] [B] [C] [D]**	**18．[A] [B] [C] [D]**

5. [A] [B] [C] [D]　　12. [A] [B] [C] [D]　　19. [A] [B] [C] [D]

6. [A] [B] [C] [D]　　13. [A] [B] [C] [D]　　20. [A] [B] [C] [D]

7. [A] [B] [C] [D]　　14. [A] [B] [C] [D]

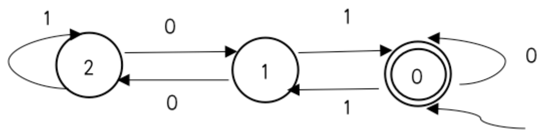
一、**Mark each statement *true* or *false* (1 points each， 20 cents)**

1. When using the subset construction method, in the worst case, an n-state NFA will be converted into a 2^n-state DFA .
2. The regular expression *"/*" "/"* ([^*/]|[^*]"/"|"*"[^/])* "*"* "*/"* can accept all C-style comments.
3. The grammar S → (S)S|ε is LL(1) .
4. Imperative-style environments is usually not considered to be implemented using hash tables.
5. A static link of a function’ s stack frame points to the its caller’ s stack frame.
6. Escaping variables must be kept in the registers.
7. A display is an array for maintaining frame pointers.
8. Lexical analysis, syntax analysis, and semantic analysis are tasks that should be completed by the frontend of a compiler.
9. The task of instruction selection is to tile the IR tree with a minimal set of tree patterns.
10. Using Maximal Munch algorithm, the optimum tiling can be obtained.
11. The liveness of variables should be analyzed from future to past.
12. The solution to the dataflow equation is a conservative approximation.
13. Via tree canonicalization, one can eliminate all the CALL instructions in the IR tree.
14. A node ***d*** dominates a node ***n*** if some path of directed edges from start node s_0 to ***n*** goes through ***d***.
15. MOVE(MEM(x), y) commutes with MEM(z) if x != z .
16. The general rewriting rule rewrites [e1, e2, ESEQ(s, e3)] into (SEQ(MOVE(t1, e1), s); [TEMP(t1), e2, e3]) if s commutes with e1, yet not with e2 .
17. For graph coloring with coalescing, if neither simplify nor coalesce applies, we should first select a node for potential spilling.
18. In an interference graph, if there is a MOVE between a and b, and for every neighbor t of b, either t already interferes with a or t is of insignificant degree, then a and b can be coalesced.
19. The records whose reference counts are not zero will not be reclaimed by the mark-and-sweep collection.
20. Using breadth-first copying collection instead of depth-first copying can

lead to the increase of cache misses.

二、 **Single Choice (1.5 points each， 30 cents)**

1. Which language can be recognized by the DFA in the graph?



[A]. String of binary numbers that can be divided by 2
[B]. String of binary numbers that can be divided by 3
[C]. String of binary numbers that can be divided by 5
[D]. String of binary numbers that can be divided by 7

2. Here is the grammar:

exp → exp addop term | term
addop → + | -
term → term mulop factor | factor
mulop → *
factor → (exp) | number

which choice is the First(factor) and Follow(factor) ($ is the *EOF*)?
[A].First(factor) = {(, number}, Follow(factor) = {+, -, *,),ε}
[B].First(factor) = {(, number, ε }, Follow(factor)={$, +, -, *,) }
[C].First(factor) = {number}, Follow(factor) = {+, -, *,)}
[D].First(factor) = {(, number}, Follow(factor) = {$, +, -, *,)}

3. Here is the LR(0) parsing table:

State	Input			Goto
	(	)	$	S
0	S2	R($S\to\varepsilon$)	R($S\to\varepsilon$)	1
1			acc	
2	S2	R($S\to\varepsilon$)	R($S\to\varepsilon$)	3
3		S4		
4	S2	R($S\to\varepsilon$)	R($S\to\varepsilon$)	5
5		R($S\to(S)S$)	R($S\to(S)S$)	

And here is the current parsing stack:

Stack	Input
#0(2S3)4S5	$

What is the next symbol on the top of the stack?
[A].S5 [B].S3 [C].S2 [D].S1

4. When using YACC, the value of ***pos*** in the declaration : ***exp : exp PLUS pos exp*** stores in:
[A]. $1 [B]. $2 [C]. $3 [D]. $4

5. Which action is not in a LR parsing table?

[A]. Match [B]. Reduce [C]. Accept [D]. Shift

6. To access a local frame-resident variable of the current function, we should find its offset from ________.
[A].The stack pointer [B]. The static link
[C].The beginning of the program [D]. The frame pointer

7. If we want to access the formal parameter of an enclosed function, _____ is used to find the activation record of the enclosed function.
[A]. Static link. [B]. Program counter .
[C]. Stack pointer [D]. Argument pointer.

8. Which of the following is commonly found in a stack frame (activation record)?
[A]. Static variable [B]. Sp pointer
[C]. Saved fp pointer [D]. Global variable

9. For symbol table of functional style, ______ is the most efficient data structure among the following structures.
[A]. Octree [B]. Hash table [C]. Binary search tree [D]. Array

10. Tiling the IR tree is the fundamental idea of instruction selection. ______ is not directly related with tiling.
[A].Tree pattern [B]. Maximal Munch
[C]. Dynamic Programming [D]. Register allocation

11. Which of the method does not be used by the dynamic-programming algorithm?
[A].Bottom-up [B]. Minimum-cost match [C]. Optimum solution [D]. Tiny tiles

12.The Tree Grammars can be used in ______ .
[A]. Fast matching [B]. Optimal tiling
[C]. Code-generator generator [D]. Maximal Munch

13. The ______ is not related to the liveness information about a node N.
[A]. Def[N] [B]. Use[N] [C]. Pred[N] [D]. Succ[N];

14. In the following descriptions about interference graphics,_____ is incorrect.
[A]. It can be expressed as a matrix.
[B]. It is only used for register allocation.
[C]. It is caused by overlapping live ranges.
[D]. It is conservative approximation.

15. Which of following choice is a valid canonical tree?
[A]. ESEQ(MOVE(MEM(TEMP fp), 10), MEM(TEMP fp))
[B]. MOVE(MEM(CALL(f, args), TEMP a))
[C]. SEQ(MOVE(TEMP r, 1), CJUMP(<, TEMP r, 0, t,f))

[D]. MOVE(TEMP a, CALL(g, args))

16. How many basic blocks should be following IR program be divided into?

```
MOV(TEMP x,MEM(TEMP fp))
CJUMP(<,TEMP x,5,t,f)
LABEL t
MOVE(TEMP r, 1)
CJUMP(>,a,b,t1,f1)
LABEL(f1)
MOVE(TEMP r, 0)
LABEL(t1)
JUMP join
LABEL f
MOV(TEMP r, 0)
JUMP join
LABEL join
RET r
```

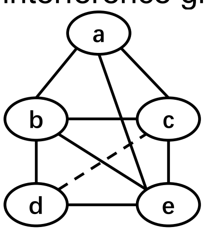
[A]. 5 [B]. 6 [C]. 7 [D]. 8

17. Among the following transformations of IR trees, which one is wrong?
[A]. ESEQ(s1, ESEQ(s2, e)) => ESEQ(SEQ(s1, s2), e)
[B]. JUMP(ESEQ(s,e1)) => ESEQ(s, JUMP(e1))
[C]. BINOP(op, ESEQ(s, e1), e2) => ESEQ(s, BINOP(op, e1, e2))
[D]. CJUMP(op, CONST 1897, ESEQ(s, e), l1, l2) => SEQ(s, CJUMP(op, CONST 1897, e, l1, l2))

18. A definition $d \leftarrow a1 \oplus a2$ may not be loop-invariant within a loop L if:
[A]. a_i are constants.
[B]. All the definitions ai that reaches d are outside L.
[C]. Only one definition of ai reaches d, and that definition is loop-invariant.
[D]. d is not used in L.

19. ___ does not suffer from external fragmentation?
[A]. Mark-and-Sweep Collection [B]. Reference Counts
[C]. Copying Collection [D]. Neither of the above

20. Suppose the target machine has 3 registers. To color the following interference graph, under what criterion can c and d be coalesced?



[A]. Briggs criterion [B]. George criterion
[C]. Both of the above [D]. Neither of the above

三、**Questions (50 cents)**

1. Draw the DFA for the number.（4 points）

nat = [0 − 9]+

signedNat = ("+"|"−")? nat

number = ***signedNat ("." nat)? (E signedNat)?***

ANSWER:

2. Given the follow grammer. (8 points)

0 A’ →A$
1 A →(A)
2 A → a

1). Construct the LALR(1) DFA for the grammar
2). Construct the LALR(1) parsing table
ANSWER:
1)

2)

3. Given a Tiger program: (6 points)

```
let
   function f(): int = let
            var a := 12
            function g(y: int): int = let
               b := 8
               function gcd(): int =
                              if a = 0 then b + y
                                    else if b = 0 then a + y
                                    else if a > b then gcd()
                                    else gcd()
```

```
                    in  gcd();  end
            in  g(10);  end
    in  f();  end
```

Function ***f()*** is at depth 1. Element ***Di*** of the display always points to the most recently called function whose static nesting depth is ***i*** .
1). Show the static links of activation records when the function ***gcd()*** is called a second time.
2). When using display instead of static links, show the pointers in the display when the function ***gcd()*** is called a second time.
3). Compare the instructions to access ***a*** in ***gcd()*** when using static links and display, where ***a*** is a local variable of function ***f()***.

ANSWER:

4. Given the following IR,

```
type Exp =
        CONST of int
      | BINOP of BinOp * Exp * Exp
      | Var of string
type Stm =
        MOVE of Exp * Exp
      | EXP of Exp
      | JUMP of string
```

```
      | CJUMP of RelOp * Exp * Exp * string * string
      | Label of string
type BinOp = PLUS | MINUS | MUL | DIV
type RelOp = EQ | NE | LT | GT | LE | GE
```

Please translate the following expression into an IR. (6 points)

```
if (a<b and c>d):
    x := y + z
else:
    x := y – z
```

For example:

```
if (b == 0):
   a = c + 3

CJUMP(EQ, Var('b'), CONST(0), 'THEN', 'IF_END')
LABEL('THEN')
MOVE(Var('a'), BINOP(PLUS, Var('c'), CONST(3)))
LABEL('IF_END')
```

ANSWER:

5. Consider the following program: (6 points)

```
L1:  t ← n % 2
     if t == 0 goto L2
     res ← res ∗ base
L2:  base ← base ∗ base
     n ← n/2
     if n > 0 goto L1
     return res
```

1) Break the program into basic blocks;

2) Determine the longest trace (one trace suffices in case of multiple longest traces).

ANSWER:

6. Perform flow analysis on the following program. (6 points)

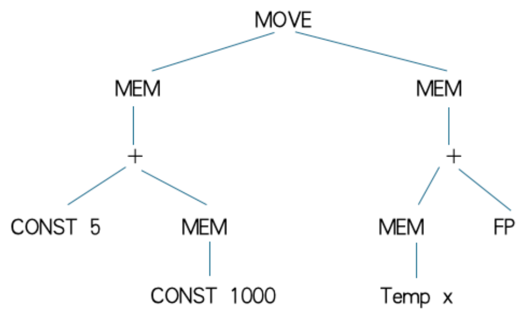
```
        s ← 0
        i ← 0
L1: if i > n goto L2
        j ← i*4
        k ←j+a
        x ← M[k]
        s ← s+x
        i ← i+1
        goto L1
L2:
```

1)Calculate live-in and live-out at each statement;

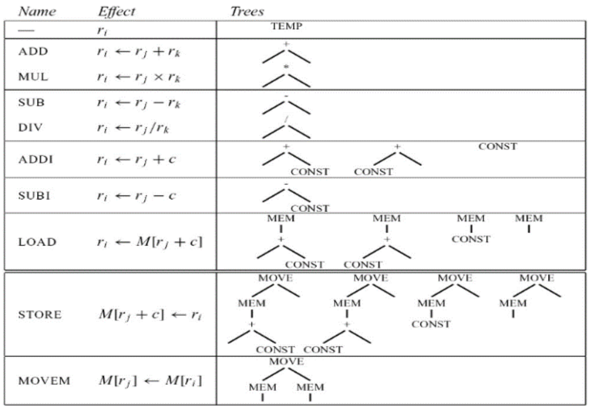
2)Construct the register interference graph.

ANSWER:

7. Given the following IR tree (6 cents)



1) Please circle the tiles and number them in the order that they are munched;
(2) Generate Jouette-machine instructions using Maximal munch. The Jouette architecture illustrated as follows.



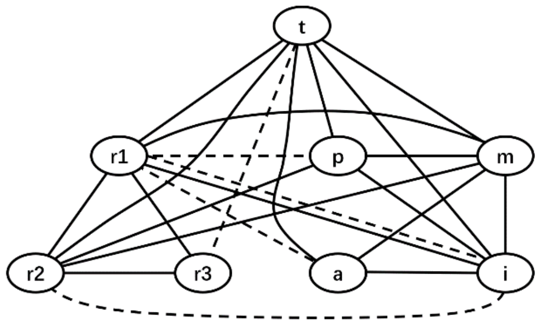
Name	Effect	Trees
—	r_i	TEMP
ADD	$r_i \leftarrow r_j + r_k$	
MUL	$r_i \leftarrow r_j \times r_k$	
SUB	$r_i \leftarrow r_j - r_k$	
DIV	$r_i \leftarrow r_j / r_k$	
ADDI	$r_i \leftarrow r_j + c$	
SUBI	$r_i \leftarrow r_j - c$	
LOAD	$r_i \leftarrow M[r_j + c]$	
STORE	$M[r_j + c] \leftarrow r_i$	
MOVEM	$M[r_j] \leftarrow M[r_i]$	

ANSWER:

8. A compiler is compiling a program for a target machine with three registers r1, r2 and r3. The compiler has produced the instruction list, and has built the inference graph for this program, as shown below.

Suppose you are performing register allocation on this inference graph. (8 points)

enter: t ← r3	The interference graph

<pre>   p ← r1
          i ← r2
          m ← M[p]
          a ← M[p+4]
loop: r1 ← a
          r2 ← i
          call f (use r1, r2, def r1)
          a ← r1
          if a>m goto end
          a ← m-a
          i ← i-1
          if i>0 goto loop
end: r1 ← i
          r3 ← t
          return (r1, r3 live out)</pre>
	

(1)The program above cannot be allocated with only 3 registers, therefore we have to choose a variable to spill.

a) Assume that the loop in the program executes an average of n times, where n≥1. Calculate the Spill Priority of the variables that need register allocation and explain which variable should be spilled.

b) Modify the program to spill the selected variable.

c) Draw the new interference graph after spilling the selected variable.

(2)Finish register allocation based on the resulting graph of question (1). Show the steps of the allocation process in detail. Please determine the opportunities for Coalesce only using the George criterion. You need to:

i. Describe what you do in each step.

ii. At least show the resulting graph after each Simplify and Coalesce phase. You can merge consecutive Coalesce steps and only draw their final graph.

iii. Show the result of the coloring.

(3)Rewrite the program based on your register allocation results.

ANSWER: