浙江大学 2002-2003 学年第二学期期终考试
《编译技术》课程试卷（闭卷）

校区______ 学院_______ 专业______ 姓名________ 学号________

考试时间长度 _2 小时___ 考试成绩 ________

一、Describe and explain the main translation phases of a compiler. （5）

二、Show that a left-recursive grammar A → A a | b cannot be LL(1). (A is a nonterminal, a and b are terminals.) （5）

三、Grammar G containing the following state **I**$_s$ ，describe the condition of this grammar being LR(1). (X is a terminal) (5)

I$_s$

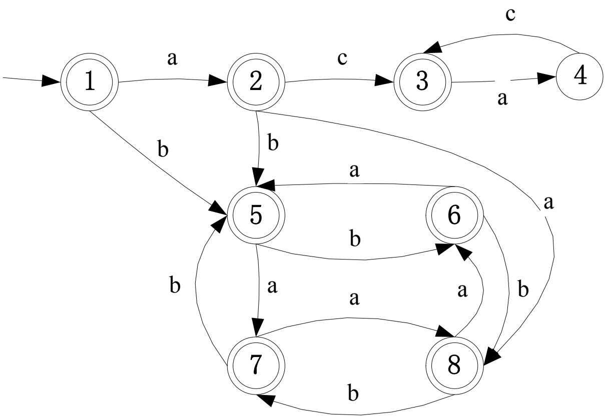
$A \to \alpha .X\beta$, a
$A \to \alpha .$, b
$B \to \gamma .$, c

四、Explain the relationships of the concepts: Attribute grammar, Dependency graph, Synthesized and Inherited Attributes. (5)

五、Give the output of the following program (in C syntax) using the four parameter passing methods. (8)

```
#include <stdio.h>
int   i = 0;
void swap(int x, int y)
{     x = x + y;
      y = x – y;
      x = x – y;
}
main ( )
{ int a[3] = {1,2,0};
swap(i, a[i]);
printf(“%d %d %d \n”, a[0], a[1], a[2]);
return 0;
 }
```

六、6.1 Apply the state minimization algorithm to the following DFA: （5）



6.2 Which of the following regular expression produce the same language as the minimized DFA? Using Thompson's construction to convert the selected regular expression into an NFA. （5）

(1) (a|b)*|ac(ac)* (2) (ac)*|(a|b)* (3) (ac)* | a (a|b)*

6.3 Convert the above NFA into a DFA using the subset construction. （5）

七、Give the grammar

S → a S b S | b S a S | ε

7.1 Give the left-most derivation and right-most derivation of the abab. (3)

7.2 Prove this grammar is ambiguity. (5)

7.3 Explain the language defined by this grammar using natural language.(2)

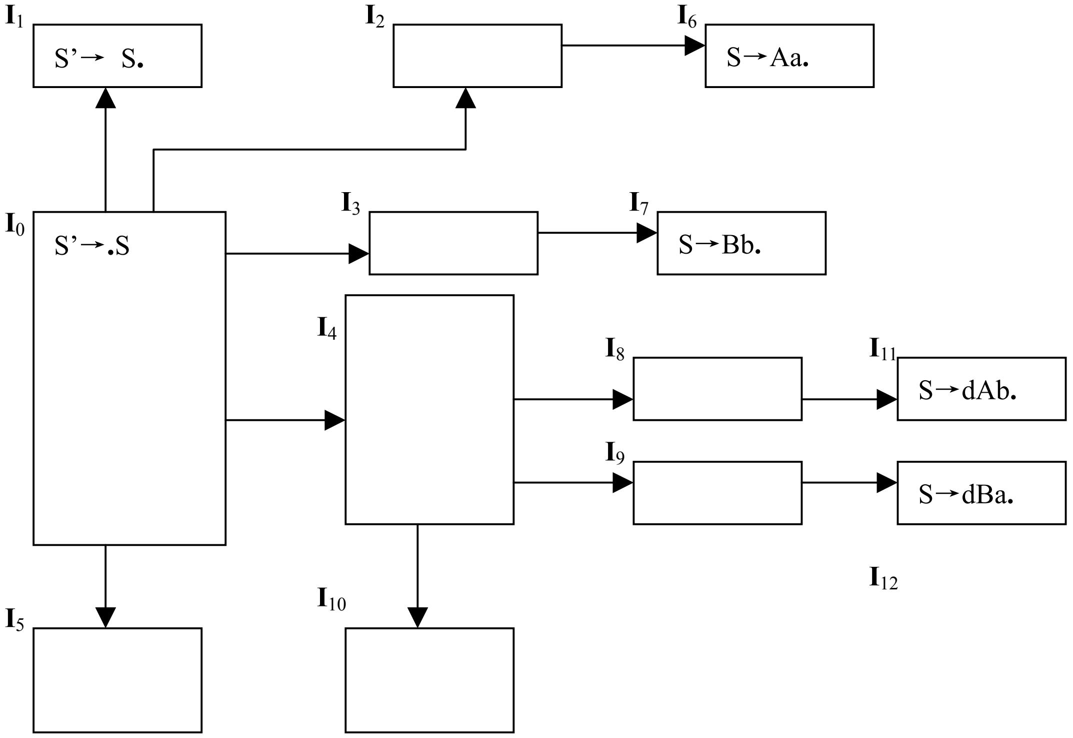
八、Consider the following grammar.

0. S'→S
1. S→Aa
2. S→dAb
3. S→Bb
4. S→dBa
5. A→c
6. B→c

8.1 Construct the DFA of LR(1) for this grammar. (7)

8.2 Construct the LR(1) parsing table. (5)

8.3 Show the parsing stack and the actions of the LR(1) parser for the input string 'dca'. (5)



LR(1) pasing table

S	ACTION					GOTO		
	a	b	c	d	#	S	A	B
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								

Parsing for input string ‘dca’

	Parsing stack	Input string	ACTION	GOTO

九、. In order to implement the most closely nested rule for block structure, please design two kinds of implementation for the HASH symbol table and use the following example (After processing the line 9) to illustrate the status of the symbol table. (10)

```
1    int i, j;
2    int f(int size)
3    { char i, temp;
4         …
5         {char j;
6           …
7         }
8         …
9         {double *j;
10          …
11        }
12   }
```

十、Consider the following program: (10)

```
program ex;
     a : integer;
     procedure   PP ( x : integer);
         begin
              x :=5;      x:=a+1;
         end;
     begin
```

```
      a :=2;
      PP(a);
      write(a)
   end.
```

Please draw the stack of activation records before the call to PP(a) and after the call to PP(a), showing the control and access links.

十一、Consider the following statement: (10)

```
if   (a>b)   while (x>0)   x = x-2;
else   y=y+1;
```

It generates the uncompleted p-code sequence, try to fill the blank in this sequence.

```
lod a
lod b
gt
_____
_____
lod x
ldc 0
gt
______
lda x
lod x
ldc 2
sbi
sto
______
_______
________
________
lda y
lod y
ldc 1
adi
sto
________
```

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

1、Both DFA and NFA can recognize regular set.

2、Context free grammar can generate language L = $\{a^n b^n c^m \mid n >=0 , m>=0\}$

3、Intermediate Code generation depends on detailed information about the target architecture, and doesn't care the characteristics of the source language.

4、A display is a data structure that may be used as a alternative to static links for maintaining access to the variables.

5. A grammar is ambiguous if it has two different derivations for a sentence.

6. In grammar, nullable symbol can only derive the empty string.

7. All LR(0) grammars can be parsed by SLR parser.

8.There might be shift-reduce, shift-shift and reduce-reduce conflicts during the LR parsing.

二、Single Choice

1、Which of the following string can be defined by the regular expression ((b|c)*a(b|c)*a) (b|c)*.

[A] abbcab [B] aaaa [C] abbbbbc [D] bbacc

2、 the output of the scanner is:

[A] token [B] syntax tree [C] target code [D] intermediate code

3、Lex is a tool that (   )

[A] break the source file into individual words, or tokens

[B] analyze the phrase structure of the program

[C] is a lexical analyzer generator

[D] is a parser generator

4、 In the following which is(are) not commonly found in a stack frame (activation record)?

[A] return address [B] static variables

[C] saved registers [D] arguments

5、which does the operation pushing the return address in the stack frame?

[A] the caller procedure

[B] the called procedure

[C] the operation system

[D] CPU

6. Which action is not in a LR Parsing table?

[A] Shift [B] Push [C] Accept [D] Reduce

7. In the production B→αAγ, Which would not be in the FOLLOW(A) Set?

[A] ε [B] FIRST(γ) [C] FOLLOW(B) [D] $

8. Which element would not appear in the stack of LR parser?

[A] nonterminal [B] terminal [C] state [D] $

9. Which grammar parser below is the most powerful?

[A] LL(1) [B] LR(0) [C] SLR [D] LR(1)

10. the parsing method of YACC is ( )

[A] LALR(1) [B]LR(1) [C] SLR(1) [D] LL(1)

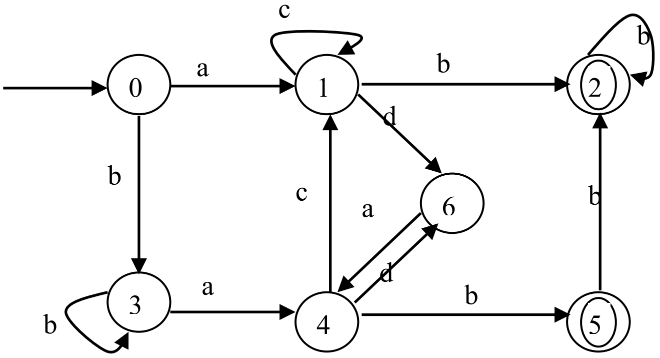
三、**Questions**

1、 Describe and explain the main translation phases of a compiler. （7 cents）

Answer:

2、Please explain the meaning of the binding {g ↦ string, a↦ int }. (5 cents)Answer:

3、Apply the state minimization algorithm to the following DFA. Then give a regular expression to describe this language. (10 cents)



Answer:

4. Consider the following statement: (10 cents)

if (a>b) x = x-2 else y=y+1

Please use Intermediate Representation Tree to describe the above code.

Answer:

5. Given the grammar

A→ A A
A→ (A)
A→ ε

Drawing two different parsing trees for the string **( )**, showing that it is ambiguous. (7 cents)

Answer:

6. Consider the following grammar (20 cents)

S→A S→B
A→ e A→ f
B→ (C) C→ S D
D→ S D D→

a. Calculate nullable, FIRST and FOLLOW for nonterminals in the grammar.

b. Construct the LL(1) parsing table for the grammar.

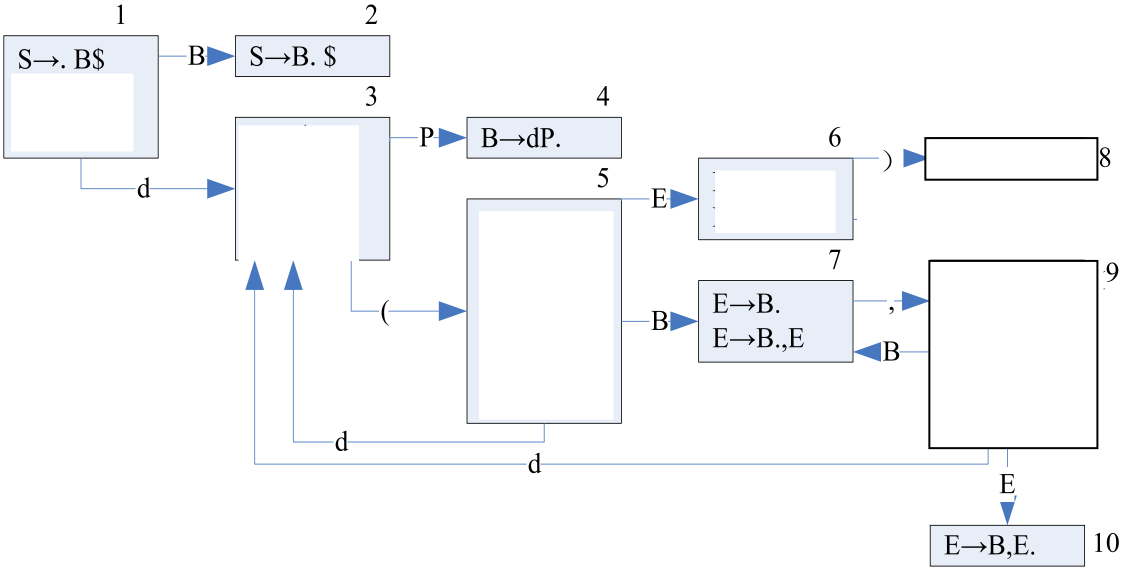
c. Show that the grammar is LL(1).

Answer:

7. Construct the LR(0) states for this grammar, and then determine whether it is an SLR grammar, give the SLR parsing table. (15 cents)

0 S→B $ 1 B→ d P 2 B→ d (E) 3 P→ 4 P→ (E)

5 E→B 6 E→B, E



Answer:

一、**Mark each statement *true* or *false* (16 cents)** （**每题 2 分**）

1 (T)

2、 (F)

3、 (F)

4、 (T)

5. (F)

6. (F)

7. (T)

8. (F)

二、**Single Choice (10 cents)** （**每题 1 分**）

1、A

2、A

3、C

4、B

5、B

6. B

7. A

8. D

9. D

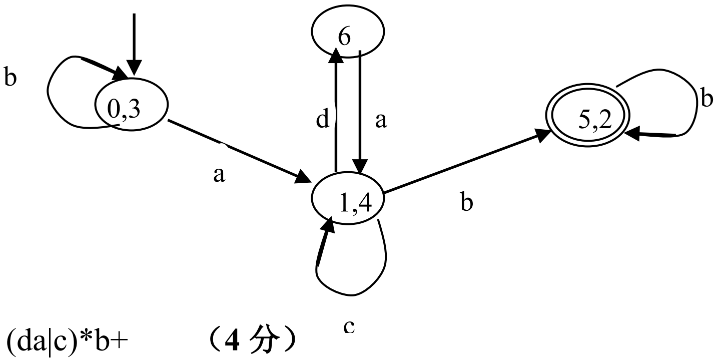
10. A

三、**Questions (74 cents)**

1、 Lexical analysis 、parser analysis 、Semantic Analysis 、code generation、optimization （7 cents）(**有一些解释就可以了，至少要有以上 5 个部分**）

2、 In the environment (σ_0) that the identifier a is an integer variable and g is a string variable. (5 cents) （**描述准确就给分**）

3、(10 cents) （最小 DFA 给 6 分）



b*a (da|c)*b+ （**4 分**）

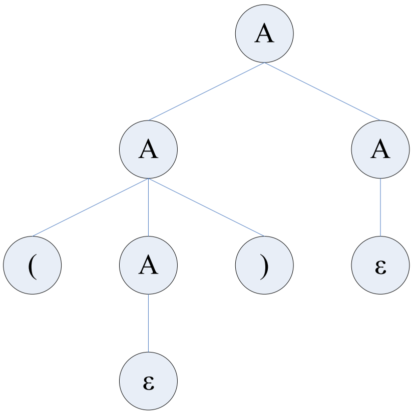
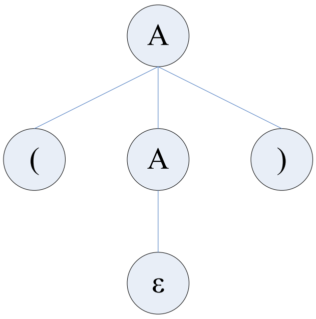
4. SEQ(SEQ(SEQ(CJUMP(>,a,b,lt,lf),SEQ(LABEL(lt),SEQ(MOVE(MEM(x),BINOP(-,x,2)),JUMP(lnext)),SEQ(LABEL(lf),SEQ(MOVE(MEM(y),BINOP(+,y,1)), JUMP(lnext)),LABEL(lnext))

(10 cents) （**画树也可以，少一个扣 1 分，扣到剩 3 分为止，只要做了，有最低分 3 分**）

5. (7 cents)

The string **()** has two different parsing trees below: （**给 5 分**）

A→AA	A→(A)
→(A)A	→()

→()A →()	
	

So it is ambiguous. （结论给 **2** 分）

6. （20 cents）

a. Calculate nullable, FIRST and FOLLOW as below: （**8** 分，**0.5** 分 **1** 格）

	nullable	FIRST	FOLLOW
S	no	e f (	e f ( )
A	no	e f	e f ( )
B	no	(	e f ( )
C	no	e f (	)
D	yes	e f (	)

b. Construct the LL(1) parsing table for the grammar: **(8** 分，**0.5** 分 **1** 格**)**

	e	f	(	)
S	S→A	S→A	S→B	
A	A→e	A→f		
B			B→(C)	
C	C→SD	C→SD	C→SD	
D	D→SD	D→SD	D →SD	D→

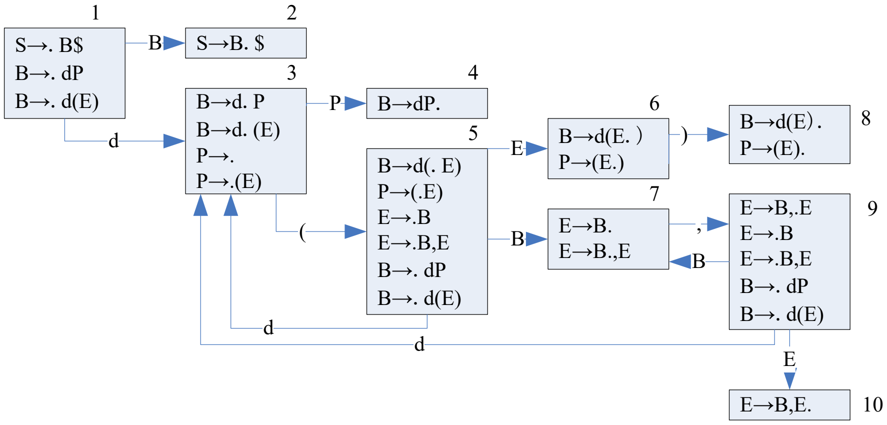
c. There are no duplicate entries in the LL(1) parsing table, so it's LL(1) grammar. （**4** 分）

7. (15 cents)

0 S→B $ 1 B→ d P 2 B→ d (E) 3 P→ 4 P→ (E) 5 E→B

6 E→B, E

(状态机共填六个状态，每个 1.5 分，9 分)



To build the SLR parsing table, we need the FOLLOW sets as below: **(.该表不一定要有)**

	nullable	FIRST	FOLLOW
S	no	d	$
B	no	d	$,)
P	yes	(	$,)
E	no	d	)

Building the SLR parsing table: （共 **4** 分） （错一行扣 **0.5** 分，扣完 **4** 分为止）

	d	(	)	,	$	S	B	P	E
1	s3						g2		
2					a				
3		s5	r3	r3	r3			g4	
4			r1	r1	r1				
5	s3						g7		g6
6			S8						
7			r5	S9					
8			r2/r4	r2/r4	r2/r4				
9	s3						g7		g10
10			r6						

The SLR parsing table contains conflicts (duplication entries), so it is not a SLR grammar. （**2** 分）

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

1. Scopes of the variables are intercrossed sometimes.

2. Yacc can not use ambiguous grammars.

3. Both DFA and NFA can recognize regular set.

4. A grammar is ambiguous if it has two different derivations or two different parse trees for a sentence.

5. Grammars with left recursion eliminated are LL(1).

6. All LL(1) grammars can be parsed by LALR parser.

7. Code generation depends on detailed information about the target architecture, and doesn't care the characteristics of the source language.

8. The best choice of data structure of the symbol table is HASH table.

9. An L-value can not occur on the right of

an assignment statement.

10.An intermediate representation is a kind of machine language with committing to too much machine-specific detail.

二、 **Single Choice (15 cents)**

1. Strings over the alphabet {a,b,c} where the length of the string is even can be generated by regular expression ___:

[A]. ((a | b | c)(a | b | c))*

[B]. (a | b | c)(aa | bb | cc)*(a | b | c)

[C]. (aa | bb | cc)*

[D]. (a | b | c)*

2. Which of the following is not only found in a stack frame (activation record)?

[A] static variables

[B] return address

[C] saved registers

[D] parameters

3. The elements in a LR Parsing table are labeled with the following kinds of actions

except___?

[A]. Shift [B]. Goto [C]. Accept [D]. Push

4. In the production B→αAγ, Which would not be in the FOLLOW(A) Set?

[A]. ε [B]. FIRST(γ) [C]. FOLLOW(B) [D]. $

5. Which grammar parser below is the least powerful?

[A]. LL(0) [B]. LR(0) [C]. SLR [D]. LR(1)

6. Here is a grammar:

X→a X→ Y→b Y→X

Z→c Z→XYZ W→d W→XY

Which symbol is not nullable?

[A]. X [B].Y [C].Z [D].W

7. Which element would not appear in the stack of LR parser?

[A].terminal [B].nonterminal [C].$ [D].state

8. Normally, we do error recovery with methods except ____?

[A]. add error production [B]. modify the parsing tables

[C]. eliminate the conflict [D]. modify the parsing engine

9.Which of the following is commonly found in a stack frame (activation record)?

[A] fp pointer

[B] sp pointer

[C] static variable

[D] global variable

10. Many compilers are divided into two largely independent passes: a front end, responsible for analyzing source code, and a back end, responsible for generating target code. (They may also include a third, intermediate phase, responsible for code improvement.) What is the most important motivation for this division into passes?

[A] to provide portability of compiler

[B] to facilitate division of labor when a compiler is built by a large team of programmers

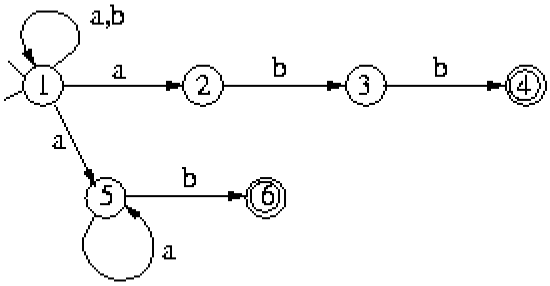
[C] to minimize memory requirements on modern machines

[D] to facilitate debugging the compiler

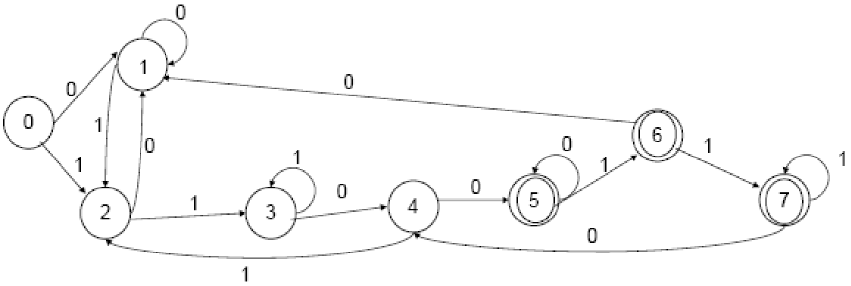
三、 Questions (65 cents)

1. Please explain the meaning of the leftmost derivation. (5 cents)

2. Convert the following NFA to DFA. You need to show the conversion steps. (8 cents)



3. Minimize the following DFA. You need to show the minimization steps. (6 cents)



4. Consider the following grammar: (12 cents)

S→ A B

A→ a

A→ +

A→ (S)

B→ ∗A B

B→

Calculate nullable, FIRST and FOLLOW for nonterminals in the grammar.

5. Show that the following grammar is LR(1) but not LALR(1), firstly you should modify this grammar with new start symbol. (15 cents)

S→ a A d　　S→ b B d
S→a B e　　S→ b A e
A→ c　　　　B→ c

6. Consider an example in Tiger language: (11 cents)

```
1  let
2
3  function f(a: int, b: int, c: int) =
4  (
5     let var j := a+b;
6         var c := a*a+b*b
7         var a := “hello”
8     in
9         print(a); print(j); print(c)
10        end
11        print(b)
12    )
13
14    in
15        f(10, 20, 30)
16    end
```

Suppose hash(a)=5, hash(b)=3, hash(c)=8, hash(j)=8, and imperative-style environments are used.

(1) Show the symbol table when line 8 is compiled

(2) Show the symbol table when line 11 is

compiled.

7. Show the activation record of the function ABC that is called. Show the frame pointer and stack pointer. (the runtime stack grows from higher to lower memory addresses, integer variables require 2 bytes of storage, char variables require 1 byte of storage, double variables require 8 bytes of storage, addresses require 4 bytes.) (8 cents)

答案

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

1. (F) 2. （F） 3. （T） 4. （F） 5. （F）

6. （F） 7. (F) 8. （T） 9. (F) 10. (F)

二、 Single Choice (15 cents)

1. A 2. A 3. D 4. A 5. A

6. C 7. C 8. C 9 A **10.** A

三、 Questions (65 cents)

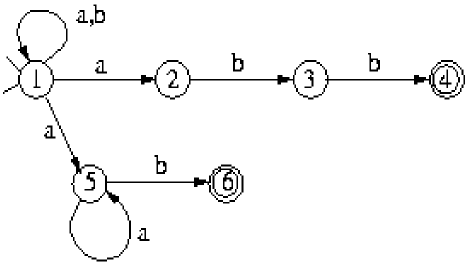
1. (5 cents)

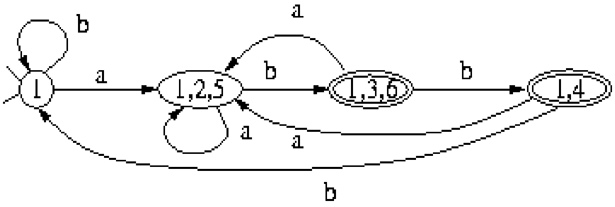
[Solution]

A leftmost derivation is one in which the leftmost symbol is always be expanded.

2. (8 cents)

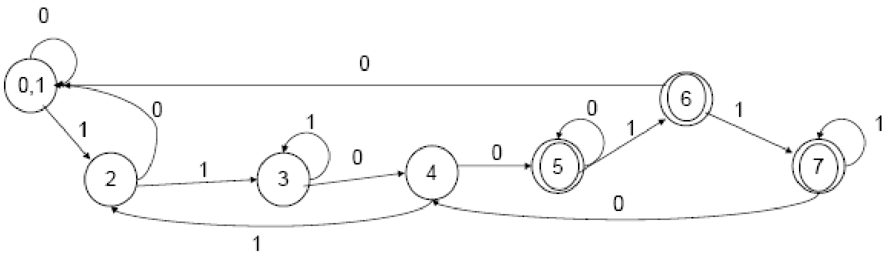
[Solution]





3. (6 cents)

[Solution]



P0={0,1,2,3,4}, P1={5, 6, 7}

$\delta(d4,0) = d5 \in P1$, but $\delta(d0/d1/d2/d3,0) \in P2$

P0={0,1,2,3}, P1={4}, P2={5,6,7}

$\delta(d3,0) = d4 \in P1$, but $\delta(d0/d1/d2,0) \in P0$

P0={0,1,2}, P1={3}, P2={4}, P3={5,6,7}

$\delta(d2,1) = d3 \in P1$, but $\delta(d0/d1,1) \in P0$

P0={0,1}, P1 = {2}, P2={3}, P3={4}, P4={5,6,7}

$\delta(d0/d1,0) = d1 \in P0$, $\delta(d0/d1,1) = d2 \in P1$

$\delta(d7,0) = d4 \in P3$, $\delta(d5,0) = d5 \in P4$, $\delta(d6,0) = d1 \in P0$

P0={0,1}, P1 = {2}, P2={3}, P3={4}, P4={5}, P5={6}, P6={7}

4. (12 cents)

[Solution]

	nullable	FIRST	FOLLOW
S	no	a,+,(	)
A	no	a,+,(	*,)
B	yes	*	)

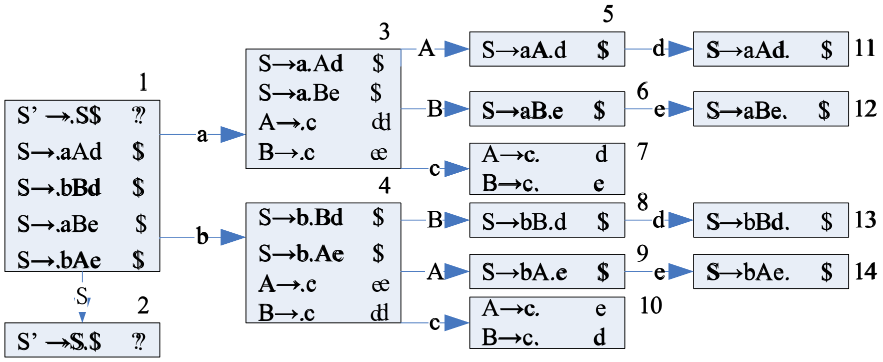
5. (15 cents)

[Solution]

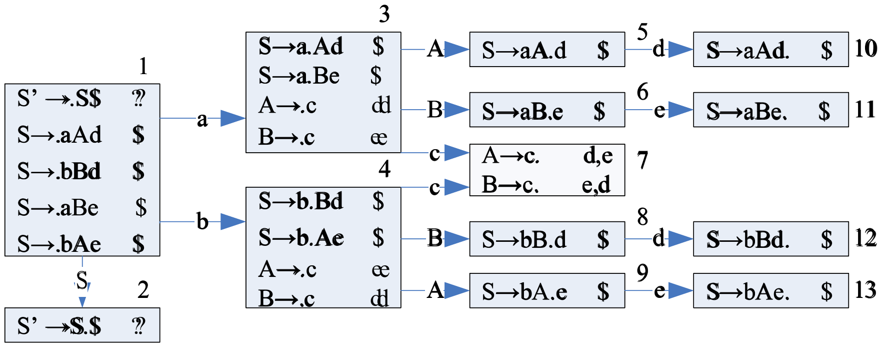
We augment the grammar with a new start symbol S’ and a new production as below:

0 S’ →S$	
1 S→aAd	2 S→bBd
3 S→aBe	4 S→bAe
5 A→c	6 B→c

The LR(1) states diagram as below:



The LALR(1) states diagram as below:

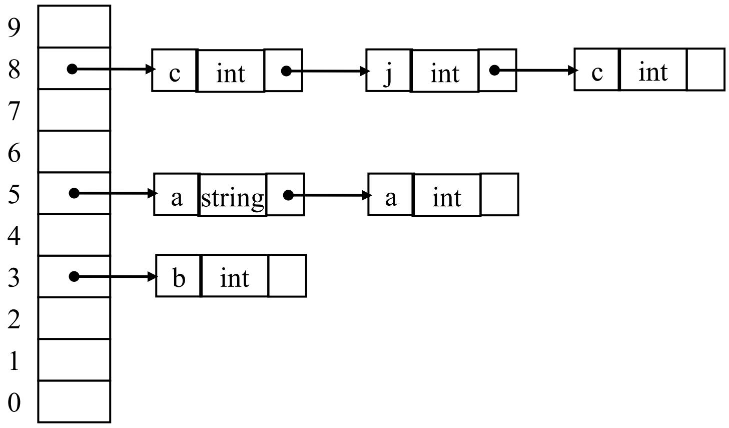


From the graphs, we know that LALR(1) table contains reduce-reduce conflicts, but the LR(1) table has none.

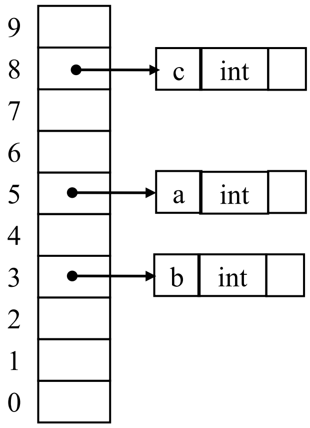
6. (11 cents)

(1) Show the symbol table when line 8 is compiled.[Solution]

[Solution]



(2) Show the symbol table when line 11 is compiled.
[Solution]



7. (8 cents)

```
void main ( )   {
      int x,y;
      x= 0;
      y = ABC (1,2,3);
      .........

}
```

```
int ABC (int a, int b, int c)   {
   double y;
   char buffer[16];
   int sum, x=1 ;
   ..........
   return sum;
}
```

[Solution]

	3
	2
	1
fp →	静态链
	y
	y
	y
	y
	buffer[0] buffer[1]
	buffer[2] buffer[3]
	buffer[4] buffer[5]
	buffer[6] buffer[7]
	buffer[8] buffer[9]
	buffer[10] buffer[11]
	buffer[12] buffer[13]
	buffer[14] buffer[15]
	sum
sp →	x=1
	return address ………

the temporary variables

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

1、Both DFA and NFA can recognize regular set.

2、Intermediate Code generation depends on detailed information about the target architecture, and doesn’t care the characteristics of the source language.

3、A display is a data structure that may be used as a alternative to static links for maintaining access to the variables.

二、Single Choice (10 cents)

1、the output of the scanner is:

[A] token [B] syntax tree [C] target code [D] intermediate code

2、Which grammar parser below is the most powerful?

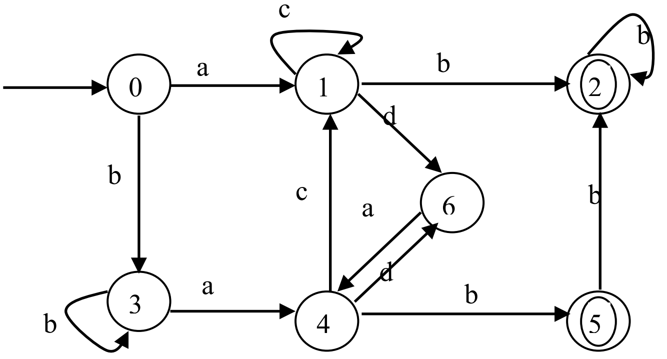
[A] LL(1) [B] LR(0) [C] SLR [D] LR(1)

3、the parsing method of YACC is ( )

[A] LALR(1) [B]LR(1) [C] SLR(1) [D] LL(1)

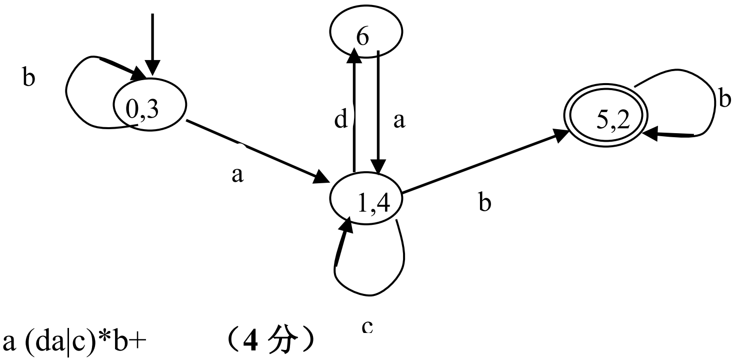
三、Questions

1、Apply the state minimization algorithm to the following DFA. Then give a regular expression to describe this language. (10 cents)



Answer:

（最小 DFA 给 6 分）



b*a (da|c)*b+　　　**（4 分）**

暂未填写邮箱信息，忘记密码将无法找回。请前去“个人设置”进行填写。
IP 错误

确定

期末考试

- 单选题
- 多选题
- 判断题
- 填空题
- 简答题
- 综合题
- 题库随机抽题
- 题库导入
- 文本导入

答题进度: 0/31
交卷
确定提交试卷吗？

确定 取消

交卷

第六、七、八章节

1. 1.

   When inheriting a previously computed synthesized attribute during LR parsing, it is suitable to treat the computed synthesized attribute as ________.

   单选题 (2 分)

   1. ○ A.

      just leaving it on the value stack

   2. ○ B.

      external data structure

   3. ○ C.

      parsing as parameter

   4. ○ D.

      return value

2. 2.

   The concept ________ is not related with automatic management of the heap.

   单选题 (2 分)

   1. ○ A.

      mark and sweep

   2. ○ B.

      memory compaction

   3. ○ C.

      display

   4. ○ D.

      stop-and-copy

3. 3.

   Which of the following is commonly found in an activation record?

   单选题 (2 分)

   1. ○ A.

      static variable

   2. ○ B.

      symbol table

   3. ○ C.

      global variable

   4. ○ D.

      temporaries

4. 4.

   In the following, ________ must be done by the caller.

   单选题 (2 分)

   1. ○ A.

      saving the used registers

   2. ○ B.

      computing the arguments

   3. ○ C.

      setting up bookkeeping information

   4. ○ D.

      saving the return address

5. 5.

   The suitable structure of the symbol table with large amount of symbols is ________.

   单选题 (2 分)

   1. ○ A.

      stack

    2. ○ B.

       linear list

    3. ○ C.

       hash table

    4. ○ D.

       tree

6. 6.

   Given the following declarations:

     t1 = array [10] of int;

     t2 = array [10] of int;

   if t1 and t2 is equivalent, it follows ________ equivalence.

   单选题 (2 分)

    1. ○ A.

       semantic

    2. ○ B.

       name

    3. ○ C.

       declaration

    4. ○ D.

       structural

7. 7.

   The space for bookkeeping information will not contain________.

   单选题 (2 分)

    1. ○ A.

       return address

    2. ○ B.

       control link

    3. ○ C.

       stack pointer

    4. ○ D.

       access link

8. 8.

The idea of delayed evaluation is used in ________ parameter passing mechanism.

单选题 (2 分)

1. ○ A.

   pass by reference

2. ○ B.

   pass by value-result

3. ○ C.

   pass by value

4. ○ D.

   pass by name

9. 9.

The output of the semantic analysis phase is _____________.

单选题 (2 分)

1. ○ A.

   a syntax tree

2. ○ B.

   an annotated tree

3. ○ C.

   a parse tree

4. ○ D.

   a token stream

10. 10.

Which of the following is used to retrieve nonlocal data declared within another procedure?

单选题 (2 分)

1. ○ A.

   activation tree

2. ○ B.

   control link

3. ○ C.

   frame pointer

4. ○ D.

   access link

11. 11.

The access link will be used in ________.

单选题 (2 分)

1. ○ A.

   stack-based environments without local procedures

2. ○ B.

   fully static runtime environment

3. ○ C.

   stack-based environments with local procedures

4. ○ D.

   fully dynamic runtime environment

12. 12.

A Yacc-generated parser implements the semantic values by keeping a ________ parallel to the parsing stack.

单选题 (2 分)

1. ○ A.

   terminal stack

2. ○ B.

   action stack

3. ○ C.

   non-terminal stack

4. ○ D.

   value stack

13. 13.

Which of the following is true ?

单选题 (2 分)

1. ○ A.

   A synthesized attribute at node N can be defined in terms of inherited attribute values at node N itself.

2. ○ B.

   A synthesized attribute at node N can not be defined in terms of attribute values at the children of N.

3. ○ C.

   A synthesized attribute at node N can not be defined in terms of inherited attribute values at node N itself.

4. ○ D.

   An inherited attribute at node N can be defined in terms of attribute values at the children of node N.

14. 14.

When computing the attributes of a language, a traverse order of the dependency graph must be _____.

单选题 (2 分)

1. ○ A.

   directed, acyclic

2. ○ B.

   directed , cyclic

3. ○ C.

   undirected, acyclic

4. ○ D.

   undirected, cyclic

15. 15.

The runtime environment dependents only on the architecture of the target machine.

判断题 (1.5 分)

1. ○ A.

   False

2. ○ B.

   True

16. 16.

The space for local temporaries is generally not allocated in the procedure activation record.

判断题 (1.5 分)

1. ○ A.

   True

2. ○ B.

   False

17. 17.

Intermediate code can be very high level, or it can closely resemble target code.

判断题 (1.5 分)

1. ○ A.

   False

2. ○ B.

   True

18. 18.

Semantic analysis is an independent phase that its computations always occur prior to code generation and after parsing.

判断题 (1.5 分)

1. ○ A.

   False

2. ○ B.

   True

19. 19.

All inherited attributes can be changed into synthesized attributes by suitable modification of the grammar, without changing the language of the grammar.

判断题 (1.5 分)

1. ○ A.

   False

2. ○ B.

   True

20. 20.

All the S-attribute grammars belong to L-attribute grammars.

判断题 (1.5 分)

1. ○ A.

   False

2. ○ B.

   True

21. 21.

The general organization of runtime storage will contain the code area, the global/static area, a stack, a free space as well as a heap.

判断题 (1.5 分)

1. ○ A.

   True

2. ○ B.

   False

22. 22.

    Yacc can not use ambiguous grammars.

    判断题 (1.5 分)

    1. ○ A.

       False

    2. ○ B.

       True

23. 23.

    An inherited attribute can be calculated by a postorder traversal of the parse tree.

    判断题 (1.5 分)

    1. ○ A.

       True

    2. ○ B.

       False

24. 24.

    When a procedure is called, an activation record P is generated on stack. There may be activation records on stack corresponding to sibling children nodes of P in the activation tree.

    判断题 (1.5 分)

    1. ○ A.

       True

    2. ○ B.

       False

25. 25.

    Three-address code has three fields.

    判断题 (1.5 分)

    1. ○ A.

       True

    2. ○ B.

       False

26. 26.

The activation record kept in the stack is always directly pointed by frame pointer (fp).

判断题 (1.5 分)

1. ○ A.

   True

2. ○ B.

   False

27. 27.

Translate the following statements into a quadruple sequence:

while (a<b and c>d) do

  if (e<f) then

        x := y + z;

else

       x := y - z;

简答题 (10 分)

附件 上传 最多可上传6张图片

28. 28.

Given the following program:

```
 #include <stdio.h>

 Int i=0;

 Void swap(int x, int y)

 { x= x+y;

  y=x-y;

  x=x-y;

 }

 main()

 { int a[3]= {2, 0, 1};

  swap(i, a[i]);

  printf(“%d %d %d %d \n”,i ,a[0], a[1], a[2]);

  return 0;

 }
```

Please show the output of the above program using four parameter passing methods (pass by value, pass by reference, pass by value-result, pass by name) respectively.

简答题 (8 分)

附件 上传 最多可上传6张图片

29. 29.

The IR code generation can be performed during a parse. Please complete the C program of the action1、action2、action3、action4、action5、action6 in the YACC file of the following grammar.

Grammar:

stmt → *while-stmt* | **break** | **other**

while-stmt → while (*exp*) *stmt*

exp → **true | false**

the IR codes as following :	the three-address codes as following:	
label L1	read x	(rd,x _,_)
<code to evaluate *exp* to t1>	if_false,t1,L1	(if_f, t1,L1,_)
if_false t1 goto L2	label L1	(lab,L1,_,_)
<code for *stmt*>	goto L1	(go,L1,_,_)
goto L1	x=t3	(asn,t3, x ,_)
label L2		

YACC:

while-stmt : **{ action1 }** while (*exp*) **{ action2 }** *stmt*

{action3 }

exp : **true** **{ action4 }**

| false **{ action5 }**

stmt: *while-stmt*

| **break** { **action6** }

| **other**

简答题 (12 分)

附件 上传 最多可上传6张图片

30. 30.

Consider the CFG of number with a one-character suffix **o** (for octal) or **d** (for decimal).

based-num → num basechar

basechar → **o** | **d**

num → num digit | digit

digit → 0|1|2|3|4|5|6|7|8|9

In this case, we need two attributes, base (for indicating octal or decimal) and val (for value of the number).

(1) Rewrite the grammar so that the computation of the attribute ***base*** does not depend on parent nodes.

(2) Given the number “**789o**”, show the dependency of attribute computation on its parsing tree of the above modified CFG.

简答题 (14 分)

附件 上传 最多可上传6张图片

31. 31.

Given a program as follow:

```
 int x=3;

 void g(int);

 void f(int n)

 {

     static int x =1;

     g(n);

     x--;

 }

 void g(int m)

 {

    int y = m-1;

    If (y > 0)

    { f(y);
```

```
       x--;
       g(y);
      }
  }
  main()
  {
      g(x);
      return 0;
  }
```

Please draw the runtime environment of the above program at the point just before the second call to g.

简答题 (10 分)

附件 上传 最多可上传6张图片

知道了
交卷
确定提交试卷吗?

确定 取消

交卷
在线测试提交
测试时间已过

确认 取消

公式编辑器

保存 取消

提示

确定 取消