

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]. $((a \mid b \mid c)(a \mid b \mid c))^*$

[B]. $(a \mid b \mid c)(aa \mid bb \mid cc)^*(a \mid b \mid c)$

[C]. $(aa \mid bb \mid cc)^*$

[D]. $(a \mid b \mid c)^*$

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 $B \rightarrow \alpha A \gamma$, 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:

$X \rightarrow a$ $X \rightarrow$ $Y \rightarrow b$ $Y \rightarrow X$
 $Z \rightarrow c$ $Z \rightarrow XYZ$ $W \rightarrow d$ $W \rightarrow XY$

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

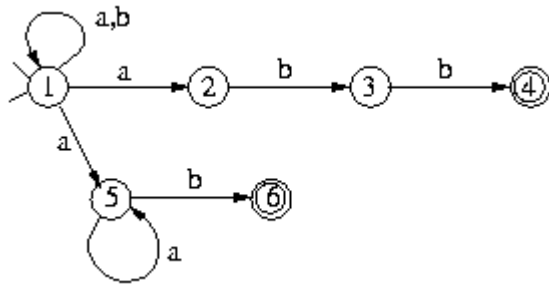
[C] to minimize memory requirements on modern machines

[D] to facilitate debugging the compiler

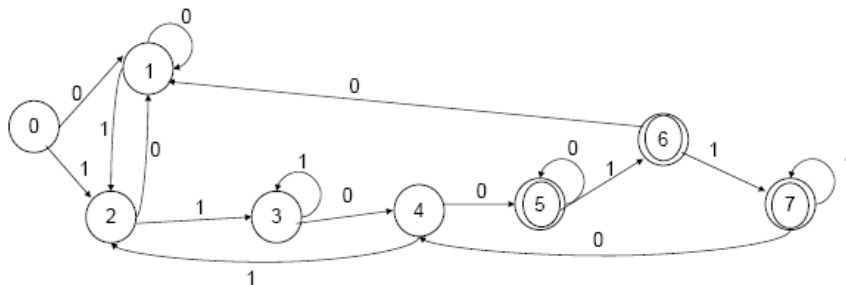
三、 Questions (65 cents)

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)

$$S \rightarrow A B$$

$$A \rightarrow a$$

$$A \rightarrow +$$

$$A \rightarrow (S)$$

$$B \rightarrow * A B$$

$B \rightarrow$

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)

$S \rightarrow a A d$	$S \rightarrow b B d$
$S \rightarrow a B e$	$S \rightarrow b A e$
$A \rightarrow c$	$B \rightarrow c$

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 $\text{hash}(a)=5$, $\text{hash}(b)=3$, $\text{hash}(c)=8$, $\text{hash}(j)=8$, 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)