2019-2《编译原理》期末考试试卷

警示：考试作弊者，不授予学士学位！

题 号	1	2	3	4	5	6	7	8	总 分
得 分									

1. [20pts.] Select the correct choice and fill the answer in the table below (10 Single choice questions, 2 points each).

Score

NO.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Answer										

(1) Which of the following phases belong to the front end of compiler? ( )

(A) Semantic Analysis and Code generation
(B) Lexical Analysis and Semantic Analysis
(C) Parsing trees and Regular Expressions
(D) Derivation and Reduction

(2) Consider the following two statements:

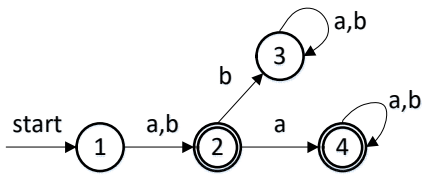
S1: $\{ 0^{2n} \mid n >= 1\}$ is a regular language

S2: $\{ 0^m 0^n 0^{(m+n)} \mid m >= 1 \text{ and } n >= 2\}$ is a regular language.

Which of the following is true? ( )

(A) Only S1 is correct
(B) Only S2 is correct
(C) Both of S1 and S2 is correct
(D) None of S1 and S2 is correct

(3) A scanner with lexical rules shown in the following finite state machine.

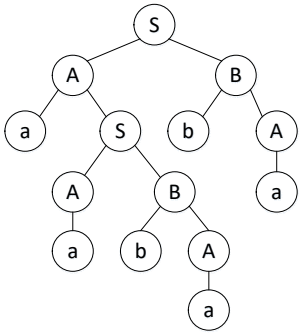


Base on Maximum Matching Rule, when the input is “baaba”, the output lexeme(s) of the scanner is ( ).

(A) “b”, “aa”, “b”, “a”
(B) “baaba”
(C) “b”, “aaba”
(D) “ba”, “aba”

(4) The parse tree below represents a rightmost derivation according to the grammar:

$S \rightarrow AB$
$A \rightarrow aS \mid a$
$B \rightarrow bA$



Which of the following are right-sentential forms corresponding to this derivation? ( )

(A) aaBba
(B) aSbA
(C) aAbaba
(D) aabAba

(5) Given the following grammar:

$E \rightarrow E * F \mid F + E \mid F$
$F \rightarrow F - F \mid id$

Which of the following is true? ( )

(A) * has higher precedence than +
(B) – has higher precedence than *
(C) + and – have same precedence
(D) + has higher precedence than *

(6) Consider the grammar

$E \rightarrow E + n \mid E \times n \mid n$

For a sentence $n + n \times n$, the handles in the right-sentential form of the reduction are ( )

(A) n, $E + n$ and $E + n \times n$
(B) n, $E + n$ and $E + n \times n$
(C) n, $n + n$ and $n + n \times n$
(D) n, $E + n$ and $E \times n$

(7) Which one of the following is true at any valid state in shift-reduce parsing? ( )

(A) At the bottom of parsing stack we find the prefixes.
(B) Stack consists of viable prefixes.
(C) Stack contains of non-terminals.
(D) None of the mentioned.

(8) The below grammar is not suitable for predictive-parsing because the grammar is ( )

$S \rightarrow Aa \mid Ab$
$A \rightarrow c$

(A) Ambiguous
(B) Right recursive
(C) Left recursive
(D) Left factor

(9) In a bottom-up evaluation of a syntax directed definition, its inherited attributes can do which of the following? ( )

(A) Always evaluated
(B) Can be evaluated if the definition is L attributed
(C) Can be evaluated if the definition has synthesized attributes
(D) Never be evaluated

(10) Symbol table is ( )

(A) Used by all phases of the compiler.
(B) Created in syntax analysis phase of the compiler.
(C) Data structure just used for storing lexical information about source program.
(D) Used for predictive-parsing.

Score

2. [5pts.] Consider the following grammar G[S]:

$S \rightarrow S0S \mid S1S \mid a$

Is this grammar ambiguous? if so, please give your reason with an example string and its parse trees.

3. [8pts.] Consider the following grammar :

Score

G[S]: $S \rightarrow (L) \mid aS \mid a$

$L \rightarrow L,S \mid S$

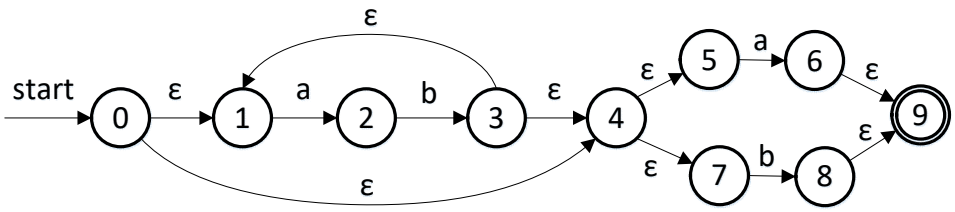
Please write the rightmost derivation for the sentential form '(S, (a))', and give the handle and the viable prefixes of this sentential form.

4. [12 pts.] Construct the minimum-state DFA for the following NFA:

Score

1) [8pts] Convert this NFA into DFA by subset construction. Both the transition table and the transition graph of DFA are required.

2) [4pts] Minimize the states of this DFA.



5. [15 pts.] Consider the grammar (decls, decl, type, varlist and varlist' are non-terminals):

Score

decls → decl;decls | ε decl → type varlist

type → int | bool varlist → id varlist'

varlist' → , varlist | ε

(1) **[8 pts.] Construct First and Follow sets for the nonterminals.**

(2) **[2 pts.] Is this grammar the LL(1) grammar? Give your reason.**

(3) **[5 pts.] Construct the LL(1) parsing table.**

Score

6. [5 pts.] Given the grammar G[S]: S→(S)A | aA , A→ BA | ε, B→ S | +S | *. (here, A→BA having higher priority on A→ ε), and the parsing table of G[S] as follow.

	a	(	)	+	*	$
S	S→aA	S→(S)A				
A	A→ ε A→BA	A→ ε A→BA	A→ ε	A→ ε A→BA	A→ ε A→BA	A→ ε
B	B→S	B→S		B→+S	B→*	

To achieve the predictive parsing, the rule "A→BA having higher priority on A→ε when selecting a production of A for derivations" is added into the grammar G[S]. Please give the parsing process for the input string (a*a) in the following table.

[Answer]:

Step	Stack	Input	Action
1	$S	(a*a)$	

Score

7. [15pts.] Consider the following augmented grammar G[S']:

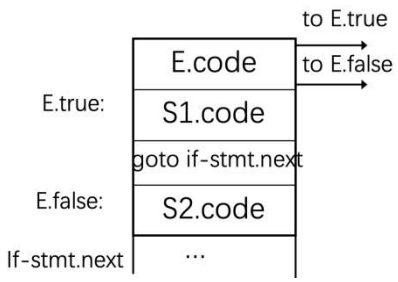
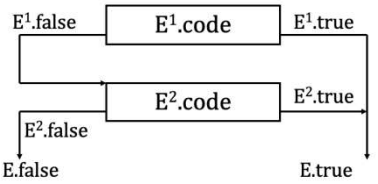
(0) S'→S (1) S→iDeD (2) S→ iD (3) D→Sb (4) D→ε

1) **[7pts.] Construct the DFA of LR(0) items for this augmented grammar.**
2) **[3pts.] Is this grammar the LR(0) or SLR(1) grammar ? Give your reason.**
3) **[5pts.] Construct the SLR(1) parsing table.**

Score

8 [20 pts.] Consider the following attribute grammar:

Grammar	Semantic Rules

if-stmt→ if E then S1 else S2	If-stmt.next=newlabel; E.true=newlabel; E.false=newlabel; S1.next=if-stmt.next; S2.next=if-stmt.next If-stmt.code=E.code || Label E.true || S1.code ||goto S.next || Label E.false || S2.code || if-stmt.next	
E → E1 or E2	E1.true=E.true; E1.false=newlabel; E2.true=E.true; E2.false=E.false; E.code=E1.code || Label E1.false || E2.code	
E → id1 relop id2	E.code = if id1.name relop id2.name goto E.true || goto E.false	
S → id = num	S.code = id.name = num.val	

Given the source code: if a<b or c<d then t= 5 else t=10

(1) [3pts.] Draw the Abstract Syntax Tree

(2) [3pts.] According to the semantic rules, calculate the inherited attributes 'true' ,'false' and 'next' on the corresponding nodes of the syntax tree, to form the semantic tree。

(3) [14 pts.]Consider the step (2) result and the synthetic attribute 'code' ， translate the three address code in a bottom- up order, recursively。

E1.code: [2pts.]　　E2.code: [2pts.]　　E.code: [4pts.]

S1.code: [1pts.]　　S2.code: [1pts.]　　If-stmt.code: [4pts.]