1. Which of the following is true.

A. The compiler executes the operations specified in the source program on inputs.

B. The scanner translates source code into intermediate code.

C. The semantic analysis is to take syntax trees as input and dependency graphs as output.

D. Both parse tree and three-address code are intermediate representations. (AST abstract syntax tree)

2. Which is the regular expression for the following finite automata.

3. Which of the following is not a LL(1) grammar.

A. G(B):B->bBb|ε B. G(A):A->bAa|ε C. G(C):C->bCc|ε D. None of the mentioned.

4. Which is correct about LL(1).

A. A LR(1) grammar is also a LL(1) grammar.

B. A grammar is called LL(1) grammar if the grammar is unambiguity.

C. All LL(1) grammars are context free grammars.

D. A grammar is LL(1) grammar if the grammar is SLR(1) grammar.

5. Which of the following is not conflict resolution in lexical analysis.

A. maximal munch rule B. panic mode C. adding priority rule D. all of the mentioned

6. Given a context free grammar E->E+E|E\*E|(E)||a, for the input string a+a\*a, which of the following may be a sentential form during the derivations.

A. E+a B. (E)\*E C. E+a\*E D. E+Ea\*

7. In parsing stack of LR parsing, we can find.

A. Only handles for reduction at the bottom of parsing stack.

B. Only non-terminals at the top of parsing stack.

C. Only terminals at the bottom of parsing stack.

D. Only state symbols at the top of parsing stack.

8. What data structure in a complier is used for managing information about variables and their attributes.

A. abstract syntax tree B. symbol table C. linked hash table D. parse table

9. In the bottom-up predictive parser.

A. Substrings of a right sentential form are handles.

B. viable prefixes always are found at the rightmost of a right-sentential form.

C. A reduction occurs when a handle is found at the top of parsing stack.

D. The input is accepted when the parsing table is empty.

10. S-attribute grammar is the attributed grammar, of which.

A. each attribute must be synthesized.

B. each attribute must be inherited.

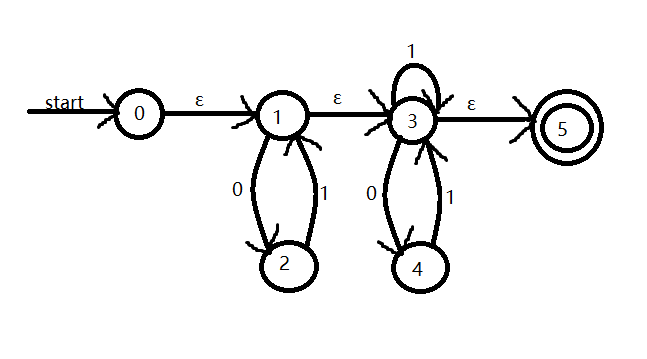
C. each attribute can be synthesized, or inherited.

D. each attribute can be inherited, which of a node on a parse tree is associated with the attributes from above or left node.

2. Consider the following grammar G[F]:F->F@F|(), please give an example string and its parse trees to prove that this grammar is ambiguous.

3. Consider the following grammar G[S]:S->(A)|a, A->SdA|S. Please give the rightmost derivation and leftmost derivation for the sentence '(ad(ada))', and circle the handle of the right sentential form '(Sd(ada))'.

4. Consider the following nondeterministic finite automata (NFA).



(1) Use the algorithm of subset construction to convert this NFA into deterministic finite automata (DFA). Complete the construction process in the following table and draw the DFA.

(2) Use the state minimization algorithm to minimize the number of states in this DFA. Please give the equivalent states in this DFA and draw the minimum-state DFA.

5. Consider the grammar G[S]:S->(R)|0S|0, R->R, S|S.

(1) Rewrite this grammar to left factor and eliminate left recursion.

(2) Construct First and Follow Sets for the nonterminals.

(3) Is this grammar the LL(1) grammar, give your reason.

(4) Construct the LL(1) parsing table.

6. Consider the grammar G[E]: ① E->@E+ ② E->@E- ③ E->ε

(1) Augmented the grammar G[E] to G[S], and draw the DFA of LR(0) items.

(2) Give your reasons to determine whether this grammar is a LR(0) grammar or a SLR(1) grammar.

(3) Construct the SLR(1) parsing table.

(4) Show the parsing stack and the action of SLR(1) parser for the input string '@@-+'.

7. Implement the following Syntax-Directed Translation. The grammar has four inherited attributes 'begin', 'next', 'true', 'false'. The synthesized attribute code for each symbol is its corresponding Three-Address-Code sequence.

Given the source code: do a=a+1 while a<b and a<c.

(1) Draw the Abstract Syntax Tree.

(2) According to the semantic rules, calculate the inherited attributes 'true', 'false', 'begin' and 'next', mark on the corresponding nodes to form the annotated syntax tree.

(3) According to the step (2) result and the synthesized attribute 'code', translate the annotated tree into Three-Address-Code.

E1.code E2.code E.code S.code do-stmt.code

Grammar Semantic Rules

do-stmt->do S do-stmt.begin=newlabel;

while E do-stmt.next=newlabel;

E.true=do-stmt.begin;

E.false=do-stmt.next;

do-stmt.code: Label do-stmt.begin || S.code || E.code || Label do-stmt.next

E->E1 and E2 E1.true=newlabel;

E1.false=E.false;

E2.true=E.true;

E2.false=E.false;

E.code=E1.code || Label E1.true || E2.code

E->id1<id2 E.code=if id1.name<id2.name goto E.true || goto E.false

S->id=id+num S.code=id.name=id.name+num.val