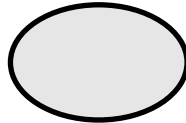


Total / 30



Please PRINT using keyboard letters :

Name :

Login :

@ucsc.edu

*No books ; No calculator ; No computer ; No email ; No internet ; No notes ; No phone. Neatness counts ! Do your scratch work elsewhere and enter only your final answer into the spaces provided.*

1. Given the grammar presented here, and using the style from the LALR(1) handout :
- (a) Construct the characteristic finite state machine (CFSM), sets of items and transition diagram, showing shifts, reductions, and acceptance. **[6✓]**
  - (b) Construct the FOLLOW sets. **[3✓]**
  - (c) Answer *yes* or *no* to each of the following questions : **[1✓]**

Is the grammar LR(0) ? \_\_\_\_\_ Is the grammar SLR(1) ? \_\_\_\_\_

- |    |                         |
|----|-------------------------|
| 0. | $S \rightarrow \$ E \$$ |
| 1. | $E \rightarrow f(A)$    |
| 2. | $E \rightarrow x$       |
| 3. | $A \rightarrow EA$      |
| 4. | $A \rightarrow$         |

2. Using **bison**, write a grammar to parse the following language. Do not show any semantic actions. Show only required declarations in section 1 (before the first %), and grammar rules (between the first and second %). Don't rearrange the grammar for the convenience of the AST constructor. Use left-associative rules wherever possible. Use right-associative rules only when necessary. For that part of this language which is poorly specified, assume the rules of C. **[5✓]**
- (a) A program is a sequence of one or more statements.
  - (b) A statement is an if statement or an expression.
  - (c) An if statement is the keyword **if**, followed by an expression in parentheses, followed by a statement.
  - (d) An if statement may optionally be followed by the keyword **else** and another statement.
  - (e) An expression is an identifier (which is a token) or a function call.
  - (f) A function call is an identifier, followed by a comma-separated list of zero or more expressions inside parentheses.

3. Draw the abstract syntax trees for each of the following code fragments. Assume the specifications of the parser project. **[3✓]**

<pre>if (a &lt; b) {     m = x * b;     f (x); }</pre>	<pre>int f (int i) {     return i * 3 + 2; }</pre>	<pre>a = b * (c + d * e);</pre>
--	--	---------------------------------

4. Using **flex**, define the following macros (section 1 before the first %): **[2✓]**
- (a) A **FRACTION** consists of one or more decimal digits, with an optional decimal point. The decimal point may precede all digits, follow all digits, or be between a pair of digits.
  - (b) An **EXPONENT** consists of the letter **E** or **e** (upper or lower case), followed by an optional plus or minus sign, followed by one or more decimal digits.

Multiple choice. To the *left* of each question, write the letter that indicates your answer. Write 'Z' if you don't want to risk a wrong answer. Wrong answers are worth negative points. [11✓]

number of correct answers		$\times 1 =$	$= a$
number of wrong answers		$\times \frac{1}{2} =$	$= b$
number of missing answers		$\times 0 =$	$0$
column total $c = \max(a - b, 0)$	11		$= c$

- The scanner **yylex** passes semantic information to the parser **yparse** using the external variable :  
(A) **yydebug**  
(B) **yyerror**  
(C) **yyval**  
(D) **yytext**
- Which statement is true about these languages ?  
(A)  $LR(0) \subset LALR(1) \subset SLR(1) \subset LR(1)$   
(B)  $LR(0) \subset LR(1) \subset SLR(1) \subset LALR(1)$   
(C)  $LR(0) \subset SLR(1) \subset LALR(1) \subset LR(1)$   
(D)  $LR(1) \subset LALR(1) \subset SLR(1) \subset LR(0)$
- The grammar  
 $A \rightarrow Axyz$   
 $A \rightarrow pqr$   
(A) is LR(0) but not SLR(1).  
(B) is SLR(1) but not LR(0).  
(C) is both LR(0) and SLR(1).  
(D) is neither LR(0) nor SLR(1).
- If there are  $n$  symbols in a regular expression, what is the worst possible number of states in the DFA ?  
(A)  $O(2^n)$   
(B)  $O(n)$   
(C)  $O(n \log_2 n)$   
(D)  $O(n^2)$
- The **flex** expression **ab|c\*** means :  
(A)  $((ab)|c)^*$   
(B)  $(a(b|c))^*$   
(C)  $(ab)|(c^*)$   
(D)  $a((b|c)^*)$
- Which of the following grammars is unambiguous, and will use up the most stack space when parsed with an LR(1) parser ?  
(A)  $A \rightarrow AA$   
 $A \rightarrow x$   
(B)  $A \rightarrow Ax$   
 $A \rightarrow x$   
(C)  $A \rightarrow xA$   
 $A \rightarrow x$   
(D)  $A \rightarrow xx$   
 $A \rightarrow x$
- We should **shift** if the precedence of the lookahead symbol is (x) than the precedence of the rule at the top of the stack ; or if they have the same precedence and are (y) associative.  
(A) (x) = higher, (y) = left.  
(B) (x) = higher, (y) = right.  
(C) (x) = lower, (y) = left.  
(D) (x) = lower, (y) = right.
- What is an example of an input that will be accepted by the following context free grammar ?  
 $A \rightarrow Ax$   
 $A \rightarrow y$   
(A) xxxxxxxxy  
(B) xyxyxyxyxy  
(C) yxxxxxxxxx  
(D) yyyyyyyxx
- An LR(1) parse table has one row for each state. The size of what set describes the number of columns ?  
(A)  $V_N$   
(B)  $V_N \cap V_T$   
(C)  $V_N \cup V_T$   
(D)  $V_T$
- Which of the following items in a state will cause a reduction ?  
(A)  $E \rightarrow \bullet E + T$   
(B)  $E \rightarrow E \bullet + T$   
(C)  $E \rightarrow E + \bullet T$   
(D)  $E \rightarrow E + T \bullet$
- What do the following statements do ?  
 $x = 3, 14; y = (3, 14);$   
Note : commas, not periods.  
(A)  $x = 14; y = 14;$   
(B)  $x = 14; y = 3;$   
(C)  $x = 3; y = 14;$   
(D)  $x = 3; y = 3;$