$Id: cmps104a-2009q4-exam2.mm,v 1.80 2009-11-13 19:38:51-08 - - $

page 1 page 2 page 3 Total / 31

Please print clearly :

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**	→	**$ X $**
1.	**X**	→	**X E**
2.	**X**	→	
3.	**E**	→	**E * P**
4.	**E**	→	**P**
5.	**P**	→	*** P**
6.	**P**	→	**m**

2. Using **bison**, write a grammar for the following language. You may make the grammar ambiguous if you correctly declare precedence and associativity. Clearly separate bison section 1 declarations from section 2 rules. Do not write any C code or semantic actions. Don't use any error productions. If the input is incorrect, the parser just crashes. The grammar follows: **[5✔]**
   (a) A program is one or more expressions separated by semicolons (`;`), not terminated by semicolons.
   (b) An expression consists of the plus (`+`) and multiply (`*`) operators with the same precedence and associativity as in C, each of which has a left and right expression operand.
   (c) An expression may be a minus sign (`-`) followed by an expression. This operator has the highest precedence and is right associative.
   (d) An expression is also an identifier, a string, or an expression in parentheses.
   (e) An expression also consists of: the **if** keyword, followed by an expression, followed by the **then** keyword, followed by an expression, followed by the **else** keyword, followed by an expression. This construct has the lowest possible precedence.
   (f) An identifier is a token consisting of one or more upper- or lower-case letters and digits, but may not begin with a digit.
   (g) A string is any sequence of characters not including a newline. It begins and ends with a double quote (`"`). If a double quote needs to appear in the string, it is doubled so as not to indicate the end of the string. Example: **`"foo'bar""baz"`**.
   (h) Spaces, tabs, and newlines are whitespace and are ignored.
   (i) Comments consist of a hash (#) followed by all characters up to but not including the newline character.

3. Write a grammar for **flex** that can scan the language in the previous question. Clearly show what goes in section 1 and section 2. Do not write any C code, except that semantic actions that recognize tokens should have an appropriate **return** statement. Do not write any code dealing with lexical information. You answer must be compatible with the previous question. Don't use any error patterns. If the input is not correct, the scanner just crashes. **[5✔]**

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		× 1 =	$= a$
number of wrong answers		× ½ =	$= b$
number of missing answers		× 0 =	0
column total $c = \max(a - b, 0)$	11		$= c$

1. Which expression will be true in C if exactly one bit in an `int` is on (1)?
   (A) `a&&!(a&(a-1))`
   (B) `a&&~(a&(a-1))`
   (C) `a||!(a&(a-1))`
   (D) `a||~(a&(a-1))`

2. If D is the set of languages recognizable by a DFA and N is the set of langauges recognizable by an NFA, then:
   (A) $D \subset N$
   (B) $D \equiv N$
   (C) $D \supset N$
   (D) None of the above.

3. Which of the following items was added to a state as part of its kernel during propagation of the gotos?[1]
   (A) $E \rightarrow \bullet\ E + T$
   (B) $E \rightarrow E\ \bullet + T$
   (C) $E \rightarrow E + \bullet\ T$
   (D) $E \rightarrow E + T\ \bullet$

4. The grammar:
   $A \rightarrow A\ x$
   $A \rightarrow$
   (A) is both LR(0) and SLR(1).
   (B) is LR(0) but not SLR(1).
   (C) is not LR(0) but is SLR(1).
   (D) is neither LR(0) nor SLR(1).

5. When converting infix expressions to abstract syntax trees:
   (A) operators are always leaf nodes.
   (B) operators are always root nodes.
   (C) operators are children of their operands.
   (D) operators are parents of their operands.

6. For a grammar $G = \langle V_N, V_T, P, S \rangle$, the symbols on the parsing stack are elements of:
   (A) V_N
   (B) V_T
   (C) $V_N \cap V_T$
   (D) $V_N \cup V_T$

7. The grammar:
   $E \rightarrow E + E$
   $E \rightarrow E * E$
   $E \rightarrow (E)$
   $E \rightarrow x$
   (A) is both LR(0) and SLR(1).
   (B) is LR(0) but not SLR(1).
   (C) is not LR(0) but is SLR(1).
   (D) is neither LR(0) nor SLR(1).

8. What is true about classes of 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) $LR(1) \subset LALR(1) \subset SLR(1) \subset LR(0)$

9. What kind of garbage collector is unable to handle cycles in the data structure?
   (A) concurrent
   (B) copying
   (C) mark and sweep
   (D) reference counting

10. If we perform the subset algorithm on an NFA to generate a DFA, what is the worst case number of states in the DFA, assuming n states in the NFA?
    (A) $O(n)$
    (B) $O(n^2)$
    (C) $O(2^n)$
    (D) $O(n \log_2 n)$

11. Earley's parsing algorithm for a context-free language using an ambiguous grammar runs in what time? Assume there are n tokens in the input stream.
    (A) $O(n)$
    (B) $O(n^2)$
    (C) $O(n^3)$
    (D) $O(n^4)$

---

1. Footnote added after test was written: Question 3 is incorrectly stated, so any of (B), (C), or (D) is correct. Item (A) was added during the closure operation. Items with a dot at the beginning of the rule were added during the closure operation. Items with a dot after some symbol were added during propagation of the gotos.