$Id: cmps104a-2011q4-exam1.mm,v 1.19 2011-10-20 19:08:47-07 - - $

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. Using as few states as possible, draw ***deterministic*** finite αὐτόματα for each of the following **flex** regular expressions. Do not show garbage states. Draw a separate transition for each symbol. **[5✔]**

   (i) `a|bc+`

   (ii) `(a|b)*c*`

   (iii) `a+b*c`

   (iv) `x[3-5]*x`

   (v) `(a|b)+`

2. Write some **flex** regular expressions as described by the following : **[5✔]**

   (i) An identifier consists of one ore more upper- or lower-case letters, digits, and underscores, but must begin with a letter. An underscore, if present, must be preceded and followed by a letter or a digit.

   (ii) A number consists of one or more digits with an optional decimal point. If the decimal point is present, it must be preceded and followed by a digit.

   (iii) A string constant begins with a double quote mark (") and may have zero or more character denotations in between. A character denotation is any character that is not a quote or a newline.

   (iv) A comment in C or Java which begins with a double slash and continues until end of line, not including the newline character at the end of the line.

   (v) A single pattern which recognizes one of the reserved words **if**, **then**, **else**, or **fi**.

3. Using ***Thompson***'s construction, convert the following `flex` regular expression into a ***nondeterministic*** finite αὐτόματον. Draw it to the right of the table, and number each one starting at 0. Call the initial state state 0. Then fillin the table with the ε-closure of each state. Scribble out state numbers that are not needed. **[5✔]**

   `abc|de*`

| $s$ | ε-closure ($s$) |
|---|---|
| 0 | |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |
| 11 | |

4. Define an unambiguous grammar similar to ETF, which contains identifiers, parentheses, and two operators. The operator `&` is right associative and has a higher precedence than the operator `@`, which is also right associative. **[2✔]**

5. For the ETF grammar given at the left, draw abstract syntax trees for each of the following expressions. **[3✔]**

   $E \rightarrow E + T$
   $E \rightarrow T$
   $T \rightarrow T * F$
   $T \rightarrow F$
   $F \rightarrow (E)$
   $F \rightarrow i$

   (i) `a*b+c*d`

   (ii) `a+b*c+d`

   (iii) `a+b+c+d`

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. What is the proper ordering of these functions ?
   (A) parsing then scanning then symbol table
   (B) scanning then parsing then symbol table
   (C) scanning then symbol table then parsing
   (D) symbol table then scanning then parsing

2. What kind of Chomsky grammar is used to build a scanner ?
   (A) context-free
   (B) context-sensitive
   (C) regular
   (D) unrestricted

3. What kind of Chomsky grammar is used to build a parser ?
   (A) context-free
   (B) context-sensitive
   (C) regular
   (D) unrestricted

4. The set of languages recognizable by a DFA is ____ the set of languages recognizable by an NFA.
   (A) a proper subset of
   (B) a proper superset of
   (C) the same as
   (D) none of the above

5. If a ***nondeterministic*** finite αὐτόματον is constructed from a regular expression $r$ and used to scan a string $s$, how much time will it take to perform the scan ? (The notation $|r|$ means : the length of $r$.)
   (A) $O(2^{|r|})$
   (B) $O(|r|)$
   (C) $O(|s|)$
   (D) $O(|r| \times |s|)$

6. If a ***deterministic*** finite αὐτόματον is constructed from a regular expression $r$ and used to scan a string $s$, how much time will it take to perform the scan ? (The notation $|r|$ means : the length of $r$.)
   (A) $O(2^{|r|})$
   (B) $O(|r|)$
   (C) $O(|s|)$
   (D) $O(|r| \times |s|)$

7. The following grammar is :
   $A \to A + A$
   $A \to A * A$
   $A \to i$
   (A) LL(1)
   (B) LR(1)
   (C) ambiguous
   (D) regular

8. How long should it take to find something in a hash table with $n$ items in it ?
   (A) $O(1)$
   (B) $O(\log_2 n)$
   (C) $O(n)$
   (D) $O(n \log_2 n)$

9. What is prohibited in a ***deterministic*** finite αὐτόματον ?
   (A) cycles in the graph
   (B) epsilon transitions
   (C) multiple final states
   (D) all of the above

10. Which parsing action pops the right hand side of a rule off of the parsing stack, performs a semantic action, then pushes the left hand side of the same rule onto the stack ?
    (A) accept
    (B) reduce
    (C) scan
    (D) shift

11. The intersection of the set of ambiguous grammars with the set of LALR(1) grammars is :
    (A) a non-empty subset of the set of context free grammars.
    (B) the empty set.
    (C) the same as the set of context free grammars.
    (D) the same as the set of regular grammars.