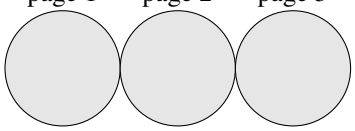
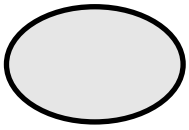
$Id: cmps104a-2010q4-exam1.mm,v 1.72 2010-10-21 16:46:31-07 - - $

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. Draw an abstract syntax tree for each of the following C expresions. **[3✔]**

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

2. What two things a permitted in an NFA that are not permitted in a DFA ? State each and show a small FA as an example of each. **[2✔]**

3. Using the ***minimum*** possible number of states, draw ***deterministic*** finite αὐτόματα for the following **`flex`** regular expressions. **[5✔]**

   (i) `xy*|cd`

   (ii) `(a|b)+y+`

   (iii) `ab|c*`

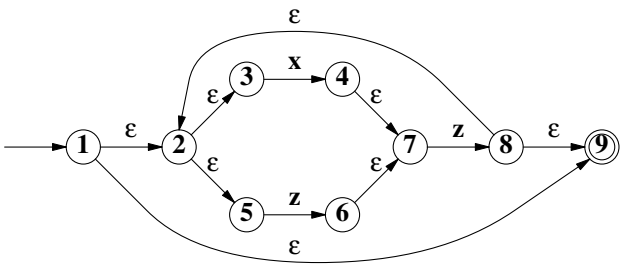
   (iv) `ab+cd|e`

   (v) `a?b?c?`

4. The following NFA was created from Thompson's construction.
   (a) What was the original regular expression ? You answer may contain only letters from the NFA, and some collection of the symbols ε, `|`, `(`, `)`, and `*`. Use as few parentheses as possible. **[1✔]**
   (b) Fill in the table showing the ε-closure foreach state. **[2✔]**
   (c) Using the subset construction, draw the equivalent DFA. ***Do not minimize.*** Draw your anwer below the NFA. In each state of the DFA, write the corresponding states from the NFA. **[2✔]**
   (d) Draw a third diagram showing a minimized DFA. Do your scratch work elsewhere and show only the final answer. Use any algorithm (or guesswork) that you like. **[1✔]**

state s	(b) ε-closure (s)
1	
2	
3	
4	
5	
6	
7	
8	
9	

(a) **Original regex :**



(c) **DFA (do not minimize) :**

(d) **Minimized DFA :**

5. Using `bison`, write an ***unambiguous*** grammar defining a `thunk` to be a sequence of one or more `bug` connected by operators, either `'+'` or `'*'`, both of which are right associative. A `bug` is a sequence of one or more `IDENT` connected by the `AND` operator, which is left associative and has a higher priority than the others. Do not show semantic actions. **[2✔]**

```
%token IDENT AND
%start thunk
%%
```

6. Using `flex`, define the following tokens. Do not define, assume, or use macros. **[2✔]**

   (i) A string which begins with either a double quote (`"`) or a single quote (`'`) followed by any number of characters not including newline up to another quote mark. The trailing quote must be the same as the opening quote, and the bounding quotes may not appear within the string.

   (ii) A floating point number whose fractional part begins and ends with a decimal digit and possibly contains an optional decimal point (`.`) within. It is followed by an optional exponent, which consists of the upper case letter `E`, an optional sign, and 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		× 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 of the following **flex** regexes will match either an **a**; or, alternatively, a **b** followed by a **c**?
   (A) `a*bc`
   (B) `a+bc`
   (C) `a?bc`
   (D) `a|bc`

2. If a function returns a `uint32_t`, the actual possible range of values is:
   (A) -2^{31} to $2^{31} - 1$
   (B) 0 to $2^{32} - 1$
   (C) 0 to 31
   (D) 1 to 2^{32}

3. Putting reserved words into a **flex** grammar instead of embedding them in the string table make the DFA use:
   (A) less CPU time
   (B) less memory
   (C) more CPU time
   (D) more memory

4. Which one of the following **flex** actions is obviously wrong?
   (A) `return "=";`
   (B) `return '=';`
   (C) `return *yytext;`
   (D) `return EQ;`

5. Compiling following C program and then running `a.out` will print what? (Choose the most reasonable answer.)
```
void main () {
   printf ("%p\n", main);
}
```
   (A) `0x4004f0`
   (B) `Segmentation fault (core dumped)`
   (C) `UHìHÇÆð@`
   (D) `a.out: Command not found.`

6. If the expression `(a*((b+c)+d))` retains the same meaning with parentheses removed, that means that the operators are (a)-associative and that `*` has a (b) precedence than `+`.
   (A) (a) = left, (b) = higher
   (B) (a) = left, (b) = lower
   (C) (a) = right, (b) = higher
   (D) (a) = right, (b) = lower

7. The **flex** regex `ab+|c` means the same as:
   (A) `((ab)+)|c`
   (B) `(a(b+))|c`
   (C) `(ab)(\+|c)`
   (D) `a((b+)|c)`

8. Which grammar is unambiguous, matches an arbitrarily large number of terminal symbols, and uses up the smallest amount of stack space when doing so?
   (A) **A** → **A A**
       A → **y**
   (B) **A** → **A x**
       A → **y**
   (C) **A** → **x A**
       A → **y**
   (D) **A** → **x x**
       A → **y**

9. Which grammar will accept a single **y** ***followed*** by an arbitrarily large number of **x**s?
   (A) **A** → **A x**
       A → **y**
   (B) **A** → **A y**
       A → **x**
   (C) **A** → **x A**
       A → **y**
   (D) **A** → **y A**
       A → **x**

10. A parser uses what level of grammar in the Chomsky hierarchy?
    (A) recursively enumerable
    (B) context sensitive
    (C) context free
    (D) regular

11. The fact that `int` is a keyword, not an identifier, is determined by:
    (A) lexical analyzer
    (B) parser
    (C) symbol table module
    (D) code generator