

# CMSC330 Fall 2010 Midterm #1

Name \_\_\_\_\_

Discussion Time (circle one): 9am 10am 11am 12pm 1pm 2pm

Do not start this exam until you are told to do so!

## Instructions

- You have 75 minutes to take this midterm.
- This exam has a total of 100 points, so allocate 45 seconds for each point.
- This is a closed book exam. No notes or other aids are allowed.
- If you have a question, please raise your hand and wait for the instructor.
- Answer essay questions concisely using 2-3 sentences. Longer answers are not necessary and a penalty may be applied.
- In order to be eligible for partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

	Problem	Score
1	Ruby features	/12
2	Regular expressions & finite automata	/12
3	RE to NFA to DFA	/20
4	DFA minimization	/10
5	OCaml Types	/18
6	Ruby programming	/28
	Total	/100

1. (12 pts) Ruby

a. (6 pts) List 3 features of Ruby that are often found in scripting languages.

b. (2 pts each) What is the output (if any) of the following Ruby programs? Write FAIL if code does not execute.

i. `puts "Found" if "College Park" =~ /[a-z]+/`      **Output =**  
`puts $1`

ii. `a = [ ]`      **Output =**  
`a[2] = 1`  
`puts a`

iii. `a = { 1 => 2 }`      **Output =**  
`a.keys.each{ |x| puts "#{ a[x] }" }`

2. (12 pts) Regular expressions and finite automata

a. (4 pts) Give a regular expression for all binary numbers (strings of 0s and 1s) beginning and ending with 1 (e.g., 1, 111, 10011).

b. (4 pts) Give a DFA for all binary numbers (strings of 0s and 1s) beginning and ending with 1 (including 1).

c. (4 pts) Explain why adding a **not** operator to the standard **union**, **concatenation**, and **closure** operators does not make regular expressions more powerful.

3. (20 pts) RE to NFA to DFA

a. (8 pts) Create a NFA for the regular expression  $abc^*d$  using the method described in lecture. Remember precedence is in the order: closure  $>$  concatenation  $>$  union.

b. (12 pts) Apply the subset construction algorithm discussed in class to convert the following NFA to a DFA. Show the NFA states associated with each state in your DFA.

