

CMSC 330, Spring 2008, Midterm 1 Practice Problems

1. Programming languages
 - a. Explain how goals for programming languages have changed since the 1960's.
 - b. List 2 desirable attributes for a programming language where Ruby is better than C. Explain why.
 - c. List 2 methods for executing a program. Which method is used by Ruby?
 - d. Explain why Ruby fits the definition of a scripting language.

2. Ruby basics
 - a. Write a Ruby method foo that takes an integer as a parameter. Call foo with 2 as its argument. Circle & label the formal and actual parameters in your code.
 - b. Using different Ruby control statements, write 4 code fragments that iterate from i=1 to i=10.
 - c. Explain the difference between explicit and implicit variable declarations.
 - d. List two advantages of static types.
 - e. Using Ruby, write a class Teacher that contains an integer field students and an integer field totalStudents that is shared across all objects of class Teacher.
 - f. Give an example of reference copy in Ruby.
 - g. Give an example of testing for structural equality in Ruby.

3. Ruby advanced features
 - a. Describe the strings accepted by the Ruby regular expression `/3{2}/` ?
 - b. Describe the strings accepted by the Ruby regular expression `/[A-Z]/` ?
 - c. Describe the strings accepted by the Ruby regular expression `/[A-Z]*[0-9]/` ?
 - d. Describe the strings accepted by the Ruby regular expression `/0$/` ?
 - e. Describe the strings accepted by the Ruby regular expression `^./` ?
 - f. What is the output of the following Ruby program?

```
“CMSC 330” =~ /[0-9]+/  
puts $1  
puts $2
```
 - g. What is the output of the following Ruby program?

```
a = [4,5,6]  
a[5] = 7  
a.delete_at(1)  
a.push(1)  
puts a  
puts a.pop
```
 - h. What is the output of the following Ruby program?

```
if “CMSC 330” =~ /1/ then  
  puts “t”  
elsif “CMSC 330” !~ /1/ then  
  puts “f”  
else  
  puts “n”  
end
```

- i. What is the output of the following Ruby program?

```
a = ["c", "b", "a"]
puts a
b = a
a.sort!
puts b
```
- j. What is the output of the following Ruby program?

```
a = "CMSC 330 CMSC 351"
b = a.scan(/[A-Z]+/)
puts b
a.scan(/[0-9]+ [A-Z]+/) { |x| puts x }
```
- k. What is the output of the following Ruby program?

```
a = {4 => 6, 5 => 7}
puts a[4]
puts a[6]
puts a.values
```
- l. What is the output of the following Ruby program?

```
h = Hash.new(0)
h["a"] = h["b"]
h["b"] = 7
h["c"] += 2
puts "#{h["a"]} #{h["b"]} #{h["c"]}"
```
- m. What is returned by “file = File.new(filename, "r"); lines = file.readlines();”?
- n. What is returned by “x = ARGV[0];”?
- o. Write a Ruby function foo that takes a code block and executes it twice.

4. Ruby programming

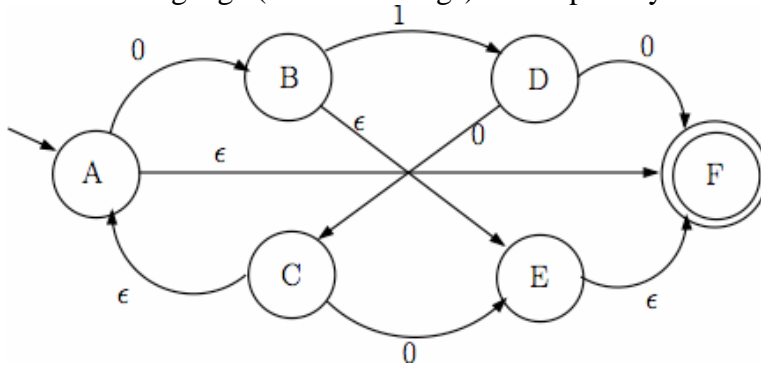
- a. Write a Ruby program that reads in lines of input from \$stdin and remembers all integers (consecutive digits) encountered. The program should stop and print out the list of integers in sorted order (from smallest to largest) when the word “Done!” is encountered.
- b. Write a Ruby program that reads the name of a text file from the command line, opens the file, reads every line of text in the file, and prints only the lines that contain exclusively the following characters: uppercase and lowercase letters, digits, and underscore. For example, lines that contain space or punctuation should not be printed.

5. Regular expressions and languages

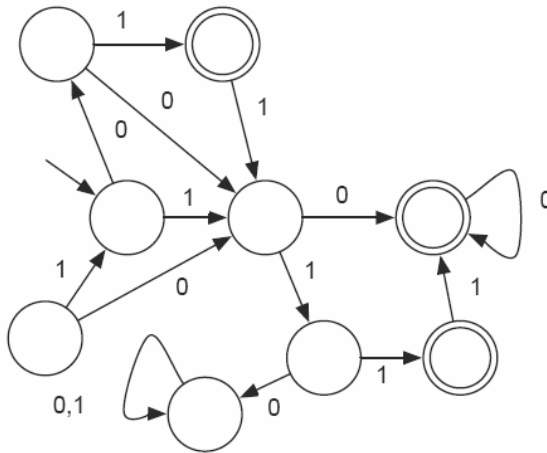
- a. From the perspective of formal language theory, what is a language?
- b. Given the language $A = \{“aa”, “c”\}$ and $B = \{“b”\}$, what is the language AB ?
- c. Given the language $A = \{“aa”, “c”\}$, what is the language A^0 ?
- d. Given the language $A = \{“aa”, “c”\}$, what is the language A^2 ?
- e. Given the language $A = \{“aa”, “c”\}$, what is the language A^* ?
- f. Give a regular expression for all binary numbers including the substring “101”.
- g. Give a regular expression for all binary numbers with an even number of 1’s.
- h. Give a regular expression for all binary numbers that don’t include “000”.

6. Finite automata

- When does a NFA accept a string?
- How long could it take to reduce a NFA with n states and t transitions to a DFA?
- Give a NFA that only accepts binary numbers including the substring "101".
- Give a NFA that only accepts binary numbers that include either "00" or "11".
- Give a NFA that only accepts binary numbers that include both "00" and "11".
- What language (or set of strings) is accepted by the following NFA?



- Compute the ϵ -closure of the start state for each of the NFA above.
- Give a DFA that only accepts binary number with an odd number of 1's.
- Give a DFA that only accepts binary numbers that include "000".
- Give a DFA that only accepts binary numbers that don't include "000".
- What language (or set of strings) is accepted by the following DFA?



- For each regular expression: 1^* , $(0|01)^*0$, $(0^*|1)^*1$, $(10^*|01)^*$
 - Reduce the RE to an NFA using the algorithm described in class.
 - Reduce the resulting NFA to an DFA using the subset algorithm.
 - Show whether the DFA accepts / rejects the strings "1", "11", "101"
 - Minimize the resulting DFA using Moore reduction
 - Are any 2 of the minimized DFA identical?