Name:_____

CMSC330 Summer 2009
Midterm 1: Ruby and Regular Languages

Instructions: Show your work for all problems. If you need extra space, ask for a sheet of paper and label your work (i.e. which problem your work corresponds to). For two points extra credit, list one thing that you like about this course and one thing that you do not like about this course on this page.

+1 for each listing

1. Typing

a. Explain the difference between dynamic types and static types without using the words "dynamic" or "static". (2pts)

static: known at compile time

dynamic: determined at run-time

b. Give an example of a language that uses static types and one that uses dynamic types (and label which language uses which). (2pts)

static: Java, C/C++, etc.

dynamic: Ruby [illegible], etc.

2. Regular Expressions and languages

a. Give a regular expression for the following Ruby regular expressions. You may use the symbol Σ to stand for "any character": (8 pts)

i. /phone[abc]/ Σ^*phone(a|b|c)Σ^*

ii. /^phone/ phoneΣ^*

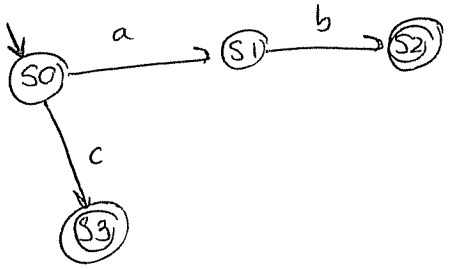
iii. /^phone.*/ phoneΣ^*

iv. /^phone$/ phone

b. Give a DFA/NFA for the following regular expressions. Do not include dead states. State whether you are drawing an NFA or a DFA (circle one). (8pts)

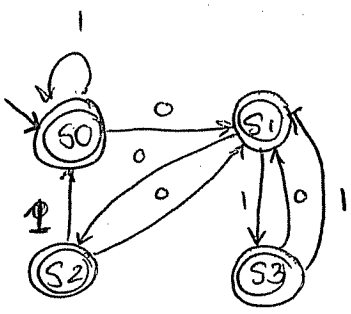
i. ab|c

DFA NFA

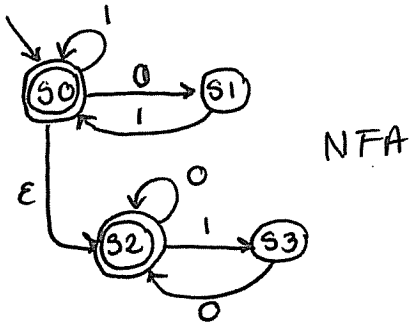


ii. (01|1)*(10|0)*

DFA NFA



DFA



NFA

c. What is the language of the following regular expressions? Give an English description. (4 pts)

i. A = {"abc", "bc"}

The strings abc and bc only

ii. (010|10|0)*

no 1s next to eachother

d. Give a regular expression for the following languages: (6 pts)

i. All binary numbers that contain at least two 0s

$(0|1)^*0(0|1)^*0(0|1)^*$

ii. All binary numbers that do not contain the substring 00

$(101|01|1)^*|0$

3. Ruby programming
(24 pts)
Write a Ruby program called phone.rb to process phone numbers contained in a text file. We are interested in recording the number of times each area code appears in the file. Your code should open a file where the file name is provided in the first argument on the command line, as:
%ruby -w phone.rb input_file.txt
You need to read in the file and find all occurrences of valid phone numbers in the following example format:
(555)-555-5555
There should be no whitespace or other characters between any digits, parentheses, or dashes as shown above. You should count the number of times each different area code appears in the input file. After you have read the file,

you should print out the area codes and the number of times each area code appeared in sorted order by area code, each area code on a new line followed by a comma, and then the count. Below are example inputs and outputs:

Assume each line in file contains only 1 valid # phone #

Input	Output
(301)-555-5555 (301) This is random text here 555-5555. (434)-555-5555 and some more text(434)-555-5555.	301,1 434,2
(703)-555-5555 (301)-555-5555 (703)-555-5555 (202)-555-5555 (301)-555-5555	202,1 301,2 703,2

Write code for phone.rb below:

```
file = File.new(ARGV[0], 'r')
lines = file.readlines()
lines.each {|line|
```

6 +1 readlines +1 lines.each +1 ARGV[0] +1 "r" +1 File.new/open +1 syntax

```
regex = /\((\d{3})\)-\d{3}-\d{4}/
hash = {}

if line =~ regex
    if hash[$1] == nil
        hash[$1] = 1
    else
        hash[$1] += 1
    end
end
}
```

6 +1 =~ +1 // or Regex.new +1 if +1 for back ref. +2 for correct RE

6 +1 for hash init. +2 for += +1 for syntax (correct ends) +1 for checking nil

```
hash.keys.sort.each {|key,value|
    puts "#{key}, #{value}"
}
```

6 +1 for |key,value| +1 for sort +1 for each +1 for syntax +2 for correct print

4. Finite Automata

a. What is the order of the runtime for a DFA to accept a string s? You may assume that s has n characters and provide the big-O notation, or you may explain in English what the runtime is based on the length of the input. (2pts)

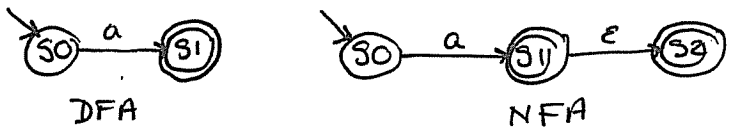
linear

b. Imagine we have an NFA and 100,000 strings for which we want to see if the NFA accepts. Name one reason why we would want to convert the NFA into a DFA in this scenario. (2pts)

NFA → DFA conversion is overhead but DFA will run in linear time based on size of input string.

c. Assume we have a DFA and an NFA for the same regular expression, neither of which have any dead states. In terms of the number of states D in the DFA and N in the NFA, which of the following is true (circle one or more)? (2pts)

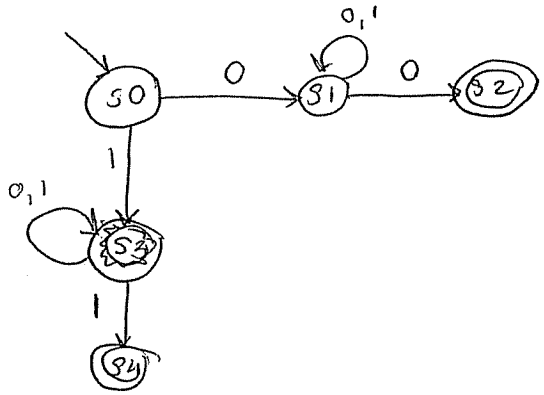
1) D < N
2) D > N
3) D = N

1): 

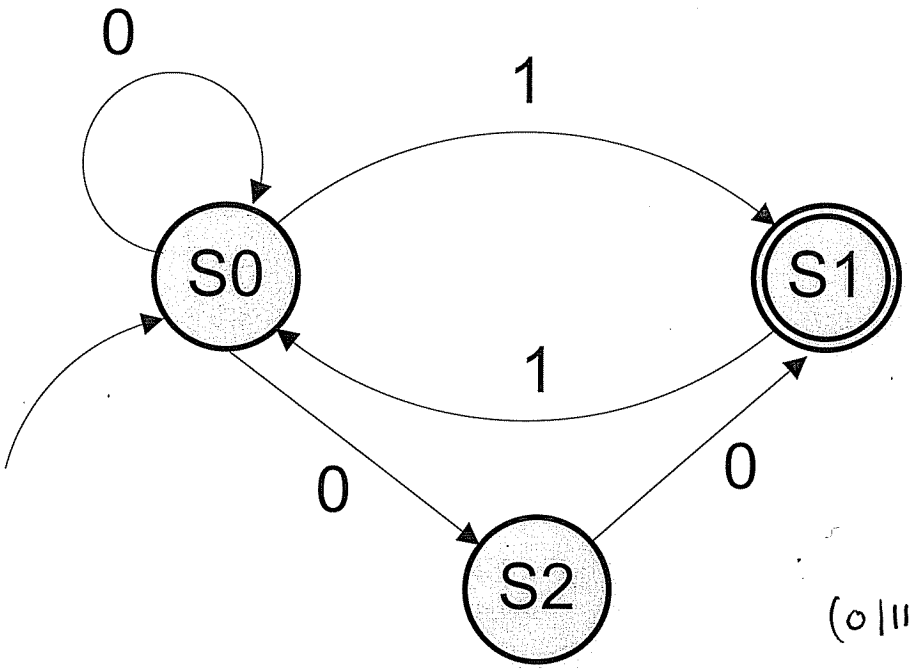
②: by definition of # of states possible under NFA → DFA conversion

③ by definition of DFA, every DFA is also an NFA

d. Draw a NFA for binary numbers that start and end in the same digit. Do not include dead states. (4pts)



e. Give a regular expression for the language represented by the ~~DFA~~ NFA below. (4pts)



$(0|11|001)^*(1|00)$

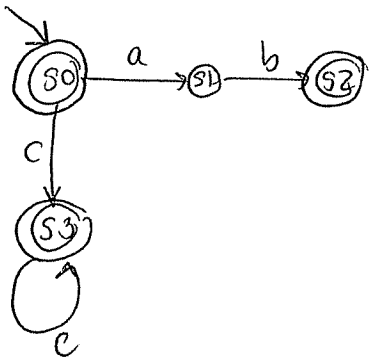
Answer to 4.e: _____

f. What does "non-deterministic" and "deterministic" mean in the context of an NFA/DFA(non-deterministic finite automata/deterministic finite automata)? (2pts)

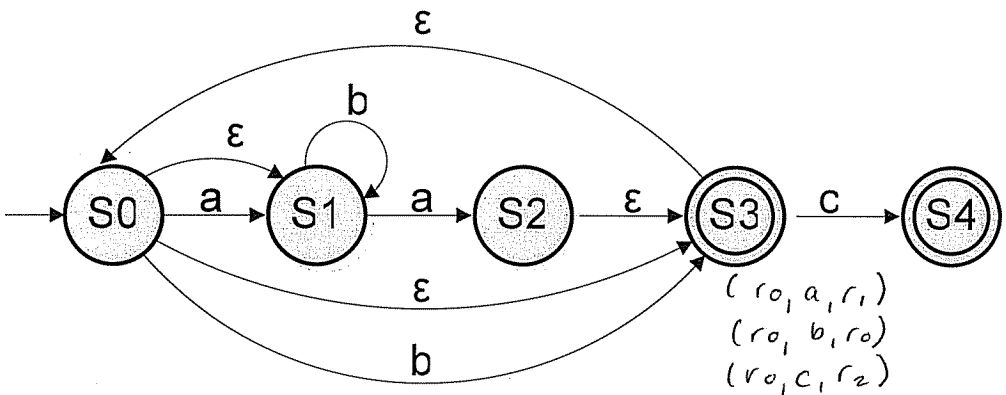
- deterministic : one possible path for string acceptance
- non-deterministic : more than 1 path explored

5. Reductions and conversions

a. Reduce the regular expression ab|c* to a NFA. You may use the algorithm given in class. Do not include dead states. (8pts)



b. Reduce the following NFA to a DFA using the construction given in class. Do not include dead states. (10 pts)



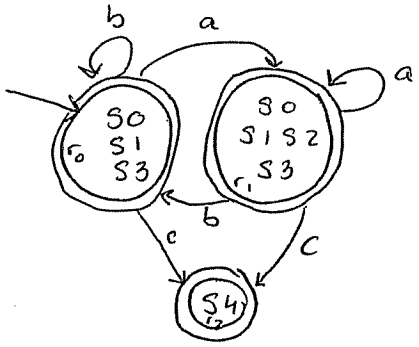
(r_0, a, r_1)
(r_0, b, r_0)
(r_0, c, r_2)
(r_1, a, r_1)
(r_1, b, r_0)
(r_1, c, r_2)

$R = \{ \underbrace{\{S0, S1, S3\}}_{r_0}, \underbrace{\{S0, S1, S2, S3\}}_{r_1}, \underbrace{\{S4\}}_{r_2} \}$

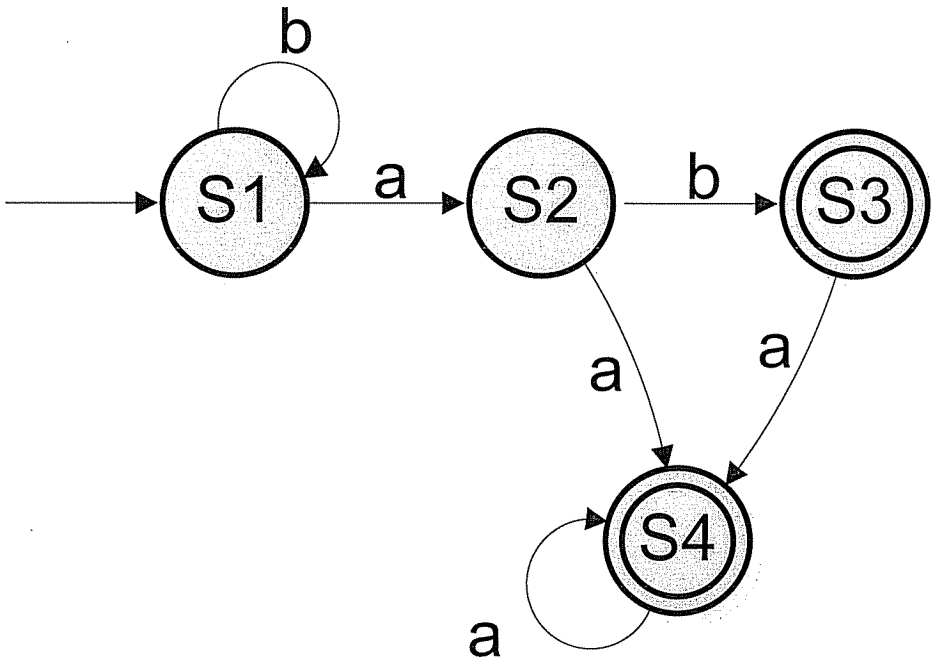
$M(r_0, a) = S1, S2 \rightarrow S1, S2, S3, S0$
$M(r_0, b) = S3, S1 \rightarrow S1, S3, S0$
$M(r_0, c) = S4 \rightarrow S4$

$M(r_1, a) = S1\,S2 \rightarrow r_1$
$M(r_1, b) = S1, S3 \rightarrow S1\,S3 \rightarrow r_0$
$M(r_1, c) = S4 \rightarrow r_2$

$M(r_2, a) = \emptyset$
$M(r_2, b) = \emptyset$
$M(r_2, c) = \emptyset$



c. Minimize the DFA below. You may use the algorithm provided in class. Do not include dead states. (8pts)



$P \neq 3$

$P1 = \{S1, \cancel{S2}\}$ $P3 = \{S2\}$

$P2 = \{S3, S4\}$

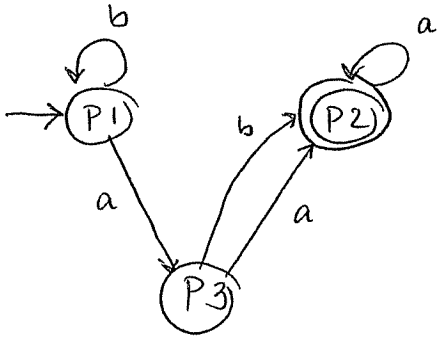
$P1_a$ = S1 → S2, S2 → P1, S2 → S4, S4 → P2

$P1_a = P1_b = \cancel{?}$ ✓

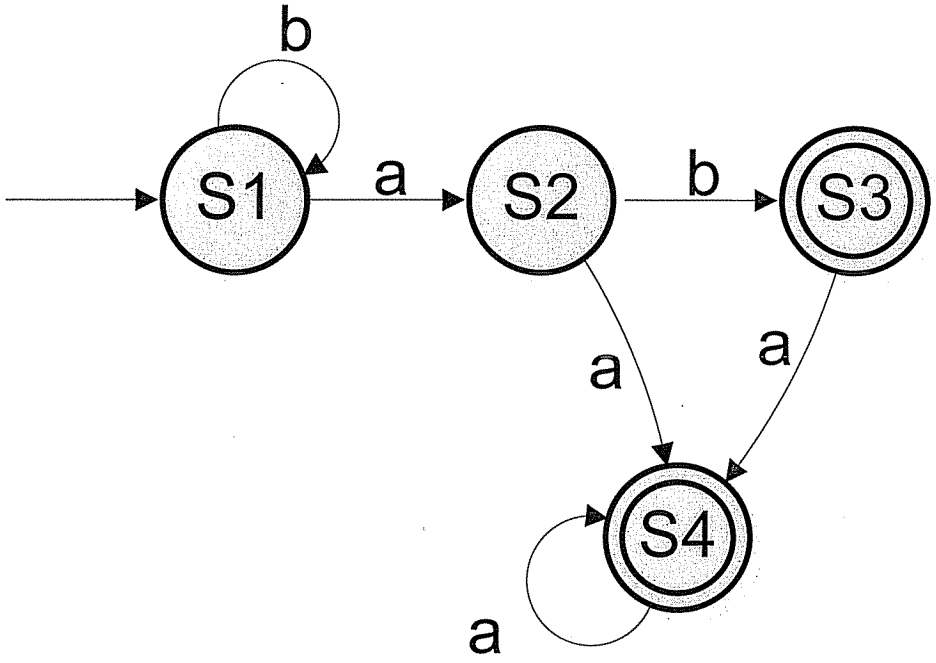
$P2_a$ = S3 → S4, S4 → S4 ✓

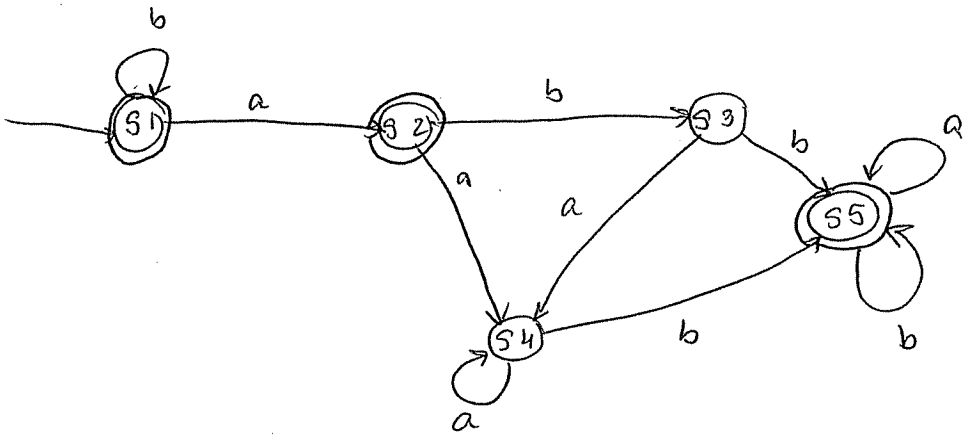
$P2_b$ = S3 → ∅, S4 → ∅ ✓

$P3_a = \checkmark$



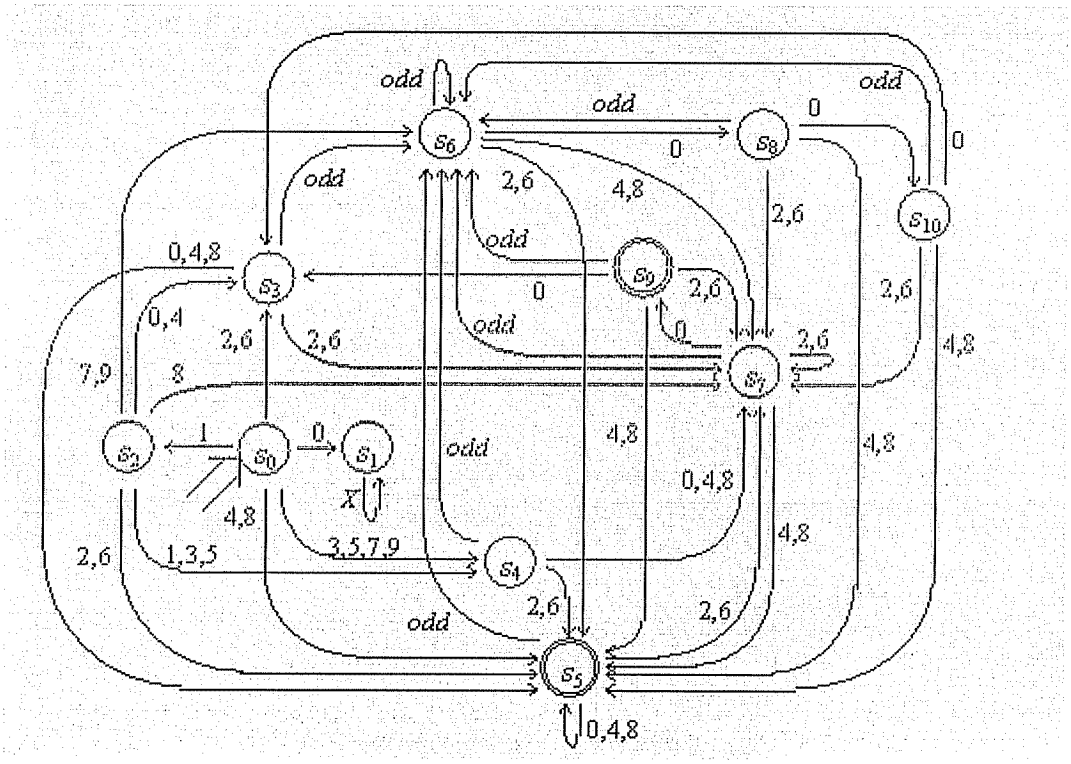
d. Give the complement of the DFA below (draw the complement DFA). (4pts)





Extra credit (for one point each):

1. What language does the following automation recognize? Hint: it accepts 2000 and 2008 but not 2009.



leap years

2. Draw an NFA that accepts all binary strings with an even number of zeros, with any number of 1s in between. Example strings in the language: 010, 00, 0101, 10101, 01111000, etc.