CMPT 413 - SPRING 2008 - Midterm #1

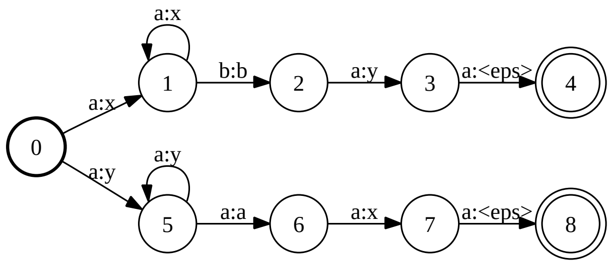
Please write down "Midterm #1" on the top of the answer booklet.

When you have finished, return your answer booklet along with this question booklet.

(1) (5pts) **Subsequential finite-state transducers**

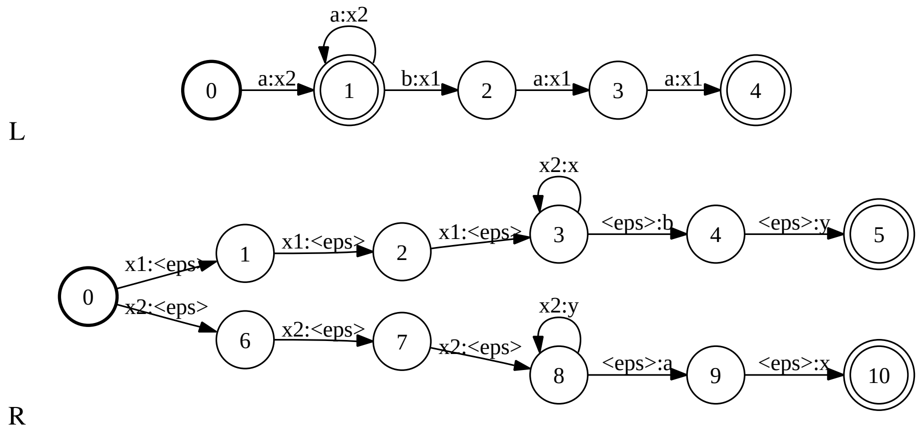
a. You are given the following definition of a regular relation: for any $n \geq 1$ if the input is $a^n baa$ then the output is $x^n by$, and if the input is a^{n+3} then the output is $y^n ax$.
For this regular relation, provide a finite-state transducer that is *not* subsequential.

Answer:



b. Provide two subsequential FSTs L and R such that the composition of L and R, $L \circ R$ captures the same regular relation. We assume that the output of L is reversed before it is given as input to R.

Answer:



(2) (7pts) **Rewrite Rules**

A rewrite rule R where $R = \alpha \rightarrow \beta \;/\; \lambda\_\; \rho$ matches α in the input and converts each match into β if the regular expression $\lambda\alpha\rho$ matches the input string. α, β, λ and ρ are regular expressions. In this question we will assume that the rewrite rules are obligatory and left to right. The alphabet is denoted by the set Σ, and every string constructed from this alphabet is matched by the regular expression Σ^*.

a. You are given two input strings: *#begˆing#* and *#singˆing#*. To each of these two strings, apply the rewrite rules shown below in sequence, where the output of the first rewrite rule is used as input to the next rewrite rule, and so on. Provide the output of each stage.
If the output is unambiguous, simply provide the output string. If the output is ambiguous, then provide the output either as a finite-state machine or a regular expression.

1. $r = \epsilon \rightarrow\, >\, /\ \Sigma^*\_\_\ \hat{}\,ing\#$
which marks locations in the string where the right context matches.

Answer: output: *#beg* > ˆ*ing#* and *#sing* > ˆ*ing#*

2. $f = \epsilon \rightarrow (<1 \mid <2) \;/\; (\Sigma \cup \{>\})^* \_\_ >?g>?\hat{}$
   which marks places where a replacement could occur, where regexp ($a|b$) indicates either a or b, and a? indicates an optional a.

Answer: output: *#be (<1 | <2) g > ˆing#* and *#sin (<1 | <2) g > ˆing#*

3. $replace_1 = g \rightarrow gg \;/\; <1\_\_>$
   which matches a regexp in the input and converts it to an output regexp.

Answer: output: *#be (<1 gg | <2 g) > ˆing#* and *#sin (<1 gg | <2 g) > ˆing#*

4. $replace_2 = > \rightarrow \epsilon \;/\; \Sigma^* \_\_ \Sigma^*$
   which removes all instances of >

Answer: output: *#be (<1 gg | <2 g) ˆing#* and *#sin (<1 gg | <2 g) ˆing#*

5. $l_1 = <1 \rightarrow \epsilon \;/\; \#\Sigma^*[\hat{}n]\_\_ \epsilon$
   which keeps the conversion to the output regexp if the left context matches.

Answer: output: *#be (gg | <2 g) ˆing#* and *#sin <2 g ˆing#*

6. $l_2 = <2 \rightarrow \epsilon \;/\; \#(\{\epsilon\} \cup \Sigma^* n)\_\_ \epsilon$
   which retains the input if the left context does not match.

Answer: output: *#beggˆing#* and *#singˆing#*

b. If x and y are rewrite rules then we define $x \circ y$ as the composition of the FST for x with the FST for y. Provide a single rewrite rule that is equivalent to the following composition:

$$r \circ f \circ replace_1 \circ replace_2 \circ l_1 \circ l_2$$

Answer:

$$g \rightarrow gg \;/\; [\hat{}n] \_\_ \hat{}ing\#$$

(3) (3pts) **Edit distance**: Assume insertion of a character has cost 1, deletion has cost 1, and substitution of one character for another has cost 2.

a. What is the minimum edit distance value between target word *goal* and source word *hole*?

b. The following is a visual display of one possible alignment between target word *goal* and source word *hole*.

```
g o a l _
  |   |
h o _ l e
```

The 1st line of the visual display shows the *target* word and the 3rd line shows the *source* word. An insertion in the target word is represented as an underscore ‘_’ in the 3rd line aligned with the inserted letter in the 1st line. Deletion from the source word is represented as an underscore ‘_’ in the 1st line aligned with the corresponding deleted character in the source on the 3rd line. Finally, if a letter is unchanged between target and source then a vertical bar (the pipe symbol ‘|’) is printed aligned with the letter in the 2nd line.

Using the above notation for alignments, provide any other possible alignments that have the same edit distance as the alignment shown above.

Answer:

```
levenshtein distance = 4
alignment number 1 for [4,4]:
_ g o a l _
    |   |
h _ o _ l e

alignment number 2 for [4,4]:
g _ o a l _
    |   |
_ h o _ l e

alignment number 3 for [4,4]:
g o a l _
  |   |
h o _ l e

total of 3 alignments
```