A. Finite automata, regular expressions

1. Give regular expressions describing the following languages:
   
   (a) The set of strings of \{a, b\}∗ starting with a and ending with b.
   (b) The set of strings of \{a, b\}∗ containing two consecutive a’s.
   (c) The set of strings of \{a, b\}∗ containing exactly two a’s.

2. Construct the minimal deterministic automaton of Σ∗abaab for Σ = \{a, b\}. Describe how you could use this automaton to search in a text for the occurrences of abaab. Let w ∈ Σ∗ be a string over the finite alphabet Σ. Show that the minimal automaton accepting Σ∗w has exactly |w| + 1 states. What is its number of transitions?

3. Give a deterministic automaton accepting the numbers x written in base 2 such that x ≡ 3 mod 4. How can you generalize the result (different base, different residue than 3, or different divisor than 4)? (Hint: assign one state to each possible rest of the division of x by 4).

B. Finite-State transducers, weighted automata, rational power series

1. Give a weighted regular expression (rational power series) representing the weighted automaton of Figure 1.

2. Give a sequential transducer giving the result of the division of x by y (y is given, x varies). What is the number of states and transitions of that transducer. Could they be reduced? (Hint: see exercise A.3.).

C. Algorithms, software library

The questions in this section should be answered by using the command-line utilities of the FSM or OpenFst library, except from the use of some simple scripts. The answers should be justified.

1. Download the CMU pronunciation dictionary from
http://www.cs.nyu.edu/~mohri/asr08/cmu_shuffled_dict.txt

This contains all the lines of the original CMU dictionary but the entries have been sorted in random order.

(a) Create a pronunciation lexicon $L_1$ for the first half of the dictionary. Some words admit multiple pronunciations, include all pronunciations. Use the closure of the pronunciation-to-word transducer to define $L_1$. What is the number of states and transitions of $L_1$?

(b) $L_1$ can be used to produce pronunciations for the second half of the dictionary, possibly multiple ones. What percentage of the pronunciations of the second half of the dictionary are correctly predicted using $L_1$? To answer this question, use as much as possible automata and transducer operations and library utilities.

(c) Answer the same questions as before by using a second half of the dictionary and a transducer $L_2$.

(d) Based on the results you obtained, which transducer, $L_1$ or $L_2$, is more accurate?

(e) Use the most accurate transducer to find all possible word parsings of ‘T UW M EH N IY P ER S AH N Z’. Give the result as an automaton and as a list of strings in order of fewest to greatest number of words per string.