Mehryar Mohri
Speech Recognition
Courant Institute of Mathematical Sciences
Homework assignment 2
October 09, 2012
Due: October 22, 2012

A. Pronunciation dictionaries

The following is a sample of a pronunciation dictionary. The phonemic transcription as well as the probability of the transcription is given for each word.

1. *cray* $\rightarrow$ *K r ey* ($P = .9$)
2. *dance* $\rightarrow$ *D ae n s* ($P = .85$)
3. *data* $\rightarrow$ *D ae T ax* ($P = .5$)
4. *data* $\rightarrow$ *D ey T ax* ($P = .5$)
5. *date* $\rightarrow$ *D ey T* ($P = .85$)
6. *day* $\rightarrow$ *D ey* ($P = .9$)

Construct a weighted transducer that represents the corresponding weighted transduction (use negative $\log$ of probabilities to assign weights to the paths, words such as 'cray' for the input alphabet, and phones such as 'K' or 'ae' for the output alphabet). Take the inverse of that transducer and make it as compact as you can.

B. Weighted Grammars

Here is a set of sentences with their corresponding probability of occurrence:

He comes (very) late, in the afternoon* (P = .2)
He comes (very) late, in the evening* (P= .3)
He will come (very) late, in the afternoon* (P = .1)
He will come (very) late, in the evening* (P = .4)

Construct a compact weighted automaton that accepts exactly these sentences (use negative $\log$ of probabilities to assign weights to the paths, use words such as 'he' to label transitions).

C. Text Operations

1. Elementary automata. Create an automaton for each of the following questions, given the alphabet $\Sigma = \{a, b, \ldots, z, A, B, \ldots, Z, \langle space \rangle\}$:

   (a) accepts a letter in Σ (excluding space),

   (b) accepts a single space,

   (c) accepts a capitalized word (where a word is a string of letters in Σ excluding space, and a capitalized word has its initial letter uppercase and remaining letters lowercase),

   (d) accepts a word containing the letter a.

2. Complex automata. Using the elementary automata of the previous exercise as the building blocks, use appropriate library operations on them to create an automaton that:

   (a) accepts zero or more capitalized words each followed by a space,

   (b) accepts a word beginning or ending in a capitalized letter,

   (c) accepts a word that is capitalized and contains the letter a,

   (d) accepts a word that is capitalized or does not contain an a,

   (e) accepts a word that is capitalized or does not contains an a (this should be done without using the union operation of the library).

3. Optimizations. Epsilon-remove, determinize, and minimize each of the automata constructed in the previous question. Give the number of states and arcs before and after these operations.

D. Trim automata

Consider the automaton:

```
0 1 1
0 2 2
1 1 1
2
3 4 4
4 3 3
4
```

1. How many states can be reached from the initial state?

2. How many states can reach a final state?

3. Compile this automaton and then remove all useless states.

E. Codes

Given the alphabet $\Sigma = \{a, b, \ldots, z, \langle space \rangle\}$,

1. create a transducer that implements the *rot13* cipher – $a \rightarrow n, b \rightarrow o, \ldots, m \rightarrow z, n \rightarrow a, o \rightarrow b, \ldots, z \rightarrow m$,

2. encode the message `"my secret message"` (assume $\langle space \rangle \rightarrow \langle space \rangle$),

3. decode the encoded message from above.

F. Numbers

Given the alphabet $\Sigma = \{0, 1, \ldots, 9\}$,

1. create an automaton that accepts numbers in the range $0 - 999999$

2. create a transducer that maps numbers (in the range $0 - 999999$) represented as strings of digits to their English read form, e.g.,

   $1 \rightarrow$ one
   $11 \rightarrow$ eleven
   $111 \rightarrow$ one hundred eleven
   $1111 \rightarrow$ one thousand one hundred eleven
   $11111 \rightarrow$ eleven thousand one hundred eleven

3. Randomly generate several numbers both as strings of digits and in their read form.

G. Spelling

Given the alphabet $\{a, b, \ldots, z, \langle space \rangle\}$, create a spelling corrector transducer that implements the (imperfect) traditional rule – 'i before e except after c'. Use it to correct the inputs 'yeild' and 'reciept'.

H. Roman numerals

Given the alphabet $\{I, V, X, L, C, D, M\}$,

1. create a weighted automaton that assigns to Roman numerals their numeric value (hint: use `fsmbestpath`).
2. ϵ-remove, determinize, and minimize this automaton. Draw the automaton before and after these operations.

I. Genome

Given the alphabet $L = \{A, G, T, C\}$,

1. create a transducer T that implements the following edit distance:
   $d(x, x) = 0, x \in L$
   $d(x, \epsilon) = d(\epsilon, y) = 1, x \in L$.
   $d(x, y) = 1.5, x \neq y \in L$
2. using T, find the best alignment between the strings 'AGTCC' and 'GGTACC'
3. find the second best alignment