Mehryar Mohri
Speech Recognition
Courant Institute of Mathematical Sciences
Homework assignment 1
Due: October 15, 2007

A. Genome

Let Σ denote the alphabet $\Sigma = \{A, G, T, C\}$.

1. Create a transducer T that implements the edit distance d based on the following insertions, deletions, and substitution costs. For all $a, b \in \Sigma, a \neq b$,
$$\begin{aligned} d(a, a) &= 0, \\ d(a, \epsilon) &= d(\epsilon, b) = \tfrac{1}{2}, \\ d(a, b) &= \tfrac{1}{3}. \end{aligned} \tag{1}$$

2. Using T, find the best alignment between the strings 'AGTCC' and 'GGTACC'. What is the cost of that alignment? What is the complexity of the algorithm you used? Find the second best alignment.

3. Let A be an automaton accepting the set $X = \{AGTCC, GTACGC\}$ and B an automaton accepting $Y = \{GGTACC, CAGTAC\}$. Using T, A, and B, find the first and second best alignment between the strings in sets X and Y. Give the complexity of your algorithm.

4. Modify the transducer T to take into account the following additional transposition cost: $d(ab, ba) = \frac{1}{4}$. Answer the same as in the previous question.

B. Counting

Let the alphabet be $\Sigma = \{a, b\}$.

1. Define a transducer T mapping each input string $x \in \Sigma^*$ to the set of its factors or substrings $\text{Fact}(x) = \{u : x \in \Sigma^* u \Sigma^*\}$.

2. Use T to define a transducer U counting substrings, that is such that $U(x, u)$ be exactly the set of occurrences of the substring u in x for any $x, u \in \Sigma^*$.

3. Give an algorithm for finding the number of occurrences of all substrings of length 3 in a document. What is the complexity of your algorithm?

C. Pronunciation dictionary

1. Consider the following words and their pronunciations (in ARPABET):

any	eh n iy
e.	iy
many	m eh n iy
men	m eh n
per	p er
persons	p er s uh n z
sons	s uh n z
suns	s uh n z
to	t uw
tomb	t uw m
too	t uw
two	t uw

   (a) Create a pronunciation lexicon L for these words – i.e., the closure of the pronunciation-to-word transducer.

   (b) using L, find all possible word parsings of 't uw m eh n iy p er s uh n z'. Give the result as a graph and as a list of strings in order of fewest to greatest number of words per string.

   (c) consider the (improbable) bigram language model that gives the cost of word β being followed by word α as:

$$Cost(\alpha|\beta) = |\,\|\alpha\| - \|\beta\|\,|\,,$$

   where $\|\gamma\|$ is the number of phonemes in the pronunciation of the word γ. Create a weighted acceptor that implements this language model. Find the best parsing of the string in (b) when constrained by this language model.