UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Department of Electrical and Computer Engineering
Instructor: Mark Hasegawa-Johnson

ECE 537 Speech Processing

Problem Set 9

Fall 2009

Issued: Wed Nov. 12, 2009 **Due:** Fri Nov. 21, 2009

Reading for problem set 9: Allen and Berkley, *Image Method...*, 1979. Chapter 6 of Flanagan, Allen, & Hasegawa-Johnson

Problem 9.1

A particular room is 5 × 4 × 3m, where 3m is the height. Alfred stands 2m tall, and 1m from each of the walls. Brigitta is also 2m tall, and stands 2m from each wall, but only 1.414m from Alfred, whom she adores, because 1100 years ago he was the king of England. The walls, floors and ceiling of the room are covered with hand-carved planking, which has a reflection coefficient of about $\gamma = 0.8$. Alfred sings in a resonant baritone. At a distance of 25cm, the sound pressure level of his voice is 94dB SPL.

(a) What is the delay between his lips and Brigitta's ear, and what is the SPL of the direct sound when it reaches her? Solution:

$$\tau = \frac{r}{c} = 1.414m340m/s = 4.16\text{ms}$$

$$\beta = 94 + 20\log_{10}\left(\frac{0.25m}{1.414m}\right) = 94 - 15 = 89\text{dB SPL}$$

(b) When does the first echo reach her, and what is its sound pressure level when it does? Solution: The first echo is the one off the ceiling, which arrives at

$$\tau = \frac{r}{c} = 2m340m/s = 5.9\text{ms}$$

$$\beta = 94 + 20\log_{10}\left(\frac{0.25m}{2m}\right) + 20\log_{10}\gamma = 74\text{dB SPL}$$

(c) One particular echo bounces twice from the ceiling, twice from the floor, and once from each wall (north, south, east, and west) before it reaches her. When does it reach her, and what is the echo SPL when it does?

Solution: If (k, l, m) indexes x-directed, y-directed, and z-directed bounces, we have $k = \pm 2$, $l = \pm 2$, $m = \pm 4$. We don't know if each of these is positive or negative without knowing, e.g., whether the sound bounces first off the floor or the ceiling. Coordinates of the virtual source are

$$\vec{r}_{\pm 2,\pm 2,\pm 4} = (x_{\pm 2}, y_{\pm 2}, z_{pm4}) = \left((-1)^{|k|}x_0 + kL_x, (-1)^{|l|}y_0 + lL_y, (-1)^{|m|}z_0 + mL_z\right)$$

$$= (1.5 \pm 10, 1 \pm 8, 0.5 \pm 12)$$

where $(x_0, y_0, z_0) = (1.5, 1, 0.5)$ is the position of Alfred's mouth. The position of Brigitta's ear is

$$\vec{r}_R = (x_R, y_R, z_R) = (0.5, 0, 0.5)$$

Assume, for example, that all signs are positive. Then the echo arrives at

$$\tau = \frac{|\vec{r}_{224} - \vec{r}_R|}{c} \approx 53\text{ms}$$

at a level of

$$\beta = 94 + 20\log_{10}\left(\frac{0.25m}{|\vec{r}_{224} - \vec{r}_R|}\right) + 20(|k| + |l| + |m|)\log_{10}\gamma = 41\text{dB SPL}$$

Problem 9.2

Suppose that sound is produced in a room whose dimensions are L_x, L_y, L_z, and whose walls and floor all have reflection coefficient γ. The speed of sound is c. Assume that the source intensity is 1 Watt/m^2 when measured at a distance of 1m.

(a) Define $R(T)$ to be the distance that sound can travel in T seconds. What is $R(T)$? Solution: $R(T) = cT$

(b) The number of echoes that arrive in T seconds, $N(T)$, is equal to the number of image sources that fit within a sphere whose radius is $R(T)$. What is $N(T)$? Solution:

$$N(T) = \frac{\frac{4}{3}\pi c^3}{V}T^3$$

where $V = L_xL_yL_z$ is the room volume.

(c) The echo density, the number of echoes per second, can be calculated as dN/dT. Find dN/dT. Solution:

$$\frac{dN}{dT} = \frac{4\pi c^3}{V}T^2$$

(d) What is the intensity, in Watts/m^2, of a typical echo arriving at time T? Note: you will need to know the number of walls that have been traversed on the path between the "typical" image source and the receiver. For this, you may write your solution in terms of L_{typ}, the "typical" path length between walls, which is the volume of the room divided by its surface area (this number is proportional to the harmonic mean of the dimensions of the room):

$$L_{typ} = \frac{V}{A} = \frac{L_x L_y L_z}{2L_x L_y + 2L_x L_z + 2L_y L_z}$$

Solution:

$$I_{echo} = \frac{I_{source}}{r^2}\gamma^{Nwallscrossed} = \frac{1}{c^2T^2}\gamma^{cT/L_{typ}}$$

(e) The reverberation intensity rate, in Watts/s/m^2, is equal to the intensity of an individual echo, times the number of echos that arrive per second. Find $R(T)$, the reverberation intensity rate, as a function of T. Solution:

$$R(T) = \frac{4\pi c^2}{V}\gamma^{cT/L_{typ}}$$

(f) The reverberation time, T_{60}, is defined by

$$10\log_{10}\left(\frac{R(T_{60})}{R(0)}\right) = -60\text{dB}$$

Find T_{60} in terms of other parameters. Solution:

$$T_{60} = \frac{6L_{typ}}{c\log_{10}(1/\gamma)}$$

(g) Now suppose that the reflection coefficient of the walls and floor is a nonzero constant (γ) for $0 \le |\Omega| \le B$, but is equal to zero (no reflections) at higher frequencies. With this modification, echoes are no longer delta functions: now they are sinc-like. Define the "diffusion" of an echo to be the width of its main lobe. What is the typical diffusion, $\Delta(T)$, of echoes arriving at time T?

Solution: Each reflection filters by $\frac{\gamma B}{\pi}\text{sinc}(Bt)$, which has a main-lobe width of $2\pi/B$. After cT/L_{typ} reflections, the main lobe width is

$$\Delta(T) = \frac{2\pi cT}{BL_{typ}}$$

(h) The room response has two parts. "Early echoes" are echoes that are heard distinctly, i.e., the average time between echoes is less than the diffusion of an echo. "Reverberation" is the set of echoes that overlap with one another, i.e., the average time between echoes is larger than the diffusion of an echo. What is the cutoff time, T, that separates early echoes from reverberation?

Solution: The echo density dN/dT is equal to $1/\Delta(T)$ when

$$T = \left(\frac{BVL_{typ}}{8\pi^2 c^4}\right)^{1/3}$$

Problem 9.3

Create a finite state transducer model of the Formulator.

(a) First, build a finite state acceptor that performs actions comparable to the lookup of syntactic templates. Let's work with a subset of the English language in which every verb is transitive, and can be expressed in either active or passive tense. The first FSA therefore generates a language with only two possible sentences:

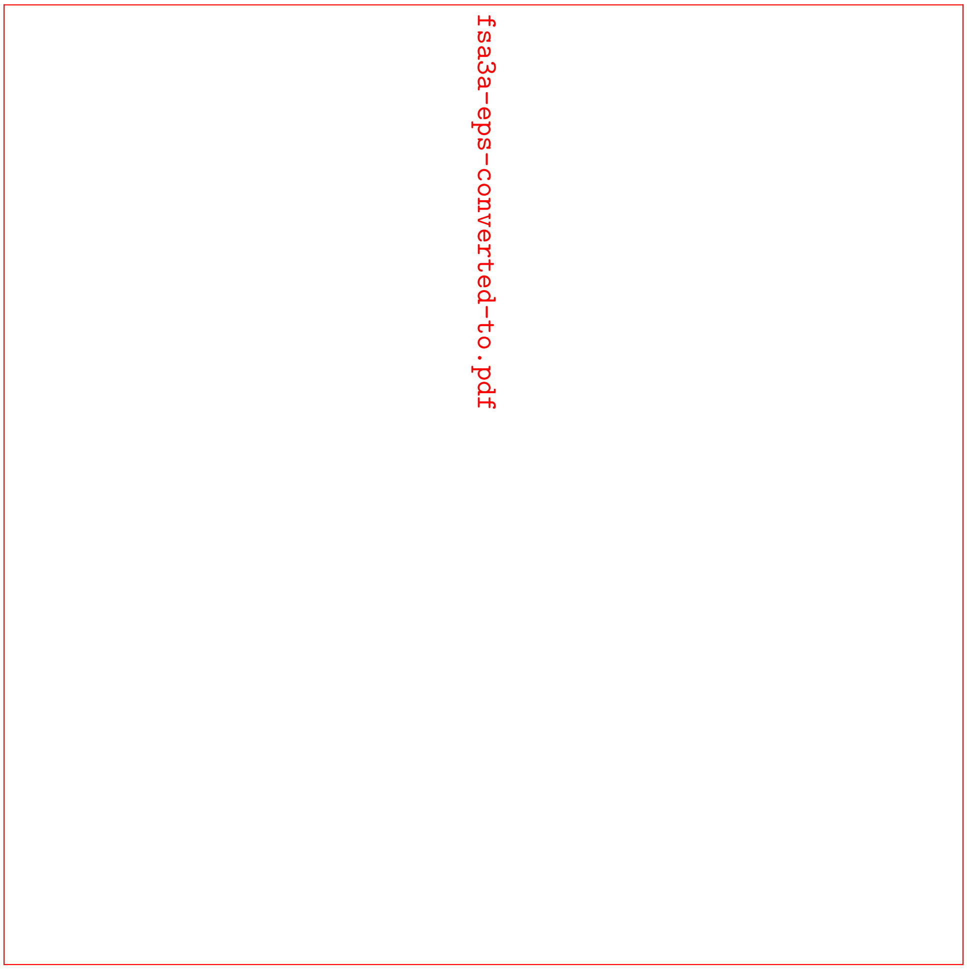
$$X, V, \mathrm{s}, Y \quad \text{(active tense, occurs with probability 0.6)} \tag{1}$$

or

$$Y, \mathrm{is}, V, \mathrm{ed}, \mathrm{by}, X \quad \text{(passive tense, occurs with probability 0.4)} \tag{2}$$

where V is the verb, X is the actor, and Y is the recipient of the action. Design an FST that generates the two sentences shown in Eqs. 1 and 2, with the probabilities shown.

Solution:



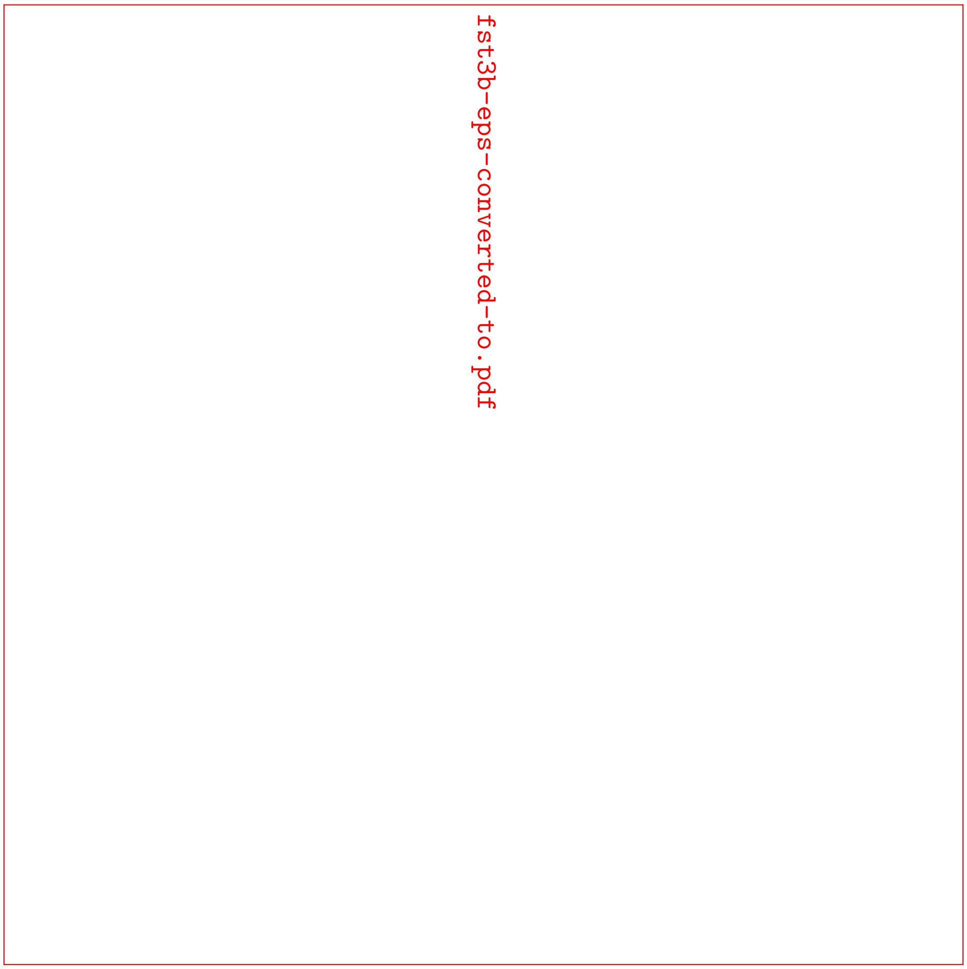
(b) Suppose that subjects in this experiment are shown one of two photographs. The first photograph shows a girl hugging a cat. The second photograph shows a dog chasing a mailman. Thus, there are two possible mappings:

(1) MAPPING 1 (probability=0.7): X →girl, Y →cat, V →hug.

(2) MAPPING 2 (probability=0.3): X →dog, Y →mailman, V →chase.

Sketch a finite state transducer that will generate either of the two translations shown above, but no others. In particular, make sure that your FST will not allow the dog to hug the cat, or the girl to chase the mailman.

Solution:



(c) Compose the two FSTs you generated in parts 1 and 2. How many different sentences are there in the resulting language?

Solution: There are four different sentences:

girl hug s cat	$P = 0.42$
cat is hug ed by girl	$P = 0.18$
dog chase s mailman	$P = 0.28$
mailman is chase ed by dog	$P = 0.12$

(d) In this language, what is the information content of the word "dog"? What is the information content of the word "chase"?

Solution:

$$I_{dog} = \log_2(1/P(\text{dog}|\text{context})) = \begin{cases} -1.8\text{bits} & \text{beginning of sentence} \\ 0\text{bits} & \text{after ``mailman is chase ed by''} \\ \infty & \text{elsewhere} \end{cases}$$

$$I_{mailman} = \begin{cases} -3.1\text{bits} & \text{beginning of sentence} \\ 0\text{bits} & \text{after ``dog chase s''} \\ \infty & \text{elsewhere} \end{cases}$$

(e) What is the entropy of this language, measured in average bits per sentence? What is the entropy in bits per morpheme? Count the elements "s" and "ed" as separate morphemes.

Solution:

$$H_{sent} = 0.42\log_2(1/0.42) + 0.18\log_2(1/0.18) + \ldots = 1.85\text{bits/sentence}$$

$$H_{morph} = \frac{0.42\log_2(1/0.42)}{4\text{morphemes}} + \frac{0.18\log_2(1/0.18)}{6\text{morphemes}} + \ldots = 0.4\text{bits/morpheme}$$

The last figure is an average, of course. It would be more useful to say that the first morphemes in the sentence has an entropy of 1.85 bits, and all succeeding morphemes have an entropy of zero bits.

(f) In the language you computed above, compute the unigram probabilities of the words "girl, cat, hug, dog, mailman, chase, s, is, ed, by." Find the cross-entropy of your language, as measured by an observer who knows nothing but the unigram probabilities.

Solution:

$$P_{uni}(\text{girl}) = P_{uni}(\text{cat}) = P_{uni}(\text{hug}) = \frac{0.42}{4} + \frac{0.18}{6} = 0.135$$

$$P_{uni}(\text{dog}) = P_{uni}(\text{chase}) = P_{uni}(\text{mailman}) = \frac{0.28}{4} + \frac{0.12}{6} = 0.09$$

$$P_{uni}(\text{s}) = \frac{0.42}{4} + \frac{0.28}{4} = 0.175$$

$$P_{uni}(\text{is}) = P_{uni}(\text{ed}) = P_{uni}(\text{by}) = \frac{0.18}{6} + \frac{0.12}{6} = 0.05$$

The cross-entropy of the language $\mathcal{L}$ computed based on informations provided by P_{uni} is:

$$H_{uni}(\mathcal{L}) = \sum_{\text{sentences}} P(\text{sentence}|\mathcal{L}) \sum_{morphs} P(\text{morph}|\text{sent}, \mathcal{L}) I(\text{morph}|\text{uni})$$

$$H_{uni}(\mathcal{L}) = \frac{0.42}{4}\left(3\log_2(1/0.135) + \log_2(1/0.175)\right) + \frac{0.28}{6}\left(3\log_2(1/0.135) + 3\log_2(1/0.05)\right) + \ldots$$

which works out to $H_{uni}(\mathcal{L}) = 3.196$bits/morpheme, quite a bit more than the 0.4 bits/morpheme of true entropy. The problem, of course, is that the unigram probabilities don't realize that the first morpheme in every sentence is sufficient to completely determine the rest of the sentence.

Problem 9.4

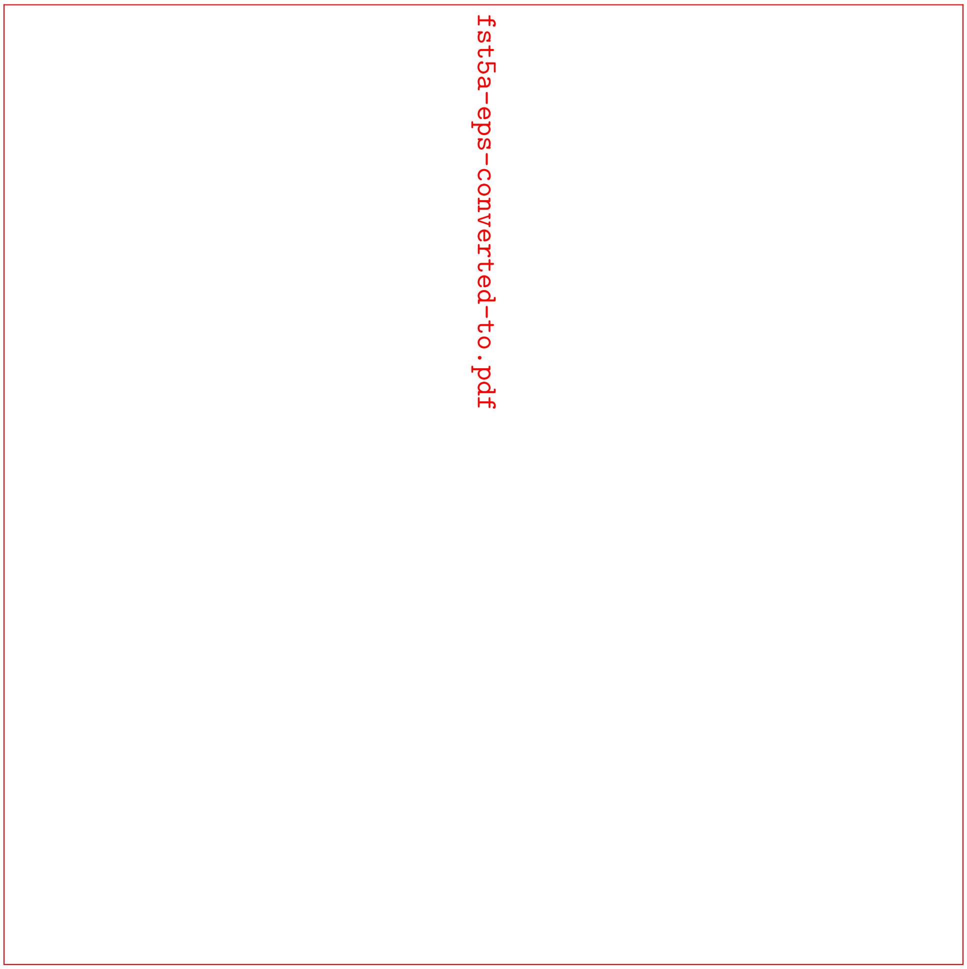
In this problem, you will build several different versions of a dictionary for a three-word language. Your dictionary will take the form of a finite state transducer that implements a mapping between the language $\mathcal{L}_I = \mathcal{A}_I$, $\mathcal{A}_I = \{$add,addle,amble$\}$, and the language $\mathcal{L}_O = \mathcal{A}_O^*$, $\mathcal{A}_O = \{$æ,b,d,l̩,m$\}$, where the symbol "l̩" means "syllabic l̩." Specifically, you want an FST that implements the following mapping:

add → [æd]
addle → [ædl̩]
amble → [æmbl̩]

There are many different ways to build an FST implementing the mapping shown above. This problem explores three of them.

(a) Create an FST that accepts any one word from this dictionary as input, and generates its pronunciation as output. Each transition in the FST should accept zero or one words as input, and generate zero or one phonemes as output. Use epsilons (null inputs and null outputs), as necessary, in order to guarantee that the word "add" shares all of its phoneme edges with the word "addle."

Solution:

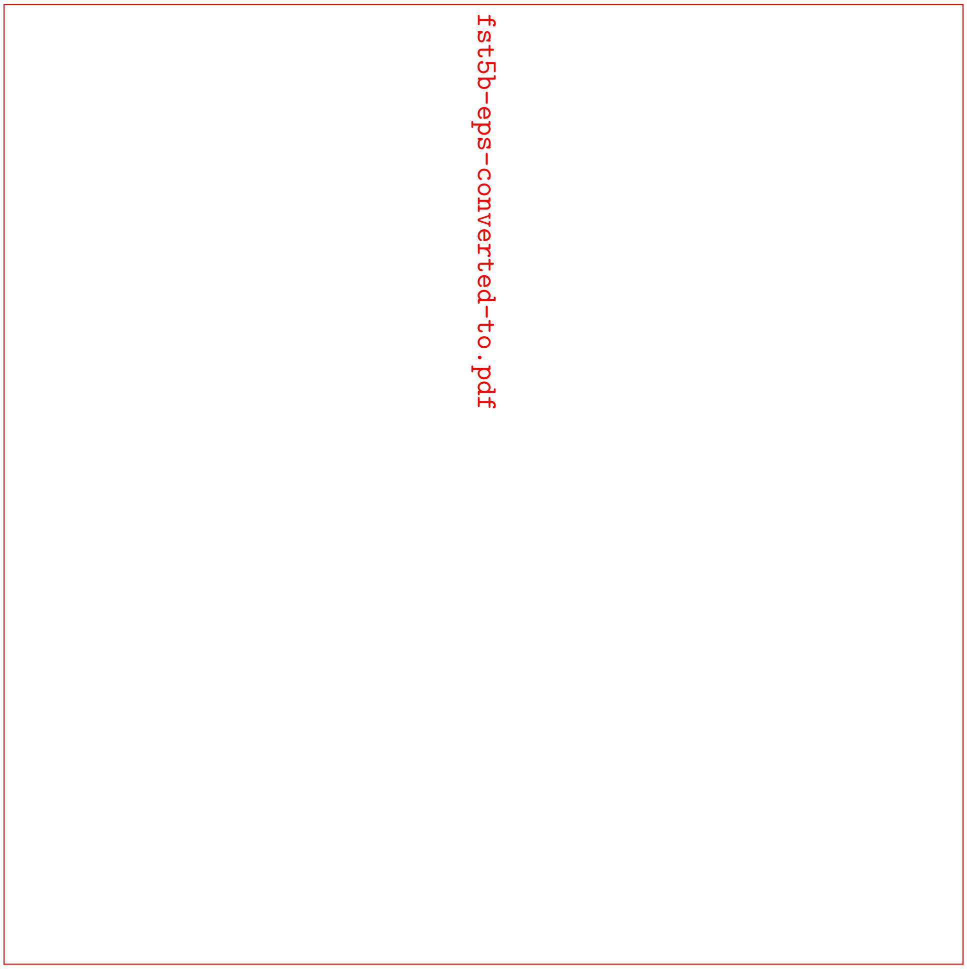


(b) Modify your tree from part (a): add edges as necessary in order to implement Kleene closure. The resulting FST should accept inputs from the language $\mathcal{L}_B = \mathcal{A}_I^*$, rather than just $\mathcal{L}_I = \mathcal{A}_I$. Usually, the cleanest way to Kleene is as follows:

(1) Add $\epsilon : \epsilon$ edges connecting all final nodes to a common "END" node, then

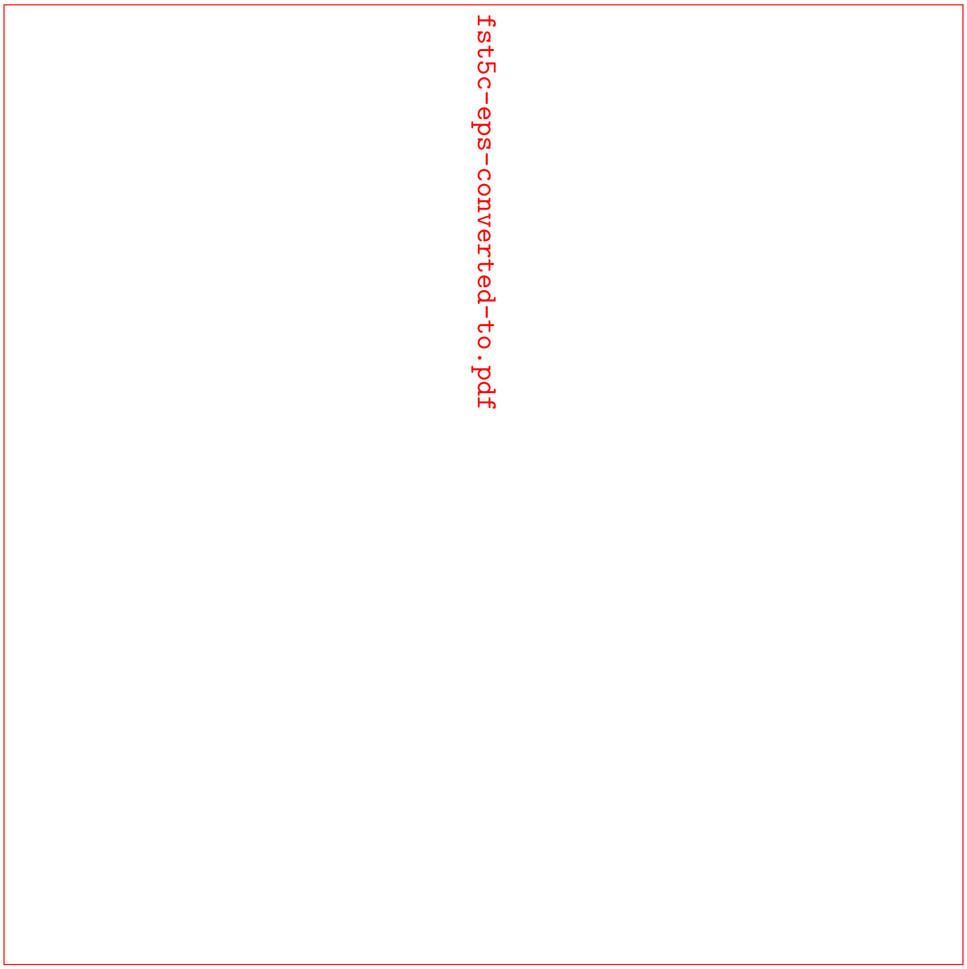
(2) Add one more $\epsilon : \epsilon$ edge, connecting the “END” node back to the “START” node.

Solution:



(c) Repeat part (a), but this time, create a left-branching tree. Note that you are repeating part (a), not part (b), therefore the input language should be $\mathcal{L}_I$, not $\mathcal{L}_B$. Which tree has fewer total edges in this case, the left-branching or the right-branching?

Solution: The right-branching tree is worse (more edges):



(d) Can you construct an FST that accepts one word as input, generates one pronunciation as output, but has fewer total edges than either the left-branching or right-branching tree? Each edge should accept no more than one input word, and generate no more than one output phoneme.

Solution: Actually, I couldn't find a dictionary with fewer total edges than the left-branching tree, but I was able to reduce the number of end states by one. When you perform Kleene closure (as in part (b)), every end state in the tree adds at least one edge, so reducing the end states is a good thing.

fst5d-eps-converted-to.pdf