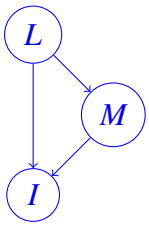
Ling 5801: Problem Set 5

Due via Carmen dropbox at 11:59 PM 11/22.

1. [10 pts.] Write an equation for a full joint distribution in terms of the following models:

   - θ_L, for using a word lemma (e.g. *find*) to communicate an idea,
   - θ_M, for using a morphological rule (e.g. */%ind/%ound/*) to inflect that lemma,
   - θ_I, for using an inflected form (e.g. *found*),

   with conditional dependencies as shown in the following network:



2. [10 pts.] Draw or describe a graphical representation of an extension of the above probability model, using a random variable for:

   - a category context for a word to be used (e.g. *participial*, such as after *pets are . . .*)

   Justify each additional conditional dependency in a sentence (e.g. for variable I: 'An inflected form like *found* would be different if the lemma is not *find* or the inflection is not */%ind/%ound/*.').

3. [10 pts.] PROGRAMMING: Using the syntax described in the lecture notes, write a program to read in models of language change over generations of speakers from standard input. Use the following format for component models of a grandparent speaker G, a parent speaker P (given grandparent), and a child speaker C (given parent) making use of the word *who* as opposed to *whom* in the position of an accusative filler (e.g. *who/whom did you invite?*):

```
G : who  = .1
G : whom = .9
⋮

P who  : who  = 1
P who  : whom = 0
P whom : who  = .3
P whom : whom = .7
⋮

C who  : who  = 1
C who  : whom = 0
C whom : who  = .5
C whom : whom = .5
```

⋮

then use these models to calculate a conditional probability distribution table for $\mathsf{P}(G \mid C)$, and print it in the following format:

```
GgivC who : who = 0.146
GgivC who : whom = 0.854
```

⋮

4. [10 pts.] PROGRAMMING: Using the syntax described in the lecture notes, write a program to read in models for all variables R, W, and O in the 'repeated trials' model shown at the beginning of the lecture notes on sequence modeling, from standard input in the following format

```
R : ohio = .5
R : phil = .5
W : neck = .6
W : knack = .4
O ohio neck : [nek] = 1
O phil neck : [nek] = .667
O phil neck : [naek] = .333
O ohio knack : [naek] = 1
O phil knack : [naek] = 1
```

and an input sequence of any number of observations in the format:

```
I [naek] [nek] [naek] ...
```

then print out a probability distribution for R given all of these input observations, in the following format (note: probabilities given observations should not necessarily match initial R model):

```
RgivenIdata : ohio = .4
RgivenIdata : phil = .6
```

Your program should be as short as possible. Hand in all inputs and outputs.