Questions for Flipped Classroom Session of COMS 4705 Week 5, Fall 2014. (Michael Collins)

Question 1 In this question our goal is to design an algorithm that takes a sentence s and a context-free grammar in Chomsky normal form as input, and as its output returns *the number of parse trees for the sentence s* as its output.

For example, if s is the sentence `a a a`, and the context-free grammar is

X → X X
X → a

with start symbol `X`, the algorithm should return the value 2, because there are two parses for the sentence under this grammar:

Question: Complete the following algorithm so that it returns the number of possible parse trees for the input sentence s.

Input: a sentence $s = x_1 \ldots x_n$, a context-free grammar $G = (N, \Sigma, S, R)$.
Initialization:
For all $i \in \{1 \ldots n\}$, for all $X \in N$,

$$\pi(i, i, X) = \begin{cases} 1 & \text{if } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

- For $l = 1 \ldots (n-1)$
  - For $i = 1 \ldots (n-l)$
    - Set $j = i + l$
    - For all $X \in N$, calculate

$$\pi(i, j, X) = \sum_{\substack{X \rightarrow Y\,Z \in R, \\ s \in \{i \ldots (j-1)\}}} \underbrace{\hspace{6cm}}_{\text{COMPLETE THE DEFINITION HERE}}$$

Output: Return $\pi(1, n, S)$

Question 2 Consider the CKY algorithm for finding the maximum probability for any tree when given as input a sequence of words $x_1, x_2, \ldots, x_n$. As usual, we use N to denote the set of non-terminals in the grammar, and S to denote the start symbol.

The base case in the recursive definition is as follows: for all $i = 1 \ldots n$, for all $X \in N$,

$$\pi(i, i, X) \quad = \quad \begin{cases} q(X \rightarrow x_i) & \text{if } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

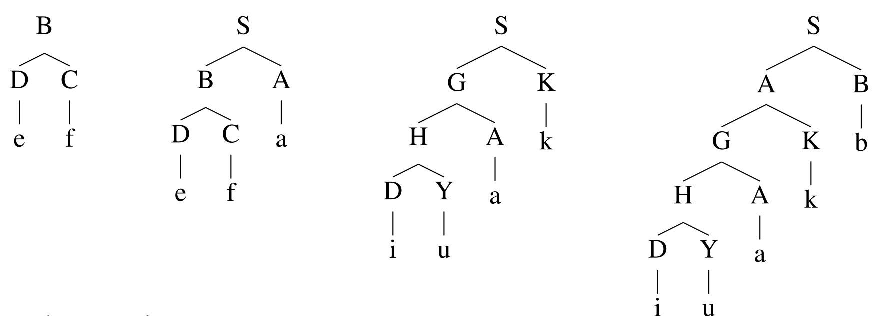
and the recursive definition is as follows: for all (i, j) such that $1 \leq i < j \leq n$, for all $X \in N$,

$$\pi(i, j, X) = \max_{\substack{X \rightarrow Y Z \in R, \\ s \in \{i \ldots (j-1)\}}} \left(q(X \rightarrow Y Z) \times \pi(i, s, Y) \times \pi(s+1, j, Z)\right)$$

Finally, we return

$$\pi(1, n, S) = \max_{t \in \mathcal{T}_G(s)} p(t)$$

Now assume that we want to find the maximum probability for any *left-branching* tree for a sentence. Here are some example left-branching trees:



(**Question continued on next page**)

It can be seen that in left-branching trees, whenever a rule of the form `X -> Y Z` is seen in the tree, then the non-terminal `Z` must directly dominate a terminal symbol.

Question: Complete the recursive definition below, so that the algorithm returns the maximum probability for any **left-branching** tree underlying a sentence $x_1, x_2, \ldots, x_n$.

Base case: for all $i = 1 \ldots n$, for all $X \in N$,

$$\pi(i, i, X) = \begin{cases} q(X \to x_i) & \text{if } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Recursive case: (Complete below)

Return:

$$\pi(1, n, S) = \max_{t \in \mathcal{T}_G(s)} p(t)$$

Question 3 Consider the following PCFG (probabilities for each rule are shown after the rule):

S $\rightarrow$ NP VP	1.0
NP $\rightarrow$ DT NBAR	1.0
NBAR $\rightarrow$ NN	0.7
NBAR $\rightarrow$ NBAR NBAR	0.3
VP $\rightarrow$ sleeps	1.0
DT $\rightarrow$ the	1.0
NN $\rightarrow$ mechanic	0.1
NN $\rightarrow$ car	0.2
NN $\rightarrow$ metal	0.7

Now consider a PCFG based on this context-free grammar. What parse tree will be returned as the highest probability tree for *the metal car mechanic sleeps*?

Question 4 Consider the following HMM:

- States in the HMM are $\{\texttt{A, B}\}$.
- q parameters of the HMM are $q(y|x)$ for $x \in \{\text{A, B, *}\}$ and $y \in \{\text{A, B, STOP}\}$.
- Vocabulary in the HMM is $\{\texttt{s}, \texttt{t}\}$.
- e parameters in the HMM of the form $e(y|x)$ for $x \in \{\text{A, B, *}\}$ and $y \in \{\texttt{s}, \texttt{t}\}$.

Our aim is this question is to write down a PCFG such that for any sentence $x_1 \ldots x_n$ and tag sequence $y_1 \ldots y_{n+1}$ with probability

$$p = p(x_1 \ldots x_n, y_1 \ldots y_{n+1})$$

under the HMM, there is a parse tree for the sentence $x_1 \ldots x_n$ with the same probability p under the PCFG.

Complete the probabilities for the following rules in the PCFG (hint: try writing down parse trees for simple sentence/tag sequences such as `s/A`, `s/A t/B` etc.):

S $\rightarrow$ A FA
S $\rightarrow$ B FB
S $\rightarrow$ A
S $\rightarrow$ B
FA $\rightarrow$ A FA
FA $\rightarrow$ A
FA $\rightarrow$ B FB
FA $\rightarrow$ B
FB $\rightarrow$ A FA
FB $\rightarrow$ A
FB $\rightarrow$ B FB
FB $\rightarrow$ B
A $\rightarrow$ s
A $\rightarrow$ t
B $\rightarrow$ s
B $\rightarrow$ t