Name: ________________ Reg #: ________________ Section: ____________

National University of Computer and Emerging Sciences, Lahore Campus

Course:	Natural Language Processing	Course Code:	CS 535
Program:	BS(Computer Science)	Semester:	Spring 2018
Duration:	60 Minutes	Total Marks:	24
Paper Date:	12-April-18	Weight	13%
Section:	ALL	Page(s):	5
Exam:	Midterm 2 Solution		

Q1) A sentence can easily have more than one parse tree that is consistent with a given CFG. How do PCFGs and non-probability-based CFGs differ in terms of handling parsing ambiguity? [2 Marks]

Answer:
PCFGs define probability of each rule which can be used to find probability of parse tree. Tree with maximum probability is selected.

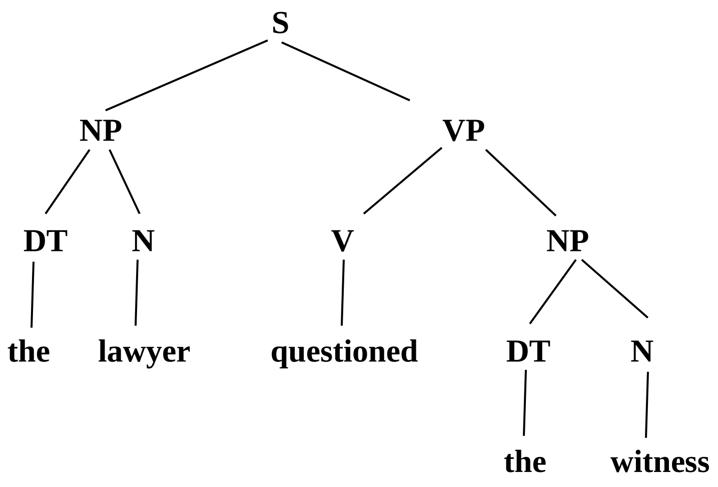
Q2) Which of the following is a false statement about PCFGs: [2 Marks]

a) The rules impose independence assumptions that effect poor modeling of structural dependency across the tree.
b) The rules do not model syntactic facts about particular words, which causes a variety of problems.
c) The joint probability of a sentence, S, and a parses of it, T, is the same as the probability of the parse, T.

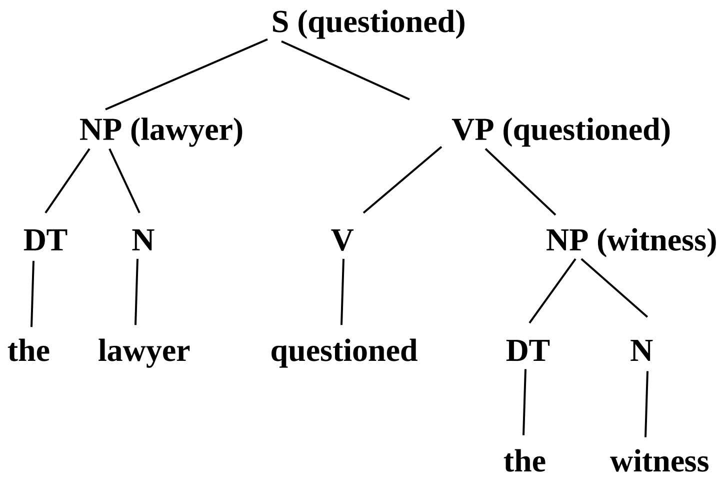
Answer: They are true

Q3) a) Which of the following trees is a lexicalized tree? [1 Mark]

Tree 1



Tree 2



Solution: Tree2

b) For the trees above, when you count and estimate the probability for rules, which tree is most likely to encounter sparseness problem? [2 Marks]

Tree2

c) How can you alleviate sparseness problem encountered in estimating probability for parse trees? [2 Marks]

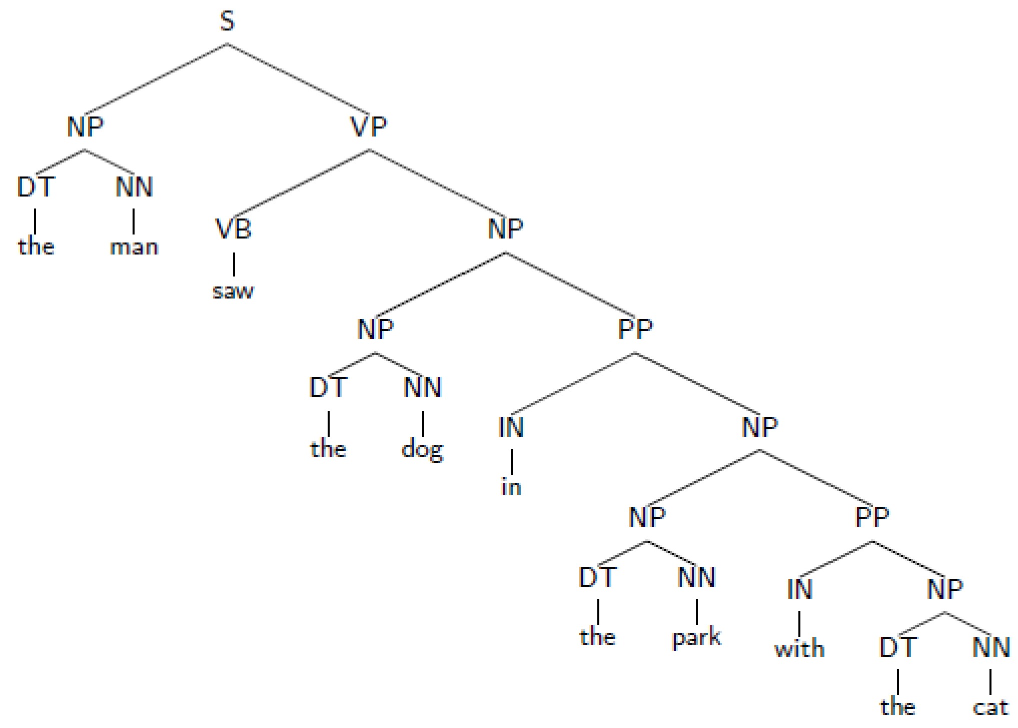
smoothing

Q4) Consider a context-free grammar with the following rules (assume that S is the start symbol):
S → NP VP
NP → DT NN
NP → NP PP
PP → IN NP
VP → VB NP
DT → the
NN → man
NN → dog
NN → cat
NN → park
VB → saw
IN → in
IN → with
IN → under

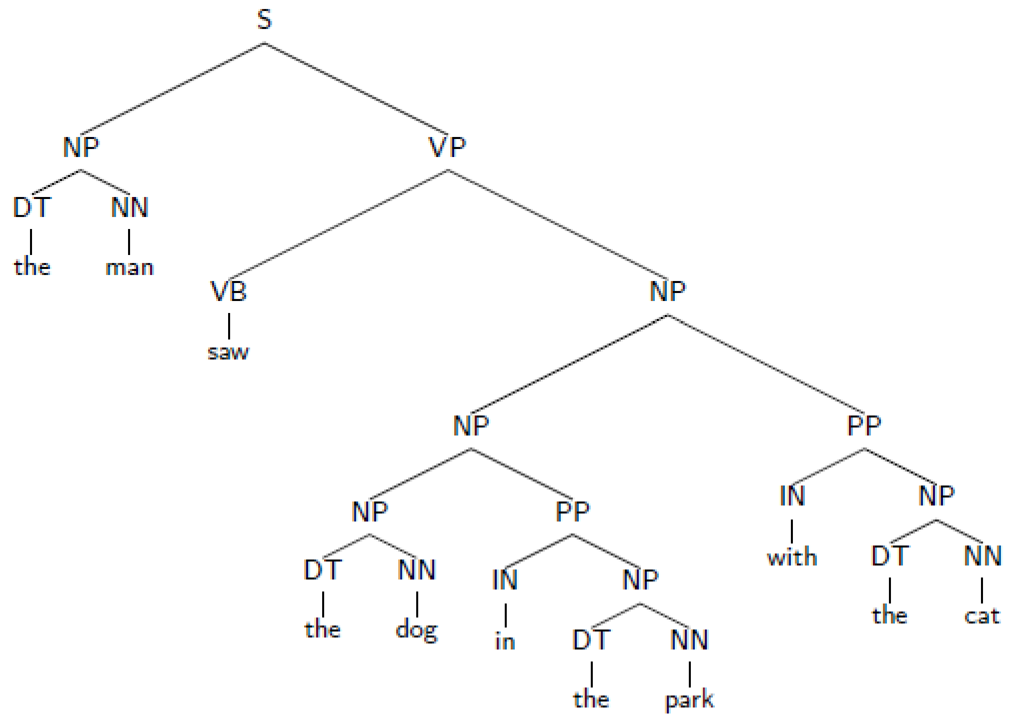
How many parse trees for “the man saw the dog in the park with the cat”? Draw all the parse trees for this sentence. [5 Marks]

Name: ________________ Reg #: ________________ Section: ____________

Two parse trees, parse tree 1:



Two parse trees, parse tree 2:



Name: ___________________ Reg #: ___________________ Section: _____________

Q3) Consider a trigram HMM, as introduced in class. We saw that the Viterbi algorithm could be used to find $\max_{y_1, \ldots, y_{n+1}} p(x_1, \ldots x_n; y_1, \ldots y_{n+1})$ where the max is taken over all sequences $y_1, \ldots y_{n+1}$ such that $y_i \in K$ for i = 1 …. n, and y_{n+1} = STOP. (Recall that K is the set of possible tags in the HMM.) In a trigram tagger we assume that p takes the form

$$p(x_1, \ldots x_n; y_1, \ldots y_{n+1}) = \prod_{i=1}^{n+1} q ¿¿$$

Recall that we have assumed in this definition that $y_0 = y_{-1} = y_{-2} = *$, and y_{n+1} = STOP. The Viterbi algorithm is shown in figure below.

```
Input: A sentence $x_1, \ldots x_n$, paramters $q(s|u, v)$ and $e(x | s)$
Dedinitions: Define $K$ to be the set of possible tags. Define $K_{-1} = K_0 = \{*\}$, and $K_k = K$ for $k = 1 \ldots n$.
Initialization: Set $\pi(0,*,*) = 1$
Algorithm: For $k = 1 \ldots n$,
        For $u \in K_{k-1}, v \in K_k$,
            $\pi(k,u,v) = \max_{w \in K_{k-2}} (\pi(k-1,w,u) \times q(v | w,u) \times e(x_k | v))$

Return $\max_{u \in K_{n-1}, v \in K_n} (\pi(n,u,v) \times q(STOP | u,v))$
```

Now consider a four-gram tagger, where p takes the form

$$p(x_1, \ldots x_n; y_1, \ldots y_{n+1}) = \prod_{i=1}^{n+1} q ¿¿$$

We have assumed in this definition that $y_0 = y_{-1} = y_{-2} = *$, and y_{n+1} = STOP. In the box on next page, give a version of the Viterbi algorithm that takes as input a sentence $x_1, \ldots x_n$, and finds $\max_{y_1, \ldots, y_{n+1}} p(x_1, \ldots x_n; y_1, \ldots y_{n+1})$ for a four-gram tagger, as defined above equation.
[10 Marks]

Input: A sentence $x_1, \ldots, x_n$, paramters $q(w|t, u, v)$ and $e(x \mid s)$

Dedinitions: Define K to be the set of possible tags. Define $K_{-2} = K_{-1} = K_0 = \{*\}$, and $K_k = K$ for $k = 1 \ldots n$.

Initialization: *Set* $\pi(0,*,*) = 1$

Algorithm: For $k = 1 \ldots n$,

```
        For $t \in K_{k-2}, u \in K_{k-1}, v \in K_k$,
            $\pi(k,t,u,v) = \max_{w \in K_{k-3}} (\pi(k-2,w,t,u) \times q(v \mid w,t,u) \times e(x_k \mid v))$
```

Return $\max_{t \in K_{n-2}, u \in K_{n-1}, v \in K_n} (\pi(n,t,u,v) \times q(STOP \mid t,u,v))$