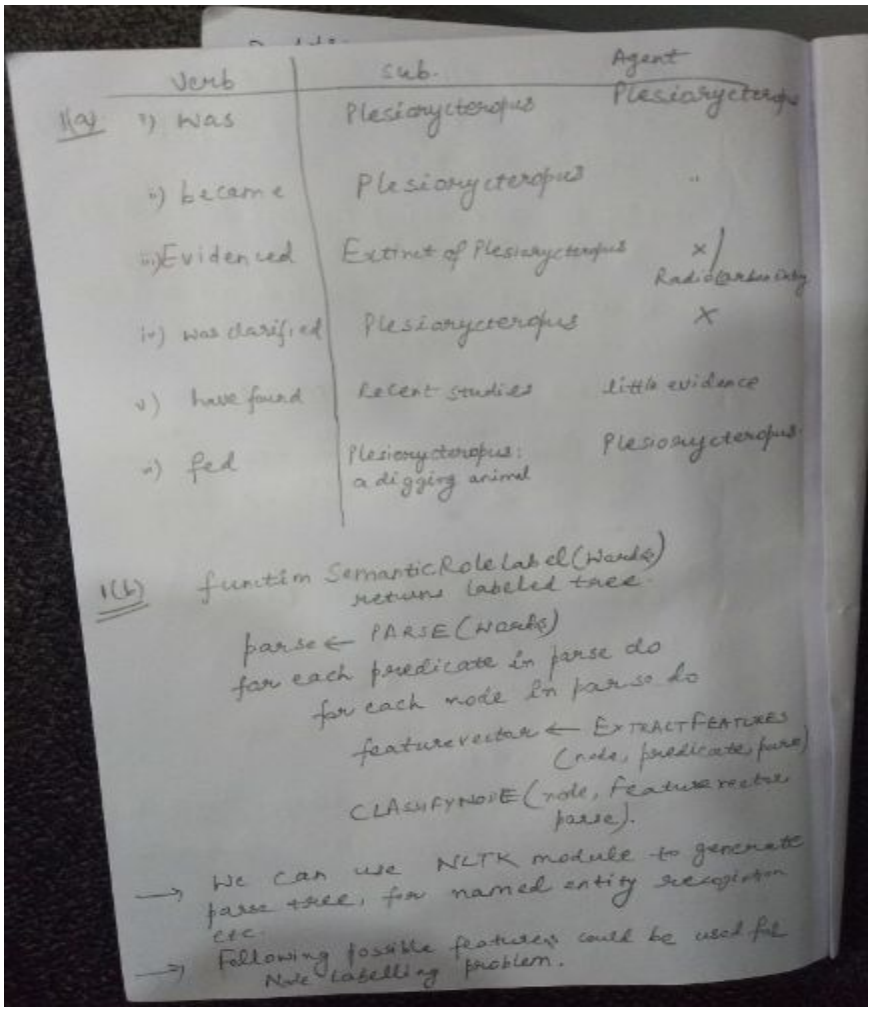
Assignment 4

By: Chanda Grover Date: March 4, 2019

Ans 1 a)



	Verb	sub.	Agent
1(a)	i) was	Plesiorycteropus	Plesiorycteropus
	ii) became	Plesiorycteropus	"
	iii) Evidenced	Extinct of Plesiorycteropus	x / Radiocarbon dating
	iv) was classified	Plesiorycteropus	x
	v) have found	Recent studies	little evidence
	vi) fed	Plesiorycteropus: a digging animal	Plesiorycteropus

1(b)

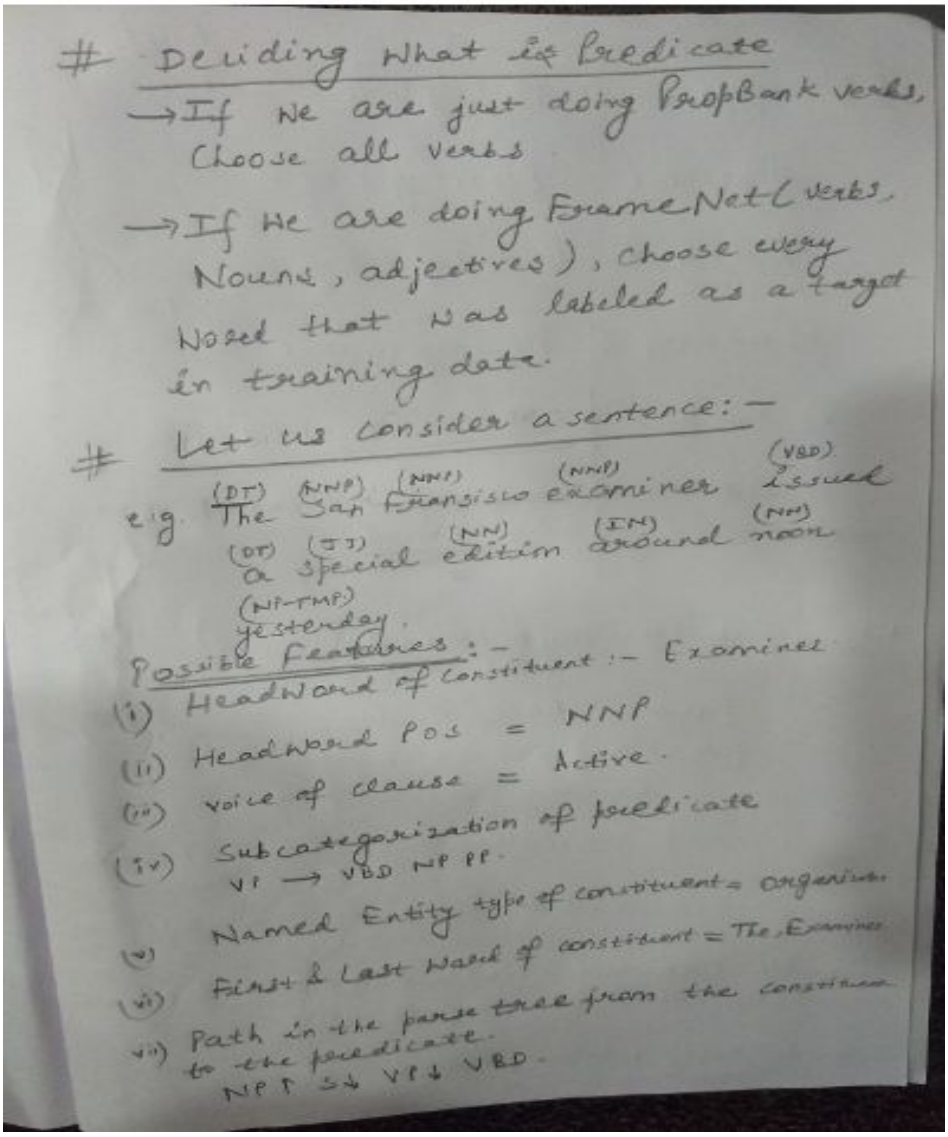
function SemanticRoleLabel(Words)
returns labeled tree

parse ← PARSE(Words)
for each predicate in parse do
  for each node in parse do
    featurevector ← EXTRACTFEATURES(node, predicate, parse)
    CLASSIFYNODE(node, featurevector, parse).

→ We can use NLTK module to generate parse tree, for named entity recognition etc.

→ Following possible features could be used for Role labelling problem.

Ans 1b)



Deciding What is Predicate

→ If We are just doing PropBank verbs, Choose all verbs

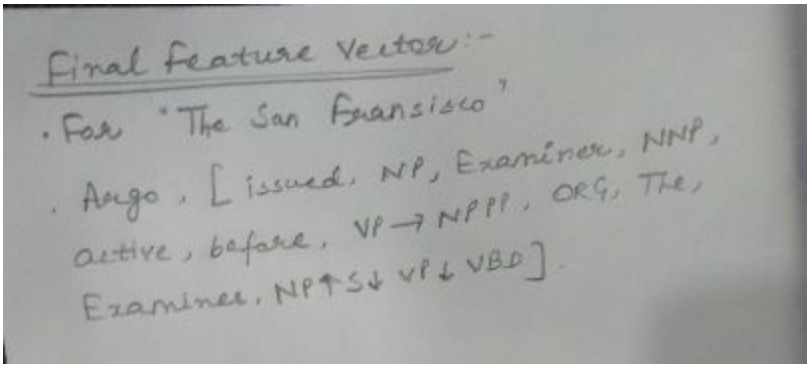
→ If We are doing FrameNet (verbs, Nouns, adjectives), choose every Word that was labeled as a target in training data.

Let us consider a sentence:—

e.g. (DT) The (NNP) San (NNP) Francisco (NNP) examiner (VBD) issued (DT) a (JJ) special (NN) editin (IN) around (NN) noon (NP-TMP) yesterday.

Possible Features :-

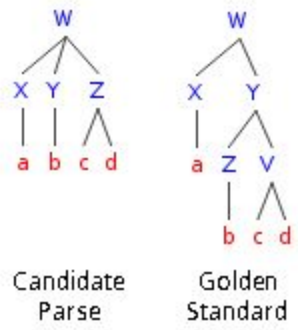
(i) HeadWord of constituent :- Examinel
(ii) HeadWord Pos = NNP
(iii) voice of clause = Active.
(iv) Subcategorization of predicate
VP → VBD NP PP.
(v) Named Entity type of constituent = organiza.
(vi) First & Last Word of constituent = The, Examine
(vii) Path in the parse tree from the constituent to the predicate.
NP ↑ S ↓ VP ↓ VBD.



Ans 2 a) The most common evaluation metrics examine the conceptual "distance" between the candidate parse generated by the parser, and the correctly annotated solution (the "gold standard"). Gold standards are sometimes annotated by hand, but more often they come from existing corpora. Generally speaking, the current standard comparison is the **PARSEVAL** metric. It defines a set of values that focus primarily on the constituency differences between the two trees. The main values are:

- **Precision**: The number of correct constituents out of the number of constituents in the candidate parse.
- **Recall**: The number of correct constituents out of the number of constituents in the gold standard.
- **F-Score**: The harmonic mean of precision and recall (see below).
- $$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

To demonstrate the concept, consider the pair of trees given below.



The constituents for each are given below in form label:yield (yield is the span of the node/constituent).

Candidate gold

X:a	X:a
Y:b	Z:b
Z:cd	V:cd
--	Y:bcd
W:abcd	W:abcd

As we can see, all of the constituents in the candidate have correct yield, so we have 4/4 = **100%**precision, but there is an extra node in the hierarchy of the gold, so we only have 4/5 = **80%** recall. These values then give us an F-score of 2*1*0.8/(1+0.8) = **88.9%**.

If we want to consider the accuracy of the labels, we can use the labelled precision and labelled recall. These metrics only count a node as correct if its yield AND label match the gold standard. Above, only 2 out of our 4 nodes with correct yield have the correct label (W & X), so we get:

P = 50% R = 40% F = 44.4%

PARSEVAL also includes one last value—the number of **crossing brackets**. This is simply a count of how many constituent boundaries in the candidate crossover constituent boundaries in the gold. These errors might be considered more serious, because they often indicate an element is in the completely wrong grouping. In the above example there are no crossing brackets.

Ans 2 b) Labelled Recall Parsing Algorithm:

Algorithm finds that tree T_G which has the highest expected value for the Labelled Recall Rate, L/Nc (where L is the number of correct labelled constituents, and Nc is the number of nodes in the correct parse). This can be written as follows:

$$T_G = \arg\max_T E(L/N_C)$$

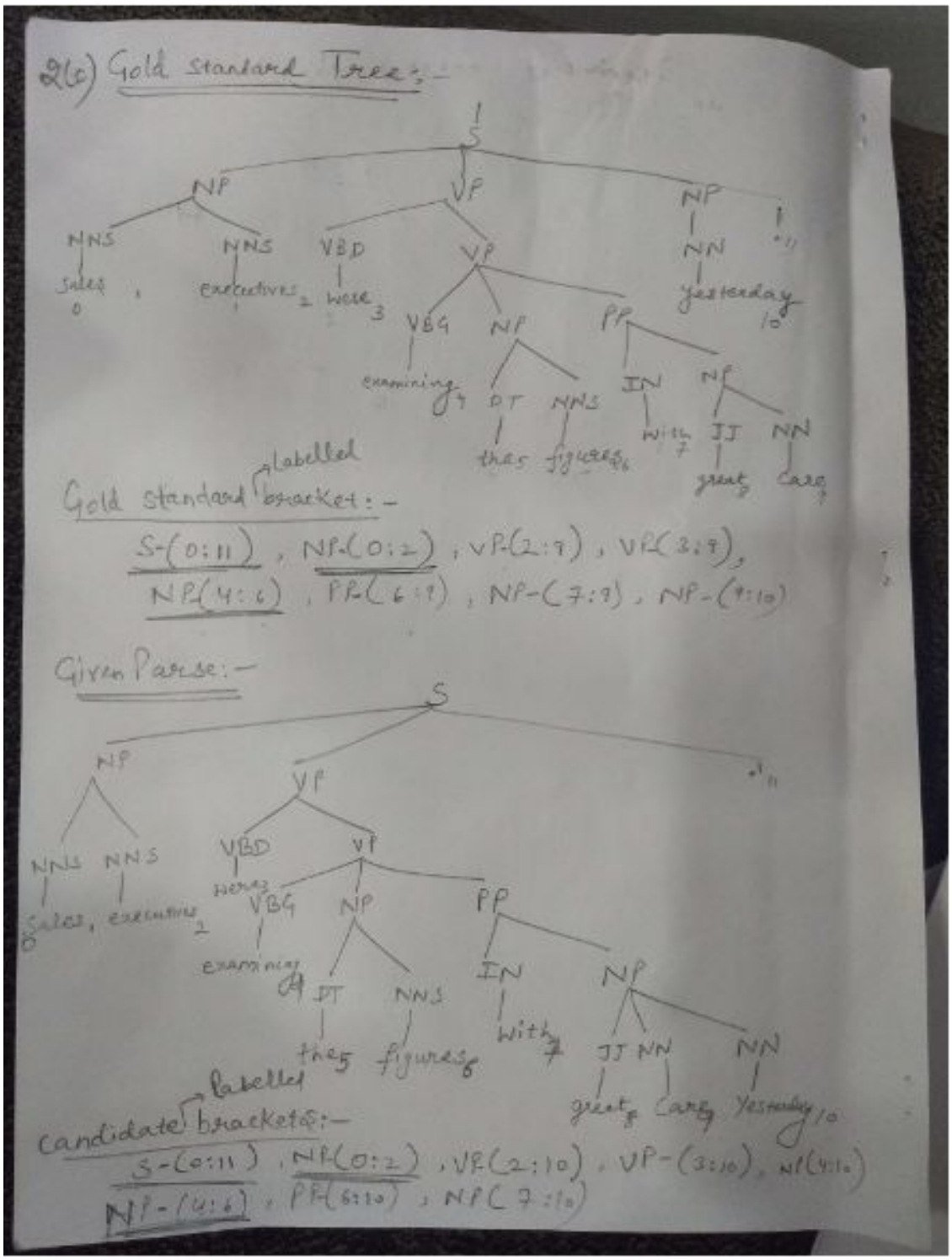
```
for length := 2 to n
        for s := 1 to n-length+l
                t := s + length - l;
                loop over nonterminals X
                        let max_g:=maximum of g(s,t,X)
                loop over r such that s <= r < t
                        let best_split:= max of maxc[s,r] + maxc[r+l,t]
                maxc[s, t] := max_g + best split;
```

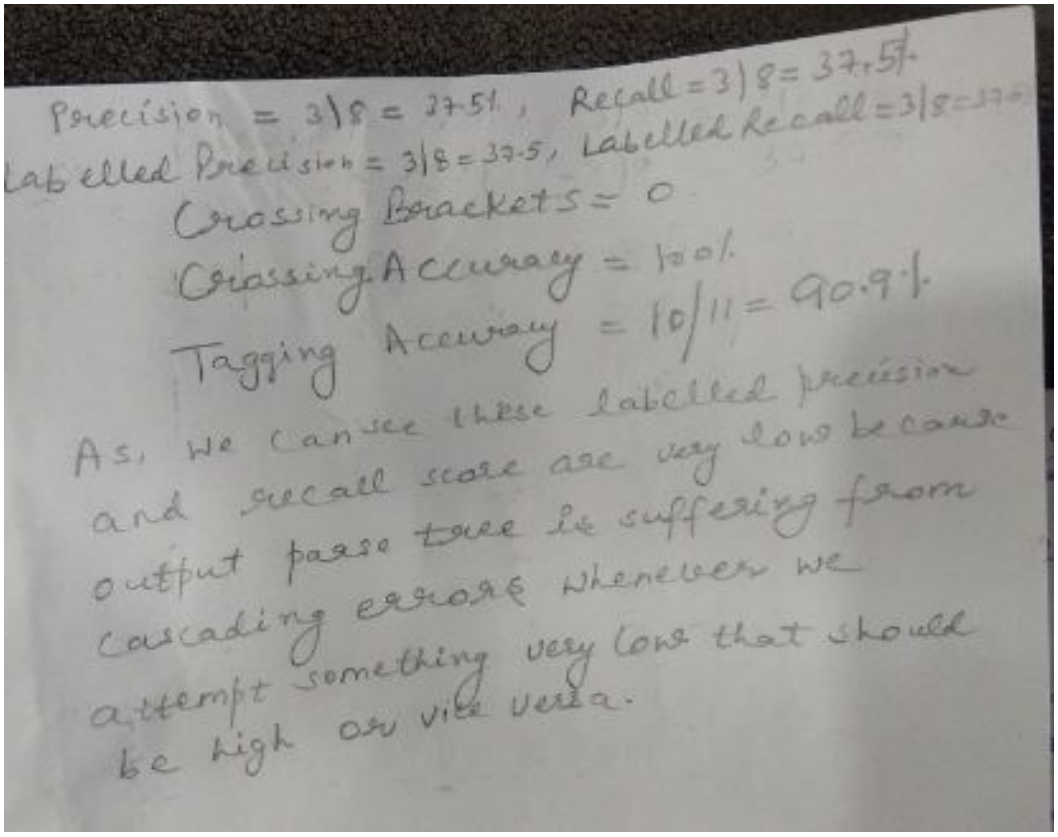
MAXC(1, n) contains the score of the best parse according to the Labelled Recall criterion.

For a grammar with r rules and k nonterminals, the run time of this algorithm is $O(n^3 + kn^2)$ since there are two layers of outer loops, each with run time at most n, and an inner loop, over

nonterminals and n. However, this is dominated by the computation of the Inside and Outside probabilities, which takes time $O(rn^3)$.

Ans 2 c)





Ans 2 d) We compare two dependency tree by finding how many of the dependencies are right and how many are wrong in a sentence. But in Penn Tree bank, dependency information is not shown. For this, we can induce dependency relationships from the phrase structured tree given.

The standard methodology for evaluating dependency parsers, as well as other kinds of parsers, is to apply them to a test set taken from a treebank and compare the output of the parser to the gold standard annotation found in the treebank. Dependency parsing has been evaluated with many different evaluation metrics. The most widely used metrics are listed here

- **Exact match**: This is the percentage of completely correctly parsed sentences. The same measure is also used for the evaluation of constituent parsers.
- **Attachment score**: This is the percentage of words that have the correct head. The use of a single accuracy metric is possible in dependency parsing thanks to the single-head property of dependency trees, which makes parsing resemble a tagging task, where every word is to be tagged with its correct head and dependency type. This is unlike the standard metrics in constituency-based parsing, which are based on precision and recall, since it is not possible to assume a one-to-one correspondence between constituents in the parser output and constituents in the treebank annotation.

- **Precision/Recall**: If we relax the single-head property or if we want to evaluate single dependency types, the following metrics can be used. They correspond more directly to the metrics used for constituent parsing.
    - **Precision**: This is the percentage of dependencies with a specific type in the parser output that were correct.
    - **Recall**: This is the percentage of dependencies with a specific type in the test set that were correctly parsed.
    - **F-measure** (β=1): This is the harmonic mean of precision and recall.

All of these metrics can be unlabeled (only looking at heads) or labeled (looking at heads and labels). The most commonly used metrics are the labeled attachment score (LAS) and the unlabeled attachment score (UAS).

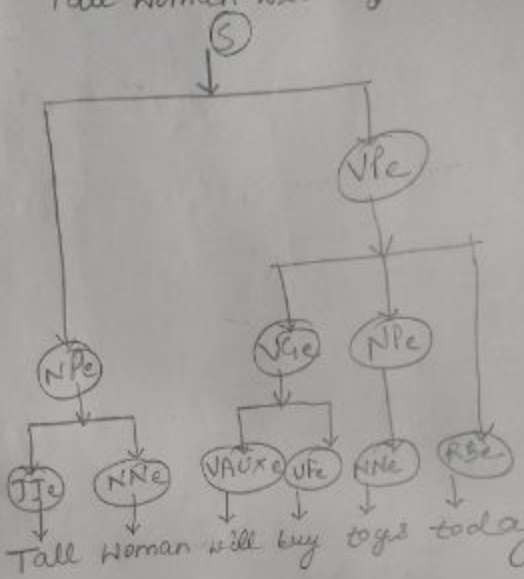
LAS, as we have presented it above, gives an evaluation of how many words were parsed correctly.However, this may not always be the point of interest. Another way of evaluating the quality of dependency parses is using sentences as the basic units. In this case,we calculate for each sentence what the percentage of correct dependencies is and then average over the sentences. The difference becomes clear when we look at a test set containing 2 sentences: Let us assume that for the first sentence, the parser assigned correct dependencies to 9 out of 10 words. For the second sentence, we have 15 out 45 words correct.

- The word-based LAS would be LAS_w = (9 + 15)/(10 + 45) = 0.436.
- The sentence-based LAS is calculated as LAS_s = (9/10 + 15/45)/2 = (0.9 + 0.333)/2 = 0.617.

In the light of this distinction, we can call the word-based LAS micro-average LAS and the sentence-based one a macro-average LAS. Another way of calculating a macro-average LAS is by averaging over all dependency types.

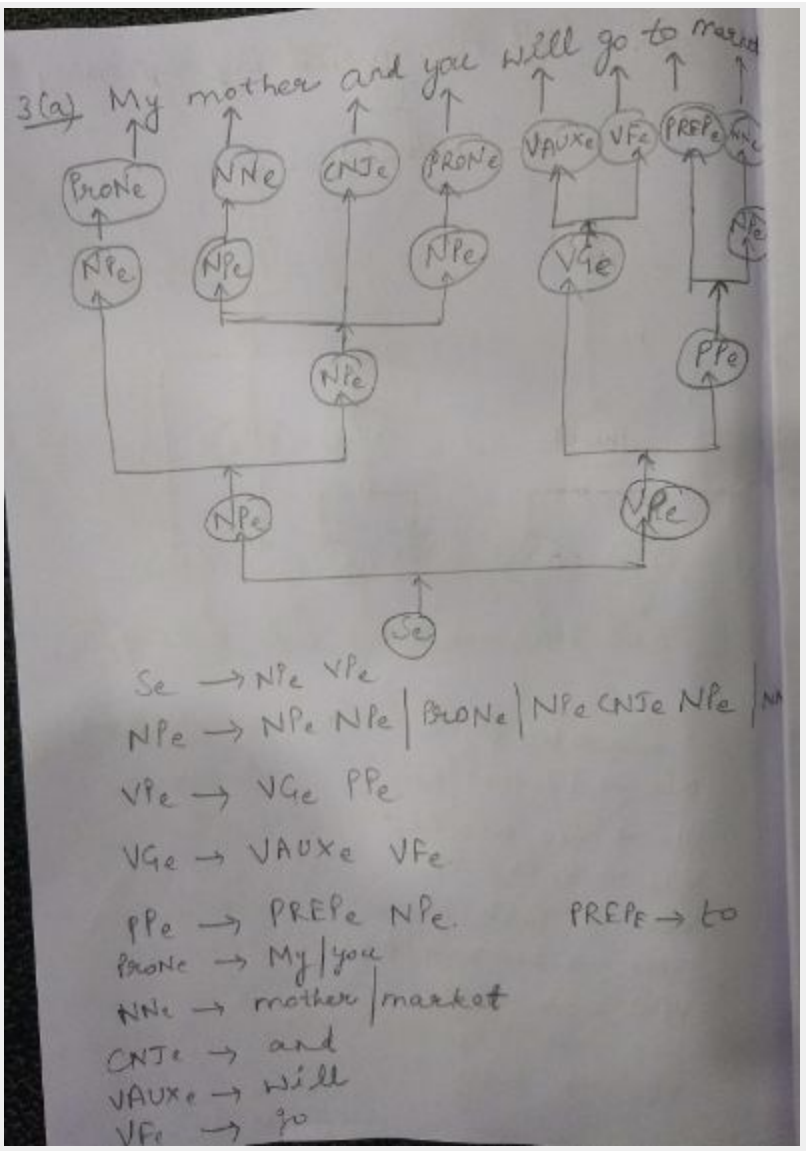
Ans 3 a)

2. Tall Woman will buy toys today.

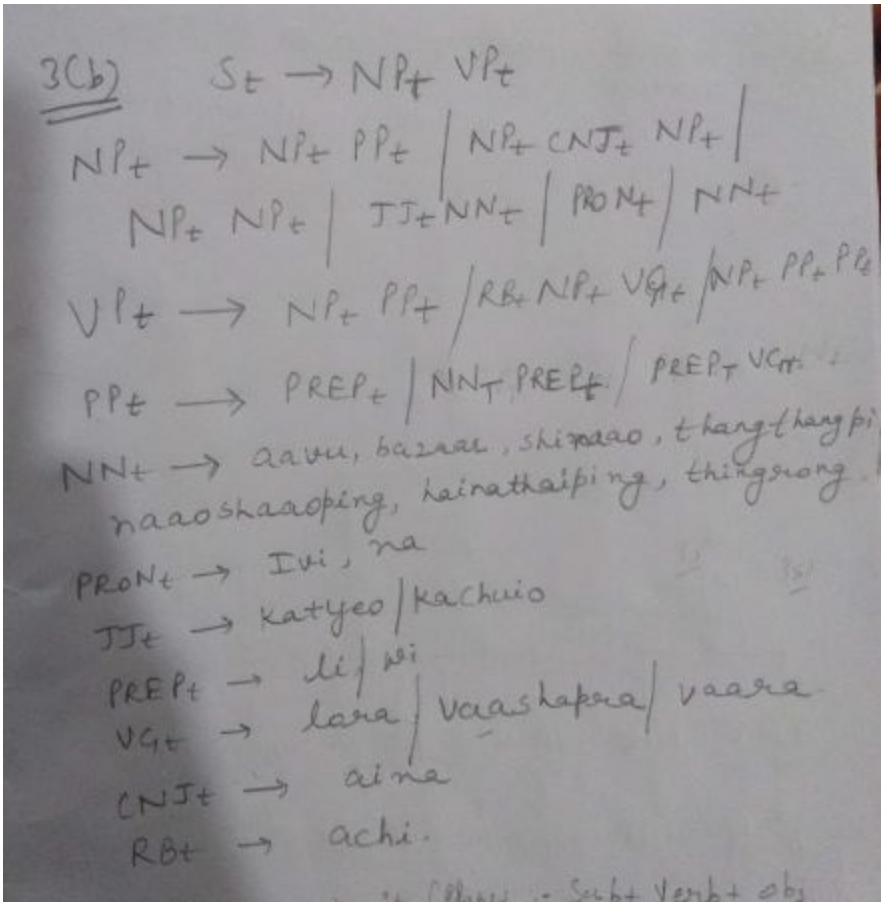


Let Woman = lady

S → NPe VPe
NPe → JJe NNe | NNe
VPe → VGe NPe RBe
VGe → VAUXe VPe
JJe → Tall
NNe → Woman | toys
VAUXe → will
VPe → buy
RBe → today.


3(a) My mother and you will go to market
ProNe
NNe
CNJe
PRONe
VAUXe
VFe
PREPe
NNe
NPe
NPe
NPe
NPe
VGe
NPe
PPe
NPe
VPe
Se
Se → NPe VPe
NPe → NPe NPe | PRONe | NPe CNJe NPe |
VPe → VGe PPe
VGe → VAUXe VFe
PPe → PREPe NPe.
PREPe → to
PRONe → My | you
NNe → mother | market
CNJe → and
VAUXe → will
VFe → go

Ans 3 b)



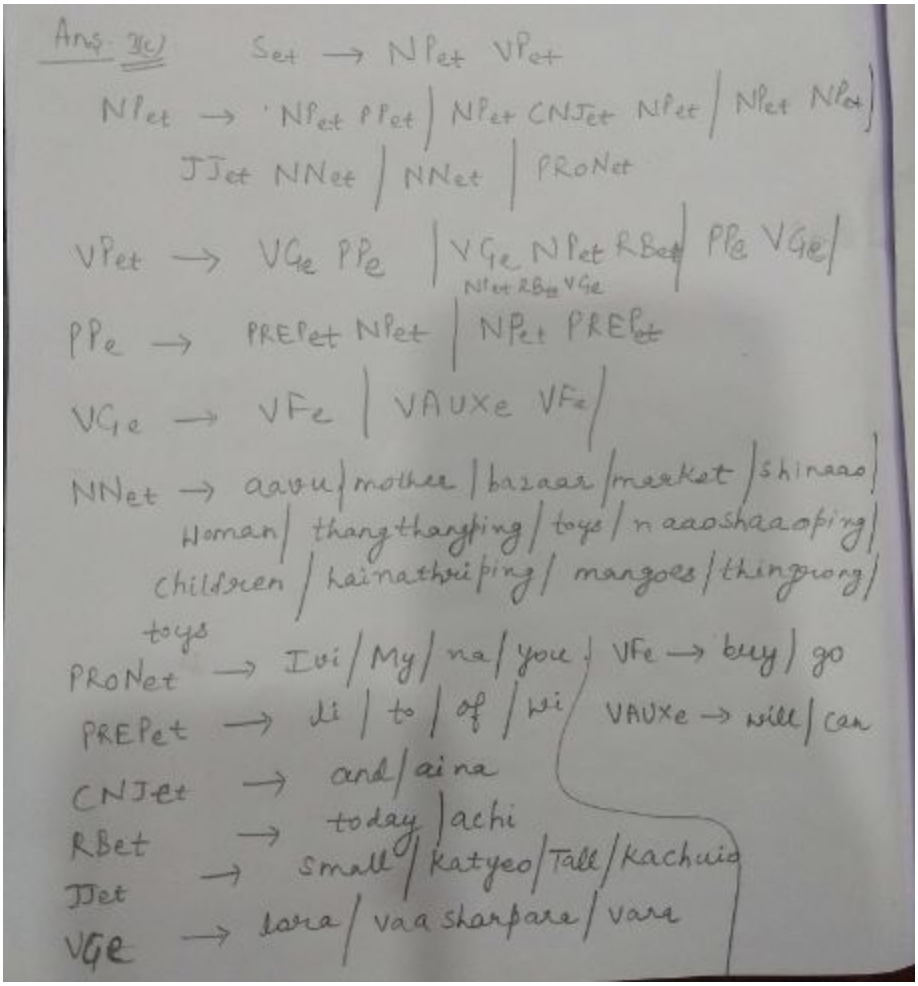
Ans 3 c)

Rules for code switching (Tangkhul-English)

i) English considers prepositions while Tangkhul considers postpositions

ii) English considers Sub + Verb +Obj order whereas Tangkhul considers Sub + Obj + Verb order.

iii) If a sentence contain English as well as Tangkhul sentences, then individual constituents like verb phrase, noun phrase, preposition/postposition phrase should be used according to each language rules.



3 d) i) Wrong : Because In Tangkhul preposition phrase should follow the order as noun phrase and then preposition

ii) Right :

iii) Right : Correct order of subject phrase and verb phrase

iv) Wrong : Preposition order is incorrect

v) Right

vi) Right