Sayar Ghosh Roy 20171047 Assignment 3

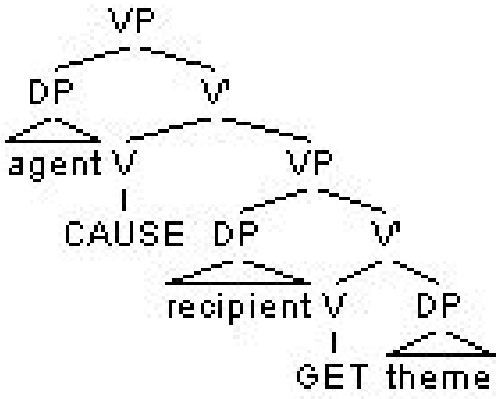
Task 1:

Summary of VP Shells:

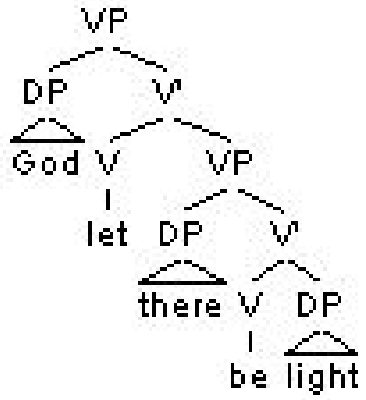
We are trying to fit the binary branching hypothesis (syntactic nodes shall have at most 2 daughters) for double object sentences.

Consider the sentence: Travis will give Betsy the receipts. Agent: Travis, Recipient: Betsy, Theme: the receipts. Now, we have to make this consistent with the binary branching hypothesis. The verb of any double object sentence can be seen as the action with the causative. For example, give is the cause of get, teach is the cause of learn, etc. This is more apparent is languages which have causative markers: In Bengali, dekhano (to show) has [ano] as the causative marker.

So, we can have the CAUSE as the main verb and the action phrase as its complement.

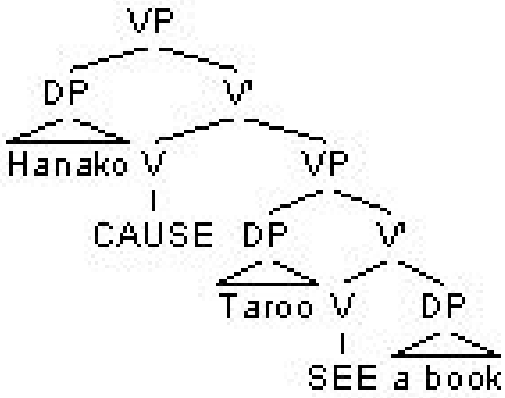


CAUSE and GET are abstract morphemes. And since we cannot have abstract morphemes in a parse structure, we must combine these morphemes at some point. Once we have this structure, we can use move operations to realise the sequence of words in the sentence we are trying to parse. Recursive Verb Phrase structures of the above type where one VP is embedded directly under another are referred to as VP shells.

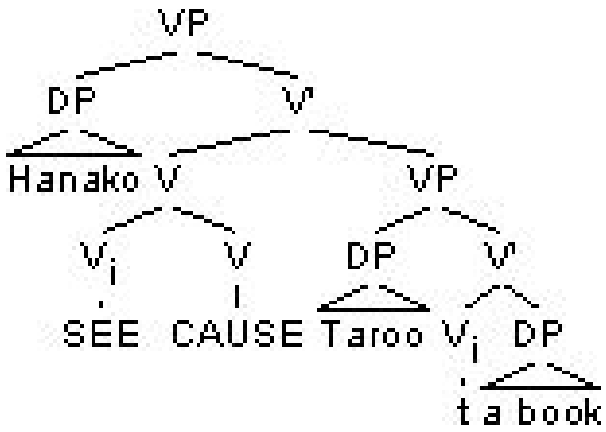


The above tree has exactly the same structure as an ordinary causative with ‘let’ as the CAUSE and ‘be’ as the action verb. In a language like Japanese where everything is mandatorily case marked, the choice of using VP shells becomes more obvious.

Consider the Bengali sentence: Hanako Taro-ke ekta boi dekhalo, meaning Hanako showed Taro a book or rather Hanako made/caused Taro to see a book which produces the following tree:



Since, one cannot have dangling abstract nodes in the structure, in order to “support” a bound morpheme, the verb moves up the tree order and this phenomenon is called abstract verb movement and is illustrated as follows.



When the agent of causation is not the agent of the action, such as in “Caesar had two legions build a bridge”, where ‘had’ is the causative and ‘Caesar’, who is the agent of the causation is not the agent of the action verb ‘build’, the structure is referred to as indirect causation.

Consider the sentence, “Caesar had a bridge built.” where ‘had’ is the cause and ‘built’ is the action verb (main verb in the subordinate clause). The agent of causation and action are the same i.e ‘Caesar’ and this is the case of direct causation. Similar arguments apply for sentences such as “Caesar built a bridge”.

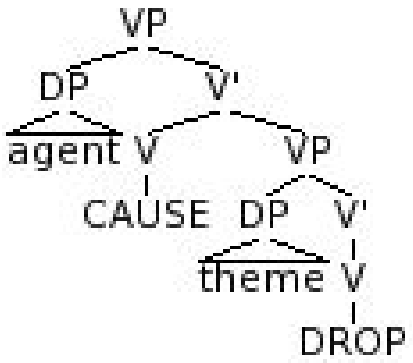
Now, consider:
1. Travis gave Betsy the receipts. (Betsy: Recipient)
2. Travis gave the receipts to Betsy. (Betsy: Goal)

We can replace Betsy by Kolkata in the above two sentences and the 1st one would not make sense semantically as Kolkata is not animate. Here, sentence 1 is a double object sentence and sentence 2 is a double complement sentence where ‘the receipts’ is the complement to the trace of ‘gave’ and ‘Betsy’ is the complement to the higher level movement of the main verb ‘gave’. Also, every double object sentence need not have a double complement counterpart and vice-versa. Hence the mapping is not one to one. Consider: “The scandal gave an idea to the reporter.” Its double complement counterpart would be “The scandal gave an idea to the reporter.”, which is an illegal sentence in English. Also, for a verb like “put” in a double complement sentence, there does not exist a double object counterpart.

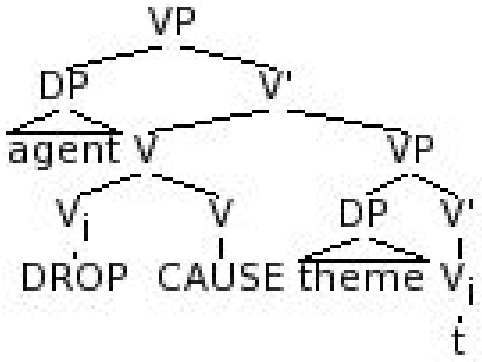
Some words can be used in 2 senses. In one case, they are inchoative verbs and in another, they act as causatives. For example, in the sentence, “The ball dropped”, ‘dropped’ is an

inchoative but in “The children dropped the ball.”, dropped is a causative. When used as inchoatives, the verbs are intransitive and denote a manner of motion, and the subject is the theme argument (expressing the entity undergoing motion). When used as causatives, the verbs are transitive, the subject is an agent or cause initiating the motion, and the theme argument appears as the direct object.

Inchoative Verb

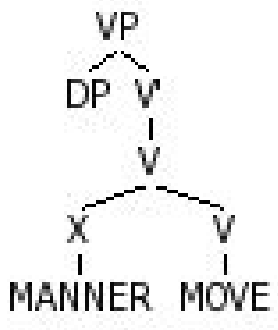


Causative before abstract verb movement



Causative after abstract verb movement

Manner of motion verbs denote some sort of motion each differing in exactly how the theme argument undergoes motion. The inchoative variants of these verbs are themselves decomposable into basic predicate (move) and a specification of manner.

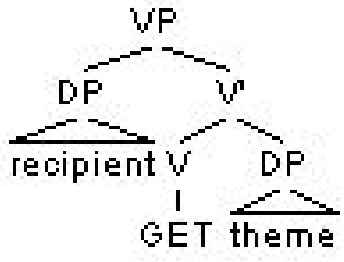


Manner-of-motion verbs.

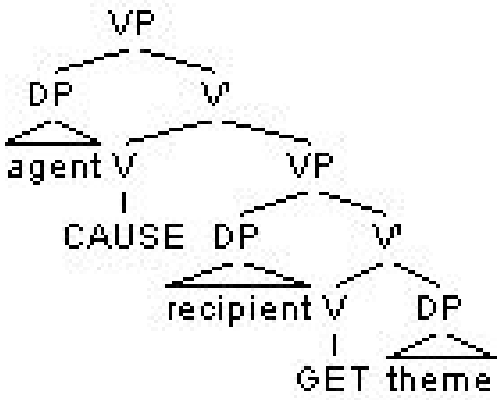
Manner incorporation is not restricted to manner-of-motion verbs. Verbs of speaking such as groan, grunt, whisper can be decomposed into predicates SAY or SPEAK and a particular manner.

Consider the sentences:
1. Betsy got the receipts.
2. Travis got Betsy the receipts.

The inchoative ‘got’ in (1) is transitive rather than intransitive but the causative in (2) introduces an additional argument which is the agent ‘Travis’.



Inchoative ‘got’

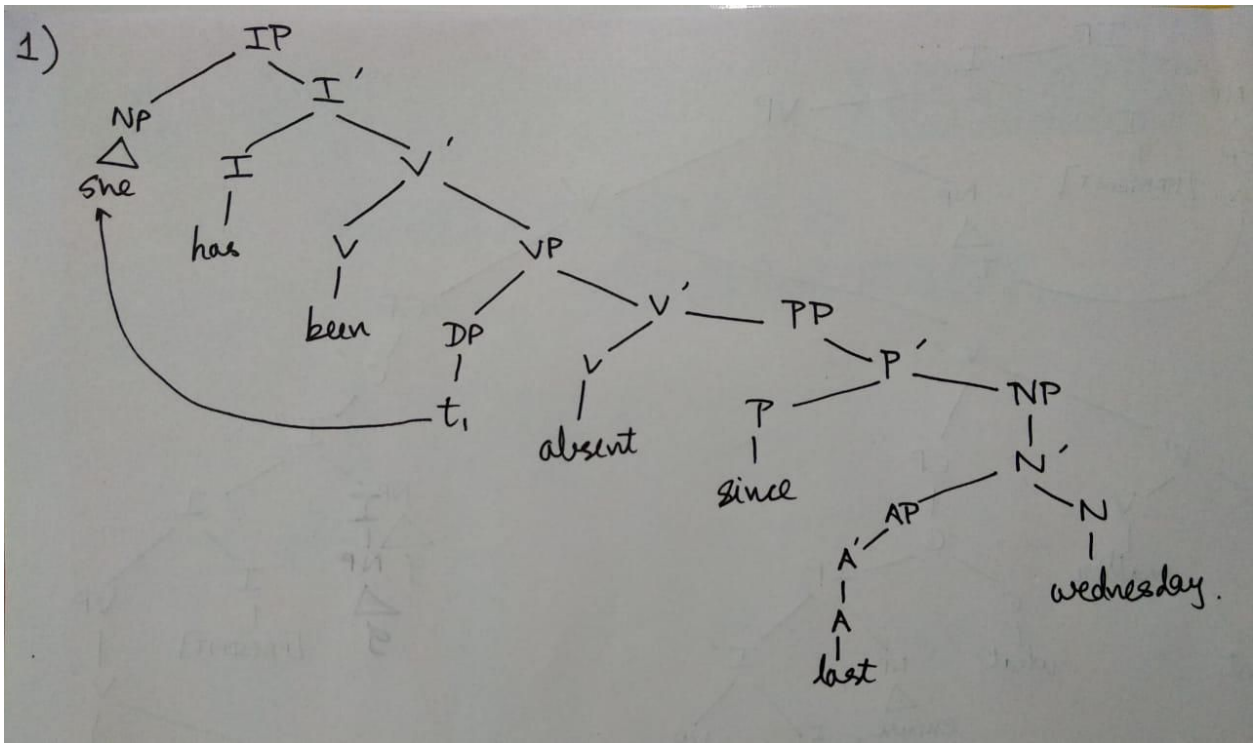


Causative with an additional agent

Task 2

1.
She has been absent since last Wednesday.

Stanford Parser Output:
[ROOT
[S
[NP [PRP She]]
[VP [VBZ has]
[VP [VBN been]
[ADJP [JJ absent]]
[PP [IN since]
[NP [JJ last] [NNP Wednesday]]]]]
]]



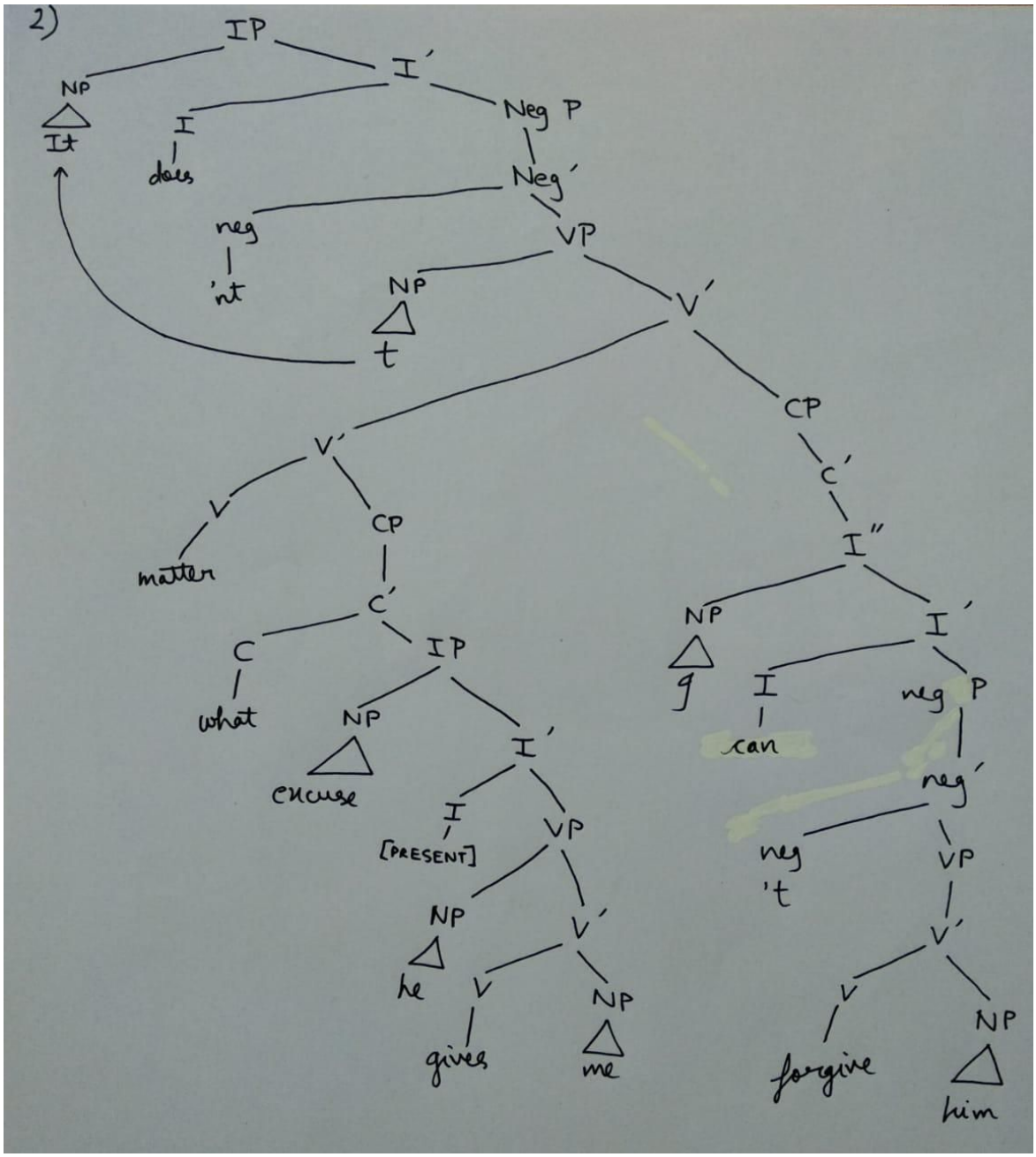
In the Stanford Parser, we see a ternary branching which is not valid in X-bar. Apart from that, the parsing is very similar leaving schema constructs aside.

2.

It doesn't matter what excuse he gives me, I can't forgive him.

Stanford Parser Output:

```
[ROOT
[S
[S
[NP [PRP It]]
[VP [VBZ does] [RB n't]
[VP [VB matter]
[SBAR
[WHNP [WP what] [NN excuse]]
[S
[NP [PRP he]]
[VP [VBZ gives]
[NP [PRP me]]]]]]]]
[, ,]
[NP [PRP I]]
[VP [MD ca] [RB n't]
[VP [VB forgive]
[NP [PRP him]]]]
]]
```

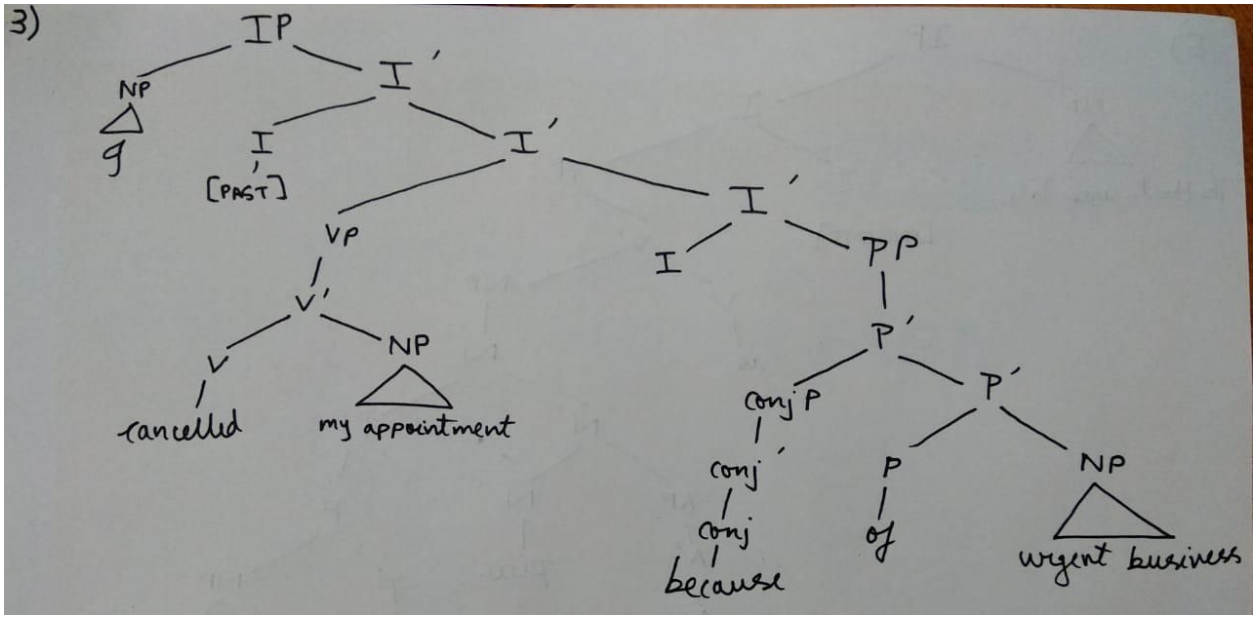


I used Neg P in my X-bar structure which is not present in the Stanford POS.

3.
I canceled my appointment because of urgent business.

Stanford Parser Output:
[ROOT
[S

[NP [PRP I]]
[VP [VBD canceled]
[NP
[NP [PRP$ my] [NN appointment]]
[PP [RB because] [IN of]
[NP [JJ urgent] [NN business]]]]]
]]

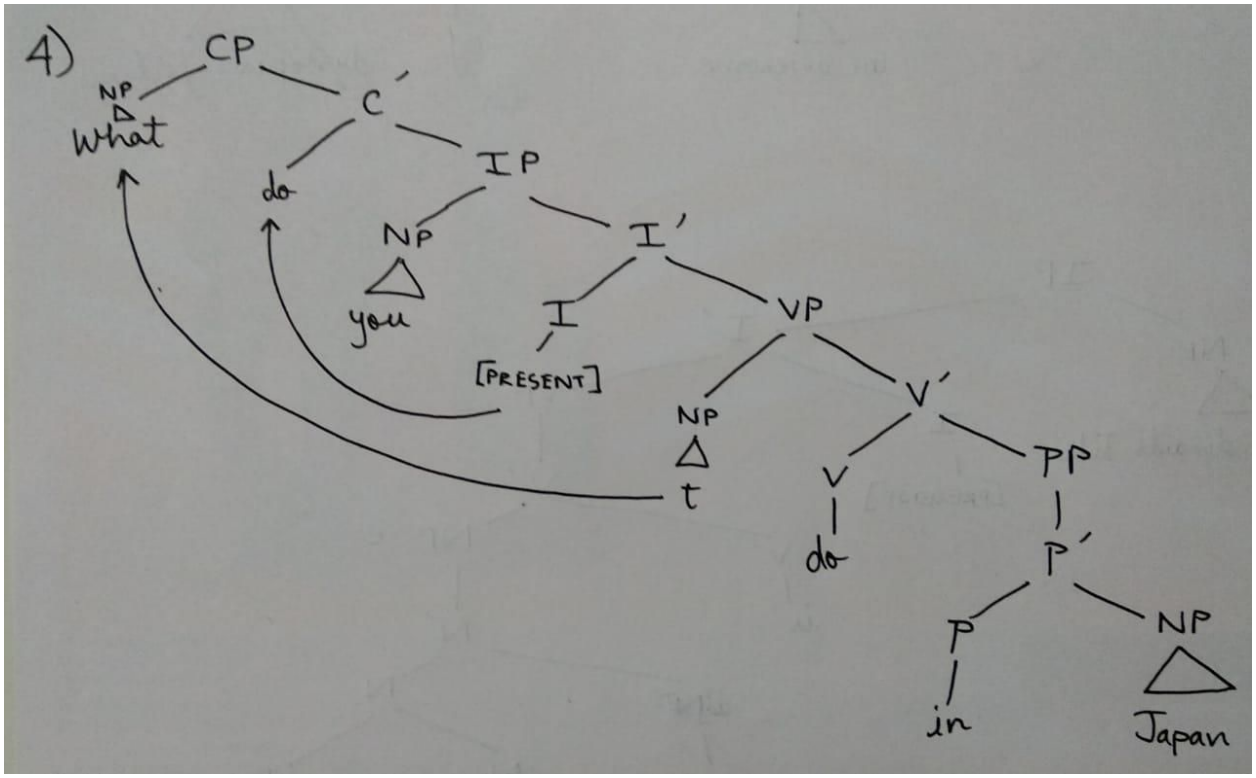


Ternary branching in PSG allowing ‘because’ and ‘of’ to be at the same level. This is not allowed in X-bar.

4.

What do you do in Japan?

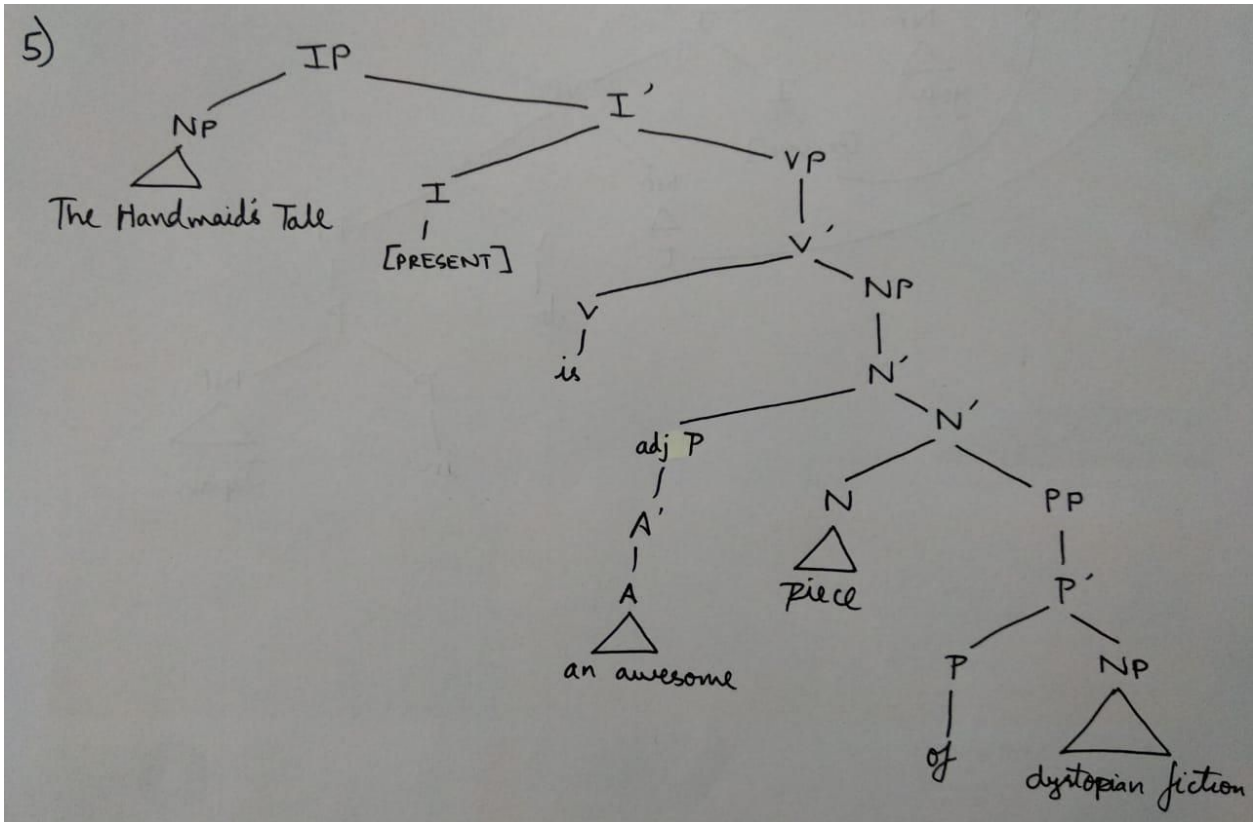
Stanford Parser Output:
[ROOT
[SBARQ
[WHNP [WP What]]
[SQ [VBP do]
[NP [PRP you]]
[VP [VB do]
[PP [IN in]
[NP [NNP Japan]]]]]
[. ?]]]



Ternary branching not allowed in X-bar. Different tag used in the Stanford Parser for ‘What’.

5.
The Handmaid’s Tale is an awesome piece of dystopian fiction.

Stanford Parser Output:
[ROOT
[S
[NP
[NP [DT The] [NNP Handmaid] [POS 's]]
[NN Tale]]
[VP [VBZ is]
[NP
[NP [DT an] [JJ awesome] [NN piece]]
[PP [IN of]
[NP [NN dystopian] [NN fiction]]]]]
]]



Stanford parser tags ‘dystopian’ as NN and uses ternary branching. Apart from that, other aspects are consistent with the X-bar schema.

6.
OK. Now what?

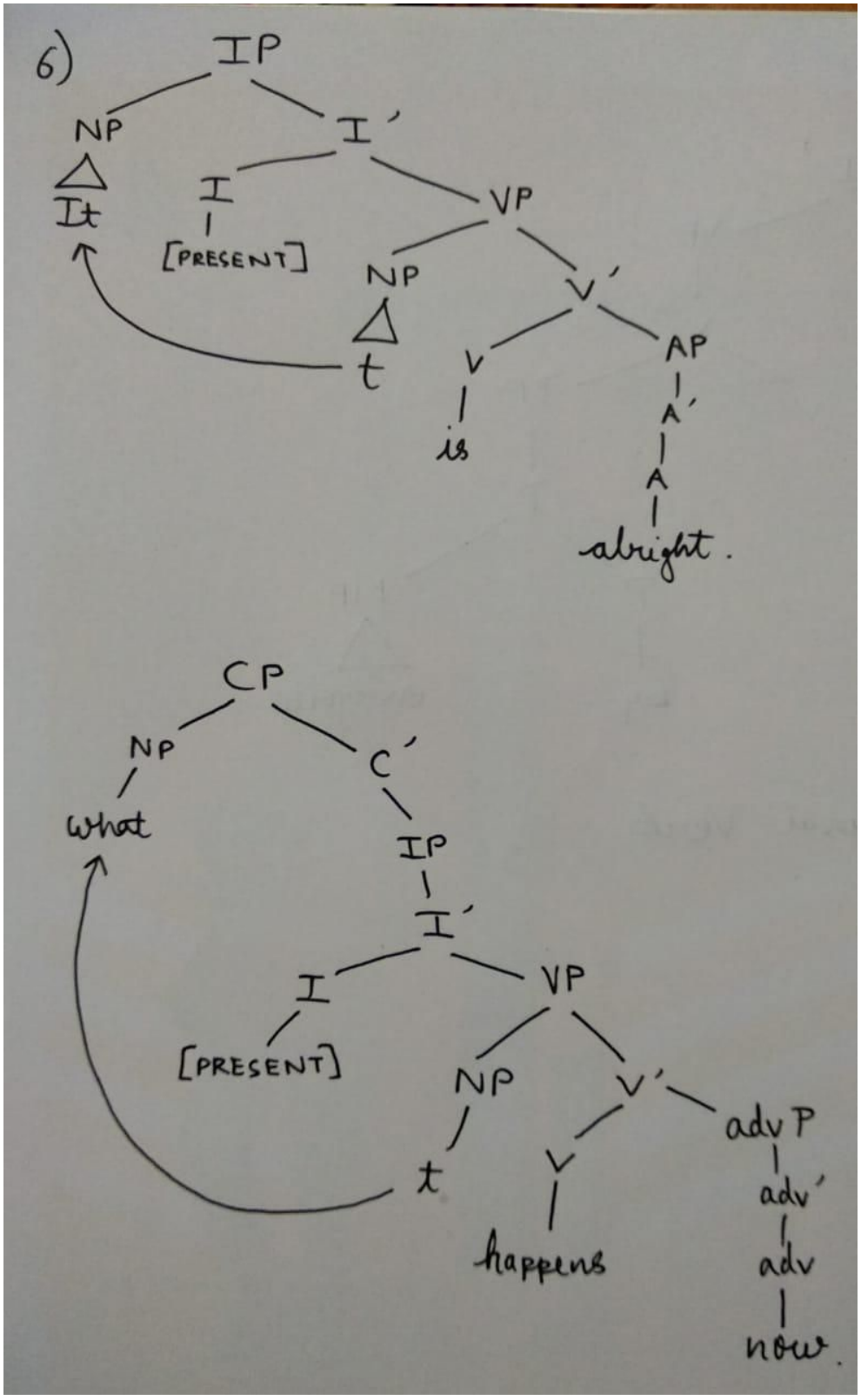
Stanford Parser Output:

[ROOT
[INTJ [UH OK]]]

[ROOT
[X
[ADVP [RB Now]]
[NP [WP what]]
[. ?]]]

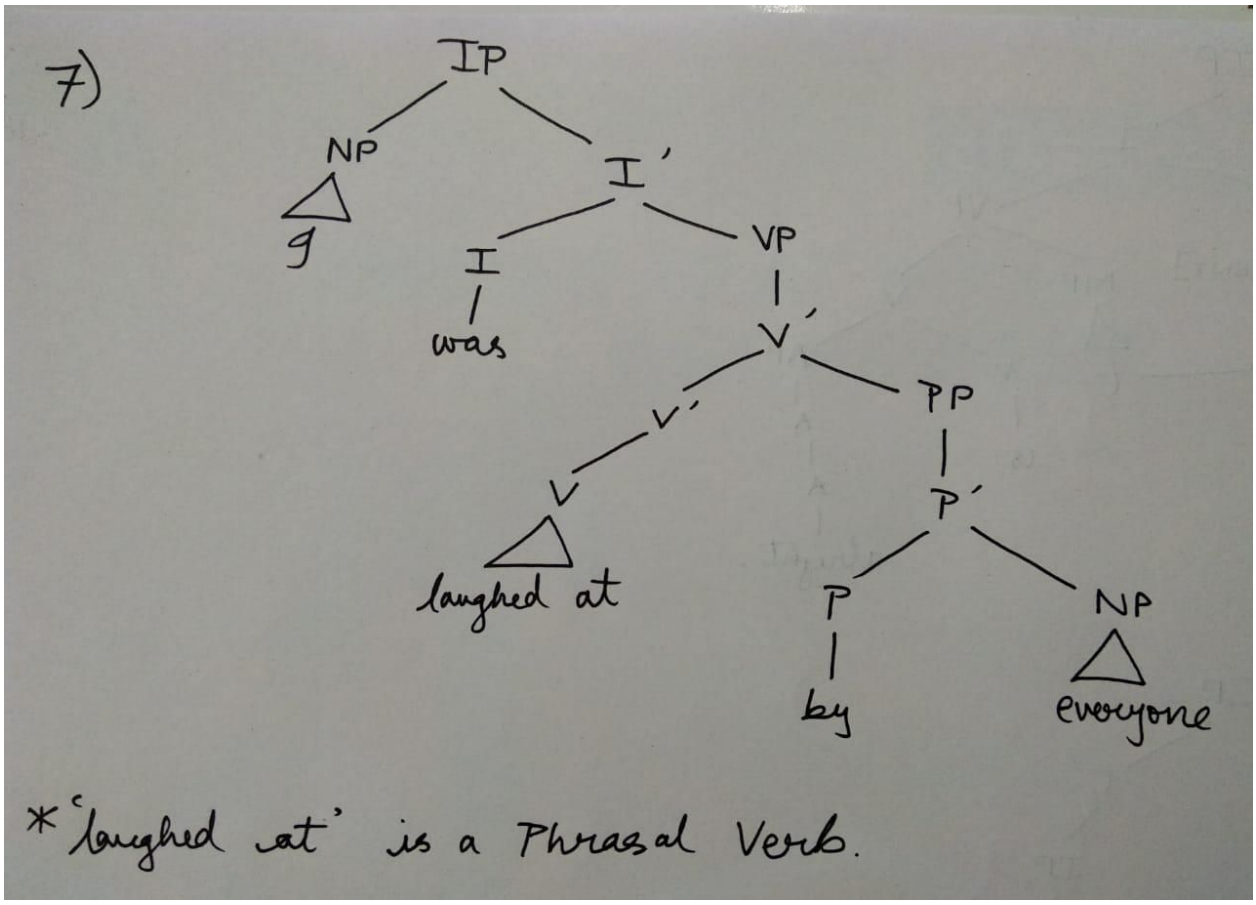
I used the sentences:

It is alright. What happens now? - Which conveys the same meaning as the above utterance.
X-bar schema can not be properly used as the given utterances are not complete sentences.



7.
I was laughed at by everyone.

Stanford Parser Output:
[ROOT
[S
[NP [PRP I]]
[VP [VBD was]
[VP [VBN laughed]
[PP [IN at]
[PP [IN by]
[NP [NN everyone]]]]]]
]]



Stanford Parser tags ‘at’ as an IN which is a preposition and fails to recognize that ‘laughed at’ is a phrasal verb expression.

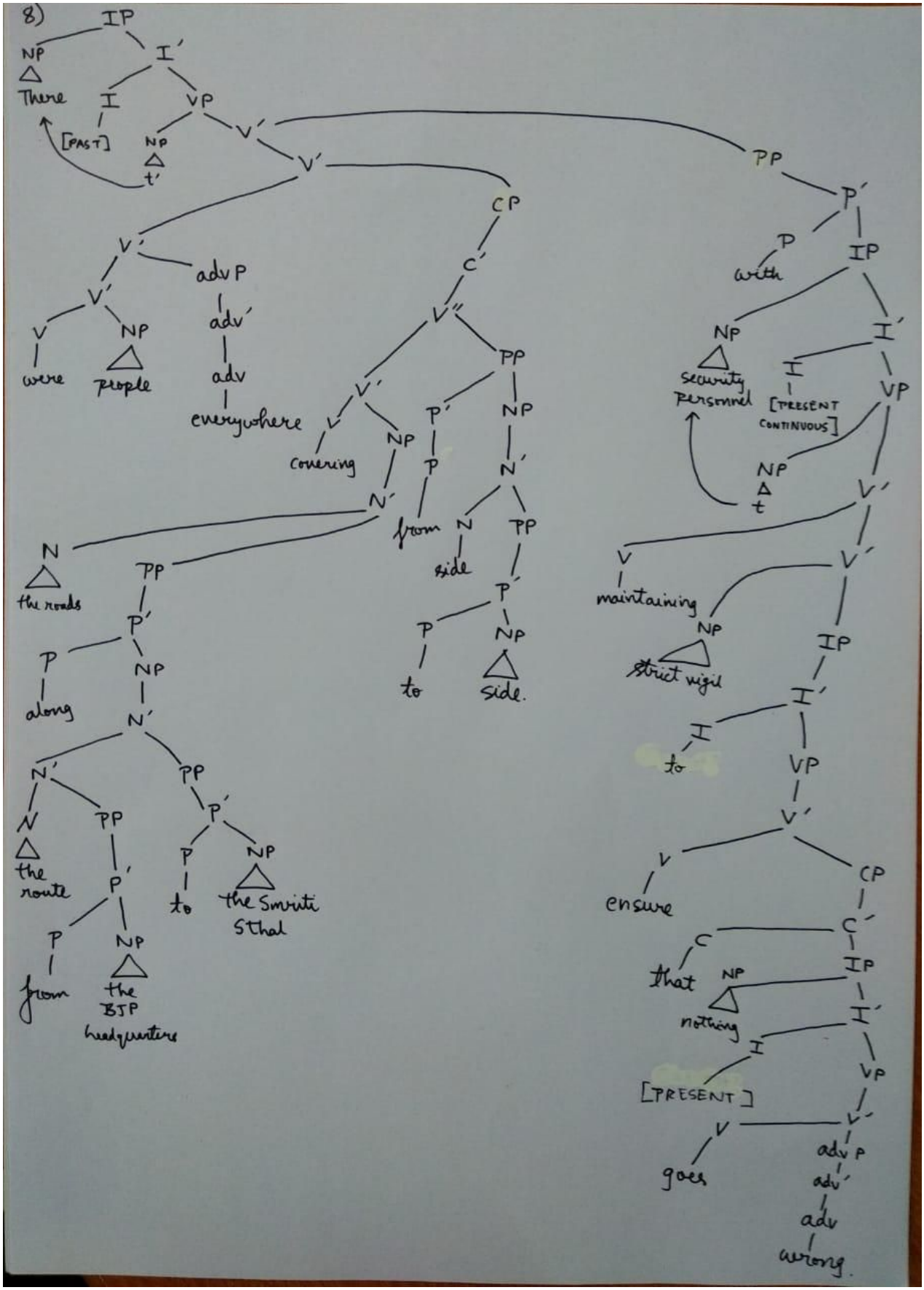
8.

There were people everywhere, covering the roads along the route from the BJP headquarters to the Smriti Sthal from side to side, with security personnel maintaining strict vigil to ensure that nothing goes wrong.

Stanford Parser Output:

```
[ROOT
[S
[NP [EX There]]
[VP [VBD were]
[NP [NNS people]]
[ADVP [RB everywhere]]
[, ,]
[S
[VP [VBG covering]
[NP [DT the] [NNS roads]]
[ADVP [IN along]
[NP [DT the] [NN route]]
[PP [IN from]
[NP [DT the] [NNP BJP] [NN headquarters]]]
[PP [TO to]
[NP [DT the] [NNP Smriti] [NNP Sthal]]]]
[PP [IN from]
[NP [NN side]]]
[PP [TO to]
[NP [NN side]]]
[, ,]
[PP [IN with]
[NP
[NP [NN security] [NNS personnel]]
[VP [VBG maintaining]
[NP [JJ strict] [NN vigil]]
[S
[VP [TO to]
[VP [VB ensure]
[SBAR [IN that]
[S
[NP [NN nothing]]
[VP [VBZ goes]
[ADJP [JJ wrong]]]]]]]]]]]]]
]]
```

Stanford parser produces a really ambiguous structure with 7 projections out of a VP. The structure makes sense but is infeasible in X-bar.


8)
IP
NP
There
I'
I
[PAST]
VP
NP
t'
V'
V'
V'
V'
V
were
NP
people
adv P
adv'
adv
everywhere
V'
V
covering
NP
N'
N
the roads
PP
P'
P
along
NP
N'
N'
N
the route
PP
P'
P
from
NP
the BJP headquarters
PP
P'
P
to
NP
the Smriti Sthal
CP
C'
V''
PP
P'
P
from
NP
N'
N
side
PP
P'
P
to
NP
side.
PP
P'
P
with
IP
NP
security personnel
I'
I
[PRESENT CONTINUOUS]
VP
NP
t
V'
V
maintaining
V'
NP
strict vigil
IP
I'
I
to
VP
V'
V
ensure
CP
C'
C
that
IP
NP
nothing
I'
I
[PRESENT]
VP
V'
V
goes
adv P
adv'
adv
wrong.

General Ideas on the 2 parsings:

The 2 approaches are very different. X-bar schema allows only binary branching while Phrase Structure Grammar allows any valid tree structure. Computationally, producing an X-bar tree for a given sentence is a very expensive operation while that for a PSG tree is far less complex. However, X-bar schema is considered very elegant as one can explain a lot of different phenomena with a very few rules. Different concepts such as movement and tense marking is used while parsing a sentence using the X-bar schema. PSG can be modelled using rules with conditional probabilistic scores for each under a context and implemented as a statistical tagger. One can use an efficient bottom up approach to find the PSG parse for a particular sentence but things are not so simple in the case of X-bar trees. Movement and tense marking are difficult to implement let alone implementation in an efficient and computationally simple manner. Hence, although intuitive and somewhat “magical” in the sense that it accommodates so many sentences with so little overhead in terms of rules, X-bar trees are not used in the modelling of sentences as it is difficult to implement in a computational environment.