

# Computational Linguistics - 1, Assignment 3

Zubair Abid, 20171076

## 1 Task 1: Reading Assignment

### 1.1 Causes for the existence of VP-shells: Binary-branching hypothesis and double object sentences

The primary reason for this chapter is because of the contradictory syntactic structures brought about when considering double object sentences in the framework of the binary branching hypothesis.

**Binary Branching hypothesis:** Earlier while discussing simple sentences and constructing X-Bar structures for verbs with more than two arguments, the author had said to discard more than 2 branches *"it is now widely assumed that syntactic structure is at most binary-branching (in other words, binary-branchingness is assumed to be a formal universal)"*.

**Double object sentences:** Double object sentences are those where the verb seems to be associated with three semantic arguments, as illustrated in the English example: **Travis will give Betsy the receipts.**

With such sentences one would ideally like to represent both the verb-related objects as complements of the verb, but naturally as one can see this seems to violate the binary branching hypothesis.

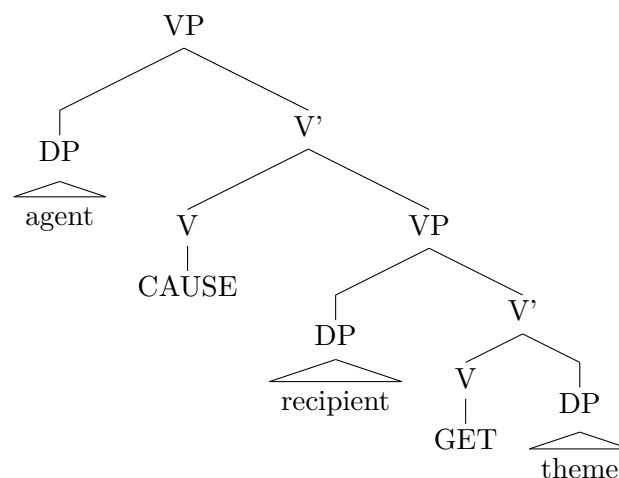
### 1.2 Grammatical structures enabling a solution; and the VP shell structure

The solution hinges on the fact that ditransitive verbs like give can be semantically decomposed into a causative part and a remainder whose meaning differs according to the verb in question. For example,

**feed:** cause to eat

**give:** cause to get

This suggest that such verbal structures in sentences like the example given can be represented by a recursive VP structure like in the one shown below:



These recursive structures are known as VP shells.

### 1.3 Ordinary causative sentences and their parallels with double object sentences; abstract verb movement

**Ordinary causative sentences:** The semantic notion of causation is overtly expressed by the verb *let*.

Now, *let* takes small clause complements. Taking these facts into consideration, it's easy to see how this sentence fits into the syntactic structure as shown above. To further investigate the parallels between causative and double object sentences, the author then takes examples in Japanese, using **Abstract verb movement** to fit grammaticality. Their final assumption that VP Shells for double object sentences in English are analogous to Japanese relies on the observation that

*"From a structural point of view, the only difference between the Japanese structures and their English counterparts is the direction in which V takes phrasal complements."*

### 1.4 Clarification on CAUSE, direct vs indirect causation

*"Before extending the VP shell analysis to further sentence types, we should explicitly state that we are not claiming that CAUSE is completely synonymous with overt causative verbs like cause, let, make, and the like. These overt causative verbs express what has been called indirect causation, where the causing event and the caused event are conceptualized as two separate events. By contrast, CAUSE expresses direct causation, and the causing event is not conceptualized as distinct from the caused event."*

### 1.5 Double complement sentences

This section summarizes some rules and clarifications on some structures involved with double complement sentences, primarily around these verbs (or their analogues):

1. **Give and Send:**

Draws a parallel between double object and double complement sentences, and uses that to reason for the similarity between *give* and *send*.

2. **Put:**

Gives a basic example

3. **Persuade:**

Gives a basic example

### 1.6 Extension of theory

Here on, the chapter focuses on extending the applicability of the VP-shell theory to other problems, such as

#### 1.6.1 Causative Alternation

Extending the VP shell analysis to

1. **The alternation between inchoative verbs and their homonymous causative counterparts**
2. **Manner incorporation, Get**

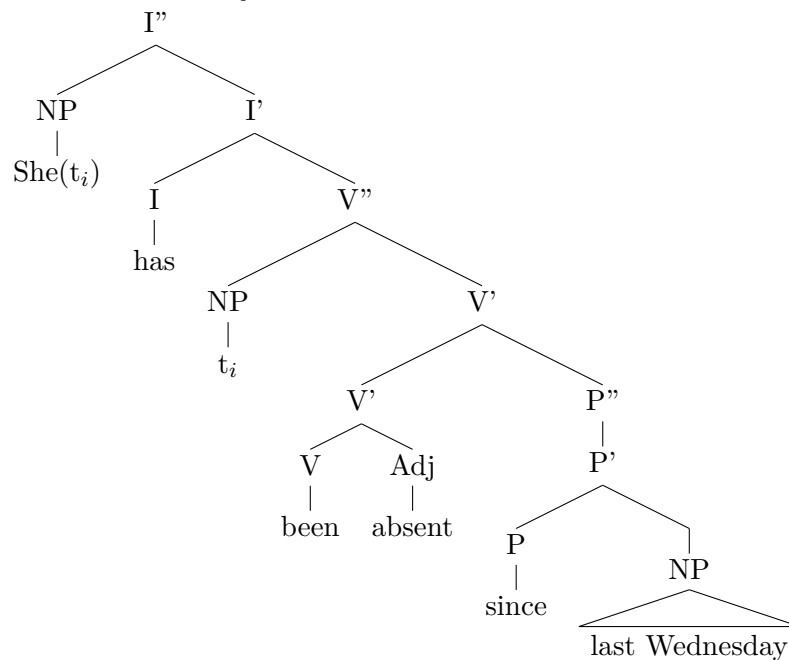
#### 1.6.2 Locality constraints on Idioms

Idioms are not the sum of their parts, and to deal with possible parse/ issues with them, it has been traditional in generative grammar (Marantz 1984) to (attempt to) impose a locality constraint on them.

The VP shell analysis methodology allows us to preserve the locality constraint over seemingly discontinuous idioms.

## 2 Task 2: X-Bar Schema and Phrase Structure

1. *She has been absent since last Wednesday.*



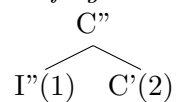
Stanford Parser output:

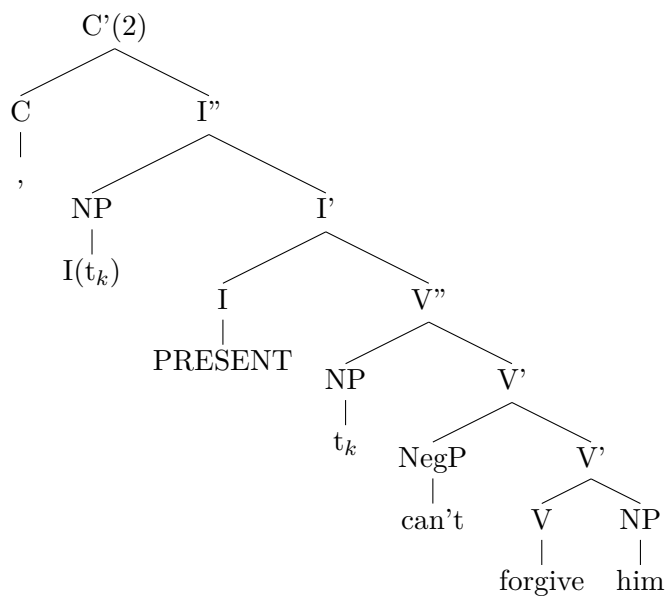
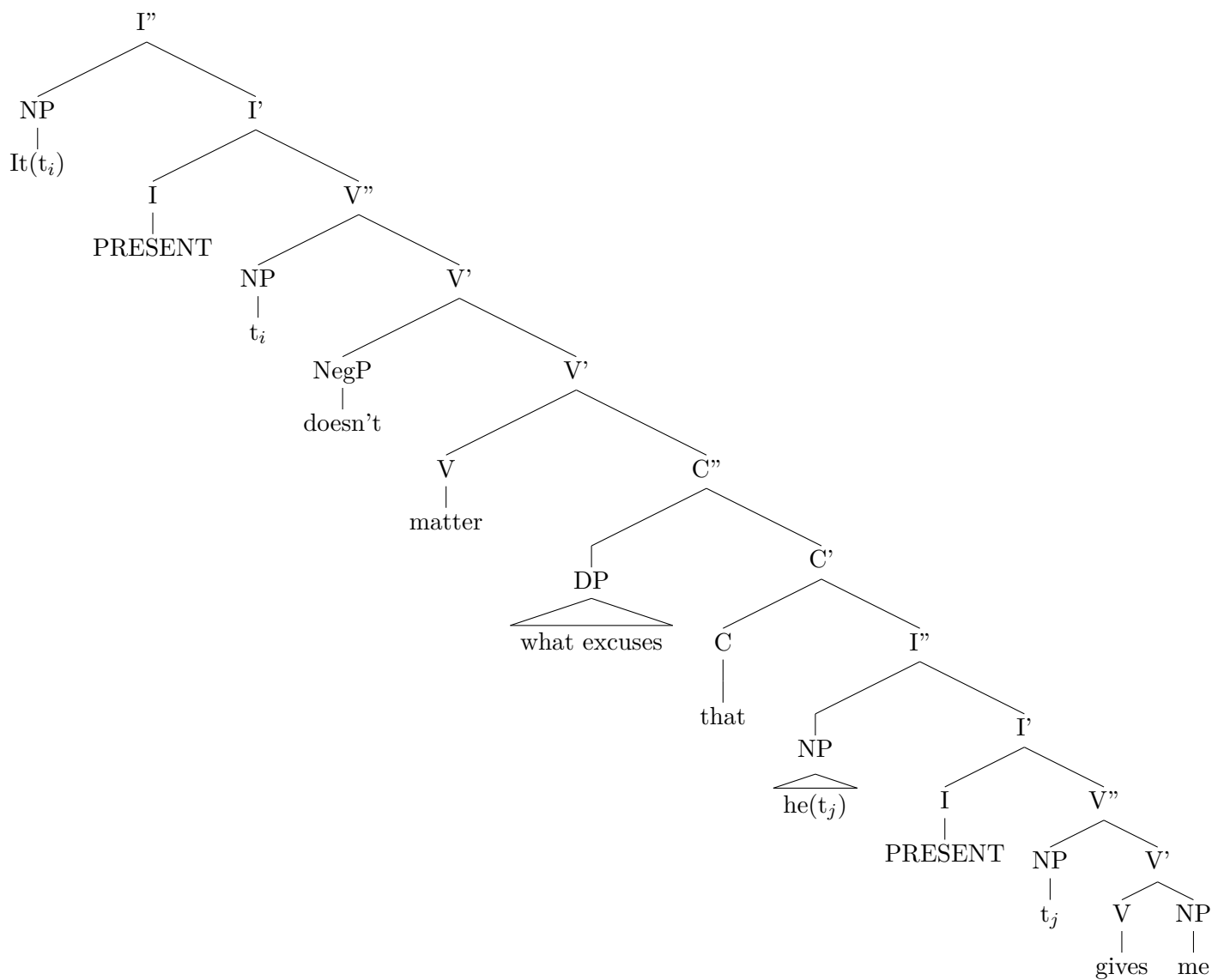
```
(ROOT
(S
(NP (PRP She))
(VP (VBZ has)
(VP (VBN been)
(ADJP (JJ absent))
(PP (IN since)
(NP (JJ last) (NNP Wednesday))))))
(. .)))
```

### DIFFERENCES WITH THE PARSES:

- (a) X-bar is binary branching, the stanford parse is not
- (b) The stanford parse is shallower
- (c) "been", "absent", and "since" are considered at the same depth of relatedness

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





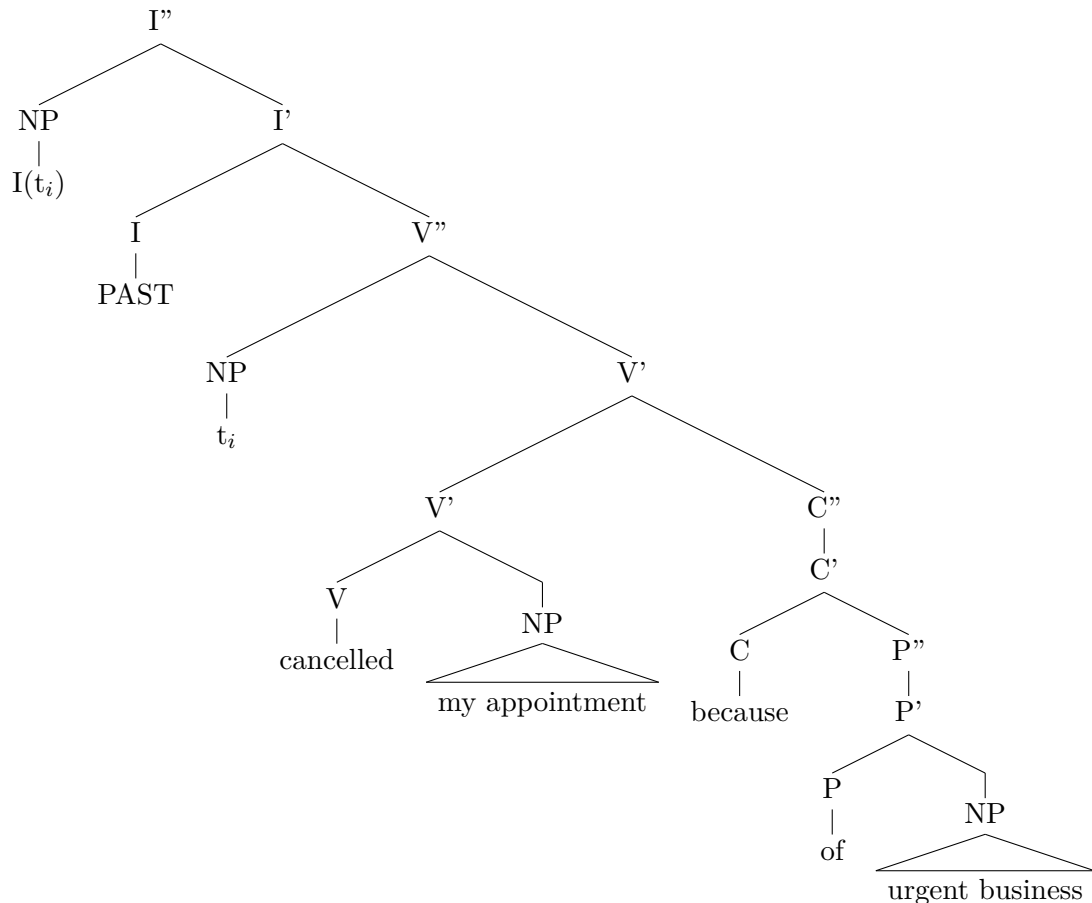
Stanford Parser output:  
 (ROOT  
 (S  
 (S  
 (NP (PRP It))

(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))))  
 (. .)))

### DIFFERENCES WITH THE PARSSES:

- (a) X-bar is binary branching, the stanford parse is not
- (b) The stanford parse is shallower

3. *I canceled my appointment because of urgent business.*



Stanford Parser output:

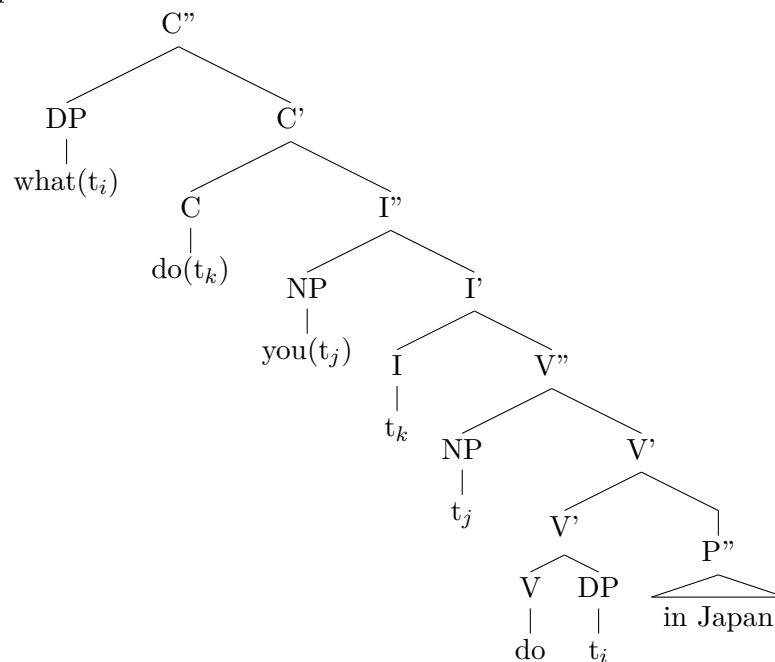
(ROOT  
 (S  
 (NP (PRP I))  
 (VP (VBD canceled)  
 (NP

(NP (PRP\$ my) (NN appointment))  
 (PP (RB because) (IN of)  
 (NP (JJ urgent) (NN business))))))  
 (. .)))

#### DIFFERENCES WITH THE PARSES:

- (a) X-bar is binary branching, the stanford parse is not
- (b) The stanford parse is shallower
- (c) "because" and "of" are closer in the Stanford Parser

4. *What do you do in Japan?*



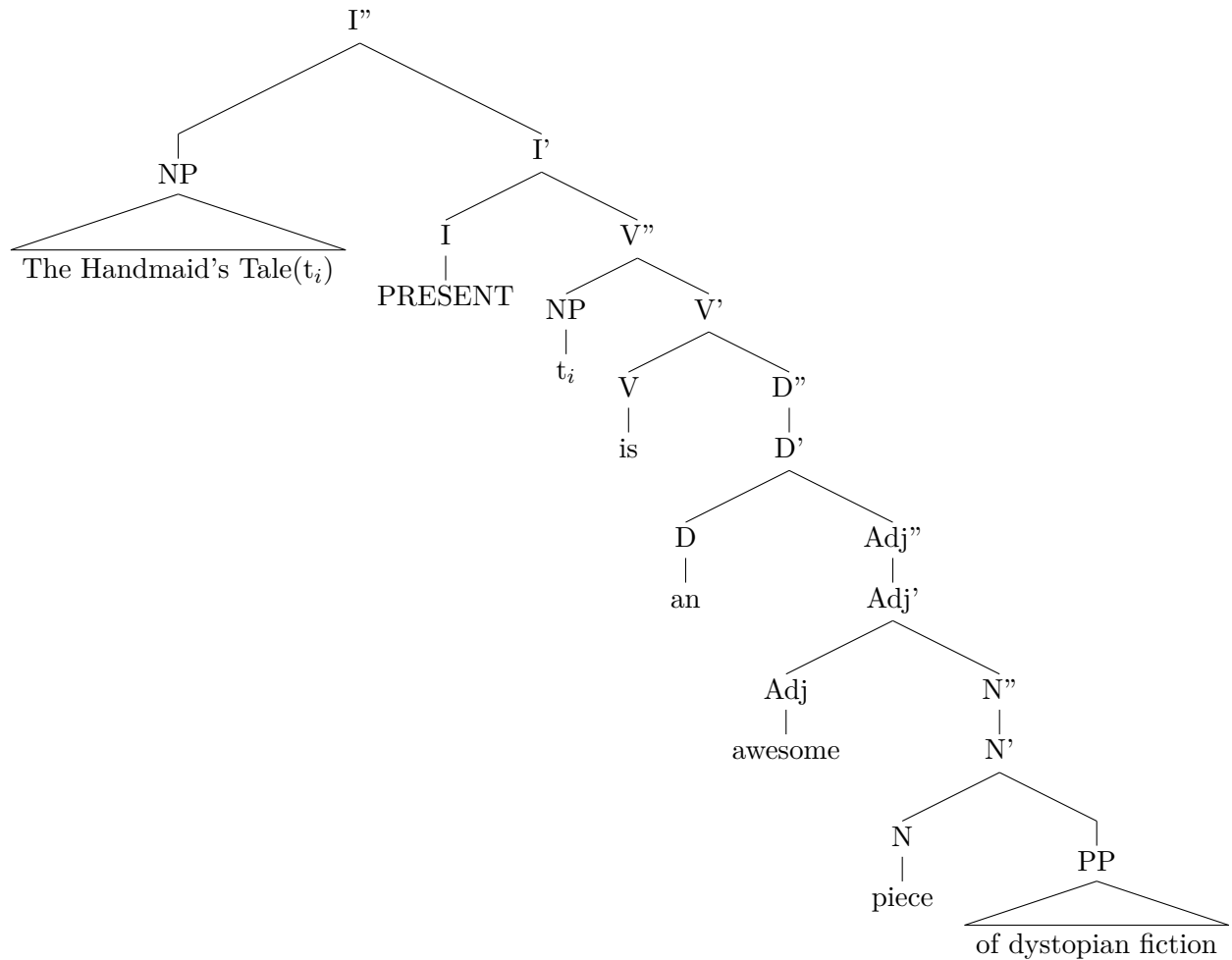
Stanford Parser output:

(ROOT  
 (SBARQ  
 (WHNP (WP What))  
 (SQ (VBP do)  
 (NP (PRP you))  
 (VP (VB do)  
 (PP (IN in)  
 (NP (NNP Japan))))))  
 (. ?)))

#### DIFFERENCES WITH THE PARSES:

- (a) X-bar is binary branching, the stanford parse is not
- (b) The stanford parse is shallower

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))))))
(. .)))

```

#### DIFFERENCES WITH THE PARSSES:

- (a) The stanford parse is shallower
- (b) The stanford parser does not NER "The Handmaid's Tale"

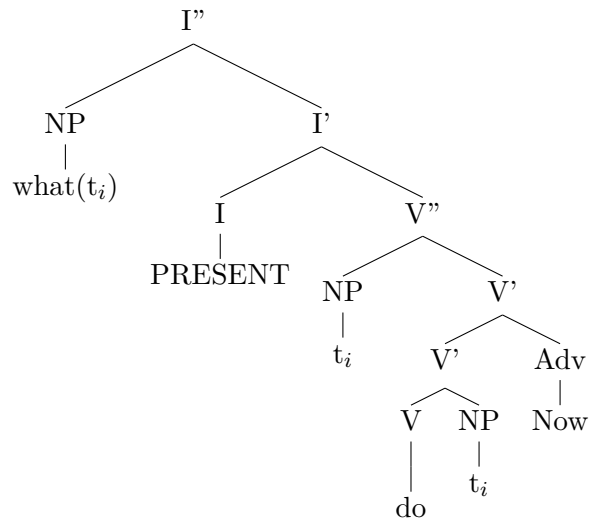
6. *OK. Now what?*

We transform the sentence to "OK. What now?", to fit the wh-scheme better.

```

INTJ
|
OK

```



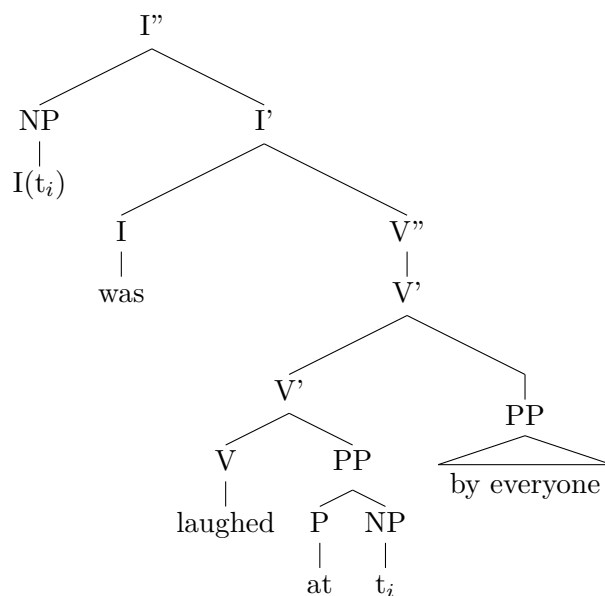
Stanford Parser output:  
 (ROOT  
 (INTJ (UH OK) (. .)))

(ROOT  
 (X  
 (ADVP (RB Now))  
 (NP (WP what))  
 (. ?)))

### DIFFERENCES WITH THE PARSES:

- (a) The stanford parse is shallower
- (b) Stanford parses the original sentence fine

7. *I was laughed at by everyone.*



Stanford Parser output:  
 (ROOT  
 (S

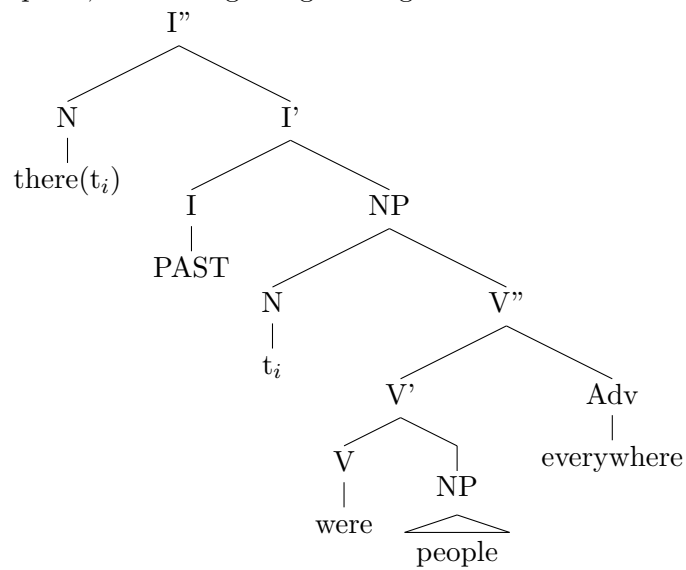


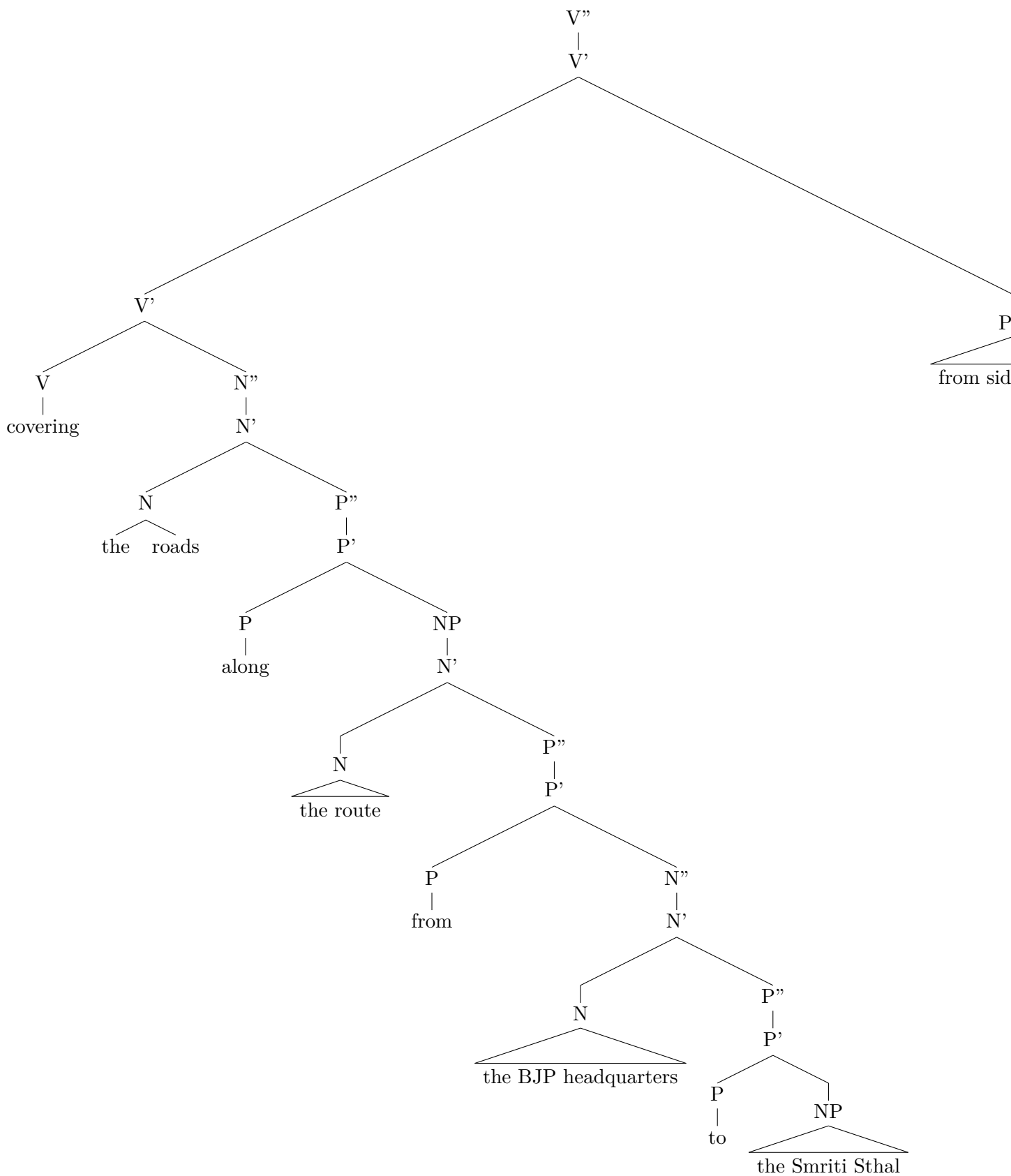
(NP (PRP I))  
 (VP (VBD was)  
 (VP (VBN laughed)  
 (PP (IN at)  
 (PP (IN by)  
 (NP (NN everyone))))))  
 (. .))

# **DIFFERENCES WITH THE PARSES:**

- (a) The stanford parse is shallower

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.*  
 The sentence was split into parts, as it was getting too big and because it makes logical sense to.





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)))))))))))))  
 (. .)))

#### **DIFFERENCES WITH THE PARSES:**

- (a) X-bar is binary branching, the stanford parse is not
- (b) The stanford parse is shallower

### **2.1 Overall differences between Stanford and X-Bar**

1. X-bar is binary branching, the stanford parse is not
2. The stanford parse is shallower
3. Stanford parser does not finely define the closeness of relation in a possible multi-branch
4. Stanford parser has issues with some NER