Language Modelling

D. Thenmozhi

Associate Professor

SSNCE

Language Modelling

- Model is a description of some complex entity or process
- A language model is a description of language
- Language Modelling
  - Handling of natural language, a complex entity that contains a large number of sentences, through a computer-based program
- Two approaches
  - Grammar-based
  - Statistical

Language Modelling Approaches

- Grammar-based language model
  - Approach uses the grammar of a language to create its model
  - Represents the syntactic structure of language
  - Grammar consists of rules
- Statistical language modelling
  - Fundamental tasks in many NLP applications (MT, IR, QA, etc.)
  - Approach creates a model by training it from a corpus (large)
  - Popular models : n-gram models

Grammar-based Language Models

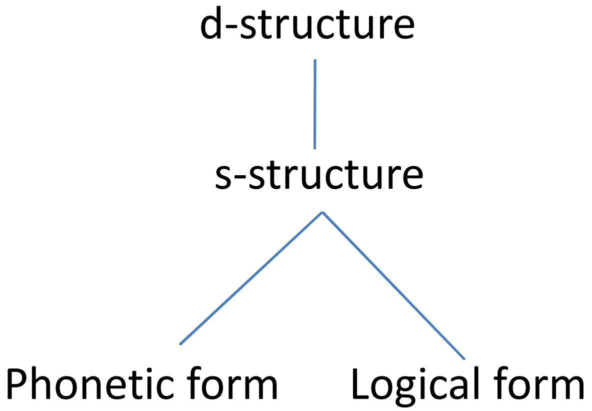
- Grammars
  - Transformational
  - Lexical functional
  - Government and binding
  - Generalized phrase structure
  - Dependency
  - Paninian
  - Tree-adjoining

Grammars

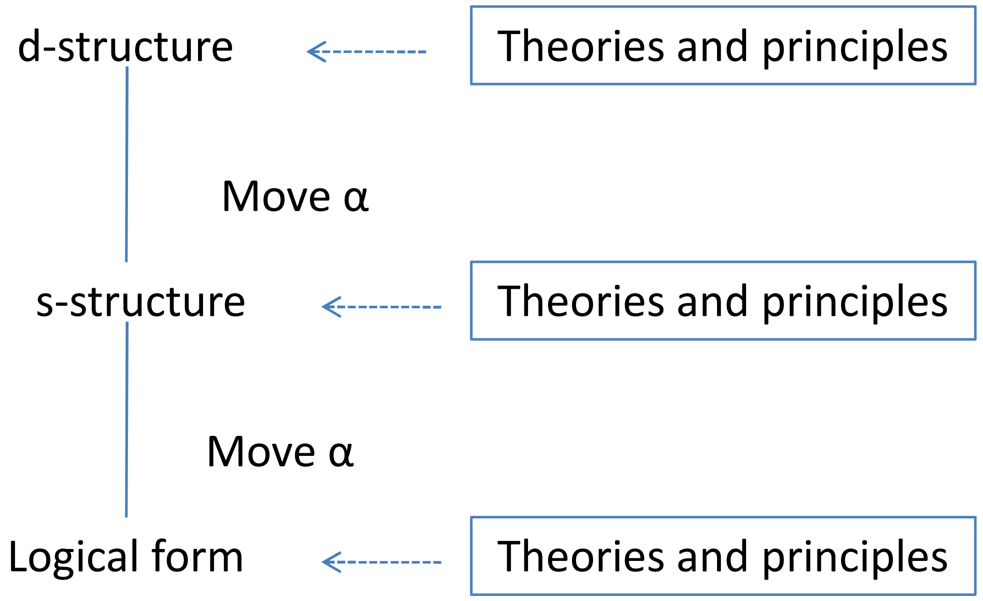
- Generative grammars
  - To generated sentences in a language
  - If we have a complete set of rules that can generate all possible sentences in a language
  - These rules provide a model for the language
- Hierarchical grammar
  - Chomskey (1956) described classes of grammar in a hierarchy
    - Type 0 – unrestricted (superset)
    - Type 1 - context sensitive grammar
    - Type 2 – context free grammar
    - Type 3 - regular

Government and Binding (GB)

- GB grammar identifies 4 levels of syntactic structure
  - Surface level structure : s-level
  - Deep level structure : d-level
  - Phonetic form
  - Logical form



Components of GB



d-structure

Theories and principles

Move α

s-structure

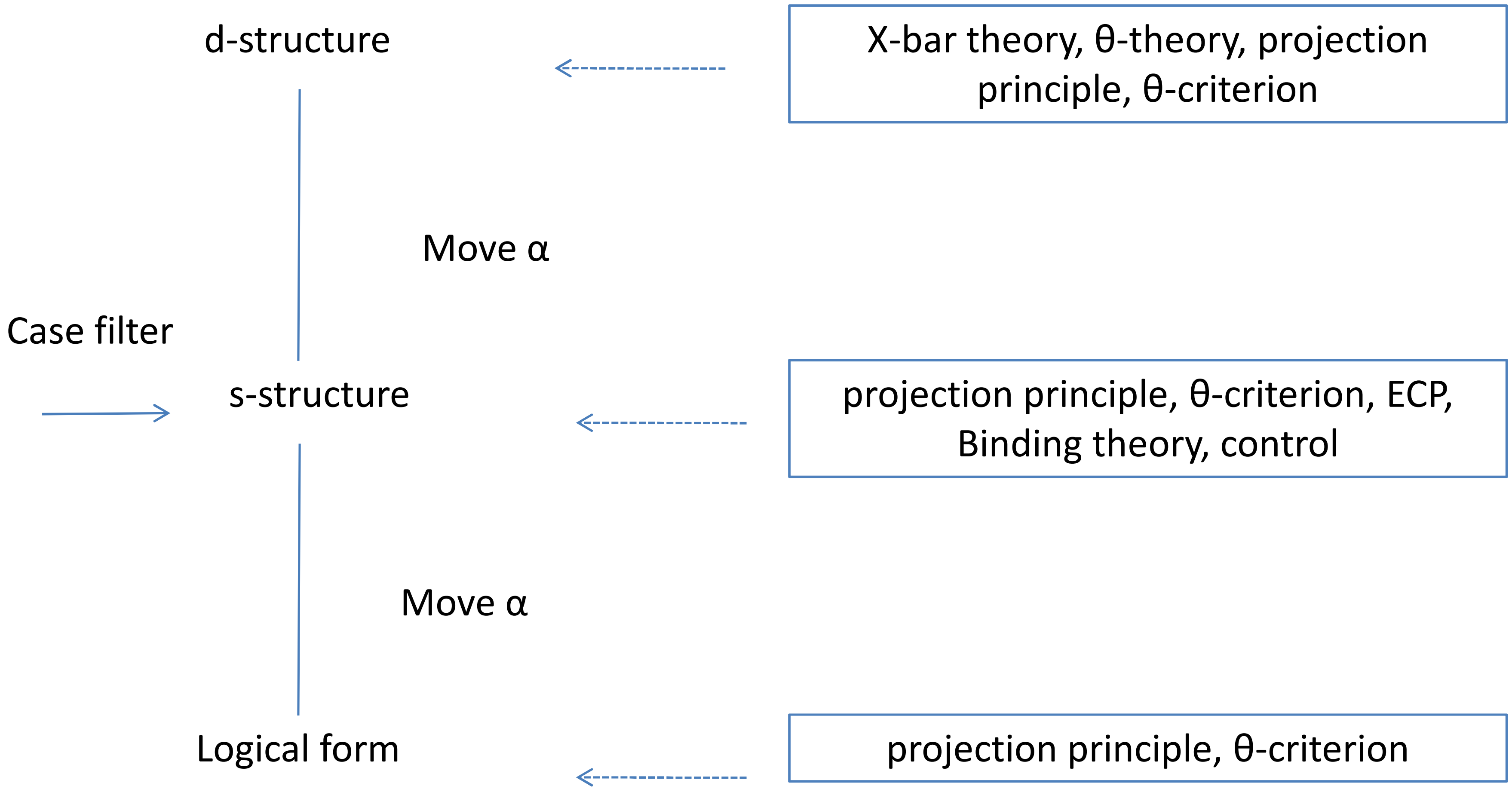
Theories and principles

Move α

Logical form

Theories and principles

Organization of GB



d-structure

X-bar theory, θ-theory, projection principle, θ-criterion

Move α

Case filter

s-structure

projection principle, θ-criterion, ECP, Binding theory, control

Move α

Logical form

projection principle, θ-criterion

X-bar Theory

Useful for appropriate substitution of phrases

Phrases

- The nodes in a syntactic tree above the word level represent **phrases**.
    - phrase = string of words that function as a unit
- Basic phrase types:
    1. Noun Phrases (NP): [intelligent leaders]
    2. Verb Phrases (VP): [shoot terrorists]
    3. Prepositional Phrases (PP): [with rifles]
    4. Adjective Phrases (AP): [more intelligent]

Phrase Phacts

- Every phrase has to have at least one constituent
    - This constituent is called the **head** of the phrase.
- The **head** determines the phrase’s function, behavior and category.
- For example, noun phrases have to consist of at least one noun.

Bob the **book**

a **picture** of Bob a **picture** of the unicorn

that weird **picture** of Bob’s unicorn

In General

- There’s a pattern to how these things work:
- **Noun** phrases (NPs) are headed by **nouns**
    - NP → N
- **Verb** phrases (VPs) are headed by **verbs**
    - VP → V
- **Prepositional** phrases (PPs) are headed by **prepositions**
    - PP → P
- **Adjective** phrases (AdjP) are headed by **adjectives**
    - AP → A
- Basic Phrase Structure Rule: XP → X

More About Phrases

- Beyond the heads, phrases can be expanded with **specifiers** and **complements.**
- **Specifiers** precede the head of the phrase;
  - they qualify or pick out a particular version of the head.
- Examples:

1. this book (Determiner specifying noun)
2. very late (Degree word specifying adjective)
3. often forgets (Qualifier/Adverb specifying verb)
4. almost in (Degree word specifying preposition)

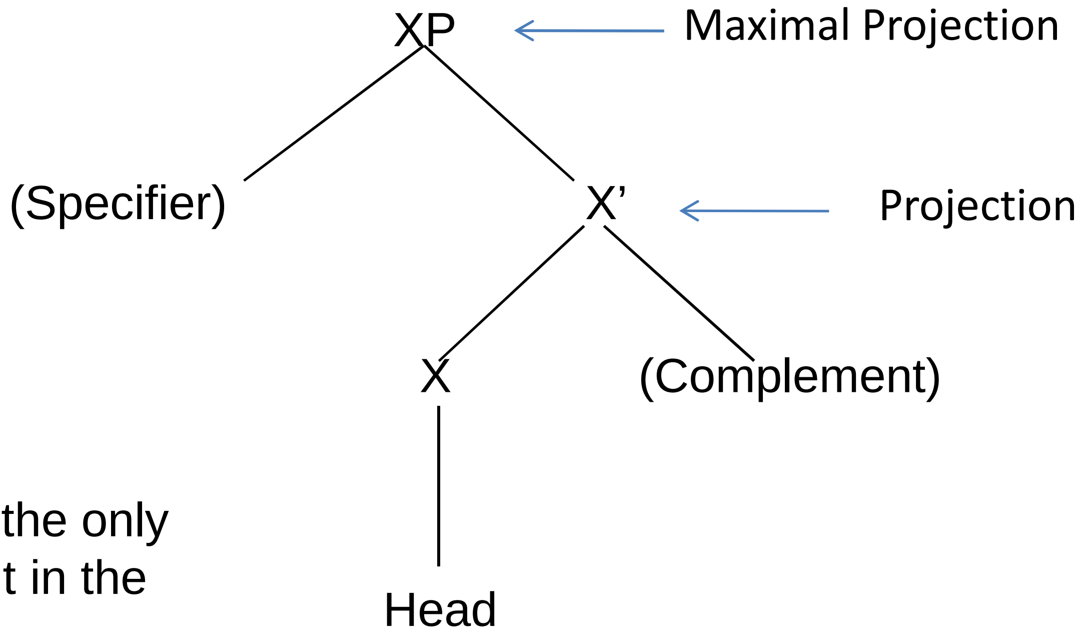
Complements

- **Complements** always follow the head of the phrase…
    - And provide more information about that head.

1. this book about unicorns
    - PP complement of the head of the NP.
2. very late to class
    - PP complement of the head of the AP.
3. often forgets his hat
    - NP complement of the head of the VP.
4. almost in the basket
    - NP complement of the head of the PP.

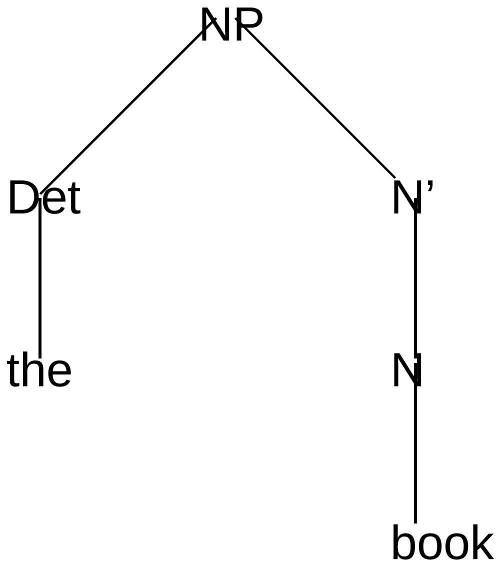
X-Bar Theory

- Together, heads and their complements form a phrasal structure known X' ("X-bar").
- Here's the way phrases (of all kinds) normally break down:

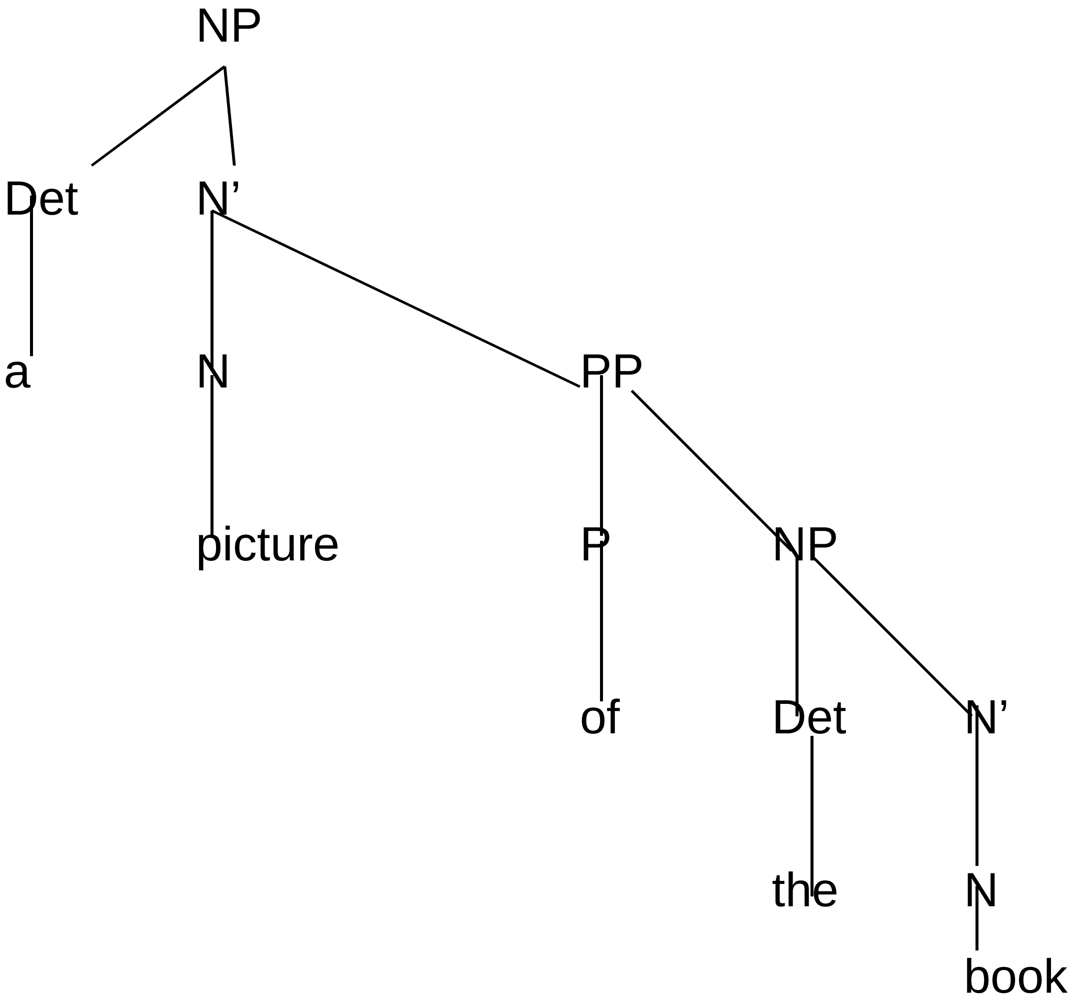


- note: heads are the only <u>obligatory</u> element in the phrase
- optional stuff is in parentheses

Example Tree

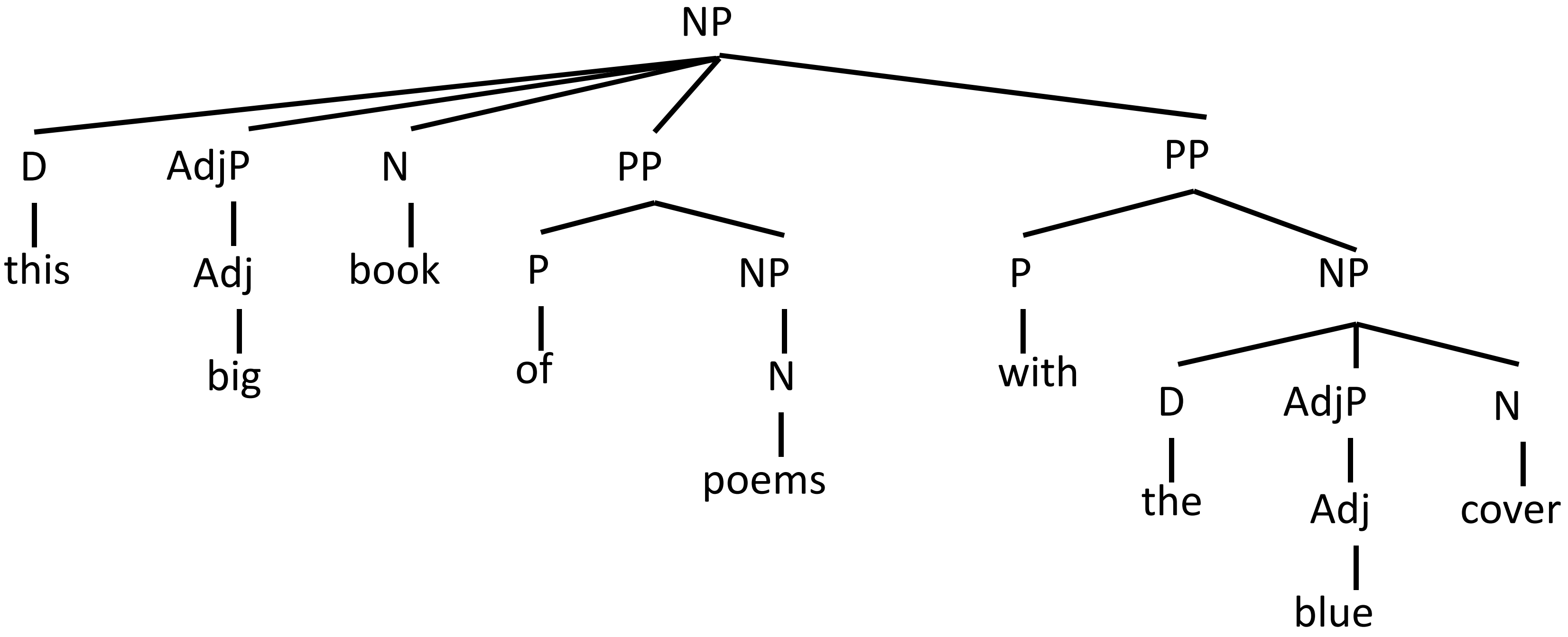


How about: “a picture of the book?”


NP
Det
N’
a
N
PP
picture
P
NP
of
Det
N’
the
N
book

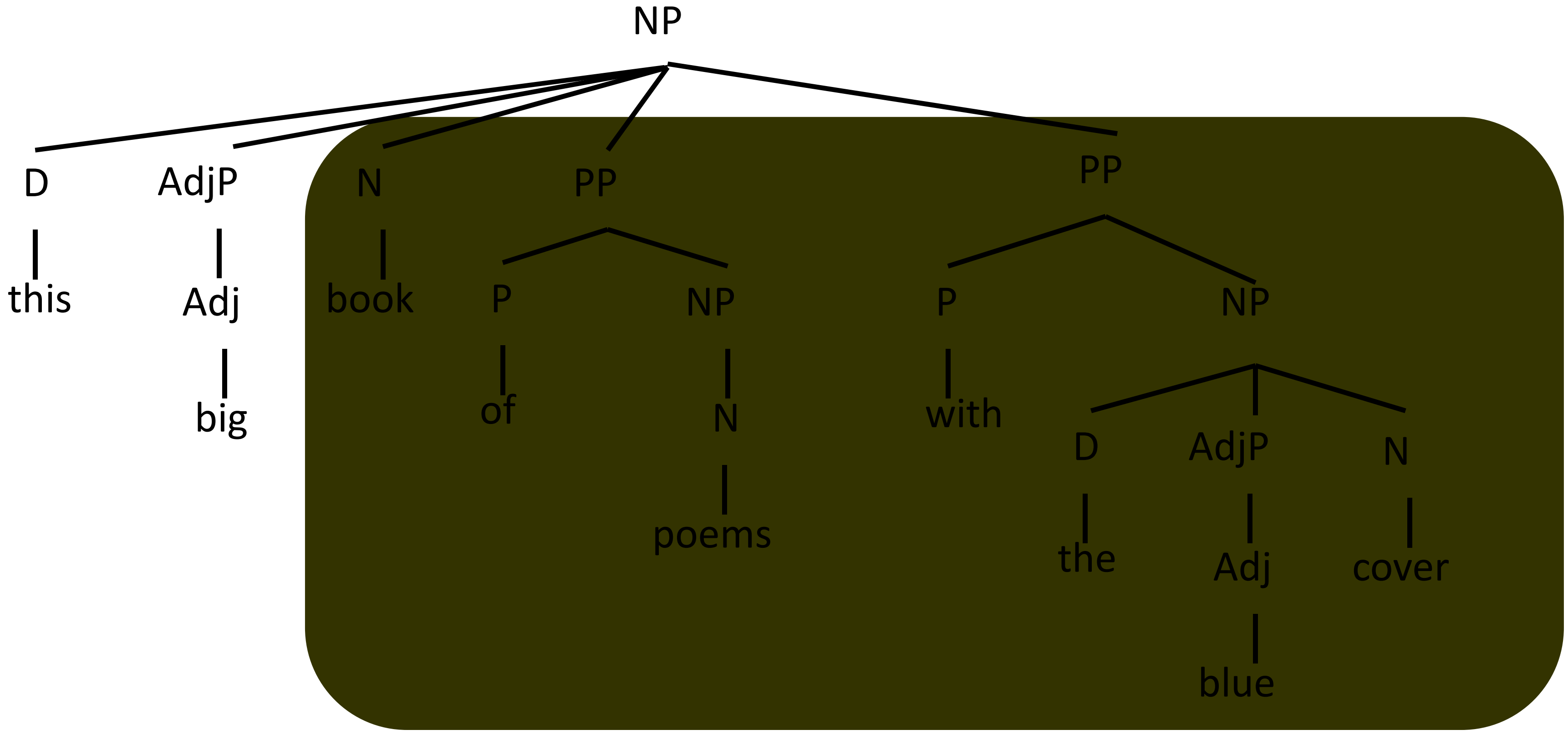
X-bar Theory

- I bought this big book of poems with the blue cover.
- You bought this small one.



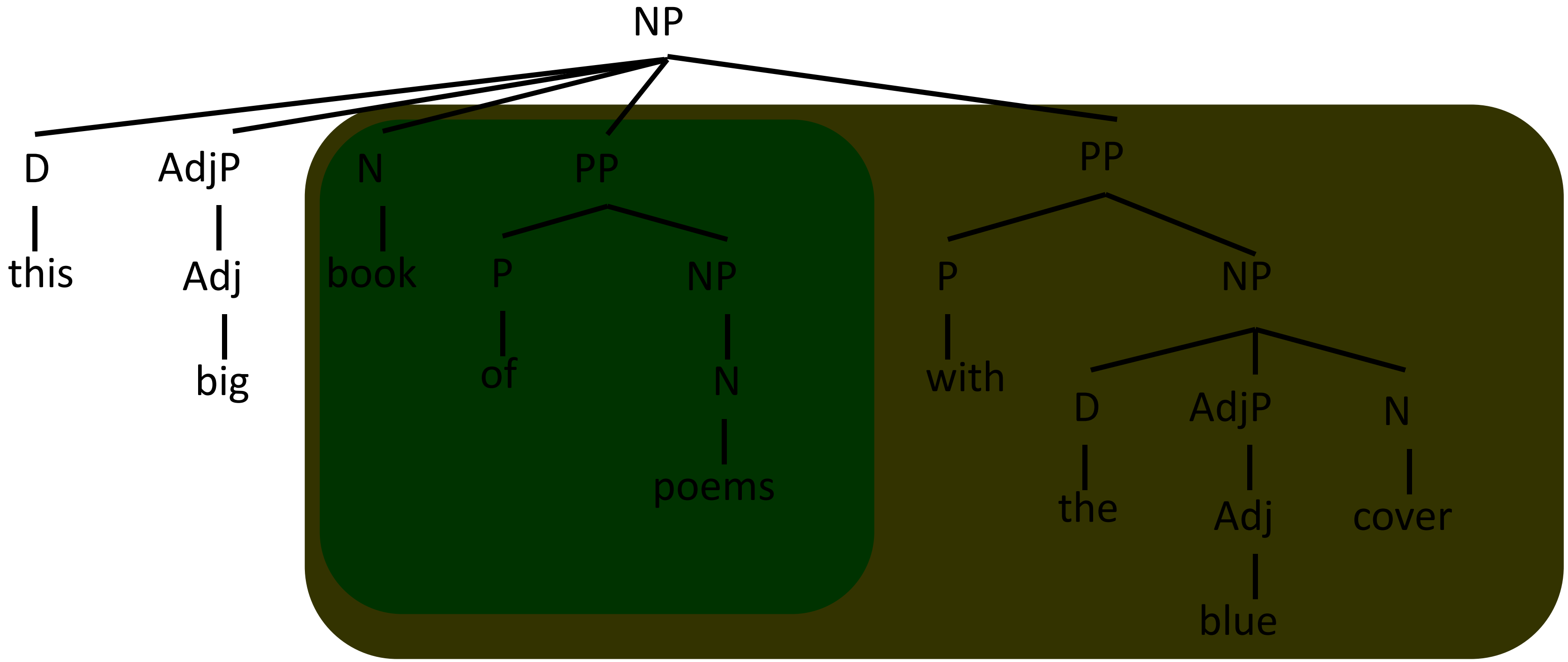
X-bar Theory - NP

- We can substitute *one* for *book of poems with the blue cover*, which should mean *book of poems with the blue cover* is a constituent, but it isn't in our structure.



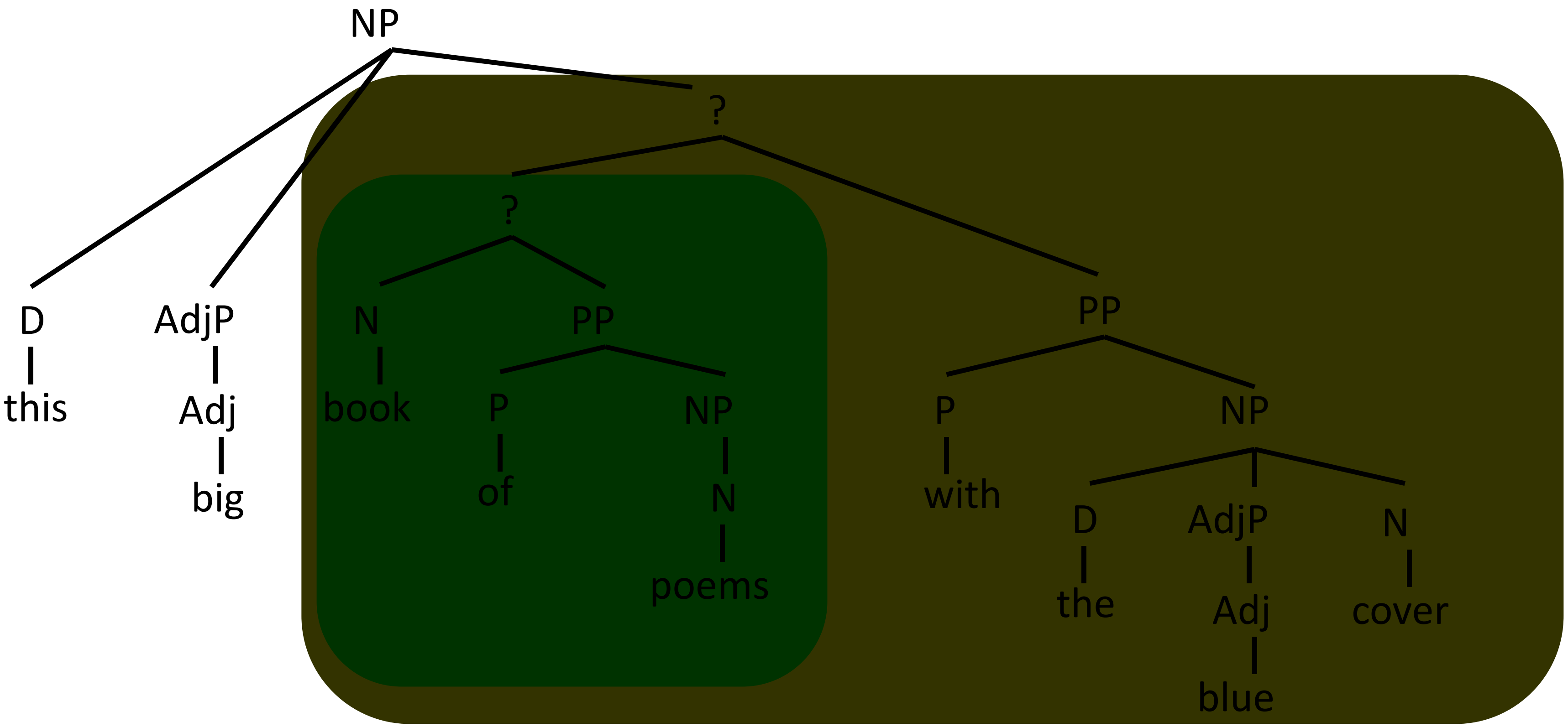
X-bar Theory - NP

- I bought this small one with the red cover.
- We can also substitute *one* in for *book of poems* alone, which should thus also be a constituent.



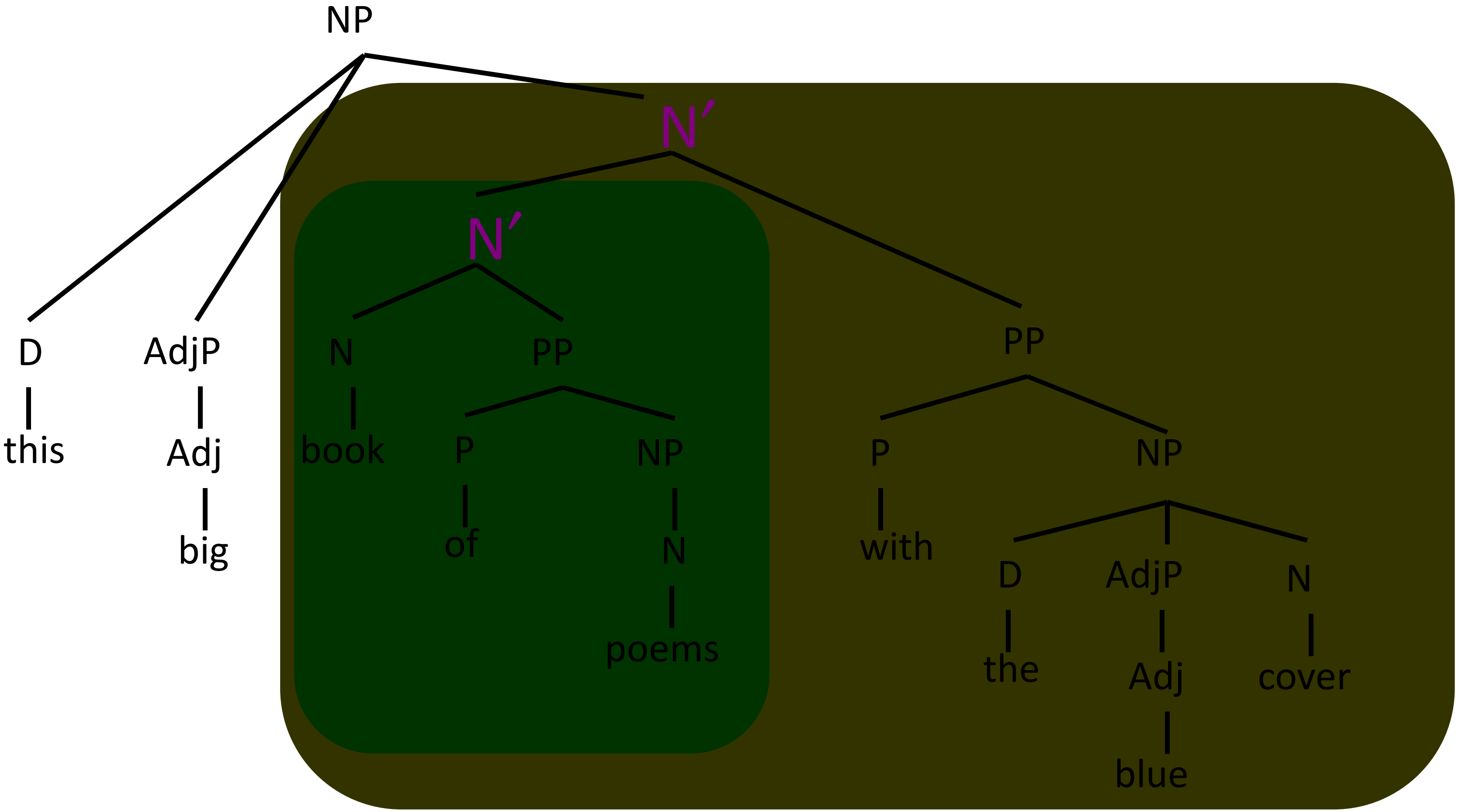
X-bar Theory - NP

This suggests a more deeply embedded structure:



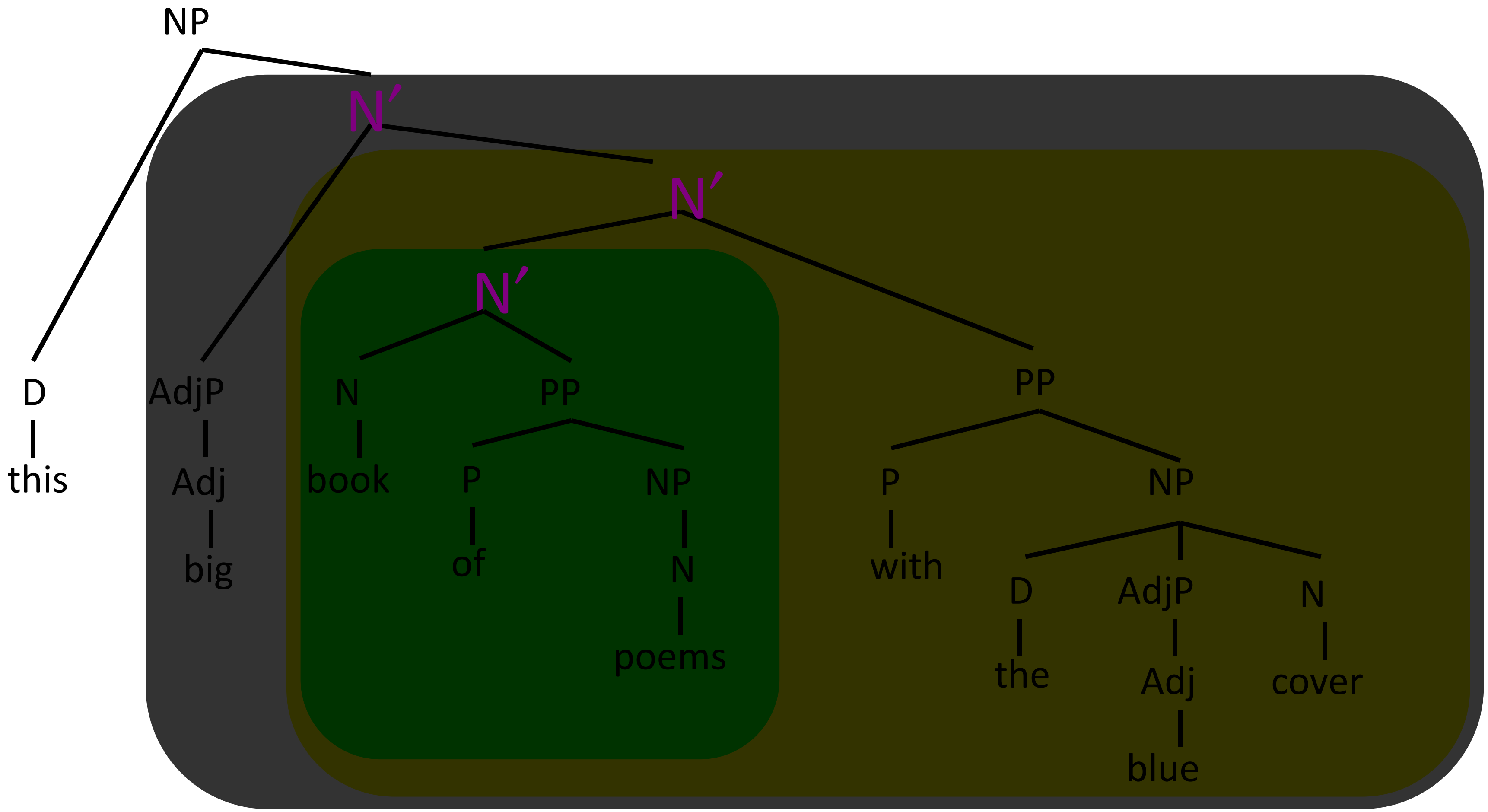
X-bar Theory - NP

- We'll call these "intermediate" nodes of NP N′ (N-bar).
- Notice that you can also say *I bought this one*.



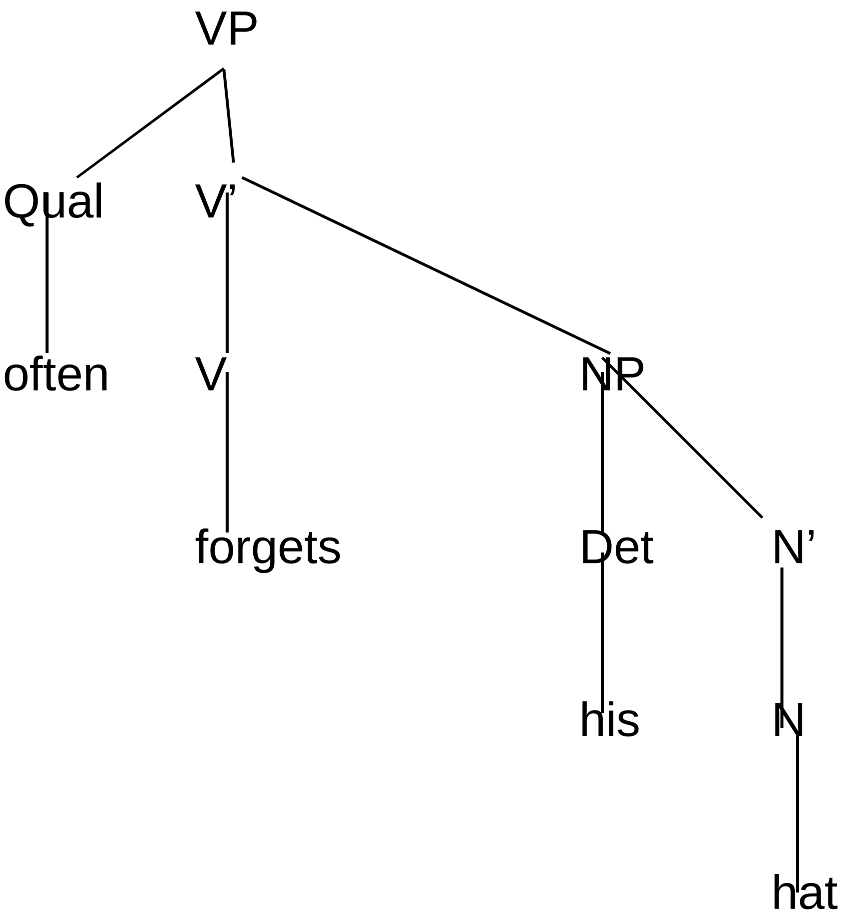
X-bar Theory - NP

So, our final NP looks like this:

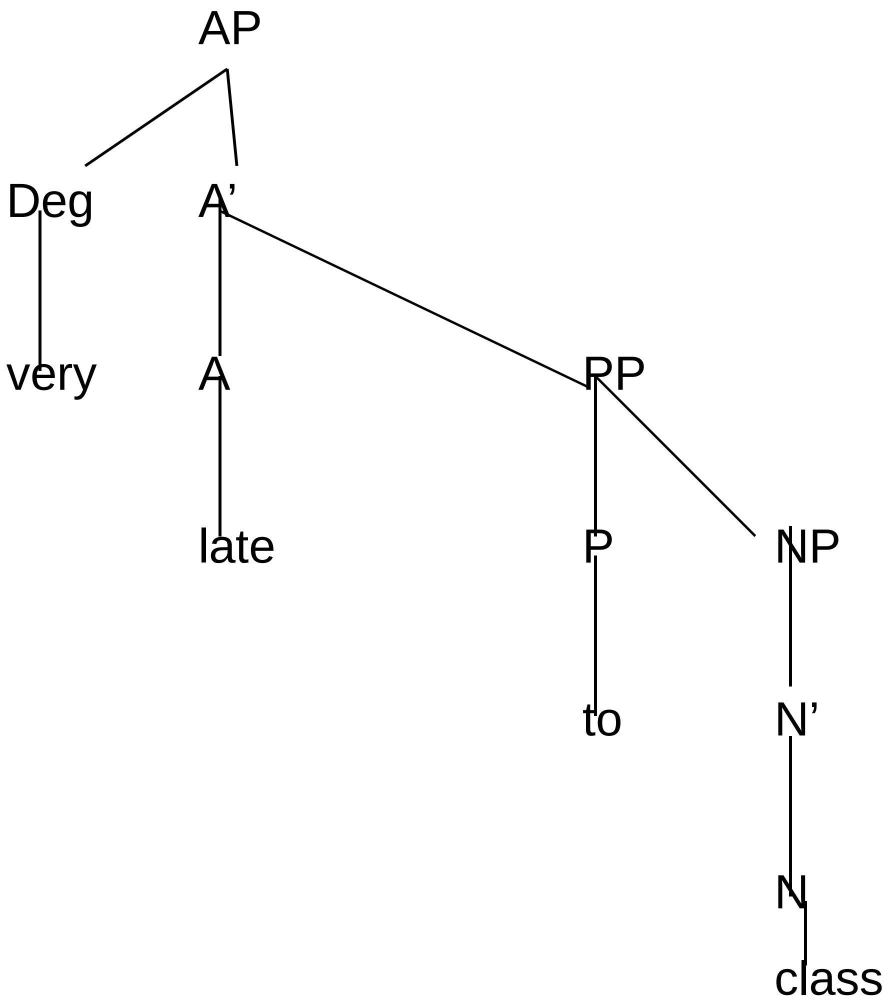


More Example Trees

- Let’s draw trees for the following phrases:
- VP: often forgets his hat
- PP: almost in the basket
- AP: very late to class



A VP Example



An AP Example

Check This Out

1. A phrase structure rule for NPs looks like:
   - NP → Det N’
2. And a PP can be a complement of a head noun:
   - N’ → N PP
3. And an NP can be a complement of a prepositional phrase:
   - PP → (Deg) P’
   - P’ → P NP

- Where can this combination of rules take us?

- There is a possibility for infinite recursion.
- NP → Det N PP
- NP → Det N P NP
- NP → Det N P Det N PP
- NP → Det N P Det N P NP
- NP → Det N P Det N P Det N PP, etc.
- Example: the book from the library in the city near the airport beside the apartment complex with the playground of the children from the school behind the train tracks...
- The fact that our grammar can generate phrases like this is why we need to know **patterns of patterns**.

Sub Categorization

- The child relied on the parent.
- The child relied the parent.
- The child relied.

Sub Categorization Restrictions

- Sub Categorization Frames
- Sub Categorization Rules

Subcategorization Frames

- Specify the categorial class of the lexical item
- Specify the environment
- Examples
  - –kick: [V; _ NP]
  - –cry: [V; _]
  - –rely: [V; _PP]
  - –put: [V; _ NP PP]
  - –think: : [V; _ S`]

Subcategorization Frames

- The information in the subcategorization frame implies that *kick* can only be inserted under a V node in a VP structure in which V has an NP sister.

- The subcategorization information is associated with the individual lexical items in their lexical entries.

Subcategorization Rules

- These rules make a specific lexical item sensitive to the subcategorization properties of the lexical item.
- Selection of a frame depends on the subcategorization properties of the verb.

Subcategorization Rule:

$$\mathrm{V} \longrightarrow y \;/ \left\{ \begin{array}{l} \_\mathrm{NP}] \\ \_\ ] \\ \_\mathrm{PP}] \\ \_\mathrm{NP}\ \mathrm{PP}] \\ \_\mathrm{S}`] \end{array} \right\}$$

Subcategorization Rules

- The child relied on the parent

1. S → NP VP
2. VP → V (NP) (PP) (S`)…
3. NP → Det N
4. V → rely / _PP]
5. P → on / _NP]
6. Det → the
7. N → child, parent

Context Free and Context Sensitive Rules

- Rule (4) ensures the non-generation of
  - *The child relied.
  - *The child relied the parent.
- Rule (4) and (5) are context sensitive rules
- Rule (1-3) are context free rules

Projection Principle

- A basic notion in GB
- The principle states that representations at all syntactic levels are projections from the lexicon
- Thus lexical properties of categorical structure (sub categorization) must be observed at each level
- This ensure correct movement and well-formed structure

Theta Theory (θ-Theory) – The Theory of Thematic Relations

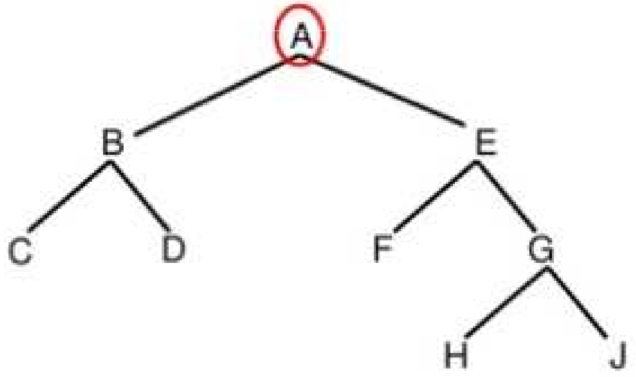
- There are certain thematic roles from which a head can select
- Ex: verb ‘eat’ can take arguments with θ-roles (agent, theme)
  - Mukesh ate food
    - Mukesh – agent
    - Food – theme or patient

- Theta-criterion states that
  - Each argument bears one and only one θ-role and each θ-role is assigned to one and only one argument
- Thus, each argument will have a unique θ-role and cannot moved to a position where it may acquire another θ-role
- θ –roles are assigned only at d-level

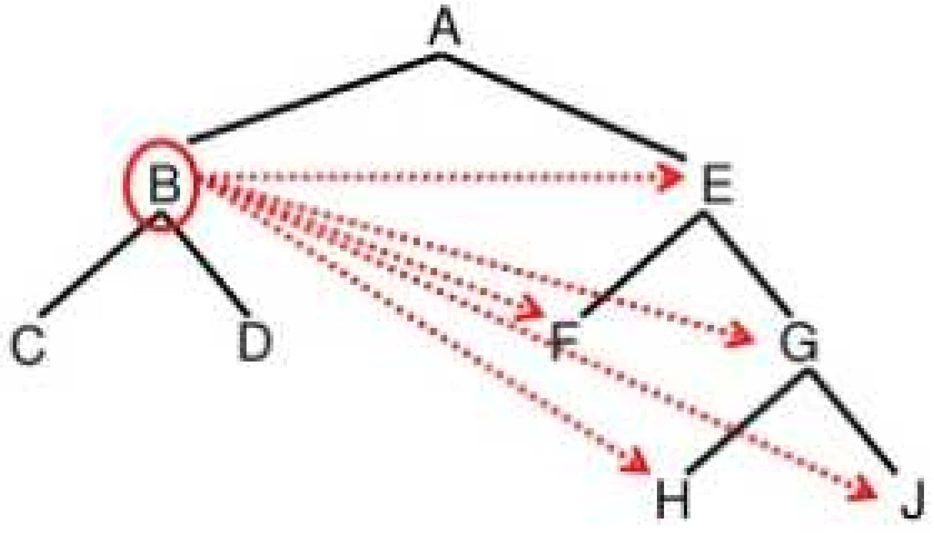
C-Command and Governments

- Governments is a special case of C-command
- C-command defines the scope of maximal projection

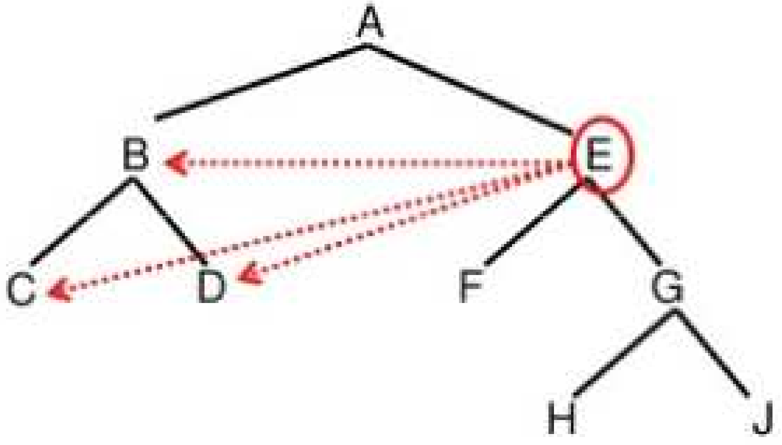
  if there are two structures α and β related in such a way that
  ‘every maximal projection dominating α dominates β’, then α c-commands β



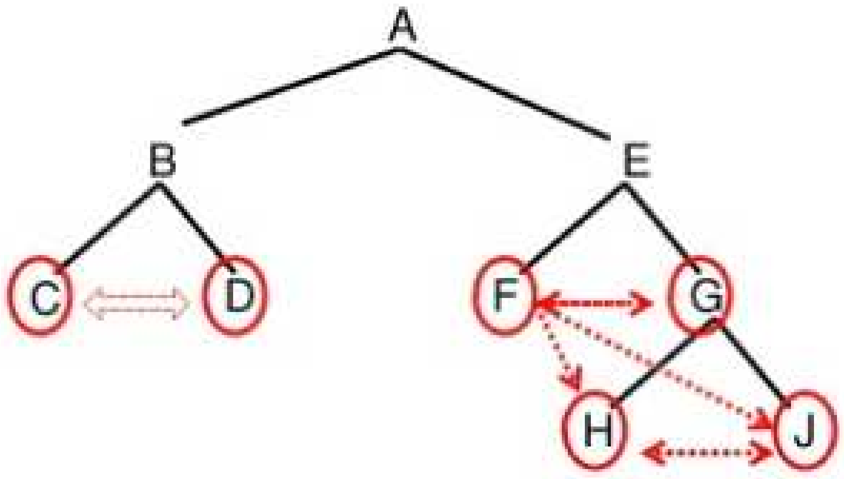
❖Mother or Root A dominates everyone.



❖B commands E, F, G, H and J



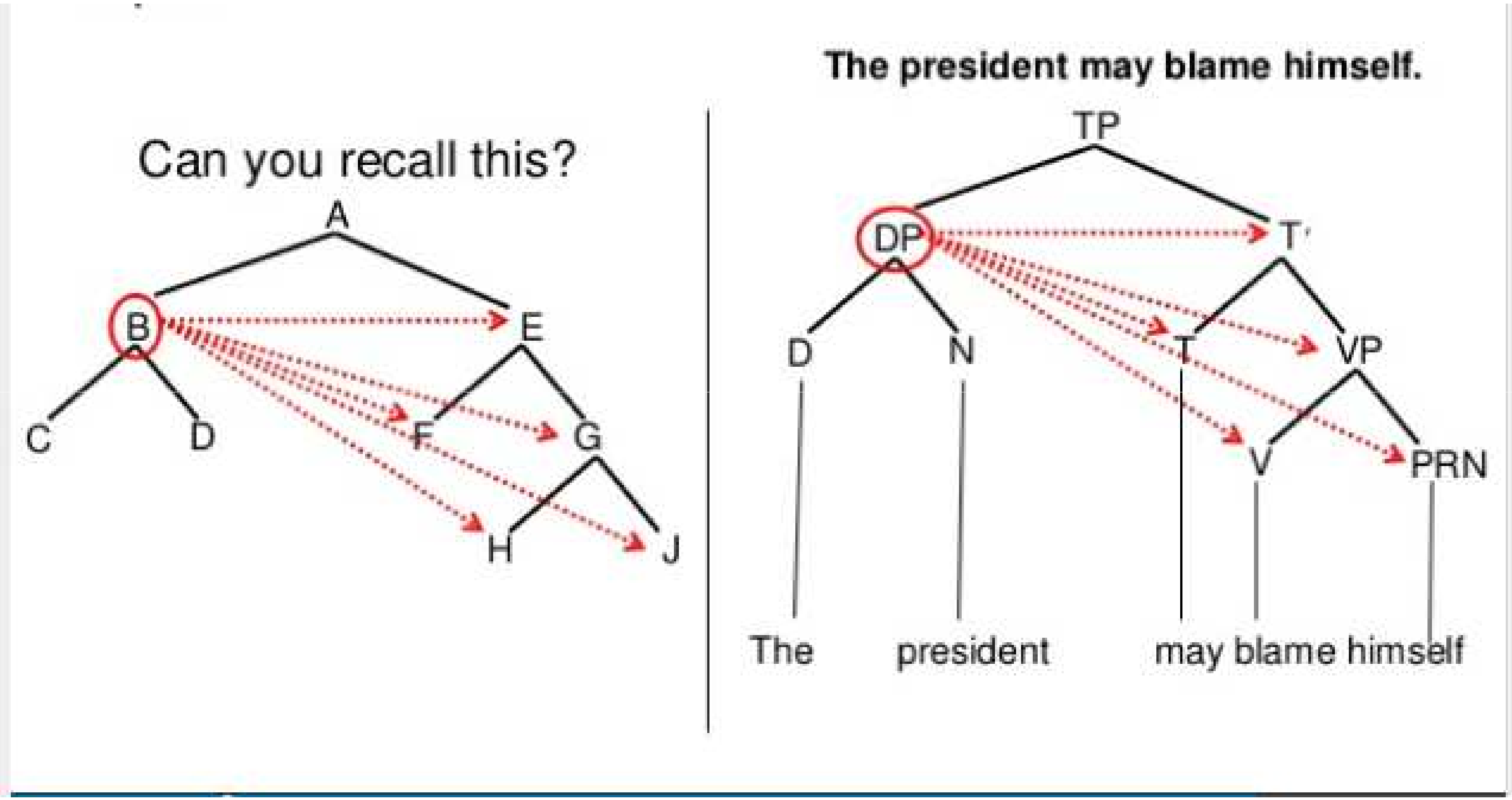
❖E commands B, C and D



- ❖C and D command each other
- ❖F commands G, H and J
- ❖G commands F

From the above examples, C-Command can be defined as follows:

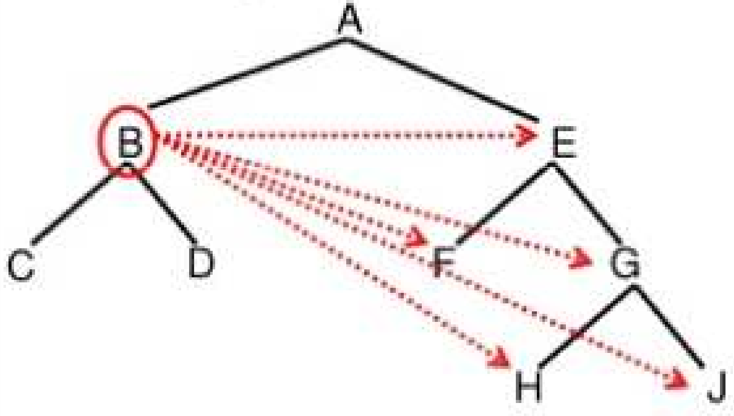
- ▪A constituent X c-commands its sister Y and any constituent Z which is contained within Y.



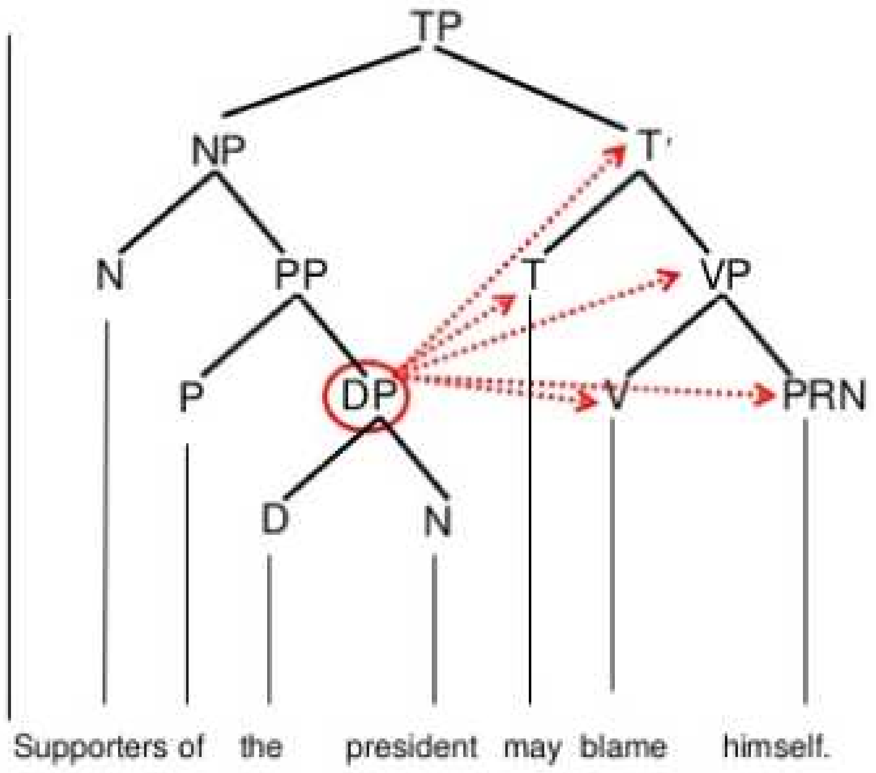
Can you recall this?

The president may blame himself.

Can you recall this?



Supporters of the president may blame himself.

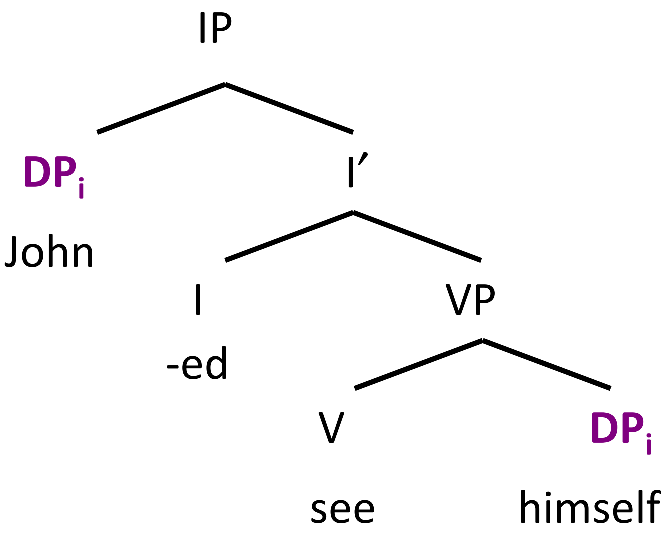


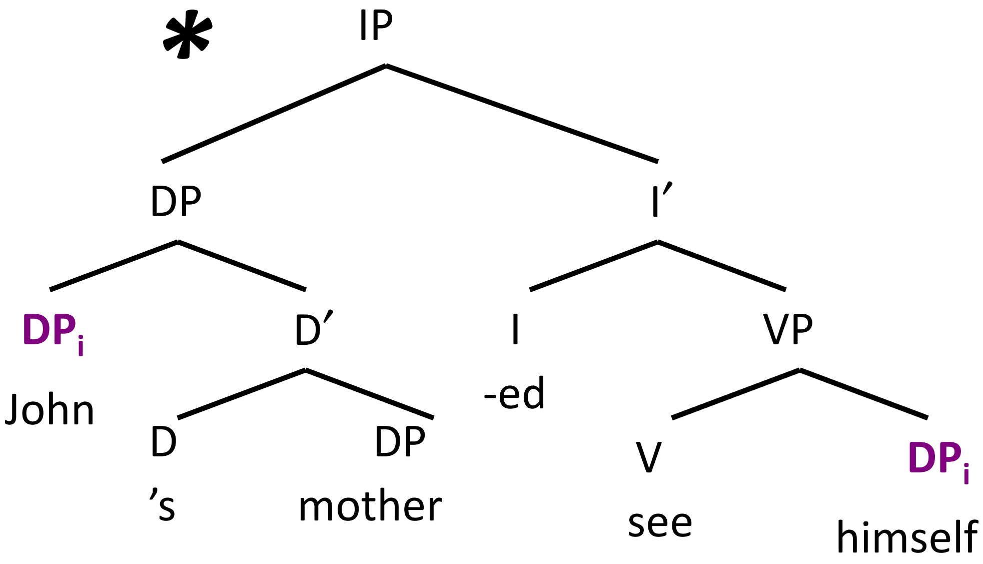
Government

- α governs β iff: α c-commands β

Binding

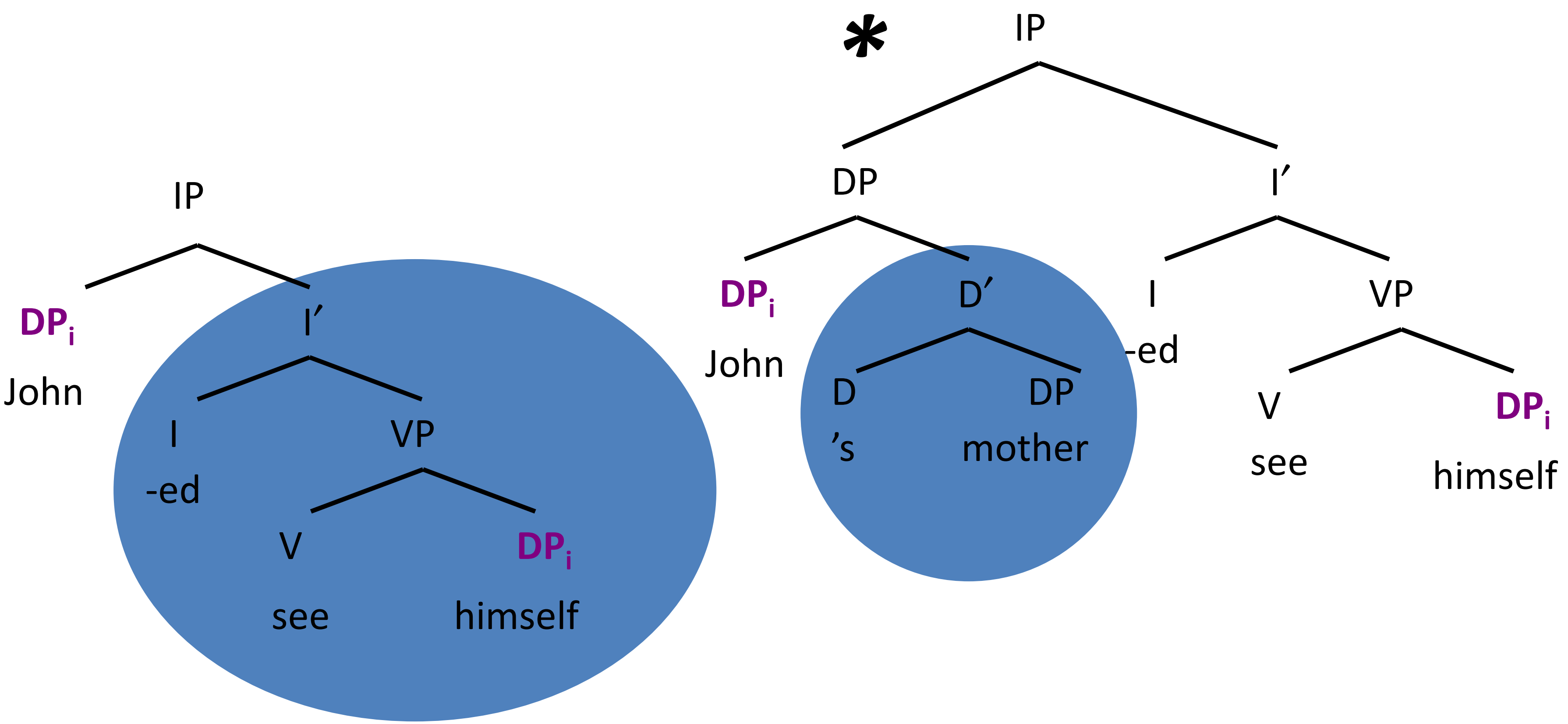
- What is the difference between the relationship between *John* and *himself* in the first case and in the second case?





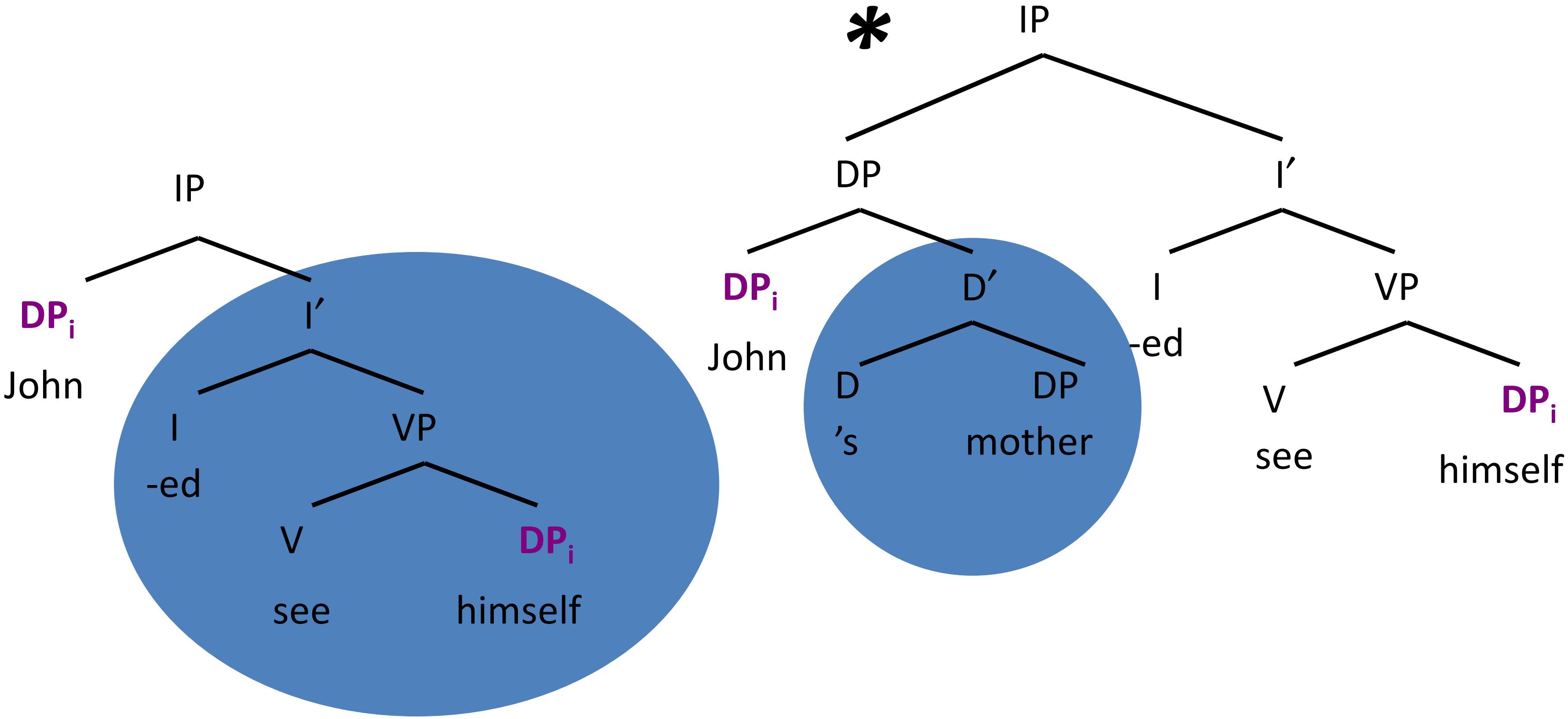
Binding

- In the first case, the DP *John* c-commands the DP *himself*. But not in the second case.



Binding

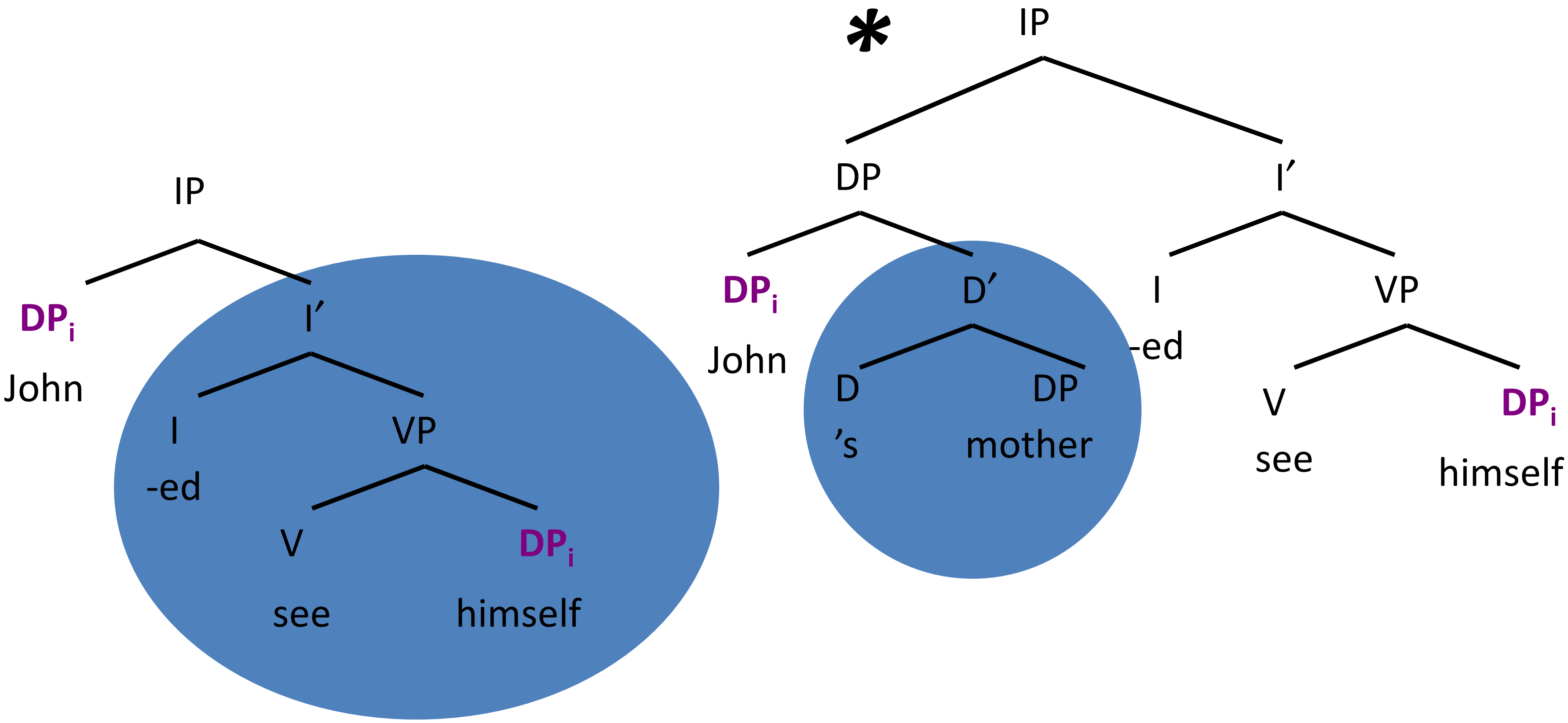
- When one DP c-commands and is coindexed with another DP, the first is said to **bind** the other.



Binding

- Definition: A **binds** B iff
  - A c-commands B
  - A is coindexed with B

"if and only if"


IP
DP$_i$
John
I′
I
-ed
VP
V
see
DP$_i$
himself
*
IP
DP
DP$_i$
John
D′
D
's
DP
mother
I′
I
-ed
VP
V
see
DP$_i$
himself