

Syntax – LIN331
Susana Bejar

Introducing minimalism

- We have been discussing **functional structure** and some enrichments:
 - little v: motivated by argument structure and verb position (Jan 17)
 - Decomposition IP into TP and AgrP: motivated by verb positions (understood as verb movement) (Jan 24)
- We'll see in the coming weeks that both of these enrichments were important precursors to the emergence^{of} of the minimalist framework (Chomsky 1993, 1995 and subsequent work).
- This week we introduce the minimalist framework, which we will continue to develop throughout the term.

The minimalist framework

The minimalist framework is an instantiation of the theory of Principles and Parameter (P&P).

- Methodological pivot away from rich syntactic modeling
- Starting point is the successes of **Government and Binding** theory (GB)
- Goal is to supercede GB by reassessing its successes in view of the **strong minimalist thesis**
 - 1) **Strong minimalist thesis (SMT):** UG reduces to the simplest computational principles which operate in accord with **interface conditions** and conditions of **computational efficiency**.

We'll see what is meant by **interface conditions** shortly. **Computational efficiency** we'll leave for another class.

Why the SMT?

- Plato's problem

About 6,100,000 results (0.61 seconds)

In the field of linguistics, Plato's Problem is **the problem of finding an explanation for how a child acquires language** though the child does not receive explicit instruction and the primary linguistic data a child does receive is limited.



https://en.wikipedia.org/wiki/Plato's_Problem :: [Plato's Problem - Wikipedia](#)

Plato's Problem <

Plato's Problem is the term given by Noam Chomsky to "the problem of explaining how we can know so much" given our limited experience. [Wikipedia](#)

[Feedback](#)

- Occam's razor

About 3,000,000 results (0.45 seconds)

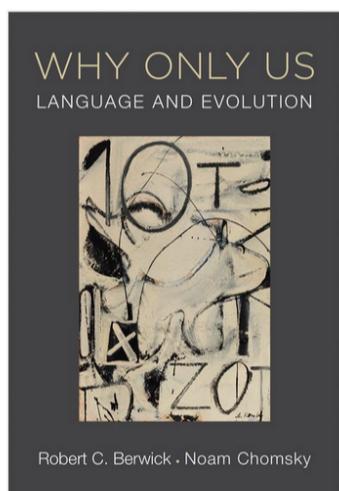
Occam's razor is a **principle of theory construction or evaluation** according to which, other things equal, explanations that posit fewer entities, or fewer kinds of entities, are to be preferred to explanations that posit more.



https://www.britannica.com/.../Philosophical_Issues :: [Occam's razor | Origin, Examples, & Facts | Britannica](#)

[Feedback](#)

- Darwin's Problem



Why Only Us: Language and Evolution

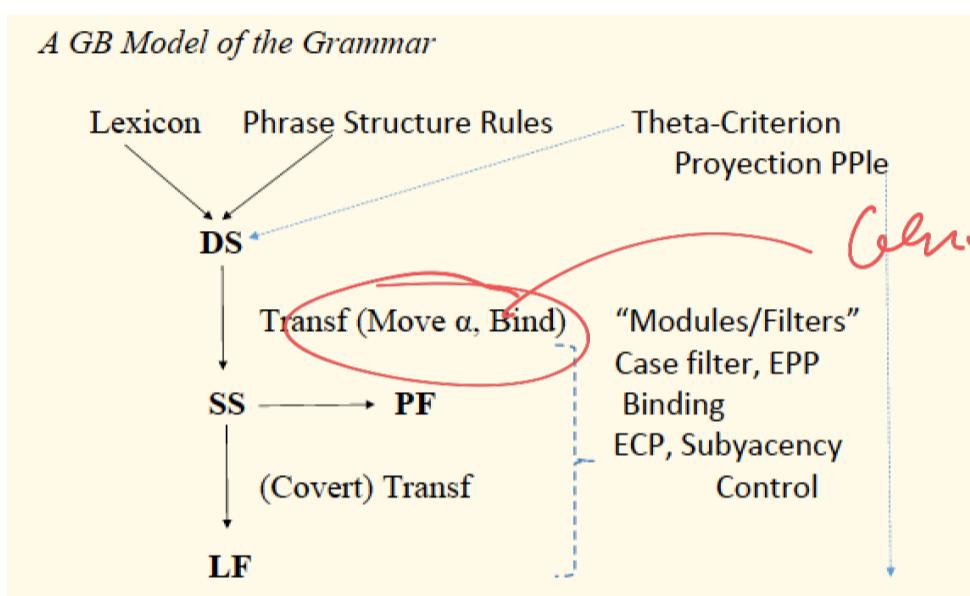
By Robert C. Berwick, Noam Chomsky

The MIT Press

DOI: <https://doi-org.myaccess.library.utoronto.ca/10.7551/mitpress/10684.001.0001>

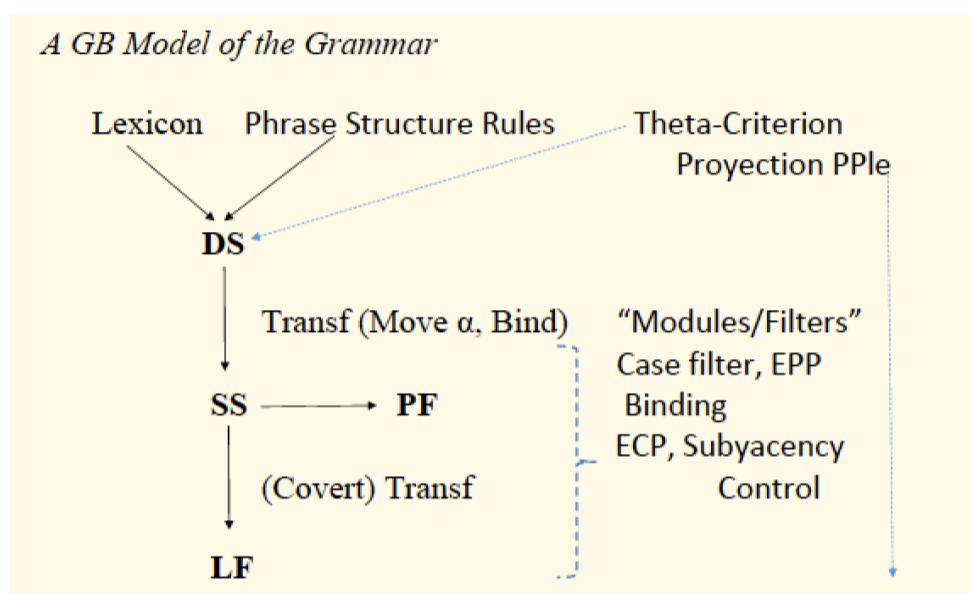
ISBN electronic: 9780262333351

GB Architecture

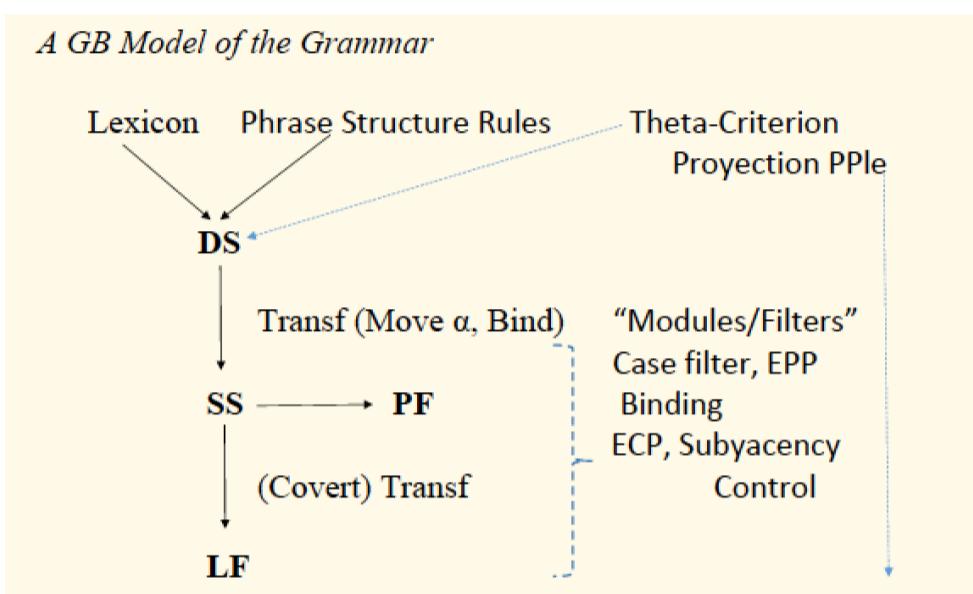


- D-structure: DS is the output of phrase-structure rules (and lexical insertion) and the input to transformations.

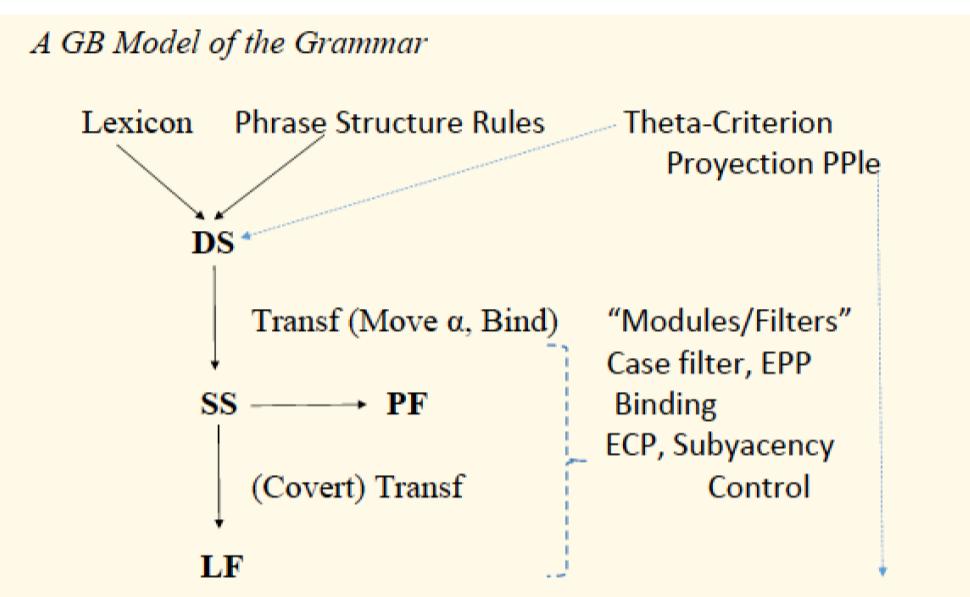
- D-structure : DS is the output of phrase-structure rules (and lexical insertion) and the input to transformations.
 - locus of recursivity (via phrase structure rules)
 - serves as common input to the computations that will lead to PF and LF, thereby ensuring the form-meaning compatibility
 - level where thematic relations are established; pure GF-theta is represented, i.e. one-to-one correspondence between grammatical function and theta-role.
 - external argument external to VP
 - internal argument(s) internal to VP



- S-structure: S-structure is the output of transformations on D-structure.
 - where many filters apply to get rid of unwanted derivations: Case, some aspects of Binding, some aspects of trace-licensing (ECP), subjacency, etc.
 - reference point for overt and covert movements, accounting for some cross-linguistic variation. (e.g. English vs. Chinese with overt vs. covert wh-movement, French vs. English with overt vs. covert V-movement, etc.)



- PF_{SEP} and LF: important interface levels in GB with different operations applying at each level.
 - PF = "phonetic form"
 - LF = "logical form"
- Projection Principle: Lexical information (such as theta roles) must be preserved at all levels of syntactic derivation. Thus the need for traces to preserve the thematic and structural information encoded at DS (plus the stipulation, known as EPP, that all clauses must have subjects)



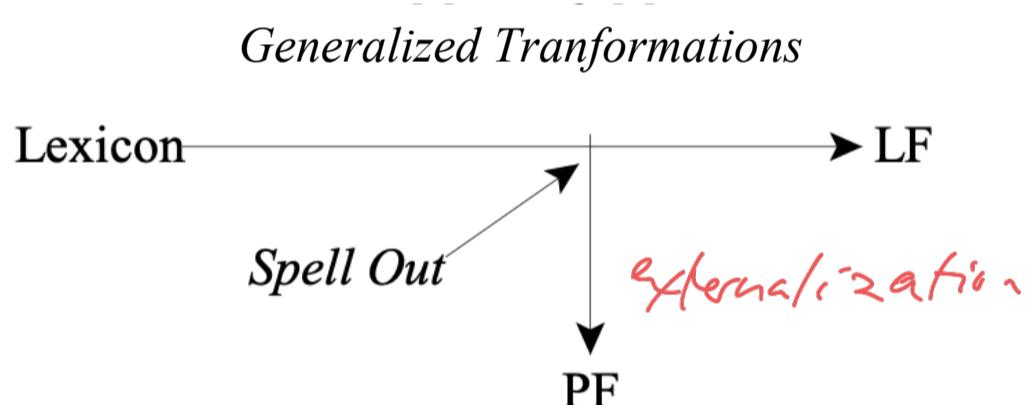
- Transformational Component: (overgenerates)
 - Bind: free indexing of DPs
 - Move: free movement of anything anywhere
- Modules: (distributed across the architecture; restrict overgeneration)

- Modules: (distributed across the architecture; restrict overgeneration)
 - Case Theory: restricts distribution of arguments
 - Theta Theory: restricts thematic relations
 - Binding Theory: restricts binding configurations
 - Empty Category Principle (ECP): restricts distribution of traces
 - Subjacency: restricts movement patterns
 - X-bar Theory: restricts phrase structure
 - Control Theory: restricts distribution of PRO
- Government: The grammatical relation that is invoked by the different modules, **uniting them.**
 - Time allowing, we may talk a little bit about government towards the end of today's class; more next week for sure.

Minimalist architecture

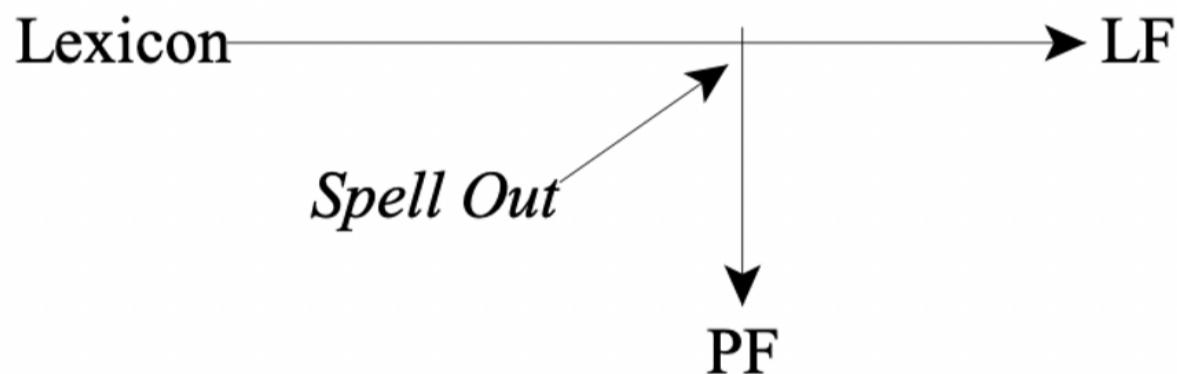
- The minimalist program aims for the elimination of architecture that is not motivated by the strong minimalist thesis (SMT), while seeking to preserving the insights and results that it was used to describe.

2) Minimalist architecture (to be revised)



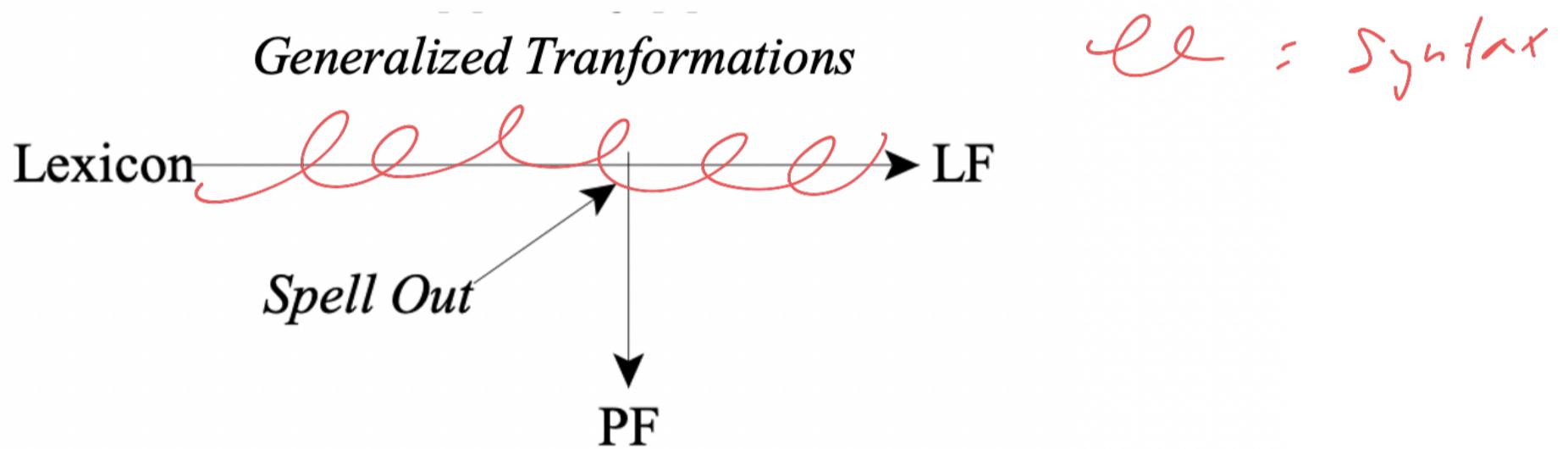
Eliminating levels of representation

Generalized Transformations



- The only levels of representation motivated by the SMT are the **interfaces** to external cognitive systems (cf. Darwin's problem):
 - Sensorimotor systems: PF
 - Conceptual systems: LF
- DS and SS are not conceptually motivated and so are eliminated

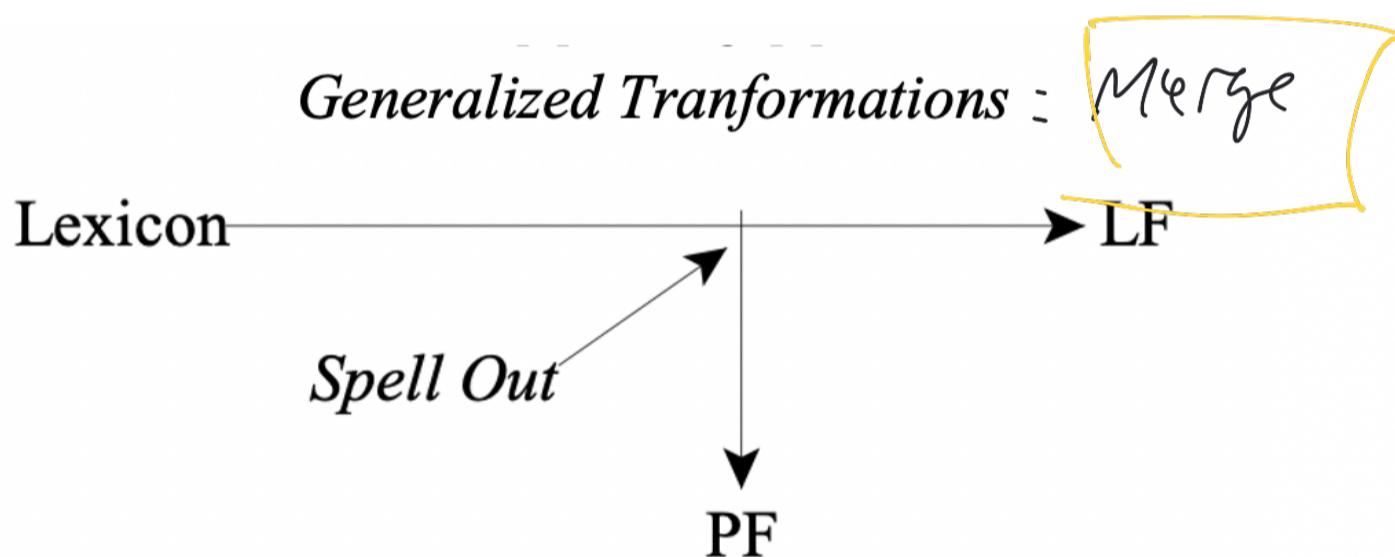
Interfaces and interface conditions



- PF (sensorimotor) and LF (conceptual) interfaces set boundary **conditions** on the nature of syntactic computation. The output of syntax must be **legible** at the PF and LF interfaces.
 - SMT postulates that syntax is an optimal system for satisfying **interface conditions**

Eliminating Phrase Structure Rules

~~Generalized Transformations: Merge, ...~~

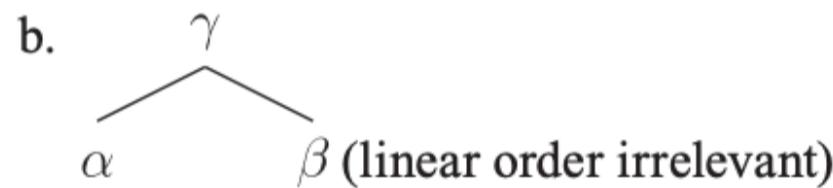


- Phrase Structure Rules (i.e. X-bar) are eliminated (cf. Occam's Razor). Their effects are subsumed under a new **generalized transformation**, the operation **Merge**.
 - Structure building and recursion moved from **Base** (in GB architecture) to generalized transformations!

Merge

- 1) **Merge**: takes two syntactic objects and forms a single object from them.

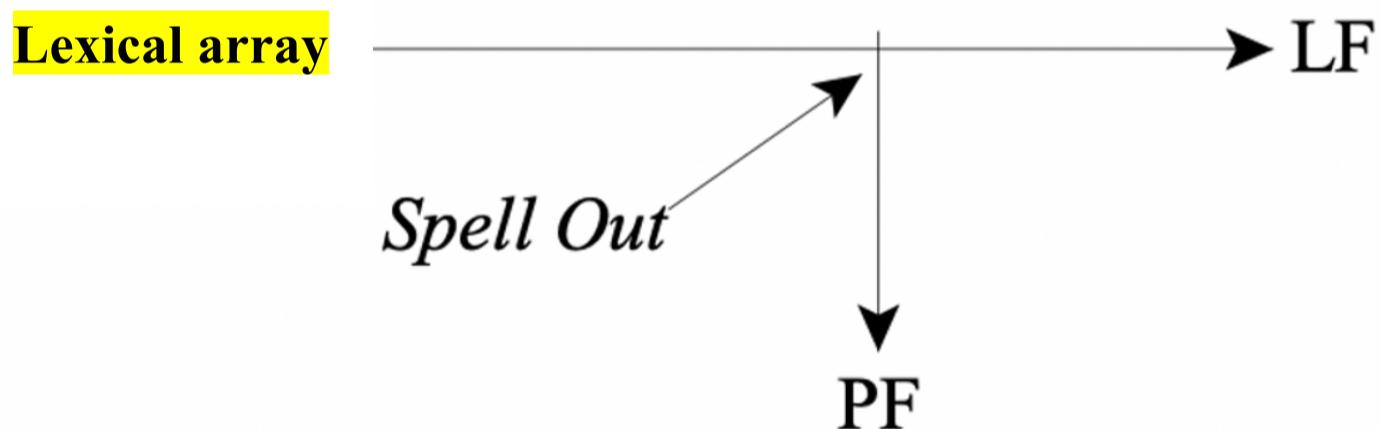
$\text{Merge}(\alpha, \beta) =$ a. $\{\gamma, \{\alpha, \beta\}\}$, where the label γ = the label of α or β .



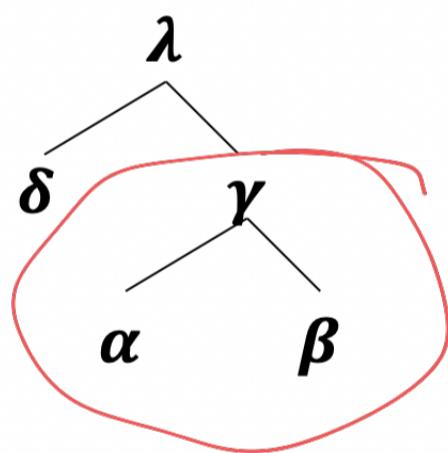
(Fukui & Narita 2014:

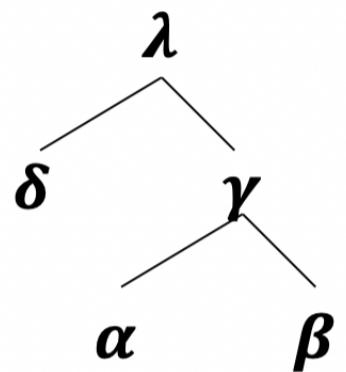
- Merge combines **lexical items** (LIs) selected from the **lexicon** and deposited in a **lexical array** (sometimes also called a **numeration**).
 - In other words, the starting point for a derivation is not the entire Lexicon but rather a subset of it.

Generalized Transformations: Merge, ...



- **Lexical items** (LIs) consist of three kinds of feature bundles:
 - Phonological features
 - Semantic features
 - Formal Features (FFs) (these are exclusively syntactic)
- Merge not only combines LIs in the Lexical Array, it also combines syntactic objects that are the output of previous iterations of Merge itself.
 - In other words, Merge is **recursive**.





- Iterations of Merge create increasingly complex syntactic objects.
- Minimalist claim: all of the core properties of phrase structure that were captured by X-bar schema in GB can be reconstrued in terms of Merge!

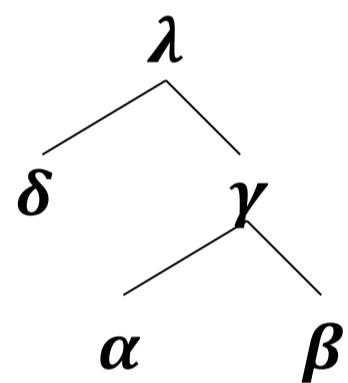
- X-bar schema captured the following key properties of phrase structure:
 - Sentences are composed of phrases organized in a **hierarchical** fashion.
 - Phrases are **recursive**
 - Phrases are **binary branching** structures
 - Phrases are **endocentric** objects with a position in the immediate projection of the head, which we call the **complement**, and a position outside the immediate projection of the head, which we call the **specifier**.
 - Phrases are composed of **heads** (X^0), **intermediate projections** (X') and **maximal projections** (XP).

Specifier $XP \rightarrow (WP) X'$

Adjuad $X' \rightarrow X' (Yp)$

Complement $X' \rightarrow X (Zp)$

- Properties (a)-(c) are captured by the fact that Merge is defined **recursively**: The output of one iteration of Merge can serve as the input to another iteration, giving recursive, binary branching hierarchical structures.



Merge is
irreducible
Binary
(always puts 2 forms
together)

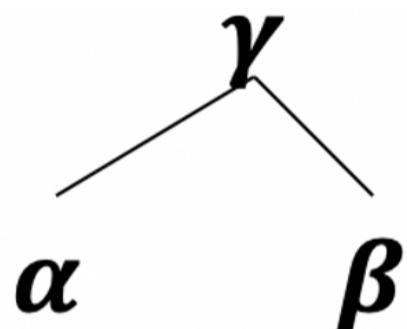
- To derive (d) and (e), i.e. the properties of **endocentricity**, **complement** and **specifier** relations to a head, and the taxonomy of **heads**, **intermediate** and **maximal projections**, we need to take a closer look at the phrase structure system that replaces X-bar, which is known as **Bare Phrase Structure**.

Bare Phrase Structure

- Bare phrase structure is the minimalist approach to phrase structure.
 - Key properties formerly accounted for by X-bar schema are reduced to properties of **Merge** and a new operation **Project** (aka **Label**)
 - We've already seen that Merge on its own gives us hierarchical structure, binary branching and recursivity. The new operation **Project** gives us endocentricity.

Projection

- What kind of syntactic object is γ ?



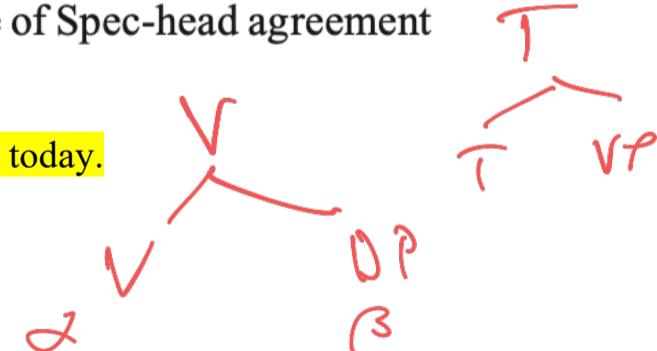
- Five possibilities:
 - γ is the same as α
 - γ is the same as β
 - γ is the union of α, β
 - γ is the intersection of α, β
 - γ is completely distinct from α, β

- Of these possibilities, only the first two are compatible with empirical evidence (e.g. if one of α, β is a V and the other is its object, then γ must pattern as a V in terms of distribution etc).
 - In other words, we want to encode **endocentricity** of phrase structure
- So the next question is how do we know whether γ will be the same as α or β ?
- We need a **labeling** mechanism:
 - 2) Projection** (Chomsky 1995, 2000, see Fukui & Narita 2014)

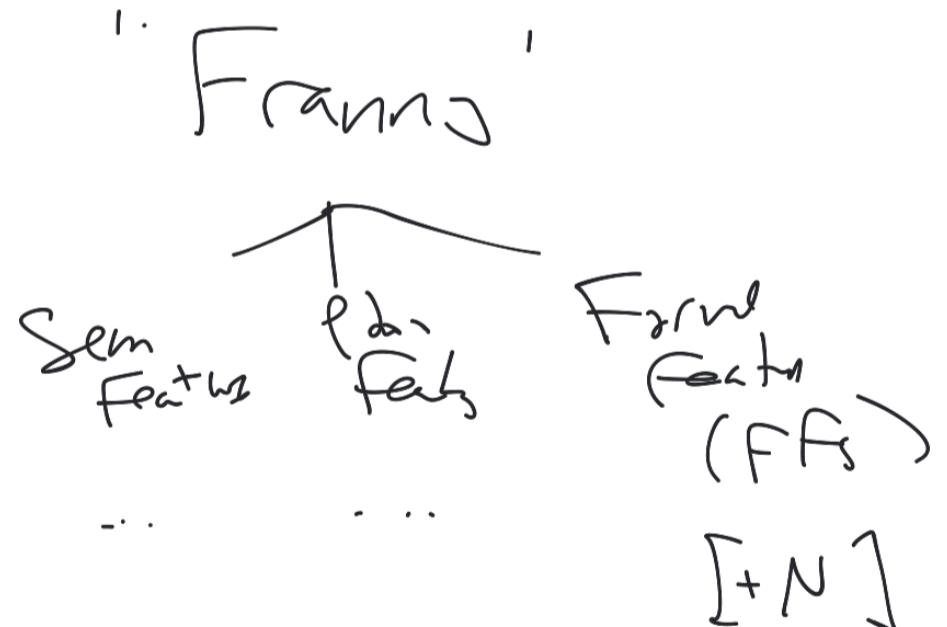
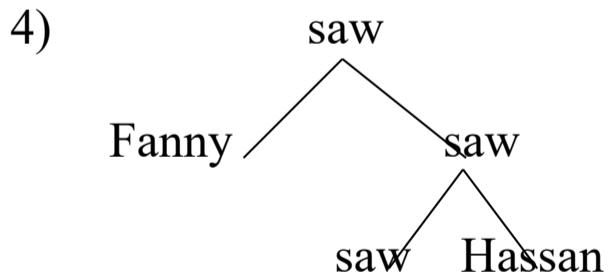
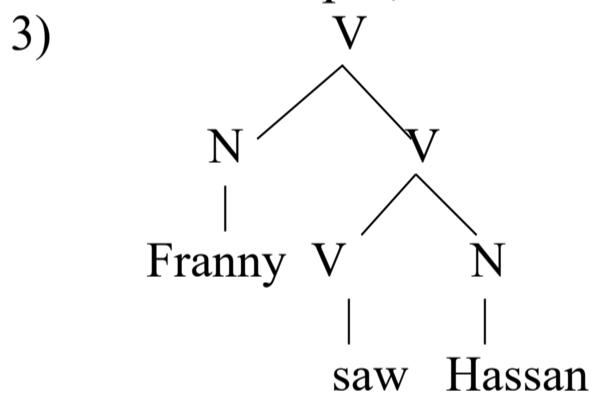
The output of $\text{Merge}(\alpha, \beta)$ is labeled by α if

- α selects β as its semantic argument, or
- α agrees with β : that is, β is attracted by α for the purpose of Spec-head agreement (feature-checking).

We won't talk about feature-checking until next week, so we'll focus on (a) today.



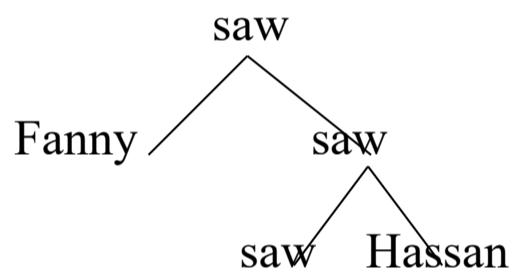
- Now let's consider what it means to be labeled by α .
- Q: Do we really need a distinction between terminal category nodes and lexical items?
 - What is the status of category labels?
 - What info does the category label N have which *Franny/Hassan* do not have? Remember, LIs include all syntactic formal features.
 - What would we lose if we replaced (3) with (4)? NB: this is a toy example, we are ignoring little v and higher structure.



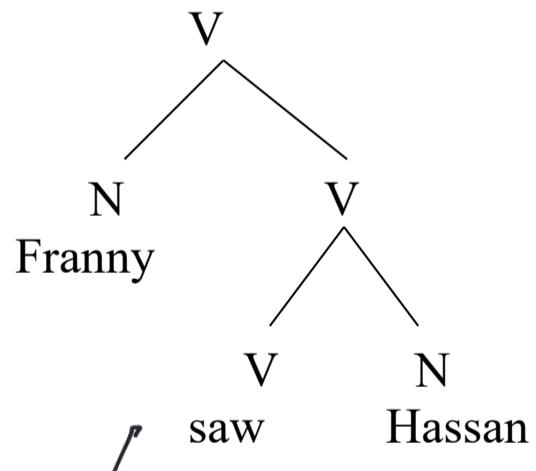
- In practice, it is easier to read trees that use labels, so we will continue to do so, but with the understanding that (6) is really (5).

-

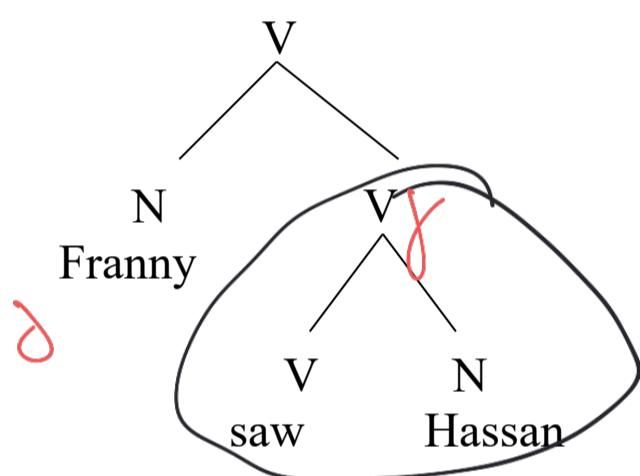
5)



6)



Complements and specifiers in bare phrase structure

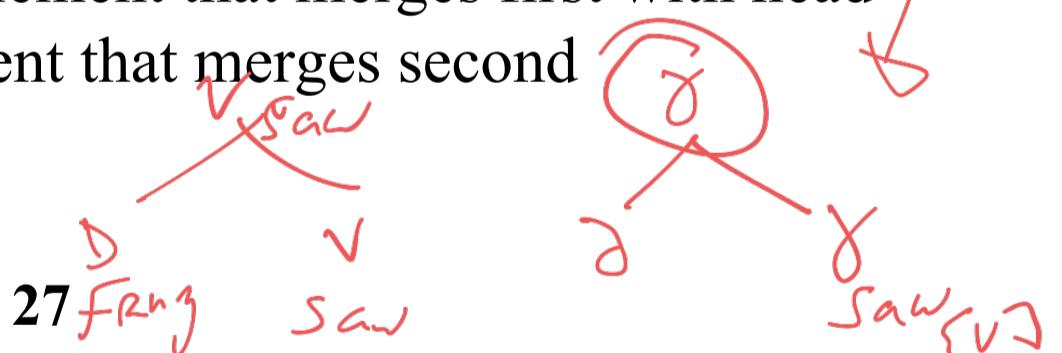


Merge 1: $\{\text{saw}, \text{Hassan}\}$
 Project : $\{\text{saw} \{\text{saw}, \text{Hassan}\}$
 Merge 2 : $\{\text{Franny}, \{\text{saw} \{\text{saw}, \text{Hassan}\}\}$
 Project 2 : $\{\text{saw} \{\text{Franny}, \{\text{saw} \{\text{saw}, \text{Hassan}\}\}\}$

- Now let's consider how we conserve the notions of **complement** and **specifier** in bare phrase structure. This is actually straightforward:

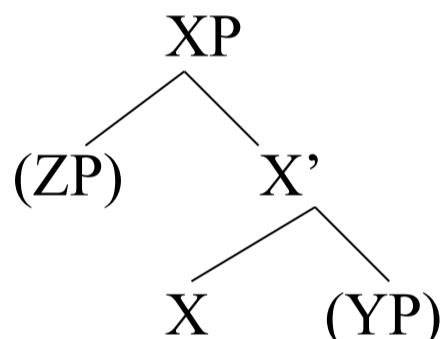
- Complement is the element that merges first with head
- Specifier is the element that merges second

N.B.: Bottom-up
structure build



Minimal, Intermediate and Maximal projections

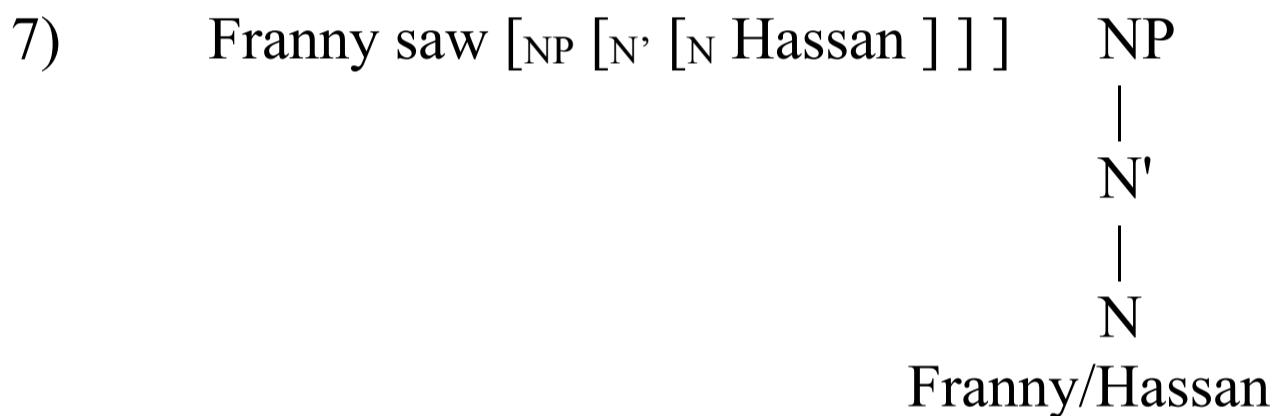
X-bar schema::



- How do we conceptualize the X-bar difference between X, X' and XP?
 - Their difference may be due to them having different intrinsic features, in roughly the same way nouns differ from verbs. Under this view, the difference is a primitive of the system. This is the X'-view.
 - Their difference may be relational, in roughly the same way a subject differs from an object. This is the minimalist view.

Q: Think about what happens in X-bar if the phrase consists of a single head?

We need vacuous levels on theory-internal grounds: *Franny* or *Hassan* in (7).



- Some questions about X-bar:
 - Why did we need these extra levels?
 - Why are complements, specifiers and adjuncts maximal projections?
 - How do we conceptualize the difference between X, X' and XP?

- The three-bar-level analysis of *Franny/Hassan* is based on the GB view that there are intrinsic differences between levels.
- The relational approach can be implemented via the definitions below:

1) *Minimal Projection*: X^0

A minimal projection is a lexical item selected from the numeration.

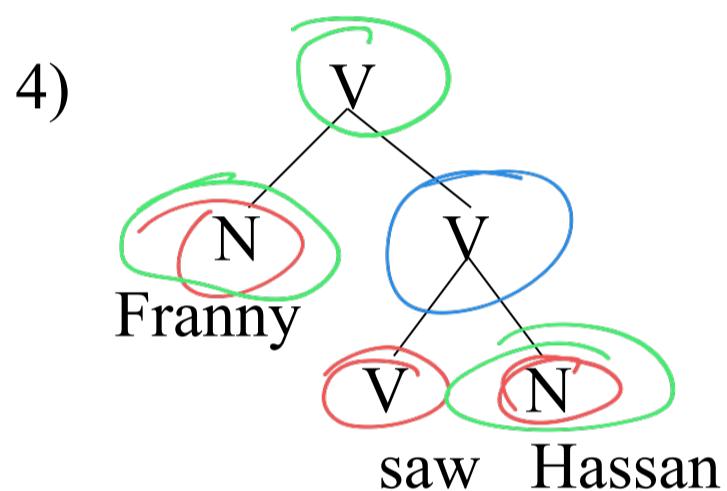
2) *Maximal Projection*: XP

A maximal projection is a syntactic object that doesn't project.

3) *Intermediate Projection*: X'

An intermediate projection is a syntactic object that is neither an X^0 nor an XP .

Exercise: Apply these definitions to the tree below to determine the status of each node in this tree structure.



- ↳ - minimal
- ↳ - internalist ($\rightarrow \text{min}, \text{max}$)
- ↳ - maximal

- The relational approach has some advantages:
 - It dispenses with vacuous intermediate projections.
 - It derives the claim that complements, adjuncts and specifiers are maximal, as they don't project further.
 - It complies with the **Inclusiveness Condition** which requires that LF objects be built from features of the lexical items. Under this view, “⁰”, “ ” and “P” aren't theoretical primes!

Adjuncts

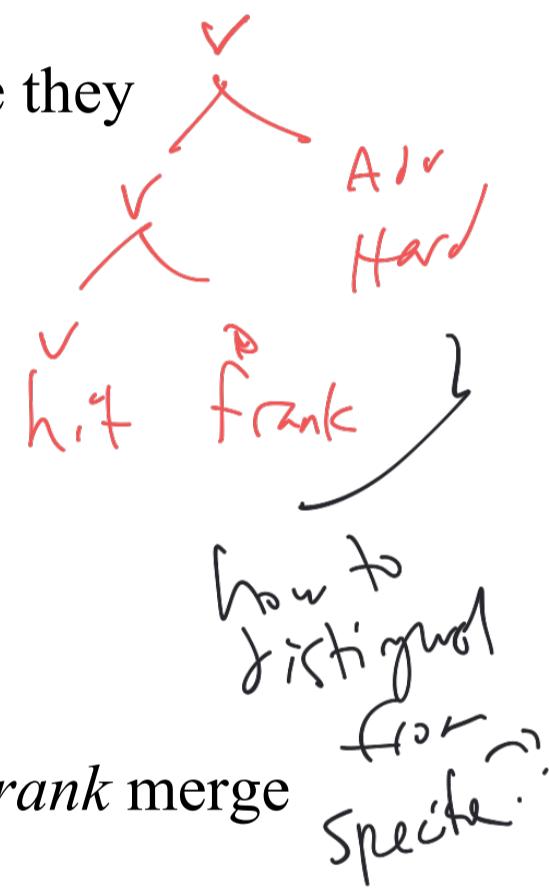
- Q: How are adjuncts treated in the system? How are they distinguished from specifiers?

Adjuncts have distinct properties: they arguably

- don't enter into agreement relations,
- have different requirements from arguments,
- are interpreted as conjuncts semantically,
- come in a wide variety of category types.

Let's take the example *hit Frank hard*. First, *hit* and *Frank* merge and *hit* projects. This is shown in (5).

5) {hit, {hit, Frank}}}



Now, at this point, *hard* merges with (5), but what is the label of the result?

6) $\{?, \{\{hit, \underline{\{hit, Frank\}}\}, \underline{hard}\}\}$

If *hit* projects in the same manner, then *hit* in (5) is no more a maximal projection and *hard* is essentially interpreted as a spec!

To avoid this problem, we need a different type of labeling mechanism for Adjuncts.

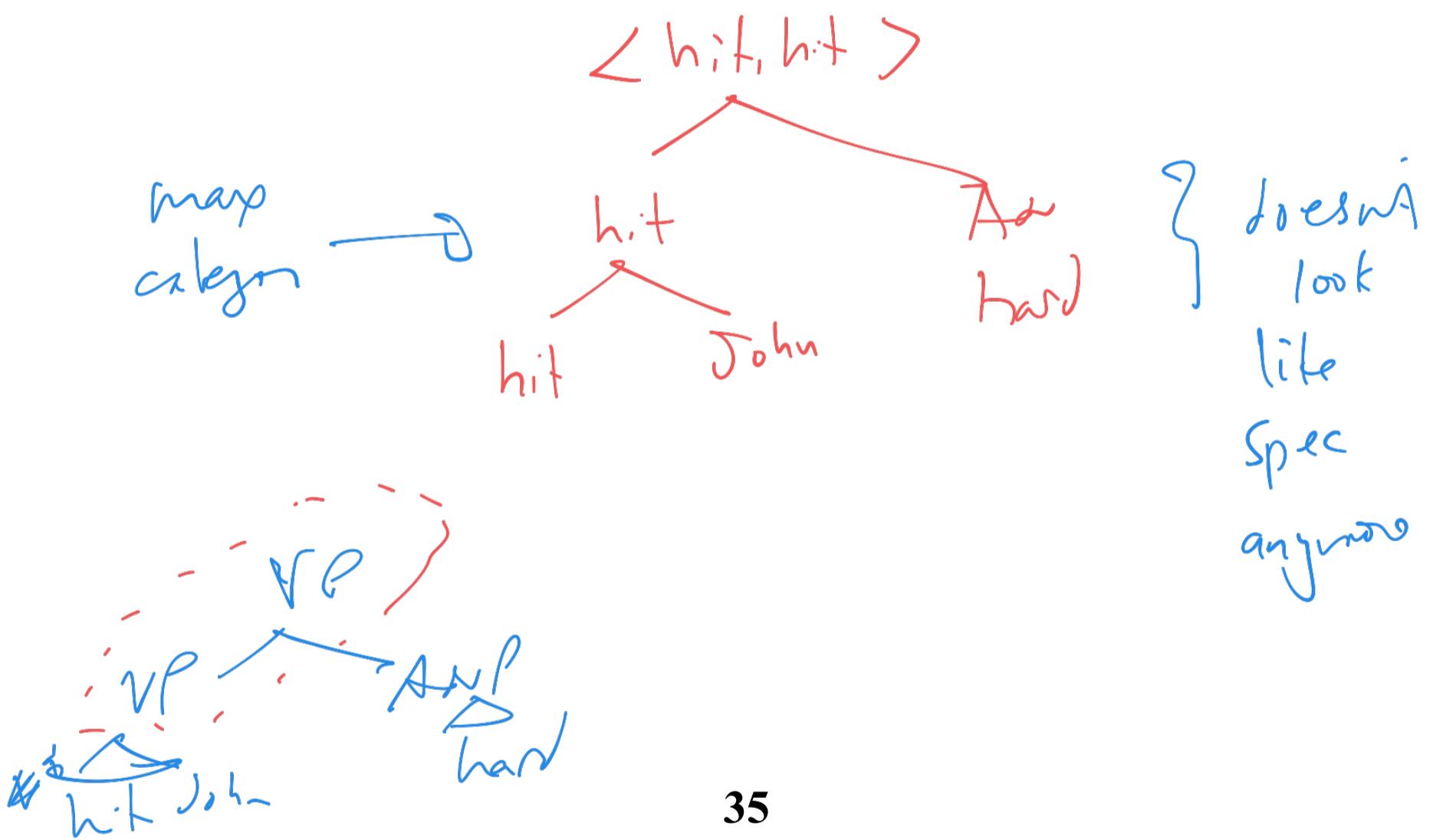
- This has been implemented by stipulating a different kind of Merge known as **pair Merge** (the original Merge, by contrast, known as **set Merge**).

In (7). The pair $\langle hit, hit \rangle$ is taken to indicate that Pair Merge has occurred. That is, the structure in (5), whose label is *hit*, determines the label of the structure in (7), but doesn't project!

7) $\{\langle hit, hit \rangle, \{\{hit, \{hit, John\}\}, hard\}\}$

{<hit, hit>, {{hit, {hit, John}}, hard}}

In a tree structure this would look like this:



Bare Phrase Structure and little v

- Little v was important for bare phrase structure because it allows for a shift to construing theta-role assignment as being entirely configurational. See discussion in Harley 2015 (reading from two weeks ago).
- To see this, let's look at intransitive verbs.

Unaccusative and Unergative verbs

- Standard GB assumption: There are two types of verbs with single arguments:
 - **unergative** verbs whose single argument behaves like an external one
 - **unaccusative** verbs whose single argument behaves like an internal one

- How do the paradigms in (8)-(10) support the unaccusative/unergative distinction?

8) Italian

- a. Giovanni [ha] / *è comprato un libro.
Giovanni has/is bought a book
 ‘Giovanni bought a book.’
- b. Giovanni [ha] / *è telefonato.
Giovanni has/is called
 ‘Giovanni called.’
- c. Giovanni [è] / *ha arrivato.
Giovanni is/has arrived
 ‘Giovanni arrived.’

transitive (EA, IA)

unergative (EA)

unaccusative (IA)

9) Portuguese

- a. A Maria comprou os livros.
the Maria bought the books
 ‘Maria bought the books.’
- b. Comprados os livros, ...
buy.PART.MASC.PL the books
 ‘After the books were bought, ...’
- c. *Comprada a Maria, ...
buy.PART.FEM.SG the Maria
 ‘After Maria bought (something), ...’
- d. Chegada a Maria, ...
arrive.PART.FEM.SG the Maria
 ‘After Maria arrived, ...’
- e. *Espirrada a Maria, ...
sneeze.PART.FEM.SG the Maria
 ‘After Maria sneezed, ...’

transitive

passive

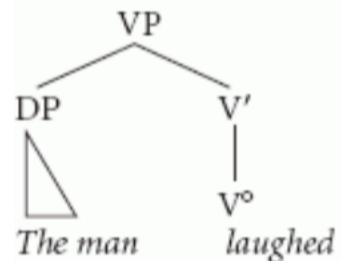
~~transitive w/ implied object~~

- 10)
- a. John smiled (a beautiful smile). *concrete* *ergative*
 - b. John arrived (*an unexpected arrival). *unaccusative*

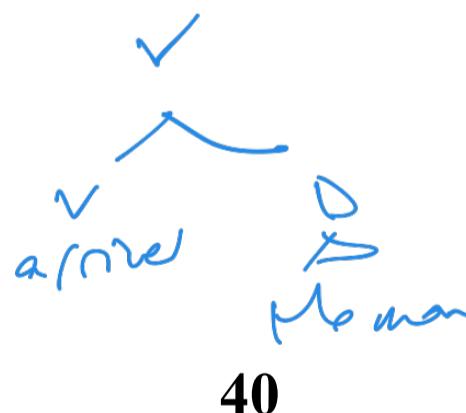
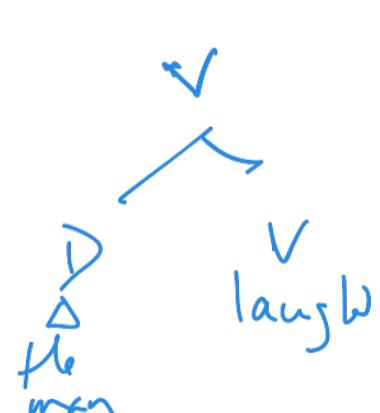
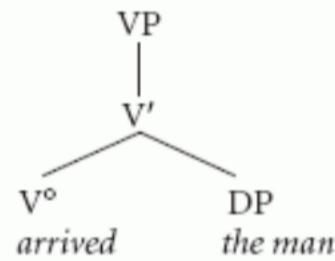
This distinction has been traditionally accounted for in terms of the position where the only argument is base-generated, as shown in (11).

- 11) a. Unergative verbs: [VP DP [V' V]]
 b. Unaccusative verbs: [VP V DP]

Unergative verbs in GB theory



Unaccusative verbs in GB theory

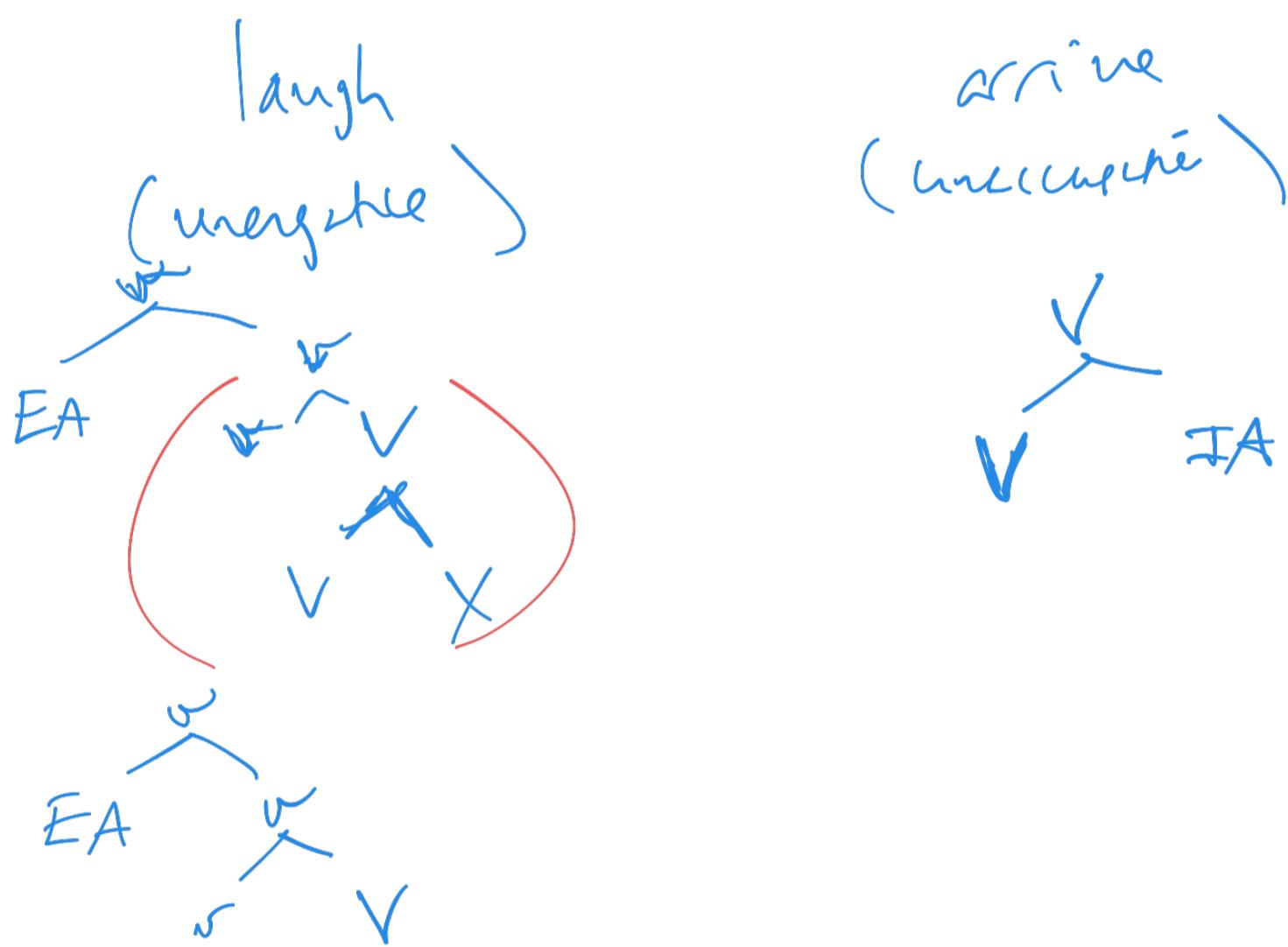


Harley 2011:(14))

identical CFC
in BPS than
is no linear order
in the structure

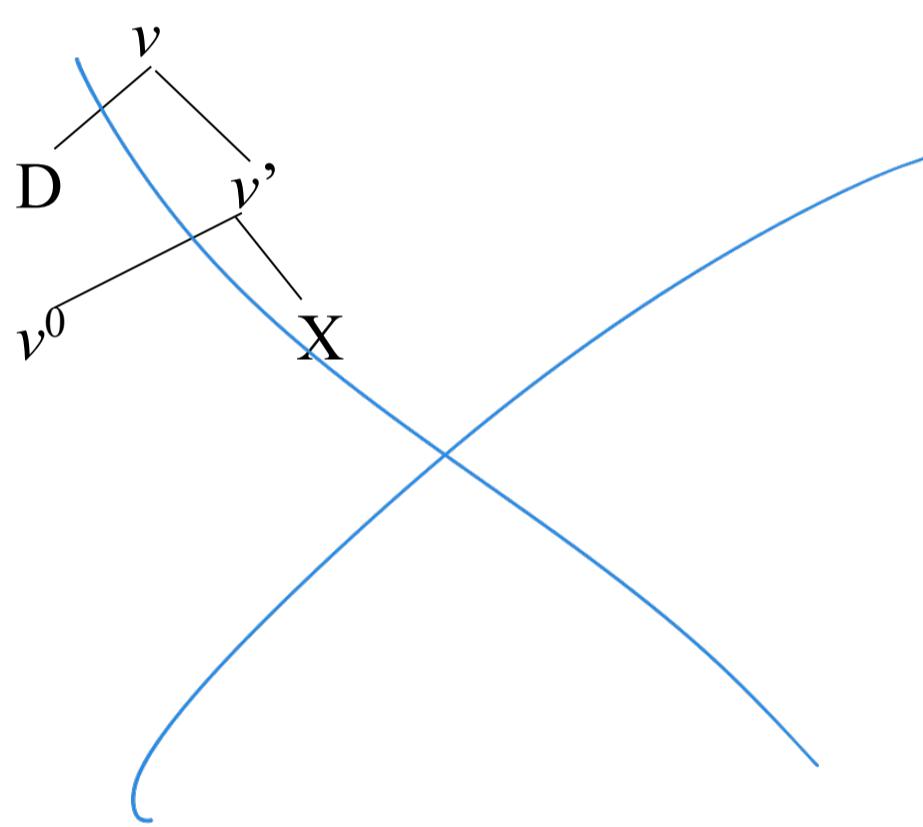
The GB structures distinguishing unaccusatives and unergatives
are problematic from a minimalist perspective. Can you say why?

- What would unaccusatives and unergatives look like in bare phrase structure, if we didn't have little v?



- To overcome the problem, we extend the light v structure to unergatives, as in (12).

12)



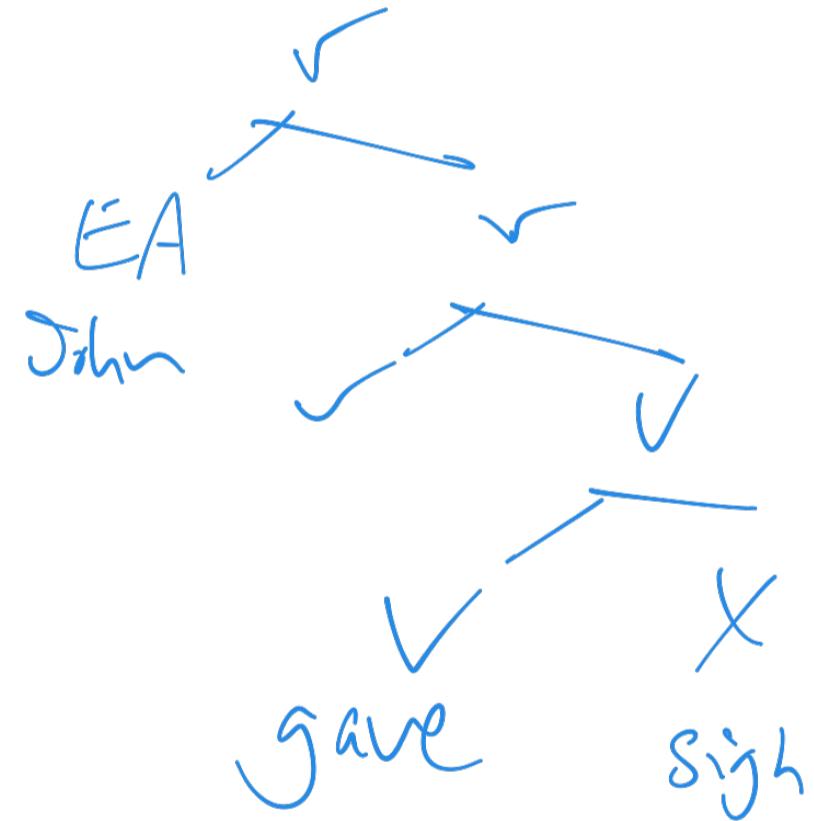
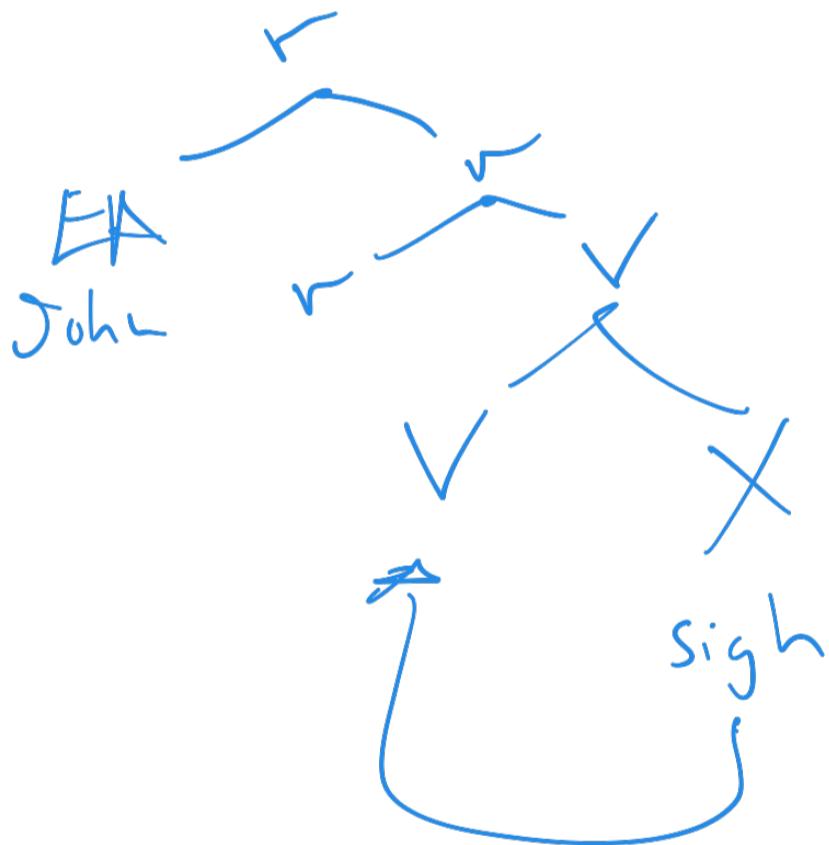
We did this
on the previous slide

- What should unaccusatives look like?

Did this
if

- This also enables us to assign a uniform configuration to pairs such as the one in (13).

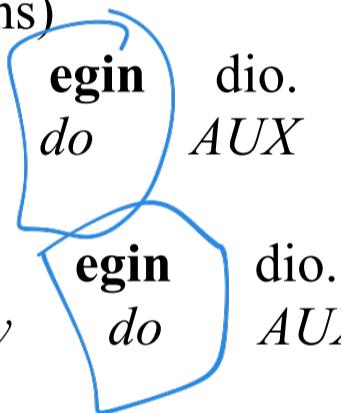
- 13) a. John sighed.
 b. John gave a sigh.



- Further evidence: Languages which have an overt light verb for both transitive and unergative but not unaccusative constructions, as shown for Basque in (14)-(16).

14) *Basque* (transitive constructions)

- a. Jonek Mireni min **egin** dio.
Jon.ERG Miren.DAT hurt **do** *AUX*
 ‘Jon hurt Miren.’
- b. Jonek kandelari putz **egin** dio.
Jon.ERG candle.DAT blow **do** *AUX*
 ‘Jon blew out the candle.’



15) *Basque* (unergative constructions)

- a. Emakumeak barre **egin** du.
woman.DEF.ERG laugh **do** *AUX*
 ‘The woman has laughed.’
- b. Nik eztul **egin** dut.
I.ERG cough do *AUX*
 ‘I have coughed.’



16) *Basque* (unaccusative constructions)

- a. Emakumea erori da.
woman.DEF.ABS fallen AUX
‘The woman has fallen.’
- b. Kamioiak etorri dira.
truck.DET.PL arrived AUX
‘The trucks have arrived.’

Looking a bit ahead

- The use of little v to enable uniform theta-role configurations dovetails with minimalist goal of eliminating stipulated relations like **government** which previously needed to state the domain of theta-role assignment. More on government next week.