

# Syntactic Theory 2

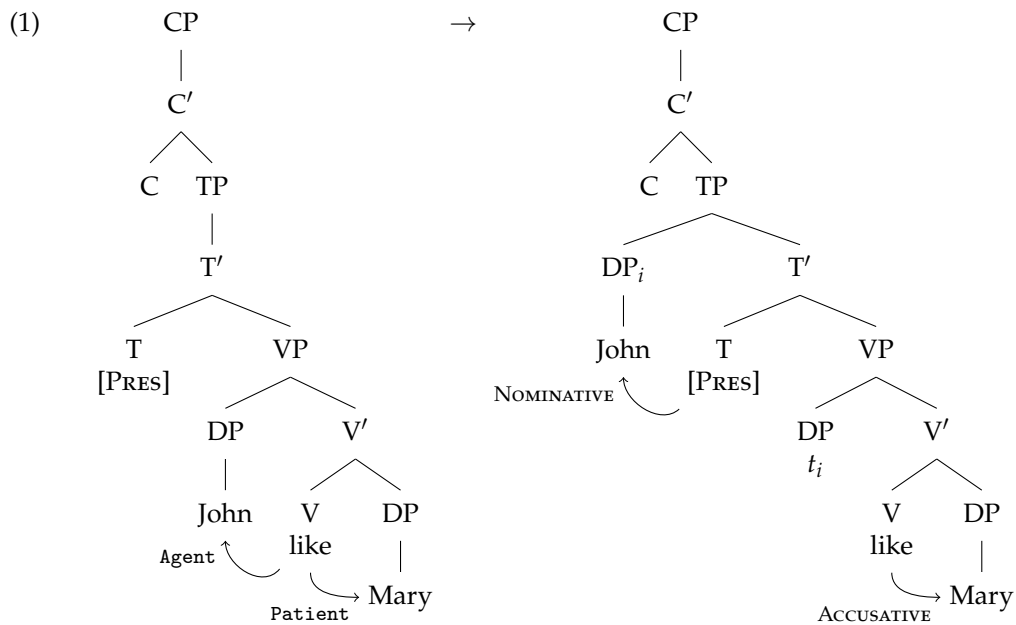
## Week 5: Merge and Linearization

Dustin A. Chacón

February 7, 2017

### 1 Introduction

- In GB, D-Structure was the representation where: phrase structure is assembled in accordance with  $X'$ -theory, theta-roles are assigned. It is also the input to movement, which produces S-Structure:



- Last time, we saw a new way to think about phrase structure, in which in place of D-Structure, we interleave Merge and Move operations, and let theta roles get assigned upon first Merge:

- (2)  $\text{Merge}(\alpha, \beta) = \{\gamma, \{\alpha, \beta\}\}$ , where  $\gamma$  is a label meeting the conditions of  $X'$  Theory
- (3)  $\text{Merge}(\text{like}, \text{DP}) = [\text{V}' \text{ like } [\text{DP} \text{ Mary}_{[\text{Case:Acc}, \text{Theta:Pat}]}]]$   
 $\text{Merge}(\text{DP}, \text{V}') = [\text{VP} [\text{DP} \text{ John}]_{[\text{Case:???, Theta:Ag}]} [\text{V}' \text{ like } [\text{DP} \text{ Mary}]_{[\text{Case:Acc}, \text{Theta:Pat}]}]]$

$$\begin{aligned}
\text{Merge}(T,VP) &= [T' T_{[PRES]} [VP [DP \text{John}]_{[Case:???,Theta:Ag]} [V \text{like Mary}]]] \\
\text{Move } [DP \text{John}] &= [TP [DP \text{John}]_{[Case:NOM,Theta:Ag]}_i [T' T_{[PRES]} [VP [DP t_i] [V \text{like Mary}]]] \\
\text{Merge}(C,TP) &= [CP [C' C [TP [DP \text{John}]_{[Case:NOM,Theta:Ag]}_i [T' T_{[PRES]} [VP t_i \text{like Mary}]]]]
\end{aligned}$$

- A move to Merge gets rid of the commitment that theta roles are assigned before any and all movements, which we saw was necessary to account for tough-movement constructions:

(4) John is easy  $OP_i$  to please  $t_i$

- At this point, we've replaced D-Structure with the operations Merge and the Numeration (or **Lexical Array**), which pre-determines the number of lexical items that will enter the derivation
- In this lecture, we will examine what the move to Merge + Numeration buys us, and what other commitments we can loosen ourselves from given Minimalist desiderata

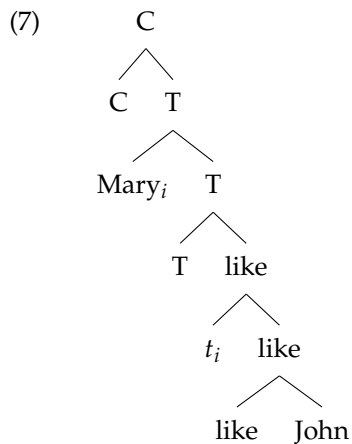
## 2 Merge and Bare Phrase Structure

- Merge forces binary branching, given Economy conditions – 2 is the smallest thing that a combinatoric operation can be defined over
- This also means there is no “unary branching”
- **Inclusiveness Condition:** The derivation does not add any information that was not a part of the lexical array

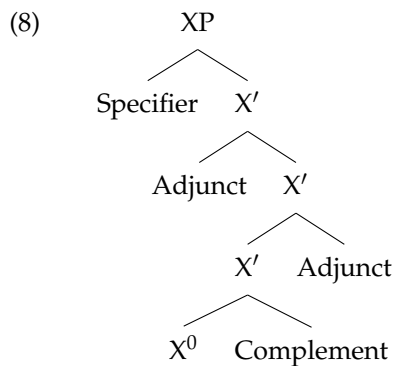
(5)  $\text{Merge}(\alpha, \beta) = \{\gamma, \{\alpha, \beta\}\}$ , where  $\gamma = \alpha$  or  $\beta$

- Additionally, the Inclusiveness Condition means we no longer have “bar-levels” in our theory of phrase structure – the theory makes no allotments for the distinction between  $X^0/X'/XP$ :

$$\begin{aligned}
(6) \quad \text{Merge}(\text{like}, \text{John}) &= [_{\text{like}} \text{like John}] \\
\text{Merge}(\text{Mary}, \text{like}) &= [_{\text{like}} \text{Mary } [_{\text{like}} \text{like John}]] \\
\text{Merge}(T, \text{like}) &= [T T [_{\text{like}} \text{Mary } [_{\text{like}} \text{like John}]]] \\
\text{Move Mary} &= [T \text{Mary}_i [T T [_{\text{like}} t_i [_{\text{like}} \text{like John}]]]] \\
\text{Merge}(C, T) &= [C C [T \text{Mary}_i [T T [_{\text{like}} t_i [_{\text{like}} \text{like John}]]]]]
\end{aligned}$$



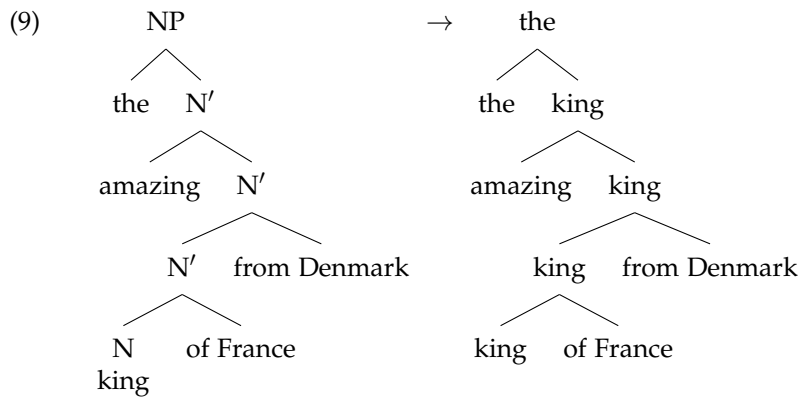
- Is this a good thing? Let's re-examine the properties of  $X'$ -Theory (Chomsky 1970)



- Properties of  $X'$  Theory:
  - One complement adjacent to the head
  - Multiple adjuncts, freely interleavable
  - A single specifier at the top
  - The properties of the head  $X^0$  "project" throughout the structure
- In Bare Phrase Structure...
  - Complement is the *first* thing to Merge with a lexical item
  - Specifiers are the *last* thing to Merge with a lexical item
  - Adjuncts are the elsewhere condition
- **Hypothesis:** Bare Phrase Structure implies that operations may target the *maximal* projection or the *minimal* projection, but nothing may target *intermediate* projection
- John invited the amazing king of France from Denmark, and Mary invited...

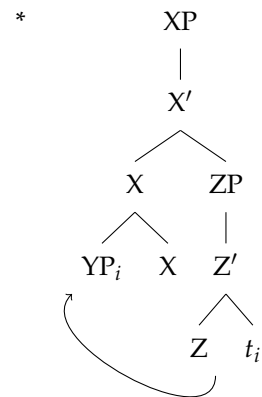
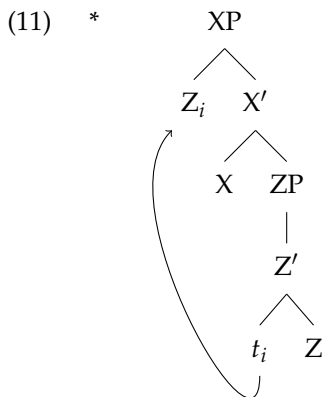
- a. \*the one (= amazing king of France from Denmark)
- b. the fantastic one (= king of France from Denmark)
- c. the amazing one from London (= king of France)
- d. the amazing one of Spain from London (= king)

- Traditionally, *one*-substitution is evidence for N'-levels (Baker 1979). However, with the DP hypothesis and BPS, we can reformulate this as an operation that targets a *non-minimal* projection

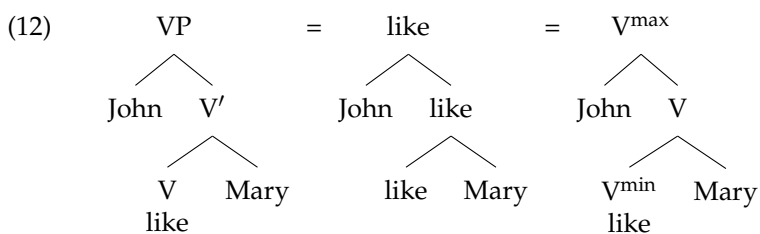


- Head-movement always targets another head position, and phrasal movement always targets a phrasal position (Emonds 1970):





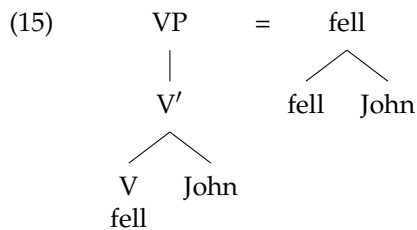
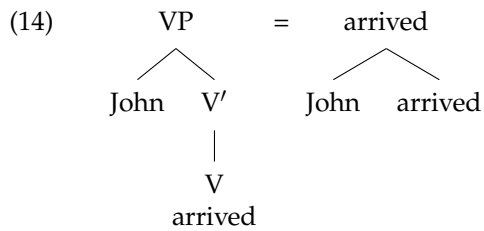
- We can say that head-movement targets *minimal projections*, and phrasal-movement targets *maximal projections*.



- BPS allows us to keep most of our analyses that rely on X' Theory in place, with minimizing our assumptions about what the computational system is actually representing. This is an effort in *explaining* why phrase structure cares about what it cares about
- One upshot of BPS – a single lexical item can be both “maximal” and “minimal” at the same time:

(13) María le quiere ver a Juan  
 Mary him want to.see ACC John  
 ‘Mary wants to see John’

- *le* seems to move from the lower clause, since it receives its thematic role from the verb *ver*. It looks like XP movement since it isn’t blocked by the intervening heads, but it doesn’t look like head movement since it phonologizes with the higher verb *quiere*, being a clitic. If *le* is *maximal/minimal*, then we have the beginning of an explanation for why clitics show special behavior
- Without binary branching, we lose the analysis of unergative/unaccusatives, due to the lack of unary branching:



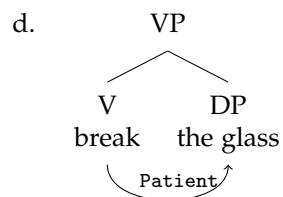
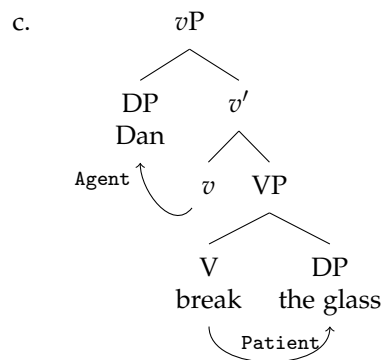
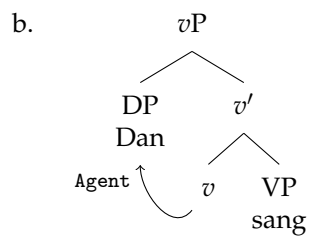
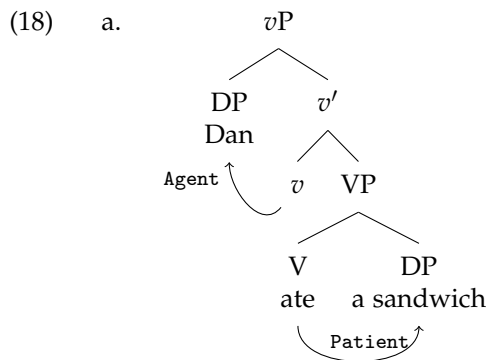
(15) However, the object seems to be “closer” to the verb than the subject. That is, the verb together with the object seems to determine how the subject participates in the event. (Kratzer 1996)

- (16)
- a. Becca took a nap
  - b. Becca took a piss
  - c. Becca took a chance
  - d. Becca took a turn
  - e. Becca took a picture
  - f. Becca took a look
  - g. Becca took a penny

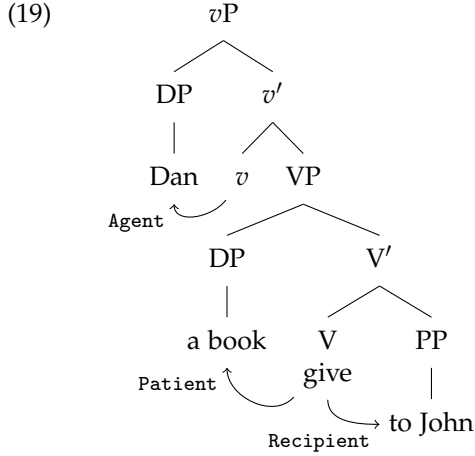
- Additionally, some languages morphologically encode the presence of an agent:

- (17)
- a. barf-u    karg-i-tu  
ice-NOM melt-PST-3SN  
'The ice melted' (Kannada, Lidz 2003)
  - b. surya barf-annu karag-is-i-tu  
sun ice-ACC melt-CAUS-PST-3SN  
'The sun melted the ice' (Kannada, Lidz 2003)

- We posit that agents are introduced by *v*, or “little *v*”, in Distributed Morphology called *voi* ('voice'). This is a “light verb”, introduces causative morphology:



- What goes in Spec,VP now? The little *v* analysis gives us an explanation for the double-object construction in English:



- Another upshot – we now say that the patient theta role is assigned upon merger with a V, and the agent theta role is assigned upon merger with  $v$  – there is **no reliance on government!** The introduction of  $v$  means that we might not need government in our theory either – that’s a good thing!

(20)

$$\begin{aligned}
 \text{Merge}(\text{like}, \text{John}) &= [\text{like like John}_{[\text{Theta:PAT}]}] \\
 \text{Merge}(v, \text{like}) &= [v v [\text{like like John}_{[\text{Theta:PAT}]}]] \\
 \text{Merge}(\text{Mary}, v) &= [v \text{Mary}_{[\text{Theta:AGENT}]} [v v [\text{like like John}_{[\text{Theta:PAT}]}]]]
 \end{aligned}$$

- In a Merge-based system, satisfying features is the name of the game. A head merges with a complement in order to satisfy its selectional features, and presumably this is the basis on which labeling occurs. That is, when *devour* merges with a DP, it’s to satisfy *devour*’s lexically specified requirement for a D complement, triggering the Labeling algorithm to label the resulting structure a projection of *devour*

(21)

$$\text{Merge}(\text{devour}, \text{it}) = [\text{devour devour}_{[\text{D}]} \text{it}_{[\text{D}]}]$$

- In Minimalism, we’ll place a lot of explanatory weight on features being introduced and then “checked” or “valued” in the derivation. Since Minimalism is a derivational theory, and Economy considerations imply that the derivation won’t do anything unless it has to, when and where features are introduced and checked will be crucial to our analyses.

### 3 Movement

- **Inclusiveness Condition:** The derivation does not add any information that was not a part of the lexical array
- Traces are theory-internal objects that GB postulates to ensure that theta-roles (a D-Structure concept) are “legible” at LF



- Reconstruction phenomena and binding phenomena imply that traces are representationally live:

(22) S-Structure: It was unknown [<sub>CP</sub> [which pictures of himself<sub>i</sub>]<sub>j</sub> John liked *e<sub>j</sub>* ]

(23) John<sub>j</sub> wondered [[*t<sub>i</sub>* how proud of himself<sub>i/\*j</sub>]<sub>i</sub> Ivan<sub>i</sub> was]

- What is the minimal theory of traces? There are none! Instead, we posit that traces are lower copies of the same phrase
- Two distinct copy theories of movement:

(24) **Copy Theory of Movement (v 1.0):** Copy a syntactic object from the structure, and the Merge it at the root, as required by the Extension Condition (Chomsky 1995)

(25) Merge(like,John) = [<sub>like</sub> like John]  
 Merge(*v*,like) = [<sub>v</sub> *v* [<sub>like</sub> like John]]  
 Merge(Mary,*v*) = [<sub>v</sub> Mary [<sub>v</sub> *v* [<sub>like</sub> like John]]]  
 Merge(T,*v*) = [<sub>T</sub> T [<sub>v</sub> Mary [<sub>v</sub> *v* [<sub>like</sub> like John]]]]  
 Copy Mary = Mary, [<sub>T</sub> T [<sub>v</sub> Mary [<sub>v</sub> *v* [<sub>like</sub> like John]]]]  
 Merge(Mary,*v*) = [<sub>T</sub> Mary [<sub>T</sub> T [<sub>v</sub> Mary [<sub>v</sub> *v* [<sub>like</sub> like John]]]]]

(26) **Copy Theory of Movement (v 2.0):** Merge may be “external” (merging from the Lexical Array) or “internal” (merging something previously merged); these are the same operation (Chomsky 2001)

(27) Merge(like,John) = [<sub>like</sub> like John]  
 Merge(*v*,like) = [<sub>v</sub> *v* [<sub>like</sub> like John]]  
 Merge(Mary,*v*) = [<sub>v</sub> Mary [<sub>v</sub> *v* [<sub>like</sub> like John]]]  
 Merge(T,*v*) = [<sub>T</sub> T [<sub>v</sub> Mary [<sub>v</sub> *v* [<sub>like</sub> like John]]]]  
 Merge(Mary,*v*) = [<sub>T</sub> Mary [<sub>T</sub> T [<sub>v</sub> Mary [<sub>v</sub> *v* [<sub>like</sub> like John]]]]]

- On CToM v1.0, Move is two operations – Copy and Merge. On CToM v2.0, they are the same operation. Functionally, these theories are almost identical (Chomsky 1995)
- However, Chomsky (1995) proposes that CToM v1.0 makes a prediction – if Move is actually 2 suboperations, then there should be a **Merge Over Move** preference – if the derivation may merge something from the lexical array or move something downstairs, it should merge from the lexical array first

(28) a. There<sub>*t<sub>i</sub>*</sub> seems *t<sub>i</sub>* to be a cat here  
 b. \*There seems a cat<sub>*t<sub>i</sub>*</sub> to be *t<sub>i</sub>* here

(29) Merge(a cat, here) = [<sub>here</sub> [a cat] here]  
 Merge(be, here) = [<sub>be</sub> be [<sub>here</sub> [a cat] here]]  
 Merge(to, here) = [<sub>to</sub> to [<sub>be</sub> be [<sub>here</sub> [a cat] here]]]  
**Merge(there,to)** = [<sub>to</sub> there [<sub>to</sub> to [<sub>be</sub> be [<sub>here</sub> [a cat] here]]]]  
 Merge(seem, to) = [<sub>seem</sub> seem [<sub>to</sub> there [<sub>to</sub> [<sub>here</sub> [a cat] here]]]]  
 Merge(T, seem) = [<sub>T</sub> T [<sub>seem</sub> seem [<sub>to</sub> there [<sub>to</sub> to [<sub>be</sub> be [<sub>here</sub> [a cat] here]]]]]]]

- Copy there** = there, [<sub>T</sub> T [<sub>seem</sub> seem [<sub>to</sub> there [<sub>to</sub> to [<sub>be</sub> be [<sub>here</sub> [a cat] here]]]]]]]  
**Merge(there, T)** = [<sub>T</sub> there [<sub>T</sub> T [<sub>seem</sub> seem [<sub>to</sub>there [<sub>to</sub> to [<sub>be</sub> be [<sub>here</sub> [a cat] here]]]]]]]]]  
(30) Merge(a cat, here) = [here [a cat] here]  
Merge(be, here) = [be be [here [a cat] here]]  
Merge(to, here) = [to to [be be [here [a cat] here]]]  
**Copy a cat** = [a cat], [to to [be be [here [a cat] here]]]  
**Merge(a cat, T)** = [to [a cat] [to to [be be [here [a cat] here]]]]  
Merge(seem, to) = [seem seem [to [a cat] [to to [be be [here [a cat] here]]]]]  
Merge(T, seem) = [<sub>T</sub> T [<sub>seem</sub> seem [to [a cat] [to to [be be [here [a cat] here]]]]]]]  
**Merge(there, T)** = [<sub>T</sub> there [<sub>T</sub> T [<sub>seem</sub> seem [to [a cat] [to to [be be [here [a cat] here]]]]]]]]]

- If (28-b) is bad because it has a less economical derivation (i.e., there's a step in which we do more than is necessary), then this implies that CToM v1.0 is the right way to go
- Either way, both versions of the copy theory of movement make the same proposal – traces don't exist, but lower copies do. If so, why is the following ungrammatical?:

(31) \*John was seen John

- (32) Merge(John, seen) = [seen seen John]  
Merge(was, seen) = [was was [seen seen John]]  
Merge(John, was) = [was John [was was [seen seen John]]]

- We propose that PF has an Economy condition of its own – delete lower copies for ease of articulation

(33) Spell-Out: John was seen John  
PF: John was seen ~~John~~

- This explains why “traces” are silent
- However, there are cases where long-distance movement has a phonological effect. First, recall that subjacency requires us to postulate that long-distance movement is successive-cyclic (Chomsky 1977):

(34) [<sub>CP</sub> who<sub>i</sub> did [<sub>TP</sub> John [<sub>VP</sub> say [<sub>CP</sub> t'<sub>i</sub> that [<sub>TP</sub> Mary [<sub>VP</sub> liked t<sub>i</sub> ]]]]]]]

- In some Germanic varieties, we find intermediate copies having phonological realizations:

- (35) a. [<sub>CP</sub> Wen<sub>i</sub> glaubst du [<sub>CP</sub> wen'<sub>i</sub> [<sub>TP</sub> sie t<sub>i</sub> getroffen hat? ]]]  
when think you when she met has  
‘When do you think she has met?’ (German)  
b. [<sub>CP</sub> Wer<sub>i</sub> tinke jo [<sub>CP</sub> wêr' t [<sub>TP</sub> Jan t<sub>i</sub> wennet? ]]]  
Where think you where that-CL Jan resides  
‘Where do you think that Jan resides’ (Frisian)<sup>1</sup>

<sup>1</sup>German and Frisian taken from Felser (2004)

- If PF can delete copies, perhaps LF can too. Recall:

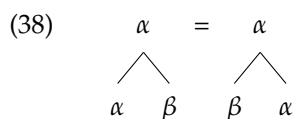
(36) Jane<sub>j</sub> wondered which picture of herself<sub>i/j</sub> Irene liked t<sub>i</sub>

(37) Spell-Out: Jane wondered which picture of herself Irene liked which picture of herself  
 PF: Jane wondered which picture of herself Irene liked ~~which picture of herself~~  
 LF: Jane<sub>j</sub> wondered which picture of herself Irene liked ~~which picture of herself~~  
 Jane wondered ~~which picture of herself~~ Irene<sub>i</sub> liked which picture of herself<sub>i</sub>

- Optionality for LF copy deletion can explain these kind of reconstruction asymmetries – if binding occurs at LF
- This raises a question – if intermediate copies can sometimes appear at PF, and if LF copies can freely delete as one remains – why does PF **in general** require the lower copy of a movement chain to delete? We’ll examine this question when dealing with linearization

## 4 Linearization

- **Inclusiveness Condition:** The derivation does not add any information that was not a part of the lexical array
- **Head-Directionality Parameter:** A head and complement may surface as [<sub>X'</sub> X YP] or [<sub>X'</sub> YP X]
- Merge is an operation that forms sets, which are definitionally unordered – i.e., the two trees are “the same” from the perspective of the syntax:



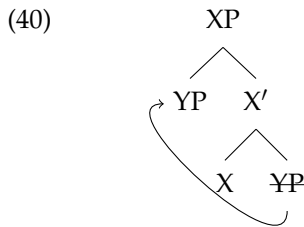
- Related issues?: Specifiers always are to the left; adjuncts are often to the left, and movement is almost always to the left
- Proposal: PF imposes linear order on structure, structure has no notion of order
- **Linear Correspondence Axiom (LCA), or Antisymmetry:** If  $\alpha$  c-commands  $\beta$ , then  $\alpha \prec \beta$  (Kayne 1994, Chomsky 1995)
- Spell-Out: [will John [will will [go go [to to [the the [store store]]]]]]  
 PF: John  $\prec$  will  $\prec$  go  $\prec$  to  $\prec$  the  $\prec$  store
- Movement is always to a c-commanding position  $\rightarrow$  movement is always “left”
- The LCA has the unfortunate consequence of implying that all languages are in some sense underlyingly SVO – the only way to get an object above a verb is to move it:

- (39) a. Rām-ne Sitā-ko pasand kartā hai  
 Ram-ERG Sita-ACC like do PRES  
 'Ram likes Sita'  
 b. Rām-ne Sitā-ko<sub>i</sub> [<sub>VP</sub> pasand kartā hai *t<sub>i</sub>* ]

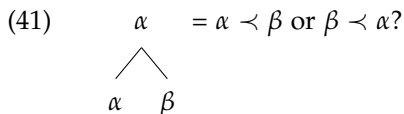
- More recent approaches have suggested that PF requires that c-command map onto linearization, but can arbitrarily choose whether  $X^0 \prec \text{Comp}$  or  $\text{Comp} \prec X^0$ . Additionally, Biebrauer *et al* (2007) have proposed that languages can choose for specified syntactic domains how Merge is linearized.
- Regardless, the idea that syntactic structures are one-dimensional, and PF imposes linearization, is still an important idea. Additionally, as you will see in the homework, linearization algorithms can be used to explain some important phenomena.

## 5 Outstanding Issues

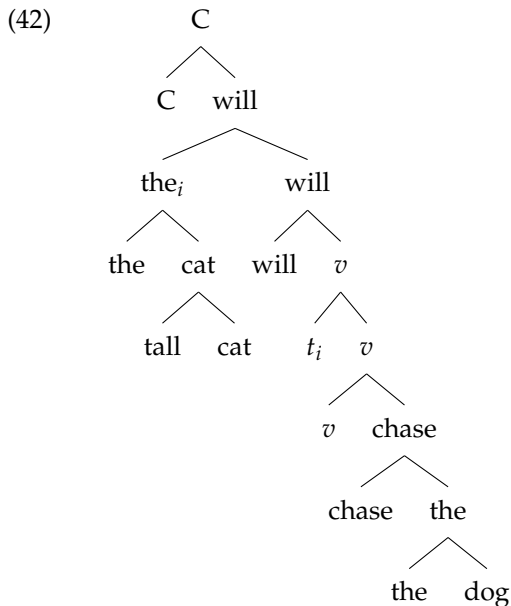
- Features drive all operations in Minimalism. Additionally, Bare Phrase Structure tells us that  $X^0 = X' = XP = X$ . What's unusual about the following derivation?:



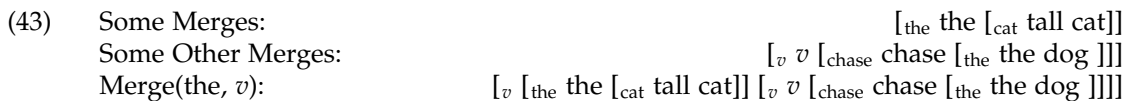
- Grohmann (2003), Abels (2003) propose that there is a “anti-locality” ban that requires movement cross at least one phrase – it may be that  $\text{Merge}(\alpha, \beta)$  followed by another instance of  $\text{Merge}(\alpha, \beta)$  should be disallowed, since the first merger should discharge whatever features  $\alpha$  needed from  $\beta$  and vice versa
- The “First Merge” problem:



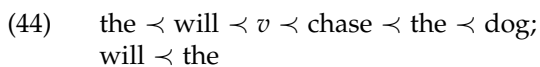
- One possibility – null structure that forces c-command? Forced movement to break up symmetric c-command (Moro 2000)? Treat foot of the tree as a single phonological word?
- How is the following tree constructed using Merge? How is it linearized?:



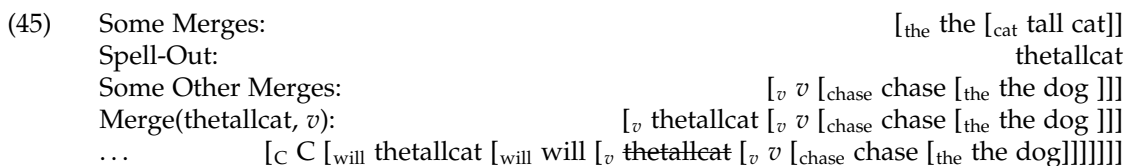
- Issue one: We seem to need to have two “work-spaces” to handle the left-branching structure:



- If we have two “work-spaces”, how do we understand the Extension Condition? In a later Homework, you will explore the implications of distinct work-spaces, and examine Nunes’s (2001) “sideways movement” approach to Parasitic Gaps and ATB movement, which relies on movement between workspaces before Merging them together
- This doesn’t solve the linearization problem – the T’ and subject DP mutually c-command one another, implying that they have to precede each other!!



- One possibility: We Spell-Out the left branching material before Merging it, then Merge it as a single word (“Multiple Spell-Out”, Uriagereka 1999)



(46) the tall cat < will < chase < the < dog

- On the copy theory of movement, how do we distinguish these two representations?

- (47) a. \*Mary<sub>i</sub> likes t<sub>i</sub>  
Spell-Out: Mary likes Mary  
PF: Mary likes Mary
- b. Mary<sub>1</sub> likes Mary<sub>2</sub>  
Spell-Out: Mary likes Mary  
PF: Mary likes Mary

- We need some way to distinguish copies of the same phrase and different instances of the same lexical item.

## Bibliography

- Abels, Klaus. 2003. *Successive Cyclicity, Anti-Locality, and Adposition Stranding*. PhD Thesis, UConn.
- Baker, C.L. 1979. Syntactic Theory and the Projection Problem. *Syntactic Theory and the Projection Problem* 10(4), 533–581.
- Bieberauer, Theresa, Anders Holmberg, & Ian Roberts. 2007. Disharmonic word-order systems and the Final-Over-Final-Constraint (FOFC). In A. Bisetto & F. Barbieri (eds.), *The Proceedings of the XXXIII Incontro di Grammatica Generativa*, 86–105.
- Chomsky, Noam. 1970. Remarks on nominalization. In R. Jacobs & P. Rosenbaum (eds.), *Readings in English Transformational Grammar*, 184–221. Waltham: Ginn.
- Chomsky, Noam. 1977. On WH-Movement. In P. Culicover, A. Akmajian, & T. Wasow (eds.), *Formal Syntax*, 71–133. New York: Academic Press.
- Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge: MIT Press.
- Chomsky, Noam. 2001. Beyond Explanatory Adequacy. *MIT Occasional Papers in Linguistics* 20. Cambridge: MITWPL.
- Emonds, Joseph E. 1970. *Root and Structure-Preserving Transformations*. PhD Thesis, MIT.
- Felser, Claudia. 2004. Wh-copying, phases, and successive cyclicity. *Lingua* 114, 543–574.
- Grohmann, Kleanthes. 2003. *Prolific Domains. On the Anti-Locality of Movement Dependencies*. Amsterdam: John Benjamins.
- Kayne, Richard. 1994. *The Antisymmetry of Syntax*. Cambridge: MIT.
- Kratzer, Angelika. 1996. Severing the External Argument from the Verb. In J. Rooryck & L. Zaring (eds.), *Phrase Structure and the Lexicon*, 109–138. Dordrecht: Kluwer.
- Lidz, Jeffrey. 2003. Causation and reflexivity in Kannada. In V. Dayal & A. Mahajan (eds.), *Clause Structure in South Asian Languages*, 93–130. Dordrecht: Kluwer.
- Moro, Andrea. 2000. *Dynamic Antisymmetry*. Cambridge: MIT Press.
- Nunes, Jairo. 2001. Sideward Movement, *Linguistic Inquiry* 31(2), 303–344.
- Uriagereka, Juan. 1999. Multiple Spell-Out. In S.D. Epstein & N. Hornstein (eds.), *Working Minimalism*, 251–282. Cambridge: MIT Press.