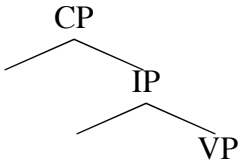
The enterprise (reminder):

- Language faculty, part of being human (genetically specific).
- Comprises a system which highly restricts the range of possible human languages.
- This system underlies all human languages, we seek to discover its properties.
  - —Looking *in depth* at the syntactic properties of individual languages
  - —Comparing properties across languages to see what the options are
- Once we see patterns, we
  - —state generalizations
  - —form a hypothesis (what causes the generalization?)
  - —check other predictions of the hypothesis
  - —revise as needed
- As the theoretical hypotheses develop, we see new things to check that we wouldn't have thought to check before. In the process, whether the hypotheses are right or wrong, we have *learned* a lot about how language works.
- **We assume there is a fact of the matter, that the Truth is Out There.**
  We are just trying to figure out what it is.
- Lots of proposals exist, some probably closer to the Truth than others.
- To evaluate them, we must see what they predict:
  - —Do they predict that all of the possible languages/constructions exist?
    English sentences must be SVO.
    I left. *Taxes* I can deal with, it's the *forms* that bother me.
  - —Do they predict that all of the impossible languages/constructions don't exist?
    English sentences may have any word order at all.
    *Like I pizza. *Bill thinks Mary that left.
- When two hypotheses both seem to predict the data, we get into aesthetics.
  - —Which hypotheses are "simpler", more "elegant"?
  - —Which hypotheses require the fewest stipulations, assumptions?
  - —Which hypotheses make the greatest range of predictions?
  - —Which hypotheses can account for the limited *crosslinguistic* options?

(1) 

The clause structure of "the olden days" (say, 1986):

CP: force (declarative, interrogative)
IP: functional morphology (tense, agreement)
VP: lexical/thematic elements

Spoiler, for Advanced Syntax: But this doesn't seem to be quite good enough to describe/explain the data—progressively, they have been split up into further functional projections.

Syntactic argumentation:

- Present the puzzle (recalcitrant data)
- Present a hypothesis to account for the data.
- **Consider what *else* your hypothesis *predicts***.
- **Check to see if the predictions are met**.
  —if so, celebrate them; if not, speculate as to why not (or revise your hypothesis).

Answers are nice. But we have to know how much to trust the answers, what assumptions those answers depended on. If we change our way of thinking about some other phenomenon, does it undercut assumptions that the answer relied on?

So, as much as possible, we want to focus our attention on the arguments (and the structure of the arguments) in the papers we are reading.

A little bit of history

This kind of generative grammar has progressed through several stages, sometimes names.

Aspects
Standard Theory
Extended Standard Theory
Revised, Extended Standard Theory
Government & Binding (GB)
—*LGB* (1981)
—*Barriers* (1986)
(Kayne 1994: Antisymmetry)

Minimalist Program (MP) (sort of)
—Minimalist Program for Linguistic Theory (1993)
—Chapter Four (1995)
—Minimalist Inquiries: The Framework (1998)
—Derivation by Phase (1999)
—Beyond Explanatory Adequacy (2001)
—On Phases (2004), etc.

There have also been offshoots along the way (again, sort of). Some you might hear of:
Lexical Functional Grammar (LFG)
Head-driven Phrase Structure Grammar (HPSG)
Relational Grammar (RG) (*promotion, advancement*)

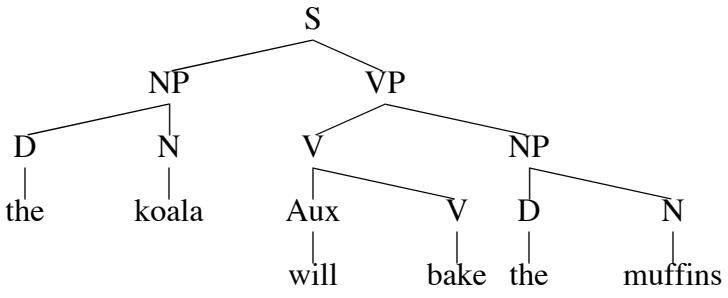
Early generative tradition

Basically, encoding the observations that sentences are *made up of* certain things. We might say: A sentence (of English) has a subject and a predicate. Subjects can be simple (*John*) or complicated (*the rumor that John asked the manager to write a comprehensive report showing opportunities for downsizing*); predicates come in various types as well (*left, surprised me, gave the union representative a headache*).

Properties of phrase structure were encoded in *phrase structure rules* ("rewrite rules").

(2) $S \rightarrow NP\ VP$
$VP \rightarrow Verb\ NP$
$NP \rightarrow D\ N$
$Verb \rightarrow Aux\ V$
$D \rightarrow$ the, a
$Aux \rightarrow$ will, PAST
$N \rightarrow$ koala, muffins
$V \rightarrow$ bake, eat

(3)



(4)

a.	S		g.	the N Aux V D N
b.	NP VP		h.	the koala Aux V D N
c.	D N VP		i.	the koala will V D N
d.	D N Verb NP		j.	the koala will bake D N
e.	D N Aux V NP		k.	the koala will bake the
f.	D N Aux V D N		l.	the koala will bake the muffins

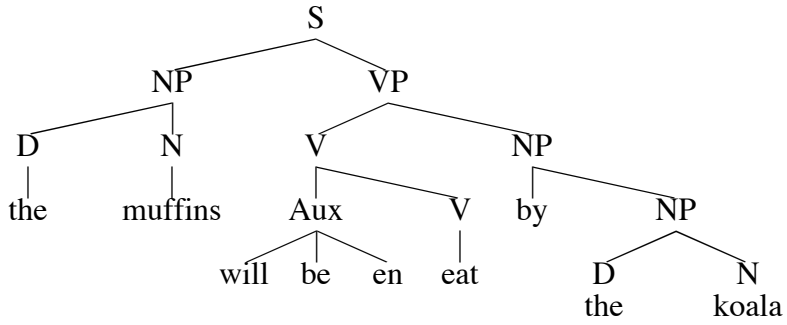
The tree structures indicate rule applications. The goal is to come up with a complete set, which is known to speakers of a language (e.g., English), that will be able to *generate* all of the sentences a native speaker considers to be English, and will not be able to generate any sentences a native speaker does not consider to be English.

Some sentences seem to be related to other sentences, and the idea is that, along with the PS rules that create the Deep Structure (DS), there are *transformations* that can then apply, that reorganize the tree is particular ways. For example, the formation of the passive, or of *wh*-questions, etc.—this is *movement*.

(5)
a. The koala will eat the muffins.
b. The muffins will be eaten by the koala.

(6) ***Passive transformation***
Structural description: $NP_1 – Aux – V – NP_2$
Structural change: $NP_2 – Aux – be+en – V – by – NP_1$

(7)



(8) Aux → (Tns) (Modal) (have+en) (be+ing)

(9) The muffins must have been being eaten by the koala

(10) The muffins must have en be ing be en eat by the koala

(11) ***Affix hopping transformation***
SD: Affix – V
SC: V – Affix

The basic structure of the system, according to this view:

(12) Phrase structure rules
↓
Deep Structure → Semantic interpretation
↓
Transformations
↓
Surface Structure → Phonetic interpretation

Gets: captures hierarchy, recursion, constituency.
But: Language seems to be relatively constrained: Lots of PS rules could exist, but don't.
Across languages, there is a high degree of similarity.
The system is **missing a generalization** in many cases.
What are the possible phrase structure rules? What are the possible transformations?

Also note/remember: What we are after (at least initially) is a characterization of the possible sentences of a language, their pronunciation and their meaning. The grammar is a way to map pronunciation and meaning to one another. This early view above is *quite* generative, pronunciation is basically derived from the meaning. This changes later.

The possible PS rules: X′ theory

It turns out that there are a number of generalizations about PS rules. For example, no rule looks like this, where a VP has no V within it:

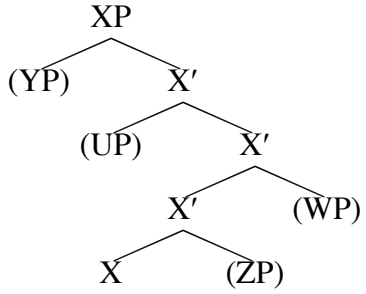
(13) VP → D N

X′-theory is the statement of possible PS rules:

(14) XP → (YP) X′
X′ → (UP) X′ (WP)
X′ → X (ZP)

"X" here is an abstract category. So the rules above would apply to VP, say, or NP. It says that all phrases have a "bar" node of the same category, which dominates a head of the same category. The phrase might have a daughter (other than the "bar" node), which is a phrase (the "specifier"). The head might have a sister that is a phrase (the "complement"). And the bar node itself might have another bar node as its daughter, with a phrasal sister.

(15)



***Note*:**
This is more accurate for 1980s-era X-bar theory; mostly, we will here assume that adjunction is to the XP level, not to the X-bar level.

Another thing that X′-theory sought to capture is certain parallelisms that seem to exist across categories: in particular with respect to gerundive nominals and derived nominals (*John's refusing the offer, John's refusal of the offer*).

Casting the sentence (S) in terms of X′-structure was a further development, where it was hypothesized that the auxiliary/tense was really the head of the sentence; hence, S became IP. (Inflection Phrase). And, then the sentence introducing complementizer was brought in as well, to form the CP, making X′ structure a fully general template for the construction of sentence structures

Deriving X′-theory

Although we'd made progress by generalizing from PS rules to X′-theory (in the process explaining why certain kinds of PS rules don't exist for natural language), the X′ schema itself is still a stipulation. And we might wonder if the *schema* could be explained.

There were a few attempts at this: The most currently prominent ones are the "bare phrase structure" of the Minimalist Program (basically what we'll be exploring here) and the "Antisymmetry" approach to phrase structure, which will get some discussion in Advanced Syntax. Both sought to posit even more general principles from which the properties stipulated under X′ theory arise. (And, perhaps, we're actually better off empirically without those that properties that were stipulated but not derived.)

Government & Binding (GB)

Government & Binding (GB) theory was a break from previous theories in its movement toward more general principles, rejecting specific rules. Asking the question: *Why* is it that way? (In general, in fact, this is how the theories move from one stage to the next.)

The basic shape of the theory is that it consists of a number of somewhat independent modules that govern the well-formedness of trees.

The idea is: You can do whatever you want, so long as you satisfy the requirements imposed by the modules, such as these:

θ-theory: Each θ-role must be assigned to an argument (chain),
and each argument (chain) must get a θ-role.
Case theory: Each NP (chain) must get Case.
X′-theory: Structures must conform to the X′-schema.
Binding theory: Indices must conform to Principles A, B, C.
Bounding theory: A moved element cannot be too far from its trace.
Control theory: Things dealing with PRO.

The Case Filter: *NP if NP has phonetic content and has no Case.

This idea was not hit upon right away, particularly because in English, you don't see a lot of evidence for case on NPs. However, if we suppose that NPs need Case (and in other languages it can even be seen), and that Case is only available to NPs in certain places, then we can derive quite a bit about syntax.

(16) a. Subject of a tensed clause: nominative.
b. Object of a verb: accusative.
c. Object of a preposition: accusative.

So, it was thought that V, P, and finite I *assign* case to NPs.

But there are restrictions: The reach does not seem to be very far:

(17) a. Pete hit me.
b. *Pete hit savagely me.

It seems as if V can only assign case to an adjacent NP. Also, it seems to be possible either to assign case to one's complement (as V and P do) or to one's specifier (as I does).

Plus, there are the cases where V or C seem to be able to reach in to the sister and assign case to the NP in the sister's specifier (the Exceptional Case Marking cases):

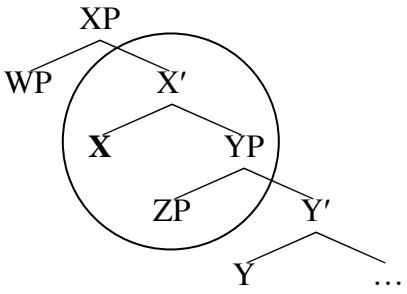
(18) a. Hank considered me to have lost the battle.
b. For me to get there on time, I will have to leave now.

So, what is the requirement on the configuration of a case-assigner and the case-recipient?

We say that the case-assigner *governs* the case-assignee. This relation of *government* holds between the head and all the positions to which it could assign case.

A head *governs*, essentially, its complement, the specifier of its complement (and sometimes its specifier). Heads assign Case to positions they govern. The properties of government differed depending on whether a head is lexical (N, V) or functional (I, C, D).

(19)



John met him
John wants him to leave.

This still leaves out the assignment of case to the specifier of IP, however. Perhaps government should be extended to WP as well… and so forth: there was a lot of effort expended by the community in detailing the precise characterization of government, and where it is used.

The way the whole GB system worked is diagrammed below.
Each of DS, SS, and LF is a syntactic structure, a tree.
You can freely form such trees, **but**, there are a number of conditions that you need to obey at the end. Each level could have conditions, so:

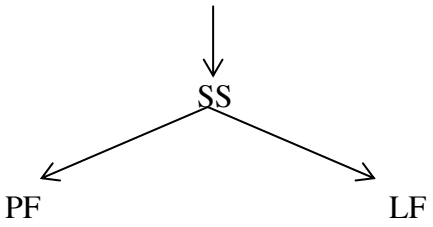
X′-theory must be satisfied by all trees.
Movement relations must satisfy Subjacency (locality, short movement).

DS: θ-theory: The θ-criterion must be satisfied.

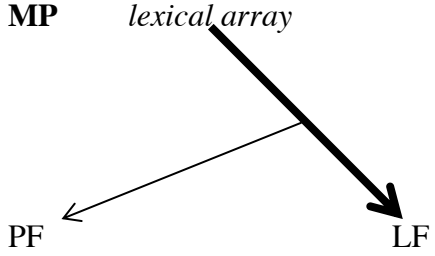
SS: Case theory: A DP must have its Case checked
Binding theory: Anaphors must be bound locally, Pronouns must be free locally, r-expressions must be free.

LF: Quantifiers must bind a variable, wh-words must move, etc.

(20) **GB**



(21) **MP**



The **government** relation was important in **licensing**—Case licensors must govern DPs that they license, traces must be governed by lexical heads, etc. Conditions could be imposed at each of the different **levels** (DS, SS, LF).

The idea was that first DS is constructed to satisfy the constraints on DS, then things moved around (Move-α), resulting in SS, which must satisfy the constraints on SS. And then possibly further movement (covert movement, morphology) occurs to create LF, PF.

Covert movement

In the structure of the system (in GB, MPLT), there is a point at which the pronunciation is (at least partially) determined. But now there is a branch of the derivation between SS and LF, unlike in the early generative version. The tree that feeds meaning can still be adjusted (in some sense) *after* the pronunciation is determined at the split. (And, incidentally, so can the pronunciation in certain ways—cf. "Distributed Morphology.")

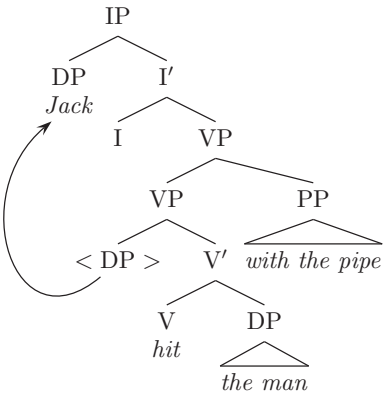
Why might people have thought something like this?

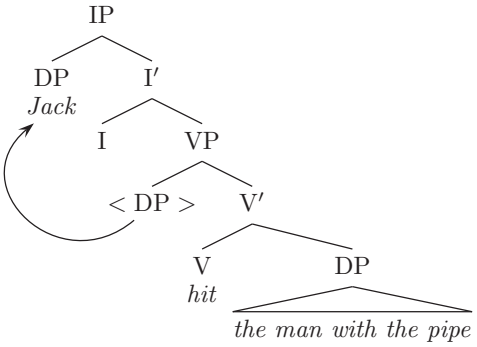
The LF structure is the tree that is supposed to feed interpretation.

Certain sentences are ambiguous:

(22) Jack hit the man with the pipe.

We think of this ambiguity as a structural ambiguity—either *with the pipe* is attached to *the man* or it is attached to the VP as an instrument. What makes the sentence ambiguous is that you can't hear the difference between these two attachment sites. The LF structure (and the DS structure and the SS structure) differs between the two interpretations the sentence can have.





The following sentence is also ambiguous.

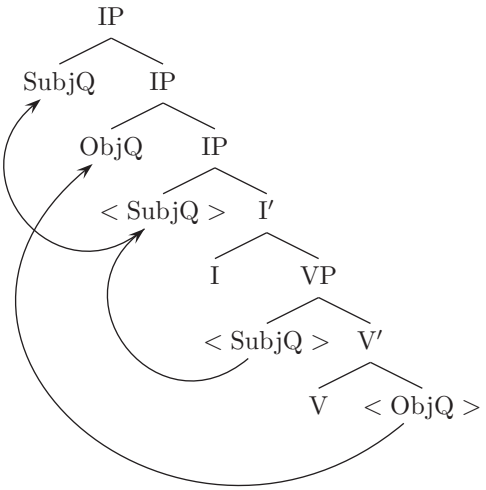
(23) Someone bought everything.

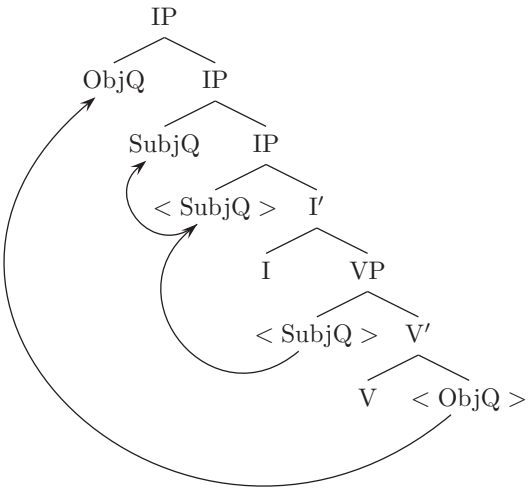
People hypothesized that perhaps this is a similar phenomenon. Specifically, that the two different meanings correspond to two different syntactic (LF) structures, allowing the "logical form" of the meaning to be transparently derived from the structure.

The two interpretations are something like:

(24) There is someone x such that for every thing y, x bought y.
(25) For every thing y, there is someone x such that x bought y.

In order to make the syntactic structure as "transparent" as possible, we might assume structures like the following, where the traces of movement represent the variables (*x* and *y*) and the moved quantifiers reflect the "There is someone" and "For every thing" part.





But of course this "movement" that is hypothesized to make the LF structure conform to the logical form is not reflected in the pronunciation. The object in both cases is pronounced after the verb. In the GB/MPLT model,

this could happen if the movement happens between SS ("Spellout") and LF. Since either of the movements illustrated are possible and lead to different interpretations, but the pronunciation is the same, the sentence is perceived to be ambiguous.

This movement of quantifiers is known as Quantifier Raising. (We'll get to this later.)

At this point, we're sort of out on a limb—it is pretty far out to suppose that there is "invisible movement." It's not the kind of thing one should accept lightly. So, the next task is to see whether one can come up with an argument that it is actually happening (rather than, as we've done so far, relying on conceptual arguments about what is a better way for syntax and semantics to work—i.e., by supposing LF is an unambiguous representation off of which the semantics/logical form can be read directly).

Among the more famous arguments for Quantifier Raising is this, from Antecedent Contained Deletion:

(26) Pat called everyone I did.

This relies on a phenomenon called VP ellipsis, which is a procedure that allows a VP to go unpronounced if it is identical to another VP in the sentence.

(27) Bob ate a cookie after I did. (after I [past] [$_{VP}$ eat a cookie])

So, to interpret that sentence, we have to "fill in the VP" in some way, perhaps by copying the antecedent VP in for the purpose of interpretation.

If we try that with (26), however, things would seem to go wrong. The meaning seems at least to be that what both Pat and I did is some calling. So, the missing verb is *call*, but the antecedent VP actually contains the elided VP.

(28) Pat [$_{VP}$ called everyone I did [$_{VP}$ Ø]]

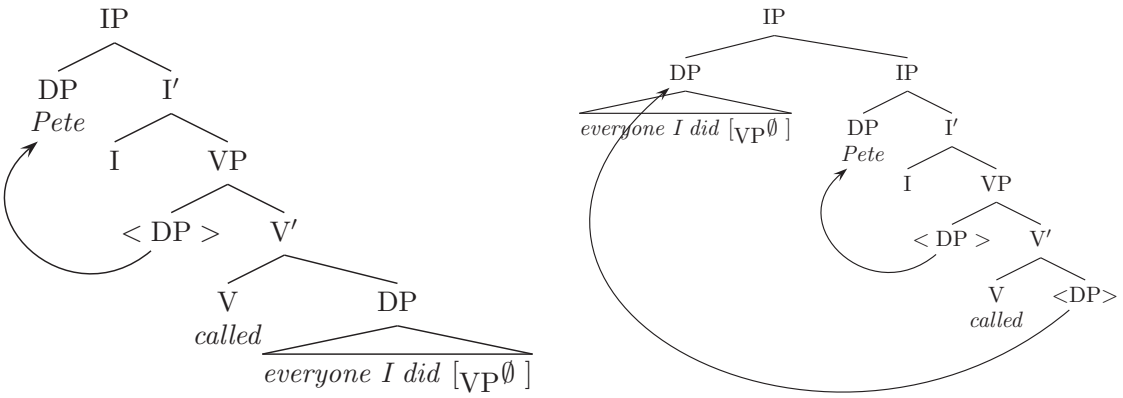
So we can't very well copy the antecedent VP into the empty VP. That would yield something like:

(29) Pat [$_{VP}$ called everyone I [$_{VP}$ called everyone I did [$_{VP}$ Ø]]]

And we haven't made any progress—we still have a missing VP. Yet, the sentence is fine.

However, if we consider that [everyone I did [$_{VP}$ Ø]] is actually a quantifier, and undergoes movement—*out* of the antecedent VP—before the copying needs to happen, we wind up with something like this:

[everyone I did [$_{VP}$ Ø]]$_i$ Pete [$_{VP}$ called t_i]



And that's perfect. We can copy [$_{VP}$ called *t*] in for [$_{VP}$ Ø] without an "infinite regress" problem. It means that everyone I called, Pete called.

Covert movement is also implicated in the language variation in *wh*-movement: We can think of languages that don't move *wh*-words as actually being languages that move them *covertly*, under the assumption that for the purposes of interpretation, all *wh*-words need to be up in SpecCP in the LF structure.

Sometimes "covert movement" is called "LF movement." (Also: in Minimalist terms, the idea of covert movement is sometimes abandoned in favor of the idea that the features that would have driven movement can be checked "long distance" via Agree without actually triggering a movement.)

Also: Reconstruction. Principle C.

Constraints on movement: Subjacency and the ECP

Very early in this approach to language as an object of scientific inquiry, it was recognized that certain movements seemed to be impossible. For example, moving a *wh*-word out of a relative clause:

(30) *What did you meet [the man [who bought at the store]]?

Ross (1967) called these "islands"—a constituent from which there is no escape.

Of course, we'd like to know why movement can't proceed out of these islands.

There are a couple of different kinds of constraints on movement that have come to be assumed. One is Subjacency, which in the days of GB had to do with movement being able to escape no more than one "barrier," where most XPs were barriers apart from specific exceptions. The other is the ECP (Empty Category Principle), which concentrated on the difference between subjects, objects, and adjuncts.

Subjacency has always worked more or less the same way, so the information from LX522 is still relevant to reading older literature that mentions Subjacency.

The ECP governs "empty categories" (most relevantly, traces), and the intuition behind it is that if something is going to be silent, it needs to be "identified." This was also taken to hold of things like *pro* (the null subject pronoun available in languages like Spanish and Italian—in those cases, the empty category is "identified" by the inflectional features of I).

The ECP says: An empty category must be properly governed.

Proper government is defined as satisfying one of two different conditions. An empty category can be properly governed if it receives a θ-role from a lexical category—so, the object of a verb would be properly governed. An empty category can also be properly governed by its antecedent (the place that the movement went next), so long as they are close enough together.

The ECP makes a distinction between traces of arguments (e.g., *what*) and adjuncts (e.g., *how*) because the first clause of the definition of proper government ensures that arguments are always properly governed, and are therefore always safe from violations of the ECP. Adjuncts, on the other hand, can only be properly governed if their antecedent is not too far away.

The effects of violating the ECP are generally worse than violating Subjacency:

(31) ??What did you wonder whether John bought t?
(32) *When did you wonder whether John bought cheese t?

Also: Coordinate Structure Constraint and the Across-the-Board exception.

The Minimalist Program

Another shift in perspective, toward deriving the principles from more basic "design properties" of the system.

C_{HL}, stuck between A–P (PF) and C–I (LF), which impose requirements.
Is human language more than an optimal solution to those requirements?
Is all of the complexity we find in syntax ultimately traceable to the *design requirements*?

Strong Minimalist Thesis:
(33) Language is an optimal solution to legibility conditions

Some starting points, assuming (33) and the idea that less machinery is better than more.

(34) a. The only linguistically significant levels are the interface levels.
b. The **interpretability** condition: Lexical items have no features other than those interpreted at the interface (properties of sound and meaning).
c. The **inclusiveness** condition: The machinery of syntax (C_{HL}) does not introduce any new features not already contained in the lexical items.

Features and the lexicon

UG makes available a set F of features (linguistic properties) and computational operations C_{HL} (the computational procedure for human language) that make use of F to form expressions. A particular language L

uses F to form a particular set of expressions {EXP}. A given EXP is a pairing of meaning (LF) and sound (PF), interface representations.

A language L takes those features and bundles them, forming a **lexicon**.

Bare phrase structure

(34c) says that C_{HL} can't add anything that wasn't already part of the lexical items. No new features.

This implies that we can't (meaningfully) mark things as being an X-bar, or an XP, or a trace. We can't make syntactic use of indices (since they weren't part of the lexical item and because the syntax can't add anything). A phrase structure tree under this view must be simply lexical items, in combination: a ***bare phrase structure***.

The derivation starts with a ***numeration*** (or ***lexical array***), which is an array of lexical choices to be reorganized by the system. (Not a set, but close).

The computational system (recursively) generates **syntactic objects** from the **objects available in the numeration** and those **syntactic objects already formed**. We "rearrange" what we had in the numeration—putting things together, sometimes making copies, and that's it. Nothing added.

We start with lexical items. We need to end up with a single syntactic object. So, we need an operation to combines things. The simplest operation would be one that takes two existing objects and puts them together to form one—**Merge**.

What kind of syntactic object do we get after Merge? Must be made of things we already had in the lexicon (bundles of features). Merge(α, β) = {α, β}? No, verb phrases act like verb phrases and not noun phrases—**we need the combined object to have the properties of the head of the phrase**. We have to label the phrase to indicate what kind of phrase it is.

The syntactic object formed by Merge has a label γ and the two merged objects α and β: {γ, {α, β}}.

We can't add anything, so γ must be something based on either α or β; since it is marking which of α or β is the head, we can just say that γ is either α or β, depending on which one is the head of the complex (that is, depending on which one ***projects***).

combine *saw* and *Mary*: *saw*

{*saw*, { *saw*, *Mary* } } *saw* *Mary*

We then read X-bar-like relations off of a tree, just by looking at the context.

(35) will { will, { {the,{the,cradle}}, {will,{will,fall}}}}

the will (detail missing in "VP" however)

the cradle will fall

A **head** is a terminal element drawn from the lexicon/numeration. The **complement** is the thing merged with the head, the most local relation. The **specifiers** are the other relations. The **maximal projection** of α is a constituent for which α doesn't project the label.

What this means is that Merge *derives* much of what X-bar theory stipulated.

- binary branching (Merge combines two things)
- one head
- properties of the head determine the properties of the phrase
- complements and specifiers must be maximal projections

But there are some things left over:

- can only have one specifier?
- can only have one complement?
- must have three levels (head, intermediate, maximal)

Also notice that *cradle* is **both a head and a maximal projection**, all at once.

Question is: Were the extra parts of X-bar theory *really* doing us any good? Can we do without the parts of X-bar theory that don't come for free?

> *(for free … derives from Merge, which we an interpretable language needs anyway… LF only interprets single syntactic objects, lexicon provides a bunch of objects in the numeration, so we have to combine them…)*

Procedure: Start with (33), then seek to disprove it. Search for apparent imperfections— things that do not seem to arise from requirements of the interfaces.

For any apparent imperfection P, any of the following could be true:

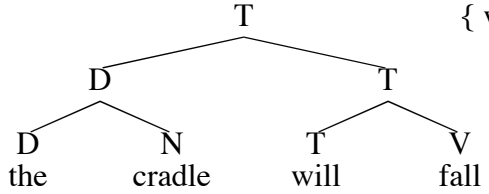
(36) a. P is a real imperfection. Therefore, (33) is false.
b. P is not real; the existence of P is only apparent, and (33) may still be true.
c. P is real, but not an imperfection—it is part of an optimal solution to the legibility conditions of the interfaces.

The goal is always (36b) or, better, (36c)—since (36c) give us insight into the legibility conditions themselves.

As you were

Now that we have a conceptual basis for phrase structure, we can go back to writing things pretty much how we did before, just with these issues in the back of our minds.

(37)

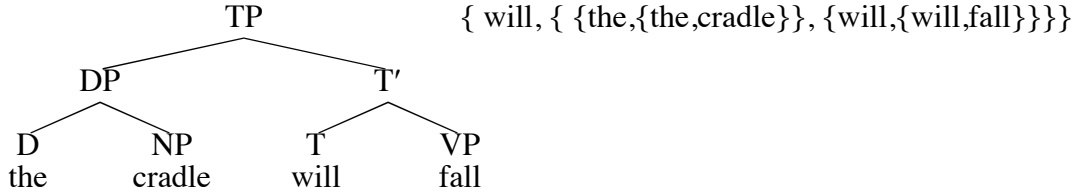


{ will, { {the,{the,cradle}}, {will,{will,fall}}}}

(detail still missing in "VP")

This is just labeling the nodes with the category feature (arguably one of the most important features of a lexical item) rather than spelling it all the way out.

We could even go back to writing in the bars and Ps, just with the understanding that they have no *meaning* for the syntax (and we don't write any "empty" things, so no non-branching nodes, etc.). People often do this, just for comfort.

(38) 

C-command and other relations

With BPS, we can define certain syntactic relations quite easily.

Sister of (other element in a Merge)
Contain (sub-elements of a syntactic object)

This also gives us ***c-command***, which has proven to be very important in syntax.

Contain (Sister of)

Movement

It is an irreducible fact about language that things *move* from the place where they're interpreted.

(39) Bill is likely to seem unhappy.
(40) What did Mary ask Sue to buy?
(41) The sandwich was eaten by Martin.

Putting aside ***why*** for a moment, how do we move things?

Prior syntactic research has given us reason to believe that the subject moves from SpecVP to SpecTP.

*DS

GB theory started by constructing a DS, the place where θ-theory needed to be satisfied.

But, DS isn't an interface level. We don't want to have a DS with special properties.
And in MP we don't any more.

There is *no more D-structure* because by the time we get to the "base positions" of elements in the higher part of the tree, stuff in the lower part of the tree has already moved around:

(42) a. (John said) that the students will all <the students> leave.

Therefore: There can't be any constraints that hold of DS—DS doesn't exist (and isn't an interface level). What was DS good for before? Can we do that in another way?

Least effort

When looked at closely, movement seems to have a *least effort* character. Often, you find that a movement will not happen if there was a shorter movement that could have accomplished the same thing.

Evidence for **Attract Closest**: Superiority in English, Raising in Icelandic.

(highest *wh*-word must move)

(43) a. Who$_i$ did John say t_i bought what?

b. * What$_i$ did John say **who** bought t_i?

(44) a. Jón telur [**mér**$_i$ virðast t_i Haraldur hafa gert þetta vel].
J.NOM believes **me.DAT** to.seem H.NOM to.have done this well
'Jon believes Harald to seem to me to have done this well.'

b. * Jón telur [**Harald**$_i$ virðast **mér** t_i hafa gert þetta vel].
J.NOM believes **H.ACC** to.seem **me.DAT** to.have done this well
'Jon believes Harald to seem to me to have done this well.'

c. Jón telur [**Harald**$_i$ virðast t_i hafa gert þetta vel].
J.NOM believes **H.ACC** to.seem to.have done this well
'Jon believes Harald to seem to have done this well.'

(45) a. * Have$_i$ John **will** t_i left by the time we get there?

b. * John$_i$ is likely for **it** to seem t_i to have left.

This suggests that the syntactic computation takes the *cheapest way out*. Longer moves are somehow more expensive, and thus avoided.

This has been variously known as **Shortest Move**, **Attract Closest**, **Minimal Link Condition**: Movement is as short as it can be.

Language variation and Spell-out

So far, there is nothing in the system that allows languages to differ except in the lexical items that come in from the numeration.

If we stick to the minimal necessary assumptions, that's the only option we have.

Language variation is restricted to the lexicon.
The computational procedure (Merge, Move, … C_{HL}) is the same for all languages.

Another thing we haven't considered: Lexical items have phonological features too ("starts with *p*") but these are assumed to be uninterpretable at LF. So they must be removed.

Phonological features are not checked. Instead, they are eliminated all at once by an operation called **Spell-out**. Spell-out removes the phonological features and sends them off to the PF interface. It's similar to the breakpoint in the old-style system called SS.

When do we do Spell-out?

The least effort intuition (combined with forethought) suggested that **Spell-out happens as early as possible**, under the assumption that doing movement you can *hear* is more work than doing movement that you can't. This is sometimes referred to as **Procrastinate**.

The computational system would prefer to send everything off to phonology immediately.
Why doesn't it?

One reason is that Spell-out is the *only* way to eliminate phonological features. **So, you can't Spell-out before you've added all of the lexical items to your tree** (except maybe for special lexical items that have no phonological features) because otherwise you would have no way to get rid of them. The tree would arrive at LF with phonological features and the derivation would **crash**.

Certain uninterpretable features are special. They are **strong**. A strong feature must be eliminated prior to Spell-out. Which features are strong may vary from language to language.

The EPP is a strong uninterpretable [D] feature on T. The [D] feature on T must be removed before Spell-out.

So, Spell-out happens as soon as it can, but that's going to be only after all of the strong features are eliminated.

In French, the verb raises overtly to T. In English it doesn't.
This can be characterized in this system by saying that French has a **strong** V feature on T. In English, assuming there is also an uninterpretable V feature on T, it is **weak** and so V doesn't move to T until after Spell-out (when we can't hear it).

(An alternative way to look at this—closer to what we did in Syntax I—is to suppose that strong features drive movement and weak features simply agree at a distance without driving movement. The difference between these two approaches is difficult to distinguish.)

Phases

An even more recent development is the idea that the derivation proceeds in stages (**phases**), with Spell-out occurring several times.

The generally accepted "phase heads" are *v* and C and probably D. So, the idea is that when a derivation reaches CP, it spells out the contents of CP (the complement of C) and the spelled out structure is no longer "visible" to the rest of the computation.

Among other things, this means that is something is going to move out of CP, it is going to have to get to the edge of CP (e.g., SpecCP) before the CP is spelled out, or else it will be "frozen" in the spelled-out CP.