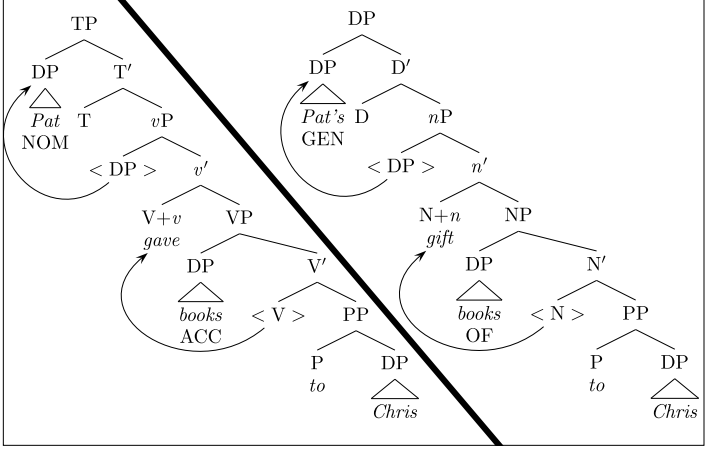
CAS LX 522
Syntax I

Finishing little *n*,
back to *do*-support

14

Little n



DP is like TP

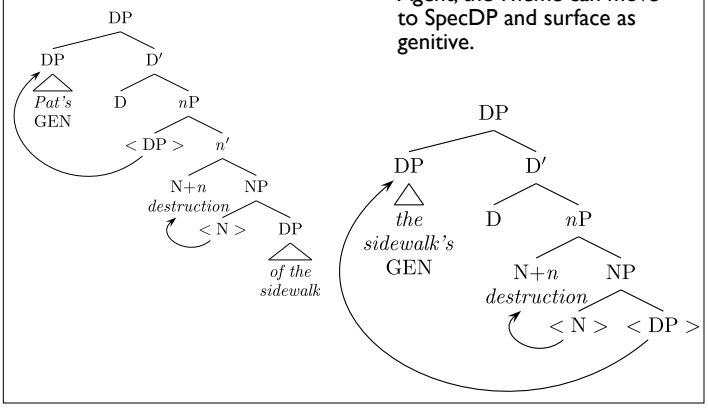
If we suppose that DP works like TP, we can extend our theoretical machinery in an exactly analogous way.

Hierarchy of Projections
D > *n* > N

UTAH
DP daughter of *n*P: Agent
DP daughter of NP: Theme
PP daughter of N′: Goal

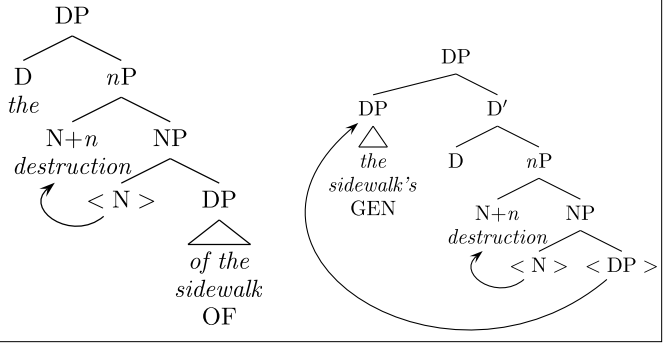
Passive nouns

Very similar to the passive, if an *n* doesn't introduce an Agent, the Theme can move to SpecDP and surface as genitive.

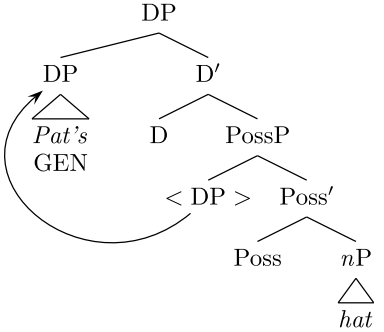


Passive nouns

If the DP has a head D like *the* that does not check genitive case, then there can be no Agent (nothing could check its case), and the Theme stays unmoved (its *of*-case checked by *n*).



Possessors



Adger proposes that Possessors are introduced by a new head, Poss.

HoP:
D > (Poss) > *n* > N

Hungarian possessors

1) Az en kalapom
   the I hat
   'my hat'
2) A Mari kalapja
   the Mary hat
   'Mary's hat'
3) A te kalapod
   the you hat
   'your hat'
4) Marinak a kalapja
   Mary the hat
   'Mary's hat'

Assuming that the DP in Hungarian has the basic structure we've been discussing, what is the structure of this kind of possessive construction?

How about that (person?) agreement on 'hat'?

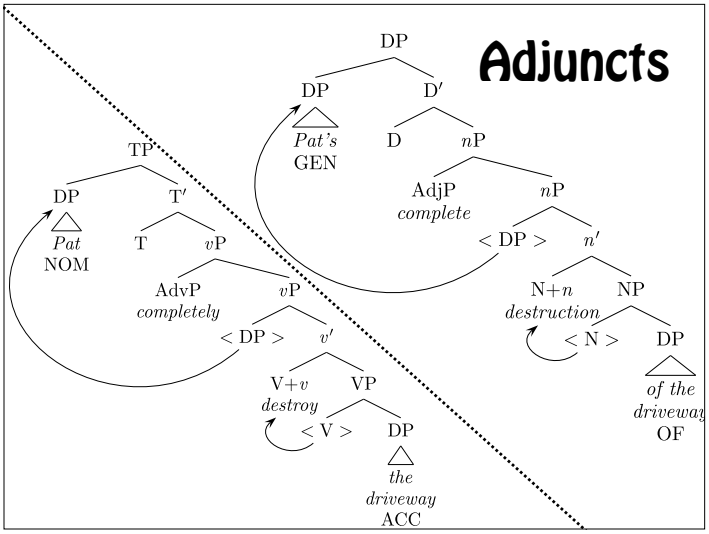
Adjectives

Adjectives are to nouns as adverbs are to verbs. So what would the structure be for *Pat's complete destruction of the sidewalk*? Or *the silly idea*? Or *the pencil on the desk*?

In *Pat completely destroyed the sidewalk*, we adjoin *completely* to *v*P. The subject moves to SpecTP.

In the same way, we adjoin *complete* to *n*P, and *Pat* moves to SpecDP.

Adjuncts



The Italian DP

In Italian, in many cases, there is simply an option (stylistically governed) as to whether you say *The Gianni* or just *Gianni*:

Gianni mi ha telefonato.
Gianni me has telephoned
'Gianni called me up.'

Il Gianni mi ha telefonato.
the Gianni me has telephoned
'Gianni called me up.'

The Italian DP

However, there is a difference with respect to the order of adjectives and the noun depending on which one you use.

1) L' antica Roma
   the ancient Rome
   'Ancient Rome'
2) *Antica Roma
   ancient Rome
3) Roma antica
   Rome ancient
4) E'venuto il vecchio Cameresi.
   came the older Cameresi
5) *E'venuto vecchio Cameresi
   came older Cameresi
6) E'venuto Camersi vecchio.
   came Cameresi older

Generalization: If there's a determiner, the noun follows the adjective. If there isn't the noun precedes the adjective.

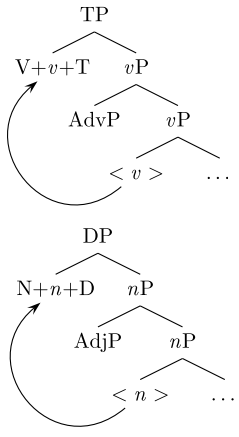
The Italian DP

- We can apply the same analysis to the order nouns and adjectives as we did to the order of adverbs and verbs.
  - Recall that in French, verbs precede adverbs, but in English, verbs follow adverbs. We conclude that in French, *v* moves to T.

In Italian, when the noun precedes the adjective it has moved over it, to D. The generalization is that this happens except if D is already filled.

- L' antica Roma
  the ancient Rome
- Roma antica
  Rome ancient

*Antica Roma
ancient Rome



Parameters

Languages differ on whether *n* moves to D, yielding some languages where nouns precede adjectives, and some languages where nouns follow adjectives.

- Likewise, languages differ on whether *v* moves to T, yielding some languages (e.g., French) where verbs precede adverbs, and some languages (e.g., English) where verbs follow adverbs.

What governs whether *n* moves to D is the strength of an uninterpretable feature checked on D or *n* by the other. One such feature is [*u*num:].

Italian: [*u*num:*] is strong on null determiners.

English: [*u*num:] is weak, even on null determiners.

- [$\varnothing_{\textbf{indef}}$ Happy students] poured forth from the classroom.

do-support

In French, verbs move to T. In English, they *don't* move to T.

That's because in French, when [tense:past] values [*u*Infl:] on *v*, it is strong, and in English, it is weak.

What this *doesn't* explain is why *do* appears sometimes in English, seemingly doing nothing but carrying the tense (and subject agreement).

- The environments are complicated:
  1) Tom **did** not **commit** the crime.
  2) Tom did not commit the crime, but someone **did**.
  3) Zoe and Danny vowed to prove Tom innocent, and prove Tom innocent they **did**.
  4) Tom (has) never **committed** that crime.

do-support

- The environments are complicated:
  1) Tom **did** not **commit** the crime.
  2) Tom did not commit the crime, but someone **did**.
  3) Zoe and Danny vowed to prove Tom innocent, and prove Tom innocent they **did**.
  4) Tom (has) never **committed** that crime.

When *not* separates T and *v*, *do* appears in T to carry the tense morphology.

When T is stranded due to VP ellipsis or VP fronting, *do* appears in T to carry the tense morphology.

When *never* (or any adverb) separates T and *v*, tense morphology appears on the verb (*v*).

So, *do* appears when T is separated from the verb, but adverbs like *never* aren't "visible", they aren't in the way.

Technical difficulties

How do we generally know to pronounce V+*v* as a past tense verb?

T values the [*u*Infl:] feature of *v*. The presumption is that *eat* +*v*[*u*Infl:past] sounds like "ate." And T doesn't sound like anything.

But this happens whether or not *v* is right next to T. *v* still has a [*u*Infl:] feature that has to be checked.

So, the questions are, how do we:

Keep from pronouncing the verb based on *v*'s [*u*Infl:] feature if T isn't right next to it?

Keep from pronouncing *do* at T if *v* is right next to it?

We need to connect T and *v* somehow.

Technical difficulties

The connection between T and *v* is that (when there are no auxiliaries), T values the [*u*Infl:] feature of *v*.

This sets up a relationship between the two heads.

Adger calls this relationship a **chain**.

We want to ensure that tense features are pronounced in exactly one place in this chain.

If the ends of the chain are not close enough together, tense is pronounced on T (as *do*). If they are close enough together, tense is pronounced on *v*+V.

Technical difficulties

Let's be creative: Suppose that the tense features on *v* (the value of the [*u*Infl:] feature) "refer back" to the tense features on T.

Agree can see relatively far (so T can value the [*u*Infl:] feature of *v*, even if it has to look past negation).

But "referring back" is more limited, basically only available to features that are sisters. Negation will get in the way for this.

So if you try to pronounce tense on *v* but T is too far away, the back-reference fails, and *v* is pronounced as a bare verb. But the tense features have to be pronounced somewhere, so they're pronounced on T (as *do*).

PTR

Adger's proposal:

> **Pronouncing Tense Rule (PTR)**
> In a chain (T[tense], *v*[*u*Infl:tense]), pronounce the tense features on *v* only if *v* is the head of T's sister.

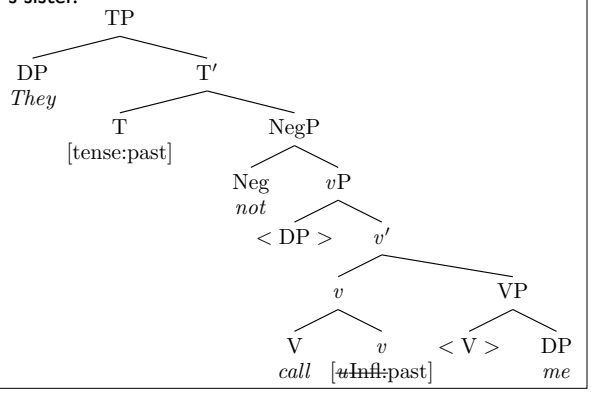
NegP, if there, will be the sister of T (HoP), but Neg has no [*u*Infl:] feature. *do* will be inserted.

Adverbs adjoin to *v*P, resulting in a *v*P. *v* has a [*u*Infl:] valued by T and adverbs don't get in the way of *v*P being the sister of T. Tense is pronounced on the verb (*v*).

If *v*P is gone altogether, *do* is inserted.

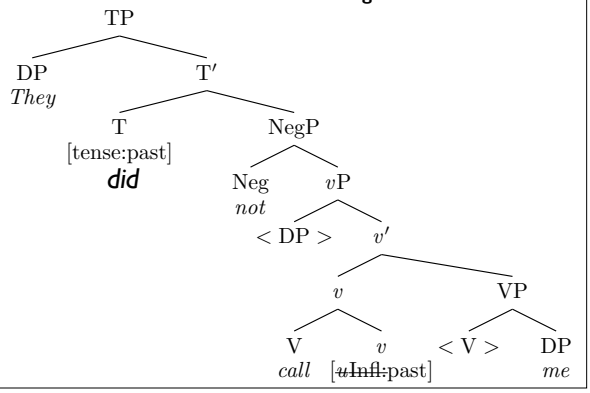
They did not call me

So, here, T and *v* form a chain because [tense:past] valued [~~*u*Infl~~:past]. But *v* is not the head of T's sister.



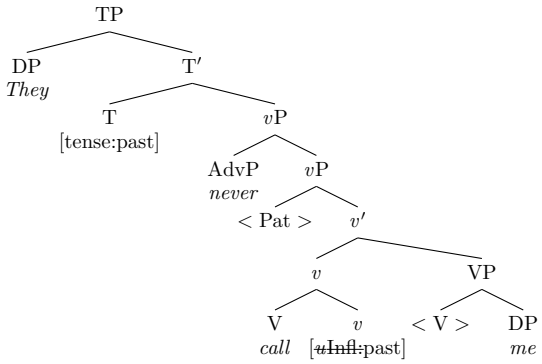
They did not call me

Do-support comes to the rescue. What this means is just that T is **pronounced** as *do* with the tense specifications on T. According to PTR, we don't pronounce them on *v*. **The tree doesn't change**.



They never called me

If there is an adverb like *never*, PTR still allows tense to be pronounced on *v* (so T doesn't have any pronunciation of its own at all).

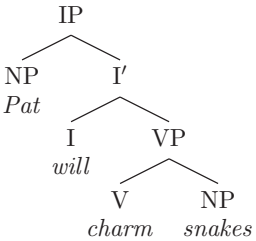


Historical interlude

Back in the days of yore, people hypothesized that *Pat will charm snakes* had a structure like this.

> The subject NP *Pat* was in the specifier of "IP" (what we call "TP"), and the VP contained only the verb *charm* and the object NP *snakes*.

Pat got an Agent θ-role by being in SpecIP, even though the fact that there is an Agent θ-role to be had is determined by the verb down in the VP.



The students will all...

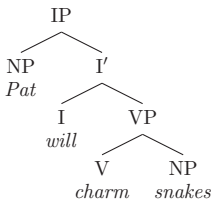
This predicts the normal word order pretty well, and so it was hypothesized that the verb simply assigned one of its θ-roles directly to SpecIP.

- No big deal, syntax works in strange and mysterious ways.

At a certain point, someone started thinking about sentences like these:

1) All the students will take the exam.
2) The students will all take the exam.

- It's fairly clear here that *all the students* is an NP, that it forms a coherent unit, a coherent concept. *All* really belongs with *the students*.



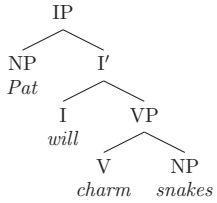
Floating quantifiers

1) All the students will take the exam.
2) The students will all take the exam.

Back in the even older days, the hypothesis was that there was a special rule that turned the first sentence into the second.

The **Quantifier Float** rule would move *all* over to the right, next to the VP.

- *all* NP … VP → NP … *all* + VP



Only some quantifiers float

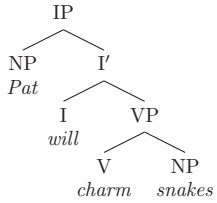
Quantifiers: *every*, *some*, *all*, *most*, *several*, *many*, *both*, *four*, …

1) Every student will take the exam.
2) *Student will every take the exam.
3) Several students will take the exam.
4) *Students will several take the exam.

- It works for *both* and *all*:

5) The students will both take the exam.
6) The students will all take the exam.

- What's a difference between *every*, *some*, *several*, *many*, and *both*, *all*?

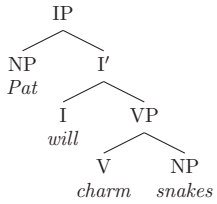


Leaving all behind

Upon further reflection, some enterprising syntacticians hit upon the idea that rather than floating *all* to its position next to VP, *all* might instead have been "left behind" by a subject that had moved.

- will [all [the students]] take the exam.
- [all [the students]]$_i$ will t_i take the exam.
- [the students]$_i$ will [all t_i] take the exam.

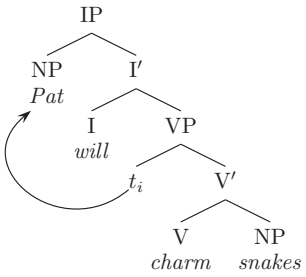
- And why would all the students have been down there? Well, that would simplify assignment of θ-roles.



The VP-Internal Subject Hypothesis

- The verb (head of VP) can assign θ-roles to other things within the VP, which is a natural explanation for how the choice of verb controls whether an Agent θ-role is assigned or not.

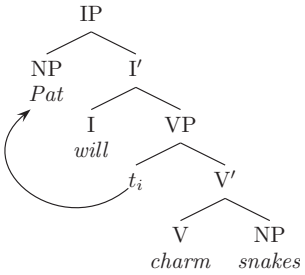
This idea became known as the **VP-Internal Subject Hypothesis**.



The VP-Internal Subject Hypothesis

For us, we've supposed from the beginning that assignment of θ-roles is necessarily local. This may not seem like a very surprising hypothesis.

But it was at the time a rather unintuitive idea, and so various people set out to see if some of the predictions this makes are borne out in the grammatical data.



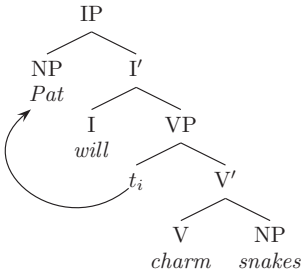
The VP-Internal Subject Hypothesis

- It turns out that as people looked, there were reasons to believe this.

The new analysis of Quantifier Float no longer relies on an idiosyncratic rule of English, but more general principles.

The assignment of θ-roles can now be more directly related to the properties of the verb.

And we can make sense of *there* constructions in a more straightforward way.



Back to the present

The basic components of the quantifier "stranding" phenomenon are:

All the students is a constituent.
The students is a DP inside *all the students*.

- [all [$_{DP}$ the students]]

Either *all the students* or just *the students* can move to SpecTP, to satisfy the [*u*D*] feature of T.

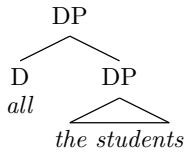
So *all the students* and *the students* are both DPs.

- [$_{DP}$ all [$_{DP}$ the students]]

So *all* is essentially a D, but one that takes a DP complement (*all*: [D, *u*D*, …]).

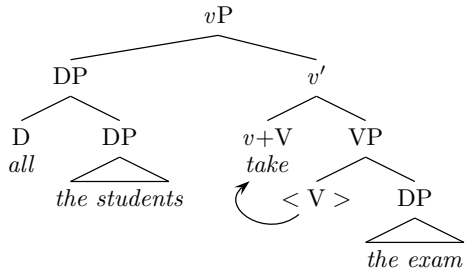
- We're assuming here that *all* is not an adjunct, but in fact a head, taking the NP as a complement. Why?

Quantifier stranding is still often referred to as "quantifier float" to this day, even though the name no longer reflects the analysis.



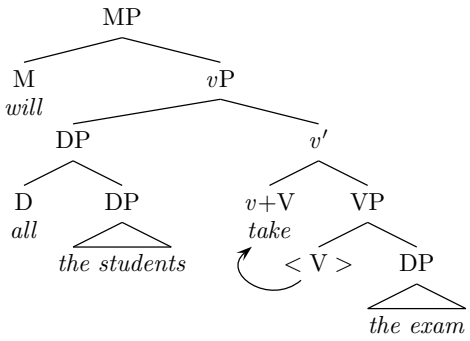
All the students will take...

- We start by building our *v*P.
  - Merge the DP *the exam* and the V *take* (checks [*u*D*] on V)
  - Merge *v* and VP (HoP)
  - Move V to *v* (checks [*u*V*] on *v*)
  - (Elsewhere,) Merge D *all* and DP *the students* (checks [*u*D*] on *all*)
  - Merge *all the students* with *v'* to complete the *v*P.



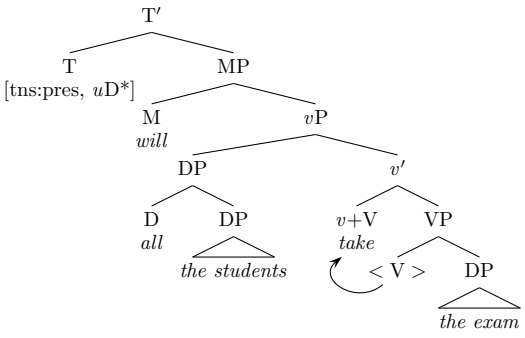
All the students will take...

- We Merge the M *will* with *v*P (HoP)
  - This values [*u*Infl:] on *v* as [~~*u*Infl:~~M].



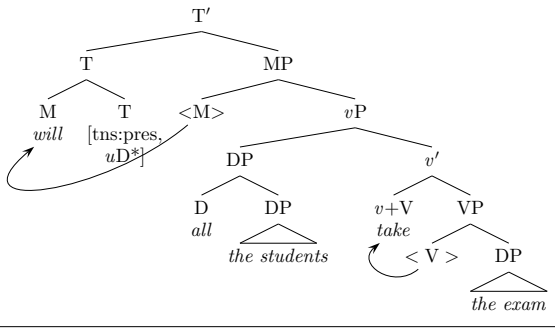
All the students will take...

- We Merge the T with MP (HoP)
  - This values [*u*Infl:] on M as [*u*Infl:pres*] (strong).



All the students will take...

- We move M up to T
  - This checks the strong [*u*Infl:pres*] on M.



All the students will take...

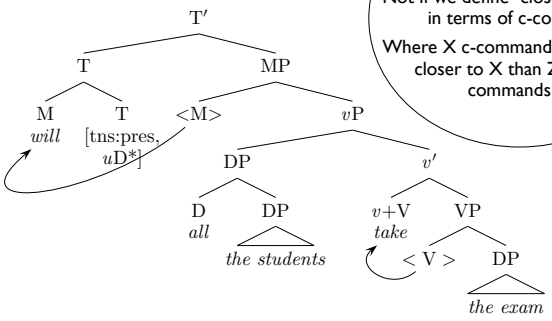
- Now, there are two possibilities:

Move the DP *all the students*.
Move the DP *the students*.

Is *all the students* closer to T than *the students* is?

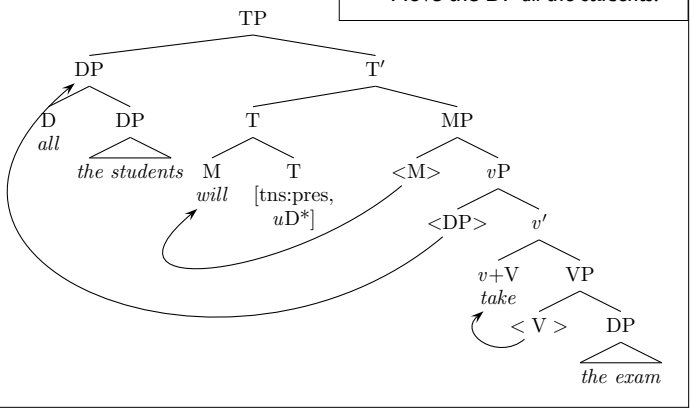
Not if we define "closer" as we did, in terms of c-command.

Where X c-commands Y and Z, Y is closer to X than Z is if Y c-commands Z.



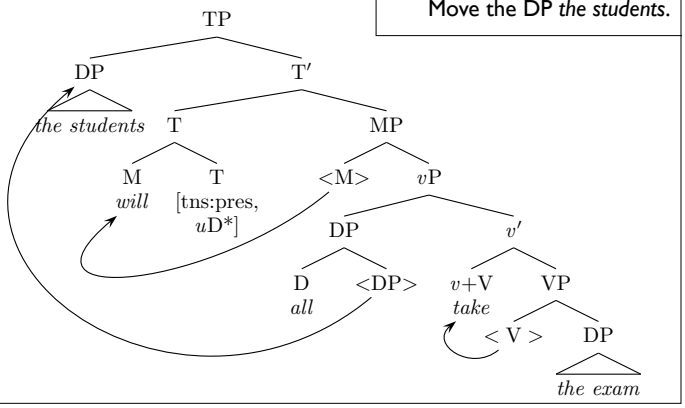
All the students will take...

Move the DP *all the students*.



The students will all take...

Move the DP *the students*.



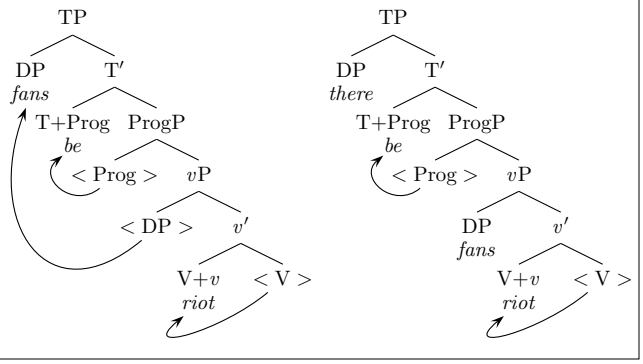
Expletive constructions

An **expletive** is an element that can be in subject position without having received a θ-role from anywhere.

- **It** had been raining.
- **There** were fans rioting on Comm Ave.

Expletive constructions

1) Fans were rioting on Comm Ave.
2) There were fans rioting on Comm Ave.



The Big Picture

Now that we've gotten some idea of how the system works, let's back up a bit to remind ourselves a bit about why we're doing what we're doing.

People have (unconscious) knowledge of the grammar of their native language (at least). They can judge whether sentences are good examples of the language or not.

- Two questions:
  - What is it that we know?
  - How is it that we came to know what we know?

History

Phrase Structure Rules

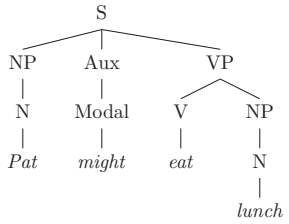
S → NP (Aux) VP	VP → V (NP) (PP)
NP → (Det) (Adj+) N	PP → P NP
Aux → (Tns) (Modal) (Perf) (Prog)	P → at, in, to, …
N → Pat, lunch, …	Modal → can, should, …
Tns → Past, Present	Prog → be -ing
Perf → have -en	

In trying to model what we know (since it isn't conscious knowledge) some of the first attempts looked like the phrase structure rules above (Chomsky 1957).

- An S can be rewritten as an NP, optionally an Aux, and a VP. An NP can be rewritten as, optionally a determiner, optionally one or more adjectives, and a noun. …
- What we know is that an S has an NP, a VP, and sometimes an Aux between them, and that NPs can have a determiner, some number of adjectives, and a noun.

Phrase Structure Rules
S → NP (Aux) VP
NP → (Det) (Adj+) N
Aux → (Tns) (Modal) (Perf) (Prog)
N → Pat, lunch, …
Tns → Past, Present
Perf → have -en
VP → V (NP) (PP)
PP → P NP
P → at, in, to, …
Modal → can, should, …
Prog → be -ing

History

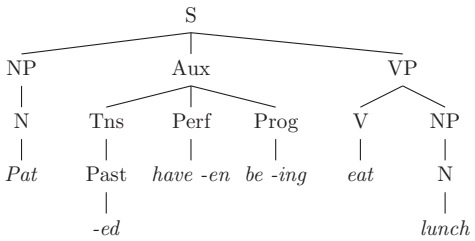


- In this way, many sentences can be derived, starting from S.
- The tree-style structure is a way to record the history of the derivation from S to the words in the sentence.
- We model our knowledge of English as a machine that (ideally, when it's finished) will generate all of the sentences of English and no others.

Phrase Structure Rules
S → NP (Aux) VP
NP → (Det) (Adj+) N
Aux → (Tns) (Modal) (Perf) (Prog)
N → Pat, lunch, …
Tns → Past, Present
Perf → have -en
VP → V (NP) (PP)
PP → P NP
P → at, in, to, …
Modal → can, should, …
Prog → be -ing

Affix Hopping

So, Chomsky proposed:
Aux → (Tns) (Modal) (Perf) (Prog)
Tns → Past, Present
Modal → can, should, …
Perf → have -en
Prog → be -ing
Past → -ed

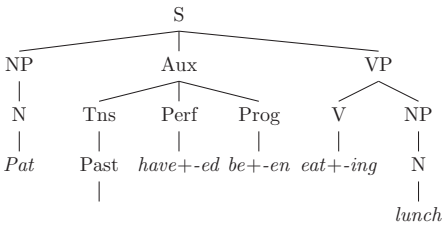


- If you build a sentence this way, things aren't in the right order, but there's a simple *transformation* that can be done to the structure to get it right.
- Empirically, tense, perfect *have*, and progressive *be* each control the form of the verbal element to their **right**.

Phrase Structure Rules
S → NP (Aux) VP
NP → (Det) (Adj+) N
Aux → (Tns) (Modal) (Perf) (Prog)
N → Pat, lunch, …
Tns → Past, Present
Perf → have -en
VP → V (NP) (PP)
PP → P NP
P → at, in, to, …
Modal → can, should, …
Prog → be -ing

Affix Hopping

So, Chomsky proposed:
Aux → (Tns) (Modal) (Perf) (Prog)
Tns → Past, Present
Modal → can, should, …
Perf → have -en
Prog → be -ing
Past → -ed



- **Affix Hopping**
  SD: afx verb
  SC: verb+afx
- The affixes all "hop to the right" and attach to the following word.
- An ancestor to the kinds of movement rules and of course the Agree operation we've been talking about.

History continues

Through the 60s there were good people working hard, figuring out what kinds of phrase structure rules and transformations are needed for a comprehensive description on English.

- As things developed, two things became clear:
  - A lot of the PSRs look pretty similar.
  - There's no way a kid acquiring language can be learning these rules.

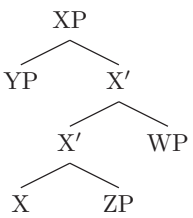
- Chomsky (1970) proposed that there actually is only a limited set of phrase structure rule types.
- For any categories X, Y, Z, W, there are only rules like:
  XP → YP X′
  X′ → X′ WP
  X′ → X ZP

X-bar theory

If drawn out as a tree, you may recognize the kind of structures this proposal entails. These are structures based on the "X-bar schema".

- XP → YP X′
  X′ → X′ WP
  X′ → X ZP
- YP being the "specifier", WP being an "adjunct", ZP being the "complement". Adjuncts were considered to have a slightly different configuration then.

Why is this better? The types of rules are much more constrained. AND it also makes predictions about structure and constituency that turn out to be more accurate.



GB

Around 1981, the view shifted from thinking of the system as constructing all and only structures with PSRs and transformations to a view in which structures and transformations could apply freely, but the grammatical structures were those that satisfied constraints on (various stages of) the representation.

- First, a "deep structure" (DS) tree is built, however you like *but*
  - Selectional restrictions must be satisfied
  - θ-roles must be assigned
  - Etc.
- Then, adjustments are made to get the "surface structure" (SS)
  - Things more or less like Affix Hopping, or moving V to *v*, or moving the subject to SpecTP.
  - Further constraints are verified here: Is there a subject in SpecTP? Etc.
- Finally, the result is assigned a pronunciation (PF), and, possibly after some further adjustments, an interpretation (LF).

Why is this better? Most of the construction-specific rules were made to follow from more general principles interacting. AND again, it caused us to look for predictions, which were better met.

Which brings us to 1993

The most recent change in viewpoint was to the system we're working with now (arising from the Minimalist Program for Linguistic Theory).

The constraints that applied to the structures in GB were getting to be rather esoteric and numerous, to the extent that it seemed we were missing generalizations.

- The goal of MPLT was to "start over" in a sense, to try to make the constraints follow from some more natural assumptions that we would need to make anyway.
- This new view has the computational system working at a very basic level, forcing structures to obey the constraints of GB by enforcing them locally as we assemble the structure from the bottom up.

Why is this better? It's a further reduction to even more general principles. The idea is that you need a few things to construct a language-like system—and there's nothing else.

Features and technology

The use of features to drive the system (uninterpretable features force Merge, because if they are not checked, the resulting structure will be itself uninterpretable) is a way to encode the notion that lexical items need other lexical items.

What the system is designed to do is assemble grammatical structures where possible, given a set of lexical items to start with.

- A comment about the technology here:
- The operations of Merge, Adjoin, Agree, and feature checking, the idea that features can be interpretable or not (or, strong or weak) are all **formalizations** of an underlying system, used so that we can **describe the system precisely enough to understand its predictions** about our language knowledge.

Features and the moon

We can think of this initially as the same kind of model as this:

$$f = G\frac{m_1 m_2}{r^2}$$

The Earth and the Moon don't compute this. But if we write it this way, we can predict where the Moon will be.

- Saying lexical items have uninterpretable features that need to be checked, and hypothesizing mechanisms (matching, valuing) by which they might be checked is similarly a way to formalize the behavior of the computational system underlying language in a way that allows us deeper understanding of the system and what it predicts about language.

The "Minimalist Program"

The analogy with the gravitational force equation isn't quite accurate, given the underlying philosophy of the MP.

The Minimalist Program in fact is trying to do this:

- Suppose that we have a cognitive system for language, which has to interact with at least two other cognitive systems, the **conceptual-intensional** and the **articulatory-perceptual**.
- Whatever it produces needs to be interpretable the vernacular of) each of these cognitive system for the representation to be of any use.
- Suppose that the properties of these external systems are your boundary conditions, your specifications.
- The hypothesis of the MPLT is that the computational system underlying language is an optimal solution to those design specifications. So everything is thought of in terms of the creation interpretable representations.