Nicholas LaCara · University of Toronto

Economy

1 Economy

- We haven't said much about economy so far except for the idea that various economy conditions exist.
  - We made an early distinction between methodological and substantive economy.
  - Substantive economy takes 'least effort notions as natural sources for grammatical principles'.
  - Economy conditions in the theory are a form of substantive economy.

See Hornstein et al. 2005, section 1.3.

- In our limited discussion of economy, we have generally talked about different options that a derivation might take.
  - For instance, Procrastinate (which we've abandoned) told us overt movement was more costly than covert movement.
  - The Preference Principle tells us which copies to reduce at LF.
  - Last Resort (briefly mentioned during our discussion of Bare Phrase Structure) penalizes superfluous derivational steps.
- Our (unstated) background assumption has been that these conditions compare derivations that begin from the same numeration.
  - In other words, we don't use economy conditions to compare derivations with different lexical items.
  - Moreover, we only use them to compare *convergent* derivations, as we discussed in the Binding lecture.
- Today we are going to review the idea that some aspects of derivational economy must be calculated locally, at particular points during the derivation, rather than globally by comparing full derivations.
- This discussion centers on a problem that was mentioned last time: Existential constructions seem to require that expletives be merged as soon as possible.
  - The punchline is that we need to check at various points during the derivation that economy conditions are being satisfied.
  - The way to do this is to Spell Out several times.
  - ...and maybe divvy up the numeration into smaller pieces.

2 Issues with existential and non-existential constructions

- Let's review the derivational difference between existential *there* constructions and non-existential constructions.
- Consider first existentials, which include the expletive *there*. This derivation follows how we did things in our discussion of Agree.

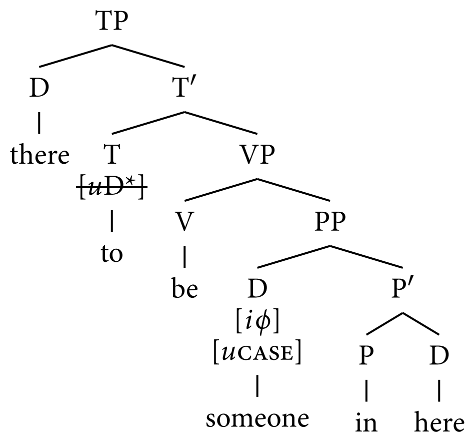
(1) There seems to be someone here.

- This is a (subject-to-subject) raising construction. As such, the expletive *there* is merged in SpecTP of the embedded clause to check the EPP feature on *to*.

Assume we can merge material directly from the numeration to check a feature on a head.

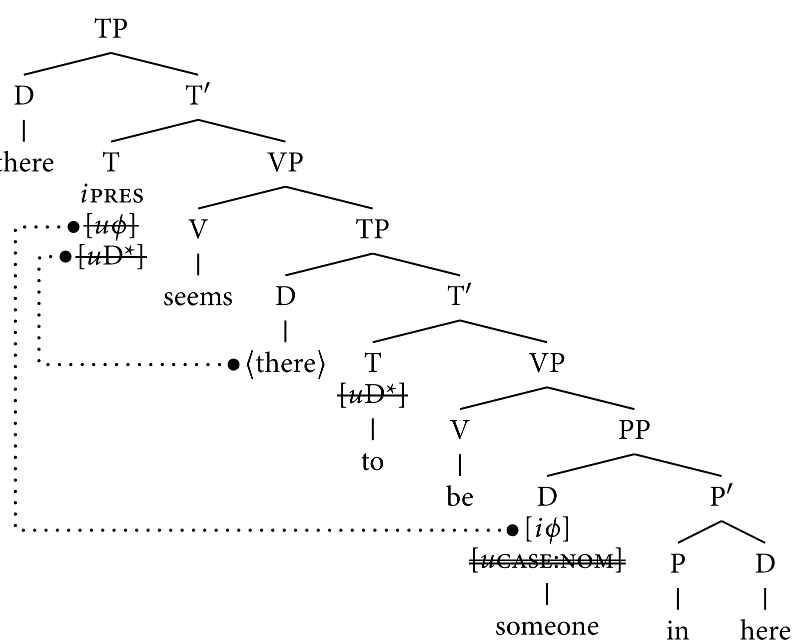
I'm going to continue assuming that the correlate is merged in SpecPP here. This is just for exposition; it could be somewhere else.

(2)



- The matrix T° merges with uninterpretable ϕ-features and an EPP feature.
  - The nearest head that can check the ϕ-features is the correlate (*someone*).
  - The nearest head that can check the EPP feature ([*u*D*]) is the expletive.

(3)



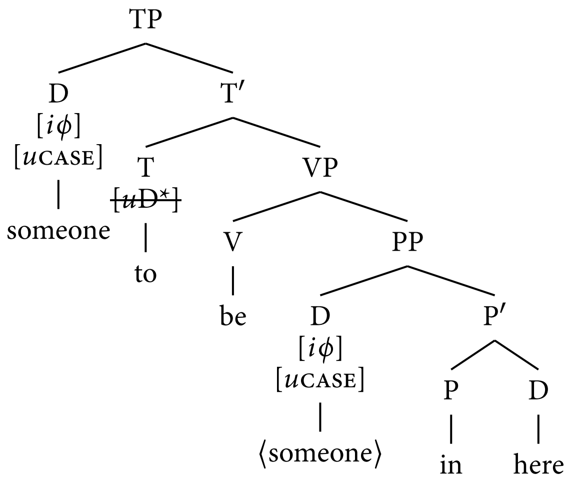
Because the probes are triggered by different features, and because the expletive is not specified for ϕ-features, the matrix verb does not have to agree with the element in its specifier.

- A non-existential sentence does not include *there* in its numeration.

(4) Someone seems to be in here.

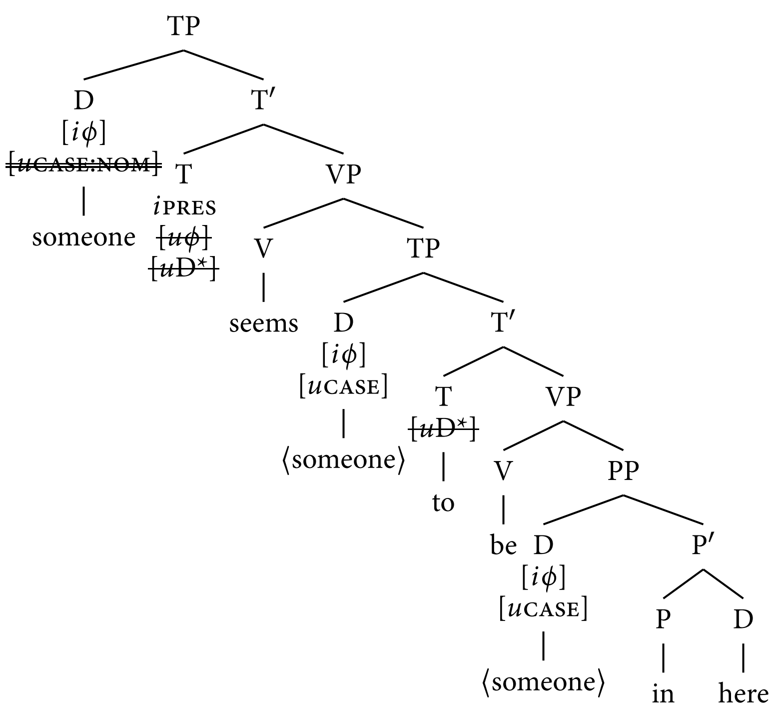
- This means at the step where *there* was merged in (2), here the only choice is to move *someone* to SpecTP of the embedded clause:

(5)



- Now when the matrix T° merges, there is only one D° that checks both the uninterpretable ϕ-features and the EPP

(6)



- So these two derivations can be distinguished based on the assumption that they begin with different numerations, one with an expletive and one without.

- This difference allows the EPP of the embedded clause to be checked in a different way in each derivation.

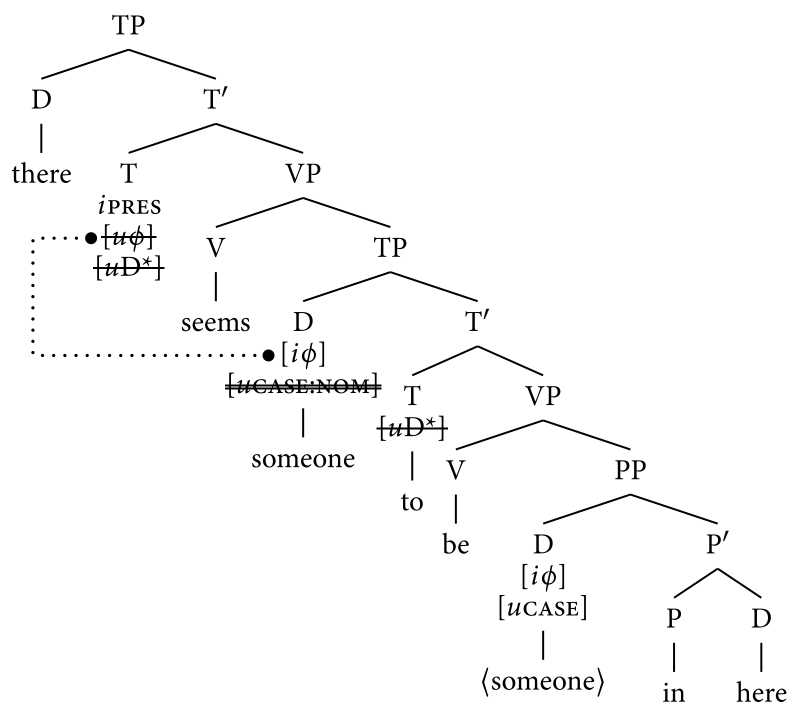
Some older theories assumed that expletives were inserted during the derivation in order to satisfy the EPP. But if we are taken the Inclusiveness Condition seriously, expletives must be part of the numeration.

3 Merge over move

- There is an issue, however, in that nothing so far prevents *someone* from moving to SpecTP in (2) as it does in (5).
  - Nothing in the numeration for (2) dictates what to do at this step.
- Given that the element that checks the EPP need not be the element that agrees with T°, this leads to a problem.
  - We should be able to merge the expletive in the specifier of the matrix TP to check its EPP feature.
  - T° should still be able to agree with the embedded subject to check its uninterpretable ϕ-features.

And assuming we must exhaust the numeration, we must do this.

(7)



- The resulting sentence is clearly ungrammatical:

(8) *There seems someone to be in here.

- The fact that movement happens in (5) tells us that such a movement should be possible, so something must rule it out in (2).

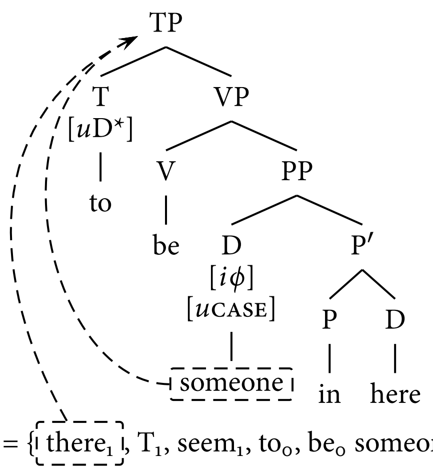
This was a problem under GB as well. Movement occurred freely under GB and illicit movement was ruled out by filters at various structural levels. No filter could rule out the illicit movement of the subject in the embedded clause in (7) without also incorrectly ruling out the licit movement in (5).

3.1 An existential quandary

- The derivations of (1) and (8) are identical up to the point where we must satisfy the EPP on the embedded non-finite T°.

- Furthermore, both (3) and (7) are convergent derivations starting from the same numeration.
    - Both check all uninterpretable features and exhaust the numeration.
- There is simply a choice of either merging the expletive or copying and merging the correlate, since both can satisfy the EPP:

(9)



$N = \{\text{there}_1, \text{T}_1, \text{seem}_1, \text{to}_0, \text{be}_0\ \text{someone}_0, \text{here}_0\}$

Subscripts indicate the number of each lexical item remaining in the numeration.

- It appears then that the fate of this derivation is determined by the choice that is made here.
- Since they are both convergent and start with the same features, they may be compared for economy purposes.
    - If economy rules out (8), then it must be less economical than (1).
- What is the economy metric that rules out (8) compared to (1), then?
    - It can't be derivational length. The derivations appear to have the same number of steps *vis-à-vis* Merge.
    - It cannot be the types of operations employed, either. The same number of Merge and Move operations happen in both derivations

(1) There seems to be someone in here.

(8) *There seems someone to be in here.

And, presumably, Agree relations, assuming that T° Agrees with the expletive in its specifier.

- So this cannot be some form of *global* economy that compares the output of the two derivations.
    - The need to compare full derivations should be reduced as much as possible anyway. From a computational perspective, it is very inefficient.
    - So if we need to compare derivations for the purposes of economy, we need to find some way to weigh economy considerations over the course of the derivation.

Imagine if the system worked in such a way that you did multiple derivations and then you went back and compared each of them after the fact.

3.2 Locality of economy

- Instead, we need an economy condition that decides between options at particular steps of the derivation.
  - The difference between (1) and (8) is determined by whether an expletive is merged or whether *someone* is copied, as sketched in (9).
  - From there the derivation is deterministic: Whichever choice is made at this point, the derivation will converge, but one output is still ruled out.
- So we want to rule out at this point the choice that leads to ungrammaticality, and that means we want to rule out moving *someone* to the embedded SpecTP.
  - This suggests an economy condition that favors merger from the numeration over moving material that has already been merged.
- But why should such a condition exist? In this case, there may be good reason:
  - Under the Copy Theory of Movement, there is no operation Move.
  - Movement is really a composite of two operations: Copy + Merge.
  - This means moving anything calls two operations (including Merge) and this is inherently more costly than just calling Merge on its own.

This sort of reasoning isn't strictly necessary, however. Here, we have a natural intrinsic reason for the economy conditions we observer, but it may be the case that some economy conditions are imposed by outside factors.

- Now, critically, both (3) and (7) employ movement. The difference is that the movement happens at different times.
  - Example (7) does it in the embedded clause.
    - But at that step, merger of the expletive is possible.
    - Assuming a local preference for Merge over Move, the derivation should choose to Merge the expletive at that step.
  - Example (3) does it in the matrix clause.
    - There is no option, at this point, to Merge an expletive from the numeration – the expletive has already been merged! The only option is to move (Copy + Merge).
    - Movement becomes viable because there is no alternative, more economical operation the derivation can make use of at this step.

Local in the sense of when the operations must apply.

If you want the derivation to converge, you have no choice.

- This is why economy needs to be calculated locally:
  - It's not about the number of times movement applies; it's about whether there's a better alternative at a point when movement might need to occur.
- Put another way, its about deciding what the most economical next step will be in a derivation.

3.3 Some consequences

- A conceptually interesting consequence of this is that derivations are only comparable up to the point where they are the same.

To the extent we need global economy conditions, anyway…

- Presumably, once a derivation takes a step that is different from some other derivation, they cannot be compared in terms of economy.
  - Two derivations that start from the same numeration will cease to be compared if at some step they merge different elements from the numeration.
  - It follows from this that two derivations that start with different numerations cannot be compared for the purposes of economy since they must diverge at some point.

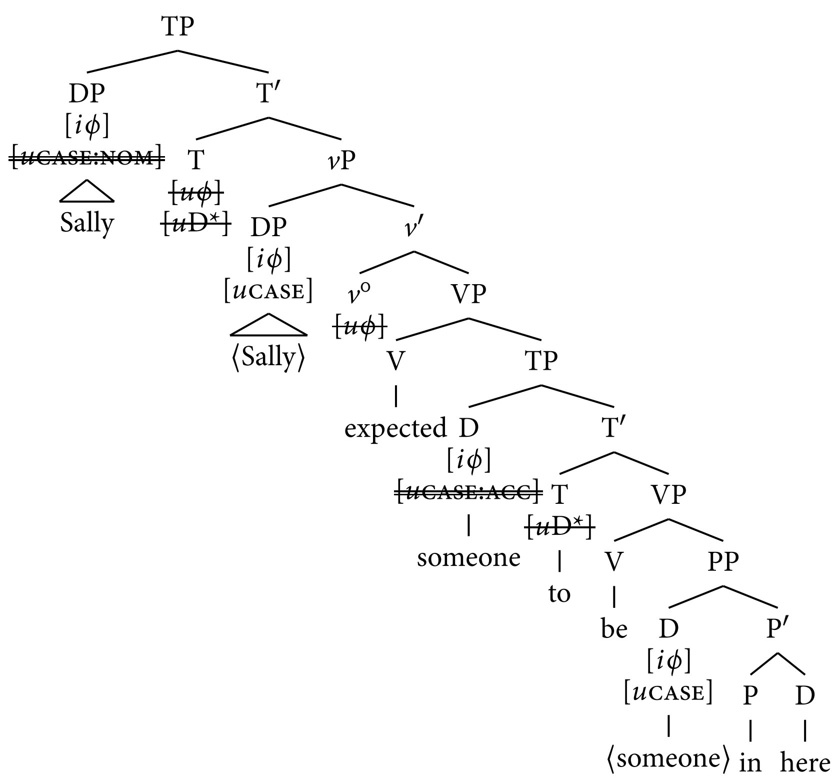
4 Non-expletive sentences

- We have to make sure that this new economy condition does not over-apply and block movement where we need it.
- Take, for instance, ECM constructions:

(10) Sally expected someone to be in here.

- In this case, we assume that *someone* moves to SpecTP of the embedded clause, as in (5), since there is no expletive in the numeration for this derivation.
- *Sally* then merges in the specifier of the matrix *v*P and moves to SpecTP, checking the EPP feature.

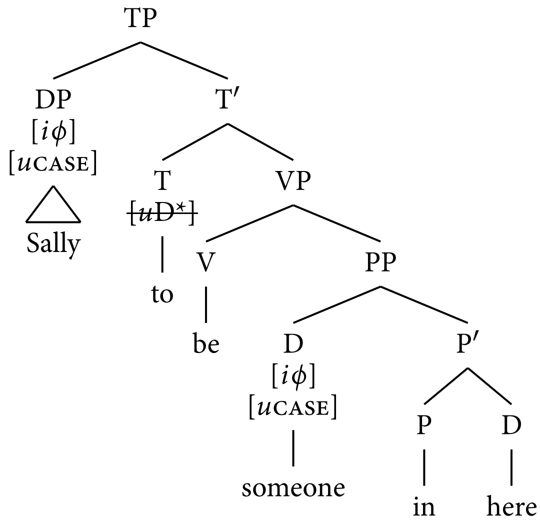
(11)



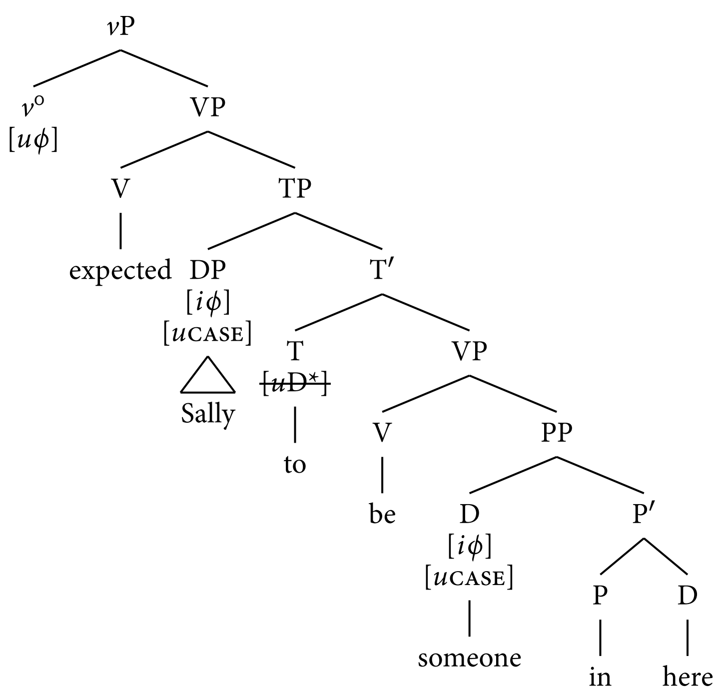
The embedded subject gets its Case feature checked by the matrix v°, as previously. But recall that this was previously done by covert movement; it is now handled by Agree!

- But if we are to prefer merging new material to moving old material, why then can we not merge Sally in the specifier of the embedded TP?

In other words, why can we not use *Sally* here the way we use an expletive to check the EPP?

(12)


- The short answer: Other things go wrong.
- Consider, for instance, how Case checking would have to proceed once we start building the matrix clause:

(13)


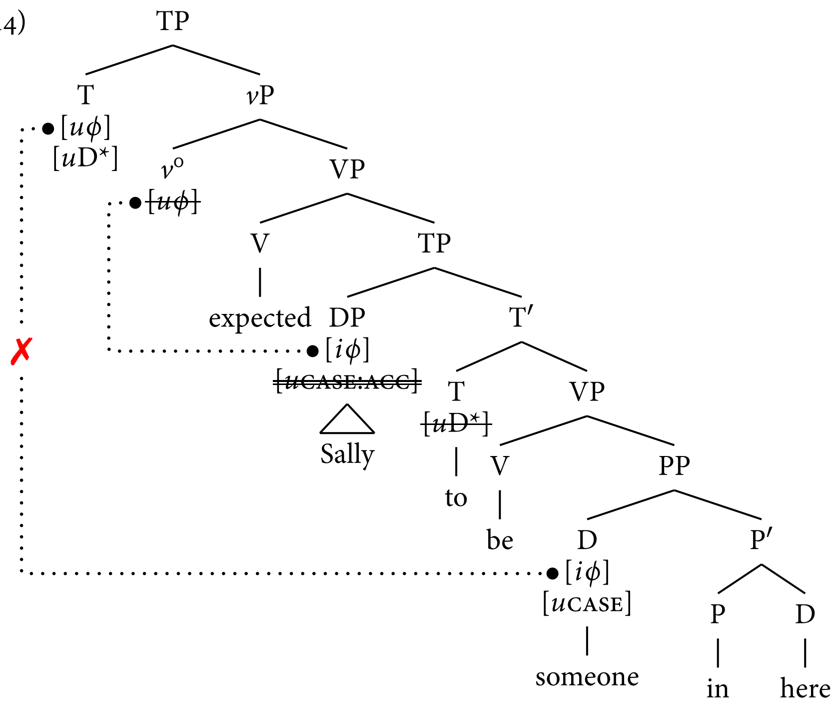
- v° will enter into an Agree relation with the first nominal it finds with interpretable ϕ-features – in this case *Sally*.

- But minimality will prevent the matrix T° from ever valuing the Case feature on *someone*, since *Sally* intervenes between the two!

Sally has interpretable ϕ-features which will always intervene for the ϕ-probe on T°.

Furthermore, *Sally* has all its uninterpretable features checked. Consequently, *Sally* can no longer be a valid goal for a probe, so T° cannot check its ϕ-features, either.

(14)



- Thus, in order to get a convergent derivation here, *someone* must move to the specifier of the embedded TP, in violation of Merge-over-Move.

5 Look-ahead

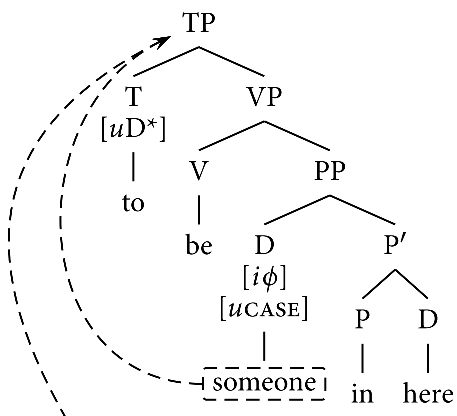
- So, given a choice between merging new material and moving old material, the Merge-over-Move economy condition requires us to choose Merge *unless the resulting derivation will not converge*.
  - How the hell does the derivation know it won't converge before it converges (or doesn't)?
- Economy calculated locally, and it is not obvious, when faced with such a decision, how the derivation will proceed.
  - Merge-over-Move will rule out future convergent derivations. And we want it to do that!
  - But the local nature of locality means it cannot know what the future of a derivation will look like.
- Let's consider again example (10), repeated here:

(15) Sally expected someone to be in here.

- When building the embedded clause, we are faced with a choice after merging non-finite *to*: Move *someone* to check the EPP, or merge *Sally*.

Both elements are of category D, so they are both capable of checking the [*u*D*] feature on T.

(16)



$N = \{$Sally$_1$, T$_1$, v_1^o, expect$_1$, to$_0$, be$_0$ someone$_0$, here$_0\}$

- Both options are grammatically possible at this derivational step.
- Merging *Sally* satisfies Merge-over-Move, and nothing goes immediately wrong.
- But things go wrong later on down the road: It will become impossible to check the Case feature on *someone*, and the derivation will crash at LF and PF.
- That is, we must build the rest of the tree to know that these crashes will occur!
  - We syntacticians know what will happen, but how could the derivation 'know' this?
  - Given that sentences can include an unbounded number recursions, it could be a long time before the derivation concludes.

Facts about expletives tell us we must be able to merge material from the numeration at this point.

- This is a problem known as LOOK-AHEAD.
  - The derivation appears to need to know about subsequent steps to know how to behave an an earlier step.
  - But this is not consistent with the assumption that this sort of derivational economy is calculated locally.
  - It also seems at odds with the entire notion of having a derivation.
- Now, in the example here, there may be a way out:
  - As soon as *Sally* merges, minimality dictates that no subsequently merged ϕ-probe will ever be able to value the Case feature on *someone*.
  - If we can get the derivation to realize this *at this point* (or very soon after), then the derivation can be canceled, preventing any further computation from occurring.

And remember, we want this. We don't want to just throw out local economy, since we can't just look at the number of steps or the number of operations. We need to calculate that something can go wrong *at this point*.

To CANCEL a derivation means to stop it before it exhausts the numeration.

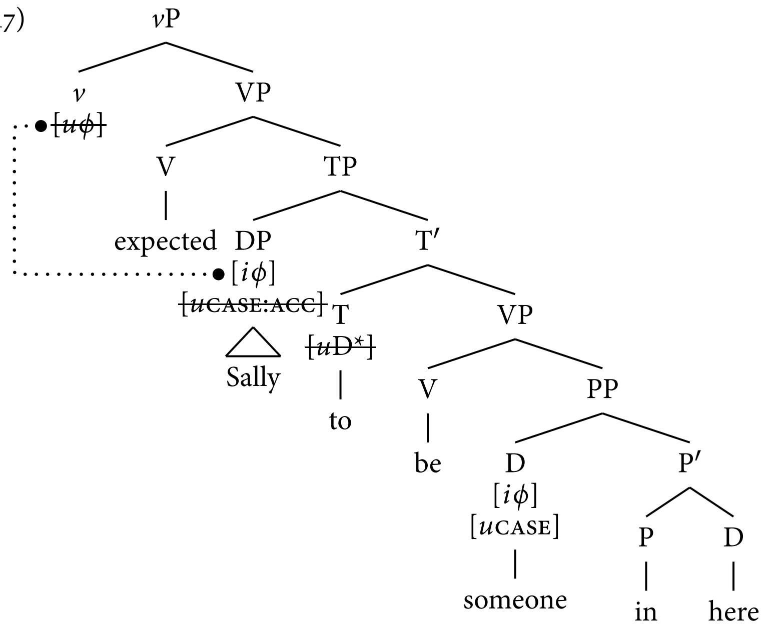
- **Spoiler:** If the derivation spells out early and often, we can solve this problem.

6 Phases

- As discussed above, if we build the structure below, the derivation will crash.

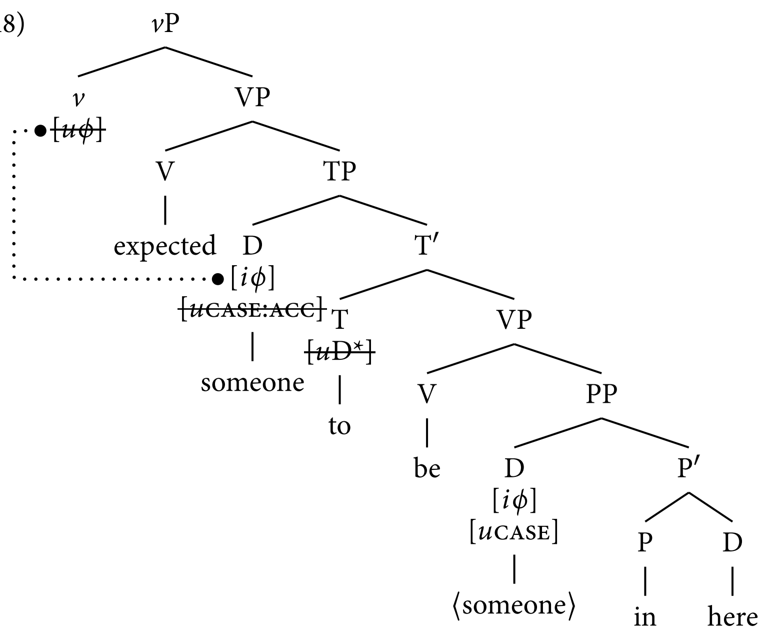
At this point, we can be sure that *someone* will never get its Case checked, regardless of what happens after this step.

(17)



- Compare this, however, to the convergent derivation where *someone* moves to SpecTP:

(18)



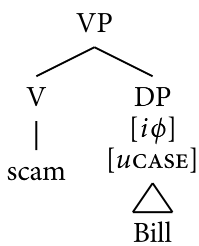
- Notably, the TP in in this tree has all of its uninterpretable features checked.

I'm assuming P checks the Case feature on *here*.

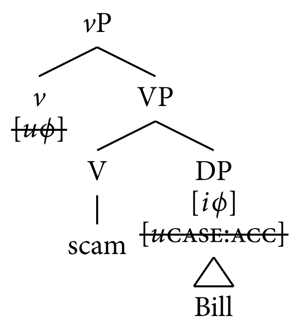
- If the derivation were to Spell Out once *v* merges we could capture the difference between (17) and (18) example.
  - Example (17) would be sent to the interfaces with an unvalued uninterpretable feature, causing crashes at LF and at PF.
  - Example (18), on the other hand, has all of the uninterpretable features checked, and so should converge at the interfaces.
- As Chomsky (2000) points out, convergence is not a property of only the final output of the derivation. Other syntactic objects created by the derivation along the way may be legible at the interfaces.
- Critically, we don't want to evaluate the derivation for legibility at every step of the derivation. It's clear that not every syntactic object is legible at the interfaces.
- A simple VP, for instance, will definitely be a malformed PF/LF object before *v* merges, since the Case feature on the internal argument will not be checked:

In other words, if we sent this VP to the interfaces, the derivation would definitely crash. Thus, we don't want to say we can just send things to the interfaces whenever we want.

(19)



(20)



- Once *v* checks the Case on the object, though, VP will be legible at LF.

6.1 Phase heads

- If VP is evaluated for convergence after *v* merges, we can immediately check whether a derivation is doomed to crash.
- Chomsky (2000, 2001) argues that syntactic derivations proceed in PHASES.
  - A phase is a syntactic object, the complement of whose head may be inspected for convergence.
- This means that upon the merger of so called PHASE HEADS, the derivation up to that point is sent to the interfaces to be inspected for conversion.
- There are two generally agreed upon phase heads: C and (Case-assigning) *v*.
- *v*P is thus a phase. When *v*P is assembled, its complement is sent to the interfaces. If it is legible at both interfaces, the derivation continues. If it is not, the derivation is CANCELED at that step.
- CP is also thought to be a phase due to the properties of TP.

D° is often thought to be a phase head too. See, *e.g.*, Bošković (2014), but see also Matushansky (2005) for complications with this view.

- First, non-finite clauses demonstrate, TPs can be assigned Case from outside the TP, either by an ECM predicate or by the non-finite complementizer *for*:

The matrix v° in ECM predicates fulfills the same role as the non-finite complementizer *for*, checking Case in a non-finite clause.

(21) a. Mary [$_{vP}$ expects [$_{TP}$ Sally to scam Bill]].
b. It would be unsurprising [$_{CP}$ for [$_{TP}$ Sally to scam Bill]].

- Imagine we try to merge one of these TPs with the finite complmentizer *that*.
  - If *that* merges with a finite TP, it will send TP to the interfaces, and since all the features in TP are checked, the derivation will continue.
  - If *that* merges with an ECM complement, the TP will not converge at the interfaces since the subject of the TP will not have its Case feature valued.

(22) a. [$_{CP}$ that [$_{TP}$ she will scam Bill]]
b. [$_{CP}$ that [$_{TP}$ Sally $\begin{bmatrix} i\phi \\ u\textsc{Case} \end{bmatrix}$ to scam Bill]]

- This can be evaluated immediately if C is a phase head that causes its complement to be sent to the interfaces.

A similar argument can be crafted from Control clauses. We can evaluate immediately whether PRO is the subject and has received Null Case (as discussed by Hornstein et al. 2005, section 10.4, page 348).

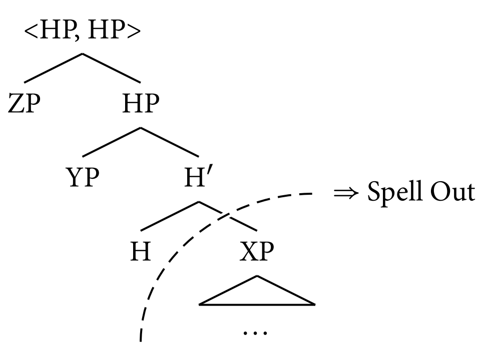
6.2 Phase impenetrability

- Once a phase-head complement has been sent to the interfaces, it becomes, by hypothesis, completely inaccessible to further syntactic computations.
- This is imposed by the Phase Impenetrability Condition, or PIC. Here, *edge* refers to the head, specifier(s), and adjuncts of HP:

(23) *Phase Impenetrability Condition* (Chomsky 2000):
In a phase α with head H, the domain of H is not accessible to operations outside α, only H and its edge are accessible to such operations.

Chomsky (2001) revises this definition to allow some limited operations outside of the domain of H before the next phase head merges.

- Essentially, this means that material from outside of HP will not be able to interact with material in the complement of H.

(24)


- The PIC is claimed to ease the computational burden:
  - A phase head complement must have all of its uninterpretable features checked in order for the derivation to continue.
  - If it is found to be convergent and the derivation continues, it must be the case that there are no more uninterpretable features in that phase head complement that need to be checked.
  - That means that every element in the phase head complement must be inactive for further computations, so it is safe to remove and the derivation need no longer consider material inside of it.

In other words, any computation that could be done with that material has been done, so there is no need for the derivation to backtrack and evaluate that material again. In that sense, the PIC is motivated (and may even result from) general principles of economy.

- Practically, this means that in order for the derivation to have access to any material that merged in the complement of H after H merges, that material must move to the edge of HP.

6.3 A quick derivation

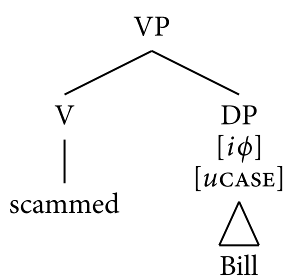
- Let's derive the embedded CP in the following sentence under phase theory.

(25) Mary believes that Sally scammed Bill.

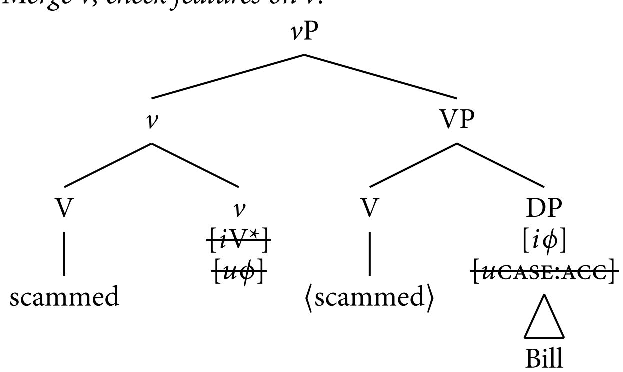
(26) *Phase 1*:

In which Doris gets her oats.

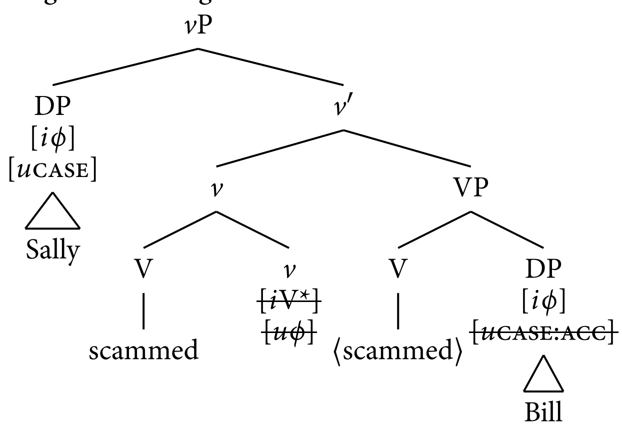
a. *Build VP*:



b. *Merge v; check features on v*:

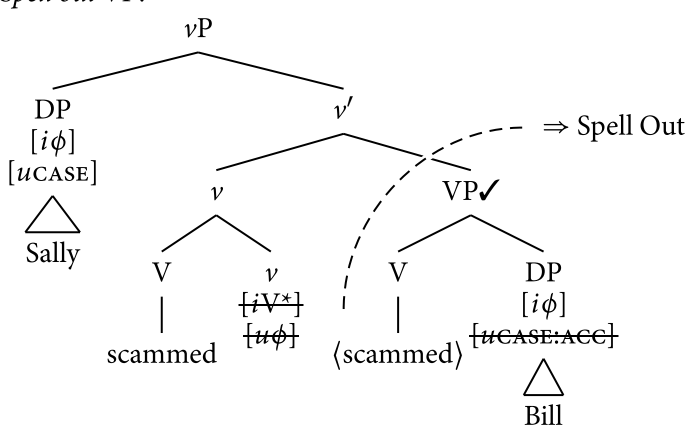


c. *Merge external argument*:



At this point the *v*P phase is complete, so it may Spell Out its complement.

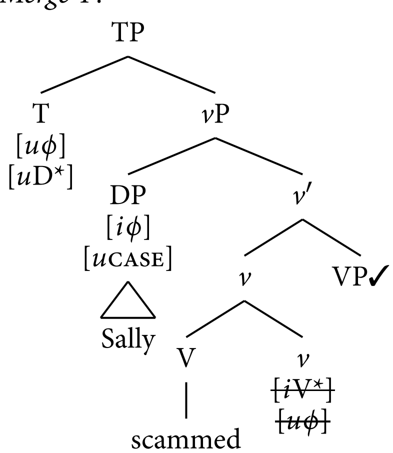
d. *Spell out VP*:



All of the uninterpretable features in VP are checked, so the VP converges at the interfaces and the derivation is allowed to proceed. Following Hornstein et al. (2005), I'll indicate conversion with a ✓ symbol.

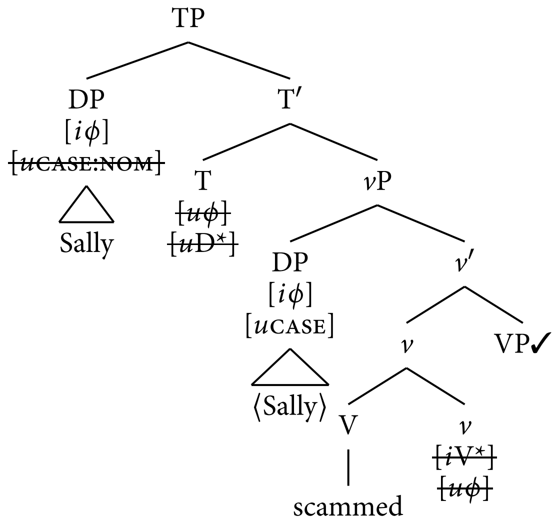
(27) *Phase 2*:

a. *Merge T*:

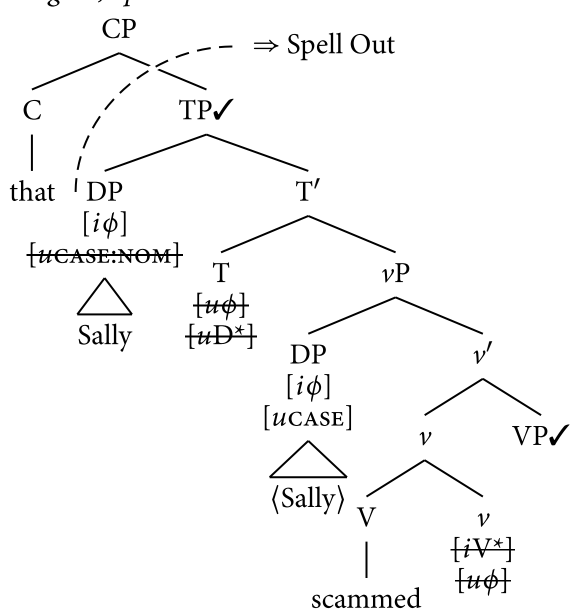


The VP having been spelled out, we no longer need to consider what was inside it. We can proceed to the CP phase.

b. *Check features on T*:



c. *Merge C; Spell out TP*:



All of the uninterpretable features in this TP are checked, so the TP converges, and the derivation is allowed to proceed!

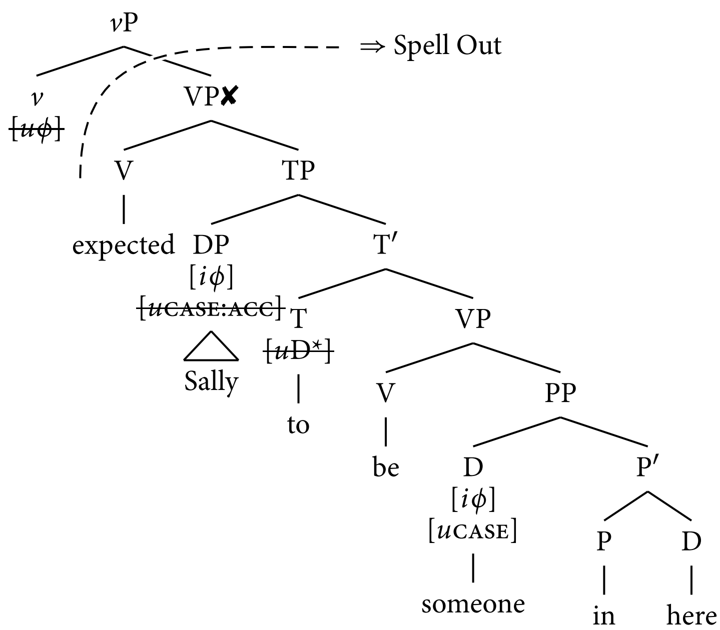
6.4 Non-existentials solved (finally)

- Returning to (10)/(15), repeated in (28), it becomes possible to rule it out as soon as *v* merges in the matrix clause, as shown in (29).

(28) Sally expected someone to be in here.

- As seen in (17), it remains impossible to check Case on *someone*, but *v*° Spells Out its complement, immediately sending the structure to the interfaces.

(29)



- Because there are unchecked uninterpretable features in VP, the VP will not be convergent at the interfaces.

Specifically, the Case feature on *someone* is still unchecked here.

- Consequently, the derivation will be canceled at this point.
- Thus, derivation by phase permits us to calculate locally that merging *Sally* at the stage of the derivation will result in a non-convergent derivation.
  - If we evaluate the economy of the derivation once each phase is sent to the interfaces, the derivation will be able to quickly determine whether merging new material rather than moving will lead to a crash.

Except. . . does it? The derivation will find out it did the wrong thing very quickly, but only *after* it merges *Sally*. Why should that mean we get to violate Merge-over-Move now?

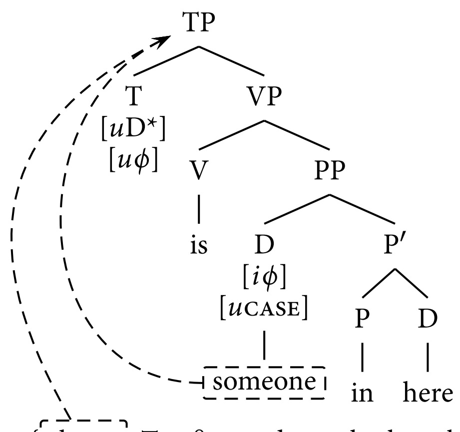
7 Lexical subarrays

- The whole goal of introducing phases here is to allow us to determine that a derivation will crash in the future without resorting to look ahead.
- However, it appears that there are pairs of sentences that can be derived from the same numeration where one of the pair should be ruled out by economy.

(30) a. Someone is wondering whether there is someone in here.
 b. There is someone wondering whether someone is in here.

- These sentences each have the same lexical items, but in (30a) the expletive is inserted earlier than in (30b), where subject movement occurs in the embedded clause.
  - Therefore, (30b) should be ruled out by Merge-over-Move
  - Expletive *there* should be inserted in the embedded clause!

(31)



$N = \{\text{there}_1, \text{T}_1, v_1^{o}, \text{wonder}_1, \text{whether}_1, \text{be}_1\ \text{someone}_1, \text{here}_0\}$

7.1 Subarrays

- One way to deal with this problem is to assume that numerations are actually structured.
- We already employ numerations as a device to limit computational complexity by using them as a mediator between the syntax and the lexicon.
- Once we introduce the notion of phase head into the syntactic computation, we can also propose that they play a role in the numeration.
- Chomsky (2000, 2001) proposes that numerations are actually composed of SUBARRAYS.
  - Whereas a traditional numeration is a multiset, under this view a numeration is a (multi)set of sets.
  - Each set in the numeration is a subarray, and every subarray contains a phase head:

(32) $N = \{\{C_1, \dots\}, \{v_1, \dots\}, \dots\}$

- This follows from the idea that phase heads introduce domains for computing convergence.
  - If the goal of the system is to reduce computational complexity across the board, then we might reasonably expect the numeration to be structured around phase heads.

Hornstein et al. (2005) also discuss the possibility that there is a Case-assigning *be* and a non-case assigning *be*. This lexical ambiguity approach does not generalize to cases of expletive constructions that lack *be*, however, so it's not the best way to describe what is going on here.

7.2 Local economy and subarrays

- Under a system with subarrays, derivations proceed one phase at a time, but these phases are determined in the numeration.
- The computation thus proceeds one subarray at a time.
  - The system will activate the first subarrary σ_1 from the numeration and build the first phase, using all the lexical items in σ_1.

  - The resulting phase is then sent to the interfaces to be inspected for convergence.
  - If the phase converges, the derivation moves on to subarray σ_2, otherwise the derivation is canceled.
  - A derivation is completed only after all the subarrays have been exhausted.

- Under this set up, (30a) will have a different numeration from (30b).
- In (30a), repeated here, the first subarray will contain the expletive:

(33) a. Someone is wondering whether there is someone in here.
b. N = {{C_1, T_1, be_1}, {someone, v_1, $wonder_1$}, {$whether_1$, $there_1$, T_1, be_1, $someone_1$, in_1, $here_1$}}

- When building the first phase, when it comes time to satisfy the EPP on T, *there* will be in the lexical subarray from which the first phase is constructed.
- Consequently, Merge-over-Move dictates that the expletive should be merged. Moving *someone* will be ruled out.
- In (30b), repeated here, the expletive is in a later subarray:

(34) a. There is someone wondering whether someone is in here.
b. N = {{C_1, $there_1$, T_1, be_1}, {someone, v_1, $wonder_1$}, {$whether_1$, T_1, be_1, $someone_1$, in_1, $here_1$}}

Another way I've seen this discussed is that once a phase is completed, the derivation gets to access the lexicon again, so the subarrays are determined as the derivation proceeds.

- Because of this, moving *someone* is the only possibility in the first phase.
- Crucially, these examples cannot be compared for the purposes of economy because they begin from different numerations.
- This ensures that Merge-over-Move is calculated locally.

This should let us understand the potential problem with (29), noted above ⇑. If *Sally* is not in the same subarray as *someone*, then the only option in that example will be for *someone* to move.

8 Cyclicity

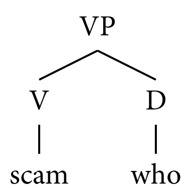
- Phases are thought to play a partial role in the explanation of successive cyclic movement, particularly *wh*-movement and other forms of A′-movement.
- It is fairly well established that *wh*-movement does not progress in one fell swoop and is in fact broken down into several smaller movements.
- This is fairly strange on the hypothesis that *wh*-movement must check a [WH] feature on C. Why should it have to stop at intermediate specifiers on the way?
- Recall that the PIC, repeated here, blocks movement out of a phase that has already been sent to the interfaces:

(35) *Phase Impenetrability Condition*:
In a phase α with head H, the domain of H is not accessible to operations outside α, only H and its edge are accessible to such operations.

- If this is right, then any element that undergoes apparently long-distance movement will have to undergo a series of shorter movements as phases are sent to the interfaces.
  - That is, every *wh*-word must get to a phase edge, otherwise it will not be accessible to operations outside that phase.
- Consider the derivation of the following:

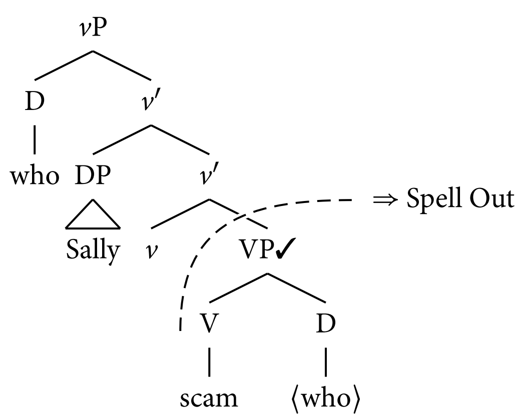
(36) Who does Mary think that Sally will scam?

- *Who* merges as the complement of *scam*:

(37)


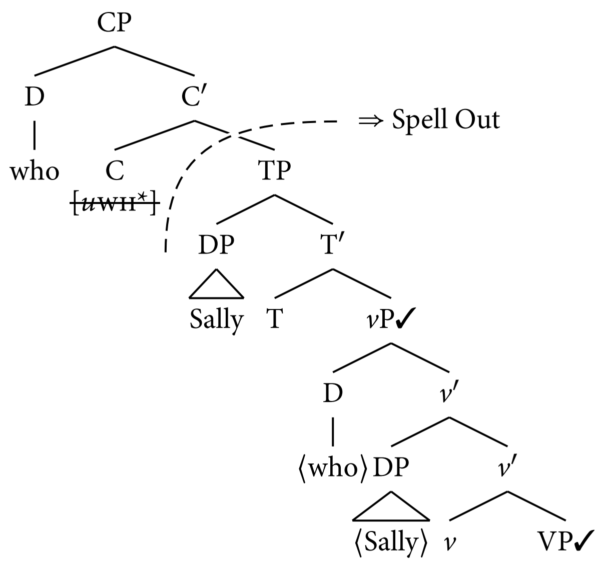
- After *v* merges, VP will be sent to the interfaces.
- In order to make sure that the *wh*-element remains accessible to operations outside the phase (*e.g.*, an Agree probe triggered by a *wh*-C head), *who* must move to the edge of the phase:

I'm suppressing things like head movement and ϕ-agreement in these trees since I'm focusing on *wh*-movement.

(38)


- This will repeat every time the *wh*-element would otherwise remain in a phase that must be sent to the interfaces.

(39)



We have to assume, though, given our current assumptions, that *who* in this derivation bears some unvalued feature that makes it a valid goal. That would explain, in part, why it keeps moving out of phases. If it stayed in the, it would fail to get checked; I believe Bošković (2007) develops an idea like this. For an alternative view, Fox and Pesetsky (2005) try to pin cyclic movement on linearization.

References

Bošković, Željko. 2007. On the locality and motivation of Move and Agree: An even more minimal theory. *Linguistic Inquiry* 38:589–644.

Bošković, Željko. 2014. Now I'm a Phase, Now I'm Not a Phase: On the Variability of Phases with Extraction and Ellipsis. *Linguistic Inquiry* 45:27–89.

Chomsky, Noam. 2000. Minimalist inquiries: The framework. In *Step by Step*, ed. R. Martin et al., 89–155. Cambridge, Mass: MIT Press.

Chomsky, Noam. 2001. Derivation by Phase. In *Ken Hale: A Life in Language*, ed. Michael Kenstowicz, 1–52. Cambridge, Mass: MIT Press.

Fox, Danny, and David Pesetsky. 2005. Cyclic Linearization of Syntactic Structure. *Theoretical Linguistics* 31:1–45.

Hornstein, Norbert, Jairo Nunes, and Grohmann. 2005. *Understanding Minimalism*. Cambridge University Press.

Matushansky, Ora. 2005. Going through a phase. In *Perspectives on Phases*, ed. Martha McGinnis and Norvin Richards, volume 49 of *MIT Working Papers in Linguistics*. Cambridge, Mass: MiTWPL.