

Constraining Merge II: Selection and dependent features

Introduction to Syntax, Lecture 7

Sandhya Sundaresan, EGG Summer School, Wrocław 2019

August 6, 2019

Now we are in a position to see why a theory of c-selection is needed for Merge:

- ☞ Essentially we can use the dependent features in c-selection as instructions, triggers for appropriate instantiations of **Merge**.
- ☞ If a syntactic object doesn't Merge with the sort of thing demanded by its dependent features, the derivation will crash, i.e. it will fail to derive a grammatical sentence.
- ☞ This is how we can ensure that only those derivations succeed in which the right sort of things have Merged.

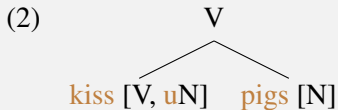
Specifically, we can now encode c-selection with dependent category features:

(1) **kiss** [V, uN]



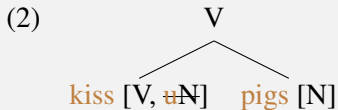
Specifically, we can now encode c-selection with dependent category features:

(1) kiss [V, uN]



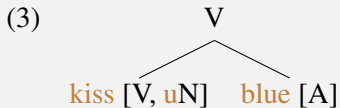
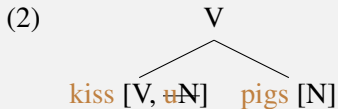
Specifically, we can now encode c-selection with dependent category features:

(1) kiss [V, uN]



Specifically, we can now encode c-selection with dependent category features:

(1) kiss [V, uN]



Specifically, we can now encode c-selection with dependent category features:

(1) kiss [V, uN]

(2)

```
graph TD; V[V] --- kiss[kiss]; V --- pigs[pigs];
```

kiss [V, uN] pigs [N]

(3)

```
graph TD; V[V] --- kiss[kiss]; V --- blue[blue];
```

kiss [V, *uN] blue [A]

- So our theory can correctly rule out sentences where the wrong category of argument combines with a predicate.
- It can also rule out sentences where a predicate doesn't combine with enough arguments.
- Either way, an unchecked dependent category feature will be left over at the end, causing a crash.

In addition to c-selection, we also need s-selection.

- This is where we encode the requirements a predicate places on the **semantic** type of its arguments.
- E.g. the object of **ask** can be of various syntactic categories, but it has to be semantically a question or piece of information that can be queried.

We won't really worry about s-selection, but you should know that it exists and seems to be independent of c-selection.

For purposes of comparison with other Minimalist theories (including the one in David Adger's 2003 book *Core Syntax*), note the following:

- ☞ The distinction that we are drawing between dependent and independent corresponds essentially to what those theories call **uninterpretable** and **interpretable**.
- ☞ This is the explanation for the **u** notation we are using for dependent features.

- ☞ I am not adopting this terminology here because it is tied to a particular set of assumptions about the status of these features which we cannot motivate.
- ☞ In our insistence on simplicity and generality, we will also depart from the standard theory of interpretability of features in other ways as we move forward.
- ☞ But most of the insights we develop here will be easily translatable into such a theory.

Back to heads

Introduction to
Syntax

Lecture 7:
Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

The structure of
VP

We can bring this all together to model the determination of the head in a given phrase:

(4) **Definition of Head**

The head of a phrase is the syntactic object which selects the other object which it Merges with to create the phrase.

- ⇒ So the object that has a dependent category feature checked off in the Merge process is the head.

And we can set down the importance of being the head:

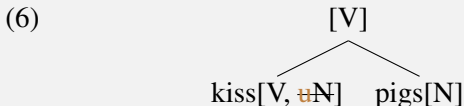
(5) **Headedness**

The item that selects is the item that projects.

- Imagine that object X selects object Y, merging with it to create object Z.
- The further properties of object Z will be projected from the head, object X.

An example:

- The constituent **kiss pigs** is headed by **kiss**, because **kiss** selects a noun like **pigs**.



- So **kiss pigs** is essentially verbal, as **kiss** is verbal, and has a distribution related to verbs, not nouns:

- (7) a. I want to [_V sing].
 b. I want to [**kiss pigs**].
- (8) a. I want [_N pigs].
 b. *I want [**kiss pigs**].

The way things are set up lets us derive an interesting corollary:

(9) **Ban on Unchecked Features on Non-heads**

If X selects Y and the two Merge, Y cannot have any unchecked dependent features.

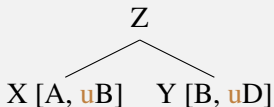
- In other words, only the head can have unchecked features.

Consider why this is:

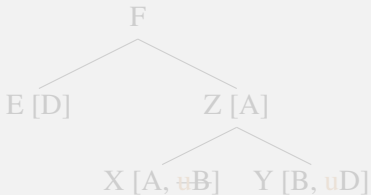
- When X and Y Merge, the features from X will project to the newly created constituent, but the features of Y won't.
- When this merges with something else, the features projected from X can be checked, but those on Y can't, because Y won't be the sister of the newly merged object.
- Any dependent features on Y will thus remain forever unchecked, leading to a crash.

In structural terms:

(10)

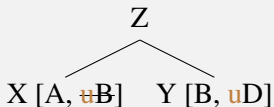


(11)

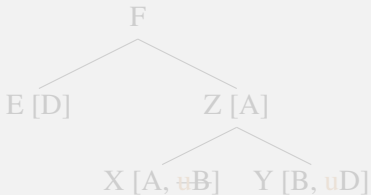


In structural terms:

(10)

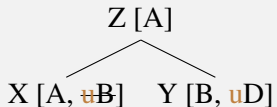


(11)

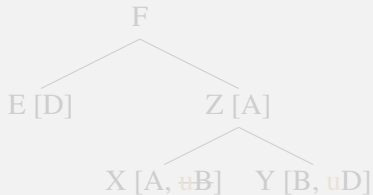


In structural terms:

(10)

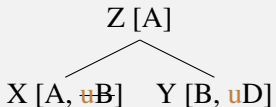


(11)

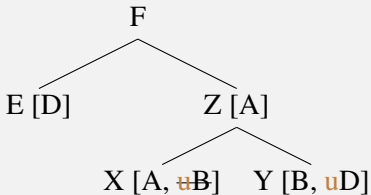


In structural terms:

(10)

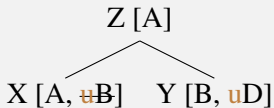


(11)

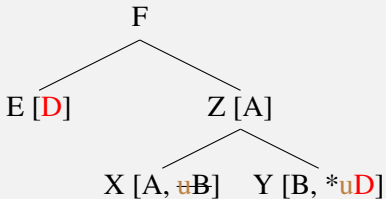


In structural terms:

(10)



(11)



There is evidence that this is actually correct. Consider:

(12) Ellie became tired of elephants.

- The verb **become** c-selects for an adjective, and the adjective **tired** c-selects in turn for a preposition, and the preposition **of** c-selects for a noun.
- 12 has all the right things for those requirements to be satisfied, but we could imagine them being combined lots of different ways.

Introduction to
Syntax

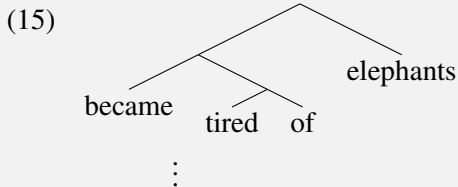
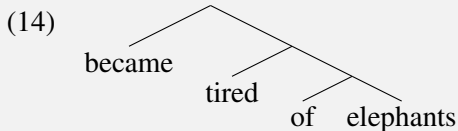
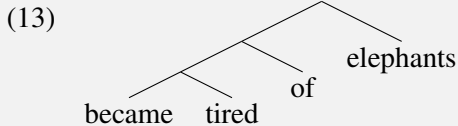
Lecture 7:
Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

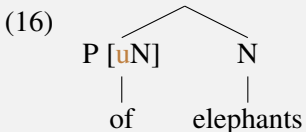
The structure of
VP



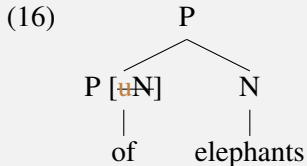
But the ban in 9 predicts that only one structure is possible: the one where the selectional feature on each object is checked before it itself is selected:

(16) N
 |
 elephants

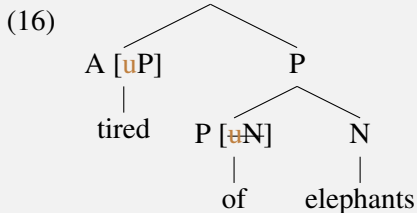
But the ban in 9 predicts that only one structure is possible: the one where the selectional feature on each object is checked before it itself is selected:



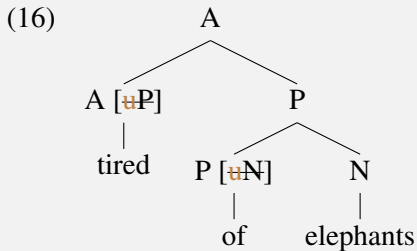
But the ban in 9 predicts that only one structure is possible: the one where the selectional feature on each object is checked before it itself is selected:



But the ban in 9 predicts that only one structure is possible: the one where the selectional feature on each object is checked before it itself is selected:

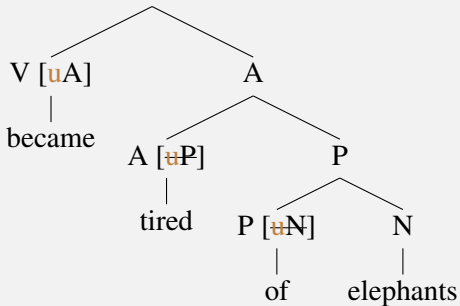


But the ban in 9 predicts that only one structure is possible: the one where the selectional feature on each object is checked before it itself is selected:

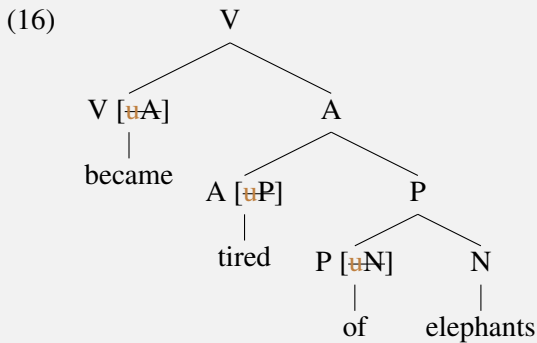


But the ban in 9 predicts that only one structure is possible: the one where the selectional feature on each object is checked before it itself is selected:

(16)



But the ban in 9 predicts that only one structure is possible: the one where the selectional feature on each object is checked before it itself is selected:



This is a good result, because constituency tests pick out the same structure. E.g.:

- (17) [Tired of elephants] is something Ellie will never become.
- (18) * [Become tired] is something Ellie never will of elephants.

The internal structure of phrases

Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

Complements

Specifiers

Adjuncts

The structure of
VP

Consider:

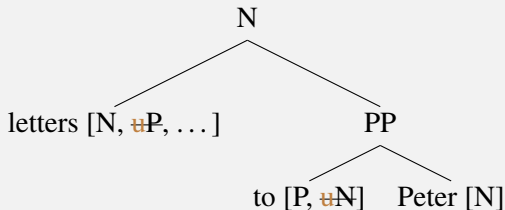
- (19) * letters to
(20) letters [to Peter]

- **to** by itself is lacking something. It selects for an N but hasn't combined with one yet, so Merging it with **letters** is ungrammatical.
- But **to Peter** is complete, the [**uN**] selection feature on **to** having been checked, so it can Merge with **letters**.

Constituents like **to Peter**, which have checked all their dependent features, are called **maximal** objects or **phrases**.

- A maximal object built around a noun is an NP, one built around a P is a PP etc.

(21)



Being Maximal depends on having no unchecked dependent features.

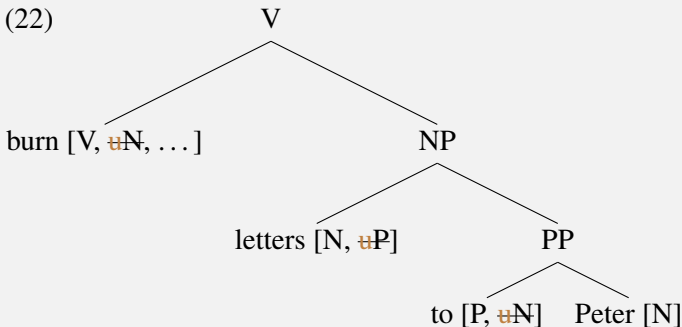
- ⇨ So there's nothing to stop something from being simultaneously Maximal and Minimal
- ☞ A simple lexical item with no selectional features, like **Peter**, will be both at the same time.

Note also that labeling a constituent as a PP or NP is just helpful notation and has no theoretical significance.

- ☞ The fact that an object is maximal is determined by its feature specification and nothing else.

Complements

A particular kind of structure arises when we Merge a simple lexical item with a category that it selects:



Peter is the **complement** of **to**, PP the complement of **letters**, NP the complement of **burn**...

The complement is the first thing selected by a head which
Merges with that head.

- ☞ Note that being a complement has nothing directly to do with linear order.
- ☞ So in many languages, complements come before heads:

(23) Hanako ga Taro o tataku. (Japanese)
Hanako nom Taro acc hit
'Hanako is hitting Taro.'

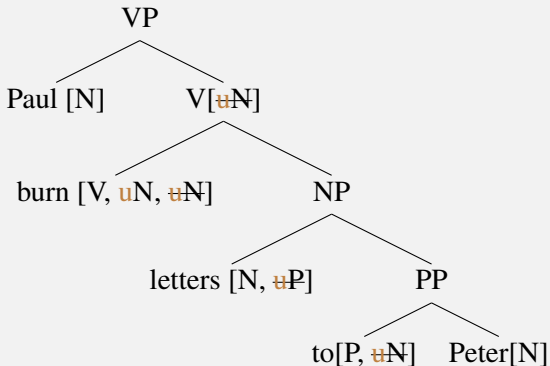
We unfortunately won't get a chance to talk in detail about how to deal with differences like this, but it's important to note that it exists.

Specifiers

Something different happens when we add the subject:

(24) Paul burns letters to Peter.

(25)



Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

Complements

Specifiers

Adjuncts

The structure of
VP

- **Paul** is selected here by one of the [uN] features on **burn**. But it doesn't Merge directly with **burn**.
- Instead, it Merges with a higher projection, after **burn** has already Merged with **letters to Peter**.
- The thing that **Paul** merges with is neither maximal nor minimal. We'll call it an **intermediate projection**, which we sometimes indicate as \bar{X} or X' , pronounced X-bar.
- ☞ Something which is selected by and Merges with an \bar{X} level projection is called a **specifier**.

Adjuncts

(26) Ellie demonized Anna every day.

- every day doesn't seem to get a θ -role from demonize or from anything else, i.e. it isn't selected.
- Instead of supplying necessary information, filling in a hole in a predicate, it gives extra information, modifying what would already be a proposition.

Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

Complements

Specifiers

Adjuncts

The structure of
VP

As such, **every day** is entirely optional, and could be left off or replaced by any number of other modifiers:

- (27) **Ellie demonized Anna at the club.**
- (28) **Ellie demonized Anna almost certainly.**
- (29) **Ellie demonized Anna very happily.**

☞ We call such modifiers **adjuncts**.

Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

Complements

Specifiers

Adjuncts

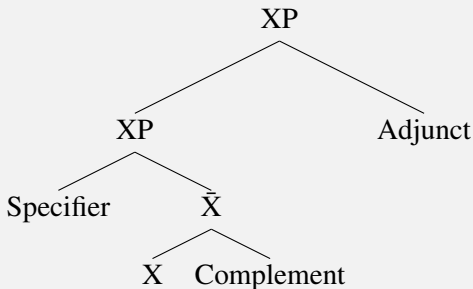
The structure of
VP

Note that the concept of adjunct, like complement and specifier, is **not** about syntactic category.

- **every day** is an NP, **at the club** is a PP, and **almost certainly** and **very happily** are AdvPs.
- So just like we have a structural definition of complement and specifier, we'll need a structural definition of adjunct.

Here's the approach we'll take:

(30)



- Adjuncts are sisters and daughters of maximal projections.
- This reflects the fact that no selectional feature is checked, and captures the fact that adjuncts are optional and recursive, i.e. you can have as many as you want.
- It also is in line with the idea that you can't add adjuncts to a structure until all of its dependent features have been checked.

Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

Complements

Specifiers

Adjuncts

The structure of
VP

If adding an adjunct doesn't affect the syntactic features of the object it's added to, it shouldn't affect that object's syntactic distribution. Here's some evidence that this is correct:

- (31) Burn the letters (quickly)!
- (32) I burnt the letters (quickly).
- (33) I plan to burn the letters (quickly).
- (34) * Burn the letters (quickly) is the best thing to do.
- (35) Burning the letters (quickly) is the best thing to do.

Adjuncts raise a technical issue that we'll have to worry about:

- Until now, the determination of the head when two items Merge – and hence what projects – has been based on which object triggers Merge via its selectional feature.
- But adjunction doesn't involve selection, so our existing procedure won't work here.
- Clearly we want the object adjoined to – and not the adjunct – to project its features, but of course it can in no way be responsible for the adjunction operation.

There are a number of different ways to approach this issue, none of which is obviously better than the others, so at this point we won't propose anything that pretends to be insightful.

- Instead we'll simply stipulate that adjuncts are marked with a feature [Adjunct] – a plain admission that there is something we don't yet understand here.
- Then for completeness we can revise our definition of headedness, again with the ugly disjunctive formulation serving as an indication that we have work to do:

(36) **Headedness** (revised)

When two objects Merge, the object that will project its features to the newly created object is

- i. the one that selects the other, if selection is involved
- ii. otherwise, the one that does not bear an [Adjunct] feature.

The structure of VP

Introduction to
Syntax

Lecture 7:
Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

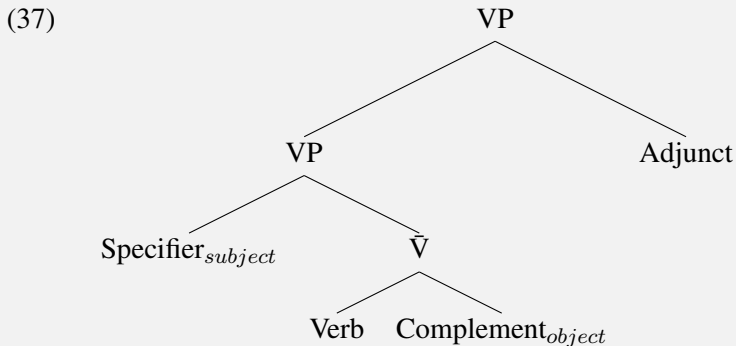
The structure of
VP

Introducing v

Motivating v

Getting v together with
 V

Here's how a full VP might look in our system:



Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

The structure of
VP

Introducing *v*

Motivating *v*

Getting *v* together with
V

This raises an issue:

- ☞ To derive such a VP with two arguments, the V would have to have two [uN] selectional features.

But this raises another problem!

Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

The structure of
VP

Introducing v

Motivating v

Getting v together with
V

The problem is:

- ? How do we make sure that the features get checked in the right order, i.e. how do we get the correct θ -roles to the complement and specifier?

As usual, there are many ways we could proceed, and it is difficult to know in advance which approach will be correct. Here are some options:

- (i) Introduce a new mechanism to manage the θ -roles that lexical items carry and ensure that they are always assigned in the correct order.
 - (ii) A different solution which avoids adding any new mechanisms and allows us to maintain an extremely simple account of θ -role assignment, but adds complexity to the structure of the verb phrase (this is the strategy we'll follow!).
- ☞ Ultimately, the choice between (i) vs. (ii) will depend on what additional empirical coverage can be achieved with their respective additional complexities.

Introducing v

Our goal is to retain the following maximally simple version of θ -role assignment:

- (38) **Blind θ -role assignment:**
 θ -roles are assigned blindly upon Merge of a θ -role assigner with an object of the correct syntactic category.

Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

The structure of
VP

Introducing v

Motivating v

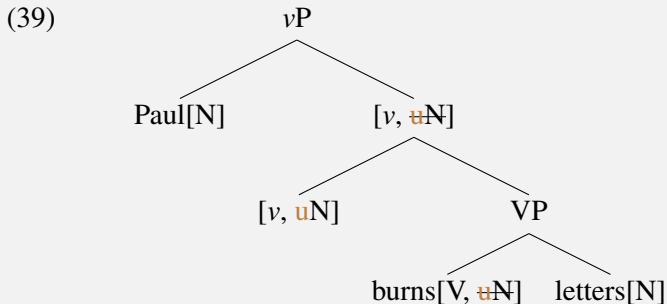
Getting v together with
 V

- I.e. θ -role assignment cares only about selection and the related categorial features.
- This means that, if a head selects two instances of the same category, there is no way to distinguish them in terms of θ -role.
- This implies that a single head should never select two instances of the same category.

So what do we do about transitive verbs?

- ☞ If we have two NPs getting θ -roles, and each head can only assign a θ -role to one instance of a particular syntactic category, then there must be two heads.
- ☞ I.e. we're going to propose that typical verbs actually involve two syntactic heads, one selecting the object, and the other selecting the subject.
- ☞ We'll continue to call the lower head V , and this is where we'll put the distinct lexical element.
- ☞ We'll call the upper head v , pronounced 'little vee'.

Here's what a simple transitive will look like then:



- **burns** has a **uN** feature, so it can Merge with the object **letters** and assign to it its θ -role.
- v also has a **uN** feature. It first Merges with VP , projecting its features up to the next level, where it Merges with **Paul** and assigns to it its θ -role.

Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

The structure of
VP

Introducing v

Motivating v

Getting v together with
 V

That solves the θ -role problem, but raises some new questions:

- ? What is the independent motivation for v ? Does it contribute anything else beyond the extra θ -role?
- ? What makes v Merge with V , and how do we determine that v should be the head rather than V ?
- ? How does the v - V combination behave for purposes of pronunciation?

Motivating *v*

Consider first that basic constituency tests show that the object is closer to the verb than the subject is:

- (40) [Eat a mongoose] I never would.
- (41) * [I/me eat] never would a mongoose.
- (42) Dolores will [eat a mongoose] and I will [eat-a mongoose] too.
- (43) * Will [you eat] a mongoose or will [you-eat] a wolverine?

Introduction to Syntax

Lecture 7: Selection

Merge and
c-selection

Back to heads

The internal
structure of
phrases

The structure of
VP

Introducing v

Motivating v

Getting v together with
 V

- ☞ Now, we can model this asymmetry by simply saying that the verb first Merges with the object and then the subject.
- ☞ But we again have no explanation for why that should always be the order if subject and object are symmetrical in both getting their θ -roles from the verb.
- ☞ On the other hand, if the subject gets its θ -role from a distinct head v which Merges with VP, we can explain the constituency facts rather than describing them.

There's also an asymmetry between subjects and objects when it comes to verbal semantics:

- (44)
 - a. throw a baseball
 - b. throw support behind a candidate
 - c. throw a party
 - d. throw a fit
- (45)
 - a. take a book from the shelf
 - b. take a bus to New York
 - c. take a nap
 - d. take an aspirin
 - e. take a letter in shorthand
- (46)
 - a. kill a cockroach
 - b. kill a conversation
 - c. kill an evening watching TV
 - d. kill a bottle
 - e. kill an audience

What does all of that tell us?

- The choice of object can apparently affect the meaning of a verb in arbitrary ways.
- Note crucially that the effects here are not (all) straightforward idioms involving specific lexical items.
- E.g. the special meaning of **kill an evening** is available with any object with the right kind of semantics.
- This makes sense if the object really is an argument of the verb, since the semantics of the verb can be made sensitive to the semantics of its arguments.

Crucially, examples with similar effects based on the choice of subject are apparently lacking.

- ⇒ So we have another asymmetry between subjects and objects in their relationship with the verb.

Again, we can make sense of this in terms of our new structure involving v :

- ☞ The facts in 44–46 constitute evidence in favor of objects being arguments of the verb.
- ☞ Thus the lack of similar facts with subjects is evidence **against** them being arguments of the verb.
- ☞ If they are instead arguments of v , this asymmetry is accounted for.

Getting v together with V

So what regulates v Merging with V ?

- The obvious possibility to consider would be that v also bears a [uV] feature, i.e. it selects for a V .
- Given our definition of headedness, this would ensure that it is v which projects after Merge and not V .

But there are some issues with such an approach:

- 1 If v selects both for a V and for an N (i.e. the subject), what ensures that it Merges with V first?
- 2 If the relevant dependent features are only on v , this will ensure that v doesn't appear without V , but there's nothing to stop V showing up without v (and without a subject). I.e. something like (47) should be grammatical:

(47) * **Burned letters.**

We need a way to express the idea that the structure of a sentence built around a verb must have **both** *v* and *V*.

- We may ultimately be able to do this in terms of the right combination of selectional (and other dependent) features localized to individual syntactic objects.
- Another possibility is that the requirement that both elements be present is not imposed by the syntax, but falls out of restrictions on semantic interpretation.
- But since it is not clear at this point how either of these solutions would work, we need something more direct.

So we're going to propose a new component to our theory:

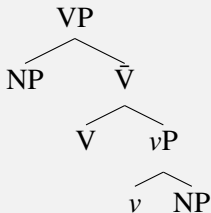
(48) **Clausal Hierarchy of Projections (1st version)**

$v > V$

- This says for now that a complete clause must (at least) involve a v taking a VP complement. (We'll add more projections later.)
- The idea is that v and V are integral parts of a single system, where v essentially extends what V has begun.

The Hierarchy places a new restriction on structures, in addition to those placed by full interpretation etc. Structures like the following that don't respect it are ruled out:

(49)



The HoP adds considerable complexity to our theory, and at this point it is a pure stipulation.

- Ultimately we will need to provide an explanatory basis for the hierarchy and justify the form it takes.
- Or we'll need to replace it with something more explanatory which can cover the same empirical ground.
- But for now, we need something to do this work, and the HoP is, at least, a relatively simple stipulation, and should help to clarify what needs to be explained rather than concealing it.

Note finally that we haven't said anything about the role of *v* in the pronunciation of sentences:

- In the examples we've looked at so far, it doesn't seem to have any effect, not being pronounced itself or changing the pronunciation of anything else.
- This would be sort of surprising if it were generally true, and in fact there is some reason to think that this head **is** pronounced in certain instances, e.g. as the verbalizing suffix **-ize** in **vaporize**.
- We're going to set this issue aside for now, but we'll hopefully come back to it later when we start developing the idea of **Movement**.