Hedde Zeijlstra
(University of Goettingen)

Principles of Projection

May 24, 2018
Colloquium talk
Universität Konstanz

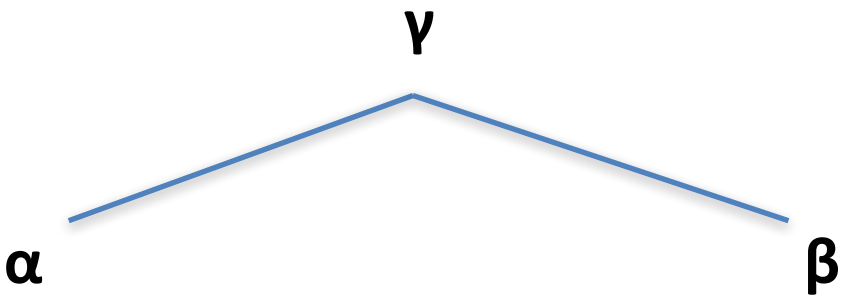
GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Göttingen
Campus

Question

I. Question

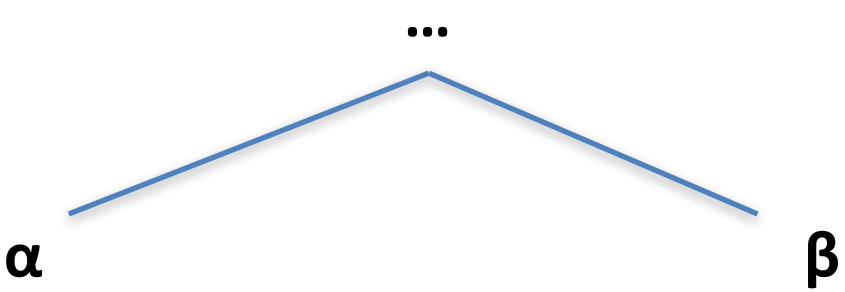
What, if any, is the label of γ?



Is labelling required, and if so, is there a unified labeling algorithm that will suffice for all cases?

I. Question

Two approaches to the central question:



- Why should the top node receive some content (i.e. why must α or β percolate)?

Most theories of Merge nowadays state that the output of Merge does not provide a label. At the same time, narrow syntax and/or the interface requires a label to be present. Hence, an independent labeling algorithm is needed.

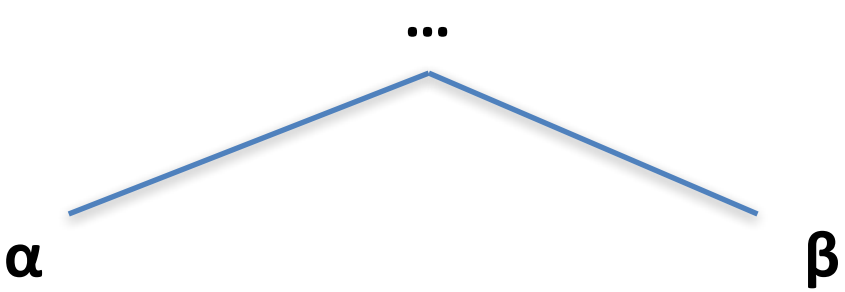
I. Question

Several suggestions for such a Labeling Algorithm have been proposed:

- Chomsky (1995, 2000, 2008): γ is either α or β (see also Cecchetto & Donati 2010)
- Collins (2002): no label at all
- Adger (2013): something else than α or β
- Chomsky (2013, 2014): either α, β or a shared feature of α and β

I. Question

Two approaches to the central question:



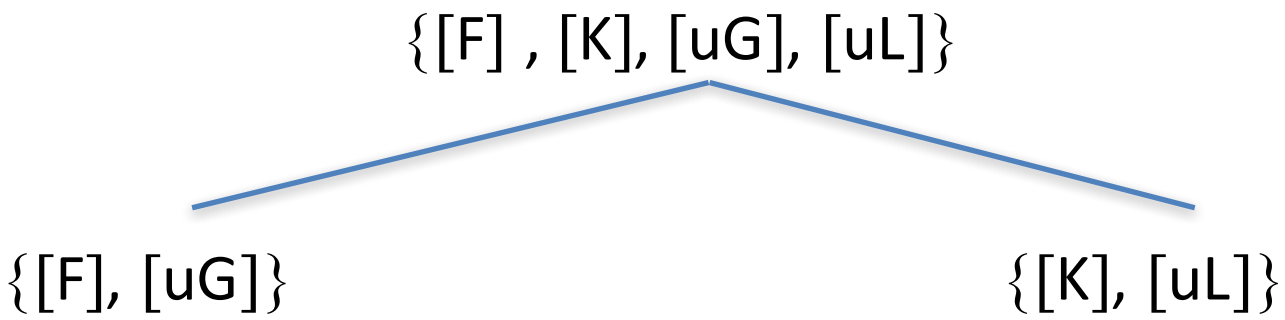
- Why don't α and β both percolate?

Most theories of labelling focus on the gain of information; hardly any theory focuses on the loss of information (Neeleman & Van der Koot 2002 being a notable exception).

I. Question

Principle of Containment of Syntactic Information:
syntactic information may not disappear in the tree (i.e. all syntactic features percolate).

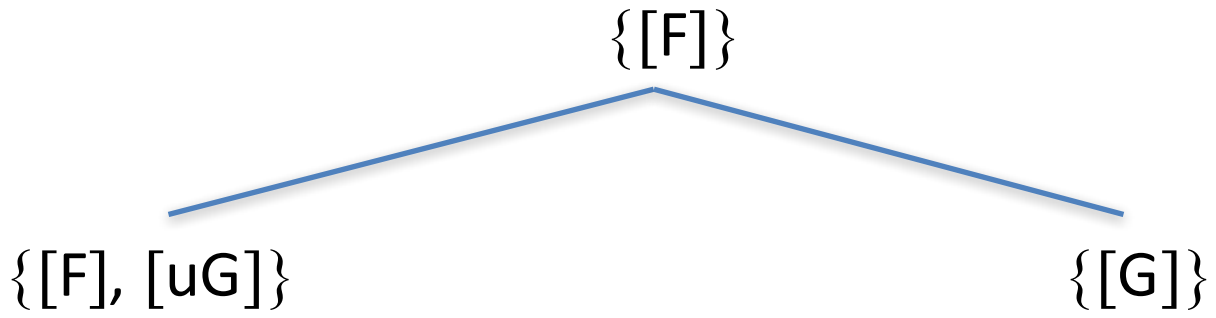
- Merge results in the union of the sets of dependent and independent features of its daughters.



I. Question

This talk: to what extent can labeling follow from constraints on feature percolation?

- Every feature percolates up, unless one interpretable feature and one uninterpretable feature stand in a sisterhood relations; then neither of these two percolate.



Background

II. Background

What features are involved in labelling and other syntactic operations?

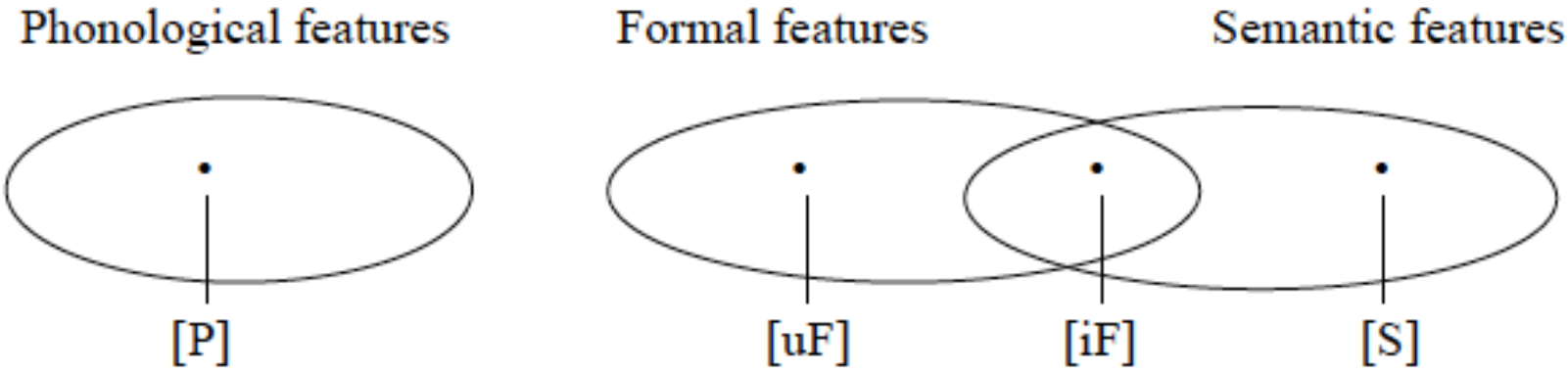
- Chomsky (1995 et seq): minimally interpretable and uninterpretable formal features.
- Later on, a variety of other features have been postulated: categorial features, selectional features, edge features, EPP-features, fully uninterpretable features, etc.

Ideally, any theory of syntactic dependencies must be based on the smallest set of types of features: minimally, interpretable and uninterpretable formal features.

II. Background

Chomsky (1995, 2000, 2001):

- The set of formal features and the set of semantic features intersect:

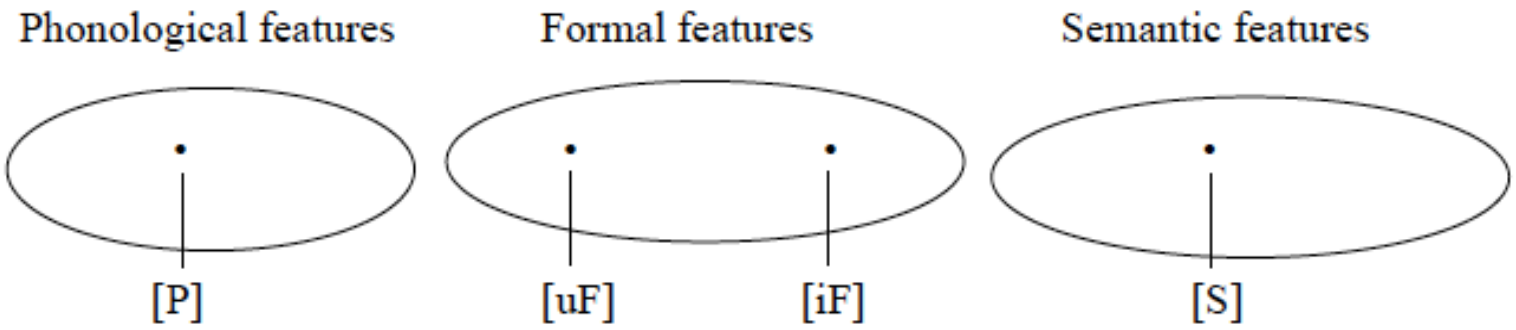


- Uninterpretable features must be deleted under checking with interpretable features

II. Background

Zeijlstra (2014):

- The sets of formal features and semantic features do not intersect:



- Language acquisition determines that most, but crucially not all, lexical items with the semantics of F are assigned a formal feature [iF].
- It is a property of syntax that [uF] features need to be c-checked by [iF] features (otherwise the sentence crashes).

II. Background

Advantages:

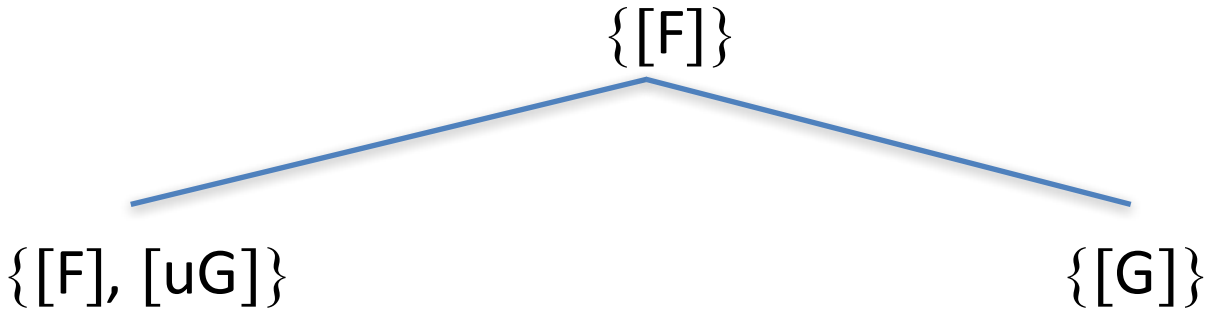
- Interpretable features are no longer ‘interpretable’; they are purely syntactic features.
- A better terminology would be dependent and independent features.
- If independent features are void of semantic content, they can be unified with categorial features.
- Lexical items then contain only two sets of formal features: dependent and independent categorial features:

		/on/
Example:	On	[P],[uD]
		LOC_ON

Proposal

III. Proposal

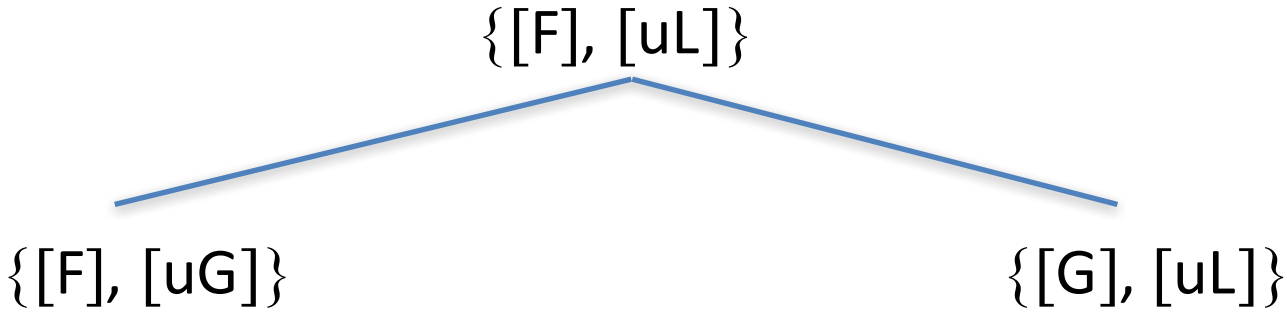
Since both dependent and independent formal features are categorial features, merger of an element with a dependent and an element with a matching independent formal feature, results in removal of both, following standard rules in categorial grammar:



Rule: Let A and B be two sets of formal features. For any pair [F]-[uF], such that [F]∈A and [uF]∈B, or [F]∈B and [uF]∈A, neither [uF] nor [F] percolate; all other features do percolate.

III. Proposal

Since both dependent and independent formal features are categorial features, merger of an element with a dependent and an element with a matching independent formal feature, results in removal of both, following standard rules in categorial grammar:

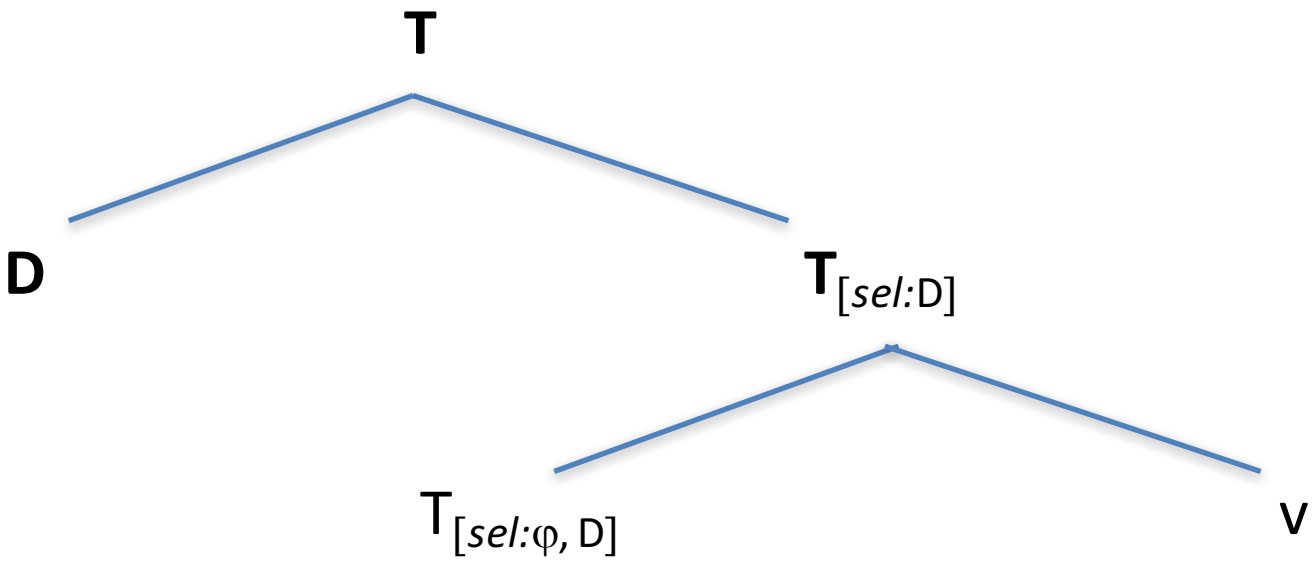


Rule: Let A and B be two sets of formal features. For any pair [F]-[uF], such that [F]∈A and [uF]∈B, or [F]∈B and [uF]∈A, neither [uF] nor [F] percolate; all other features do percolate.

IV. Selection by projection

This view is very similar to the orginal view that *what selects projects (*Chomsky 1995, 2000, Adger 2003, Boeckx 2008)*:*

- If α, β Merge and if α selects for (a feature present on) β, α is the label of the merger of α and β.



IV. Selection by projection

Advantages:

- Unified LA (except for adjunction): applies to Head-Comp configurations and to both internally and externally specifiers
- Determines the syntactic behaviour of the top node

IV. Selection by projection

Problems:

- Unmotivated why the selector should project
- Does not apply to adjunction
- Ordering problem (why not have T first merge with DP and then with vP)?
- Various selectional features are taken to be semantic features (s-selection vs. c-selection). But why would semantic features play a role in syntax?
- What happens if the complement also selects a feature of the head (e.g. case feature checking to sisters of V or P)?
- If selectional features are uninterpretable features, why can't they be checked on a distance (via Agree)?

V. Aims

To formulate an account of selection and labeling that:

- Motivates why the selector projects
- Can account for labelling in all configurations (including adjunction)
- Can overcome the ordering problem
- Can circumvent the problems relating s-selection vs c-selection and mutual selection (e.g. with case)
- Can account for the differences between selection and (long-distance) Agree

Application

The selector projects:
labelling of head-complement configurations and configurations involving specifiers

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Göttingen
Campus

VI. Head-Complements

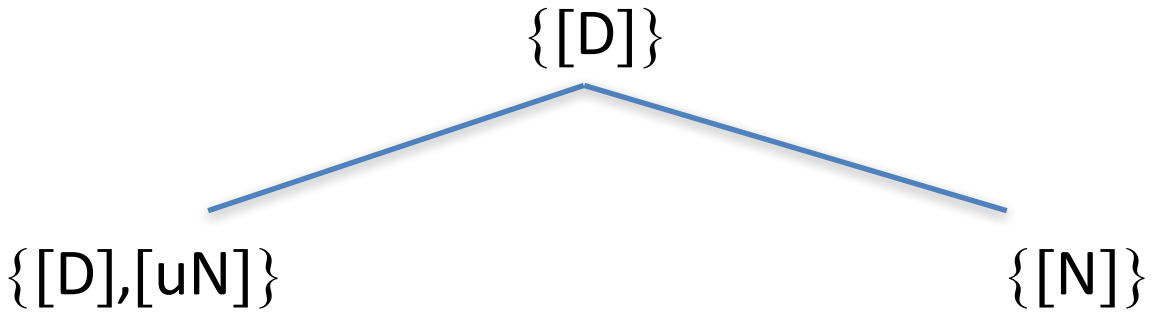
Given the categorial status of both dependent and independent features, sisterhood of two matching dependent and independent features results in removal of both from the top node.

This accounts for the fact that:

- The selected feature does not project
- the selecting feature does not project
- that all other features still project.

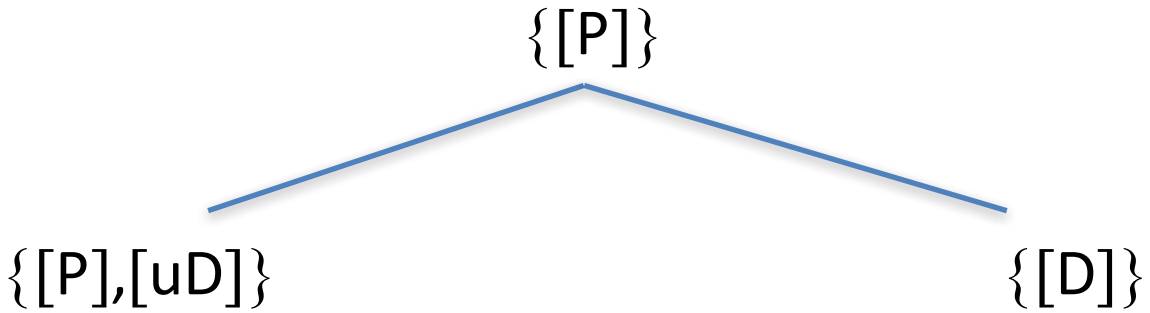
VI. Head-Complements

Example: D selecting NP



VI. Head-Complements

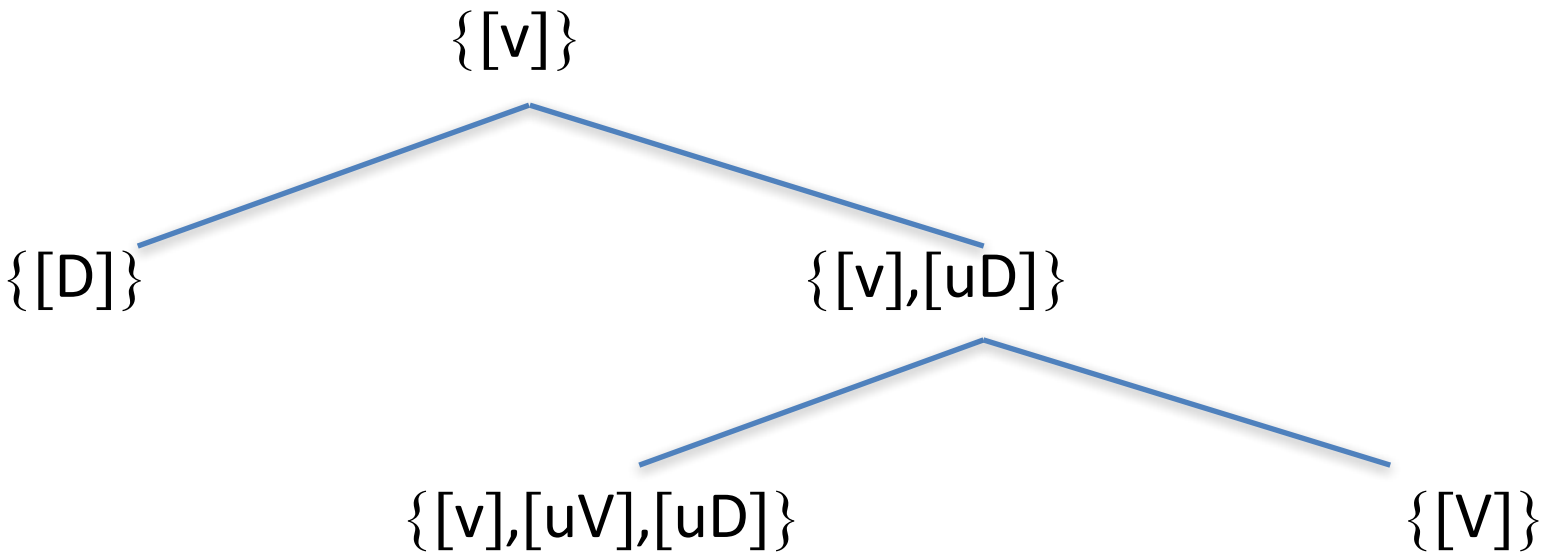
Example: P selecting DP



VII. Specifiers

Following the earlier proposals, the same mechanism applies to specifiers merging with bar-levels.

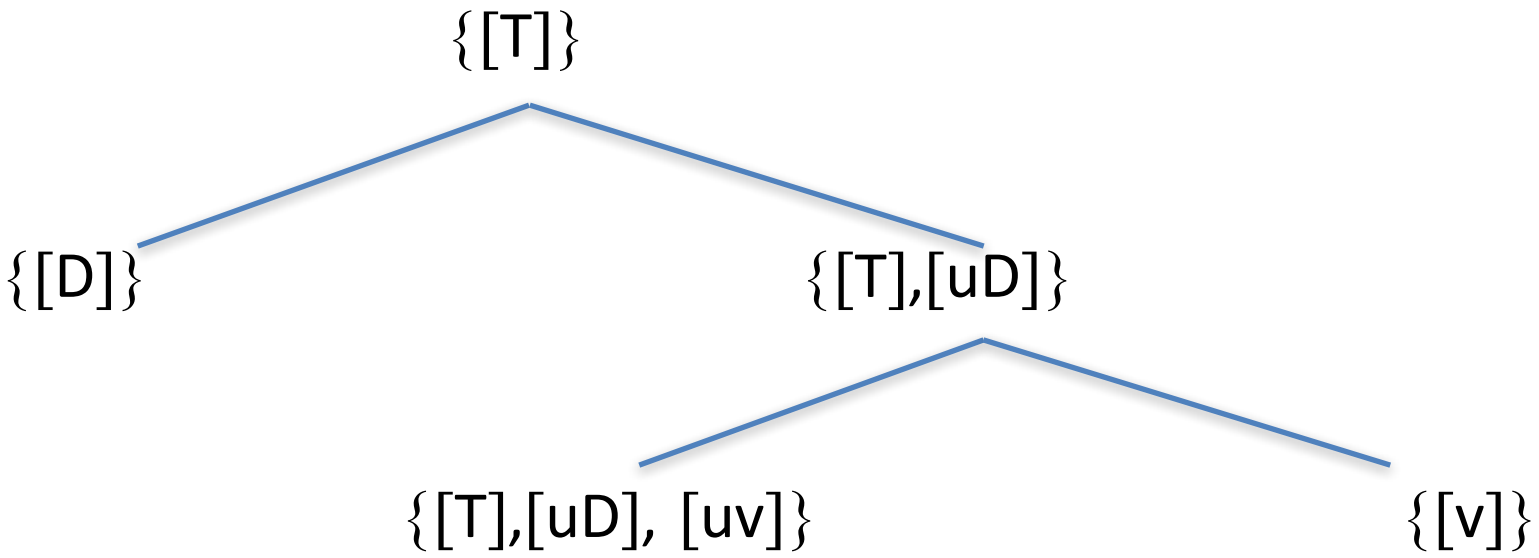
Example: vP (involving External Merge):



VII. Specifiers

Following the earlier proposals, the same mechanism applies to specifiers merging with bar-levels.

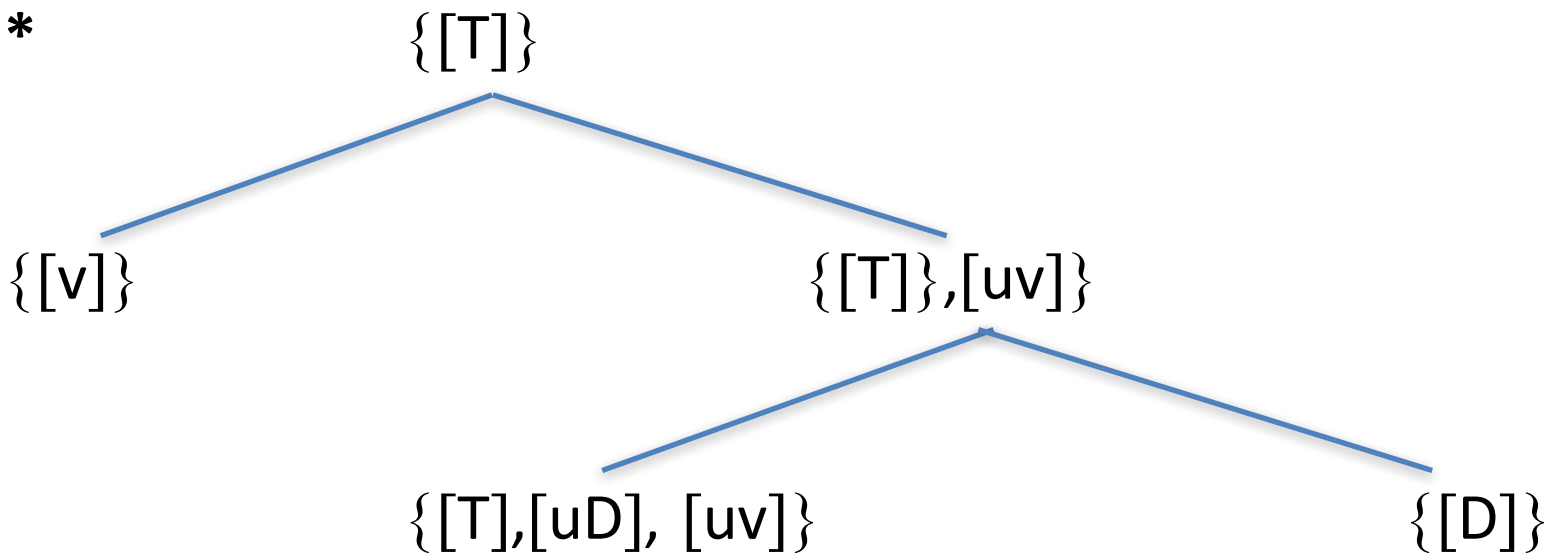
Example: TP (involving Internal Merge):



VII. Specifiers

Hence, the proposal so far explains why the selector probes, and also how labeling works for Head-Complement relations and specifiers (irrespective of their original position).

- However, an problem for unifying labelling in both Head-Comp and Spec-Head configurations concerns the ordering of the fulfilment of the selectional requirements.



VII. Specifiers

Two solutions suggest themselves:

- Add ordering diacritics: for some reason, the [uv] feature on the T-head needs to be checked first, and only then the [uD] feature (same for [uV] and [uD] on v).

- Rule out the unwanted orders by means of narrow syntactic and/or interface conditions: the semantics of T require a semantic complement that can only be realized by a vP, not by a DP; mutatis mutandis the same for v.

VII. Specifiers

Theoretical consequences:

- Apart from the addition of (ordering) diacritics, the first solution runs against the more central principle of the proposal, namely that the set of (in)dependent features present on a lexical item is unordered.

- Not immediately clear how every unwanted selection order can be ruled out by narrow syntax or at the interfaces, something required by the second solution.

VII. Specifiers

Empirical consequences: the two solutions make clear different empirical predictions:

- Under the ordered features solution flexible selection orders are never possible;
- Under the interface solution flexible selection orders are expected when the interfaces do not rule them out;

!!! Spoiler alert !!!

VII. Specifiers

Empirical consequences: the two solutions make clear different empirical predictions:

- Under the ordered features solution flexible selection orders are never possible;

- Under the interface solution flexible selection orders are expected when the interfaces do not rule them out;

Spoiler: we will indeed attest cases where selection orderings are flexible.

Adjunction

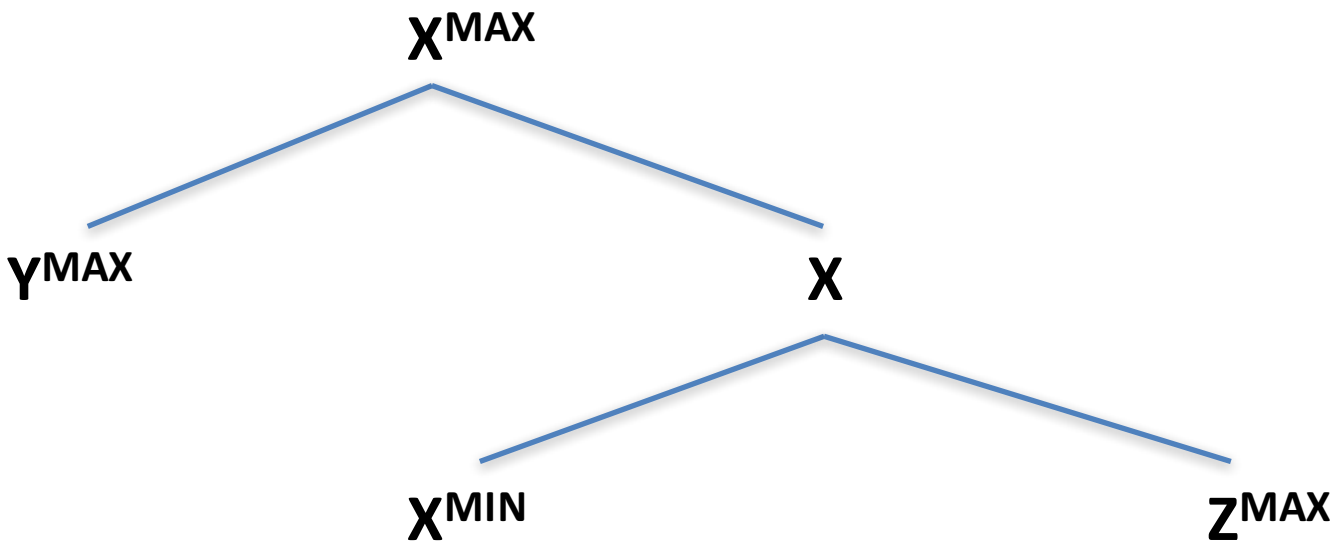
GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Göttingen
Campus

VIII. Adjuncts

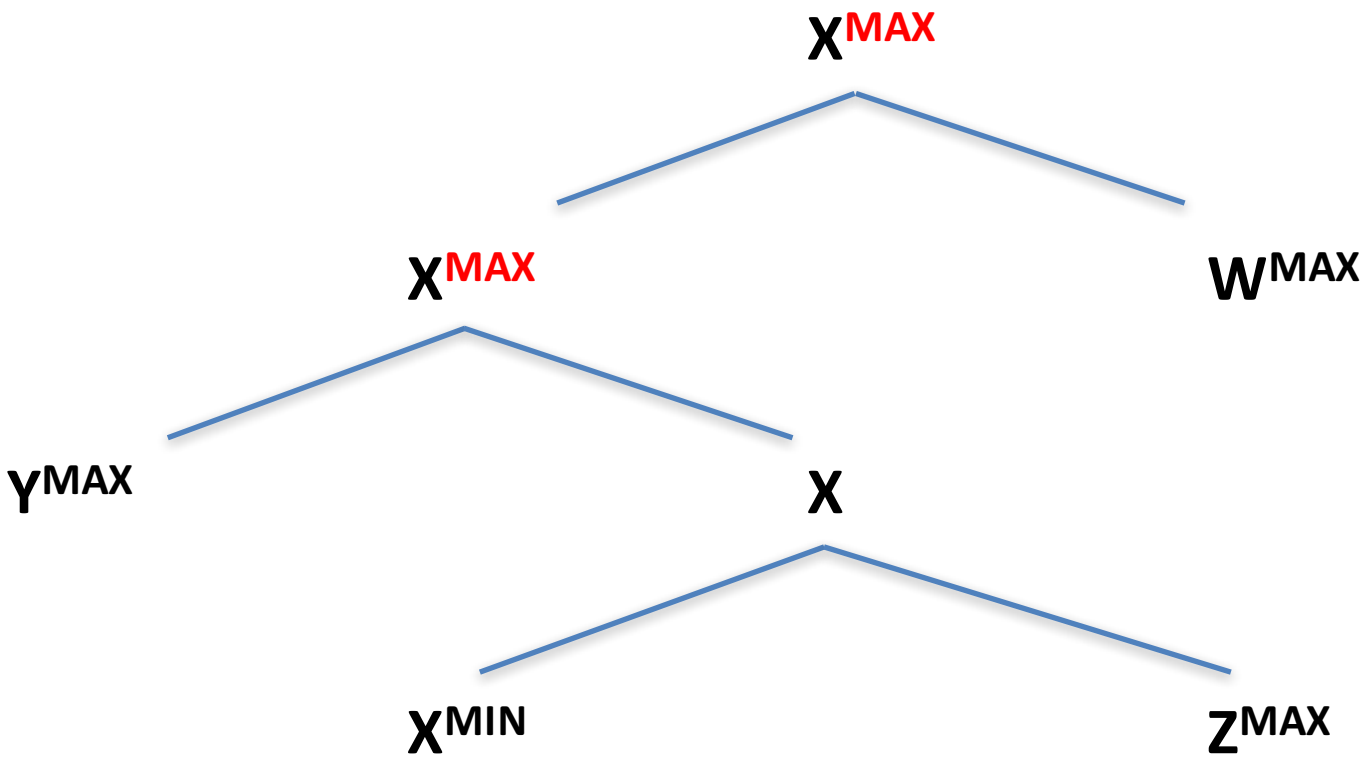
Adjuncts form a notorious problem for labelling under Bare Phrase Structure (cf. Hornstein & Nunes 2009 and references therein).

- Under Bare Phrase Structure Bar levels and Maximal projections are structurally defined:



VIII. Adjuncts

But adjunctions consists of two layers of the same feature that should both count as maximal:



VIII. Adjuncts

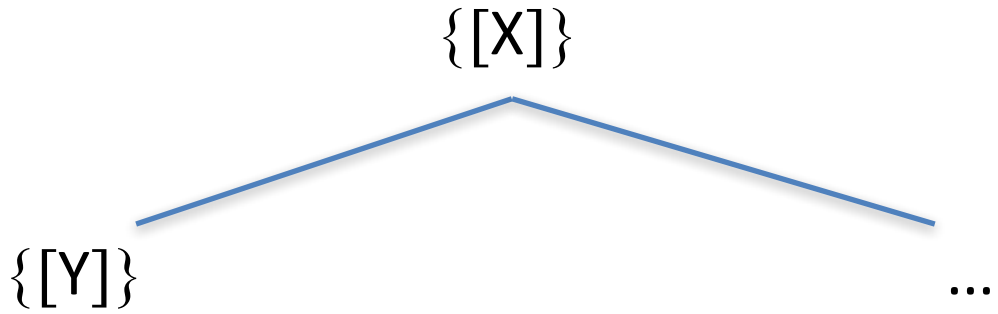
For this reasons, adjuncts have been taken outside the system that derives structures by means of set-merge and labelling:

- Chomsky (2001): Set-Merge vs Pair-Merge
- Lebaux (1989): Late insertion of adjuncts into already labelled structures
- Hornstein & Nunes (2009): Unlabelled adjuncts

All these approaches have been primarily introduced to account for the special status of adjuncts under Bare Phrase Structure.

VIII. Adjuncts

However, under the proposal proposed here, adjunction can be derived under Bare Phrase Structure. Take the following structure

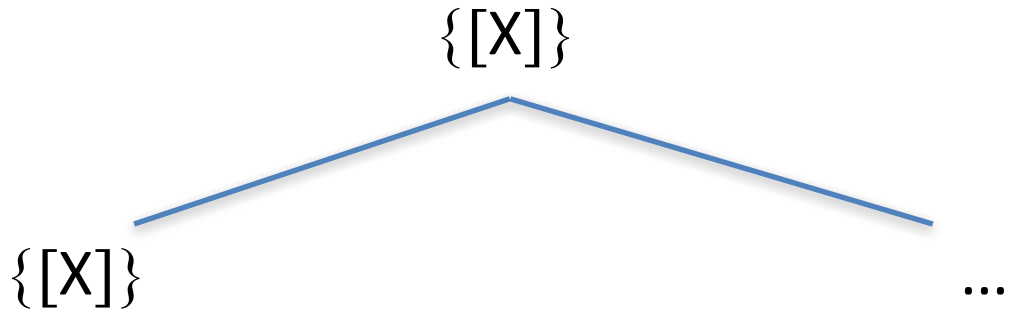


Irrespective of the phrasal status of the elements represented by {[X]} and {[Y]}, we can compute the feature representation of the unknown sister/daughter …, which must be:

{[X], [uY]}

VIII. Adjuncts

Adjuncts are cases where the top node must be featurally identical to one its sisters, otherwise its distribution would not be identical.



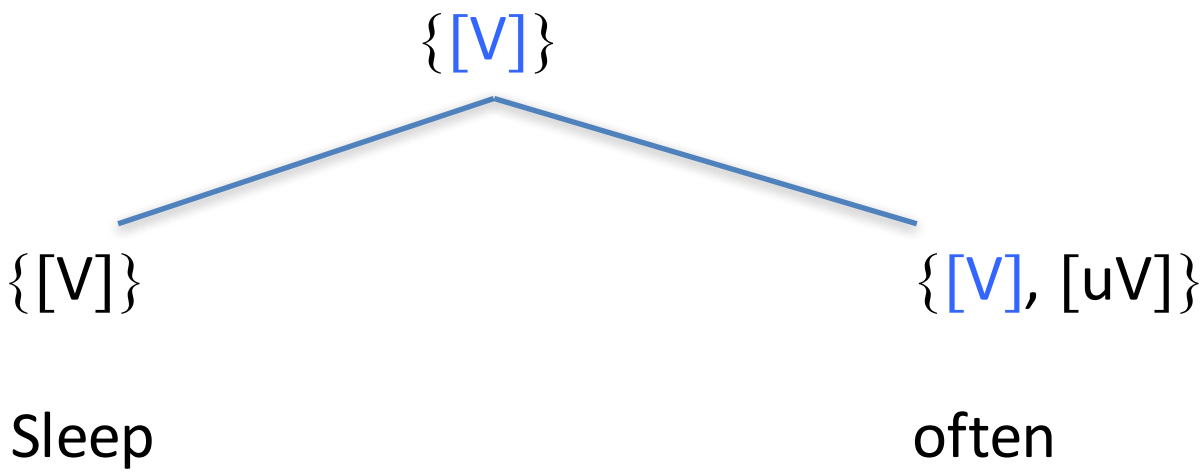
But that means that every X-adjunct, should have a representation:

{[X], [uX]}.

VIII. Adjuncts

VP adjuncts, e.g. adverbs, should then be taken to be elements with a featural representation {[V], [uV]}.

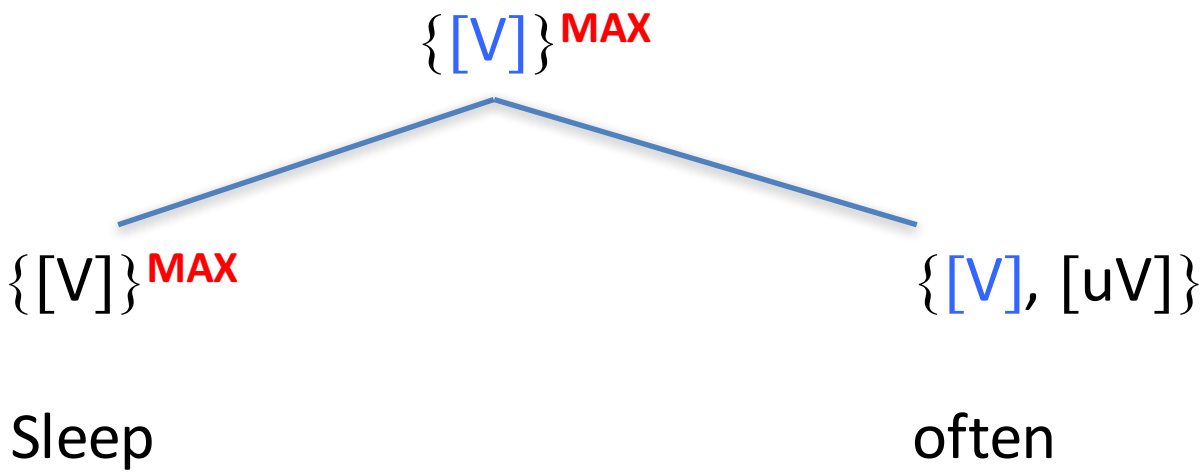
But, as the picture shows, this solves the adjunct problem. In the configuration below, both V-layers are maximal projections (where colours reflect projection lines):



VIII. Adjuncts

VP adjuncts, e.g. adverbs, should then be taken to be elements with a featural representation {[V], [uV]}.

But, as the picture shows, this solves the adjunct problem. In the configuration below, both V-layers are maximal projections (where colours reflect projection lines;



VIII. Adjuncts

Note that, as things stand now, adjuncts must be specified for the phrases they adjoin.

- For VP-adverbs we can indeed say that they are elements with a categorial feature set {**[V], [uV]**}.

- But how about other types of adjuncts (in the verbal domain), such as PP adjuncts?

IX. PPs

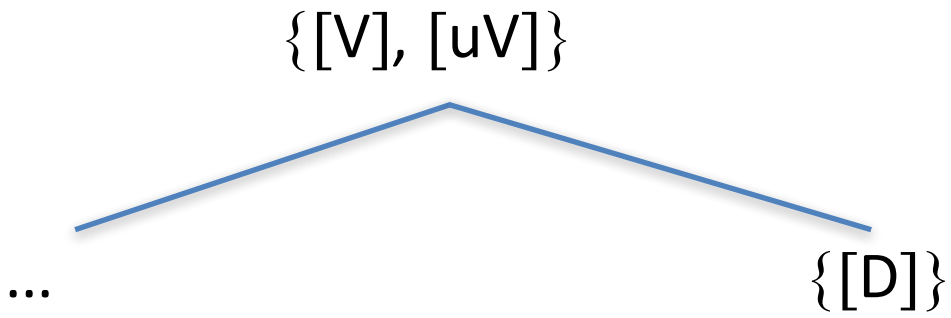
If PPs behave as VP adjuncts, they should be analysed as {[V], [uV]}. But such an analysis gives rise to the following two issues:

- If PPs are {[V], [uV]}, what are Ps?
- PPs may adjoin to (at least) NPs, APs and VPs.

I will address both questions in this talk, starting with the first one.

IX. PPs

P selects DP

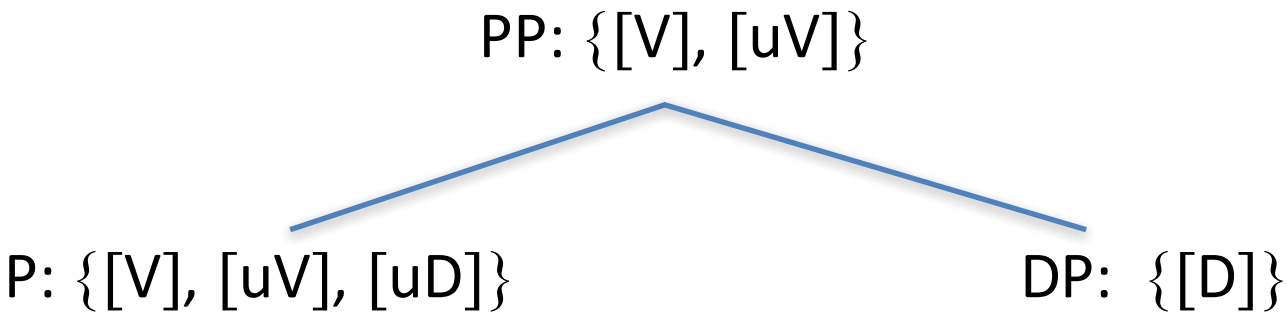


If a PP is {[V}, [uV]}, then P must be:

{**[V], [uV], [uD]**}.

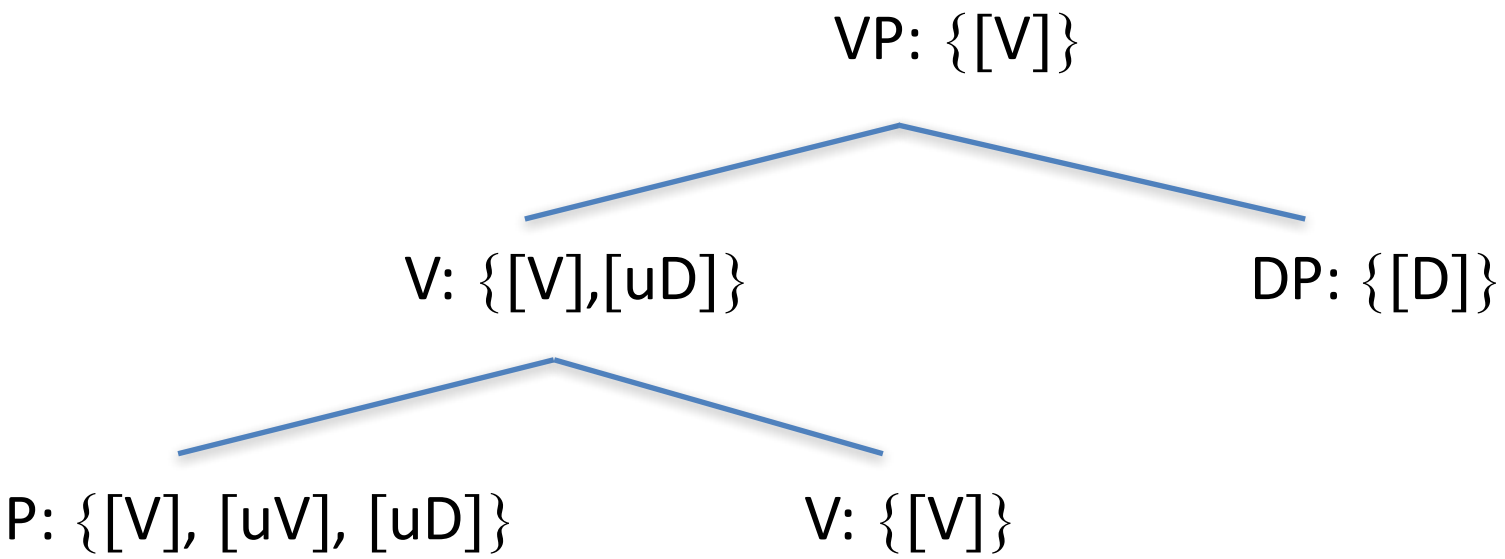
IX. PPs

Ps selecting DPs



But now the ordering problem re-appears: why could P, then, not first select / merge with a V, and then with a DP? The proposal predicts the following to be grammatical:

IX. PPs



But how bad are these structures?

IX. PPs

Many languages allow prepositions to form syntactic structures with verbs (so called particle verb constructions), before they merge with their arguments.

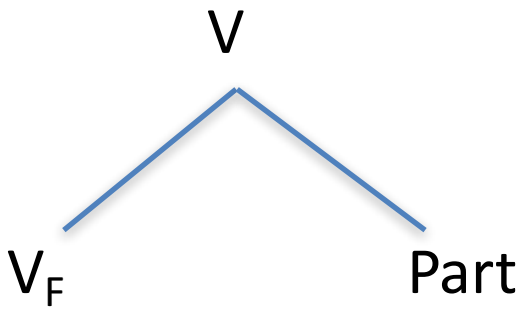
- [[Eat up] [the sandwich]]
- Ich rufe [Marie [an-~~rufe~~]]

Even though such constructions are strongly constrained (not every preposition and verb can form a particle verb), the fact that some can shows that an analysis of prepositions being able to select verbs and jointly select a subsequent argument is necessary.

IX. PPs

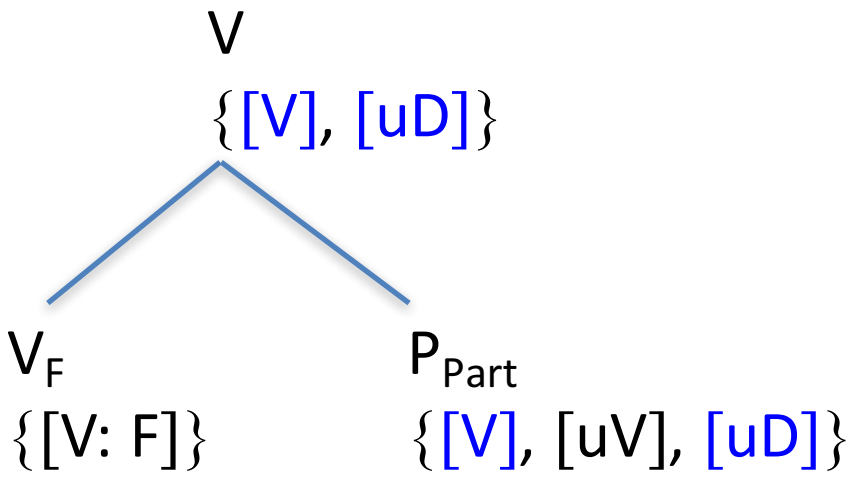
According to Van Riemsdijk (1978), Baker (1988), Koopman (1995), Neeleman (1994, 2002), Zeller (2001), among many others, particle verbs are complex verbal heads:

- Zeller (2001): particle verbs are complex heads, where the verbal subfeatures of the verb do not percolate to the verb-particle complex (as only the verbal part can receive inflectional morphology and may undergo movement by itself):



IX. PPs

This is indeed what is derived by the proposal: the verb is adjoined by the prepositional particle, but the verbal feature on the top node stems from the particle, not form the verb itself.



The subfeatures of the verbal part of the complex verb do not percolate to the higher node.

IX. PPs

The existence of particle verbs

- thus supports the proposed analysis;
- and forms an empirical argument for flexible selectional ordering (and therefore for unordered feature sets as categorial representations), showing that constraints on selectional ordering must follow from the interfaces (with semantics/phonology) or other syntactic properties, and not be encoded on the selectional features themselves; when the interfaces do not rule out one of two logically possible orderings, both orderings are indeed ruled in.

IX. PPs

Summing up:

- In this way, the system meets several of the desiderata for any labelling account: it can account for all labelling configurations (including adjunction) and it can circumvent the ordering problem.

- And, of course, the question how to deal with the fact that PPs can also modify NPs and APs is also still in need of explanation.

- However, the difference between Agree and selection, as well as various issues concerning VP-internal labeling, are still in need of explanation.

Selection in the V-domain

X. Selection in the V-domain

So far, the proposal predicts that every selector forms the label. This is, however, not particularly clear for VP-internal selection.

- Are the selectional requirements of a V s-selectional or c-selectional requirements? For this proposal it is necessary that V selects its arguments syntactically, not semantically. Otherwise the label of merger of V and D can never be V.

- How do case properties fill in? For this proposal, again, it is necessary that selected DPs do not check their selecting heads in return (as is often assumed in case theory).

X. Selection in the V-domain

The idea that verbs do not c-select, but rather s-select (DP) arguments comes from two reasons:

- Vs often select non-DP arguments, but PP or CP arguments:

  Mary knows Bill
  Mary knows about Peter
  Mary knows that Theo is ill

X. Selection in the V-domain

The idea that verbs do not c-select, but rather s-select (DP) arguments comes from two reasons:

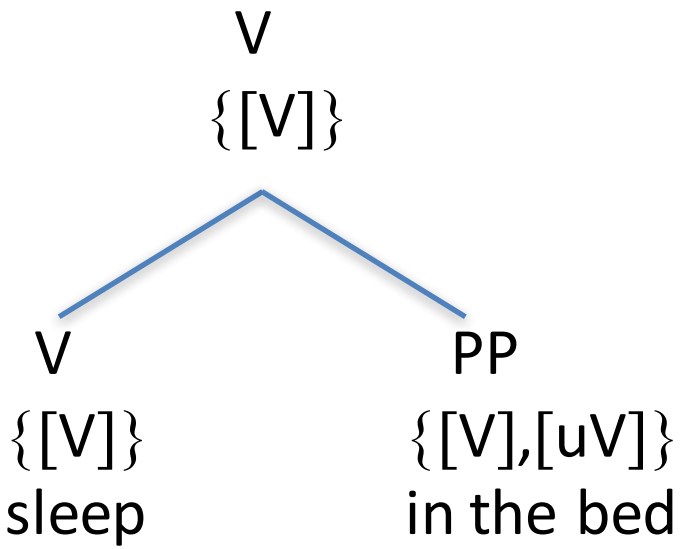
- Not every DP is selected as an argument

  John works this week

As the latter type of argument could be analysed in terms of covert prepositions ([$_{PP}$ P [$_{DP}$ this week]]) (cf. Corver 2015), we focus on the first argument, which is generally taken to be the strongest argument against c-selection.

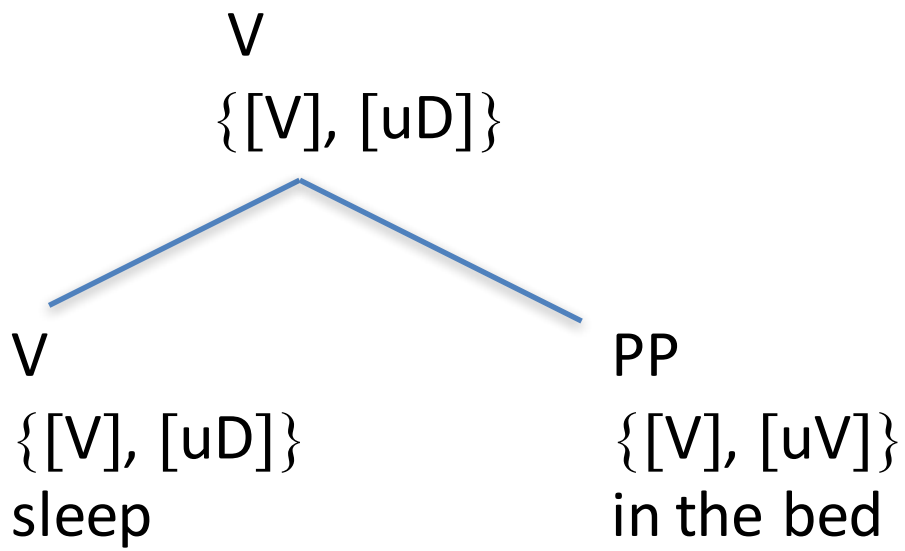
XI. PP arguments

However, as shown before, PP arguments do not require any V to select them; PPs select VPs



XI. PP arguments

In fact, even if V selects for a DP argument, it can be modified by a PP argument as well:



XI. PP arguments

Under this proposal, Verbs do not syntactically select for PP arguments (or adjuncts), but they can be modified by PPs; (verbal PPs select V-constituents).

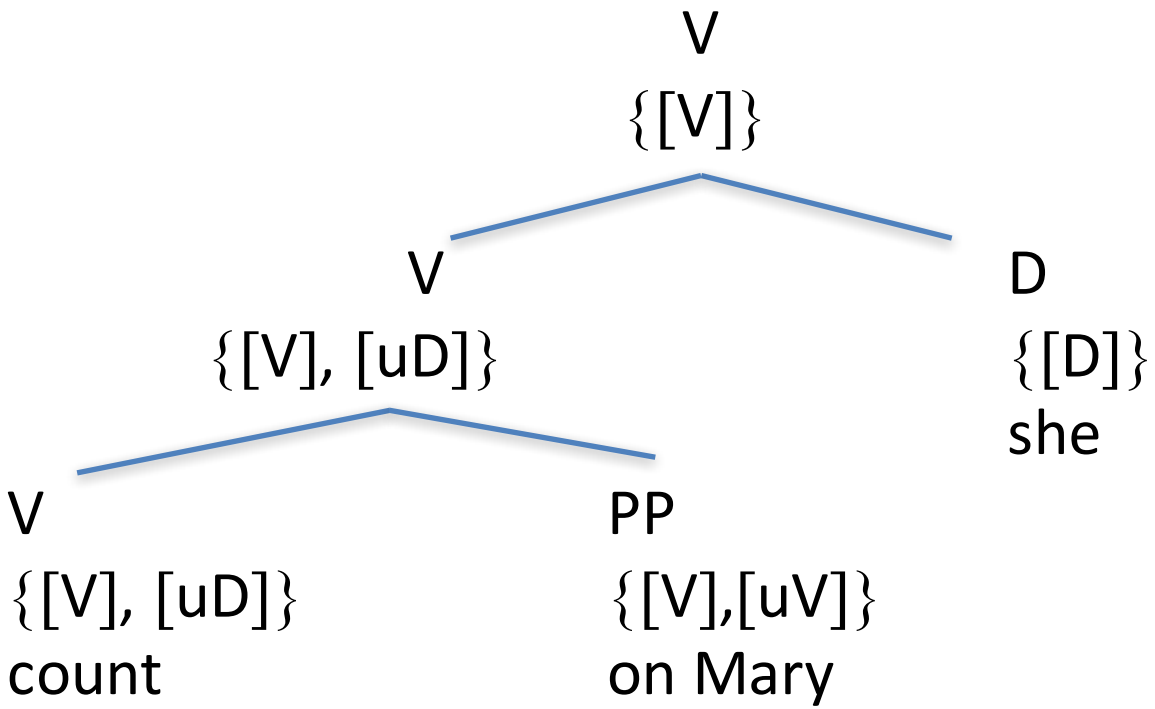
- PP arguments and PP adjuncts are syntactically identical; their differences follow from the (different) semantic properties of argument and adjunct PPs and the the verb.

- At the same time the question arises if, and of so, how the argument-adjunct distinction of PPs is syntactically reflected.

XI. PP arguments

PP adjuncts and PP arguments of a verb can be syntactically distinguished, as long as this verb also selects for a DP.

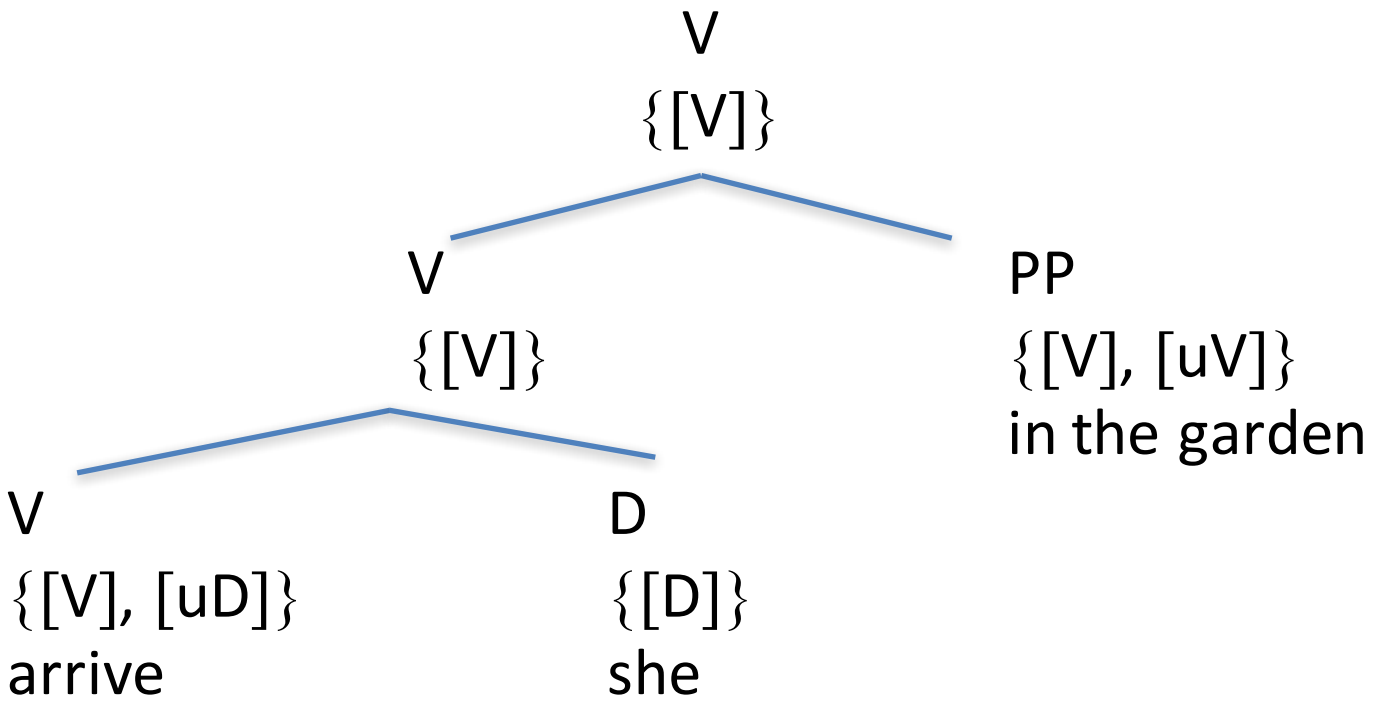
- A PP sister of a verb behaves argumental if it is merged before the verb selects other (DP-)arguments:



XI. PP arguments

PP adjuncts and PP arguments of a verb can be syntactically distinguished, as long as this verb also selects for a DP.

- A PP sister of a verb behaves adjunct-like if it is merged after the verb selects other (DP-)arguments:



XII. CP arguments

So, PPs do not form an argument against c-selection of Vs, while the PP-adjunct/argument distinction still remains available.

- Hence, the only argument against c-selection is the difference between DP arguments and CP arguments.
- In order to assess these differences, first it must be established what the syntactic features of CP arguments are.
- Most crucially, it should be determined what the differences and correspondences between (argument) CPs and (argument) DPs are.

XII. CP arguments

Closer scrutiny shows that (argument) CPs share a number of prototypical properties of (argument) DPs.

- They control (3rd singular) agreement: That it rains is clear.
- They can be referred to by pronouns: That (John is ill) I know.
- They have case (in a clause with a CP subject, the DP receives dependent case): That Bill left Susanne shocked her.

Note that this not necessarily apply to every CP. It only holds for those CPs that can be used as (verbal) arguments.

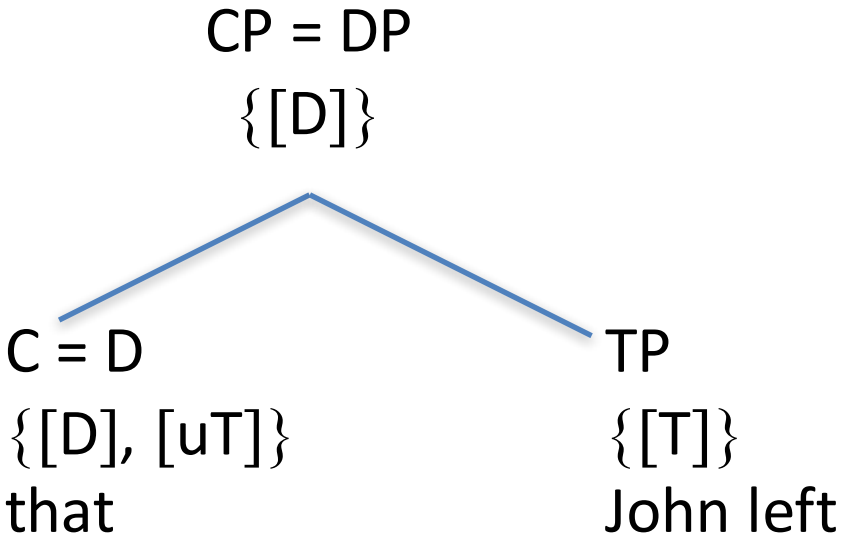
XII. CP arguments

CPs differ from DPs in the sense that complementizers select clauses (TP, vP, VP). Focusing on (argumental) *that*-CPs:

- *That* selects a TP
- The merger of *that* and a TP behaves like a DP
- On these grounds, it makes sense to think of complementizers like *that* as elements that change TPs into DPs. (Note that the pronominal nature of complementizers has been proposed earlier by Kayne 1994, among others).

XII. CP arguments

Proposal: *that* carries {[D], [uT]}.



XII. CP arguments

But this means that every verb that selects a DP or CP argument must carry [uD] (cf. Wurmbrand 2014). The lexical semantics of these verbs determines whether the argument is an individual or a proposition.

- But now the question arises as to how verbs that select for multiple (DP) arguments encode these selectional properties

- In order to address this turn question it is necessary to discuss case.

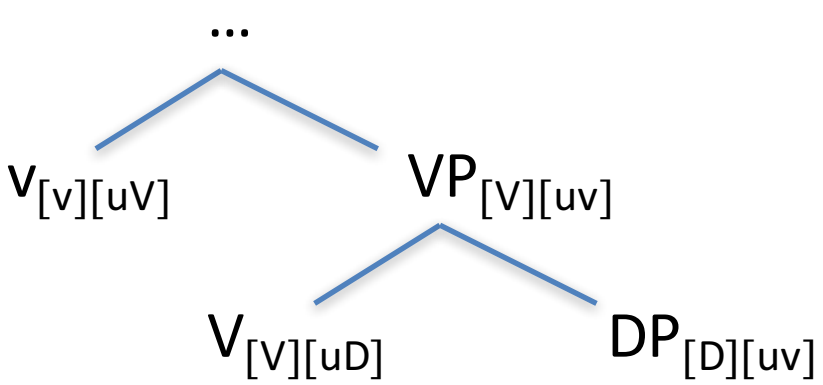
XIII. Case

Another desideratum for a labelling theory is how to deal with case. Under the Chomskyan view that case manifests an Agree relation between a DP and some (functional) head, our proposal faces severe problems.

- To see this, assume that accusative case is [uv].
- Once [uv] is the sister of v (carrying [v]) their merger would be labelless:

XIII. Case

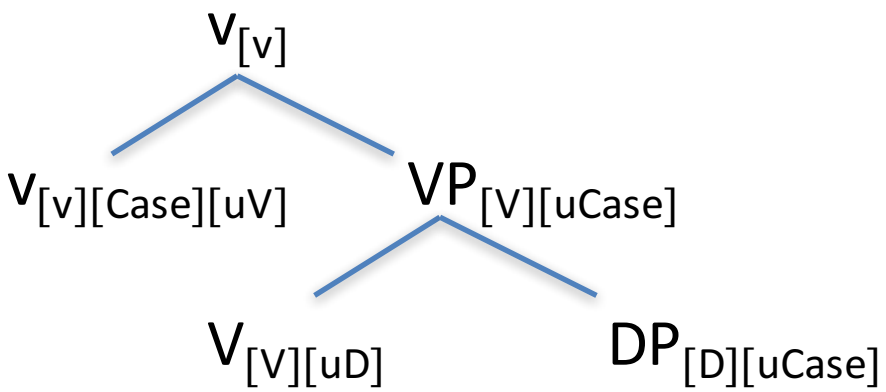
The result would yield an element that lacks any (independent) feature:



- Note that assigning v a [uD] feature would not solve the problem, as then merger with a DP would then render the merger labelless again.

XIII. Case

Of course, this problem can be circumvented by ad hoc solutions; e.g., accusative reflects some feature [uCase] that at PF is realized as dependent accusative case:



- But apart from lacking independent motivation, this solution is problematic for a number of other reasons.

XIII. Case

- No principled motivation for [(u)Case] features
- Not clear whether morphological case reflects Agree (cf. Marantz 1990, Baker 2015)
- No understanding why DPs should exhibit abstract Case in the first place.

XIII. Case

Major problem for any case theory:

- Morphological case seems to be PF-determined (cf. Marantz 1990, Bobaljik 2008, Legate 2008).
- But DPs require licensing by some functional head (generally captured by syntactic Agree): abstract Case.
- Is there any way how the DP requirement to be licensed by a functional head can follow without encoding this in a direct feature-checking relation?

XIII. Case

Answer: yes, there is. Let's solve this problem with the previous question: what happens with verbs selecting for multiple DP arguments.

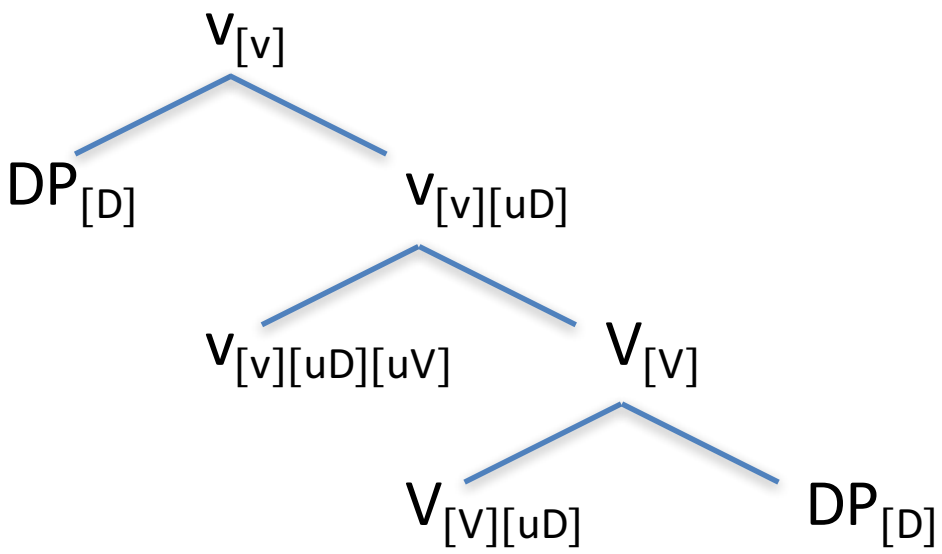
- Suppose that a verb selects two DP arguments:

- Such a verb cannot carry two [uD]-features:

$$\{\{[V], [uD], [uD]\} = \{[V], [uD]\}$$

XIII. Case

In order to be able to select for more than one DP argument, a verb needs to merge with another DP-selecting element (P or v):



Note that v is almost featurally identical to P.

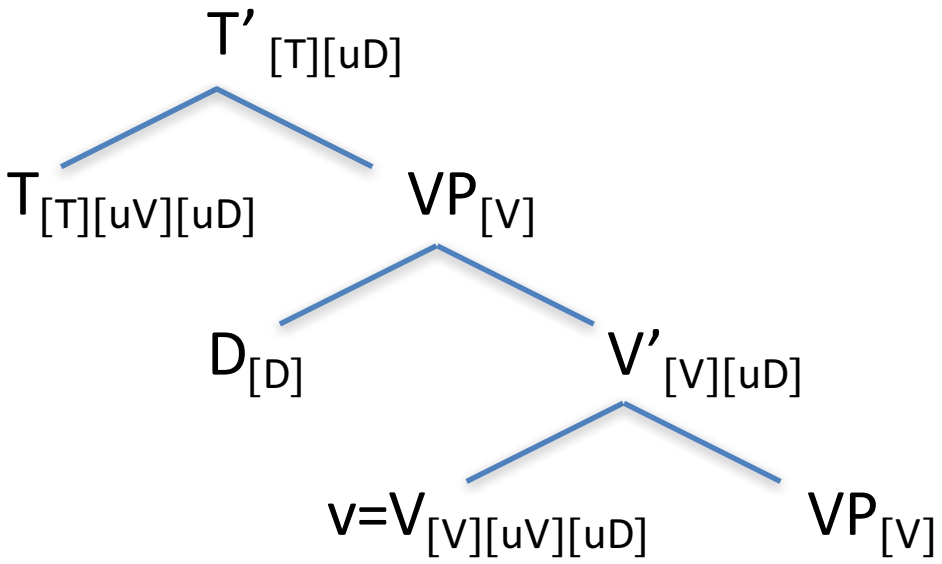
XIII. Case

In fact, we can think of v indeed as a kind of (verbal) preposition:

- V = {[V], [uV], [uD]}
- Note that this would first strengthen the resemblance between what look like two different 'assigners of accusative case', v and P.
- Second, this would unify vP/VP-selection by T:

XIII. Case

TP selecting vP=VP



For, now, we will just stick to the notion of v for illustrative purposes.

XIII. Case

Abstract Case follows from the requirement that DPs need to be selected.

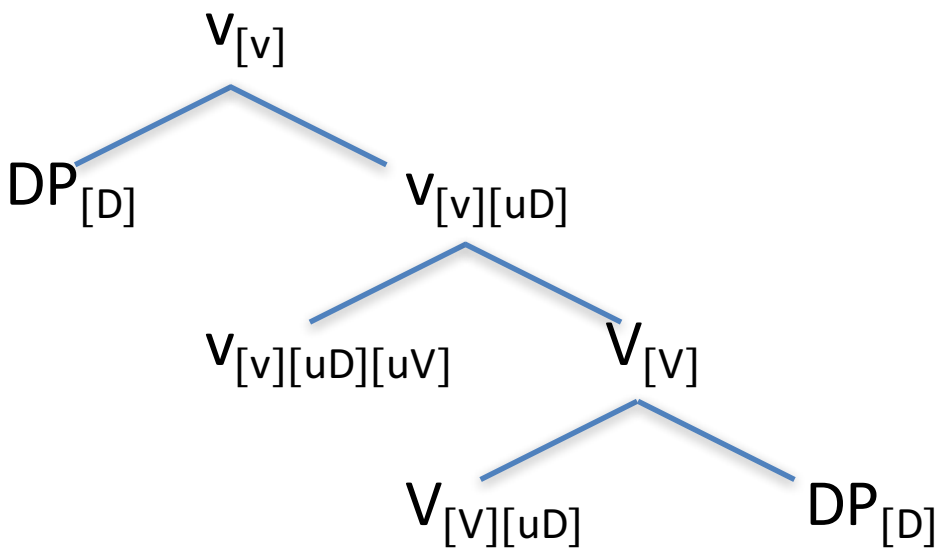
- Even if semantically a verb allows for multiple arguments, syntactically it cannot select multiple DPs, given that the set of selectional features is unordered.

- However, such an intuition of what is behind abstract Case is only valid if non-DP selecting elements cannot be merged with DP.

XIII. Case

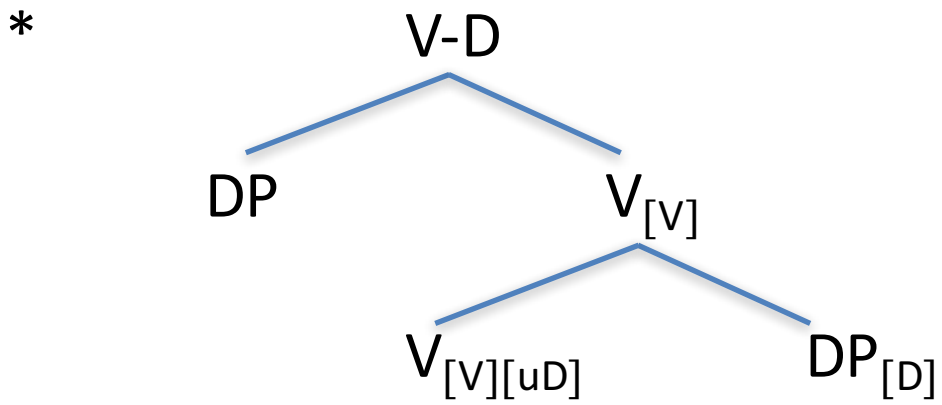
Indeed, every licenser of abstract Case is a head that selects for (phi-valued) DPs: Finite T, v, P.

- The function of introducing a head that can 'assign' abstract Case is nothing but selecting a new DP.



XIII. Case

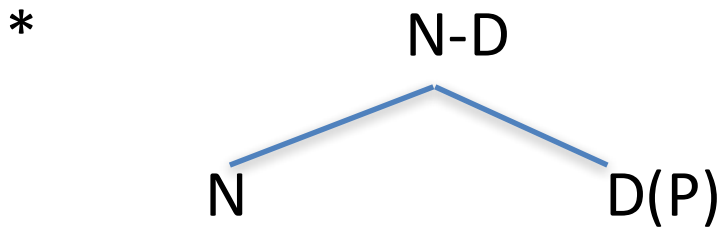
It needs to be prevented that DPs (and other elements) do not merge with elements that do not select them:



- There are various ways of implementing this. The most straightforward one would be to allude to a more general constraint on Merge that it may only take place if it leads to feature checking (cf. Pesetsky & Torrego 2006, Wurmbrand 2014).

XIII. Case

Such a constraint also prevents that DPs (and other elements) do not merge with elements that do not select them, such as nouns:



- This also derives that nouns selecting a DP argument require an intermediate D-selector (a.k.a. a case-assigner) between N and D.

XIII. Case

Under this proposal, abstract Case is nothing but the requirement that every DP is to be selected, i.e. to be merged with, an element carrying [uD]. This correctly predicts that

- Every known 'case-assigner' (P, finite T, v) is an element selecting for DPs.
- DPs must Merge with these selectors.
- If the difference between elements carrying structural case and inherent case is that only elements with structural case have syntactically active ϕ-values, every agreement probe agrees with the closest DP 'carrying' structural case (cf. Bobaljik 2008, Bjorkman & Zeijlstra 2015).

XIV. DP-selection

Now we can also address the previous question: which verbs select for DPs (i.e., carry a [uD] feature)?

- All arguments need to be base-generated inside the vP/VP (VISH)
- Every DP needs to be selected
- Every verb requires one DP subject
- A verb cannot select for more than one DP

- Conclusion: each verb must carry exactly one feature [uD].

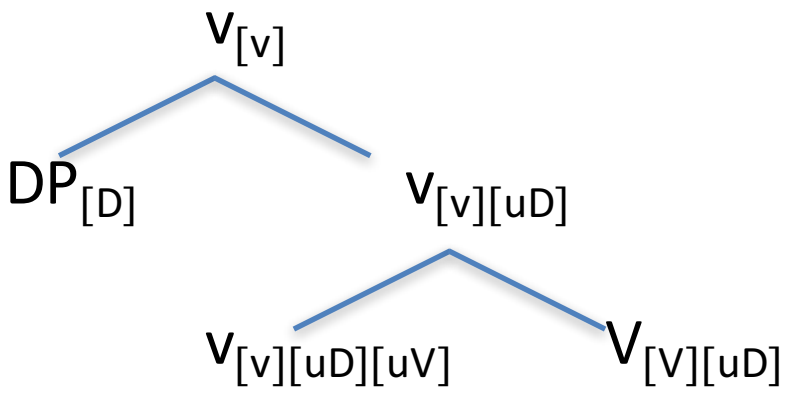
XIV. DP-selection

How to distinguish verb types:

- Every transitive verb carries [uD] (which selects the object DP); a second verbal head (‘v’) selects the subject.
- Transitive verbs ‘selecting’ PP arguments, are actually intransitive verbs (carrying [uD]) being selected by a PP-argument.
- Unaccusative intransitive verbs carry a feature [uD] (which selects the subject DP).
- Unergative intransitive verbs carry a feature [uD], but merge first with v:

XIV. DP-selection

Unergatives are intransitive verbs that select a DP but merge with v first:



- Note that this entails that the fact that unergatives lack objects is purely semantic; syntactically they could select an object (cf. cognate objects):

  I walked a walk
  I dreamed a dream

XIV. DP-selection

Summing up:

- Every verb (perhaps even by definition) selects for a DP (i.e., it carries [uD]).
- Selection of multiple DPs arguments requires introducing additional functional heads (v, P, Appl), an effect known as abstract Case.
- Different syntactic verb types exist by virtue of semantic differences between types of verbs, and the fact that these differences can be reflected by different syntactic configurations.

Selection and Agree

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Göttingen
Campus

XV. Agree

So far, the system proposed exploits uninterpretable (or better: dependent) features to encode selectional requirements. But if (local) selection results from feature checking, how can Agree apply on a long distance?

- Proposal: Long-distance Agree is the result of local checking plus asymmetric feature percolation.

XVI. Upward and Downward Agree

Currently there are two versions of Agree.

- **Downward Agree (Chomksy 2001, Preminger 2013, Preminger & Polinsky 2015):** an *uninterpretable* feature must c-command an *interpretable* matching feature.

- **Upward Agree (Wurmbrand 2011, 2012, Zeijlstra 2012, Bjorkman & Zeijlstra 2015):** an *interpretable* feature must c-command an *uninterpretable* matching feature.

XVI. Upward and Downward Agree

Major arguments come from different perspectives:

- **Downward Agree:** certain cases of long-distance agreement involve clear cases where the goal never c-commands the goal.

- **Upward Agree:** all syntactic dependencies (binding, strict NPI-licensing, Negative Concord, Sequence of Tense) except morphological co-varying phi-agreement involve clear 'upward' checking (adhering to (Neeleman & Van der Koot's 2002 configurational matrix); long-distance agreement is often dependent on other instances of Upward Agree and defective.

XVI. Upward and Downward Agree

Solution: separate valuation from checking (Bjorkman & Zeijlstra 2015)

- **Checking:** every uninterpretable feature must be checked by a c-commanding matching interpretable feature.

- **Valuation:** every unvalued feature must be valued; only if the checker does not value this feature itself, a lower interpretable feature can do this to (provided that this lower goal stands in a checking relation with the goal itself).

XVI. Upward and Downward Agree

Examples: In languages with nominative-accusative case systems, agreement is with the subject.

- $[[DP_{NOM}]\ T_{[u\phi:\ \_\_]} \ldots [DP_{ACC}]]$ (English)

In languages with ergative-absolutive case systems, agreement is either with the subject or the absolutive (cf. Bobaljik 2008). If ergative case is structural, the phi-features on the subject are active and can value the probe (e.g. Nepalese); if ergative case is inherent, these phi-features are inactive and the absolutive object ((indirectly) receiving case from T) fulfils the valuation requirement (e.g. Hindi).

- $[[DP_{ERG}]\ T_{[u\phi:\ \_\_]} \ldots [DP_{ABS}]]$

XVI. Upward and Downward Agree

So far:

- Under this approach instances of downward agreement are compatible with the Upward Agree approach. For our purposes here, it suffices to say that every [uF] needs to be checked by a matching [F], the latter c-commanding the former.

- If valuation cannot take place under checking, another goal can complete the valuation process (under particular configurational conditions).

XVII. Checking under sisterhood

Now Agree (i.e. feature checking) is fully compatible with selection. Recall the Principle of Containment of Syntactic Information: *syntactic information may not disappear in the tree (i.e. all syntactic features percolate).*

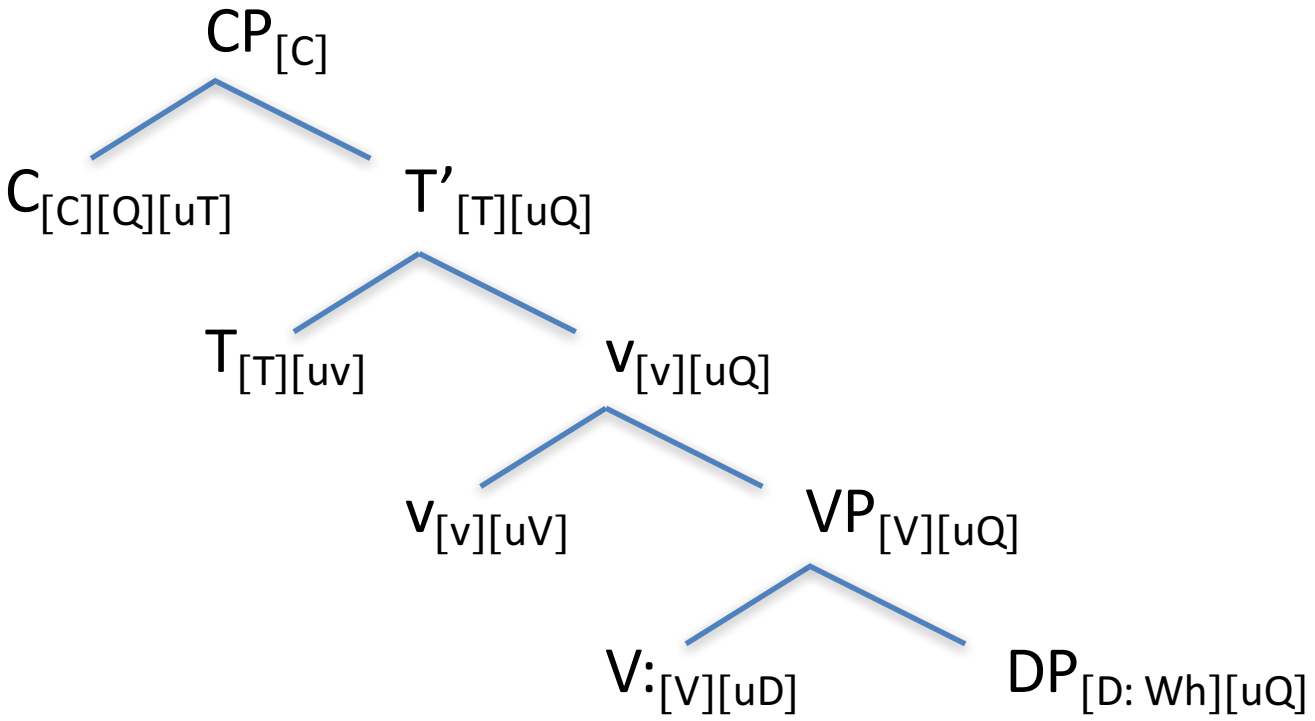
- By definition, interpretable features can never percolate beyond their maximal projection.

- Uninterpretable features always percolate up, until they stand in a sisterhood relation with a matching interpretable feature.

This derives Upward Agree.

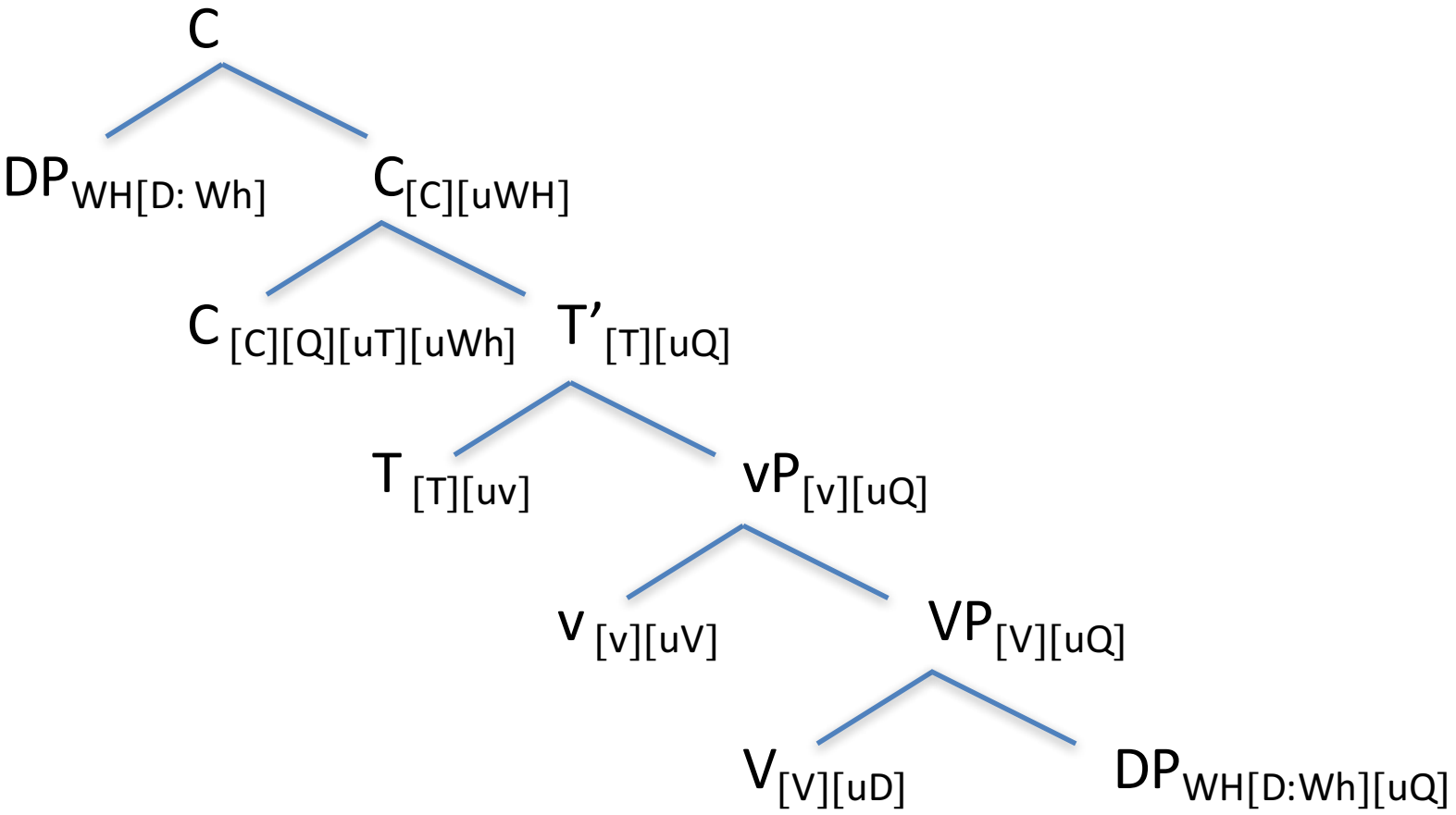
XVII. Checking under sisterhood

- Assume that all Wh-elements carry a feature [D: Wh] and a feature [uQ]; and that interrogative C carries a feature [uWh] and [Q]. Then, the Wh-elements can have its [uQ] feature checked off in situ by percolating it up till the C-level:



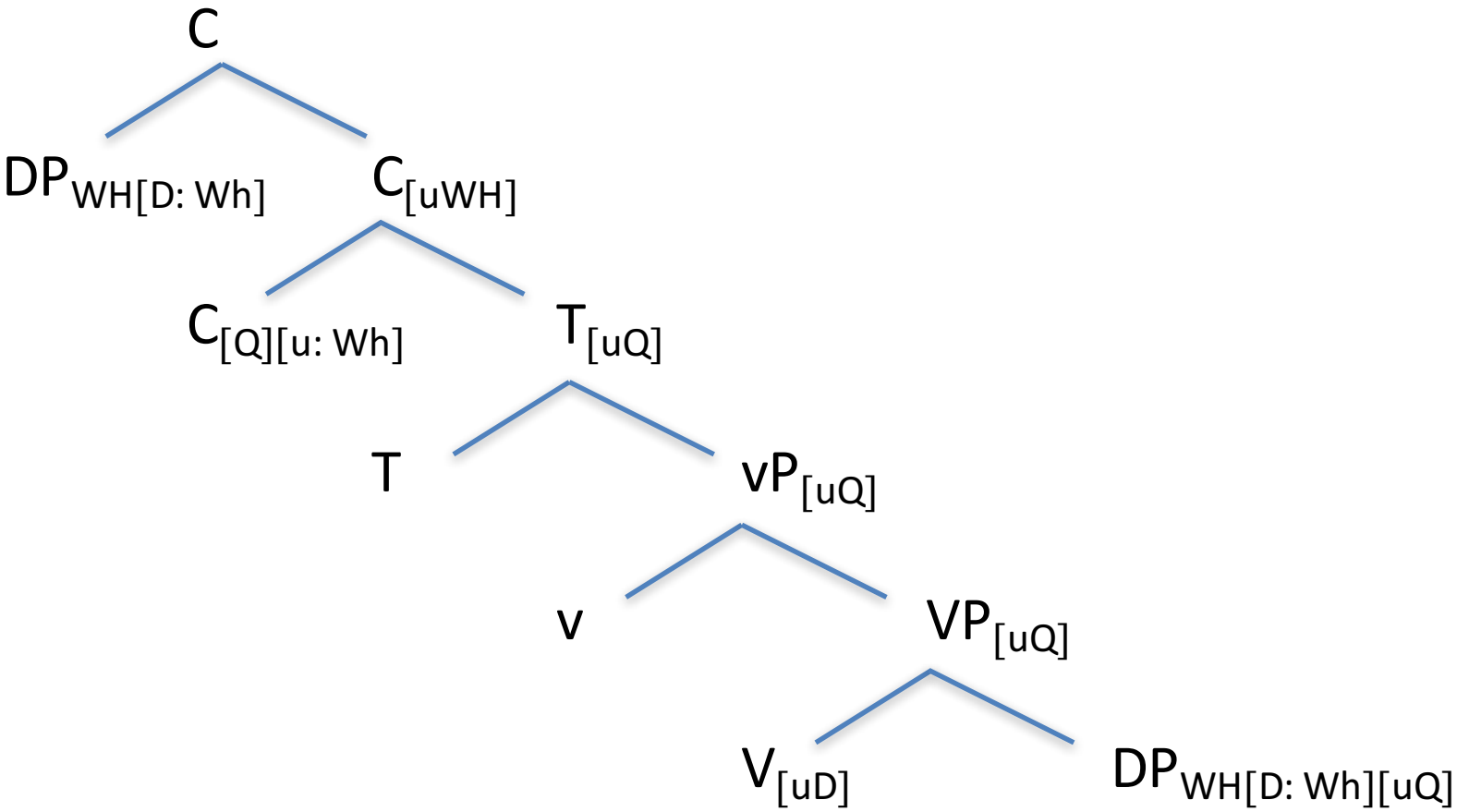
XVII. Checking under sisterhood

- Movement is subsequently triggered by the need to check the higher [u: Wh] feature on C:



XVII. Checking under sisterhood

- Movement is subsequently triggered by the need to check the higher [u: Wh] feature on C:



XVII. Checking under sisterhood

Summing up:

- The proposal is fully compatible with uninterpretable features triggering Selection, (Upward) Agree and Movement.

- Upward Agree follows directly from the proposal (it was, in fact, developed as a motivation to derive the asymmetric nature of Upward Agree).

- Note: *Agree* is not the same as *agreement*; not every checker (fully) values its checkee; in those cases alternative, lower valuers may complete the valuation process.

Consequences and (open) questions

XVIII. So far ...

The proposal for a labelling algorithm:

- Motivates why the selector projects
- Can account for labelling in all configurations (including adjunction)
- Can overcome the ordering problem
- Can circumvent the problems relating s-selection vs c-selection and mutual selection (e.g. with case)
- Can account for the differences between selection and (long-distance) Agree

XVIII. So far ...

Moreover, it provides:

- A principled understanding of a syntax-internal labelling mechanism.
- A motivation why Agree (in the sense of feature checking) applies in an upward fashion.
- A motivation why DPs need to be licensed by functional heads (a.k.a. abstract Case).

XVIII. So far ...

At the same time, many questions are still open. For now, I will focus on three pressing questions (but there are many more, of course):

- PPs modifying NPs and APs
- DP-internal structures
- Overgeneration and valuation

PPs modifying NPs and APs

XIX. Predicates

An open question is how to deal with PPs modifying NPs and APs.

- In principle PPs may modify NPs.

  The book about Obama

- Also, (predicative) adjectives may be modified by PPs (albeit more rarely, presumably a consequence of the fact that most adjectives assign only one theta-role):

  The doctor is afraid of the patient
  *The afraid of the patient doctor committed suicide

XIX. Predicates

If PPs (as any other adjuncts) select for their sisters, and PPs do not only select for verbs, there are two logical options:

- PPs are ambiguous between verbal, nominal and (predicatively used) adjectival PPs (and there are, thus, three types of prepositions).

- There is a supercategory above verbs, nouns and (predicatively used) adjectives.

The first option can hardly be supported as virtually every PP can be used to modify VPs, NPs and A_{Pred}Ps.

XIX. Predicates

The second option, however, has been proposed for a variety of other reasons in the literature:

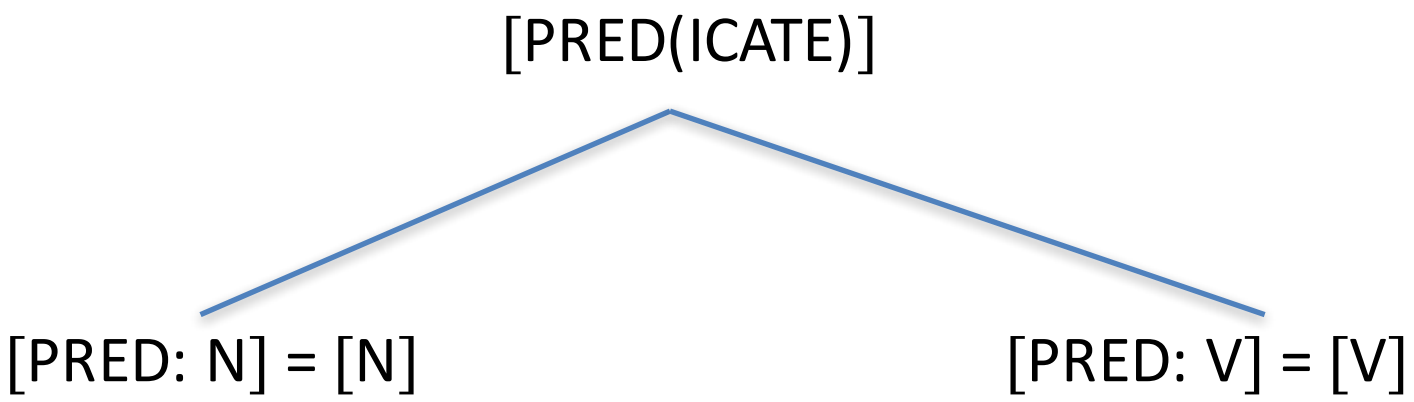
- In several languages various lexical items (if not all) can be used both verbally and nominally.
- The idea that lexical items are not stored in the lexical as nouns/verbs, but as roots, which are rendered nominal/verbal by having them merge with a N-/V- feature is very much in line with elements having some kind of a supercategorial feature.
- Semantically, verbs, nouns and adjectives all seem to denote predicates, with additional argumental or other requirements; this semantic core would then be reflected in the syntactic featural inventory.

XIX. Predicates

Implementation:

- There is a superfeature [Predicate]
- This feature can receive a feature value (i.e. become a subfeature) V or N.
- Predicatively used adjectives may be unvalued predicates.

XIX. Predicates

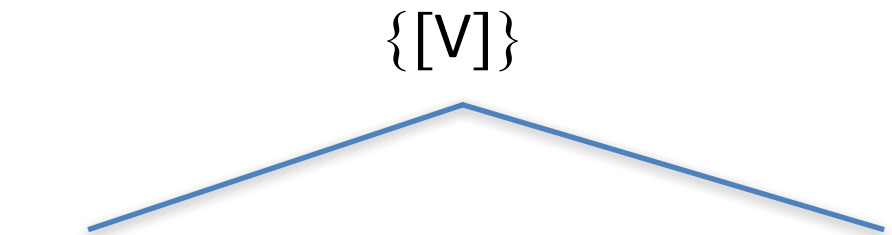


Now, if this is correct, Ps are nothing but elements that are:

- {[PRED], [uPRED], [uD]}
- Immediately valued for V/N (and thus become {[V], [uV], [uD]} and {[N], [uN], [uD]} respectively) when merged with a V- or N-labelled element.

XIX. Predicates

V PP merger:

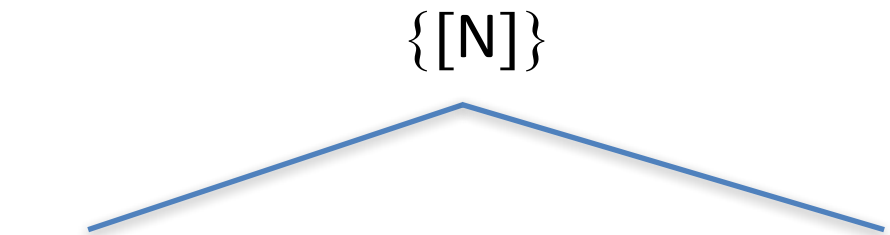


{[Pred: V]} {[Pred: V], [uPred: V]}

{[Pred: V]} {[Pred:], [uPred:]}

XIX. Predicates

N PP merger:



{[Pred: N]} {[Pred: N], [uPred: N]}

{[Pred: N]} {[Pred], [uPred]}

XIX. Predicates

If preprositions are indeed elements carrying {[PRED], [uPRED], [uD]}:

- Their behaviour of PPs as N, V and A modifiers follows naturally.
- The analysis of adjuncts as elements selecting their modifiee can be maintained.

DP-Internal Syntax

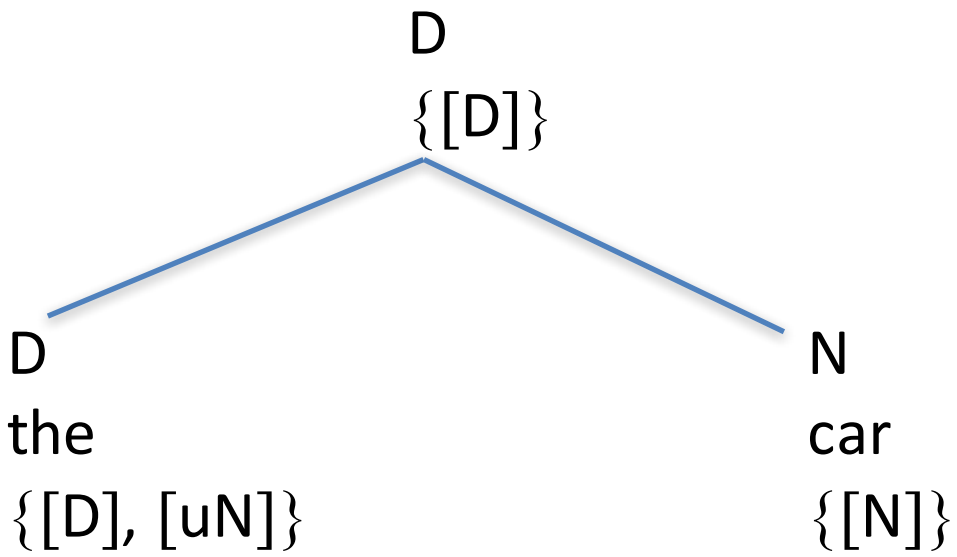
GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Göttingen
Campus

XX. D-NP selection

If predicative adjectives are indeed elements carrying {[PRED]}, questions concerning the internal structure of DPs arise .

- Determiners select nouns, or nominal elements:



XX. D-NP selection

Under this system, proper names and pronouns are D-heads that do not select for any NPs.

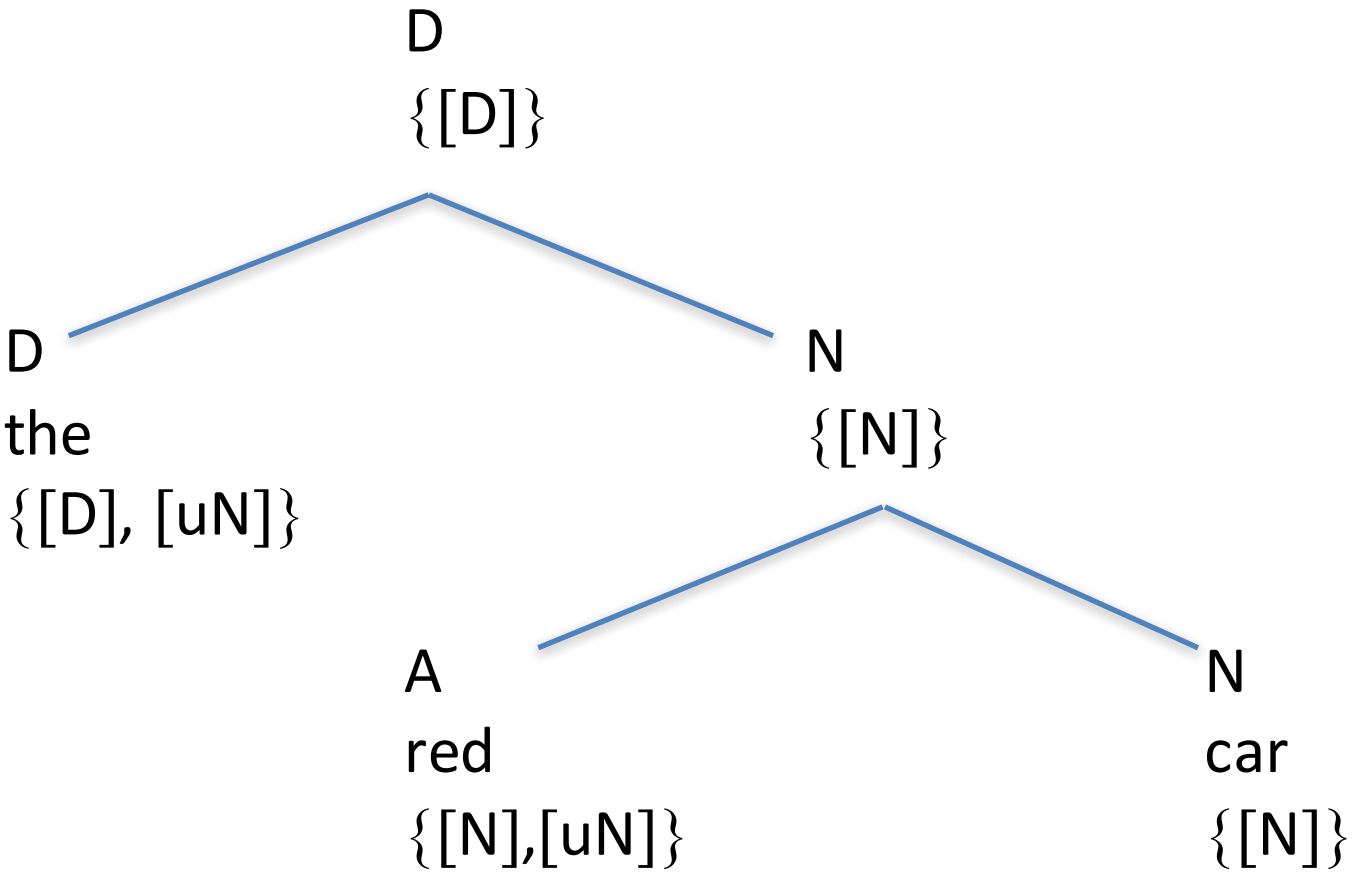
- *John, Mary, the US*: {[D]}
- *They, her, we*: {[D]}

Consequence:

- No need for a complex internal covert NP complex inside the DP for pronouns / proper names (pace Abney 1997 *et seq*).

XXI. Two types of adjectives

But if Ds select NPs, nouns modified by (attributively used) adjectives must be NPs too.



XXI. Two types of adjectives

But now we face a problem. If predicatively used adjectives are {[Pred]}, how can attributively used adjectives be {[N],[uN]}?

- In some languages (e.g. Dutch) only attributively used adjectives receive morphological inflection, predicatively used adjectives do not.

  De auto is rood/*rode
  The car is red

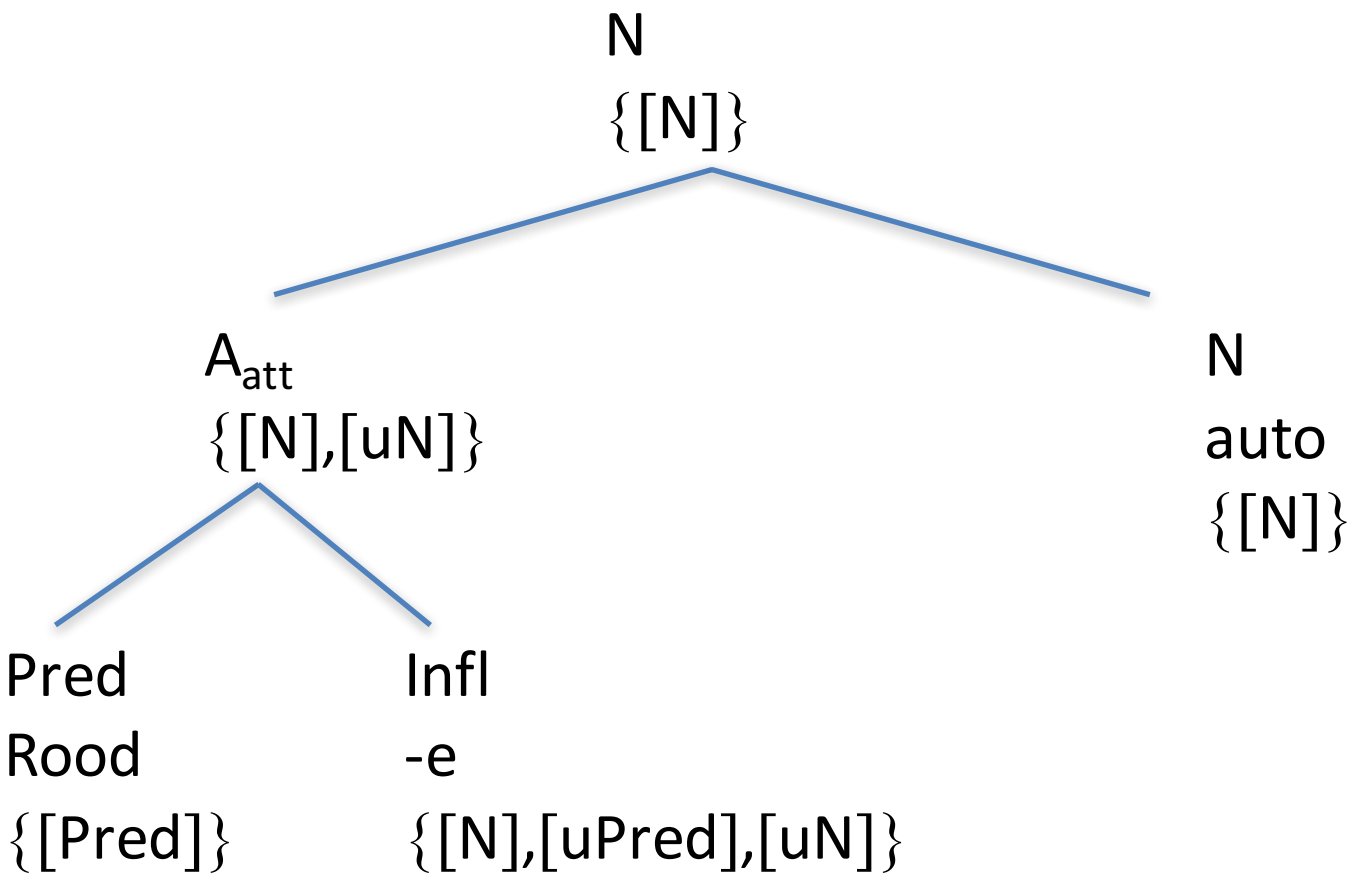
  De rode /*rood auto
  The red.INFL car

XXI. Two types of adjectives

Arguably, this shows that inflectional morphology on attributively used adjectives has a different formal function than that on predicatively used adjectives.

- This allows us to hypothesize that inflectional morphology on adjectives functions as a type-shifter: it changes predicates into nominal adjuncts.

XXI. Two types of adjectives



XXI. Two types of adjectives

Evidence for the idea that inflectional morphology changes predicates into nominal adjuncts comes again from PP modification.

- As in English, Dutch predicatively used adjectives can be modified by (right-adjoined) PPs; attributively used adjectives cannot be modified by (right-adjoined) PPs

De student is bang van Marie
The student is afraid of Marie
*De bang van Marie(-e) student
The afraid of Mary student

XXI. Two types of adjectives

Evidence for the idea that inflectional morphology changes predicates into nominal adjuncts comes again from PP modification.

- But attributively used adjectives can be modified by left-adjoined PPs:

  De van Marie bang-e student
  The of Mary afraid.INFL student

- [De [[[van Marie bang]-e] student]]

XXI. Two types of adjectives

Under this proposal, at least for some instances of inflectional morphology their presence can be motivated in formal terms (instead as some piece of redundant uninterpretable information).

- Moreover, note that this way, adverbs are the verbal counterparts of attributively used adjectives; they are elements carrying {[V],[uV]} that can be created in the same way from predicates.

- adverbial *–ly* would then just be {[V],[uPred],[uV]}: merged with a predicate like *quick*, this would then yield: *quick-ly*, carrying {[V],[uV]}.

XXI. Two types of adjectives

Finally, one question needs to be addressed. If prepositions carry {[Pred],[uPred],[uD]}, motivated by their ability to modify nouns and verbs in the same way, why don't we have *particle noun constructions*?

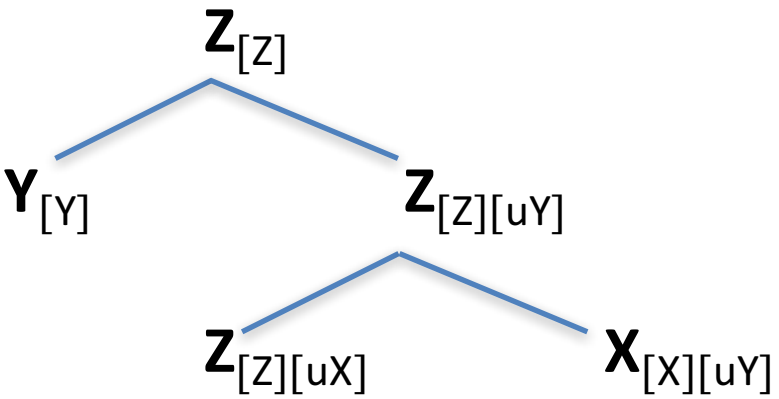
*The in car seat (*int.: the car-internal seat*)

- The proposed system ruled this out. Apart from theta-theoretic requirements, *in car seat* would end up carrying {[N],[uD]}. But determiners carry {[D],[uN]}. Merger of these two would yield a label-less element.
- Of course merger of a proper name / pronoun and a particle noun construction should then be ruled out independently (in terms of semantic mismatches).

Overgeneration and valuation

XII. Overgeneration

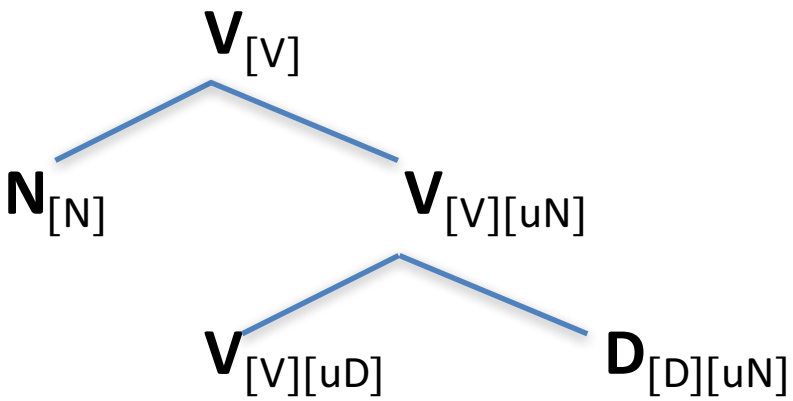
One particular problem should however be discussed: overgeneration: selection often 'can't wait', but selectional features can (and should) be able to percolate up quite unrestictedly (for Agree to be a long-distance effect):



XII. Overgeneration

This should be ruled (in many cases). D needs to select first for an NP, before it gets selected itself:

*



XIII. Valuation

Apparently, D has a special requirement for NP-selection. That needs to take place before it is selected itself.

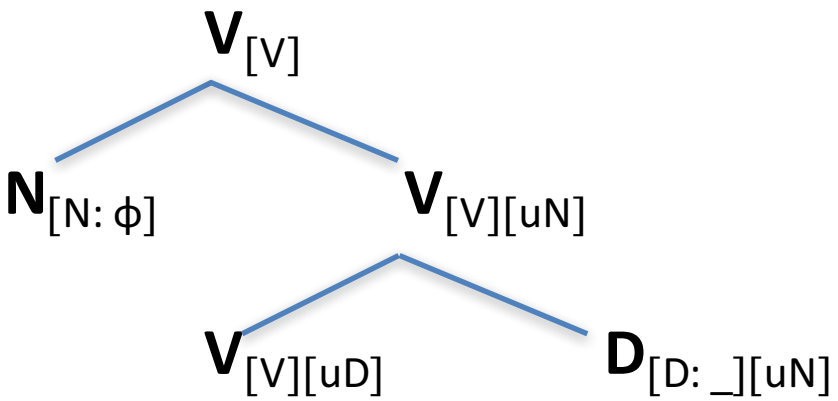
This is where valuation comes in:

- D is the carrier of ϕ-features
- D is not the source of ϕ-features (that is N)
- D must be valued by N before it gets selected again

XIII. Valuation

If D does not merge with NP first, D cannot be valued for any ϕ-features:

*



XIII. Valuation

If D does not merge with NP first, D cannot be valued for any ϕ-features.

- Hence, valuation is needed in syntax to 'prioritize' selectional requirements. It is not only motivated within the system to account for PRED-valuation.
- Valuation is not a PF mechanism; it is an operation that can also be postponed till PF.
- Valuation is a *must-if-you-can* mechanism, it takes place as soon as possible.

To what extent valuation also calls for 'prioritized selection' on other domains (outside the DP) still needs to be investigated.

Conclusions

XXIV. Conclusions

The proposal for a labelling algorithm:

- Motivates why the selector projects.
- Can account for labelling in all configurations (including adjunction).
- Can overcome the ordering problem.
- Can circumvent the problems relating s-selection vs c-selection and mutual selection (e.g. with case).
- Can account for the differences between selection and (long-distance) Agree.

XXIV. Conclusions

Moreover, it provides:

- A principled understanding of a syntax-internal labelling mechanism.
- A motivation why Agree (in the sense of feature checking) applies in an upward fashion.
- A motivation why DPs need to be licensed by functional heads (a.k.a. abstract Case).

XXIV. Conclusions

And, although many questions are still open, it does account for:

- PPs modifying NPs and APs
- DP-internal syntax
- Valuation as a constraint on potential overgeneration

XXV. Questions

?

Thank you!

Hedde Zeijlstra
University of Goettingen
hzeijls@uni-goettingen.de
http://www.heddezeijlstra.org

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Göttingen
Campus