



# Towards a predictive theory of concord across categories

Antonio Fábregas

<antonio.fabregas@ntnu.no>

# Full handout



# Main claims

- Agreement features unify participants across distinct syntactic constituents ( $\approx$  what HPSH has always said).
- Agreement minimally involves a phi feature sequence and a Match head that instructs LF to unify the relevant participants picked by phi features.
  - Those features are always present when unification is necessary; however, most languages spell them out with one single exponent in predicative position.
  - Modification position involves an extra JOIN head which intervenes between AP and phi features  $\Rightarrow$  agreement more likely to emerge (as Concord) in modifiers.

# Assumptions (standard Nanosyntax)

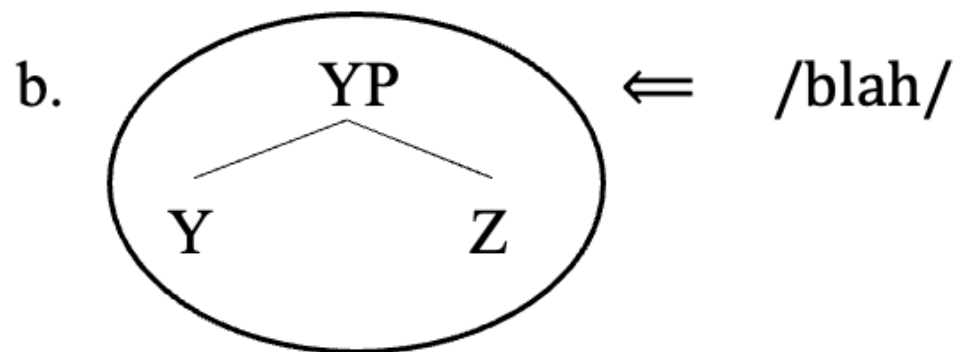
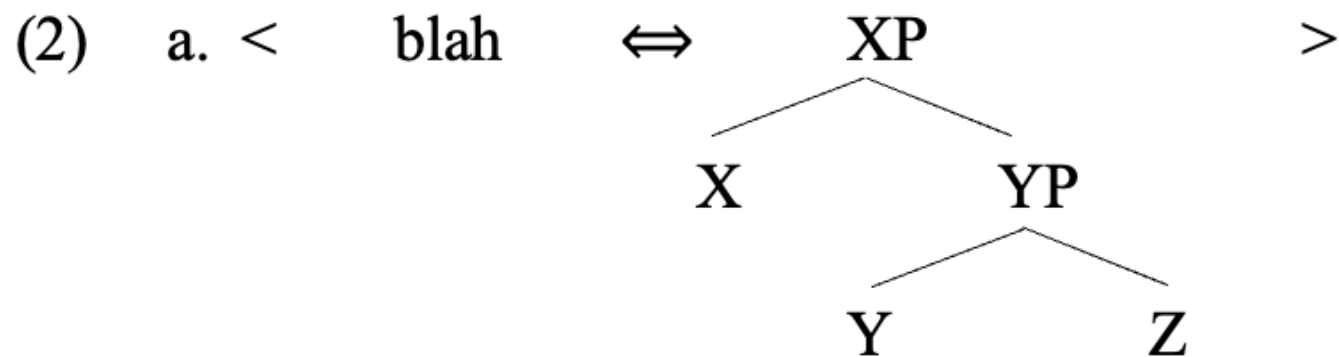
- Features are universal.
- Operations are universal.
- Variation reduces to the size and shape of lexical items.

# What I assume

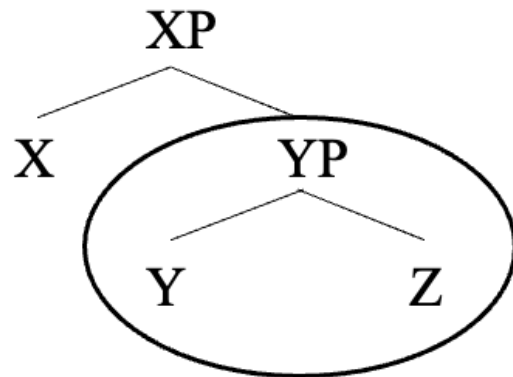
- (1) Spell out algorithm (Cortiula 2023, Caha et al. 2024)
  - a. Merge F and lexicalize.
  - b. If fail, evacuate the closest labelled non-remnant constituent and try (a)
  - c. If fail, evacuate the immediately dominating constituent and try (a) (recursive).
  - d. If merge F fails, spawn a new derivation providing F and project F in the spine.

In this work I will explore two consequences of this algorithm to throw light on the nature of concord.

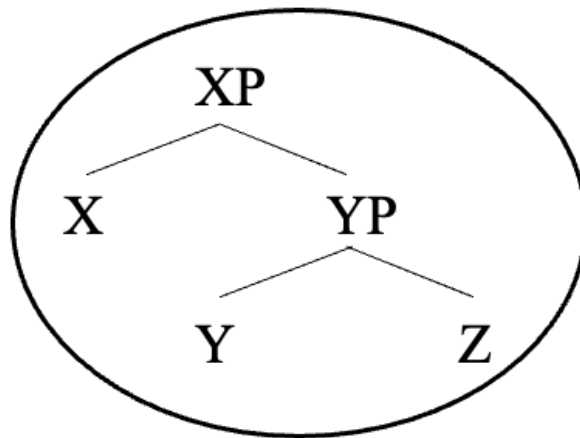
# No affix



c.



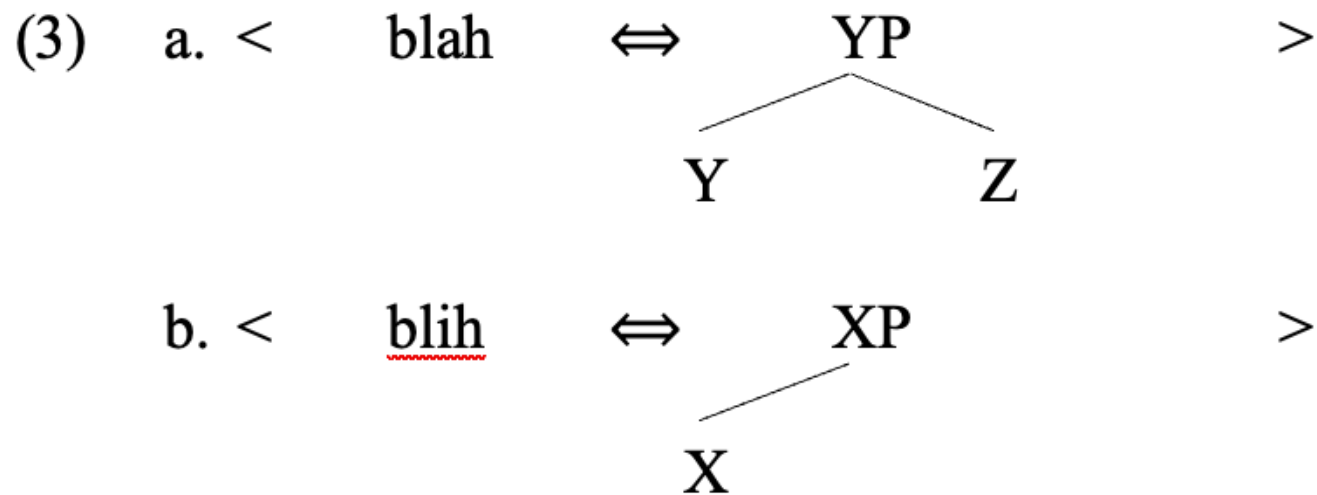
d.



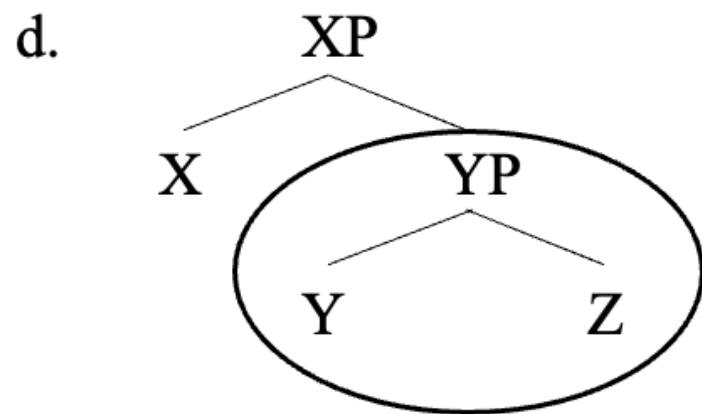
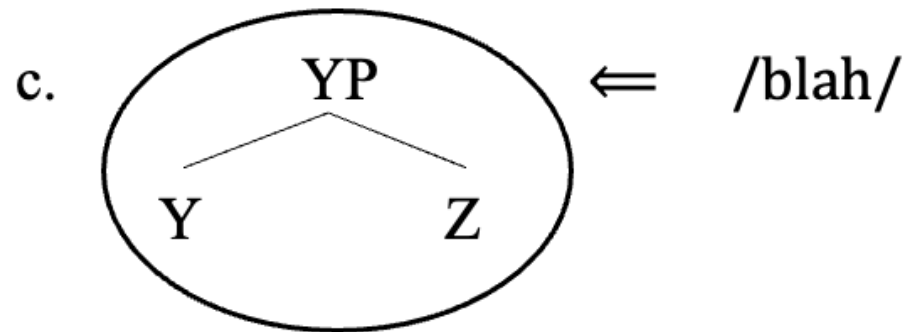
⇐ /blah/

# Suffix

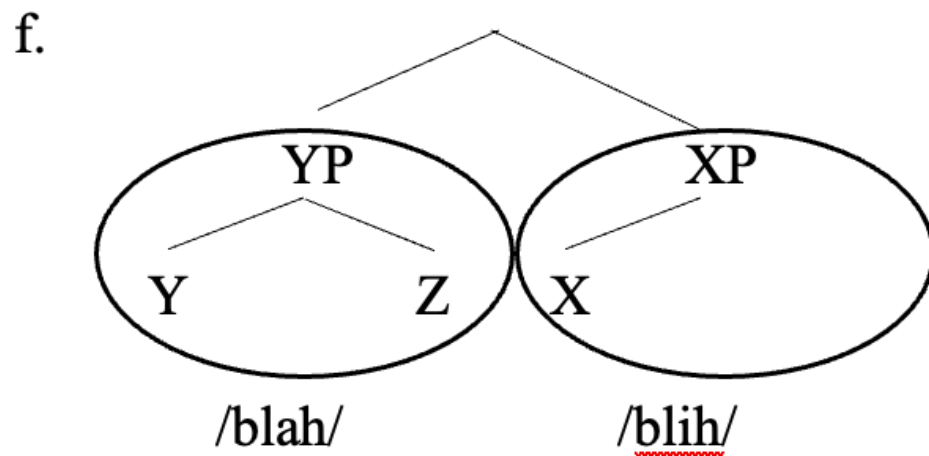
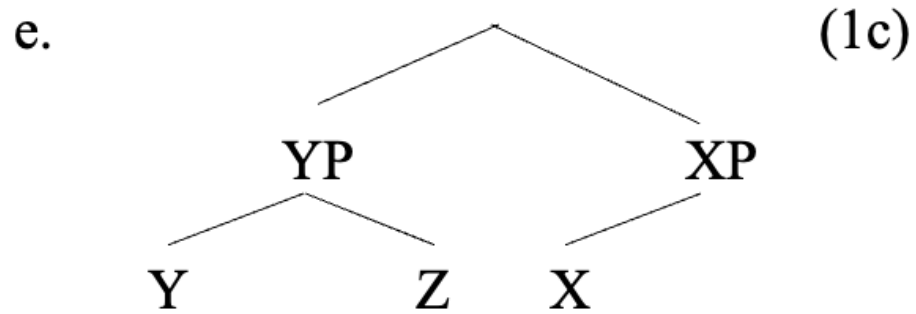
Suffix blih (1c)





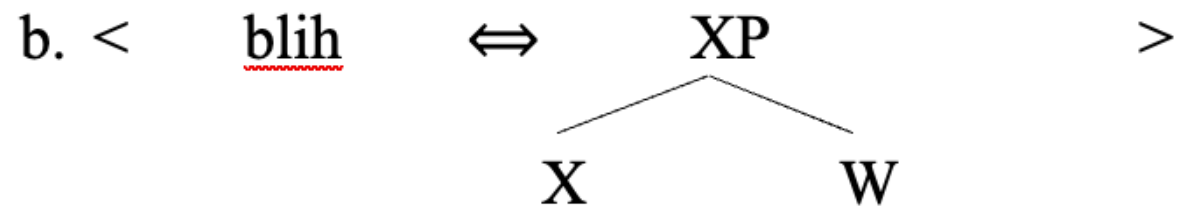
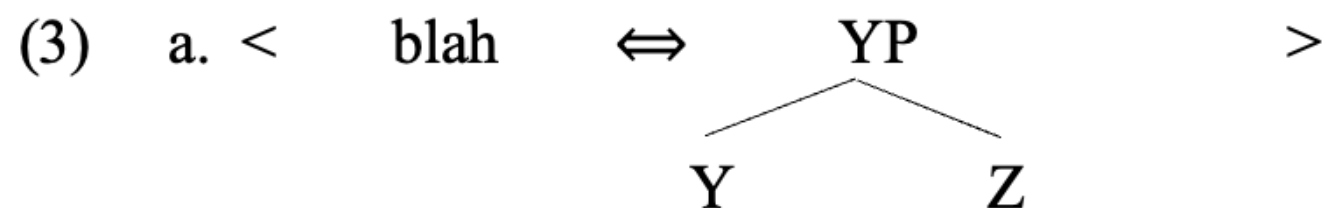


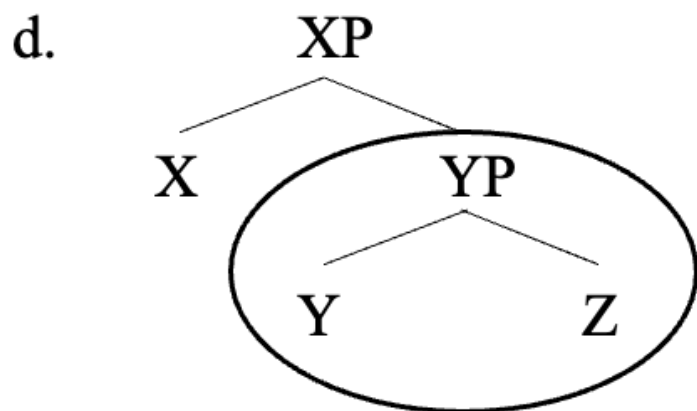
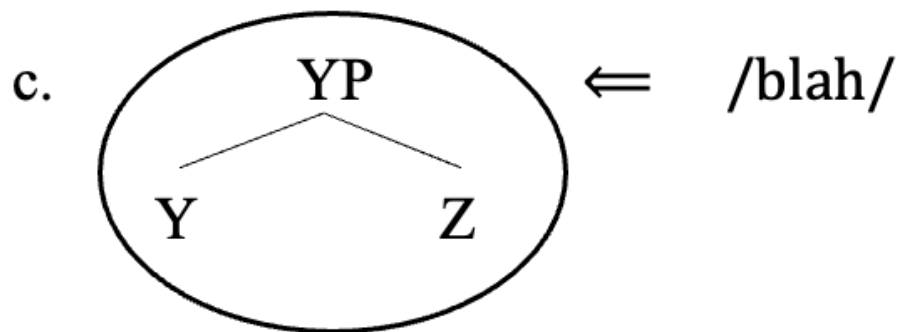
(1a) fails; (1b) fails --no specifier--



# Prefix

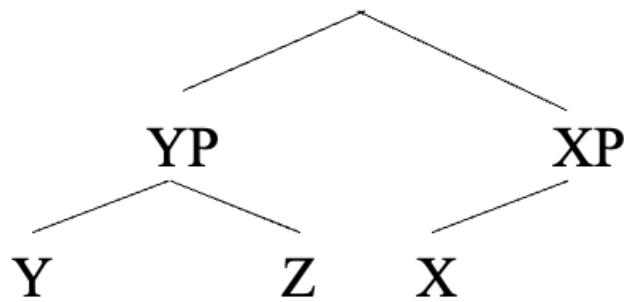
Prefix blih (1d)





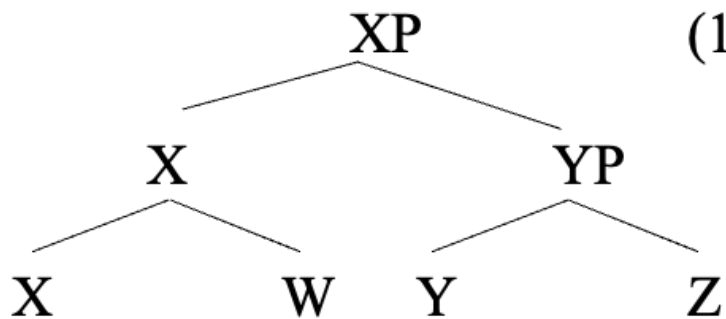
(1a) fails; (1b) fails --no specifier--

e.



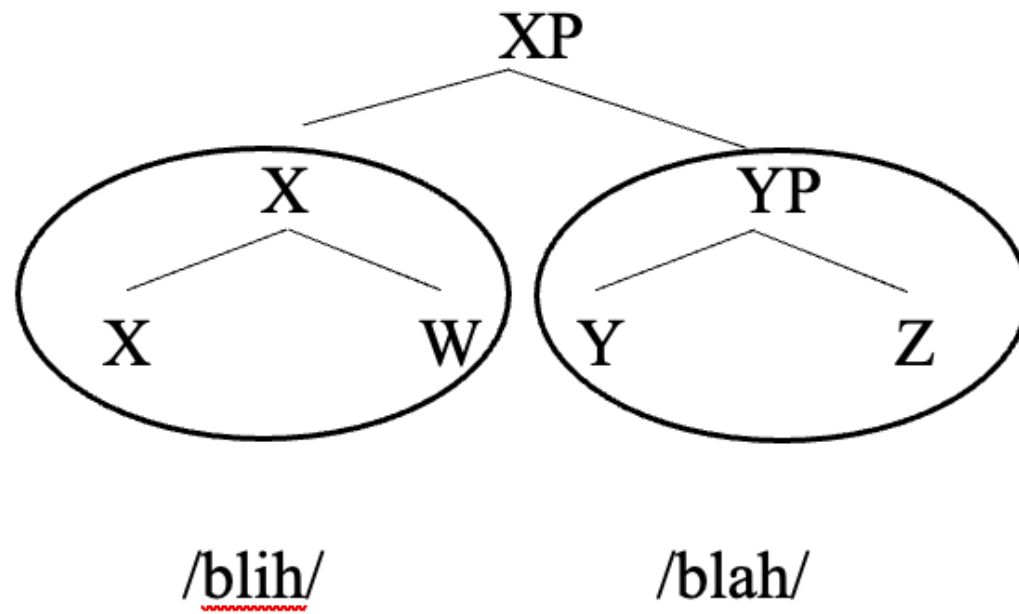
(1c); (1c) fails [all Xs have a complex foot]

f.



(1d)

g.



## Consequence:

- Prefixal material is always syntactically complex [anchors must be binary]
- Suffixal material can be syntactically complex or not [unary anchors compatible with one or several heads]

# Concord = P

- (4) a. un hombre con bigote  
a man with moustache  
b. un hombre bigot-ud-o  
a man moustache-ed-m.sg  
'a man with a moustache'

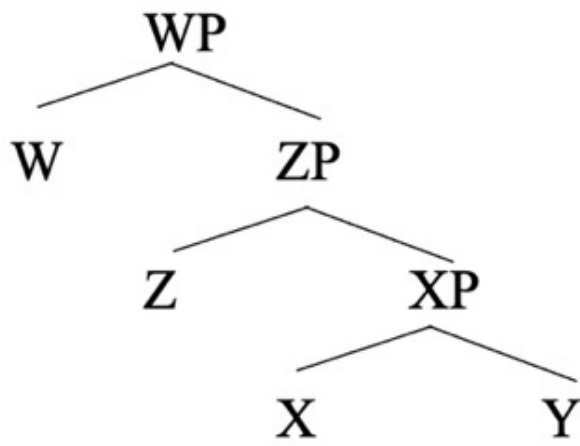
– Adjectival agreement and P cannot co-occur. One of the two must emerge, but never both.

- (5) a. un-a-s mujer-es con(\*-a-s) bigote  
a-f-pl woman-pl with- f-pl moustache  
b. un-a-s mujer-es bigot-ud-\*(a-s)  
a-f-pl woman-pl moustache-ed-f-pl

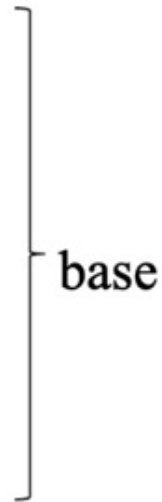


- The Spell Out Algorithm predicts that the same exponent is kept in the derivation as much as possible.
- Hence there will always be a strict complementary distribution relation between the material spelled out by any two items within the same syntactic space.

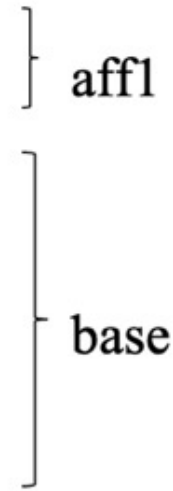
(6) Syntactic tree



Pure  
suppletion



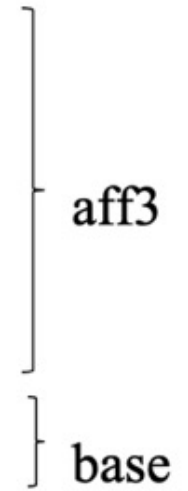
Base-affix1



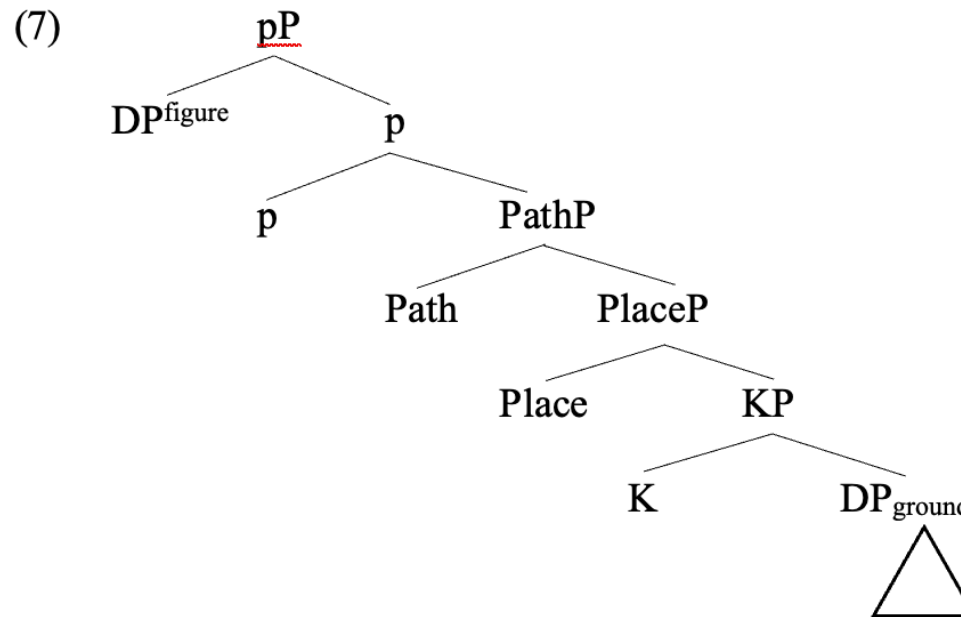
Base-affix 2



Base-affix 3



- The highest element in an adjective is a concord exponent.
- The highest element in a preposition is a relational exponent.



(8) Concord = little p

– In other words, concord is related to a relational head.

Which relation? I will argue for the following idea.

– Identity.

(agreement is used to unify participants)

# Suffixal ezafe in Balochi and Gilaki

– Ezafe is roughly characterised as some 'glue' material introduced between a nominal modifier and the head.

– There are grammatically different animals living under this label.

Balochi / Galaki: a clearly attributive marker.

Balochi / Galaki: a clearly attributive marker.

(9) mazan-en asp  
bigg-ez horse  
'a big horse'

[Balochi, Jahani & Korn 2009: 655]

(10) zərd-ə čərm-ə kif  
yellow-ez leather-ez bag  
'a yellow leather bag'

[Gilaki, Kahnemuyipour, Shabani & Taghipour 2024: 6]

– Kahnemuyipour et al. (2024): this type of ezafe is the materialisation of the semantic function JOIN (Truswell 2004, Belk 2017).

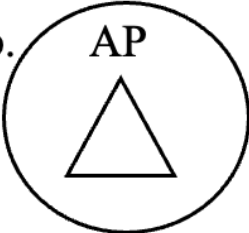
- Bierwisch (1986), Partee (1986), Zamparelli (2000): adjectives are basically predicates (type  $\langle e,t \rangle$ )
- In noun-internal position they must have a  $\langle \langle e,t \rangle, \langle e,t \rangle \rangle$  type.
- JOIN is a type shifter that moves from predicate to modifier.
- [Predicate Modification à la Heim & Kratzer 1998 or Theta-Identification à la Higginbotham 1985 is not good enough because it equals coordination]

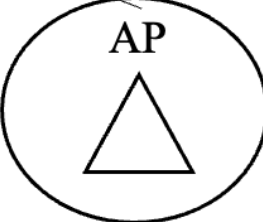
- Ezafe contains a JOIN head.



(11) Balochi / Gilaki ezafe [to be revised]

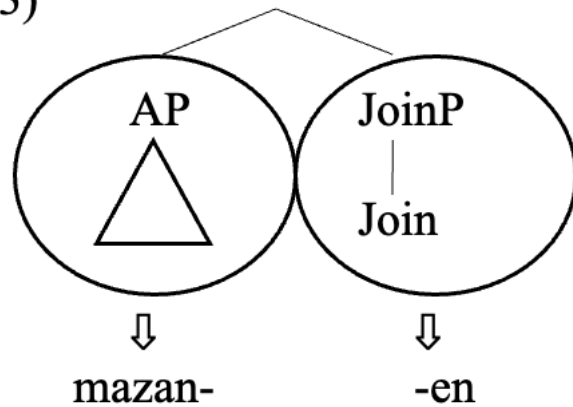
a. < JoinP      ⇔    -en / -ə >  
    /      \  
    Join

b.  ⇒    mazan-

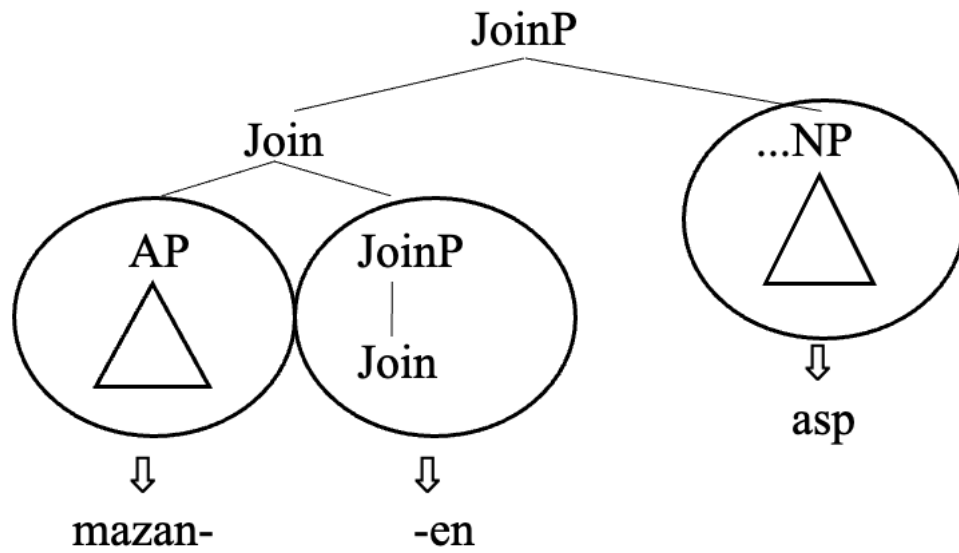
c.      JoinP  
    /      \  
    Join      

- (12) a. Attempt lexicalisation (1a)  
b. Lexicalisation fails; attempt specifier movement (1b)  
c. No specifier; attempt complement movement (1c)

(13)



(14) Project Join in the spine



# Prefixal ezafe in Persian and Kurdish

- However, the best studied cases of ezafe are prefixal. They precede the modifier.
- [Admittedly, morphophonological sensibility to the final segment of the previous form; there may be a prosodic side to the ezafe, which I ignore here]

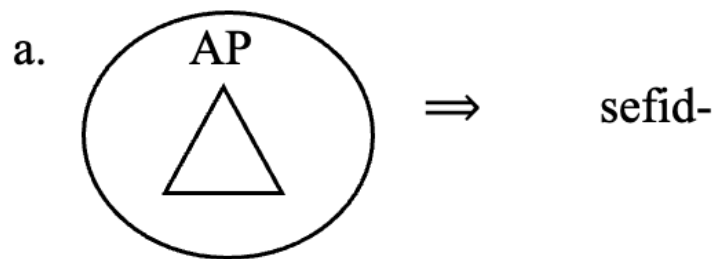
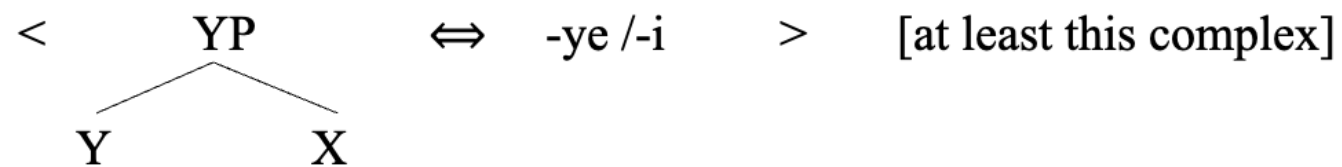
- (15) a. gorbe-ye iraani-ye sefid-e Maryam  
cat-ez Persian-ez white-ez Maryam  
'Maryam's white Persian cat'  
[Persian, Kahnemuyipour 2014: 2]
- b. kteb-i sur  
book-ez red  
'a red book'  
[Kurdish, Karimi 2007: 2164]

– Ezafe never appears with prenominal elements, but prenominal elements happen to coincide with quantifiers (eg., the 'adjective' *only*, which happens to be a quantifier declaring an intersection between sets of cardinality |1|)

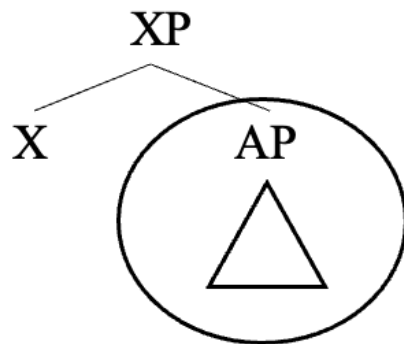
- (16) a. The prefixal nature of ezafe means that it must have a branching foot (by 1d).  
b. Any movement operation will be unable to rescue the derivation.

- Ezafe must contain more material beyond JOIN.

(17) Persian ezafe



b.



Attempt lexicalisation (1a)

Lexicalisation fails; attempt specifier movement (1b)

No specifier; attempt complement movement (1c)

Lexicalisation fails, and any movement will fail because ezafe has binary foot.

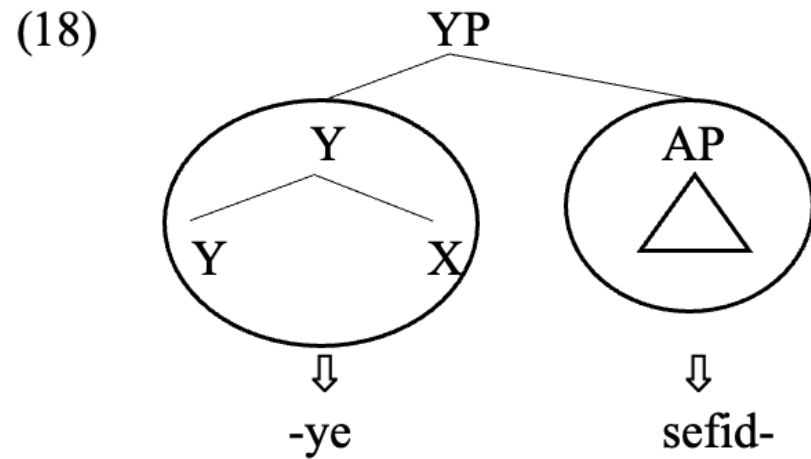
Spawn new derivation.

– DeClercq (2019): once you open a new working space, exploit it as much as possible.

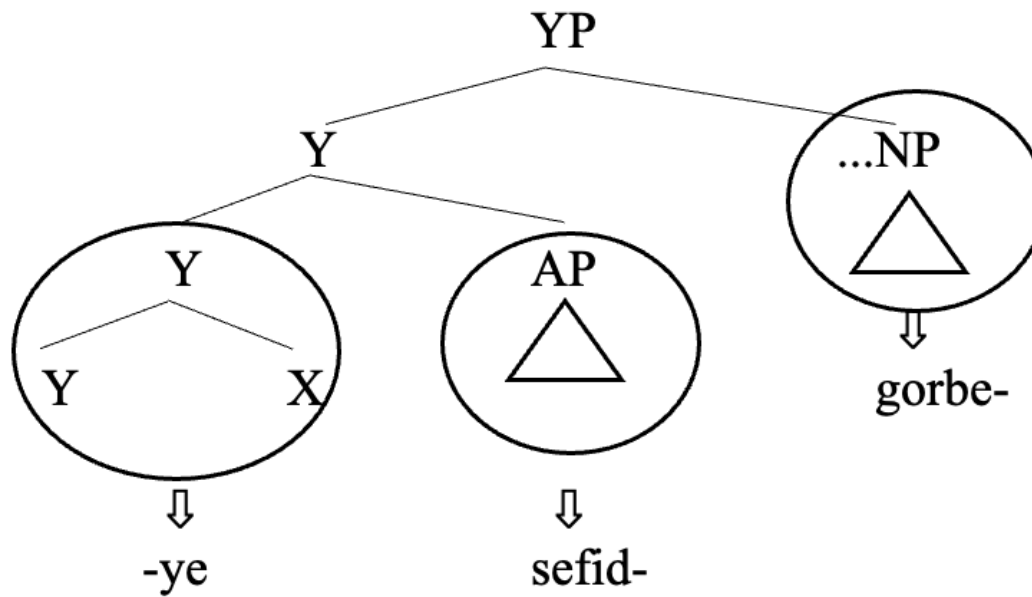


– DeClercq (2019): once you open a new working space, exploit it as much as possible.

This translates as 'merge all the structure of the ezafe'.

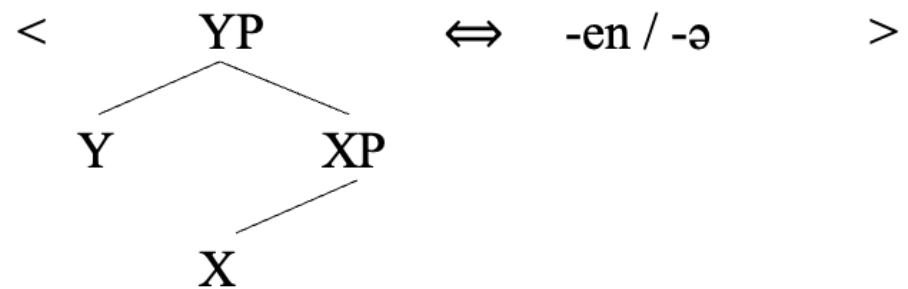


(19) Project Y in the nominal spine.

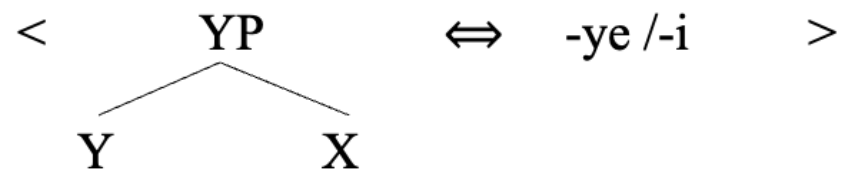


- Two desiderata:
  - Features are universal.
  - Languages differ only on the size and shape of exponents.

(20) Balochi / Gilaki ezafe, revised version



(cf. Persian ezafe)



# Agreeing prefixal ezafe in Zazaki

– So what are Y and X? Let us look to Zazaki.

- (21) a. Ju    bız=a                      gırs=e                      vaş                      wen-a.  
          one   goat=ez.f.sg.nom    big=f.sg.nom    grass                      eat-3sg.f  
          'A big goat is eating grass'
- b. Kutık=o                      gırs=ø                      mi                      vinen-o.  
          dog=ez.m.sg.nom            big=m.sg.nom 1sg.obl    see-3sg.m  
          'The big dog sees me'

[Kahnemuyipour & Peters 2019]

– There is agreement in the ezafe.

– The agreeing ezafe is distinct from non-ezafe agreement

# Ezafe agreement $\neq$ other agreement

Table 1: Zazaki Case Paradigm

		NOM.	OBL.
SG.	M.	$-\emptyset$	$-i/y$
	F.	$-e$	$-e$
PL.	M./F.	$-i/y$	$-u(n)$

Table 2: Zazaki Ezafe Paradigm

		NOM.	OBL.
SG.	M.	$-o$	$\hat{e}$
	F.	$-a$	$-a$
PL.	M./F.	$-\hat{e}$	$-un-\hat{e}$

- Therefore:

- Ezafe = [Concord [Join]]

- Consequently, ezafe contains agreement

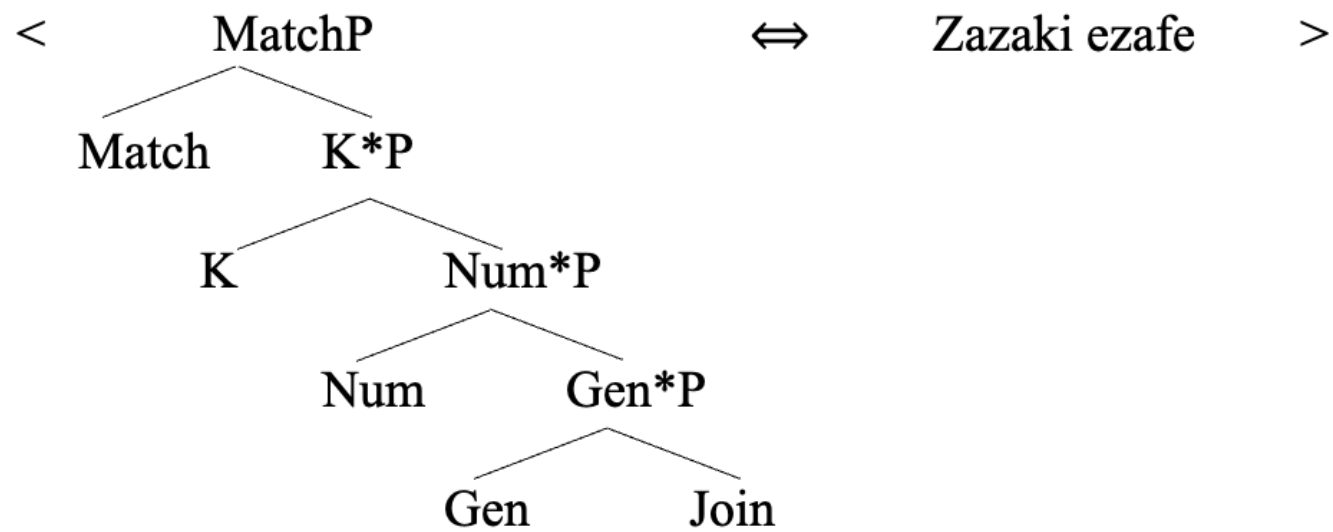
– Agreement is necessary to semantically unify the two elements as describing the same referent (HPSG, Sag & Pollard 1984)

– There needs to be a Match operation that checks the two sets of features for compatibility.

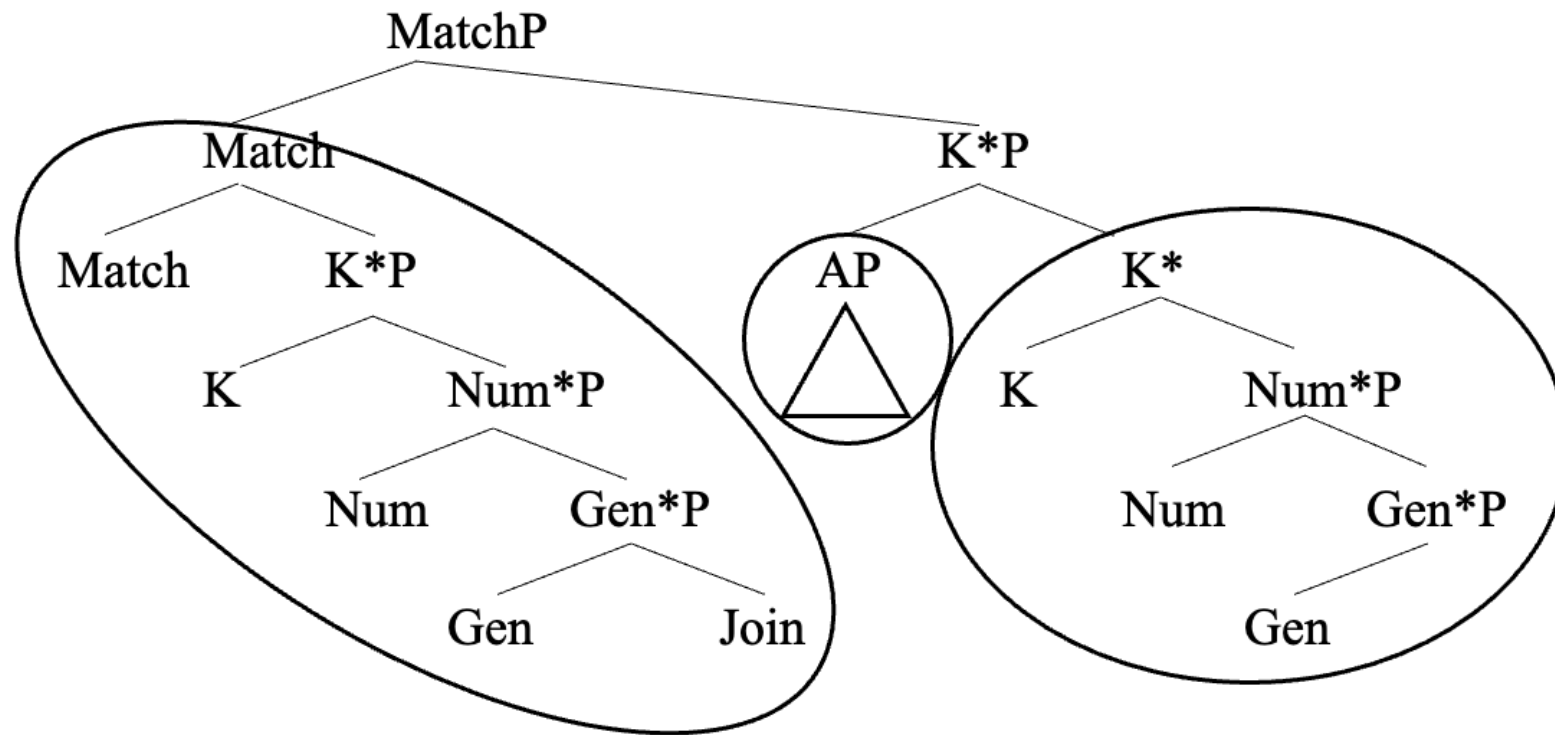
- (23)
- a. JOIN: type shifter from  $\langle e, t \rangle$  to modifier
  - b. Concord: introduces information to syntactically identify the modified entity
  - c. Match: associates the modified with the modifier



(24) Lexical entry



(25) Combine with adjective



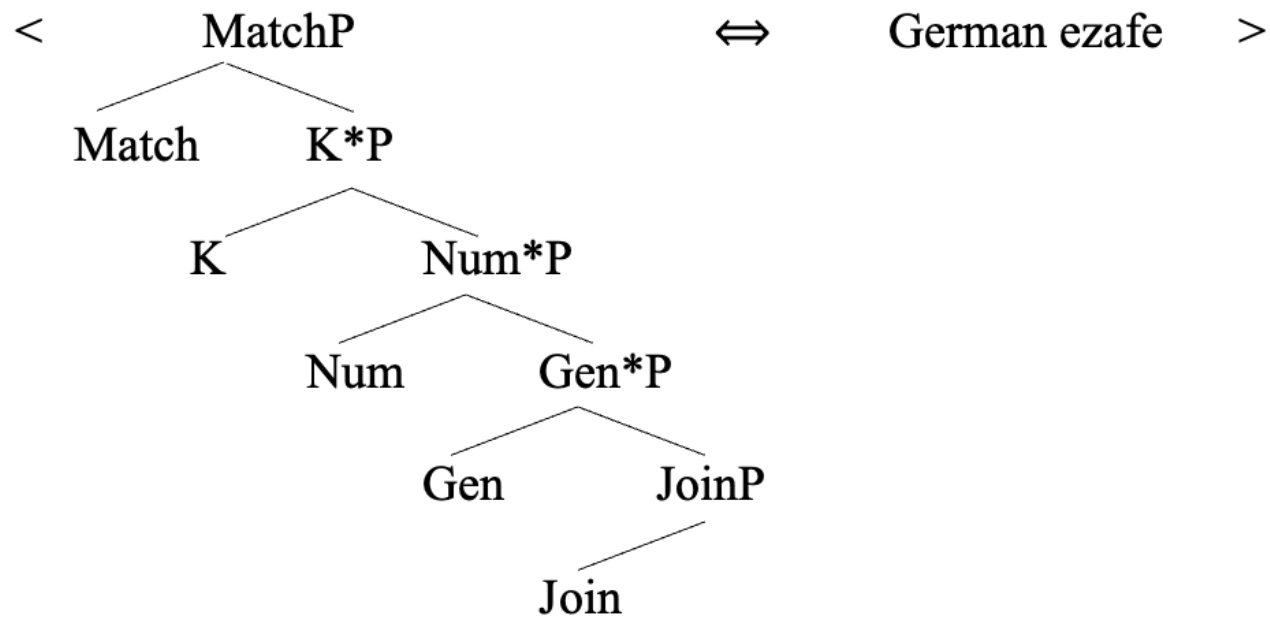
# Agreeing suffixal ezafe in German

(27) German concord is clearly an ezafe

a.	ein	grün*(-er)	Bus
	a	green(-Conc)	Bus
	'a green bus'		
b.	Ein	Bus	ist grün(*-er).
	a	bus	is green(-Conc)
	'A bus is green'		

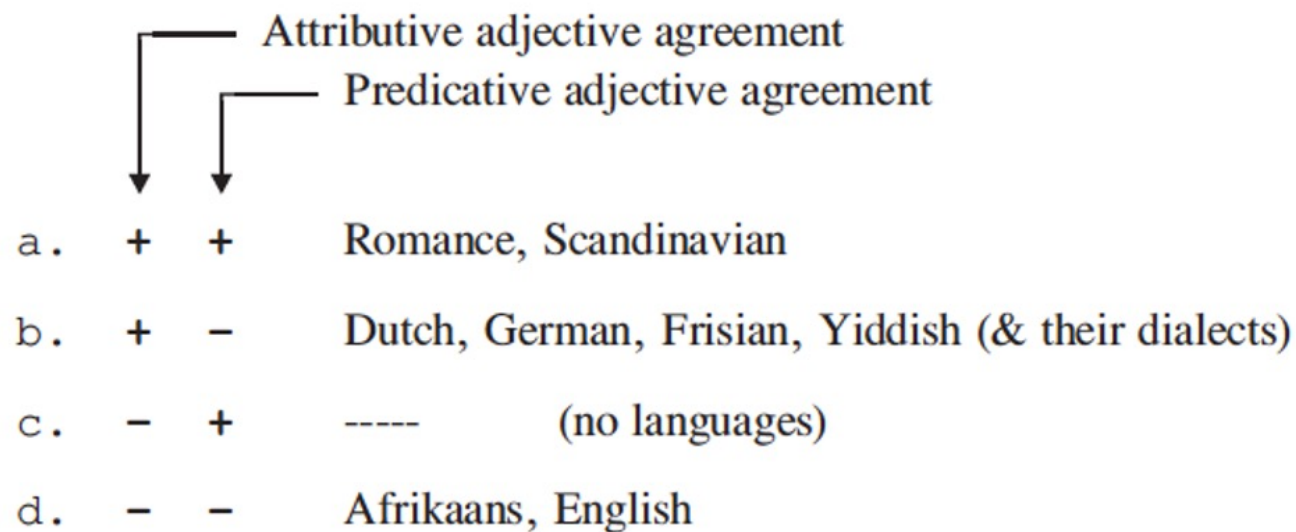
- Noun - Adjective agreement only emerges when the adjective is a modifier.
- When the adjective is a predicate, Concord is ungrammatical.
- This is what we expect if German is like Zazaki, only that ezafes are suffixal here.

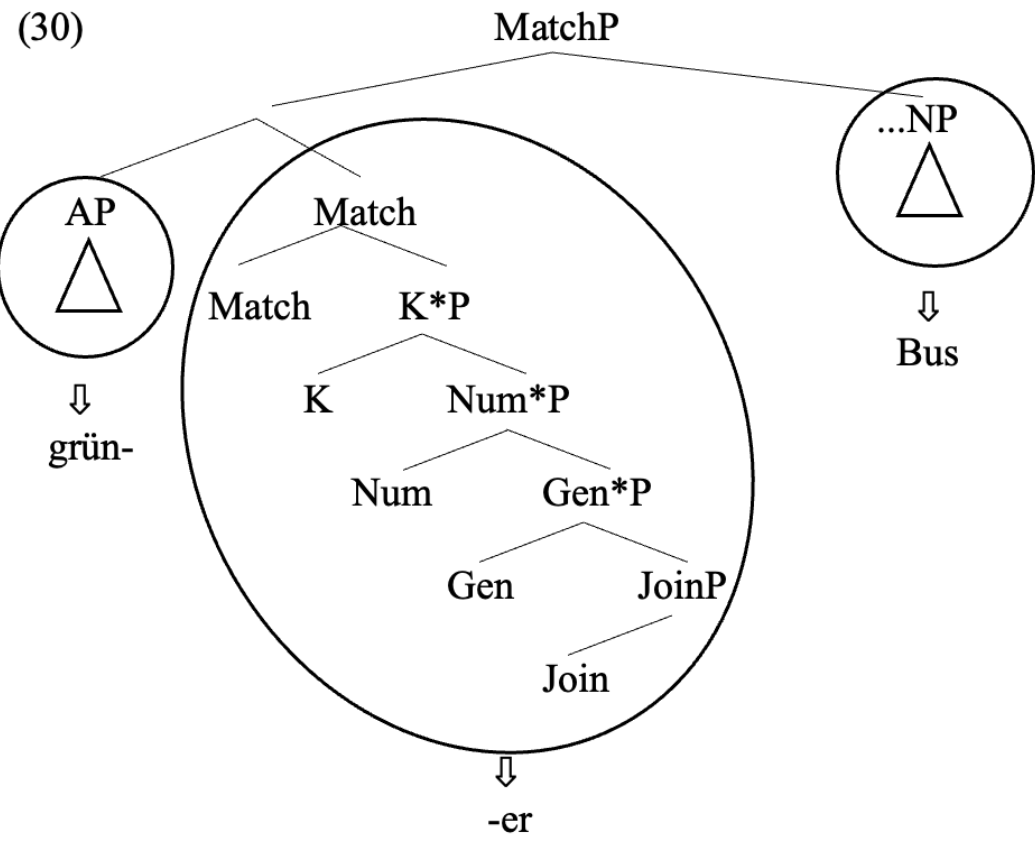
(28) Lexical entry



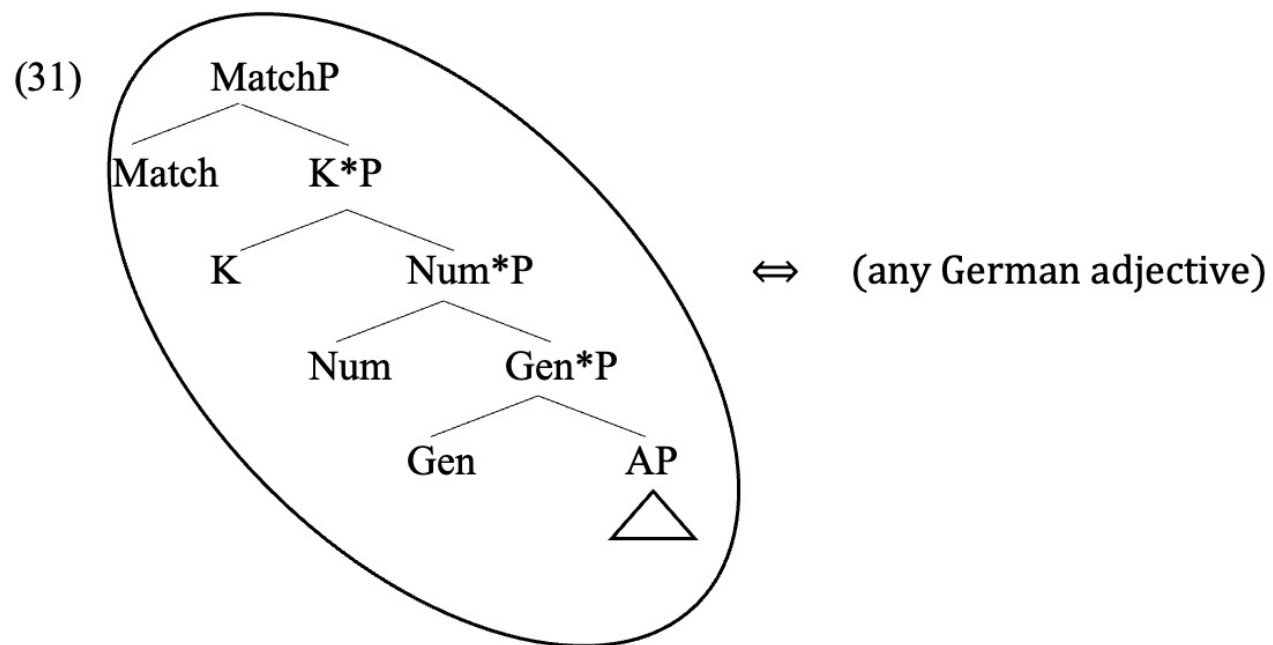
Vikner (2009): languages may have agreement in both predicative and attributive contexts; they may lack it in either, or they may only have it in attributive position, but no language has it only in predicative position.

(29) Vikner (2009: 9, his example 23)



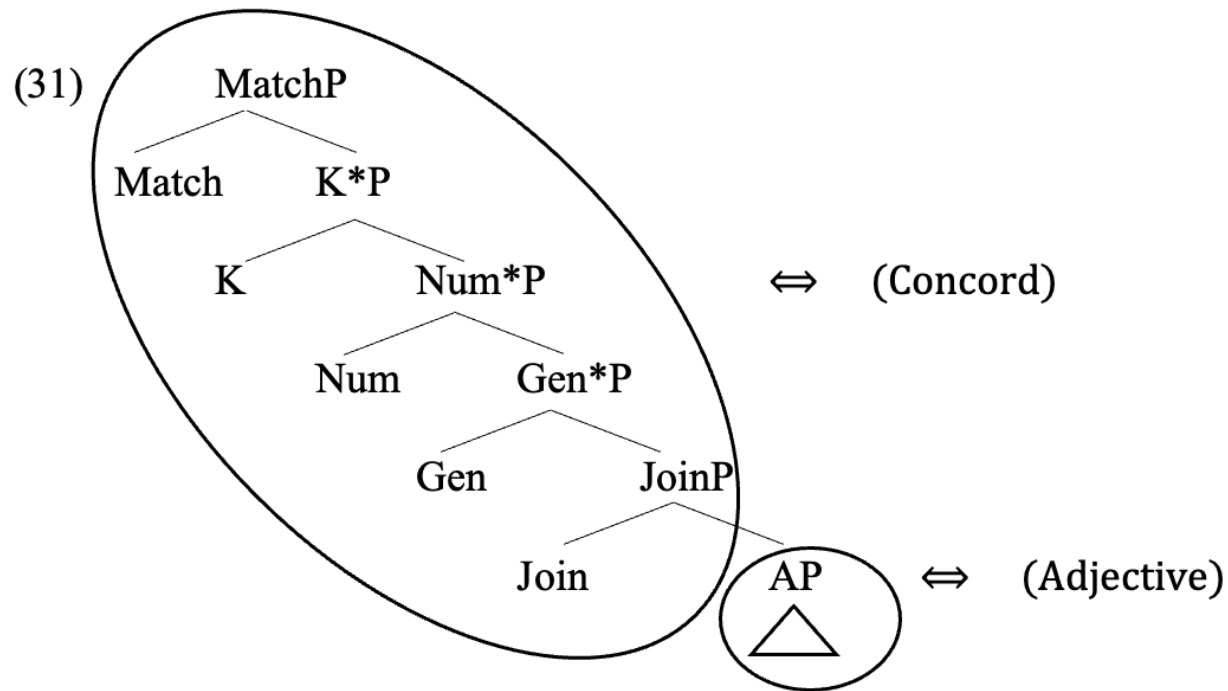


– Without JOIN: German adjectival exponents go up to MatchP, and spell out the relevant set of phi features.





- When JOIN is present between AP and Gen\*, the constituent gets broken.

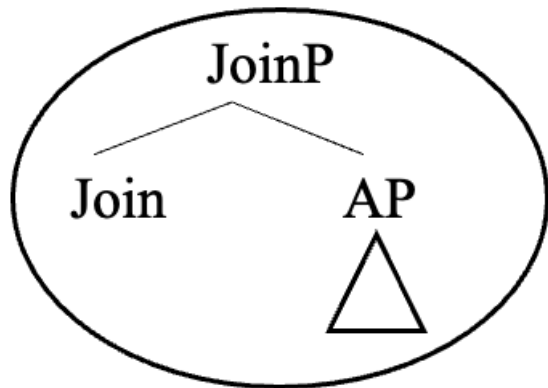


# Ezafe in Romance

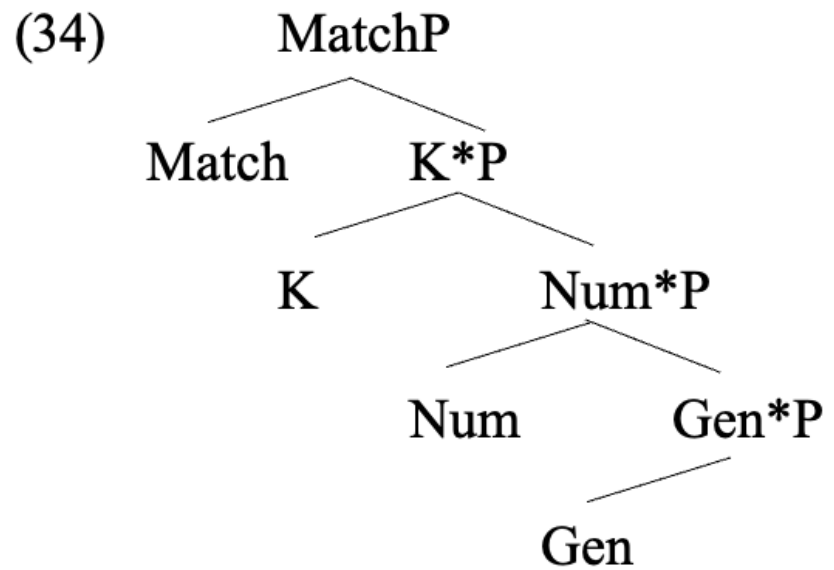
– Spanish and many other languages must have concord both in attributive and predicative position.

- (32) a. buen \*(-a-s)      notici-a-s  
      good    -f-pl      news-f-pl  
      'good news'
- b. L-a-s      notici-a-s      son      buen \*(-a-s).  
          the-f-pl    news-f-pl      are      good    -f-pl  
      'The news are good'

(33)



⇐ [any adjective that can be attributive]



# Conclusions

- Agreement is a relational head in the sense that it is necessary to establish a particular type of relation.
- That relation is unifying participants when their descriptions are distributed across distinct syntactic constituents.

# References



# Appendixes

## Appendix I. Determiners that have agreement.

Obviously, in a standard account determiners do not have JOIN, and ezafe normally does not affect them. So what happens when they agree?

I have to adopt a generalised version of Leu (2015): Ds contain As in all cases.

(i) D takes [xAP] as complement.

[<sub>xAP</sub> the AGR HERE] (ibidem 2015:12)

– There is a silent adjective within Ds, and agreement associates to the adjective. In Leu this is the case of many Ds, but not all Ds: the articles lack an adjective.

– It is crucial in my approach that all Ds with agreement have an adjective that adds properties to the noun.

– Hardest case: definite article

(ii) [Agr [AP 'the unique X in context']]

– I would need to treat the D head (external to xAP) as a function that turns the adjectival property into a quantifier.



### **Appendix III. Languages without agreement or concord.**

They are probably not a single group of languages. Three options, with distinct predictions, come to mind:

- Pointer connecting JOIN and the rest of the tree so that JOIN does not destroy the constituent (prediction: it would be a big coincidence if NO adjective showed some form of change)
- The language lacks adjectives.
- Adjectives in the language are actually roots.

## Appendix IV. Left branch movement and morphological richness

Bluemel (2022): languages that allow left branch movement (Ross 1967) are languages that inflectionally differentiate between case suffixes.

(vii) BSC (Bošković 2008)

a. Čijeg<sub>i</sub> si video [*t<sub>i</sub>* oca]?  
whose are seen father

'Whose father did you see?'

b. Lijepe<sub>i</sub> je video [*t<sub>i</sub>* kuće].  
beautiful is seen house

'Beautiful houses, he saw'

(viii) a. \*Whose did you see father?

b. \*Beautiful, he saw houses.

– I agree with Blumel that this has to do with labeling, but in a different sense:

(Evacuation movement does not relabel)

(1) Spell out algorithm (Cortiula 2023, Caha et al. 2024)

a. Merge F and lexicalize.

b. If fail, evacuate the closest labelled non-remnant constituent and try (a)

c. If fail, evacuate the immediately dominating constituent and try (a) (recursive).

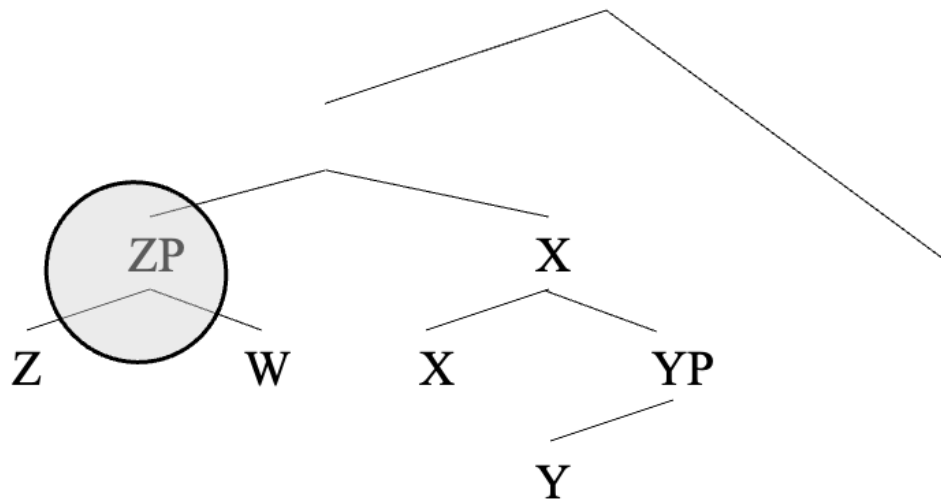
– Notice the contrast between (1b) and (1c).

- This distinction reflects then two ways of obtaining (complex) left branches:

- (ix) a. By evacuation movement (non-projecting)
- b. By spawning a new working space (projecting)

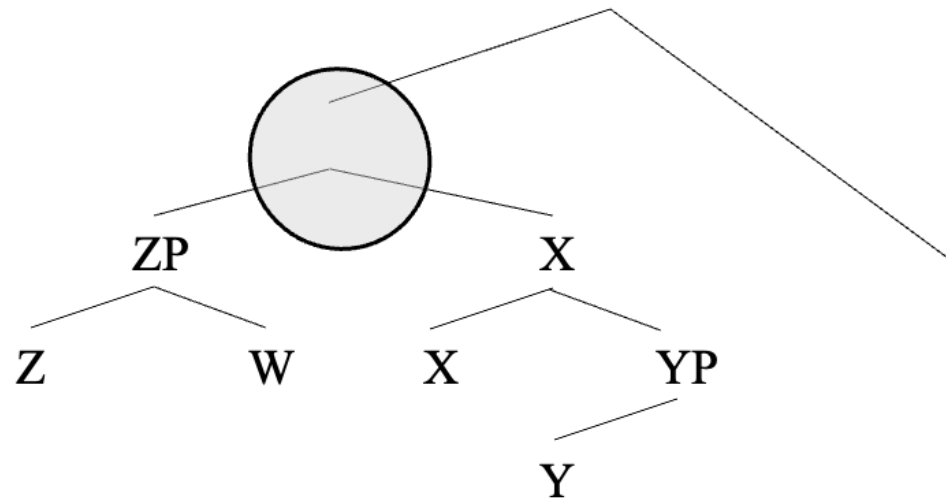
– Complex left branch by evacuation

(x)

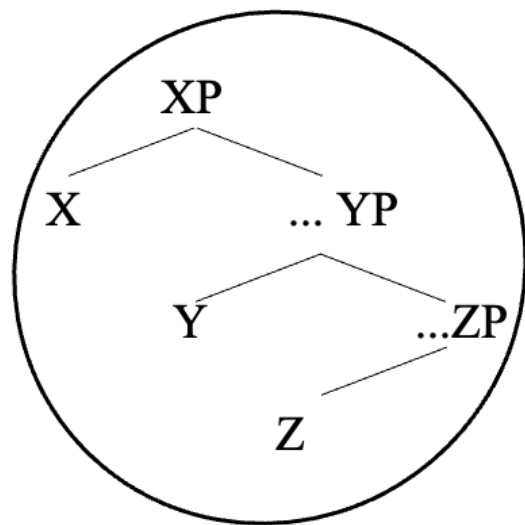


If that fails...

(xi)



(xii) Suffixes in languages without D and with case suffixes:



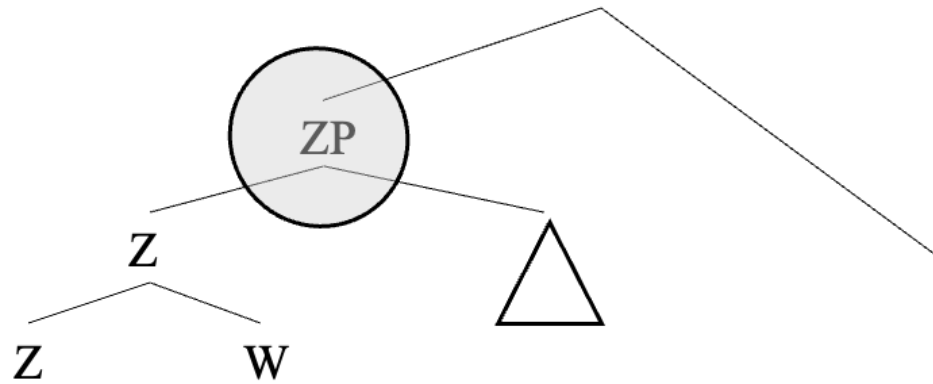
⇒ highest suffix must include D and K\*

– Consequently, anything generated anywhere in the D / K\* area will have to evacuate the suffixal material (which leads to x).

- Left branch condition languages
  - No case suffixes and D in the other set of languages: project a new derivation.



(xiii)



In (ix) the first movement attempted will be Z, and one will never try to move the smaller left branches.