Two paradigms of Balkar agreement and the shape of Nanosyntactic trees[1]

Daniar Erlanovich Kasenov (HSE University) antidanyar@protonmail.com

Intro: In Balkar (a Turkic language, spoken in Kabardino-Balkariya, Russia), there are two verbal person agreement paradigms presented in tables (a) and (b). Affixes of (a) paradigm are found after aspectual affixes and affixes of (b) paradigm are found after past tense, optative, and conditional morphology. This work aims to give a morphosyntactic analysis of this pattern in Nanosyntax, a realizational syntactic theory of morphology, building on recent Nanosyntactic work on agreement paradigms (Starke 2021).

Problems: Nanosyntax, unlike its neighbour, Distributed Morphology, disallows context-sensitivity of realization rules. Hence, whenever there are different forms, a nanosyntactician expects different structures to be realized there. Therefore, the existence of two agreement paradigms implies that affixes corresponding to the same φ-feature set in the paradigm have to realize different parts of the structure. On a less theory-specific note, another interesting problem is the fact that the paradigm of first-second person possessive affixes (table (c)) consists of one half of (a) and one half of (b). Clearly, this property should follow from an adequate analysis of Balkar data.

The framework: Nanosyntax (Starke 2009) is a theory of morphology where morphology realizes abstract structures built from partitive features acting as syntactic heads. The exponence rules of Nanosyntax match constituents of the S-tree (the result of derivation) with the L-trees (structure-form pairs, parts of the lexicon) via the Superset Principle (an L-tree may lexicalize its subconstituents). Following Starke 2020, I assume the following hierarchy of verbal and agreement feature-heads: Asp–T–PST–#–PL–π–PART–SPKR. Of course, the PL will be absent in singular agreement, PART will be absent in 3rd person agreement and so on. Finally, I will use the idea proposed by Starke 2022 and used in Caha and Taraldsen Medová (2022) that Nanosyntactic process of spell-out is sensitive to the derivational history of structures (via structural reflexes of movement).

The solution: The L-trees are provided below. Here, I should specify the motivation behind them. The first observation is the paradigm form of (b)'s plural subparadigm: the affix *-lA* is overwritten by *-ʁIz* (2PL) and *q* (1PL). Accordingly, the *m* and *ŋ* affixes are structured in a parallel fashion (without the PL feature). A final observation about the (b) subparadigm is the fact that *-dI* realizes 3SG φ-features – everything follows from this observation combined with backtracking (see the derivations below).

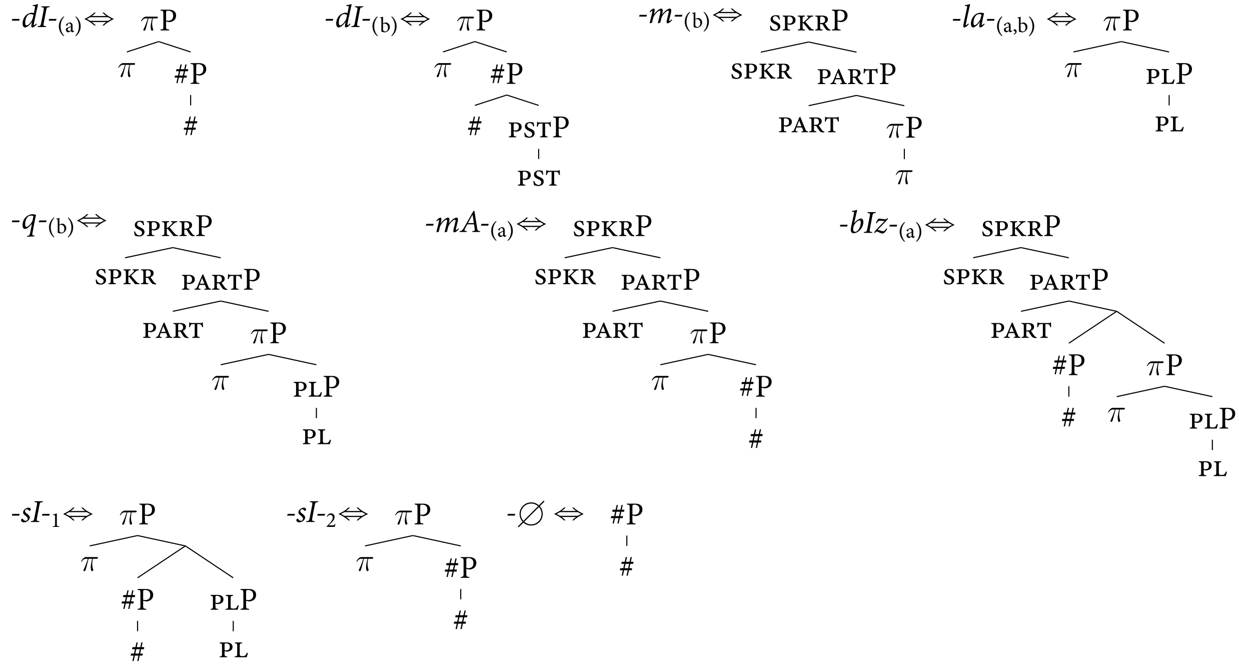
The most important observation about the paradigm (a) is the fact that *-bIz* and *-sIz* overwrite two affixes at the same time (*-dI-lA*). This observation motivates the complex (so-called movement-containing, Caha and Taraldsen Medová 2022) structure for *-bIz* – see derivations for how this happens. Other than that, the paradigm (a) is rather straightforward: the *-dI* affix there only realizes φ-features, which allows for #-including singular affixes and the complex structure for plural affixes.

From the analysis of verbal paradigm, important observations about possessive paradigm follow: firstly, *m* and *ŋ* affixes do not realize the # feature, and we need to postulate a null realization of # in the posessive paradigm. On the side of the plural affixes, forcing a complex structure appears necessary to end up with *-bIz* and *-sIz*, which is done with the plural version of *-sI*. I should note that both the null affix and the homophony of *-sI* are not inevitable, but the main claim of this abstract is the possibility of a uniform analysis for the first/second person affixes across the three paradigms.

Conclusion: This work has presented an analysis of three agreement paradigms of Balkar in Nanosyntax, a bottom-up derivational approach to lexicalization of fine-grained syntactic structure. See below for L-trees and derivations with movement-containing structures and which shows the L-trees below in action.

[1] This work is supported by RSF grant # 20-512-26004 'Morphosyntax of agreement'.

L-trees for the three paradigms. Second person affixes are first person affixes without SPKR layer.



Paradigm (a):

	SG	PL
1	bar-a-ma go-IPFV-1SG	bar-a-bɨz go-IPFV-1PL
2	bar-a-sa go-IPFV-2SG	bar-a-sɨz go-IPFV-2PL
3	bar-a-dɨ go-IPFV-3SG	bar-a-dɨ-la go-IPFV-3-PL

Paradigm (b):

	SG	PL
1	bar-dɨ-m go-PST-1SG	bar-dɨ-q go-PST-1PL
2	bar-dɨ-ŋ go-PST-2SG	bar-dɨ-ʁɨz go-PST-2PL
3	bar-dɨ- go-PST.3SG	bar-dɨ-la go-PST-3PL

Possessive paradigm (c):

	sg	pl
1	bala-m child-1SG	bala-bɨz child-1PL
2	bala-ŋ child-2SG	bala-sɨz child-2PL
3	bala-sɨ child-3	bala-sɨ child-3

References: • Caha, P. and Taraldsen Medová, L. (2022). On the feminine singular declension of the Russian demonstrative. Talk given at FDSL 15, https://is.muni.cz/publication/2222757/fdsl15-dem.pdf. • Starke, M. (2009). Nanosyntax: A short primer to a new approach to language. *Nordlyd*, 36(1):1–6. • Starke, M. (2018). Complex left branches, spellout, and prefixes. *Exploring nanosyntax*, pages 239–249. • Starke, M. (2021). Reply to comments on Universal Morphology. *Isogloss. Open Journal of Romance Linguistics*, 7:1–10. • Starke, M. (2022). Movement-containing trees in Nanosyntax. Unpublished ms.

(1) Nanosyntax lexicalization algorithm (Caha and Taraldsen Medová 2022).

a. Merge F and spell-out

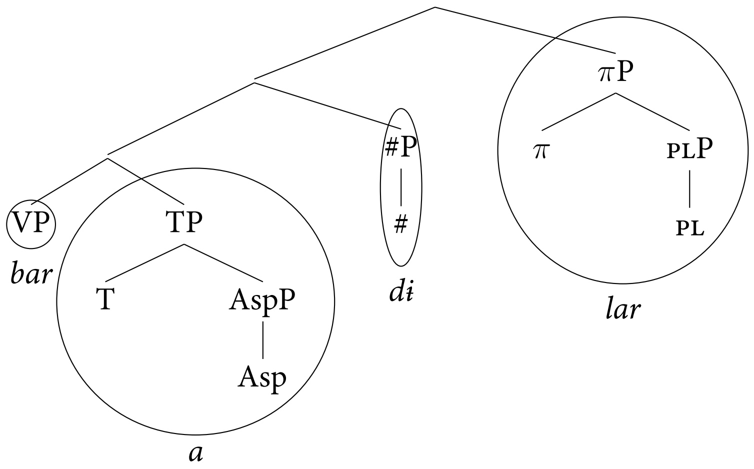
b. If previous step failed, move the closest non-remnant constituent

c. If previous step failed, move the dominating node (recursive)

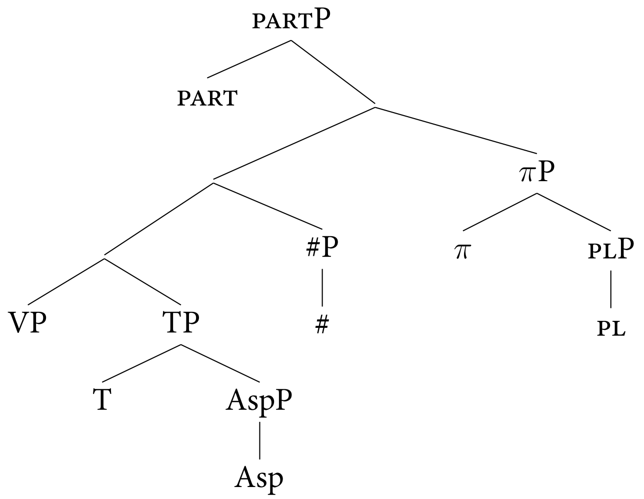
d. Go back to the last cycle (backtracking)

(2) Partial for plural subparadigm of paradigm (a)

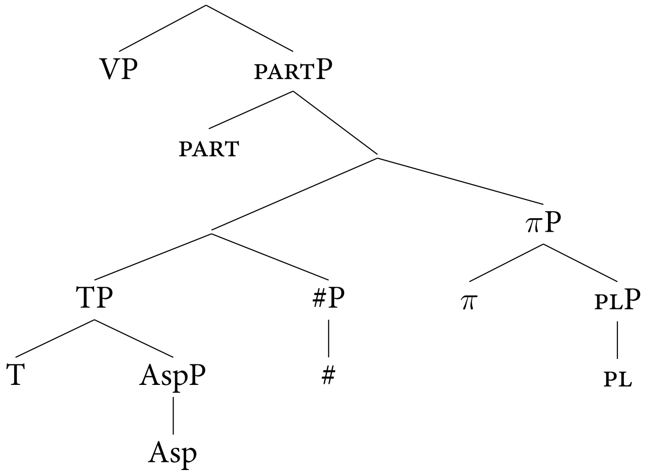
a. Step 0



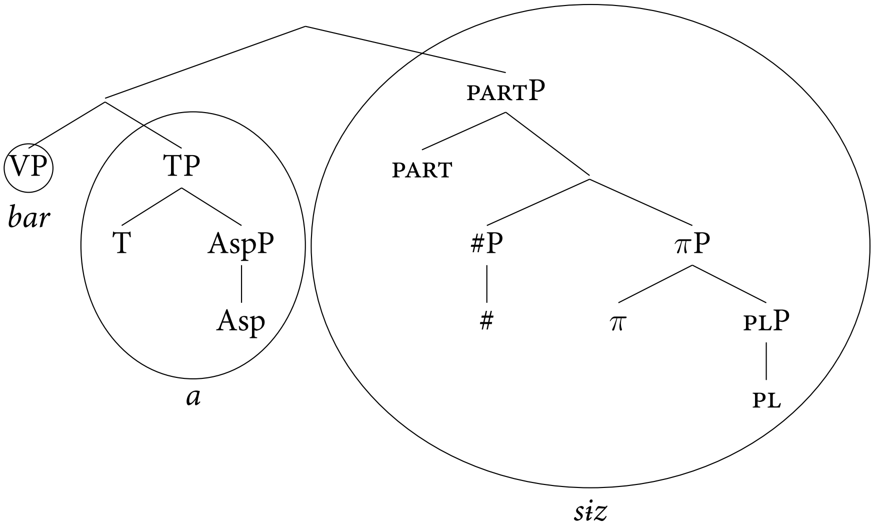
b. Merge PART



c. Move the closest non-remnant constituent

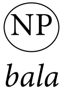


d. Move the dominating node

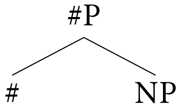


(3) Derivation for plural subparadigm of paradigm (c)

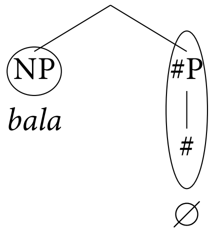
a. Step 0



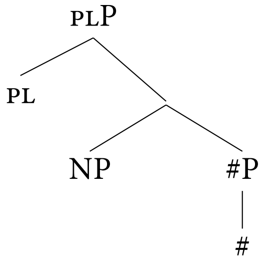
b. Merge #



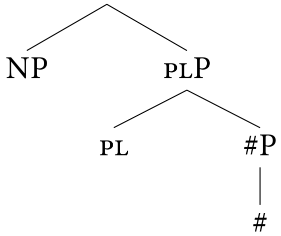
c. Move the closest non-remnant constituent



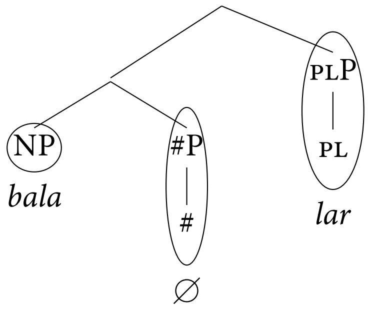
d. Merge pl



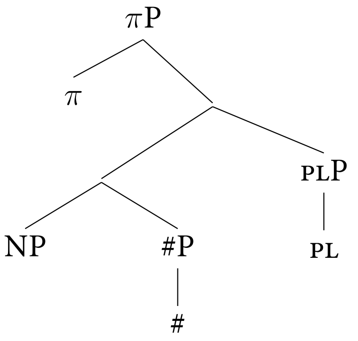
e. Move the closest non-remnant constituent



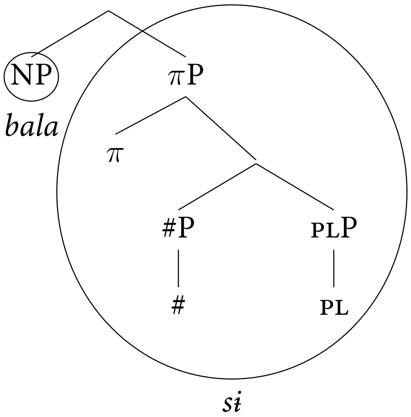
f. Move the dominating node



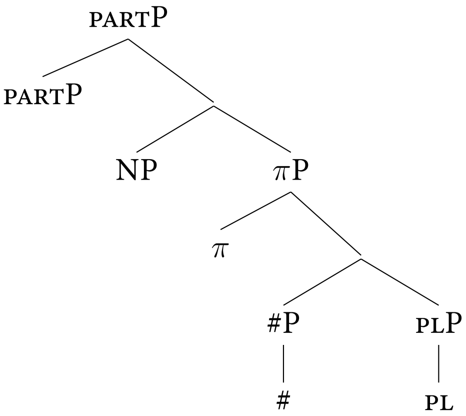
g. Merge π



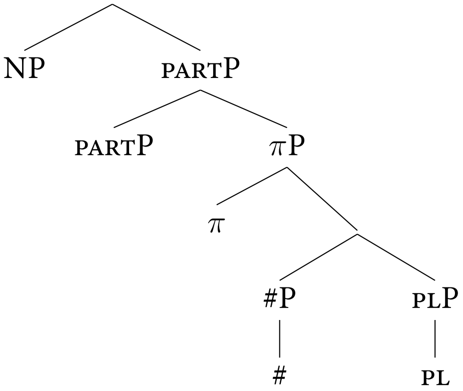
h. Move the closest non-remnant constituent



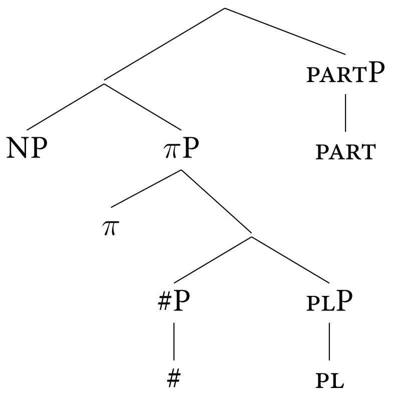
i. Merge PART



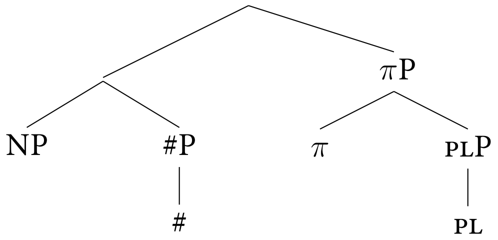
j. Move the closest non-remnant constituent



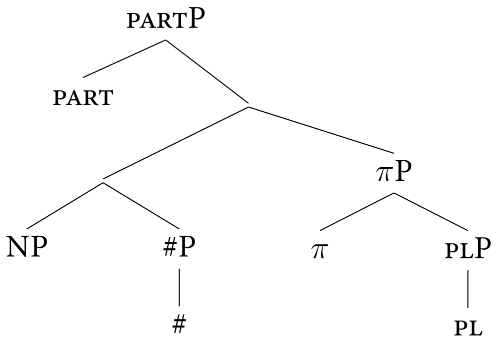
k. Move the dominating node



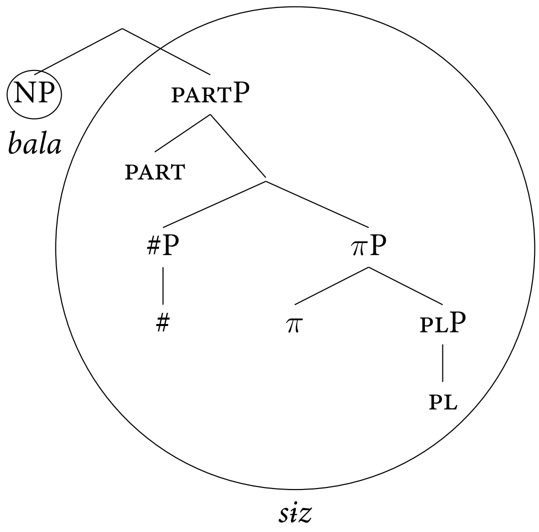
l. Backtracking. Move the closest dominating node



m. Merge part



n. Move the closest non-remnant constituent


NP
bala
partP
part
#P
πP
#
π
plP
pl
siz