

## \*ABA in Romeyka Verbal Paradigms

Merve Yazar, Ömer Demirok\*

**Abstract.** In this work, we analyze present-tense and past-tense paradigms in Romeyka, an understudied Indo-European language belonging to the Hellenic sub-branch, spoken mainly in the Trabzon province in Turkey. We argue for decomposing the endings in the plural forms based on the regularities across present and past tense paradigms. While the proposed decomposition is empirically motivated, it leads to a theoretical challenge known in the literature as \*ABA pattern, which has been extensively argued to be non-existent in morphology (e.g., Caha, 2009; Bobaljik, 2012; Middleton, 2020). We demonstrate this challenge for Nanosyntax (Starke, 2009), which predicts that syncretism will only be possible for structurally contiguous features. In an attempt to address this puzzle, we develop an account which relies on the idea that languages may differ in whether they single out the Speaker or the Addressee features in building first and second-person structures. Accordingly, the \*ABA in Romeyka dissolves into a predicted pattern where the syncretism is the non-offending AAB pattern.

**Keywords.** \*ABA pattern; syncretism; Romeyka; Nanosyntax

**1. Introduction.** Bobaljik (2012) presents a sketch of comparative and superlative patterns across languages, concentrating on roots where we observe suppletive allomorphy as in (1):

(1) good-better-best

In order to refer to suppletive forms, he adopts using variables A, B, C; where the same letter is used to indicate syncretic forms. Assuming that we label *good* in (1) as A, the comparative root *bett-* is marked as B due to the forms being phonologically distinct. Considering that the comparative and the superlative roots in (1) are phonologically similar, the superlative root *be-* is also labeled as B; overall the paradigm corresponding to an ABB pattern.

Following this logic, ABA pattern refers to syncretisms between non-adjacent forms in a given paradigm; e.g., syncretisms between the first and the third items in a sequence, here exemplified as the positive and the superlative root *good* in (2):

(2) \*good-better-goodest

Bobaljik (2012) further argues that ABA patterns are non-existent in morphology. The absence of the pattern across languages led Bobaljik & Sauerland (2018, p. 1) to come up with the \*ABA *generalization*, stating that “...morphological patterns in which, given some arrangement of the relevant forms in a structured sequence, the first and third may share some property ‘A’ only if the middle member shares that property as well”. This translates to the idea that first-third syncretisms, namely ABA forms, are not attested across languages; hence, the asterisk in \*ABA.

The ban of the emergence of ABA has been extensively reported in the literature as well. The earliest observation dates back to Blansitt (1988)’s work where he demonstrates a

---

\* We are grateful to Pavel Caha for sharing his comments. We also thank the reviewers for their helpful feedback. Authors: Merve Yazar, Boğaziçi University ([merve.yazar@boun.edu.tr](mailto:merve.yazar@boun.edu.tr)) & Ömer Demirok, Boğaziçi University ([omerfaruk.demirok@boun.edu.tr](mailto:omerfaruk.demirok@boun.edu.tr)).

ban of syncretism over forms that are not adjacent in the linear sequence of DATIVE-ALLATIVE-LOCATIVE. In other words, DATIVE and LOCATIVE forms cannot be syncretic excluding the ALLATIVE. Caha (2009) makes similar observations on case by providing cross-linguistic data. Further work involves Lander & Haegeman (2016)’s study on spatial deixis, Middleton (2020)’s work on pronominal forms, Starke (2009) on Germanic verbs.<sup>1</sup>

Different accounts have been proposed in an attempt to explain \*ABA. This work mainly adopts the one proposed by Bobaljik (2012) but also see others (e.g., Kramer, 2016; Caha, 2017 based on Bobaljik & Sauerland, 2018; Graf, 2017). Bobaljik comes up with the *Containment Hypothesis*, stating that what is realized as the superlative contains the comparative across languages. Assuming the following representation for the superlative in (3c), he concludes that the superlative is built as a result of merging the SUPERLATIVE feature upon what is realized as the comparative in (3b):

- (3) a. [ ADJECTIVE ]  
 b. [ [ [ ADJECTIVE ] COMPARATIVE ]  
 c. [ [ [ ADJECTIVE ] COMPARATIVE ] SUPERLATIVE ]

With this representation, ABA is logically impossible to derive. Assume that a form X lexicalizes (3a). Considering that a suppletive form Y lexicalizes (3b), X cannot lexicalize (3c) since in this decomposition, (3c) structurally *contains* (3b). In fact, there are only two ways to lexicalize (3c) with this representation: either with Y or a novel suppletive form Z. The former option results in having syncretic comparative and superlative forms where the paradigm overall corresponds to an ABB pattern while the latter brings about an ABC pattern where all the forms are distinct. Bobaljik & Sauerland (2018) further restate the scope of this generalization and assume \*ABA for any paradigm with three exponents.

Nanosyntax, a realizational model of morphology, also assumes a containment relation between syncretic forms (Starke, 2009). In other words, syncretisms are only possible for structurally contiguous features in a given functional sequence. This means that its algorithm cannot derive ABA forms, either.

This work presents an ABA puzzle for Nanosyntax from Romeyka, an understudied Indo-European language belonging to the Hellenic sub-branch, spoken mainly in the Trabzon province in Turkey. In section 2, we present the data and the theoretical problem it brings about. Section 3 deals with the proposal by focusing on the decomposition of person features. In section 4, we provide the Nanosyntactic derivation of the data by giving background on the tools of Nanosyntax. Section 5 is the conclusion.

**2. Data.** Romeyka has different paradigms for present and past tenses. The paradigms of the verb *val-* ‘put’ are given in Table 1. Notice that there is a systematicity in the way overt present tense suffixes relate to their counterparts in the past paradigm; initial vowels emerge as their lowered and unrounded counterparts in the past paradigm:

	Present Indicative	Past Indicative
1.SG	val-o	e-val-a

---

<sup>1</sup> There are also counter-arguments on \*ABA such as Andersson (2018). They argue for the existence of the pattern in verbs in Swedish, Low German, and Gammalsvenskbymålet, an endangered Germanic language spoken in Ukraine. Based on the observation that Proto-Germanic exhibits zero ABA patterns in verbal paradigms, they have a diachronic explanation as to why we observe ABA; regular sound changes resulted in having the ABA pattern.

2.SG	val-is	e-val-es
3.SG	val-Ø	e-val-en
1.PL	val-ume	e-val-ame
2.PL	val-ite	e-val-ete
3.PL	val-une	e-val-ane

Table 1: The verb “put” in Present Indicative and Past Indicative

We observe two patterns when we compare the present and past tense paradigms. First, we see a regularity in *me*, *te*, and *ne* emerging in the plural forms in both paradigms. Second, when *me*, *te*, and *ne* are removed from the suffixes, we observe syncretism between 1.PL and 3.PL forms in both past and present tense paradigms, suggesting that these patterns are non-accidental.

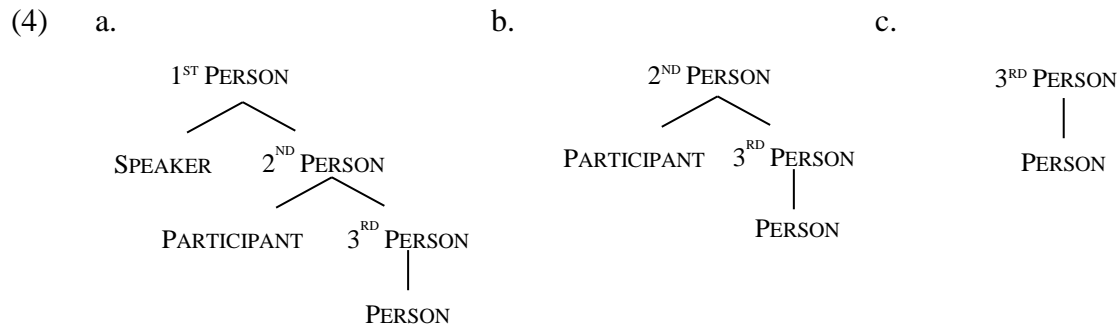
Both the syncretism between 1.PL and 3.PL forms and the commonality of having *me*, *te*, and *ne* in both paradigms suggest systematicity, which we argue justifies a decomposition of the endings in the plural forms. However, this leads to a potential ABA pattern, as shown in Table 2, assuming that first and third person features are not contiguous features:

	Present Indicative	Past Indicative
1.PL	u	a
2.PL	i	e
3.PL	u	a

Table 2: ABA in Present Indicative and Past Indicative

### 3. Proposal.

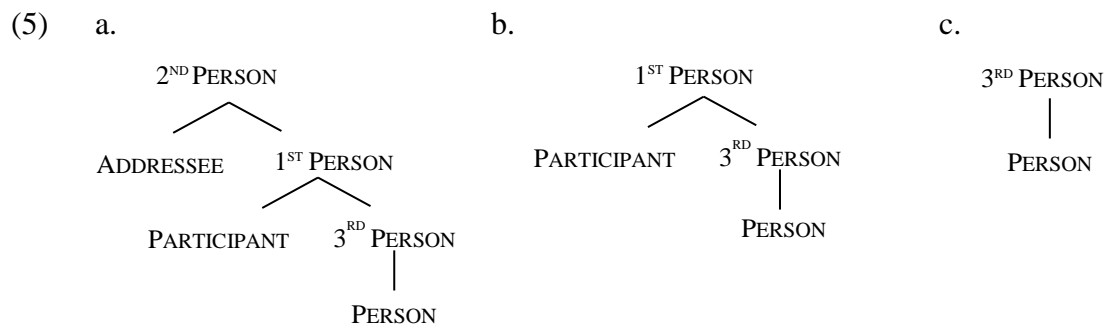
3.1. DECOMPOSING THE PERSON FEATURES. The proposal in this work is going to be motivated through person features. A widely adopted decomposition for first, second and third persons are given in (4) below:



Notably, this decomposition relies on privative features where the absence of SPEAKER feature in (4b) leads the hearer to reason that the structure picks out a participant which is not the speaker; therefore, the addressee. Similarly, the absence of the PARTICIPANT feature in (4c) leads to the reasoning that the structure refers to a third person, a person not in the conversation.

Adopting this decomposition (Starke, 2013; Wyngaerd, 2016), Nanosyntax predicts syncretisms to be restricted to contiguous features. Therefore, it predicts syncretisms between first and second and syncretisms between second and third persons but it rules out syncretisms between first and third persons, given the decompositions in (4).

3.2. AN ALTERNATIVE DECOMPOSITION. As an alternative, we propose that Romeyka has the decomposition for person features in (5).<sup>2</sup>



The decomposition in (5) offers an alternative look at person features by involving different privative features; instead of having a *SPEAKER* feature, we now have *ADDRESSEE*. The introduction of the *ADDRESSEE* feature in (5a) necessarily translates to a structure that involves a person that is both the participant and the addressee; hence, the hearer ends up reasoning that the structure opts for the second-person. The absence of the *ADDRESSEE* feature in (5b); on the other hand, is automatically interpreted as the first-person; a person that is a participant but not the addressee. Similarly, the absence of the *PARTICIPANT* feature in (5c) still translates to a third person that is not a participant in the conversation.

Harley & Ritter (2002) have already entertained this possibility for the pronoun system in Pirahã. Introducing this flexibility in the functional sequence of person structures allows us to account for the systematic syncretism between first and third persons in Romeyka. This is because the features that spellout the first and the second persons; namely *ADDRESSEE* and *PARTICIPANT*, are now contiguous in the hierarchy with this decomposition. In other words, the structure that is realized as the second-person is built on what is realized as the first-person. The next section offers a detailed Nanosyntactic derivation by adopting (5) for Romeyka.

#### 4. Nanosyntactic Derivation.

4.1. THEORETICAL BACKGROUND. Nanosyntax is a late-insertion model of grammar that is based on phrasal lexicalization. Each structure formation (i.e., Merge) activates the lexicalization algorithm to find a lexical entry for the newly created structure. The main computational tools in the framework are *the Superset Principle* (Starke, 2009), *the Elsewhere Condition* and *the Spellout Algorithm* (Starke, 2018). The formal definitions are provided below:

- (6) The Superset Principle (Starke, 2009, p.3): A lexically stored tree matches a syntactic node if and only if the lexically stored tree contains the syntactic node.
- (7) The Elsewhere Condition (Caha, 2019, p.99): When two entries can spell out a given node, the more specific entry wins. Under the Superset Principle governed insertion, the more specific entry is the one which has fewer unused features.

<sup>2</sup> A reviewer brings the question of whether this proposal speculates on the rarity of first-third syncretisms as opposed to other syncretism patterns to our attention. We acknowledge that our proposal does not say anything about the possibility that first-third type syncretisms could be rarer than second-third or first-second type syncretisms (and vice versa). Regardless of the actual frequency of the first-third type syncretism in the typology, the fact that it is attested is the empirical basis for our proposal.

(8) The Spellout Algorithm (based on Starke, 2018 & Caha, 2019):

- a. Merge F and spellout.
- b. If (a) crashes, move the specifier of the complement of the newly merged node (*aka. spec-to-spec movement*).
- c. If (b) crashes, move the complement of the newly merged node (*aka. comp-to-spec movement*).
- d. If (c) crashes, unmerge the newly merged feature and try the next option for that cycle (*aka. backtracking*).
- e. If (d) crashes, create a derivational auxiliary workspace with F and close the workspace once FP is successfully lexicalized, projecting F to the top (*aka. Early AUX Closure*).

4.2. DERIVING THE FORMS. The lexicon we assume is given in Table 3, Table 4 and Table 5, Table 6 and Table 7 in the form of lexicalization tables.

	VP	ASPP	TP	NUMBERP	PLURALP	PERSONP	PARTP	ADDP
PRS.2.PL	val		-i	-te				
PRS.1.PL	val		-u		-me			
PRS.3.PL	val		-u		-ne			

Table 3: The lexicon of present tense plural exponents

	VP	ASPP	PASTP	NUMBERP	PLURALP	PERSONP	PARTP	ADDP
PST.2.PL	val		-e	-te				
PST.1.PL	val		-a		-me			
PST.3.PL	val		-a		-ne			

Table 4: The lexicon of past tense plural exponents

	VP	ASPP	PASTP	NUMBERP	PERSONP	PARTP	ADDP
PST.2.PL	val		-es				
PST.1.PL	val		-a				
PST.3.PL	val		-en				

Table 5: The lexicon of past tense singular exponents

	VP	ASPP	TP	NUMBERP	PERSONP	PARTP	ADDP
PST.2.PL	val		-is				
PST.1.PL	val		-o				
PST.3.PL	val		-Ø				

Table 6: The lexicon of present tense singular exponents

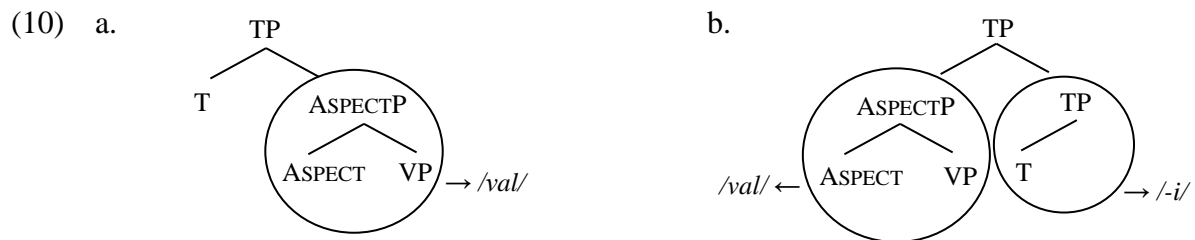
	ASP	TP	SHIIFTP
PST.2.PL	e-		
PST.1.PL	e-		
PST.3.PL	e-		

Table 7: The entry of the past tense prefix

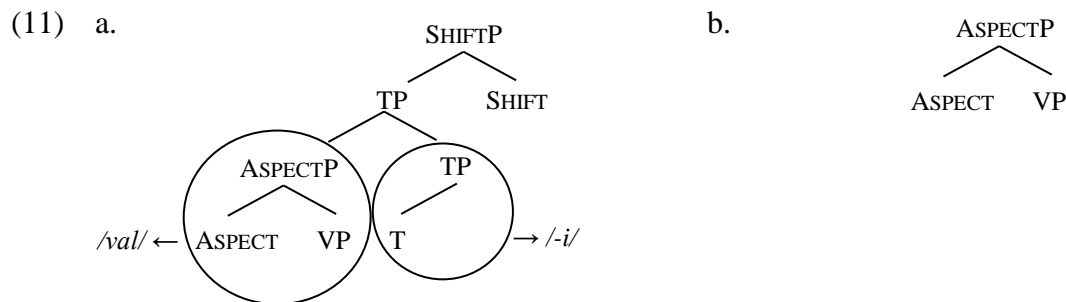
The functional sequence (fseq) we follow is given below in (9). Here, we adopt Gök’s (2023) novel SHIFT feature within the sequence. This feature is part of the fseq whenever there is a shift from the time of utterance (now) (for details, please refer to Gök, 2023). Accordingly, SHIFT is not part of the fseq when the tense is present but is part of the fseq when the tense is past. In other words, the fseq that yields present tense forms includes all the features in (9) except for SHIFT and PAST.

- (9) V > ASPECT > TENSE > SHIFT > PAST > NUMBER > PL > PERSON > PART > ADDRESSEE

We begin the derivation of the past forms with constructing the VP, which finds a match in the lexicon: *val* ‘to put’. In the next cycle, when we merge ASPECT with VP and create ASPECTP, *val* matches again the entire structure due to being an exact match. Next, we merge TENSE with ASPECTP, yielding the structure in (10a). In our lexicon, there is no entry that can spellout (10a); therefore, we resort to spec-to-spec movement. However, the complement ASPECTP does not have any specifier to move; therefore, we move on to the next step and do comp-to-spec movement. This results in (10b), *-i* successfully being inserted to TP.

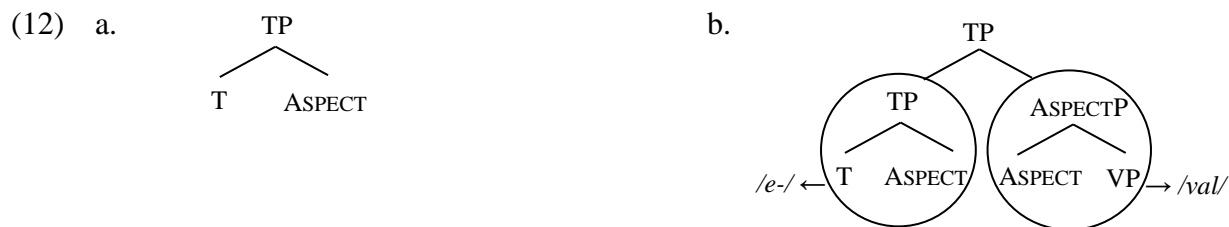


We further merge SHIFT on top of (10b). No match is found; therefore, we should try spec-to-spec movement. However, evacuating ASPECTP does not lead to successful lexicalization since there is no entry for SHIFTP dominating a unary-branching TP. Therefore, we do comp-to-spec movement. This operation gives us (11a). However, since there is still no match, we move on to the third option in the Spellout Algorithm and we backtrack. We delete SHIFTP from the structure and try the next step for the cycle that we are left with once we remove SHIFT, which is (10b). Comp-to-spec movement lexicalized (10b), so we try the next option in the Spellout Algorithm for that cycle. This means that we once again backtrack; delete the TP and end up with (11b).

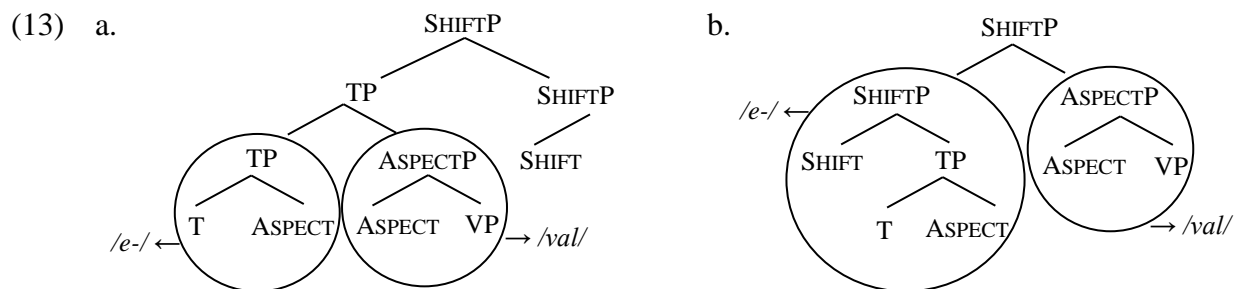


At this point, we are still stuck since neither spec-to-spec nor comp-to-spec movement saves the structure; there is no specifier to move and when we do comp-to-spec movement, there is no entry that can match ASPECTP. This leads to the activation of the complex left branch in (12a). At this step, formation of the prefix involves merging the feature in the previous cycle and the one next in the functional sequence; in this case ASPECT and T respectfully. Once we merge these

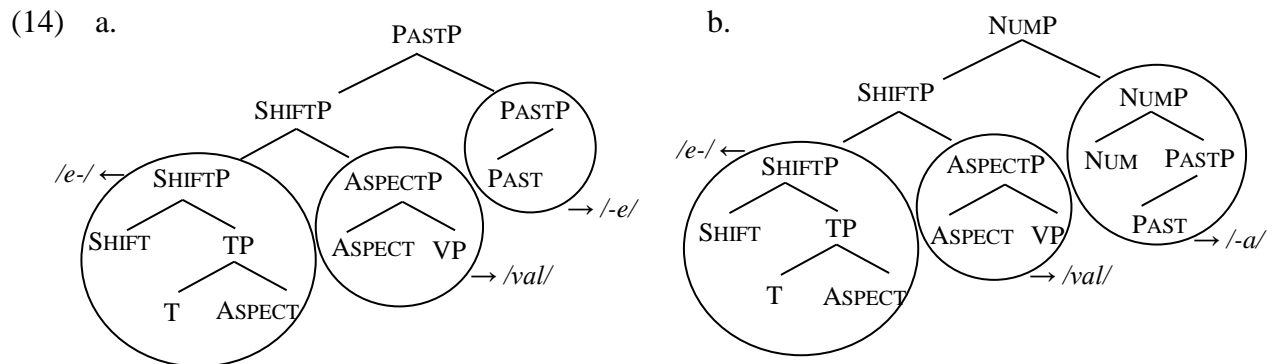
two features, we close the workspace due to Early AUX Closure, and continue with the derivation by merging (12a) with (11b). The result is (12b), the prefix *e-* successfully lexicalizing the TP due to Superset Principle and the root *val* matching ASPECTP.



We continue the derivation by adding SHIFT on top of (12b). Since there is no exact match, we once again resort to movements. Following Caha (2019) and assuming that projecting complex left branches do not undergo spec-to-spec movement, we move on to comp-to-spec movement, which gives us (13a). However, there is no exponent that can lexicalize SHIFTP. Therefore, we backtrack. We unmerge SHIFT from the structure and end up with (12b). Doing comp-to-spec movement is reasonable yet we follow Caha (2019) in that the first backtrack operation following the opening a complex left branch triggers the complex left branch to be opened up again. This means that we add SHIFT on top of (12a) and once again close the workspace due to Early AUX closure. Now we merge our newly created prefix and (12b). The result is (13b) where the prefix *e-* successfully lexicalizes SHIFTP since it is an exact match.

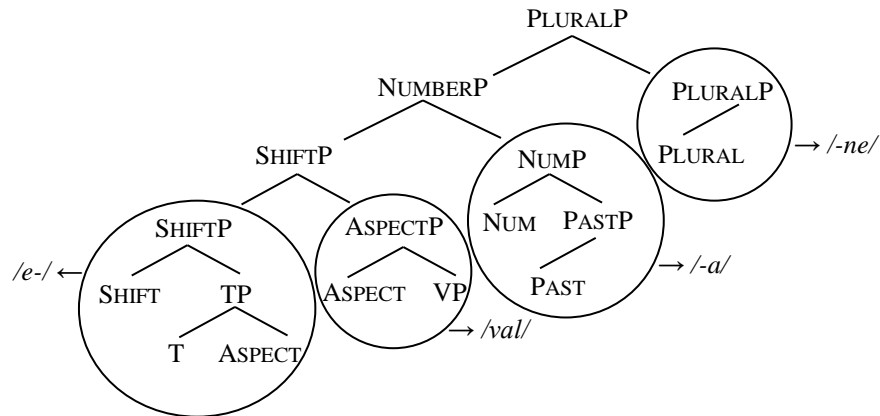


The next step is to merge PAST with (13b). There is no exact match and we cannot do spec-to-spec movement since we once again have a projecting complex left branch. We do comp-to-spec movement and successfully lexicalize PASTP with the exponent *-e*, as shown in (14a). The following feature we merge with (11a) is NUMBER. Since there is no exact match, we do spec-to-spec movement and we successfully lexicalize NUMERP, *-a* overriding *-e* as shown in (14b).



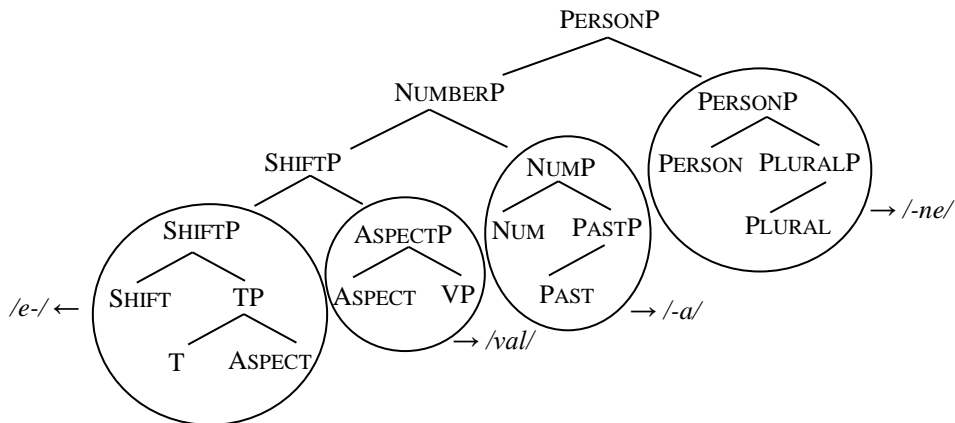
We further merge PLURAL with (14b). Spec-to-spec movement doesn't give us a successful match yet we lexicalize the structure with the exponent *-ne* with regards to Superset Principle as a result of doing comp-to-spec movement as in (15).

(15)



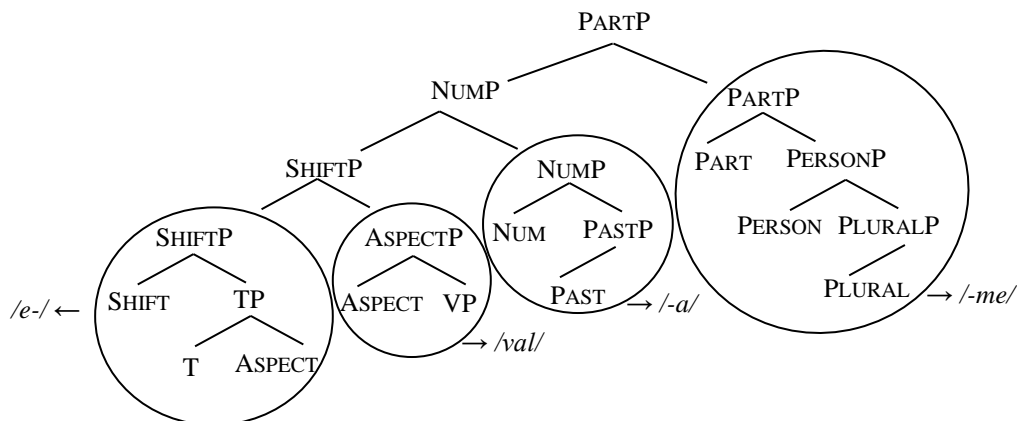
Upon merging PERSON, we resort to spec-to-spec movement and end up with (16) where once again *-ne* lexicalizes the PERSONP due to being an exact match. This is how we build the third-person plural form in the past paradigm, namely *evalane*.

(16)



The next feature we merge is PARTICIPANT. We once again do spec-to-spec movement and we lexicalize the PARTICIPANTP with *-me*, shown in (17). This is how we derive *evalame*, the first-person plural form in the past paradigm.

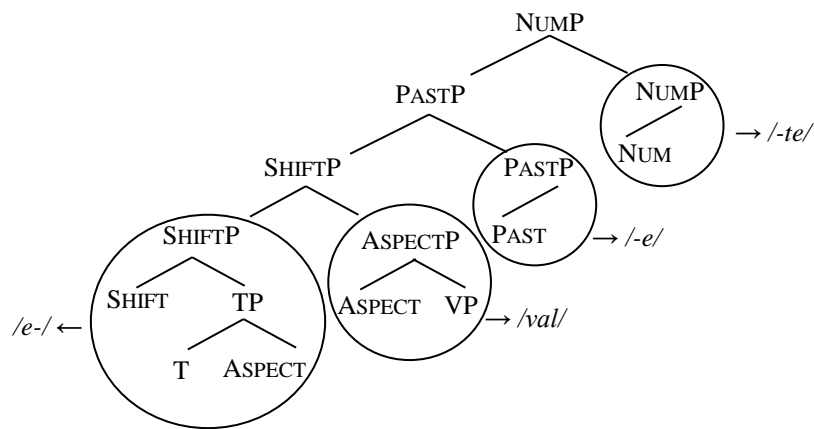
(17)





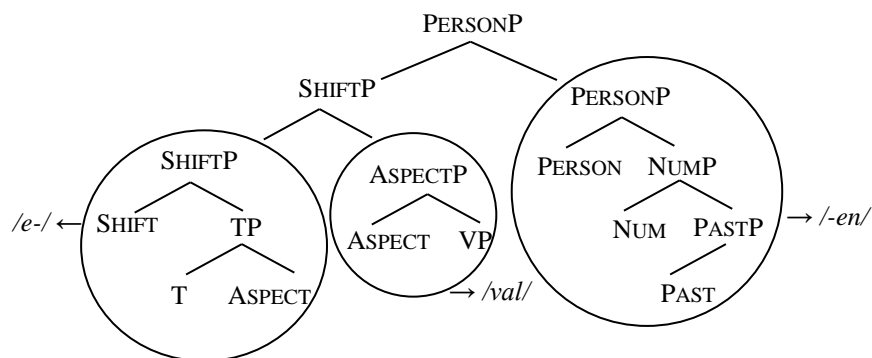
The final feature we merge is ADDRESSEE. Neither spec-to-spec nor comp-to-spec movement manages to lexicalize the structure; hence, we backtrack. We unmerge ADDRESSEE and end up with (17). Since spec-to-spec movement lexicalized (17), we try comp-to-spec movement. However, this does not give us a structure that our exponents can lexicalize. Therefore, we continue backtracking and go to the previous cycle, namely (16). What lexicalized (16) was spec-to-spec movement; therefore, we try the next option for that cycle. However, doing comp-to-spec movement doesn't help us, either. We once again backtrack. In fact, we need to keep backtracking until we reach (14b). The aim is to alter the structure so that the derivation doesn't crash. Since spec-to-spec movement lexicalized (14b), we do comp-to-spec movement instead. This gives us (18). This time we have a match for NUMBERP: *-te* thanks to the Superset Principle. Notice that PASTP is lexicalized by *-e*. In this way, we capture the syncretism between the first and the third person plural forms with the exponent *-a* to the exception of the second-person plural form where we see *-e*. This is how we derive *evalete* in the past paradigm.

(18)



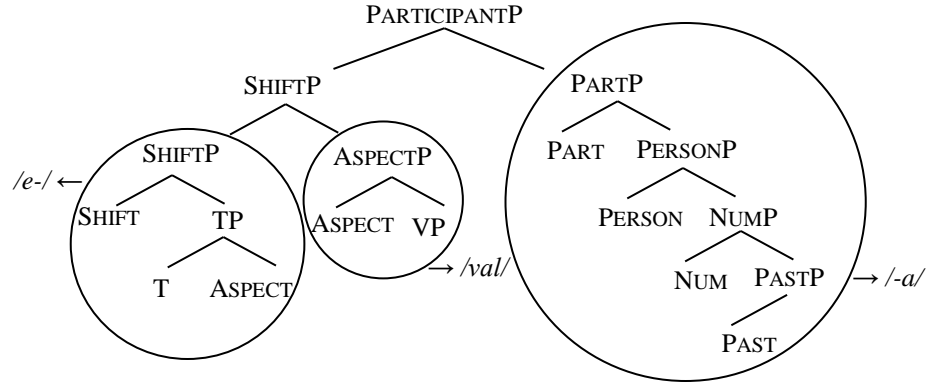
In order to build the singular forms, we follow the same steps until we reach (14b). Instead of merging PLURAL, we merge PERSON with (14b). No match is found; thus, we do spec-to-spec movement. This translates to (19), *-en* overriding *-a* and lexicalizing PERSONP. This is how we derive the third person singular form in the past tense, namely, *evalen*.

(19)



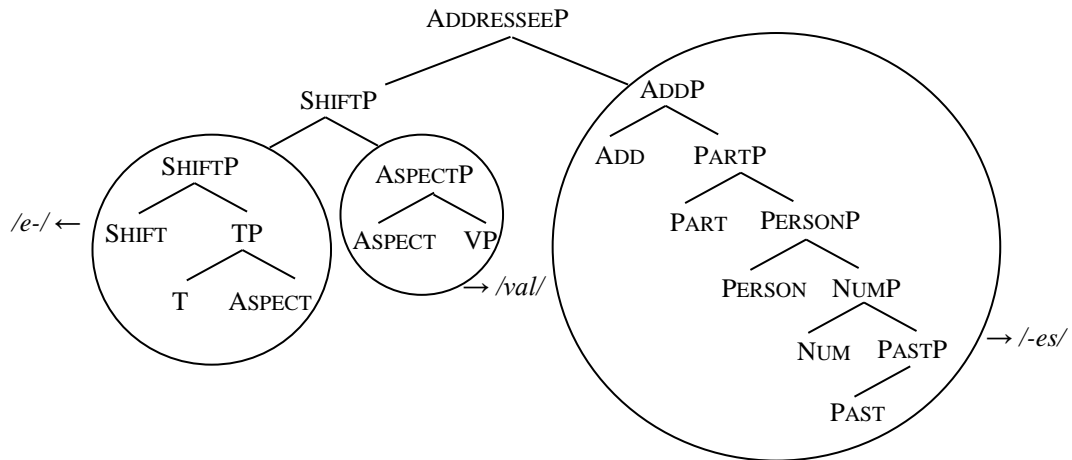
We further merge PARTICIPANT with (19). Spec-to-spec movement once again applies and we lexicalize PARTICIPANTP, *-a* overriding *-en*. In this way, we build the first-person singular form (20) in present tense; *evala*.

(20)



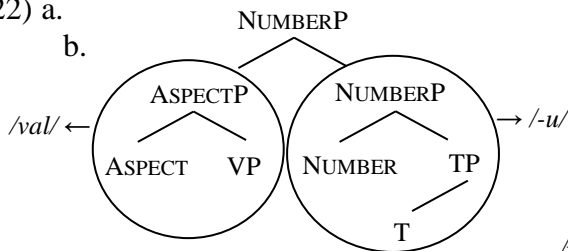
Finally, merging ADDRESSEE with (21) and doing spec-to-spec movement, we derive the second person singular form *evales*.

(21)

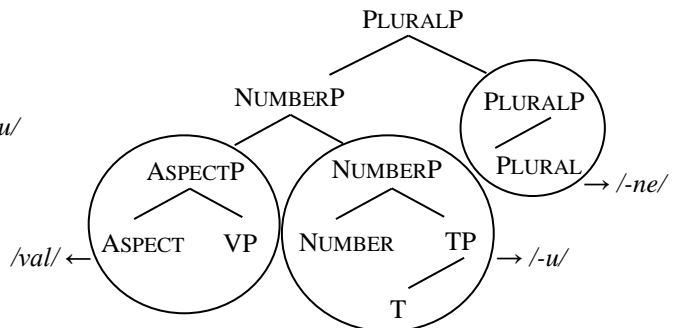


The derivation of the present tense forms mostly involves the same steps to a great extent; as mentioned, with the exception of SHIFT and PAST features in the functional sequence. Therefore, we follow the same path until we reach (10b). Next, we merge NUMBER with (10b). Since there is no match, we do spec-to-spec movement. This gives us (22a), *-u* overriding *-i* and lexicalizing NUMBERP. The next feature in line is PLURAL. Merging PLURAL on top of (22a) and finding no match, we are once again forced to do movements. Spec-to-spec movement doesn't give us a structure we can lexicalize; therefore, we do comp-to-spec movement and insert *-ne* to PLURALP as in (22b) thanks to the Superset Principle.

(22) a.

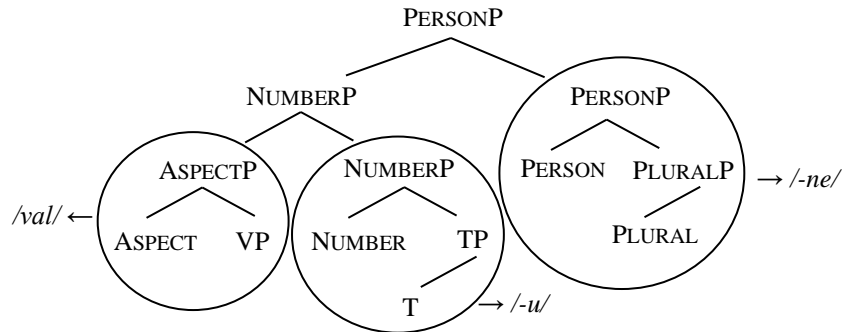


b.



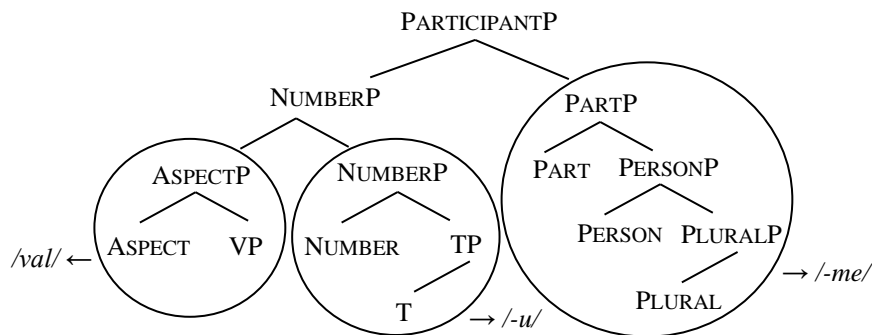
Upon merging PERSON with (22b), we once again do spec-to-spec movement. *-ne* is inserted to PERSONP. We build the third-person plural form *valune* in the present tense paradigm, as in (23).

(23)



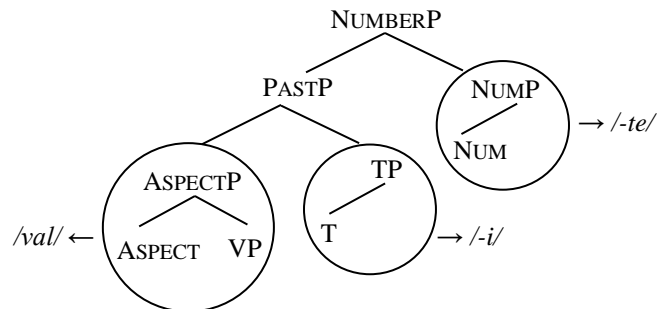
We continue the derivation by merging PARTICIPANT with (23). Spec-to-spec movement gives us (24) where *-me* lexicalizes the PARTICIPANTP. This is the first-person plural form *valume*.

(24)



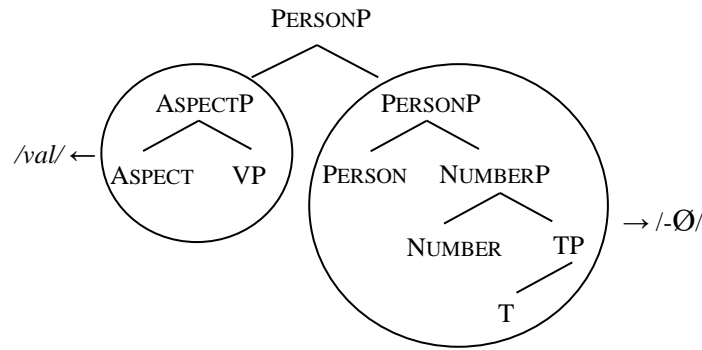
The last feature we merge is ADDRESSEE. We encounter the same problem we came across when we were building the plural past forms; there is no match and neither spec-to-spec nor comp-to-spec movement lexicalizes the structure. We once again backtrack until we reach NUMBERP, namely (22a). Since spec-to-spec movement lexicalized (22a), we do comp-to-spec movement and successfully insert *-te* to NUMBERP, shown in (25). Once again, notice that TP is lexicalized by *-i*. Similar to the derivation of the past forms, we derive the syncretism between the first and the third person plural forms with the exponent *-u*, excluding the second-person plural form with *-i*. This is how we derive the second person plural form *valite*.

(25)



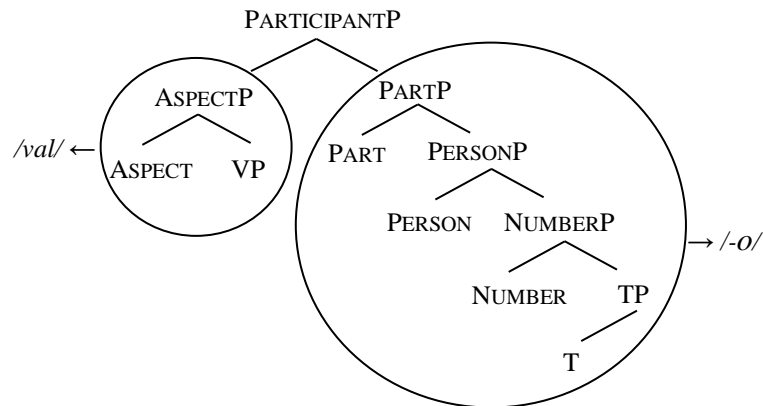
Building the singular forms of the present tense forms involves the same steps until (22a). Similar to the past forms, we merge PERSON with (22a) instead of PLURAL. This triggers spec-to-spec movement to take place and as a result, we insert  $\emptyset$  to PERSONP and eventually derive the third person singular form in present tense, *val*, as seen in (26).

(26)



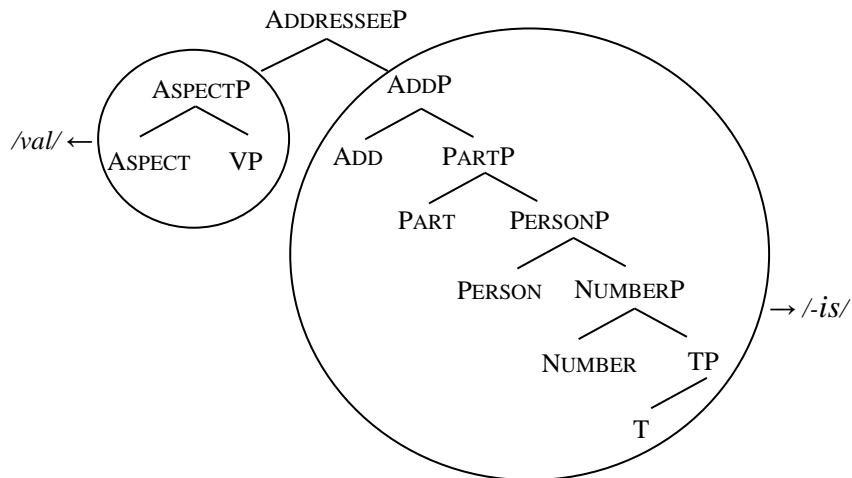
The next step is to merge PARTICIPANT. We do spec-to-spec movement again and insert *-o* to PARTICIPANTP in (27). This is how we derive *valo*, the first-person singular form in present tense.

(27)



The last feature we merge is ADDRESSEE. Spec-to-spec movement gives us (28) where *-is* lexicalizes ADDRESSEEP. In this way, we derive *valis*, the second person singular form in present tense.

(28)



**5. Conclusion.** In this work, we tackle the ABA puzzle in Romeyka by proposing an alternative decomposition for person features. By providing a Nanosyntactic derivation of present and past forms, we show that the illusionary ABA resolves into an AAB pattern. It is worth nothing that

the data can be approached by adopting different alternatives; one such option being phonological and involving the decomposition of the vowels in the paradigms following Element Theory as in the second person forms *evalate* being *eval+A+Ite* and *valite* being *val+U+Ite*. Assuming that the past forms all involve *A* and present forms *U*, this suggestion also stays as an alternative that leads to a regular paradigm and a non-ABA analysis (Caha, personal communication, June 10, 2023). Nevertheless, the proposal in this work justifies the status of first-third syncretism as a metasyncretism in the language in the end. When we take a look at the bigger picture, this gives credibility to the idea that languages differ in foregrounding either the ADDRESSEE or the SPEAKER features. Although variation in the universal functional sequence of features may not always be desirable, we argue that the proposal here is relatively innocent. This is so because the status of the flexibility here does not involve reverse hierarchy but rather simply provides alternative decompositions for first and second persons that crucially do not contradict each other. Needless to say; however, further research is needed to see whether there are any other differences between languages that exhibit one feature rather than the other.

## References

- Andersson, Samuel. 2018. (\*)ABA in Germanic verbs, *Glossa: A Journal of General Linguistics* 3(1): 119. <https://doi.org/10.5334/gjgl.733>.
- Blansitt, Edward. 1988. Datives and allatives. In Michael Hammond, Edith A. Moravcsik & Jessica Wirth (eds.), *Studies in Syntactic Typology* 17, 173–191. Amsterdam/Philadelphia: John Benjamins.
- Bobaljik, Jonathan David. 2012. *Universals in Comparative Morphology: Suppletion, Superlatives, and the Structure of Words*. Cambridge, MA: MIT Press.
- Bobaljik, Jonathan David & Uri Sauerland. 2018. ABA and the combinatorics of morphological features. *Glossa* 3(1). <http://doi.org/10.5334/gjgl.345>.
- Caha, Pavel. 2009. *The Nanosyntax of Case*. Tromsø: University of Tromsø dissertation.
- Caha, Pavel. 2019. Case competition in Nanosyntax: A study of numerals in Ossetic and Russian. Ms. Masaryk University
- Gök, Serra. 2023. *A Nanosyntactic Account of the Turkish Copula and Agreement Paradigms*. Master's thesis, Boğaziçi University.
- Graf, Thomas. 2017. Graph transductions and typological gaps in morphological paradigms. Presented at *the 15th Meeting on the Mathematics of Language*. Queen Mary University of London. <http://doi.org/10.18653/v1/W17-3411>.
- Harley, Heidi & Elizabeth Ritter. 2002. Person and number in pronouns: A feature-geometric analysis, *Language*, 78 (3), 482-526.
- Kramer, Ruth. 2016. Syncretism in paradigm function morphology and distributed morphology. In Daniel Siddiqi & Heidi Harley (eds.), *Morphological Metatheory*, 95–120. Amsterdam: John Benjamins.
- Lander, Eric & Liliane Haegeman. 2016. The nanosyntax of spatial deixis. *Studia Linguistica*. 72(1), 362–427. <http://doi.org/10.1111/stul.12061>.
- Middleton, Hannah Jane. 2020. *\*ABA Syncretism Patterns in Pronominal Morphology*. London: University College London dissertation.
- Starke, Michal. 2009. Nanosyntax: A short primer to a new approach to language. *Nordlyd*, 36(1). 1–6. <http://doi.org/10.7557/12.213>.
- Starke, Michal. 2013. Nanosyntax, part II. Lecture series at CRISSP, Brussels.

- Starke, Michal. 2018. Complex left branches, spellout, and prefixes. *Exploring Nanosyntax* 239–249. <http://dx.doi.org/10.1093/oso/9780190876746.001.0001>.
- Vanden Wyngaerd, Guido. 2016. The feature structure of pronouns: A probe into multidimensional paradigms In: KU Leuven, CRISSP. Retrieved from <http://ling.auf.net/lingbuzz/003166>.