Nanosyntax: some key features

Pavel Caha*
Masarykova univerzita Brno

June 19, 2020

1 Introduction

Nanosyntax (Starke 2002, 2009; Caha 2009) is a theory of morphosyntax whose central tenets overlap to a certain extent with those proposed in Distributed Morphology (DM, Halle and Marantz 1993, 1994). For instance, DM's famous dictum of 'syntax all the way down' is something that Nanosyntax would subscribe to just as much as DM. Similarly, both theories converge on a 'late insertion' approach to morphology, where syntactic computation, consisting minimally of Merge and Move, applies before spellout. The shared features are given in (1).[1]

(1) Nanosyntax and DM, shared features:

a. *Late Insertion*: All syntactic nodes systematically lack all phonological features. The phonological features are supplied – after the syntax – by consulting lexical entries in the postsyntactic lexicon.

b. *Syntax all the way down*: Terminal nodes are organized into hierarchical structures determined by the principles and operations of the syntax. Since terminal nodes correspond to units smaller than words, it follows that syntactic principles govern the internal structure of words. There is no sharp boundary between the traditional syntax and morphology.

At the same time, there are also differences. The first major difference is that in Nanosyntax, each morphosyntactic feature corresponds to its own syntactic terminal (cf. Kayne 2005, Cinque and Rizzi 2010). I will refer to this as "No Bundling." The second important difference is that there are no post-syntactic operations in Nanosyntax ("No Morphology,"

*My work on this chapter has been supported by the Czech Science Foundation (GAČR), grant no. GA17-10144S. I thank Karen De Clercq, Antonio Fábregas, Michal Starke and Guido Vanden Wyngaerd for their helpful comments on a previous version of this paper.

[1] I have used Halle and Marantz (1994) as the basis of (1). I have changed the formulations slightly in an attempt to show what both frameworks share, compared to other approaches, such as, e.g., A-morphous Morphology (Anderson 1992) or Lexicalist approaches (Di Sciullo and Williams 1987).

see, e.g., Koopman 2005, Kayne 2010). In this article, I shall begin from No Bundling, and explain the idea behind No Morphology later on.

Let me start by noting that as a consequence of No Bundling, it follows that any complex object like, e.g., the set of features [1st PLURAL], must be created by (binary) Merge, and therefore, correspond to a syntactic phrase. This position reflects a more fundamental hypothesis about language, namely that any complex grouping of more primitive building blocks is the product of a single generative system (namely syntax).

DM does not share this view. Ever since its early days until the most recent incarnations, the input to syntactic computation in DM are not only terminals with single features, but also complex sets of features, called feature bundles. As Bobaljik (2017) puts it in his recent overview of the theory, the input to the syntactic computation in DM is "a list of the syntactic atoms [...]. Items on this list would include [...] (possibly language-particular) bundles of features that constitute a single node: for example English (plausibly) groups both tense and agreement (person and number) under a single INFL node in the syntax." The reliance on 'syntactically atomic' feature bundles is characteristic for the work done in DM despite the fact that a lot of its home-grown research has been devoted to uncovering the rich structure of these bundles (cf. Harley and Ritter 2002b). In fact, this type of work lays bare a fundamental tension inside the DM model: on the one hand, it is becoming increasingly clear that feature bundles have a rich internal structure. On the other hand, this structure cannot be generated by syntax, because it is already in place when syntax starts assembling such bundles together.

Feature bundles are a point of concern for Nanosyntax. As Starke (2014a) puts it, "a 'feature bundle' is equivalent to a constituent," because "enclosing elements inside square brackets is a notational variant of linking those elements under a single mother node." From this perspective, feature bundles are equivalent to n-ary trees, with n typically greater than 2. Starke further points out that this is equivalent to having a "second syntax" with "a new type of Merge for the purpose of lexical storage." The issue of what generative system lies behind the formation of these bundles remains a largely unaddressed question within DM.

Nanosyntax, as already mentioned, entirely dispenses with feature bundles. The framework (as a core hypothesis about the architecture of grammar) simply rejects the possibility that feature bundles may be generated outside of the core syntactic computation:

(2) *No Bundling (a property of Nanosyntax, but not DM):*
The atoms (terminal nodes) of syntactic trees are single features. All combinations of morphosyntactic features arise as the result of (binary) Merge. Pre-syntactic feature bundles do not exist, they correspond to phrases assembled by syntax.

The elimination of feature bundles eradicates the residue of Lexicalism in the theory of grammar. To see that, consider the fact that feature bundles are language specific (recall the quote from Bobaljik's 2017 introduction to DM). If that is so, then the list of "pre-syntactic building blocks" from which syntactic structures are constructed in DM is also necessarily language specific, as used to be the case in Lexicalist theories. DM of course differs from Lexicalist theories in many important respects, but the idea that syntax draws its atoms from a language-particular list of objects is shared between DM and Lexicalism.

In Nanosyntax, this residue of a language-particular presyntactic lexicon is eliminated. What we are left with as the building blocks are universal features. Concerning the inventory of such features, Nanosyntax in line with Cartography adopts "the strongest position one could take; one which implies that if some language provides evidence for the existence of a particular functional head (and projection), then that head (and projection) must be present in every other language" (Cinque and Rizzi 2010:55). As a result, the set of syntactic building blocks is the same in all languages, and we have no trace left of a language particular list that feeds syntax. Nanosyntax thus completes the shift from a model with a presyntactic lexicon to a postsyntacic one.

As a consequence, the postsyntactic lexicon assumes within Nanosyntax a crucial role as the only source of cross-linguistic variation. The goal of this chapter is to elaborate on the technical consequences of the theoretical choices highlighted above.

2 Constituent spellout

Once feature bundles are dispensed with, it follows that markers like *we*, which correspond to multiple features ([1 PL]), spell out multiple terminals of the syntactic tree. The first question is how to delimit the sets of terminals that may be spelled out by a single marker. This question arises because it is not the case that just about any two terminals may be pronounced together regardless of their position in the tree. To give an example: a morpheme like *we* cannot lexicalize the person of the subject and the number of a fronted XP, so that a sentence like *In god* ***we*** *trust* would be the spellout of a meaning corresponding to *In god****[-s I]*** *trust*. Here the bracket in the second sentence encloses two elements that jointly provide both the feature of the first person and that of a plural, yet their adjacency is not sufficient for joint spellout. So clearly, *some* restrictions on which features may be lexicalized together must be a part of the spellout mechanism: the sheer reduction of the number of morphemes in a string cannot be the right criterion.

Starke (2002, 2009, 2014b, 2018) as well as much current work in Nanosyntax proposes that the theoretically simplest way to define sets of terminals eligible for joint spellout is to rely on sets that are already needed for independent reasons. The sets of terminals that syntax provides for free are constituents, and hence, the zero theory of non-terminal spellout is one where morphemes spell out constituents (cf. McCawley 1968, Weerman and Evers-Vermeul 2002, Neeleman and Szendrői 2007, Radkevich 2010 for similar approaches outside of Nanosyntax).

Constructing theories along these lines has been the golden standard of work in generative grammar. Consider, for instance, the work done on ellipsis or movement. Here, we also encounter situations where ellipsis/movement targets multiple terminals at once. The standard way of explaining why several terminals undergo ellipsis/movement jointly is to say that they are all contained in a single constituent, and it is this constituent that actually undergoes ellipsis/movement. Nanosyntax adopts the same methodology (just applying it to spellout) and adheres to the view that when multiple terminals are joined inside a single morpheme, this is so because the morpheme spells out a constituent containing these terminals. All observable restrictions on joint lexicalisation should follow from this (in the same way as restrictions on 'joint movement' or 'joint ellipsis').

In following this logic, Nanosyntax not only adheres to a standard theory-building procedure, but also increases our model of grammar incrementally (rather than changing it completely). The reason is that in both theories, insertion targets syntactic nodes. As a consequence, where DM has complex feature bundles located under a terminal (subject to insertion), Nanosyntax has a run-of-the-mill syntactic phrase (subject to insertion). But crucially, all syntactic nodes of standard DM are still syntactic nodes in Nanosyntax – just phrasal.

This is not so in sequence-based approaches to spellout (including the so-called Spanning). These approaches are close to Nanosyntax in that a single morpheme may correspond to several terminals. The difference is that in this theory, spellout targets multiple terminals that form "a functional/linear sequence" (Williams 2003, Abels and Muriungi 2008, Ramchand 2008, Taraldsen 2010, Dékány 2012, Svenonius 2012, Julien 2015, Merchant 2015, Haugen and Siddiqi 2016, Blix 2020). These approaches lead to a much more radical departure from feature constituency used in DM. In such approaches, the feature bundles of DM (targeted by insertion) are no longer constituents at all: they have been replaced by a new type of object, a "sequence."

2.1 Underspecification vs. Overspecification

Let me now move on to the observation that if we want to have a theory where morphemes target phrasal nodes, we must define our insertion principles differently from what is usually assumed in DM. The reason is that non-terminal insertion clashes with one of DM's key features (as defined in Halle and Marantz 1994), namely *Underspecification*, see (3).

(3) *Underspecification (a feature specific to DM)*. In order for a Vocabulary Item to be inserted in a terminal node, the identifying features of the Vocabulary Item must be a subset of the features at the terminal node.

Underspecification is embodied in the well-known insertion principle used in DM, the *Subset Principle*, given in an abbreviated form below. Notice that this principle explicitly states that it governs insertion only at terminal nodes.

(4) *The Subset Principle* (abbreviated, Halle 1997)
The phonological exponent of a Vocabulary Item is inserted into a morpheme of the

terminal string if the item matches all or only a subset of the grammatical features specified in the terminal morpheme.

Can this insertion principle govern insertion at phrasal nodes? It turns out that it cannot. Consider, for instance, the suppletive comparative *worse*. Bobaljik (2012:14-5) suggests that it spells out a non-terminal composed of the root $\sqrt{\text{BAD}}$ and an associated CMPR head. Its lexical entry is as shown in (5):

(5) CMPRP $\Leftrightarrow$ /worse/

CMPR $\sqrt{\text{BAD}}$

To allow for the insertion of such lexical items, one could simply drop the restriction on terminal nodes from the Subset Principle. This would preserve the spirit of *Underspecification* and, at the same time, allow insertion into all nodes in general. To reflect the more general nature of insertion sites in such a reformulation of the Subset Principle, given in (6), I will call it the 'Generalized' Subset Principle. The boldfaced parts highlight the modifications introduced in (6) compared to the standard formulation in (4).

(6) *The Generalized Subset Principle* (a made-up principle that would fail, if proposed)
The phonological exponent of a Vocabulary Item is inserted into a **node** if the item matches all or only a subset of the grammatical features specified in the **node**.

This principle would allow the insertion of *worse* (5) into a non-terminal built by syntax, as shown in (7). The tree here depicts the structure built by syntax, and the circle indicates that the spellout of CMPRP is successful, since *worse* in (5) matches a (non-proper) subset of the features specified inside the CMPRP, namely the root $\sqrt{\text{BAD}}$ and the CMPR feature.

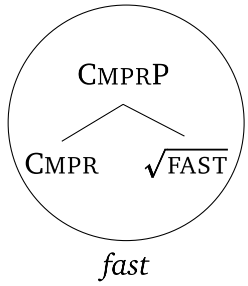
(7) CMPRP

CMPR $\sqrt{\text{BAD}}$

worse

Such a system would lead to a number of problematic scenarios. As an example, consider the regular (non-suppletive) root *fast*, with an entry as given in (8a). The Generalized Subset Principle would allow that such a root spells out the CMPRP in (8b), because the vocabulary item *fast* matches a (proper) subset of the features specified in the CMPRP node, namely the root $\sqrt{\text{FAST}}$.

(8) a. $\sqrt{\text{FAST}} \Longleftrightarrow$ *fast*

b. 

This is obviously the wrong result, since *fast* does not have a comparative meaning, and so insertion at CMPRP must be blocked in this case. In fact, taking this logic to its extreme, a whole sentence containing the root $\sqrt{\text{FAST}}$ could be spelled out as *fast*. Similarly, any node (e.g., the full sentence) could be spelled out by the plural marker *-s* as long as that node contains the plural feature.

There are various ways in which theories based on *Underspecification* may block *fast* in spelling out the whole CMPRP in (8b). The main strategy is to augment the Subset Principle by additional principles that restrict *Underspecification* at non-terminals. For instance, Bobaljik adopts the Vocabulary Insertion Principle proposed by Radkevich (2010) for the case at hand; cf. Feature Portaging in Newell and Noonan (2018) or negative features in Siddiqi (2006). I shall not discuss these theories in any detail here, since the general point is exactly this: in order to allow for non-terminal spell out, one needs to re-think how insertion works: *Underspecification* on its own is not enough.

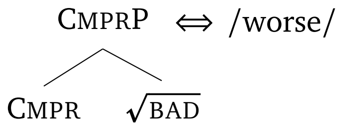
Nanosyntax – rather than proposing additional principles on top of *Underspecification* – replaces *Underspecification* by a similar (but inverted) condition, namely *Overspecification*, see (9).

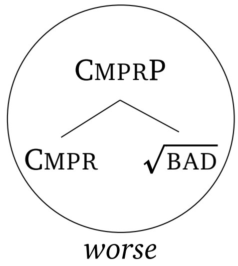
(9) *Overspecification (a feature specific to Nanosyntax)*. In order for a lexical item to be inserted in a node, the lexical entry must fully contain the syntactic node, including all its daughters, granddaughters, etc., all the way down to every single feature dominated by the node to be spelled out, and in exactly the right geometrical shape.

Technically, *Overspecification* is implemented by the so-called Superset Principle:

(10) *The Superset Principle, Starke (2009)*:
A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

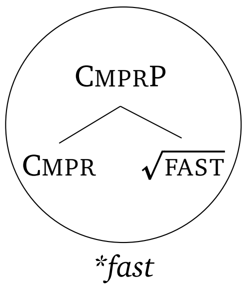
Once the Superset Principle is in place, non-terminal spell out works as needed (and without the need to add principles). Specifically, *worse* (recall (5), repeated in (11)) can still spell out the structure (12), repeated from (7). Spellout is allowed because the syntactic tree (12) is (non-properly) contained in the lexically stored tree in (11).

(11) 

(12) 

However, *fast* (with the entry as in (13a), repeated from (8a)) can no longer spell out (13b) (which is indicted by the asterisk in (13b)). The reason is that the syntactic tree (13b) is not contained in the entry of *fast*.

(13) a. √FAST ⟺ *fast*

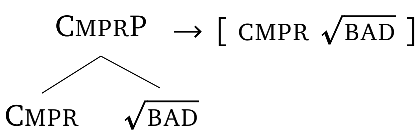
b. 

This is the correct result. We see that once *Overspecification* is adopted, we no longer need to search for any additional principles. As a result, we have a simple rule for insertion (the Superset Principle) that is of exactly the same complexity as the Subset Principle (just inverted); yet it is a rule that applies to all nodes (both phrasal and terminal).

What is important to realize at this point is that abandoning Underspecification is not a matter of choice: it is a virtually forced move in a system where phrasal nodes become targets of insertion.

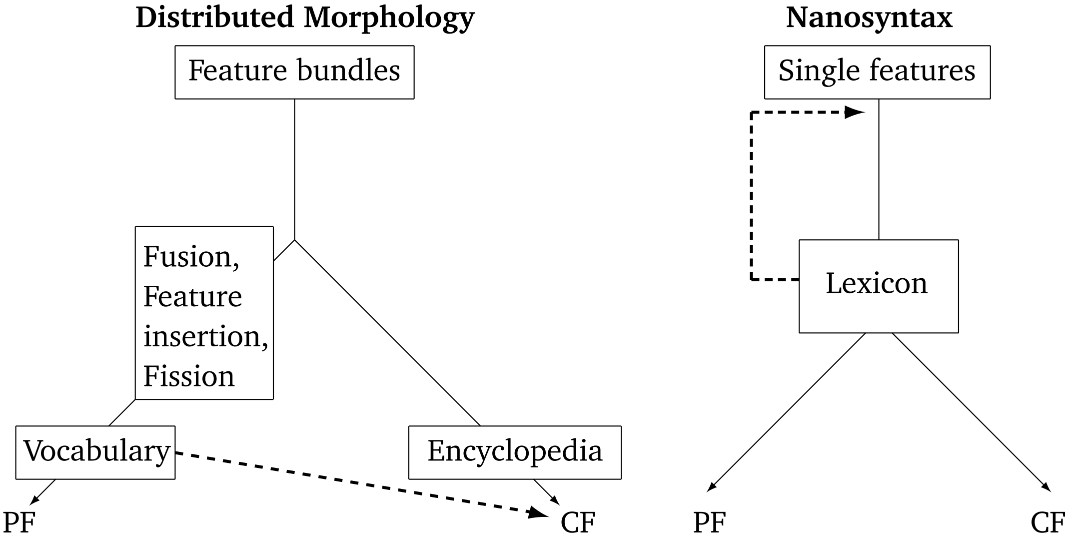
2.2 The architecture of grammar and No Morphology

In "canonical" DM, however, insertion is usually taken to target only terminals (Embick and Marantz 2008, Embick 2016). Trivially, this move eliminates the need to augment *Underspecification* by any additional principles to deal with phrasal spellout. In such case, however, the need to associate *worse* to multiple terminals requires something else. One possible approach within DM is to introduce a post-syntactic operation called Fusion. What Fusion does is that it takes the relevant non-terminal as an input, and turns it into a terminal, see (14). (Bobaljik 2012:147 proposes this as an alternative to phrasal spellout.)

(14) Fusion: 

The availability of such an implementation rests on a particular architecture of grammar. Specifically, because of the fact that Fusion does not seem to be a run-of-the-mill syntactic operation, it is assumed to take place on a separate branch of the derivation, namely a branch that only affects PF. The overall model is thus as shown in (15) on the left-hand side, which is a picture based on Harley and Noyer 1999).

(15) The architecture of Distributed Morphology (left) and Nanosyntax (right)



The special component of grammar where Fusion takes place will be called here the Morphology component (represented by a rectangle on the PF branch, preceding Vocabulary). This component plays a central role in the research done within DM. As Bobaljik (2017) puts it, in DM, "the investigation of mismatches between syntactically-motivated representations, and those observed in the morphophonological string" assumes "a central role," and "a variety of devices [=postsyntactic operations] serve together to constitute a theory of possible mismatches." Some of the operations are listed in the rectangle in (15).

In contrast to this, Nanosyntax proposes to eliminate the Morphology component: the box with morphological operations is missing in the Nanosyntax model, see (15). On this approach, all the 'mismatches' investigated in DM must be accommodated by either updating our theory of lexical insertion (e.g., enrichment by phrasal spellout) or by updating our syntax (e.g., making it more fine grained as a consequence of No Bundling). However reference to post-syntactic operations represents a type of a solution that would not be considered satisfactory in Nanosyntax. In fact, there is no way for such a solution to be stated in this architecture; a point I shall come back to shortly.

(16) *No Morphology (a feature of Nanosyntax).* There is no component of grammar other than syntax that has the power to manipulate syntactic structures. No nodes or features may be added or deleted outside of syntax, no displacement operations take place outside of the syntactic computation. Since there is no Morphology component, there can also be no Morphology-specific features; all features relevant for morphological phenomena are syntactic features by definition.

Note that deciding whether we do – or don't – need a Morphology Component of the sort envisaged in DM is not a matter of simple empirical observation (as researchers working in DM sometimes tend to suggest). This is not to say that facts play no role, but it is an indisputable fact that for as long as the theory of syntax is not ready and finished once and for all (accounting for all the facts there are), mismatches between surface forms and syntactic structures are bound to occur: their very existence does not constitute evidence for anything. The question is rather how we proceed when mismatches are uncovered: do we treat them as illusions, which disappear once syntax and insertion are properly set up, or do we accommodate the facts by adding 'mismatch-removing operations' on top of an existing syntactic theory? Nanosyntax (along with other approaches) rejects the latter and opts for the former. As the chapter unfolds, I would like to give the reader a sense of the technology that Nanosyntax uses to deal with the scenarios that DM covers by the post-syntactic operations given in (15).

Note further that as a consequence of the presence/absence of the Morphology component, a rather fundamental difference emerges between the two frameworks concerning the role of the Lexicon. In Nanosyntax, due to the absence of the Morphology component, the Lexicon appears at the juncture of the three language-related systems: syntax, PF and CF. The position of the Lexicon at this particular place in the grammatical architecture reflects the view that lexical items are nothing but links between well-formed representations belonging to the three modules. Maximally, a lexical item (e.g., *worse*) links a PF representation (/*wɜːs*/), to a CF representation (**bad'**) and to the syntactic representation (the comparative of a gradable adjective = [CMPR A]). When syntax builds the appropriate representation, the Lexicon uses the relevant lexical item to 'translate' it onto its PF representation and the corresponding CF representation.[2]

At the same time, this position of the Lexicon makes any dedicated Morphology component unthinkable; any operation that precedes lexical insertion will be a syntactic operation by definition. Similarly, all features we postulate will be syntactic features.

This is different from DM, which has no unified Lexicon in the same sense as Nanosyntax. This is because the insertion of phonological features must follow the Morphology component, and therefore happens down on the PF branch. This location makes it difficult for a lexicon to feed any concepts to the CF, and so the PF lexicon, called the Vocabulary in DM, only determines the phonological realization of nodes.

Conceptual information in DM is provided by yet another type of lexicon, called the Encyclopedia. The Encyclopedia is a list that is separate from the Vocabulary and links syntactic nodes to concepts. The Encyclopedia is located at the end of the CF branch, symmetrically to the Vocabulary.

Under this setting, a technical issue is how to make sure that the Vocabulary and the Encyclopedia are 'in synch;' i.e., the issue is how to make sure that when the Vocabulary sends /*wɜːs*/ to PF as the spell out of a root node, the Encyclopedia sends the concept

[2]Of course, either the concept or the phonology may be missing from a lexical entry. There may be silent lexical items (so-called Ø morphemes), which lack any phonology. There may also be lexical items that link phonology and syntax, but have no concept associated to them; e.g., the plural marker -*s*.

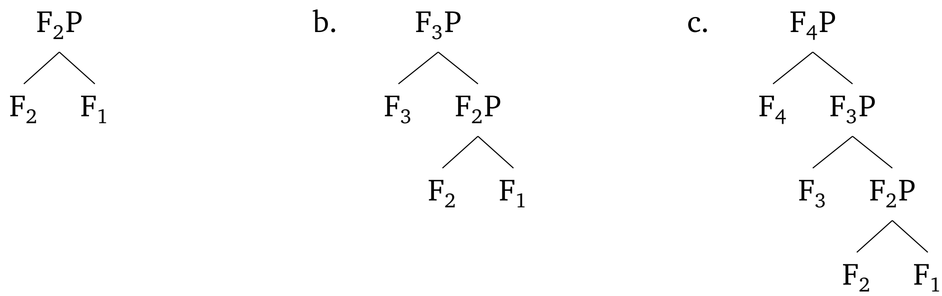
bad' to CF as the interpretation of the same node. One option would be to suggest some kind of a link between the Vocabulary and the Encyclopedia/CF, which is designated by the dashed arrow in (15) (see Harley and Noyer 1999). However, there are also alternative approaches in DM. For instance, Embick and Noyer (2007) propose that items containing CF information (i.e., roots) are inserted early, and carry their phonology as well as meaning throughout the syntactic derivation. Under this conception, the link between the Vocabulary and the Encyclopedia does not need to be assumed (cf. Harley 2014).

For the purpose of this chapter, it is not crucial which of these synching procedures is adopted; the point is simply that while Nanosyntax has a single Lexicon at the juncture of syntax, PF and CF – making any Morphological component unthinkable, DM has a Morphological component and hence, it has two separate post-syntactic lexicons in the sense of two separate 'language-specific lists that interpret the output of syntax.' The presence/absence of a Morphology component and a split/unified lexicon seem to be related choices.

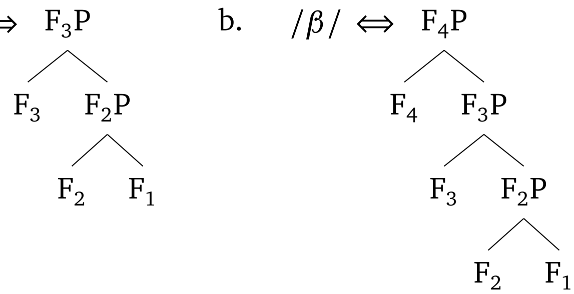
3 Features, Paradigms and the *ABA

In order to see the working of Nanosyntax spellout in more detail, consider the abstract syntactic trees in (17) and the lexical entries in (18).

(17) a. b. c.



(18) a. $/\alpha/ \Leftrightarrow$ b. $/\beta/ \Leftrightarrow$



What we know so far is that the match between a lexical item and the structure is determined by the Superset Principle. According to this principle, the lexical entry for β matches all the structures in (17), because it contains every single one. Still according to this principle, α matches only (17a,b), but it does not match (17c).

This means that for (17a,b), both α and β are candidates for insertion. In such cases, the entries compete, and the so-called Elsewhere Condition (Kiparsky 1973) determines the winner. The Elsewhere Condition says that the more specific entry should always win, where ‘the more specific entry’ is the one that applies in a proper subset of cases compared to the other entry. In our case, this is α, because it applies in a proper subset of structures compared to β. As a rule of thumb, the more specific entry is the one that has fewer superfluous features:

(19) *The Elsewhere Condition*:
When two entries can spell out a given node, the more specific entry wins. The more specific entry is the one which has fewer features.

With the basics of insertion in place, let me now turn to how the theory is put to use in modeling morphological paradigms. As the discussion unfolds, I will first address paradigms that have a single dimension (e.g., case) and I will incorporate two-dimensional paradigms (e.g., case and declension class) as the discussion unfolds.

In a tradition going back at least to Jakobson (1936 [1962]), it has become customary to characterize the cells of paradigms in terms of more primitive units of analysis, namely features, where each cell of the paradigm is defined by a unique set of features. This approach is also widely adopted in DM, where the relevant features usually form a bundle at the relevant terminal. In Nanosyntax, feature bundles are dispensed with, and so each cell in a paradigm corresponds to a constituent containing the relevant features.

Consider, for instance, the trees in (17). These trees can be taken to define a paradigm like the one in (20). Specifically, the tree (17a) contains the features F_1 and F_2; these features can be understood as defining a cell in a paradigm containing those features, see the first row (Cell 1). The tree (17b) corresponds to Cell 2 (it contains the same features as Cell 2), and (17c) corresponds to Cell 3.

(20) An abstract paradigm

	features	α matches	β matches	insertion
Cell 1	F_1, F_2	yes	yes	α
Cell 2	F_1, F_2, F_3	yes	yes	α
Cell 3	F_1, F_2, F_3, F_4	no	yes	β

Consider now how these cells (each cell representing a particular constituent) get spelled out by lexical items. In (18), I have given two lexical entries such that α matches Cell 1 and Cell 2, and β matches the features of all the cells (with Match determined by the Superset Principle). Whether a lexical item matches or not is recorded in the middle two column of Table (20). Where both items match, competition arises with a winner determined by the Elsewhere Condition. The winners are recorded in the final column, which corresponds to the surface paradigm generated by the system.

The interest of “translating” the abstract structures in (17) onto a paradigm like (20) is that once the possibility of such a “translation” is established, we can start doing it also the other way round (i.e., starting from the forms, inferring what the features are and how they are structured). If we succeed, we ultimately reduce surface paradigms (the sequences of

αs and βs) to manifestations of syntactic structures of a rather familiar kind.

A fundamental question in this enterprise is how we come to know—given a set of forms—what features they contain and what structure the features have. An important stepping stone on this path is the investigation of the so-called *ABA patterns. ABA (without the asterisk) refers to a pattern where in a particular arrangement of cells (say Cell 1-2-3), the first cell and the last cell are the same, while the middle cell is different (1=3≠2). When ABA is preceded by an asterisk, this means that such a syncretism is not found, and that all syncretisms target adjacent regions in the sequence 1-2-3.

It can be shown that if the cells in a paradigm are ordered in terms of growing complexity (as in (20)), then it is impossible to generate an ABA pattern. To show that, I will try to set up lexical entries that would actually yield such an ABA pattern. However, as we shall see, this will turn out to be impossible. Because of this, we can conclude that the particular feature decomposition in (20) derives the *ABA restriction.

Let me then try to set up lexical entries so as to produce a β-α-β pattern in (20). In order for Cell 3 in (20) to be spelled out as β, the entry of β must be as in (18b): it must contain all the features of Cell 3, otherwise it would not be an eligible candidate (recall the Superset Principle). Now by virtue of containing all the features of Cell 3, and because we are dealing with a paradigm of growing complexity, β necessarily also contains all the features of Cell 2 and of Cell 1. If that is so, then β also automatically applies in all cells of the paradigm (still by virtue of the Superset Principle).

Now since β is in principle applicable in all the cells, the only way how it may fail to appear in the middle cell is that there is a more specific competitor—α in our case—which outcompetes β due to the fact that it has fewer superfluous features. Further, since this α must both be able to spell out the middle cell and be more specific than β, the only available lexical entry it can have is one where its features exactly match the middle cell; see (18a).

Crucially, once we have β in the most complex cell and α in the middle cell, we realize that the least complex cell (C1) can never be spelled out by the same β that spells out the most complex cell. As a result, the impossibility of an ABA type of pattern is derived.

The conclusion to be drawn here is therefore the following: if it is true that surface paradigms derive from syntactic structures of the sort in (17), we expect such paradigms to exhibit rather stringent restrictions on syncretism. For this reason, the study of *ABA patterns has been an important empirical domain to look at within Nanosyntax.[3]

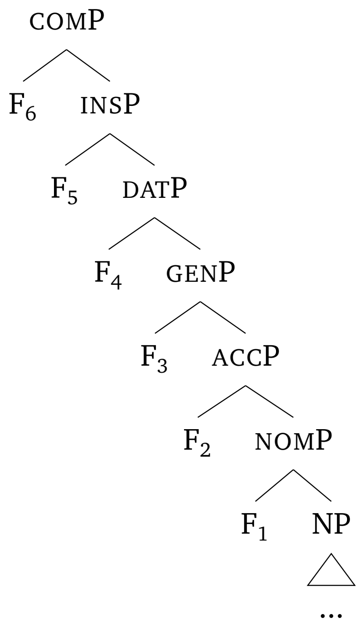
3.1 Case paradigms in Nanosyntax

In one of the first approaches along these lines, Caha (2009) has addressed this issue for case morphology and found a number of individual languages where such linear constraints have been independently proposed. Moreover, Caha suggested that the results of such studies

[3]See, e.g., McCreight and Chvany (1991), Plank (1991) or Johnston (1996) for the investigation of *ABA patterns outside of both DM or Nanosyntax. See Bobaljik (2012) for an important discussion of *ABA within DM, which has provided much inspiration for this kind of work.

can be generalized into a universal linear restriction on syncretism in case, such that in the sequence NOM—ACC—GEN—DAT—INS—COM, only adjacent functions can be syncretic. Leaving the subsequent ramifications of this ordering aside (see Hardarson 2016, Starke 2017, Zompì 2017, Caha 2018, McFadden 2018, Van Baal and Don 2018, Smith et al. 2019), Caha (2009) proposed that such a constraint can be explained by organizing the cells of case paradigms in a cumulative fashion, as depicted abstractly in table (20), so that ultimately, the full case structure looks as given in (21).[4]

(21)



The proposal embodied in this structure is that the nominative case (corresponding to NOMP in (21)) is characterized by the feature F_1, added to the extended NP as a separate head). The accusative (corresponding to ACCP in (21)) is derived by adding to the nominative yet another feature, etc. The proposal has the effect that the cases NOM—ACC—GEN—etc. stand in a containment relation, exactly as the abstract structures in (17). The labels of the phrasal constituents are apparently exocentric, but this is only for clarity: the 'true' label of the nominative is F_1P, but this would be a rather opaque label, so the non-terminal nodes carry the name of the case, which is defined by the collection of Fs it dominates (e.g., ACC = [F_1, F_2]). I am leaving the content of the features aside, see Caha (2013) for some remarks.

The novel part of this proposal is not so much the idea that case decomposes into various features; this has been a common stance ever since Jakobson's (1936 [1962]) pioneering work. The novel part is rather that each case feature corresponds to a head of its own (a consequence of No Bundling), and therefore, that each case (a collection of such features) corresponds to a syntactic constituent. In sum, the proposal reduces the morphological patterns found in case paradigms to a type of architecture that is characteristic for syntactic

[4]In order for the case features to be spelled out by a case suffix, they must form a constituent to the exclusion of the noun. I shall address this in Section 5.2.

derivations.

By now, a number of researchers working within Nanosyntax (and beyond) have looked at various phenomena through a similar lens, and used *ABA patterns as a tool for uncovering the underlying features and their hierarchical organisation into 'nesting' structures of the type in (21) (see in particular Pantcheva 2010, Taraldsen 2010, Wiland 2012, De Clercq 2013, Lander 2015, Baunaz and Lander 2018a,c, Lander and Haegeman 2018, Bergsma 2019, Taraldsen Medová and Wiland 2019, De Clercq 2020). And even though some more recent contributions point out that *ABA patterns can also be derived with non-nesting types of structures (e.g., Caha 2017, Bobaljik and Sauerland 2018, Neeleman and Van de Koot 2020, Truong 2020, Vanden Wyngaerd et al. 2020), this changes nothing to the fact that for a number of domains, the existence of *ABA patterns has led to the confirmation of an architecture where the atoms of syntax are not feature bundles, but single features.

3.2 Case as an M-feature in DM

In DM, the standard treatment of case morphology is different in a way that I think is symptomatic for the larger architectural differences between the frameworks (see Halle 1997, Halle and Vaux 1998, McFadden 2004, Embick and Noyer 2007, Calabrese 2008). Consider, for instance, one specific proposal taken from Embick and Noyer (2007), a state-of-the-art paper on DM. Their feature decomposition, intended to capture the facts of the Latin declension, is given in (22). I have taken the freedom to re-label their ablative as instrumental, since in Latin, the ablative marks also instruments.

(22) Case decomposition in DM

	NOM	ACC	GEN	DAT	INS
Oblique	-	-	+	+	+
Structural	+	+	+	+	-
Superior	+	-	-	+	+

The decomposition offered in (22) differs in two aspects. (i) The features are equipolent (they have a +/- value), and (ii) they form a feature bundle in a single terminal. To see the generative power of such a decomposition, consider, for instance, the triplet NOM—ACC—GEN. In this sequence, no ABA pattern is attested in Latin, which is in line with the fact that this would be very rare cross-linguistically. In particular, Baerman et al. (2005) report that if one of NOM/ACC is the same as an oblique case (frequently a genitive), this is going to be the accusative and not the nominative.

However, the decomposition in (22) cannot rule out such ABA patterns, and so it does not allow us to capture the asymmetry reported by Baerman et al. (2005). Consider the reasoning: In the proposal (22), the three cases under discussion (NOM–ACC–GEN) share the feature [+Structural], and so any exponent marked for [+Structural] can appear in all the cases, yielding an AAA pattern (recall that DM uses *Underspecification*). When such a 'default' AAA pattern interacts with competing entries, ABA patterns emerge. Specifically, if the 'default' entry would compete with an entry tailor-made for ACC ([-Oblique, +Structural, -Superior]), the competition would yield an ABA pattern. Hence, as Caha (2009) argues, the Nanosyntactic proposal that features are privative and assembled by Merge is not only theoretically attractive (consistent with No Bundling), it allows us to capture important generalizations.

3.3 M-features, Fission and the issue of complex feature structures

The standard DM account of case has an additional feature that is worth mentioning in this context. Following Marantz (1991) and McFadden (2004), Embick and Noyer (2007) report that in DM, case features are not a part of the syntactic derivation at all. Rather, "[a]t PF, case features are added to DPs [...], based on the syntactic structure that the DP appears in. [...] These features are added at PF, and are not present in the syntactic derivation." I will refer to such features introduced post syntax as morphological features, 'M-features' for short (cf. Bobaljik 2006 and Alexiadou and Schäfer's chapter on case in this volume).

The important point is that the postulation of M-features amounts to the introduction of yet another generative component (this time post-syntax), where complex feature structures can be assembled. In addition, when case is expressed as a separate affix (as in agglutinative languages), the case features would be introduced in a separate terminal node. This extra terminal is also created for them post-syntax by the special operation of Fission, as Embick and Noyer (2007) make clear in their footnote 25. In effect, both features and nodes may be added to the morphological structure after syntax. I find it difficult to reconcile such an array of structure-building operations with the explicit statement that in DM, "all complex objects, whether words and phrases, are treated as the output of the same generative system (the syntax)" (Embick and Noyer 2007).

One can of course always backtrack from specific proposals about case features, but the fact remains that the multitude of sources for complex feature structures within DM is "an embarrassment of the riches," as Bobaljik (2017) puts it. To make explicit the implications of this situation for the general architecture assumed in Nanosyntax and DM, I will start by quoting a passage from Embick and Noyer (2007), originally meant as a guideline for comparing Lexicalist and non-Lexicalist approaches. They say: "It is often objected in discussions of non-Lexicalist versus Lexicalist analyses that the patterns analyzed syntactically in the former type of approach *could* potentially be stated in a theory with a Lexicon. This point is almost certainly correct, but at the same time never at issue. [...] The Lexicalist position, which posits two distinct generative systems in the grammar, can be supported only to the extent that there is clear evidence that Lexical derivations and syntactic derivations must be distinct."

DM (compared to Nanosyntax) has exactly the same issue of multiple systems that can generate (or minimally provide) complex feature structures: (i) pre-syntactic feature bundles, (ii) syntax (Merge), (iii) nodes and feature bundles constructed 'at PF.' So if the reasoning quoted above is followed consistently, it must be concluded that Nanosyntax has an architectural advantage of exactly the same sort that differentiates between Lexicalist and non-Lexicalist approaches. In particular, to the extent that feature bundles can be generated by syntax (corresponding to vanilla-flavour syntactic constituents), they should not be created at PF or taken simply as a given (i.e., drawn from a pre-syntactic list). The fact that analyzes that do not rely on the existence of multiple sources for complex feature structures "could potentially be stated" in a theory with a whole array of generative devices "should never be at issue."

4 Roots and inflection classes

The prreceding section has summarized one prominent aspect of work in Nanosyntax, which focusses extensively on a single paradigmatic dimension. More recent work addresses another aspect of paradigm structure, which is its vertical dimension (see, e.g., Caha and Pantcheva 2012, De Clercq and Vanden Wyngaerd 2017, Holaj 2018, Vanden Wyngaerd 2018, Caha et al. 2019a, Taraldsen 2019, Janků and Starke 2020, Vanden Wyngaerd et al. 2020, Ziková and Faltýnková in press, Caha to appear). The vertical dimension becomes relevant when, for instance, a particular case (say the dative) is expressed by different markers depending on the gender or the declension class of the root.

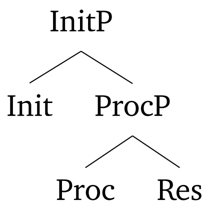
The work in this domain builds on a particular understanding of two partly independent areas of research. The first domain is the analysis of roots: in order to know how a root influences the shape of the suffix, we have to know what the analysis of roots in Nanosyntax is. The second domain we need to understand is how exactly the root and the affix interact with each other, or, more generally, how any two morphemes interact in a phrasal spellout model. I start by presenting the essential features of how roots are analyzed in Nanosyntax.

4.1 Roots and their sizes

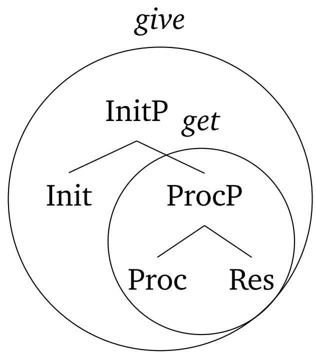
In DM, it is proposed that roots do not have a syntactic category and occupy a special, category-neutral $\surd$ node in syntax. The $\surd$ node is then categorized by special category-forming heads (little *a*, *v*, *n*). The productive nature of the combinations pairing $\surd$s and categorial heads is meant to account for the categorial flexibility of roots (Borer 2005).

While Nanosyntax is in general compatible with $\surd$ nodes (Caha et al. 2019b), a number of Nanosyntactic analyses adopts instead the approach pioneered in Ramchand (2008) and replaces the $\surd$ node by a (sequence of) functional head(s). Specifically, Ramchand proposes that (what used to be) the lexical V head is decomposed into a sequence of three sub-eventive functional heads called the Initiation, Process and Result (Init, Proc, Res). These heads form a hierarchy such that the maximal VP corresponds to (23), leaving aside the arguments which interleave with the relevant heads. In this hierarchy, the heads are linked by a ‘leads to’ relation, such that an (optional) initiation/causation sub-event leads to the process sub-event, which in turn leads to some result being attained (Res is optional).

(23)



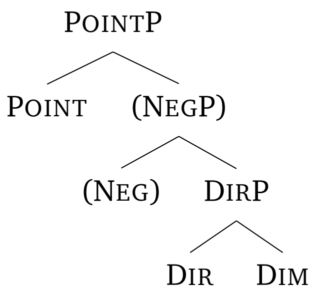
(24)



Individual verbal roots spell out various portions of the maximal decomposed VP, with a specific example in (24). The idea depicted there is that the root *give* spells out all the three heads (Ramchand 2008:108), while *get* lacks *Init* (Taraldsen Medová and Wiland 2019; cf. Richards 2001 on the relationship between *give* and *get*). What is important is the general idea that various roots can be characterized by a different amount of functional heads, yielding different verb classes.

In Vanden Wyngaerd et al. (2020), an analogous proposal is made for gradable adjectives. The starting point of the analysis is that such adjectives contain at their core a scale. Semantically, the scale corresponds to a totally ordered set of points (degrees) pertaining to a particular dimension (see, e.g., Kennedy and McNally 2005). In syntax, as Vanden Wyngaerd et al. (2020) propose, such a scale is decomposed into two syntactic components, DIR and DIM, see (25). The lowermost DIM head encodes the relevant dimension of measurement (length, temperature, speed, etc.), while the ordering of degrees is expressed by the DIR head. The full DIRP thus corresponds to the semantic notion of a scale.

(25)

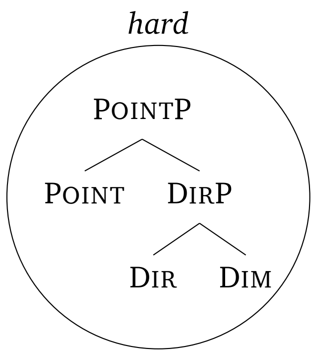


On top of the scalar component is an optional negative head NEG. The NEG feature (when present) reverses the direction of the scale and we get the so-called negative adjectives. The presence/absence of this head thus differentiates pairs of antonymous adjectives containing the same scale, such as *hard—soft, fast—slow, long—short*). While a positive adjective like *wide* would only have a structure with DIM and DIR, a negative adjective like *narrow* would include NEG in addition (cf. De Clercq and Vanden Wyngaerd 2019).

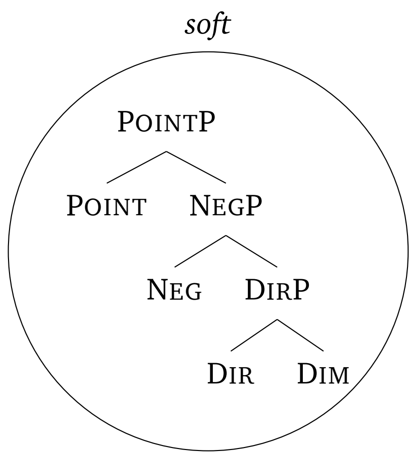
On top of the scale (corresponding either to DIRP or NEGP) comes a POINT projection, which introduces a point on the scale. When the adjective is combined with its argument, the argument of the adjective must have a degree such that it exceeds this point. The point may be specified by a measure phrase, but it may also be left unspecified. When it is unspecified, it is provided by the context and the so-called context-sensitivity of gradable adjectives arises.

In sum, the lexical entry of a positive adjective like *hard* contains the three heads Dim, Dir and Point, which makes it capable to spell out a structure like (26). A negative adjective such as *soft* has a Neg head in addition, and spells out a structure like (27). The crucial idea (as in the case of verbs) is that different adjectival roots spell out different sets of functional heads.

(26)



(27)



When it comes to the categorial flexibility of roots, Nanosyntax handles this on a par with the ambiguity of functional morphemes, namely by *Overspecification*. For instance, Caha et al. (2020) observe that in English, there are (at least) two types of verbs based on adjectives. One type is identical to the adjective (28a), the other has a suffix, see (28b).

(28)	a.	to cool	b.	to hard-**en**
		to dry		to soft-**en**
		to narrow		to wid-**en**

All these verbs have both an inchoative use (29a,c) and a causative sense (29b,d):

(29)	a.	The road narrow-ed	b.	The workers narrow-ed the road
	c.	The road wid-**en**-ed	d.	The workers wid-**en**-ed the road

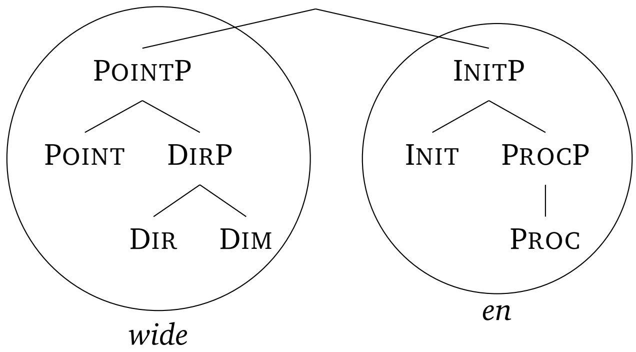
According to Ramchand (2008:108), the causative structures in (29b,d) contain the init and proc heads in addition to the adjectival base. Caha et al. (2020) therefore take the overt suffix in (29d) to realize these two heads, as depicted in (30).[5]

(30) [init [proc]] ⇔ /-en/

[5]Unlike in the case of regular open-class [init proc] verbs, *-en* is not linked to any concept; it is a pure verbalizer without any conceptual meaning.

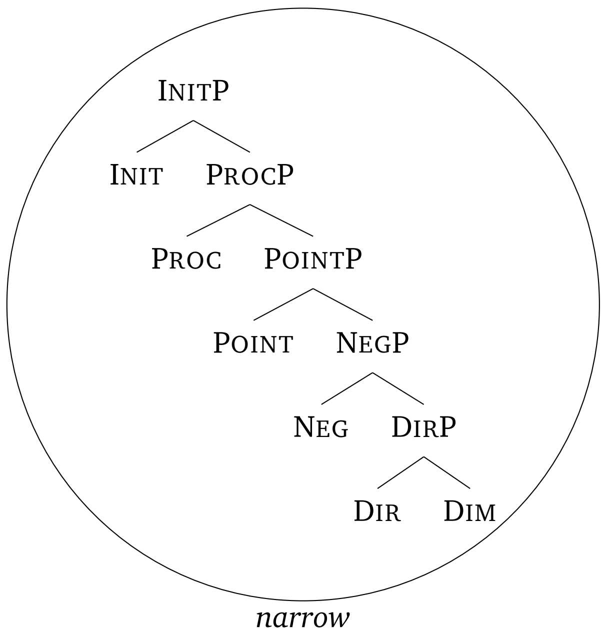
This suffix will combine with adjectives to produce verbs like *wid-en*, see (31). Here the adjective (which corresponds to PointP) has moved across the Init and Proc heads from the complement position of Proc.

(31)

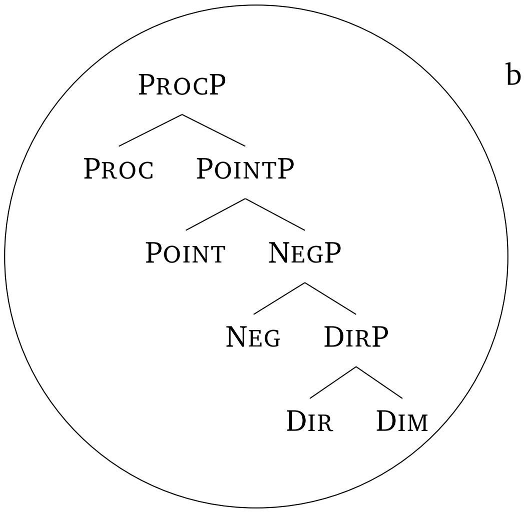


According to Caha et al. (2020), 'zero-derived' verbs (such as *narrow* in (29b)) should be analyzed as a case where the root is capable of realizing all the meaning components, see (32).

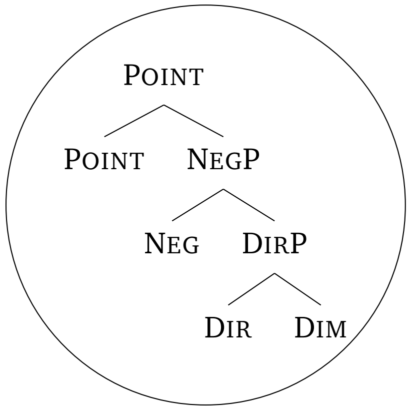
(32)



The ambiguity of such roots between the causative and inchoative verb as well as a positive degree adjective falls out as a consequence of the Superset Principle (10), which allows that a lexical entry can spell out any subtree which it contains. Both the inchoative sense (33a) and the adjective sense (33b) correspond to proper parts of (32). As a result, we derive these readings without the need to postulate special category-neutral nodes in syntax.

(33) a. 

narrow

b. 

narrow

The proposal makes it possible not only to capture the flexibility of some roots (e.g., *narrow* in (33)), but also the restrictions on other roots. Consider, for instance, the fact that the adjective *wide* does not allow a verbal use parallel to *narrow* (**to wide the road* vs. *to narrow the road*). This can be captured if the root *wide* is associated to a tree of the size PointP in the lexicon, which makes it impossible for it to be used as a InitP or ProcP. As a consequence, we predict that *wide* cannot be used as a verb, and must rely on *-en* to spell out Init and Proc, recall (31).

Finally, let me note that in Nanosyntax (just as in DM), the choice of a particular root is not determined by the Elsewhere Condition. In Nanosyntax, this can be seen as a consequence of the fact that roots are associated to a concept. Because of this, it is not the case that one of them is used in a proper subset of environments compared to the other, and the Elsewhere Condition does not apply in such a situation. This entails that the choice between different roots is left to the 'free will;' i.e., we insert the root we 'want to talk about' (provided its lexical tree matches the syntactic tree).

4.2 Allomorphy as a consequence of root size

What we have learned in the previous section is that roots in Nanosyntax are similar to functional morphemes and spell out various sets of functional heads. One of the ideas that has gained prominence in Nanosyntax is that 'root size' can also explain the existence of (partly arbitrary) inflection classes. In fact, differences in root size (which is just a consequence of the arbitrary nature of lexical storage) may very well be all that one needs in order to account for the existence of inflection classes.

To show how root size helps us analyze arbitrary inflection classes, consider the following four paradigms of Russian (34).

(34) Four inflection classes of Russian (Timberlake 2004)

	factory I_A (MASC)	place I_B (NEUT)	lip II (FEM)	notebook, III (FEM)
NOM	zavód-Ø	mést-o	gub-á	tetrád'-Ø
ACC	zavód-Ø	mést-o	gub-ú	tetrád'-Ø
GEN	zavód-a	mést-a	gub-ý	tetrád'-i
LOC	zavód-e	mést-e	gub-é	tetrád'-i
DAT	zavód-u	mést-u	gub-é	tetrád'-i
INS	zavód-om	mést-om	gub-ój	tetrád'-ju

Each paradigm represents a particular declension (the divisions are as in Timberlake 2004). Animate nouns aside, each declension is occupied by nouns of a particular gender. The pairing of genders and declensions for inanimate nouns is as indicated in the header of the table (34), exceptions are very few.

The basic question I shall be addressing in this section revolves around the two feminine classes; these are classes II and III (*gub-a* 'lip' and *tetrád'* 'notebook' respectively). (A full discussion can be found in Caha to appear.) The basic question is how to pair the roots of Declensions II and III with the correct endings in a way that, e.g., the ACC Class II ending *-u* fails to appear in the accusative of a root belonging in Class III. Ideally, as Müller (2004) notes in his DM analysis, "one might hope that information that is inherently present on a stem – like gender, phonological, or semantic features – will suffice. This is not the case, though." The reason for this, as Müller notes, is that "a feminine, inanimate noun stem ending in a soft consonant may belong to class II or class III." The relevant conclusion is, therefore, that class membership cannot be fully reduced to some independent property of the root; it is (at least in part) arbitrary.

In Müller's approach, the above observation translates into a theory where "arbitrary inflection class features must be assumed as inherent properties of noun stems." The specific way in which Müller analyzes inflection classes is shown in (35) (cf. Alexiadou and Müller 2008). What we see here is that each of the classes is defined by two diacritic binary features α and β.

(35) Decomposition of inflection classes in Russian:

a. I_A $[+\alpha, -\beta]$ *zavod* ('factory')

b. I_B $[+\alpha, +\beta]$ *mest* ('place')

c. II $[-\alpha, +\beta]$ *gub* ('lip')

d. III $[-\alpha, -\beta]$ *tetrad'* ('notebook')

The proposal brings out an important point, which is, as Müller puts it, that "[i]nflection class features are arbitrary and irreducible by definition; this is reflected in the [arbitrary] labels $[\alpha, \beta]$." While such labels may not be widely used, Müller finds it "worth emphasizing that the features $[+/-\alpha]$, $[+/-\beta]$ are no more arbitrary than standardly adopted features like [class I], [class II]" (as in, e.g., Halle 1997, Halle and Vaux 1998).

The need to use arbitrary and language-specific features (diacritics) is something that we should be concerned about. If these features are present in the syntax, this entails that we are giving up on the idea that syntactic features are drawn from a set provided by the UG, which (by definition) would have to be language invariant. Moreover, since these features are only relevant for inserting the correct ending, the features are irrelevant for the inner working of the syntactic derivation. From this perspective, the postulation of such features within the core syntactic computation is sub-optimal.

The alternative is that these features are (similarly to case features) introduced after syntax, which (once again) brings architectural challenges, which concern the number of structure-building (generative) components in the grammar.

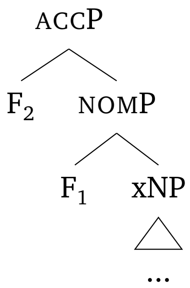
There is also a long tradition of critique directed at declension diacritics in the Word and Paradigm model (see, e.g., Blevins 2016). The basic idea of this model is that inflected forms should not be decomposed into its constituent parts (morphemes) at all, because the word (on this conception) is more than a pure sum of its parts. Equivalently, the parts on their own do not fully suffice to recover all the information that words contain. According to Blevins (2016:71), the need to postulate declension diacritics represents nothing else but a blatant admission of this fact: once words are decomposed, it becomes impossible to rebuild them again using only the information inherent in the pieces. Declension features are artificial devices that are intended to overcome this problem; but according to Blevins, they are nothing but "assembly instructions" in disguise, which are only needed to "restore information about the co-occurrence of stems and exponents" (which has been lost after decomposition).

In a Nanosyntactic analysis (see Caha to appear), it is possible to generate the correct combinations of roots and affixes without declension features, relying on the notion of root size instead. On this approach, the features of the forms are drawn from the universal inventory proposed by Harley and Ritter (2002a). At the bottom of the hierarchy, there is a referentiality feature REF (nouns are referential). Above REF, the gender of the noun is specified. Harley and Ritter use the feature CLASS for this purpose. CLASS is interpreted as non-feminine by default. Feminine nouns require the feature FEM in addition. Above gender, number features are found, Caha uses #P (from Borer 2005) as a countability feature. The feature # is interpreted as singular by default (plural requires an extra PL feature). Above number come the case features.

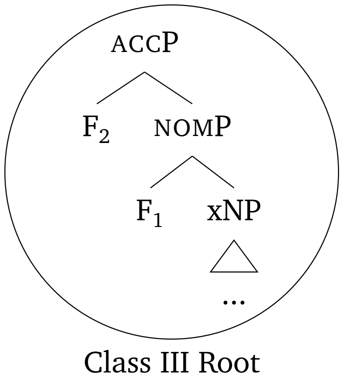
In Caha's analysis, each feature of the feminine declensions is a self-standing head. The heads are structured as shown in (36), the optional PL feature is in brackets.

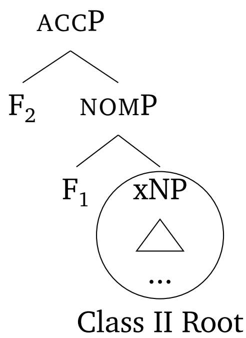
(36) [CASE [(PL) [# [FEM [CLASS REF]]]]]

To explain how declensions II and III arise in a system where all feminine nouns have the same set of features (36), let me first abbreviate all the features below case as xNP, see (37).

(37) 

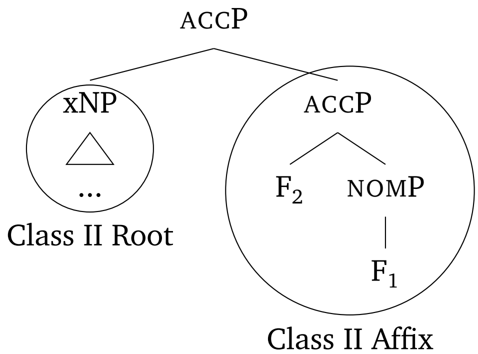
The first observation is that nouns of Declension III have no ending in NOM/ACC. We can capture this by saying that their roots spell out all the features up to the projection of the accusative case, as in (38).

(38) 

(39) 

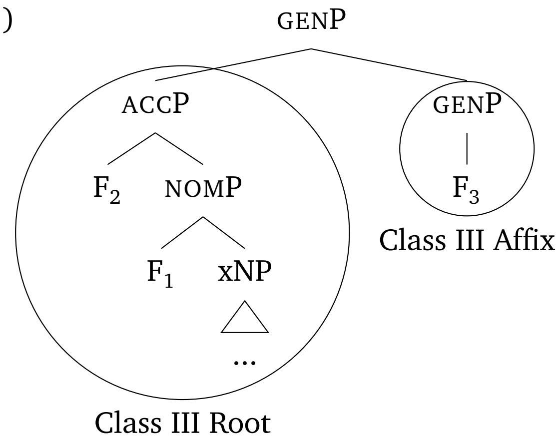
The Declension II root (*gub-* 'lip') has an ending in the nominative singular (e.g., *gub-**a*** 'lip-NOM'). We can capture this if their root is lexically smaller than NOMP and as a consequence, it leaves some features unlexicalized, see (39).

Now in order for the additional features in (39) to undergo spellout, the root must move and the ending appears as the realization of the leftover features. The tree in (40) shows the structure of the accusative singular. In sum, the difference between Class II roots and Class III roots is perfectly analogous to the difference between a root like *narrow* and *wide*.

(40) 

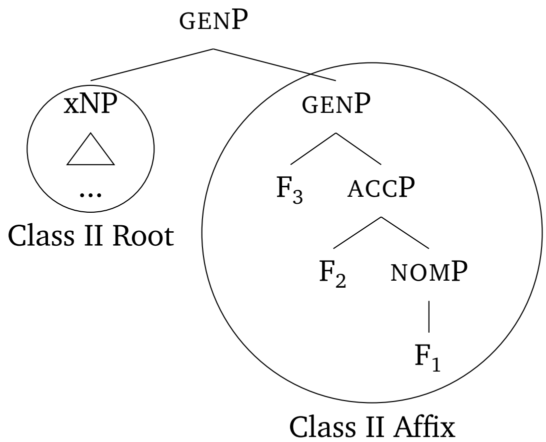
In the oblique cases (which add features on top of the accusative), both declensions have an ending. However, the endings are different. This can be explained as a consequence of the fact that Class III nouns (*tetrad'*) only need F_3 to be spelled out, see (41).

(41)



On the other hand, the nouns of Declension II leave all the case features unlexicalized, see (42). As a result, a difference between the two roots is maintained throughout the paradigm, because the different endings spell out different sets of features.

(42)



The result is that as a function of an arbitrary root size, we get a difference in the suffixes the nouns take. At the same time, we are not forced to postulate any declension diacritics (e.g., [+/- α]). This is a good result, because such diacritics lack any independent justification. As a consequence, the existence of different (arbitrary) declensions is derived fully in line with No Morphology (which, recall, bans any morphology-specific features).

While successful in reaching these goals, the proposal entails that different sizes of roots give rise to different syntactic structures. In (41), the whole ACCP moves from below F_3. In (42), only the xNP moves. In the next (final) section, I explain how such derivations work. The general idea in presented in the following section is that the syntactic derivation is allowed to interact with individual lexical items, which have a say in how the derivation unfolds. This idea is crucial not only for explaining the difference between two Russian declensions, but also for explaining variation among languages: trivially, two different languages will have different lexical items. If lexical items can influence how derivations unfold, different languages will have different structures.

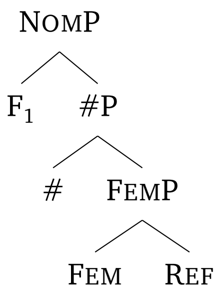
5 Cyclic spellout and spellout driven movement

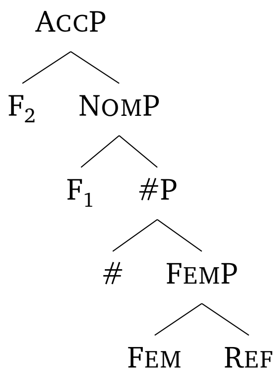
The technical implementation relies on two features of the current work done in Nanosyntax. These are *Cyclic phrasal spellout* and *Spellout-Driven Movement*. I introduce these in turn (cf. Starke 2018, Baunaz and Lander 2018b, Bergsma 2019, Caha et al. 2019a, De Clercq 2019, Wiland 2019, Vanden Wyngaerd et al. 2020, Caha to appear).

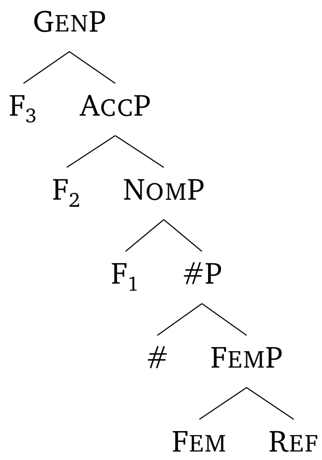
5.1 Cyclic spellout

(43) *Cyclic phrasal spellout*. Spell out must successfully apply to the output of every Merge F operation. Since Merge proceeds bottom-up, so does spellout. After successful spellout, the derivation may terminate, or proceed to another round of Merge F.

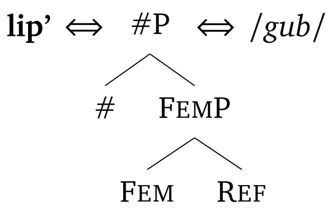
To see how cyclic spellout works, let us assume the very same toy scenario that we have been working with in the previous section. The syntactic structures we are interested in are given in (44), and they represent the nominative, accusative and genitive singular of a feminine noun. The representations are slightly simplified for brevity, the CLASS node (found between FEM and REF) is left out.

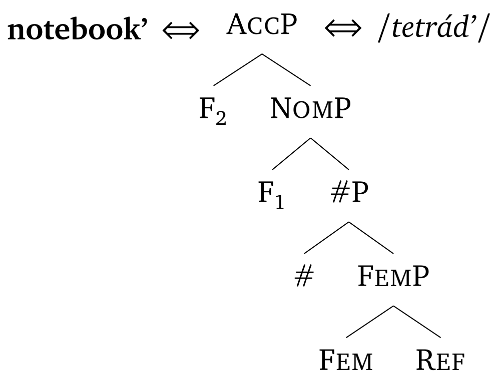
(44)


(45)


(46)


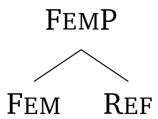
The lexical entries of the roots are in (47) and (48); we have the Class II root *gub-* 'lip,' and the Class III root *tetrád'* 'notebook.'

(47)


(48)


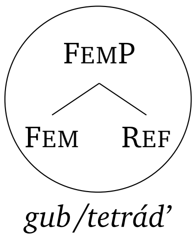
Suppose now that spellout proceeds cyclically, where each cycle begins by Merging a new feature F and ends by finding a matching lexical item for the feature F and its FP projection. This means that syntax first merges Fem to Ref, forming FemP shown in (49).

(49)



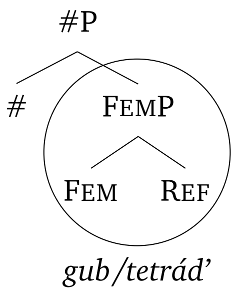
After Merge F has applied, spellout applies. Spellout means that the lexicon is searched for an item matching the phrase in (49). In our toy scenario, FemP is contained in the lexical entry for both *gub*, recall (47), and *tetrád'*, recall (48). Each of these may be inserted, since roots do not compete with each other. Regardless of which root is inserted, spellout is successful (a matching item has been found). This is depicted in (50) by the circle around the relevant constituent.

(50)

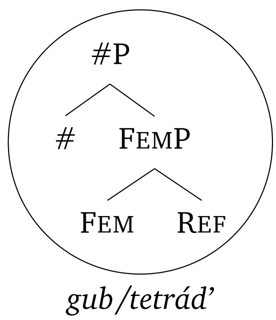


After the successful application of spellout at FemP, the lexicalization procedure remembers minimally that FemP can be lexicalized by the relevant root. If no more features were to be added (we are finished constructing the intended meaning), the derivation would terminate and FemP would ultimately be pronounced by the root. However, if we want to add more meaning, the derivation continues—without being immediately pronounced. Suppose it continues, which means that FemP is fed back to syntax for an additional Merge F operation. The feature # is added, and spellout applies to the #P depicted in (51). In (51), the tree contains the information (accessible to the lexicalization procedure, not to syntax) that FemP has been matched by a particular root at the previous round of spellout.

(51)



(52)

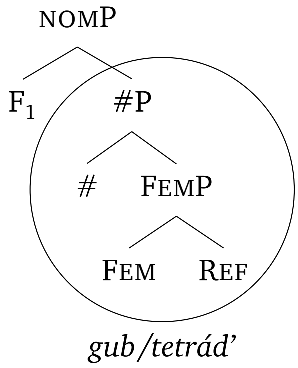


When a tree like (51) is fed to spellout, we again find two possible matches for #P in our toy lexicon, namely *gub* and *tetrád'*. Hence, the structure (51) can be pronounced by either of these roots, see (52). The match at #P is remembered, and all previous matches inside the #P are forgotten (over-ridden). Should no more features be added, #P would be pronounced the same as FEMP, namely by either of the roots.

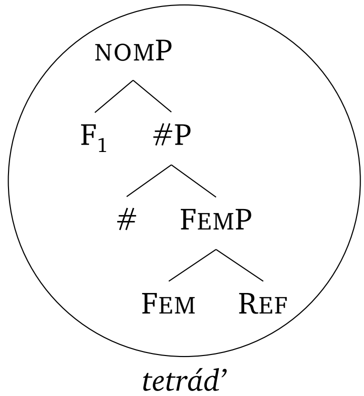
In the abstract Nanosyntax model in (15), the cyclic nature of Merge—Spellout—Merge—Spellout... is depicted by the dashed 'feedback' loop from the lexicon to the start of the syntax.

Let me now come back to the stage of the derivation (52). When the nominative feature F_1 is introduced, as in (53), a new situation arises. That is because only the root *tetrád'* can spell out F_1P, see (54). However, the Declension II root *gub* can't do this, because its entry in (47) does not contain this constituent.

(53)



(54)



In order to see how the derivation of the Declension II proceeds, I must now introduce spellout-driven movements (spellout movements for short).

5.2 Spellout-driven movement

Spellout movements are rescue movements, activated when the spellout of some FP fails. The goal of the rescue movements is to modify the structure in a way that the FP will ultimately be spelled out.

(55) *Spellout-driven movement.* When Merge F produces an FP that cannot be spelled out (no lexical item matches the FP), the FP is rejected at the interface and Merge F (the addition of new features) cannot continue. Syntax then tries to rescue the structure by performing one in a predefined hierarchy of movement operations, before it sends the structure for spellout again. A number of different types of movements may apply until spellout at FP succeeds. Once lexicalization succeeds, the derivation either terminates or continues by Merge F.

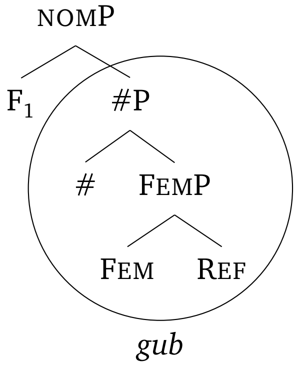
The algorithm that governs spellout movement is given in (56). The algorithm is formulated as a ranking of spellout options: spellout without movement is preferred, and only if this option fails, movements are activated. Among the movements, Spec movement is preferred over complement movement. This particular ranking is what will allow us to derive different structures depending on the size of the root.

(56) Spellout Algorithm (based on Starke 2018)

a. Merge F and spell out.
b. If (a) fails, move the Spec of the complement and spell out.
c. If (b) fails, move the complement of F and spell out.

To see how this works, let us come back to the Declension II noun in the nominative, see (57). What we concluded is that when the nominative feature is merged, yielding (57), the full structure cannot be spelled out by the root *gub*; its lexical entry does not contain the case feature F_1.

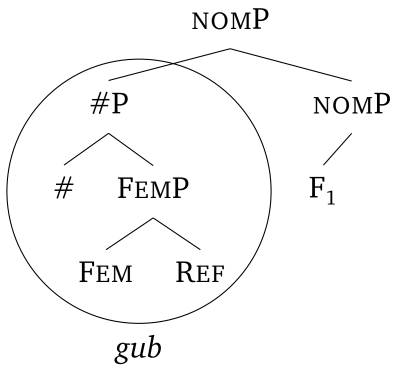
(57)



The structure is therefore rejected at the interface, and Merge F cannot continue. The reason is that the nominative feature is not lexicalized, which is not allowed (cf. Fábregas 2007). A repair movement (driven by the need to spell out) is therefore attempted. The various options for such movements are always applied in the succession given in (56).

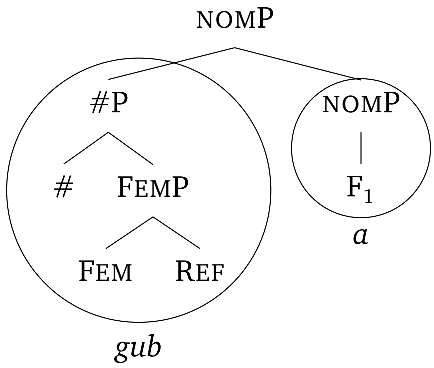
The first option (introduced under (b)) is moving the Spec of the complement. In our specific case, this entails that the movement of Spec,#P should be tried first. However, #P has no Spec in (57), so this option is skipped. The next option down the list is the movement of the full complement, which is the #P. The output of such a movement is shown in (58).

(58)

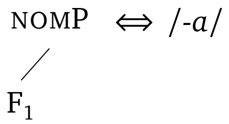


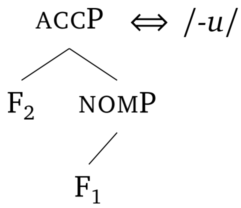
#P relocates to Spec,NOMP, leaving no trace behind. According to Starke (2018), the absence of traces is a general property of spellout movement, and this is also how spellout movement differs from, e.g., *wh*-movement. This is in turn related to the fact that spellout-driven movement (unlike *wh*-movement) never shows any reconstruction effects, and so there is never any evidence for two different interpretive positions; see also Caha (to appear) for a discussion.

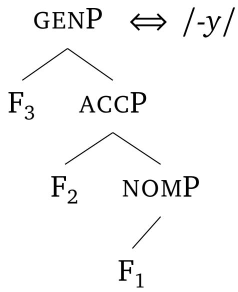
After movement, the spell out of the lower NomP in (58) is tried again. NomP now lacks the #P inside, and so there is a chance that lexicalization succeeds. We know that it does so in Russian, inserting the Nom marker *-a*, see (59).

(59)


The lexical entry of this *-a* must therefore be as shown in (60a). It is easy to see that this entry perfectly matches the NomP in (59). In (61) and (62), I show the accusative and the genitive case markers respectively.

(60)


(61)


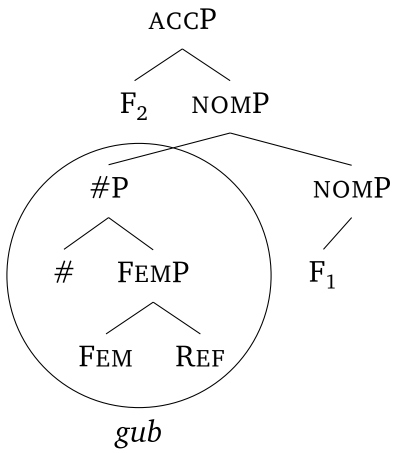
(62)


Note incidentally that unlike roots, case markers compete for insertion, because they are not associated to any concept. Specifically, the case markers in (61) and (62) are also candidates to spell out the NomP in (59); however, they lose in competition with *-a* (they have superfluous features).

At this stage, the structure (59) is successfully spelled out, and if no more features are added (i.e., if we would want to derive the Nom case), the structure would be pronounced as the sequence of *gub* and *-a*. Note that as a result of spellout-driven movement, NomP follows the #P, so on the surface, *-a* follows *gub*.

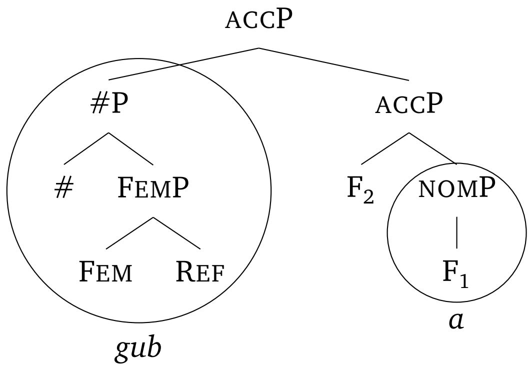
The derivation now proceeds by merging F_2 on top of (59), with the result shown in (63).

(63)



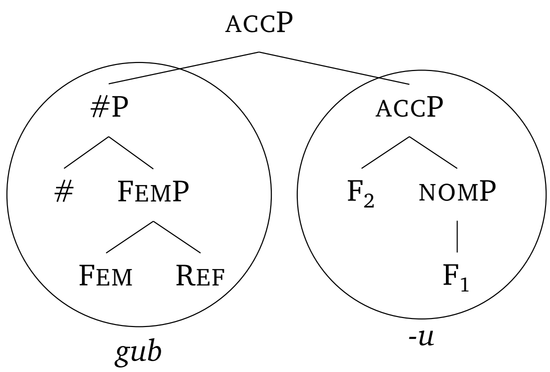
This structure cannot be spelled out as is, triggering spellout-driven movement. According to the spell out algorithm, the first operation that must be tried is the movement of the Spec of the complement of F_2. This phrase corresponds to the #P in (63), and so the #P is moved out, with the result in (64).

(64)



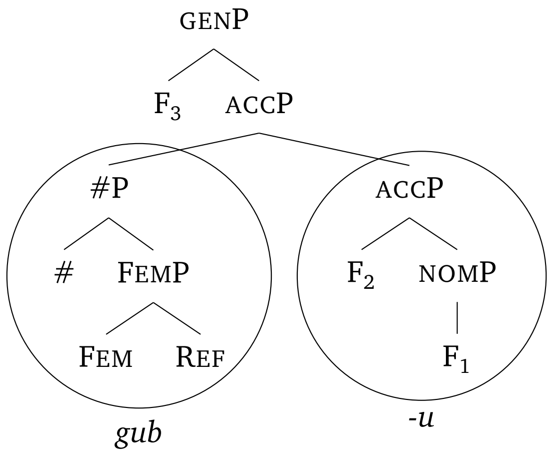
In this structure, the lower ACCP can be lexicalized by either the accusative *-u* (recall (61)) or by the genitive *-y* (recall (62)); both entries match. The accusative *-u* wins as it has fewer superfluous features. The result of the lexicalization is given in (65).

(65)



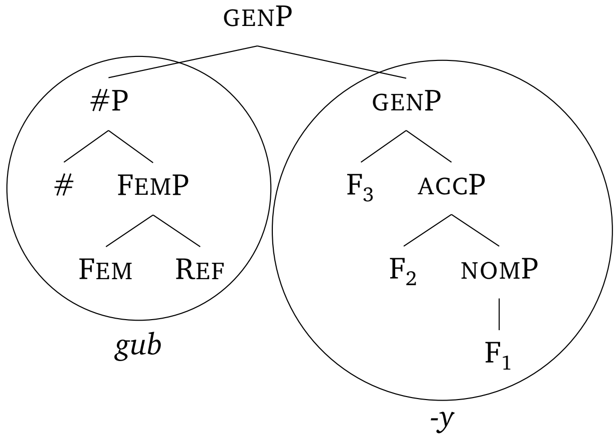
When the GEN feature F_3 is merged, the structure (66) arises.

(66)



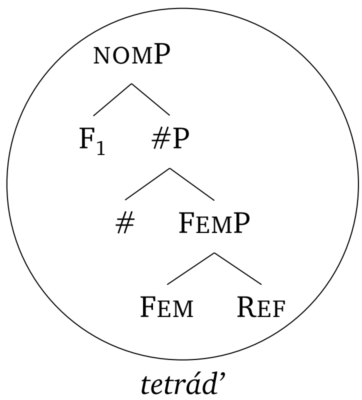
The structure is fed to spellout, and the by-now familiar sequence of steps is followed: lexicalization without movement is attempted but fails, because the full structure (66) is not contained in any lexical entry. Therefore, the Spec of the complement is moved, yielding (67). Lexicalization at the lower GENP is successful. The structure (67) shows the result.

(67)



Let me now come back to Declension III. We have already seen that the nominative is spelled out without the need for any movement, using just the root, recall (58) repeated in (68).

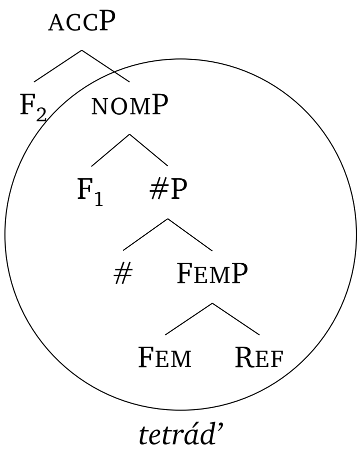
(68)



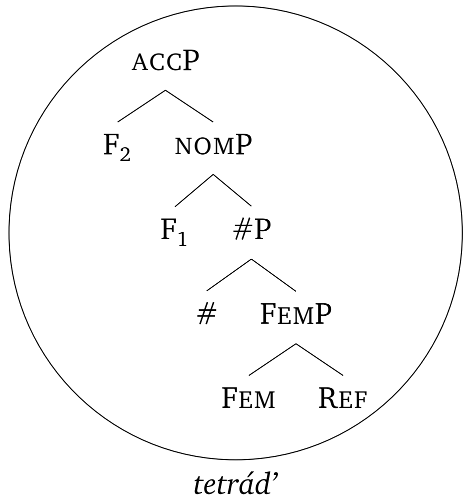
The reason for this is that spellout without movement is preferred by the spellout algorithm. Therefore, as long as the root can spell out the relevant features, this is the preferred type of lexicalization. And since the root of Declension III nouns is big, there is no need for movements in NOM.

The same holds for the accusative case. When the feature F_2 is added to the nominative, we first try to spell out without any movement. Given that Declension III roots can (by definition) spell out the whole structure, the spellout procedure yields (70) as the relevant output. This is again preferred to performing any movement, which only applies in Declension II (because the root is small, again by definition).

(69)

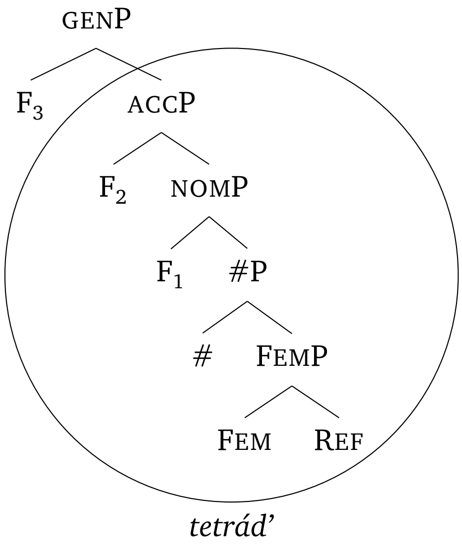


(70)

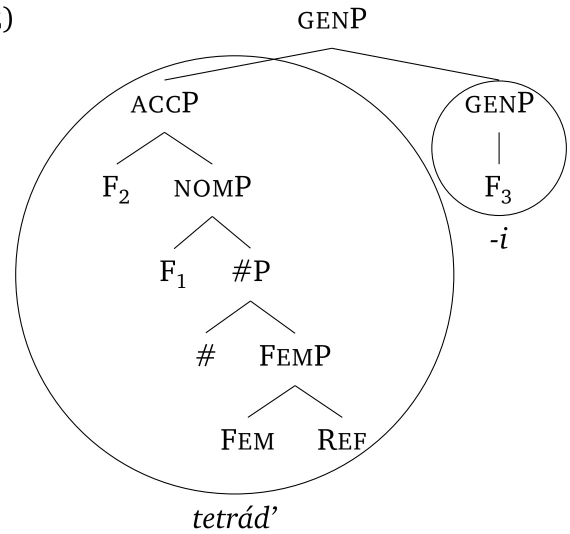


After successful spellout, (70) is fed into syntax again, yielding (71). This time, spellout without movement fails. Therefore, movement of the Spec should be tried, but the complement has no Spec. Therefore, the full complement moves, yielding (72).

(71)



(72)



Note that the lower GENP cannot by spelled out by any of the Declension II markers with entries as in (60) to (62). The reason is that the GENP constituent – with just the feature F_3 hanging down – is not contained in any of them. Instead, a special lexical entry is used, one which is depicted in (73). This entry perfectly matches the relevant constituent, and the correct form *tetrad’-i* is derived.

(73) GENP $\Leftrightarrow$ /*-i*/

$\quad$ F_3

In sum, cyclic spellout interacts with the spellout algorithm (56) in a way that lexical entries determine how the derivation unfolds. This first of all derives the correct ordering of markers, and at the same time, allows for different declensions to have a slightly different syntactic structure. This consequence can then be trivially generalized to the variation observed among different languages: since different languages have different lexical items, they will have different structures as a consequence.[6]

5.3 Further ramifications of the spellout algorithm

So far, I have been discussing the interaction between roots and suffixes. In this final section, I shall briefly introduce how prefixation works. Prefixation is a domain that is currently still under development within Nanosyntax. There is nevertheless a consensus concerning the core ideas, which I shall present here (see Starke 2018, De Clercq and Vanden Wyngaerd 2018, Caha et al. 2019a, De Clercq 2020, Caha to appear for some current work on prefixes).

Let me start from the observation that the suffixal nature of the Russian case endings is determined by the shape of the lexical tree associated to them. Consider, for instance, the nominative ending (60), repeated in (74) for convenience. The relevant property is that the lexical tree has just a single feature dependent on the lowest phrasal projection NOMP.

(74) NOMP $\Leftrightarrow$ /*-a*/

$\quad$ F_1

In a model like that of Chomsky (1995) (Bare Phrase Structure), such a configuration only arises in syntax when the second daughter of NOMP moves out of NOMP. And since movement is only to the left (as in Kayne 1994 a.m.o.), this means that *-a* will only ever be inserted as a suffix.

In this respect, prefixes differ from suffixes. Specifically, the lexical tree associated to a prefix has two daughters, as in (75).

(75) XP $\Leftrightarrow$ /*prefix*/

$\quad$ X $\quad$ Y

If such an entry is to be inserted, no movement may take place from within XP. And since

[6]Once the system has been introduced, let me make it explicit that its correct working presupposes that individual terminal nodes cannot be spelled out on their own. For example, if we allowed F_3 to be spelled out as a terminal in the structure (71) (on the grounds that the terminal is contained in the entry (73)), we would never be able to trigger movement as a consequence of a failed spellout. The absence of terminal spellout is encoded in the formulation of the spellout algorithm (56), which requires that spellout always targets an 'FP.'

nothing moves, the entry ends up as a prefix.

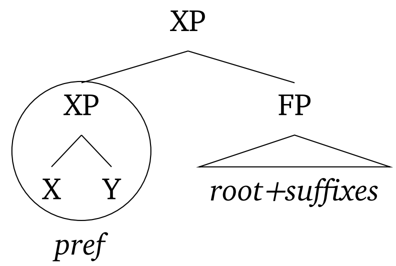
One example of such an entry we have already seen is the entry of roots. In the linear string, these are prototypically on the left edge (unless further prefixed). That is because they themselves move from below suffixes, but nothing has moved from below the root.

The second use for prefixes is when they spell out higher functional heads (e.g., the English preposition *of*, or the infinitive prefix *to*). In order for prefixes to be used this way, the structure underlying the prefix must be assembled in a separate workspace. This is relatively costly, and so in the hierarchy of rescue options, prefixes are the absolute last resort. Therefore, prefixes are only used when suffixal derivation fails. This is stated in (76).

(76) If the suffixal spell out of F fails, remove F from the main workspace. Start a new workspace, and build a phrase containing F that can be spelled out. Once done, Merge that phrase back with the main projection line.

Spec formation yields structures such as (77). Here on top of an FP, we want to add the feature X. Assume now that the lexicon has no suffix that spells out X: we only have the prefix in (75). All rescue movements therefore fail, and X must be removed from the main spine. A new derivational workspace is created and we build a phrase that is spelled out by the prefix. Once we are ready, the workspace is attached back to the main spine, yielding (77).

(77)



For reasons of space, I do not discuss any specific empirical cases; see the references at the beginning of this subsection.

6 Summary of key features

To sum up, let me repeat here the most important features of the Nanosyntax framework as introduced above, starting from the features it shares with DM:

(78) Nanosyntax and DM, shared features:

a. *Late Insertion*: All syntactic nodes systematically lack all phonological features. The phonological features are supplied – after the syntax – by consulting lexical entries in the postsyntactic lexicon.

b. *Syntax all the way down*: Terminal nodes are organized into hierarchical structures determined by the principles and operations of the syntax. Since terminal nodes correspond to units smaller than words, it follows that syntactic principles govern the internal structure of words. There is no sharp boundary between the traditional syntax and morphology.

The following properties have been identified as features that differentiate Nanosyntax from DM, though not necessarily from other frameworks (like Cartography Cinque and Rizzi 2010 or the work by Kayne 2005, Koopman 2005 and others):

(79) Features differentiating Nano from DM, but not other frameworks:

a. *No Bundling:* The atoms (terminal nodes) of syntactic trees are single features. All combinations of morphosyntactic features arise as the result of (binary) Merge. Pre-syntactic feature bundles do not exist, they correspond to phrases assembled by syntax.

b. *No Morphology.* There is no component of grammar other than syntax that has the power to manipulate syntactic structures. No nodes or features may be added or deleted outside of syntax, no displacement operations take place outside of the syntactic computation. Since there is no Morphology component, there can be no Morphology-specific features; all features relevant for morphological phenomena are syntactic features by definition.

Finally, the following features are specific to Nanosyntax; they are not present in any other framework, as far as I am aware.

(80) Features specific to Nano:

a. *Overspecification*. In order for a lexical item to be inserted in a node, the lexical entry must fully contain the syntactic node, including all its daughters, granddaughters, etc., all the way down to every single feature dominated by the node to be spelled out, and in exactly the right geometrical shape.

b. *Cyclic phrasal spellout*. Spell out must successfully apply to the output of every Merge F operation. After successful spellout, the derivation may terminate, or proceed to another round of Merge F, in which case a new round of spellout is initiated, and so on.

c. *Spellout-driven movement.* When Merge F produces an FP that cannot be spelled out (no lexical item matches the FP), the FP is rejected at the interface. Syntax then tries to rescue the structure by performing one in a predefined hierarchy of movement operations, before it sends the structure for spellout again. A variety of spellout-driven-movement operations may apply until spellout at FP succeeds.

References

Abels, Klaus and Peter Muriungi. 2008. The focus particle in Kîîtharaka: Syntax and semantics. *Lingua* 118: 687–731.

Alexiadou, Artemis and Gereon Müller. 2008. Class features as probes. Ms., Uni Stuttgart and Uni Leipzig.

Anderson, Stephen R. 1992. *A-morphous morphology*, vol. 62. Cambridge University Press, Cambridge.

Baerman, Matthew, Dunstan Brown, and Greville G. Corbett. 2005. *The Syntax-Morphology Interface. A Study of Syncretism*. Cambridge University Press, Cambridge.

Baunaz, Lena and Eric Lander. 2018a. Deconstructing categories syncretic with the nominal complementizer. *Glossa: a journal of general linguistics* 3 1.

Baunaz, Lena and Eric Lander. 2018b. Nanosyntax: the basics. In *Exploring Nanosyntax*, edited by Lena Baunaz, Karen De Clercq, Liliane Haegeman, and Eric Lander, pp. 3–56. Oxford University Press, Oxford.

Baunaz, Lena and Eric Lander. 2018c. Ontological categories. In *The Unpublished Manuscript*, edited by Pavel Caha, Karen De Clercq, and Guido Vanden Wyngaerd, pp. 1–18. lingbuzz/003993.

Bergsma, Fenna. 2019. Mismatches in free relatives–grafting nanosyntactic trees. *Glossa: a journal of general linguistics* 4 1.

Blevins, James P. 2016. *Word and paradigm morphology*. Oxford University Press, Oxford.

Blix, Hagen. 2020. Spans in South Caucasian agreement: Revisiting the Pieces of inflection. *Natural Language and Linguistic Theory* .

Bobaljik, Jonathan. 2006. Where's ϕ? Agreement as a post-syntactic operation. In *Phi-Theory: Phi Features Across Interfaces and Modules*, edited by Daniel Harbour, David Adger, and Susana Béjar, pp. 295–328. Oxford: Oxford University Press.

Bobaljik, Jonathan. 2012. *Universals In Comparative Morphology*. MIT Press, Cambridge, MA.

Bobaljik, Jonathan and Uli Sauerland. 2018. *ABA and the combinatorics of morphological features. *Glossa* 3: 15.1–34.

Bobaljik, Jonathan David. 2017. Distributed morphology. In *Oxford research encyclopedia of morphology*, edited by Rochelle Lieber. Oxford University Press.

Borer, Hagit. 2005. *In Name Only*. Oxford: Oxford University Press.

Caha, Pavel. 2009. *The Nanosyntax of Case*. Ph.D. thesis, University of Tromsø, Tromsø.

Caha, Pavel. 2013. Explaining the structure of case paradigms through the mechanisms of Nanosyntax. *Natural Language and Linguistic Theory* 31: 1015–1066.

Caha, Pavel. 2017. How (not) to derive a *ABA: the case of Blansitt's generalization. *Glossa* 2: 84.1–32.

Caha, Pavel. 2018. Syncretism as Merge F. Ms., Masaryk University, lingbuzz/004340.

Caha, Pavel. to appear. *Case competition in Nanosyntax. A study of numeral phrases in Russian and Ossetic*. Language Science Press, Berlin.

Caha, Pavel, Karen De Clercq, and Guido Vanden Wyngaerd. 2019a. The fine structure of the comparative. *Studia Linguistica* 73 1.

Caha, Pavel, Karen De Clercq, and Guido Vanden Wyngaerd. 2019b. On the difference between $\surd$ and root. Ms., Lingbuzz.

Caha, Pavel, Karen De Clercq, and Guido Vanden Wyngaerd. 2020. Zero morphology and change-of-state verbs. Talk at the Derivational zero affixes, SLE 2020.

Caha, Pavel and Marina Pantcheva. 2012. Contiguity beyond linearity. Talk at Decennium: The first 10 years of CASTL.

Calabrese, Andrea. 2008. On absolute and contextual syncretism. In *Inflectional Identity*, edited by Asaf Bachrach and Andrew Nevins, pp. 156–205. Oxford University Press, Oxford.

Chomsky, Noam. 1995. Bare phrase structure. In *Government binding theory and the minimalist program*, edited by Gert Webelhuth, pp. 383–439. Oxford University Press, Oxford.

Cinque, Guglielmo and Luigi Rizzi. 2010. The cartography of syntactic structures. In *The Oxford Handbook of Linguistic Analysis*, edited by Bernd Heine and Heiko Narrog, pp. 51–65. Oxford University Press, Oxford.

De Clercq, Karen. 2013. *A unified syntax of negation*. Ph.D. thesis, Ghent University.

De Clercq, Karen. 2019. French negation and feature conservation. In *Cycles in Language Change*, edited by Miriam Bouzouita, Anne Breitbarth, Lieven Danckaert, and Elisabeth Witzenhausen, pp. 199–227. Oxford University Press, Oxford.

De Clercq, Karen. 2020. *The Morphosyntax of Negative Markers*. De Gruyter Mouton, Berlin, Boston.

De Clercq, Karen and Guido Vanden Wyngaerd. 2017. *ABA revisited: evidence from Czech and Latin degree morphology. *Glossa* 2 1: 69: 1–32.

De Clercq, Karen and Guido Vanden Wyngaerd. 2018. Unmerging analytical comparatives. *Jezikoslovlje* 19 3: 341–363.

De Clercq, Karen and Guido Vanden Wyngaerd. 2019. Negation and the functional sequence. *Natural Language & Linguistic Theory* 37 2: 425–460.

Dékány, Éva. 2012. *A profile of the Hungarian DP: The interaction of lexicalization, agreement and linearization with the functional sequence*. Ph.D. thesis, Universitetet i Tromsø, Tromsø.

Di Sciullo, Anna-Maria and Edwin Williams. 1987. *On the definition of word*. MIT Press, Cambridge, MA.

Embick, David. 2016. On the targets of phonological realization. In *The Morphosyntax-Phonology Connection: Locality and Directionality at the Interface*, edited by Vera Gribanova and Stephanie S. Shih, pp. 255–283. Oxford University Press, Oxford.

Embick, David and Alec Marantz. 2008. Architecture and blocking. *Linguistic Inquiry* 39 1: 1–53.

Embick, David and Rolf Noyer. 2007. Distributed morphology and the syntax/morphology interface. In *The Oxford Handbook of Linguistic Interfaces*, edited by Gillian Ramchand and Charles Reiss, pp. 289–324. Oxford University Press, Oxford.

Fábregas, Antonio. 2007. An exhaustive lexicalisation account of directional complements. In *Nordlyd, special issue on Space, Motion, and Result*, edited by Monika Bašić, Marina Pantcheva, Minjeong Son, and Peter Svenonius, no. 34.2 in Tromsø Working Papers on Language and Linguistics, pp. 165–199. CASTL, University of Tromsø, Tromsø.

Halle, Morris. 1997. Distributed morphology: impoverishment and fission. *MIT Working Papers in Linguistics* 30: 425–449.

Halle, Morris and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In *The View from Building 20*, edited by Ken Hale and Jay Keyser, pp. 111–176. MIT Press, Cambridge, MA.

Halle, Morris and Alec Marantz. 1994. Some key features of distributed morphology. In *Papers on Phonology and Morphology*, edited by Andrew Carnie, Heidi Harley, and Tony Bures, vol. 21 of *MIT Working Papers in Linguistics*, pp. 275 – 288. Cambridge, Mass.

Halle, Morris and Bert Vaux. 1998. Theoretical aspects of Indo-European nominal morphology: The nominal declensions of Latin and Armenian. In *Mir Curad: Studies in Honor of Clavert Watkins*, edited by Jay Jasanoff, H. Craig Melchert, and Lisi Olivier, Innsbrucker Beiträge zur Sprachwissenschaft, pp. 223–240. Institut für Sprachwissenschaft der Universität Innsbruck, Innsbruck.

Hardarson, Gísli Rúnar. 2016. A case for a weak case contiguity hypothesis—a reply to Caha. *Natural Language & Linguistic Theory* 34 4: 1329–1343.

Harley, Heidi. 2014. On the identity of roots. *Theoretical Linguistics* 40: 225–276.

Harley, Heidi and Rolf Noyer. 1999. State-of-the-article: distributed morphology. *Glot International* 4: 3–9.

Harley, Heidi and Elizabeth Ritter. 2002a. Person and number in pronouns: A feature-geometric analysis. *Language* 78 3: 482–526.

Harley, Heidi and Elizabeth Ritter. 2002b. Structuring the bundle: A universal morphosyntactic feature geometry. In *Pronouns: Grammar and Representation*, edited by Horst J. Simon and Heike Wiese, p. 23–39. John Benjamins, Amsterdam.

Haugen, Jason D. and Daniel Siddiqi. 2016. Towards a restricted realization theory: Multimorphemic monolistemicity, portmanteaux, and post-linearization spanning. In *Morphological metatheory*, edited by Daniel Siddiqi and Heidi Harley, pp. 343–385. John Benjamins, Amsterdam.

Holaj, Richard. 2018. Balancing between roots and thematic vowel. In *The Unpublished Manuscript*, edited by Pavel Caha, Karen De Clercq, and Guido Vanden Wyngaerd, pp. 81–94. Lingbuzz.

Jakobson, Roman. 1936 [1962]. Beitrag zur allgemeinen Kasuslehre: Gesamtbedeutungen der russischen Kasus. In *Selected writings, vol. 2*, pp. 23–71. Mouton, The Hague.

Janků, Lucie and Michal Starke. 2020. Growing roots in Italian nominals: Nanosyntactic account. Talk at ConSOLE28.

Johnston, Jason. 1996. *Systematic Homonymy and the Structure of Morphological Categories. Some Lessons from Paradigm Geometry*. Ph.D. thesis, University of Sydney.

Julien, Marit. 2015. Theme vowels in north sámi: Spanning and maximal expression. *Lingua* 164: 1–24.

Kayne, Richard. 1994. *The Antisymmetry of Syntax*. MIT Press, Cambridge, MA.

Kayne, Richard. 2005. Some notes on comparative syntax: With special reference to English and French. In *Movement and Silence*, pp. 277–333. Oxford University Press, Oxford.

Kayne, Richard S. 2010. Toward a syntactic reinterpretation of Harris & Halle (2005). In *Selected papers from "Going Romance," Groningen 2008*, edited by R. Bok-Bennema, B. Kampers-Manhe, and B. Hollebrandese, pp. 145–170. John Benjamins, Amsterdam.

Kennedy, Christopher and Louise McNally. 2005. Scale structure, degree modification, and the semantics of gradable predicates. *Language* 81: 345–381.

Kiparsky, Paul. 1973. 'Elsewhere' in phonology. In *A Festschrift for Morris Halle*, edited by Stephen Anderson and Paul Kiparsky, pp. 93–106. Holt, Rinehart & Winston, New York.

Koopman, Hilda. 2005. Korean (and Japanese) morphology from a syntactic perspective. *Linguistic Inquiry* 36 4: 601–633.

Lander, Eric. 2015. *The nanosyntax of the Northwest Germanic reinforced demonstrative*. Ph.D. thesis, Ghent University.

Lander, Eric and Liliane Haegeman. 2018. The nanosyntax of spatial deixis. *Studia linguistica* 72 2: 362–427.

Marantz, Alec. 1991. Case and licensing. In *Proceedings of ESCOL '91*, edited by Benjamin Ao German Westphal and Hee-Rahk Chae, pp. 234–253. Cornell Linguistics Club, Ithaca.

McCawley, James. 1968. Lexical insertion in a transformational grammar without deep structure. *CLS* 4: 71–80.

McCreight, Katherine and Catherine V. Chvany. 1991. Geometric representation of paradigms in a modular theory of grammar. In *Paradigms: The Economy of Inflection*, edited by Frans Plank, pp. 91 – 112. Mouton de Gruyter, Berlin, New York.

McFadden, Thomas. 2004. *The position of morphological case in the derivation*. Ph.D. thesis, UPenn.

McFadden, Thomas. 2018. * aba in stem-allomorphy and the emptiness of the nominative. *Glossa: a journal of general linguistics* 3 1.

Merchant, Jason. 2015. How much context is enough?: Two cases of span-conditioned stem allomorphy. *Linguistic Inquiry* 46: 273–303.

Müller, Gereon. 2004. A distributed morphology approach to syncretism in russian noun inflection. In *Proceedings of FASL*, vol. 12, pp. 353–373.

Neeleman, Ad and Hans Van de Koot. 2020. Deep and surface *ABA syncretism patterns. Ms., UCL.

Neeleman, Ad and Kriszta Szendrői. 2007. Radical pro-drop and the morphology of pronouns. *Linguistic Inquiry* 38: 671–714.

Newell, Heather and Maire Noonan. 2018. A re-portage on spanning; feature portaging and non-terminal spell-out. In *McGill Working Papers in Linguistics*, pp. 156–167. McGill University, Montreal.

Pantcheva, Marina. 2010. The syntactic structure of locations, goals, and sources. *Linguitics* 48: 1043–1081.

Plank, Frans. 1991. Rasmus Rask's dilemma. In *Paradigms: The Economy of Inflection*, edited by Frans Plank, pp. 161–196. Mouton de Gruyter, Berlin.

Radkevich, Nina. 2010. *On Location: The structure of case and adpositions*. dissertation, University of Connecticut, Storrs, CT.

Ramchand, Gillian. 2008. *Verb Meaning and the Lexicon*. Cambridge University Press, Cambridge.

Richards, Norvin. 2001. An idiomatic argument for lexical decomposition. *Linguistic inquiry* 32 1: 183–192.

Siddiqi, Daniel. 2006. *Minimize Exponence: Economy Effects on the Morphosyntactic Component of the Grammar*. Ph.D. thesis, University of Arizona.

Smith, Peter W, Beata Moskal, Ting Xu, Jungmin Kang, and Jonathan David Bobaljik. 2019. Case and number suppletion in pronouns. *Natural Language & Linguistic Theory* 37 3: 1029–1101.

Starke, Michal. 2002. The day syntax ate morphology. Class taught at the EGG summerschool, Novi Sad.

Starke, Michal. 2009. Nanosyntax: A short primer to a new approach to language. *Nordlyd* 36: 1–6.

Starke, Michal. 2014a. Cleaning up the lexicon. *Linguistic Analysis* 39: 245–256.

Starke, Michal. 2014b. Towards an elegant solution to language variation: Variation reduces to the size of lexically stored trees. In *Linguistic Variation in the Minimalist Framework*, edited by Carme M. Picalo, pp. 140 – 153. Oxford University Press, Oxford.

Starke, Michal. 2017. Resolving (dat = acc) ≠ gen. *Glossa* 2 1: 104.1–8.

Starke, Michal. 2018. Complex left branches, spellout, and prefixes. In *Exploring Nanosyntax*, edited by Lena Baunaz, Karen De Clercq, Liliane Haegeman, and Eric Lander, pp. 239–249. Oxford University Press, Oxford.

Svenonius, Peter. 2012. Spanning. Ms., CASTL.

Taraldsen, Knut Tarald. 2010. The nanosyntax of Nguni noun class prefixes and concords. *Lingua* 120 6: 1522–1548.

Taraldsen, Knut Tarald. 2019. Nanosyntax and syncretism in multidimensional paradigms. *Linguistics Vanguard* 5 1.

Taraldsen Medová, Lucie and Bartosz Wiland. 2019. Semelfactives are bigger than degree achievements. *Natural Language & Linguistic Theory* 37 4: 1463–1513.

Timberlake, Alan. 2004. *A reference grammar of Russian*. Cambridge University Press, Cambridge.

Truong, Tran. 2020. * aba effects in kinship allomorphy & syncretism. *Proceedings of the Linguistic Society of America* 5 1: 345–353.

Van Baal, Yvonne and Jan Don. 2018. Universals in possessive morphology. *Glossa: a journal of general linguistics* 3 1.

Vanden Wyngaerd, Guido. 2018. The feature structure of pronouns: a probe into multidimensional paradigms. In *Exploring Nanosyntax*, edited by Lena Baunaz, Karen De Clercq, Liliane Haegeman, and Eric Lander, pp. 277–304. Oxford University Press, Oxford.

Vanden Wyngaerd, Guido, Michal Starke, Karen De Clercq, and Pavel Caha. 2020. How to be positive. *Glossa* 5 1.

Weerman, Fred and Jacqueline Evers-Vermeul. 2002. Pronouns and case. *Lingua* 112: 301–338.

Wiland, Bartosz. 2012. Prefix stacking, syncretism, and the syntactic hierarchy. In *Slavic Languages in Formal Grammar*, edited by Mojmír Dočekal and Markéta Ziková, p. 307–324. Peter Lang, Berlin/Frankfurt.

Wiland, Bartosz. 2019. *The spell-out algorithm and lexicalization patterns: Slavic verbs and complementizers*. Language Science Press, Berlin.

Williams, Edwin. 2003. *Representation Theory*. MIT Press, Cambridge, MA.

Ziková, Markéta and Michaela Faltýnková. in press. How to derive allomorphy: A case study from Czech. *The Linguistic Review* .

Zompì, Stanislao. 2017. *Case decomposition meets dependent-case theories*. Master's thesis, Pisa: Universita de Pisa.