Minimalism and Nanosyntax: Reconciling Late Insertion and the Borer-Chomsky Conjecture

Pavel Caha
Masarykova univerzita

August 14, 2021

1 Introduction

Nanosyntax (Starke 2009, 2018) is an approach to grammar within the Principles & Parameters framework, which shares with standard minimalist theories (Chomsky 1995 et seq.) the idea that the core of syntax consists in a bottom-up application of (external and internal) Merge, which is arguably the main (and perhaps only) syntactic operation (cf. [CITKO], [ZWART]).[1] The Nanosyntax framework can be easily distinguished within current approaches by its adherence to three core ideas: (i) the idea that the atoms of syntax are just single features; (ii) the idea that insertion of phonology and/or concepts happens after syntax (Late Insertion); and (iii) the idea that lexical insertion targets phrasal nodes.[2]

Due to its focus on syntactic atoms (recall (i) above), a lot of the empirical work in Nanosyntax focuses on traditional morphological phenomena such as syncretism, allomorphy, suppletion and others. However, one of the main driving forces of this research is to show that these phenomena can be fully explained by subjecting traditional morphological features to the same set of rules as observed in phrasal syntax.

Like Minimalism, Nanosyntax makes no reference to syntax-internal levels of representation (such as Deep Structure, Surface Structure or Morphological Structure), placing much of the explanatory burden on the interface between syntax and the external systems (PF and CF). As highlighted under (iii) above, the Nanosyntactic theory of the interface is based on the idea of phrasal spellout, which assigns phonological and/or conceptual interpretation to phrasal nodes. In recent incarnations of the model, phrasal spellout is cyclic (applying after every Merge) and it must be successful at every cycle. When spellout fails,

[1] I am grateful to an anonymous reviewer for their comments that helped improve this chapter. I also benefited from talking to Karen De Clercq about aspects of this article. My work on this chapter has been supported by a grant from GAČR (The Czech Science Foundation) number GA19-07004S.

[2] A brief note on history: Starke's work on Nanosyntax goes back to the early 2000s, but it has remained unpublished. In addition to Starke's (2009) article, early published sources include Caha (2007, 2009, 2010), Fábregas (2007, 2009), Pantcheva (2009, 2011), Taraldsen (2009, 2010). More recent work is cited at appropriate places in the chapter.

the derivation crashes at the interface and it must be repaired in specific ways, namely by (spellout-driven) movement.

The current chapter looks at the three core ideas highlighted in (i)-(iii) above. In section 2, I describe the particular view on the atoms of language, which is at the core of the framework. In section 3, I turn to the crucial role that Late Insertion has in the model. In Sections 4, 5 and 6, I provide a more specific information about the postsyntactic lexicon and describe how its interaction with a language-invariant lexicalisation procedure leads to cross-linguistic variation. Section 7 concludes by highlighting how Nanosyntax resolves one of the tensions in current Minimalist thinking, which relates to the lexicon.

The specific tension I have in mind arises when we juxtapose two independent ideas about the role of the lexicon in grammar. The first idea is Late Insertion. According to this proposal, syntax is modular and deals only with syntactic information. The lexicon therefore comes late in the derivation, providing phonological/conceptual features only after syntax has finished its job (cf. [KALIN&WEISSER], [SVENONIUS-LATE], [AKKUŞ]). The second idea is that the principles of language are invariant, and that the lexicon is therefore the only source of crosslinguistic variation (the so-called Borer-Chomsky conjecture).

The two ideas, while both attractive, lead to a tension. Specifically, if the lexicon is placed *after* syntax, it is difficult to see how it can influence parametric variation *inside* syntax. The point of Section 7 is to show how the architecture of grammar proposed in Nanosyntax resolves the tension and allows us to maintain both ideas at the same time. The key to the answer is the cyclic nature of spellout, where spellout applies after every Merge. Because of this, we get an interleaving of Merge–Spellout–Merge–Spellout etc.; as a result, some Merge applies after spellout. That way, even though the lexicon comes in late, it can still be the only source of variation (while preserving modularity).

2 The atoms

In this section, I address the nature of LEX in Nanosyntax. My starting point is the fact that one of the major changes that arrived with the Minimalist Program (Chomsky 1995) was the elimination of the 'syntax-internal' levels of representation, Deep Structure and Surface Structure. In the new architecture, structures are no longer created by the so-called rewrite rules conforming to the general template of the $\bar{X}$-theory. Rather, derivations now unfold in a stepwise manner, starting from a set of atoms, and combining these into ever larger trees, ultimately producing the whole sentence.

This change in perspective brings along many consequences. One of these is that in the new setting, syntactic structures are no longer made of abstract 'positions,' but rather of something more tangible, namely of some pre-existing elements, standardly referred to as syntactic atoms. As a result, the question what the atoms are becomes of central importance.

The standard view on syntactic atoms, adopted in Chomsky's work and more or less by everyone in the field today, is that they are language-particular lexical items. In Chomsky's *Beyond Explanatory Adequacy* (2004:104, 107), the idea is described as follows: "FL appears to be a species property, close to uniform across a broad range. It has a genetically-determined initial state S_0, which determines the possible states it can assume. [...] S_0 determines the set {F} of properties ("features") available for languages. Each language makes a one-time selection of a subset [F] of {F} and a one-time assembly of elements of [F] as its lexicon [LEX]."

This view introduces a two-step procedure by which syntactic atoms (the smallest elements subject to Merge) arise from the universal features provided by the UG. The first step is a reduction: starting from the set of properties provided by the UG, each language restricts this inventory to the set of features relevant for that particular language. For instance, one language may lack PLURAL while another language has it. The second step corresponds to producing language particular groupings of the selected features: in order to form syntactic atoms, language-specific features are grouped into lexical items. This view is summarised in (1).

(1) The atoms of Merge in standard Minimalism

- a. Start from {F}, the set of features provided by UG
- b. Reduce {F} to the set of features used by individual languages [F]
- c. Construct the atoms of Merge (LEX) as language-particular sets (assemblies) composed of language particular features, members of [F]

Ultimately, this is also the theory of how cross-linguistic differences arise, since the rules and operations of narrow syntax are assumed to be uniform across languages. The theory of variation that emerges from this picture can thus be summarised by saying that each language has different atoms, and that languages differ as a consequence of how the distinct atoms interact with the invariant narrow-syntactic computation.

This general picture, however, leaves a number of alternatives open for exploration. In particular, various theories within the Principles & Parameters framework adopt different positions on the relationship between the atoms of syntax (LEX) and the universal set of features {F}.[3] One of the central features of Nanosyntax is the hypothesis that can be expressed by the formula {F} = [F] = LEX. In words, the idea is that the set of atoms each language uses to build structures (LEX) corresponds to the set of features provided by the UG ({F}).

The hypothesis, as described above, has strictly speaking two parts. The first part ([F] = {F}) says that all languages use the same features, namely the set provided by the UG; see (2b). The second part (LEX = [F]) says that LEX contains only single features rather than sets/bundles (assemblies in Chomsky's terms); see (2c).

[3] An example of such an alternative is Hegarty (2005:29ff). Hegarty builds on Giorgi and Pianesi (1996) and proposes that features are assembled into syntactic atoms for each derivation (numeration), rather than once and for all, as in Chomsky's work.

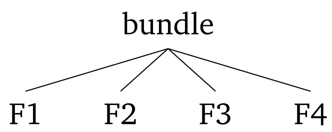
(2) The atoms of Merge in Nanosyntax

a. UG provides the set {F} of features (properties) available to language
b. The set of features used by individual languages [F] is identical to {F}
c. The atoms of Merge in individual languages (LEX) are individual features, members of [F]

The consequence of the claims in (2) is that Nanosyntax has a different starting point compared to other theories. In virtually any framework, the input to syntax is a language-particular list, the presyntactic lexicon. In Nanosyntax, there is no language-particular presyntactic list: all languages start from the same features. As a consequence, Nanosyntax also has a different take on the variation among languages. Since they cannot reside in the presyntactic lexicon, they must reside in the postsyntactic lexicon, and this requires a rather different architecture of grammar compared to the standard view. I shall be exploring the architecture in this chapter.

Let me start by providing some reasons for (2), starting from the second statement (LEX = [F]). According to Starke (2014a), the main reason why syntax should not start from bundles is that such bundles are equivalent to "[e]nclosing elements inside square brackets." This is in turn "a notational variant of linking those elements under a single mother node. Feature bundles are thus trees, typically flat n-ary trees with n > 2. This means that a syntactic representation with 'feature bundles' in its terminals is composed of two types of trees, each with their own conventions: the binary branching syntactic nodes, and the n-ary branching lexical nodes at the bottom. [...] In other words, we just invented a second syntax and a new type of merge, for the purpose of lexical storage." An example of a feature bundle that Starke talks about is in (3).

(3) *Feature bundle = flat* n-*ary tree*



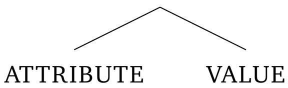
In abandoning feature bundles, Nanosyntax comes close to various other frameworks, often related in spirit. For instance, Kayne (2005) (cf. Collins and Kayne 2021) has proposed that "UG imposes a maximum of one interpretable features per lexical item." Also in Cartography (Cinque and Rizzi 2010), the maxim "one (morphosyntactic) property – one feature – one head" has been adopted as a research guideline.

It is worth pointing out that as a result of insisting on the atomic nature of features, Nanosyntax only uses privative features. These are features like PLURAL or PARTICIPANT. They have no internal structure and no values: they are either present or absent. When they are present in the structure, their grammatical meaning is included in the interpretation. When they are absent, their meaning is not included.

Binary features like [+/-PLURAL] or multivalent features [NUMBER:PL] are not used in

Nanosyntax. The reason is that both of these feature types are internally complex objects, decomposable into an attribute and a value. The attribute and the value are “enclosed in square brackets,” i.e., we get an object like [ATTRIBUTE:VALUE]. Rather than enclosing the attribute and the value in brackets, we could also represent this object as in (4), which makes its composite nature clear.

(4) *Attribute-value pair as a complex object*

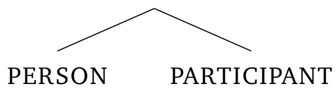


We already know that Nanosyntax (as one of its architectural properties) avoids postulating complex presyntactic objects like (4). Therefore, the general approach to objects such as (4) followed in Nanosyntax is to understand the value and the attribute as syntactic atoms, and understand attribute value pairs (where needed) as complex objects created by Merge.

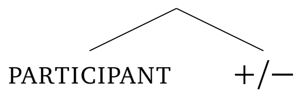
For reasons of space, I cannot devote too much attention to this issue, but for concreteness, let me provide an example. Consider, for instance, two common representations of the privative PARTICIPANT feature (which is a feature characteristic for the 1st and 2nd person). In multivalent theories, this feature is usually rendered as [PERSON:PARTICIPANT], which is equivalent to the structure (5a). In a theory with binary features, one would use [+/−PARTICIPANT], which is equivalent to (5b).

(5) *Multivalent and binary* PARTICIPANT *feature*

a.



b.



When it comes to multivalent features like (5a), their reinterpretation into privative features is almost trivial. Specifically, where objects such as (5a) are needed, Nanosyntax analyses them as the combination of two atomic features (PERSON and PARTICIPANT), a strategy followed, e.g., in Vanden Wyngaerd (2018). What he proposes is that (5a) is literally constructed by syntax, merging an atomic PERSON feature with an atomic PARTICIPANT feature, producing a complex syntactic structure [PARTICIPANT PERSON]. This structure is equivalent to using a presyntactic “atom” like (5a), and that is precisely the point. In sum, multivalent features are equivalent to structures created by Merge, and they are therefore treated as such in Nanosyntax.[4]

Similar considerations apply to binary features, even though here the issues are admittedly a bit more complex. As pointed out in Harbour (2011), binary features generally allow for a three-way distinction between the absence of a feature Ø, its positive value [+F]

[4] In Vanden Wyngaerd’s proposal, the first person is then created by further merging the atomic SPEAKER feature, producing a standard syntactic hierarchy of the type [SPEAKER [PARTICIPANT PERSON]], which we shall discuss in greater detail later on. However, it is also possible to imagine a system where 1st person is created by merging the SPEAKER feature directly as the value of PERSON.

and its negative value [−F]. An interesting hint as to how such a contrast can be restated in privative terms is related to Harbour's (2011:562) observation that the interpretation of [−F] is derived compositionally from [+F] by applying negation to it, i.e., according to Harbour, $[\![-F]\!] = \neg[\![+\mathrm{F}]\!]$. If that is so, it seems possible to encode the three way contrast that Harbour talks about as an opposition between the absence of a feature Ø, its presence [F] and a two-feature combination of F and a regular negation feature Neg, yielding [Neg F]. While it would be interesting to look at the potential differences (if any) between the [−F] and the [Neg F] notations, this is beyond the scope of the current chapter (cf. [BEJÁR], [MATUSHANSKY]).

In sum, the requirement of atomicity precludes the Nanosyntactic theory from using multivalent and binary features as syntactic atoms, even though objects corresponding to such features can be, where needed, understood as complex structures produced by Merge.

Let me now leave the atomic nature of the elements in LEX behind and let me turn to the second idea in (2), which is that the language-specific selection of features [F] is the same set as the set of features provided by UG {F}, i.e., that [F] = {F}. This idea has been probably most systematically argued for in the Cartography framework (cf. [SVENONIUS-CARTO]). Cinque and Rizzi (2010:55) state it as a general methodological guideline 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, whether the language offers overt evidence for it or not." While this guideline is certainly controversial, I mention it here since it underlies much of the work done within Nanosyntax as well, a point I shall return to in Section 5.

I close this section by briefly repeating the main message, which is that in Minimalism, the nature of syntactic atoms (LEX) becomes central. In Chomsky's own work, few restrictions are associated with syntactic atoms: they correspond to language-specific collections of grammatical features used by individual languages. Nanosyntax, on the other hand, places the most severe restrictions on LEX; specifically, as one of its core hypotheses, it makes LEX (the list of syntactic atoms used by particular languages) non-distinct from {F} (the set of features provided by UG). Since these sets are non-distinct, the distinction among LEX, [F] and {F} can be eliminated and replaced by just a single set, namely the set of universal features. As a further consequence, the features are taken to be privative.

This view leads to a theory without any language-particular presyntactic lexicon. In other words, not only are the operations of narrow syntax identical across languages, the list of syntactic atoms is also identical for all languages as well (cf. Starke 2014b).

3 Late Insertion

The question that immediately arises in this setup concerns language variation. If all languages have the same syntactic atoms, and if the rules of their combination are also identical (as is standard in Minimalism), then how can it be that languages differ? It turns out that this is possible because of the fact that Nanosyntax subscribes to Late Insertion (cf. [KALIN&WEISSER], [SVENONIUS-LATE], [AKKUŞ]). In Late-Insertion theories, syntax operates over features that lack any phonology or concepts. Phonological and conceptual information is only activated ('inserted') after syntax, during the so-called 'spellout.' And this is also where cross-linguistic differences arise: late, but still before the PF. This model is depicted in Figure 1.

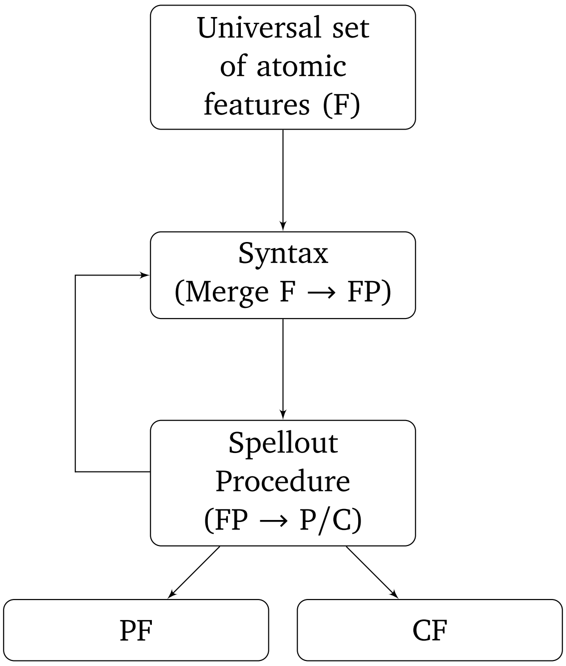


Figure 1: A late-insertion model of grammar (Nanosyntax)

We can see here that the model of grammar starts from single features. These are fed into syntax. Syntax assembles these features into syntactic phrases (FPs). FPs produced by syntax are then fed into the lexicon and mapped onto phonological form (PF) and conceptual form (CF) by the so-called spellout procedure (cf. [IRURTZUN] and [SCHEER]). In Nanosyntax, this is a (language-invariant) procedure that searches the postsyntactic lexicon for lexical items matching the syntactic configurations provided by syntax. The postsyntactic lexicon is represented in the diagram by the formula FP → P/C, since in Nanosyntax, it links syntactic constituents (FPs) to phonology (P) and/or concepts (C).

As we shall see in the following sections, the spellout procedure interacts with the postsyntactic lexicon and it can trigger language variation in a number of ways. The most straightforward way corresponds to the simple fact that, for instance, the plural is realized as *-s* in English and *-er* in Norwegian. However, as we will see, the role of the lexicon goes also beyond such simple facts, and this is the meaning of the 'feedback-loop' arrow leading from Spellout back to Syntax. However, even without knowing the details of how exactly the spellout procedure and the postsyntactic lexicon influence syntax, it is easy to see that having a feedback loop of this sort is something that actually allows for the radical position described in the previous section, which is that all languages have the same syntactic atoms and the same rules for their combination.

Apart from allowing for language-invariant syntactic atoms, Late-Insertion models (of which Nanosyntax is a representative) have an independent motivation. Since Late Insertion directly ties with the idea that syntax operates on language-invariant universal features (rather than on language-specific lexical items), let me spell this out in more detail.

The main reason for thinking that Late Insertion is independently needed is that it implements a modular view on syntax. In order to see how this is relevant, it is instructive to compare a Late-Insertion model with an Early-Insertion model. In Early-Insertion models, the computation operates over lexical items that contain also phonology and/or concepts (i.e., non-syntactic information). To give a concrete example, in Chomsky's (1995) approach, "[t]he lexical entry for *airplane*, for example, contains three collections of features: phonological features such as [begins with vowel], semantic features such as [artifact], and formal features such as [nominal]. The phonological features are stripped away by Spell-Out and are thus available only to the phonological component; the others are left behind by Spell-Out, and the formal ones may continue to be accessed by the covert computation to LF."

The passage makes it clear that syntactic atoms in such a model contain also phonological and conceptual information. Now since the syntactic derivation builds on such lexical items, it follows that minimally at its early stages, the syntactic derivation also contains phonological and conceptual information. This is depicted in 2, where a language-specific presyntactic lexicon feeds the syntactic derivation, and the syntax box therefore contains all three kinds of features (F, P, C).

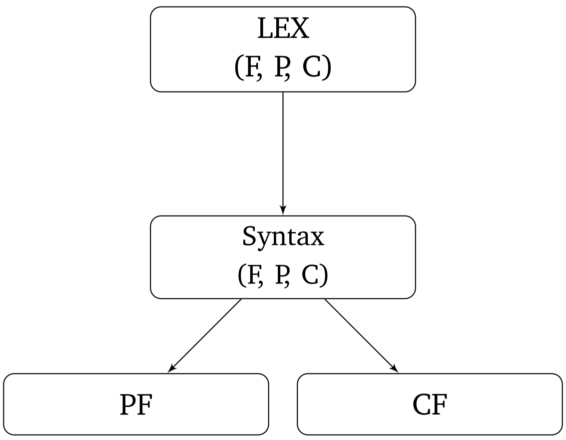


Figure 2: An early-insertion model of grammar

With this in mind, consider now the observation that syntactic operations are only driven by syntactic features and never by phonological features (Zwicky 1969, Zwicky and Pullum 1986, Marantz 1996, Miller et al. 1997). For example, across languages, there are classes of verbs with special behavior depending on grammatical features such as modality, aspect, voice, etc. However, there are no special classes of verbs depending on features such [begins with vowel], etc. The literature cited above has summarised such effects under the label of

Phonology-Free Syntax, see (6).

(6) *Principle of Phonology Free Syntax*
Rules of syntax make no reference to phonology.

A related observation (Marantz 1994) is that concepts also do not influence the operations of syntax. For example, languages display differential object marking depending on the animacy and/or specificity of the direct object, but they do not differentially mark objects that correspond to an [artifact].

(7) *Principle of Concept Free Syntax*
Rules of syntax make no reference to concepts.

In sum, the observation is that neither conceptual or phonological features influence the working of the syntactic computation, an observation that is a part of a larger hypothesis referred to as modularity (Fodor 1983).

These observations naturally follow in Late Insertion models, recall Figure 1. We can see that in such a model, syntax starts from single features (Fs) and assembles them into syntactic trees. Such trees are then the input to the lexicon, which links such representations to their corresponding representations at PF and CF. Thus, instead of "stripping" the narrow syntactic derivation of phonological and conceptual information, spellout (the lexicalisation procedure) *introduces* both phonology and concepts (by activating the relevant PF/CF representations). We shall look at this process in more detail in the next section; for now, the point is that in such a model, the principles of phonology/concept-free syntax (recall (6) and (7)) directly follow from the architecture.

These effects do not follow from the architecture in the classical Early-Insertion model, recall Figure 2. We can see that here, the syntactic computation begins with traditional lexical items that contain not only syntactic information, but also phonological and semantic information. Clearly, in such a model, something extra needs to be said as to why syntax can make reference to only one type of information that is present in the atoms, but not to the other two types. And while some answers could clearly be given, the point is simply that neither of the principles (6) or (7) is *predicted* on the basis of such an architecture, and an independent explanation must be provided *post hoc*.

The architectural conclusion that Nanosyntax draws from this is that a model that proposes Late Insertion of both phonology and concepts has the right type of architecture from which these observations naturally follow. Moreover, once the lexicon is placed after syntax for these independent reasons, it becomes possible to explore the option that the postsyntactic lexicon is, in fact, the *only* component of grammar that is language specific. The following sections visit a couple of relevant examples showing how this can be achieved.

4 The Postsyntactic Lexicon in Nanosyntax

In order to serve the purpose of mapping syntax to sound and conceptual meaning, the lexicon is a storage place for memorised links between syntactic structures on the one hand and sound/meaning on the other hand. The basic idea is that when a syntactic structure arrives at the interface, the spellout procedure tries to match this structure against the stored information in the lexicon, thereby mapping the syntactic structure onto the corresponding sound and meaning. If the procedure succeeds to match the structure against the stored information, we can say that the derivation converges. If matching fails, the derivation crashes (there is no way to map such a derivation to sound/meaning).

As an example of a lexical item, consider, for instance, the suppletive comparative *worse*. The idea is that when a child hears *worse*, the child will remember it and store it in the form of a lexical entry like (8). The lexical entry is a link between a particular syntactic structure (the comparative of an adjective) and a particular phonology. The lexical entry will then be used as a "translation" instruction: when syntax builds the structure as in (8), it will be realised as *worse* (see Caha et al. 2019 for a discussion of root suppletion).

(8) CMPRP ⇔ /*wɜːs*/

CMPR A

Constructing lexical items as links between syntactic structures and their phonological/conceptual representation leads to a particular consequence. The consequence is that the lexicon only contains well-formed syntactic structures: this is because only such structures are produced by syntax and can therefore be remembered as somehow special (i.e., associated to a non-compositional form or meaning).

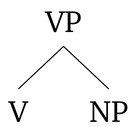
The general conclusion we can draw from this is that the format of lexically-stored trees is restricted: the lexicon only contains well-formed syntactic trees (respecting binary branching, the universal hierarchy of projections, etc.). This is an interesting conclusion, because it contrasts with the standard view that language-specific lexicons are formed in an essentially unprincipled manner as haphazard collections of syntactic features. In the Nanosyntax model, these collections of syntactic features are not grouped into lexical items inside the lexicon. Rather, it is syntax who forms the collections (i.e., syntactic constituents), and the lexical items only link independently existing constituents to phonology and/or meaning. As in most accounts, the most important reason for storage is non-compositionality: the reason why we must remember *worse* is that it is non-compositional. On the other hand, we do not need to remember *long-er* because it is compositional. (There is, of course, the logical possibility that also compositional forms may be stored.)

The fact that the format of lexical entries is restricted is important. Since the lexicon is considered to be the only source of variation in Nanosyntax, a constrained lexicon automatically entails a constrained theory of variation (Starke 2014b). Specifically, all cross-linguistic variation must be expressible as a variation in the size and shape of lexically stored trees (where size refers to the number of features/projections inside the entry, and shape refers to the specific structural configuration of these features).

Before we look into the details of lexical insertion, let me note that the post-syntactic nature of lexical insertion in Nanosyntax is an important feature that distinguishes between Nanosyntax and other models with phrasal/complex lexical entries, such as the Simpler Syntax (SS) model of Culicover and Jackendoff (2005). In this model, phrasal lexical entries are understood to be the atoms of syntax, and this leads to redundancies. To see that, consider the quote from Culicover and Jackendoff (2006:416), where the authors describe their idea as follows:

"SS enables storage of [...] complex structures with associated meanings. [...] Once pieces of syntactic structure can be stored in the lexicon associated with meanings, it is a simple step to store pieces of syntactic structure that have no inherent meaning [...] such as [(9)].

(9)



This piece of structure is equivalent to a traditional phrase structure rule VP → V–NP. Thus, it is possible to think of the lexicon as containing all the rules that permit syntactic combinatoriality. These are put to use directly in processing, as pieces available for constructing trees."

From this quote, we can see that while the possibility to store phrases is shared between Nanosyntax and the Simpler-Syntax model, there is a clear difference here. In Nanosyntax, it is impossible to conceive of the lexicon as containing rules that "permit syntactic combinatoriality." On the contrary, syntax operates according to its own rules and principles, merging one feature at a time. There is no way how a post-syntactic lexicon could *provide* a ready-made syntactic structure for direct use because the only thing that the Nanosyntactic lexicon does is that it links the outputs of syntax to their phonological and/or conceptual representation. The lexicon in Nanosyntax thus remains a passive list, consulted only during the mapping from syntax to PF/CF.

As the last point in this section, let me mention that the matching between lexicon and syntax is based on identity. A particular lexical entry only matches a given structure if it contains a piece that is identical to this structure. This is the content of the insertion principle in (10), which is standardly referred to as the Superset Principle.

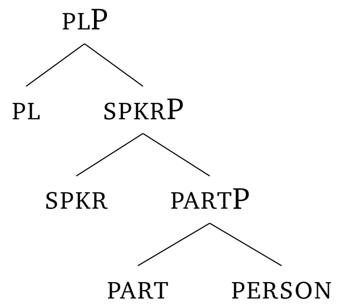
(10) The Superset Principle (Starke 2009)
A lexically stored tree matches a syntactic node iff the lexically stored tree contains the syntactic node.

5 Syncretism as an alternative to language-particular features

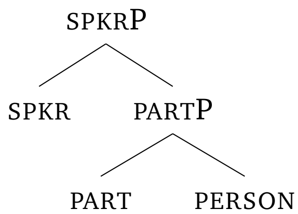
In this section, I return to Chomsky's idea that each language may only uses a subset of the features provided by UG, recall (1b). My goal is to show how Nanosyntax treats the relevant cases by relying on the postsyntactic lexicon only (keeping the feature structures constant across languages).

In order to work with a specific example, I focus on personal pronouns like *I, you, we,* etc., drawing on Vanden Wyngaerd (2018). The first thing that we must put in place are the features that these pronouns have. Following Vanden Wyngaerd (2018) (cf. Harley and Ritter 2002), I shall be using here the features PERSON, PARTICIPANT and SPEAKER.[5] In addition, I will be using the privative feature PL. When all these features are present in the structure, as in (11), we get the first person plural structure. When the PL feature is missing, we get the first person singular, see (12).

(11)

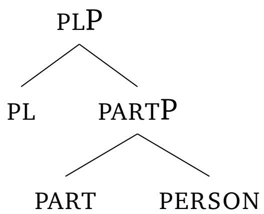


(12)

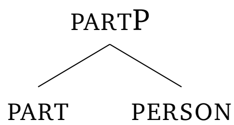


(13) gives the second person plural, (14) corresponds to the second person singular.

(13)



(14)



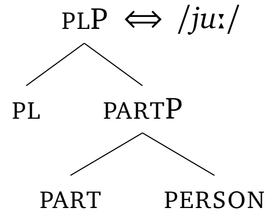
Note that the interpretation of these structures has a 'Gricean' component, in that the structure in (14) only says that we are looking at a participant in the discourse (which could be either the speaker or the addressee). However, since syntax has a special feature for the speaker, and since the feature is not used in (14), the structure is interpreted as referring to the addressee. Similarly, even though the structure (14) is unspecified for number, it is interpreted as singular, since the plural feature is absent.[6]

[5]Vanden Wyngaerd (2018) calls these features 3, 2 and 1, but this is just a matter of labelling conventions.

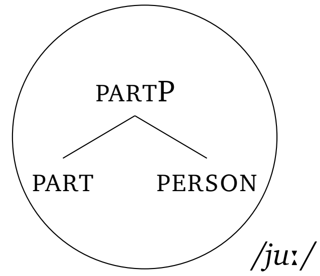
[6]There are also proposals according to which the plural is the number-neutral form, and the singular is semantically marked Sauerland et al. (2005). I do not try to resolve this tension here and follow Vanden Wyngaerd's (2018) approach, where the singular is unmarked.

Now, when a child encounters the expression *you* in the function of the 2nd person plural pronoun, she will associate the structure of the 2nd person plural with the relevant sound, and she will store this association in the lexicon as in (15).

(15)



(16)

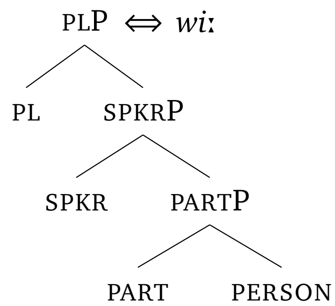


Suppose now that syntax builds a structure like (16). At spellout, the phrase must be linked to a PF representation. The lexicon is therefore searched with the goal to find a structure that is identical to this constituent. Since this constituent is found inside the lexical item (15), the structure can be linked to the phonology associated with the lexical entry. This is the content of the Superset Principle, recall (10).

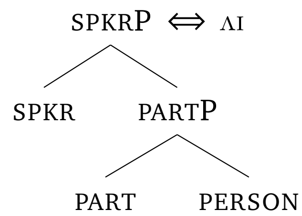
In (16), I depict by a circle the fact that this structure can be lexicalised by (15), which associates this structure to */juː/*. As a result, syncretism between the second person plural and the second person singular arises.

In cases where there is no syncretism, like in the first person, the lexicon must contain two lexical entries as in (17) and (18).

(17)



(18)



Notice that both entries match the first person singular, because both entries contain its structure. If there are two candidates for spelling out a particular structure, competition arises. In this competition, /ʌɪ/ wins, because it is a better match. (Competition among multiple matching entries, resolved by 'best fit' is a standard part of Late insertion theories.)

This mechanism allows for an elegant and restrictive account of syncretism, as argued for pronouns in Vanden Wyngaerd's (2018) article, and as further demonstrated in various strands of research within Nanosyntax (Caha 2009, 2013, 2017, 2021, Pantcheva 2010, De Clercq 2013, Lander and Haegeman 2016, Taraldsen 2017, Baunaz and Lander 2018a,b, Phan and Duffield 2019, Taraldsen Medová and Wiland 2019).

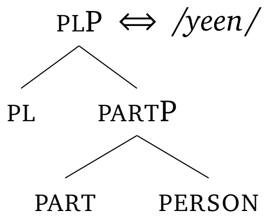
For the present chapter, the main interest of looking at syncretism is to show how we can use it to model variation. One aspect of this variation is trivial. Consider, for instance, the following dataset from Dakar Wolof (from Vanden Wyngaerd 2018).

(19) Dakar Wolof pronouns

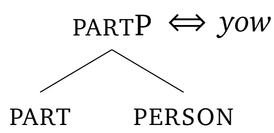
	SG	PL
1	man	ñum
2	yow	yeen
3	moon	ñoom

The difference from English is that there is no syncretism in the second person between the singular and the plural. This is because a child growing up in a Wolof-speaking environment will have enough evidence to postulate (in addition to the 2nd person plural entry) an independent entry for the singular, constructing a pair of entries as in (20) and (21).

(20)



(21)



The lexical entry (20) could in principle also pronounce the second person singular. However, since there is a better-fitting entry (namely (21)), this better-fitting entry is used. As a result, Wolof has no syncretism between the second person singular and plural.

While this type of variation appears trivial, interesting results emerge when we push it to the extreme. Suppose, for instance, that a language would only have pronouns like the English *you*, i.e., lacking a singular-specific counterpart in the lexicon. I.e., in all persons, we would get the singular-plural syncretism because there would never be any better-fitting competitior.

A language like that is Salt-Yui, where all independent pronouns are ambiguous between the singular and plural, see (22). (The third person uses demonstratives, which are also identical across the two numbers.)

(22) Salt-Yui (Cysouw 2009:116)

	SG	PL
1	na	na
2	ni	ni
3	DEM	DEM

Such a language-wide syncretism is sometimes referred to as 'absolute' syncretism or 'meta-syncretism' (Calabrese 2008, Harley 2008).[7] One way to analyse such a language is to say that the language lacks the relevant feature (plural in the case of (22)), since the language does not show any overt evidence for it. The idea is that since the child has no evidence

[7] Salt-Yui happens to have other means of distinguishing singular and plural reference. I am not reflecting on this in any detail, since my point is going to be independent of this.

for that feature, the feature will not be selected into the set [F] of features relevant for that particular language (where, recall, [F] is a subset of the features made available by UG). In other words, we can capture this by varying the inventory of features.

Nanosyntax, however, prefers not to go down this path (see, e.g., the discussion in Caha 2009:109ff). Rather, it accounts for paradigms such as (22) by pushing the notion of syncretism to its extreme. Under this approach, Salt-Yui does have a PL feature, and the pronouns in the right column of (22) spell it out. The absence of singular-specific forms in the singular column is analysed simply as an instance of syncretism: each form in (22) is syncretic between singular and plural in the same way as *you*. Under this approach, the paradigm in (22) represents just one (expected) logical option among many others as to how language-specific lexicons may reflect a universal set of features. This, then, is one concrete example how the postsyntactic lexicon allows us to capture cross-linguistic variation without proposing different feature inventories.

6 Tree size as an alternative to bundling

Let me now turn to the issue of presyntactic bundles proposed in standard minimalism as a way to handle certain aspects of cross-linguistic variation. The point of this section is to show how Nanosyntax replaces presyntactic bundling (which is unavailable in this theory) by the idea that lexical trees vary in their size.

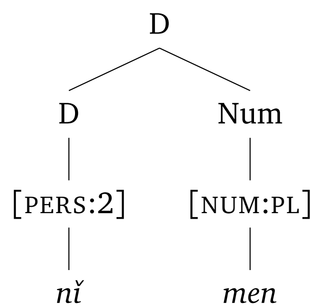
To have a concrete example in hand, consider the Mandarin data in (23) (as discussed in Vanden Wyngaerd 2018).

(23) Mandarin pronouns

	SG	PL
1	**wǒ**	**wǒ**-men
2	**nǐ**	**nǐ**-men
3	**tā**	**tā**-men

There is an obvious difference between the paradigms we have seen so far and Mandarin. Specifically, the Mandarin plural has two morphemes, one expressing person and another number (plural). On the other hand, English and Dakar Wolof have just one morpheme in the plural, expressing both person and number. One could conclude here that Mandarin (24) has two syntactic heads (and two positions of exponence), while English (25) has just one head:

(24)



(25) D

[PERS:2, NUM:PL]

we

In standard Minimalism, the differing structures would be a consequence of different groupings of features in LEX. A classical example of such work within the minimalist program is the paper by Bobaljik and Thráinsson (1998).

In Nanosyntax, the contrast between English and Mandarin can be captured by proposing that Mandarin person markers are lexically 'small.' Specifically, if no pronominal person marker in Mandarin spells out the plural feature in addition to person features, a paradigm like (25) emerges. In (26) and (27), I show the lexical entries for the first and second person markers respectively. Note that none of them contains the plural feature. This is unlike the English *you* in (15) and the Wolof *yeen* in (20).)

(26) SPKRP $\Leftrightarrow$ / *wǒ* /

SPKR PARTP

PART PERSON

(27) PARTP $\Leftrightarrow$ /*nǐ*/

PART PERSON

The plural in Mandarin is expressed by a separate lexical entry that Vanden Wyngaerd (2018) depicts as in (28).

(28) PLP $\Leftrightarrow$ /*men*/

PL

When embedded within the general theory of spellout assumed in Nanosyntax (due to Starke (2018)), the postsyntactic lexical entries as given above are all that one needs to account for the difference between English/Wolof and Mandarin. In order to show how this works in detail, we must look now at the details of the Nanosyntax spellout procedure.

As I will be introducing this procedure, it should be kept in mind that the spellout procedure is language invariant. However, as an integral part of its operation, the procedure interacts with the postsyntactic (language-specific) lexicon. As a result, it leads to different outputs in different languages, despite the fact that (i) the atoms of syntax are the same across languages; (ii) the operations of narrow syntax are constrained by the same principles across languages and (iii) the spellout procedure itself is also language-invariant.

The first idea that we need to put in place is that spellout is cyclic, applying potentially many times during a single derivation (cf. Uriagereka 1999). In Nanosyntax, spellout in fact happens after every step of external Merge. More specifically, every time external Merge applies – merging a new feature F and forming an FP – the FP must be lexicalised, else the derivation crashes. 'Be lexicalised' in this context does not mean directly 'be pronounced;' it means that a matching item for the FP created by external merge must be found.

If the lexicon finds a matching item for the FP, this means that the derivation can be mapped onto a PF/CF representation (it can be externalised). From there, the derivation may follow one of two routes. It either terminates, in which case the phonology and concepts are sent to PF and CF, following the downward arrows leading from the Spellout box in Figure 1. Alternatively, the derivation continues by further Merge (with its pronunciation delayed until it terminates). In this case, the structure (which had converged at the interface) is fed back to syntax via the cyclic feedback loop that leads from the Spellout box back to Syntax, see Figure 1.

However, if matching fails, this means that the derivation cannot be externalised and it must be rescued. Rescue in this context means that syntax performs various types of movement in order to change the configuration and allow for the lexicalisation of FP to succeed. The precise series of rescue steps is given in (29). Keep in mind that 'spell out FP' in the definitions in (29) means 'match FP by a lexical item.'[8]

(29) Spellout Algorithm (based on Starke 2018)

a. Merge F and spell out FP
b. If (a) fails, try moving the Spec of F's complement and spell out FP
c. If (b) fails, move the complement of F and spell out FP

It is important to keep in mind that the movements in (29) are not regular feature-driven movements. Their motivation is not to create a new interpretation (e.g., different scope); the goal is to create a new spellout configuration. Because of their motivation, the movements triggered by (29) are called spellout-driven movements, and they are expected to have different properties from feature-driven movements. For feature-driven movement in Nanosyntax, see De Clercq (2019).

It is also worth noting that the movements are performed by syntax, and not by PF. The idea is that when the derivation crashes, the lexicalisation procedure in (29) sends the derivation back to syntax, and it is syntax who has to perform the movement. This is again a part of the cyclic feedback loop in Figure 1.

In order to see in detail why spellout-driven movement is a syntactic movement (rather than a postsyntactic movement), it may be instructive to compare spellout movements to features like the EPP in standard minimalism. The movement-triggering EPP feature is standardly associated to particular elements of LEX *before syntax*, but this does not mean that the EPP triggers "presyntactic" movement. Rather, the conclusion is that while the EPP is a presyntactic trigger, it is syntax who performs the movement and satisfies the EPP.

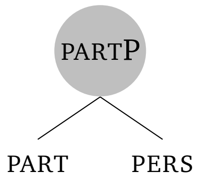
In the case of the spellout algorithm (29), the situation is similar. The spellout algorithm

[8] In its content, the algorithm is taken from Starke (2018). I have changed the wording slightly to (hopefully) facilitate understanding.

serves as the postsyntactic trigger for movement, but the movement itself is performed by syntax. Specifically, the movement takes place after the derivation is shipped into the Syntax box in Figure 1 using the feedback loop.[9]

With this issue clarified, let me now illustrate the working of the algorithm on the example of the Mandarin, Wolof and English second person plural pronouns. In all of these languages, the derivation begins by merging PERSON and PARTICIPANT, yielding (30).

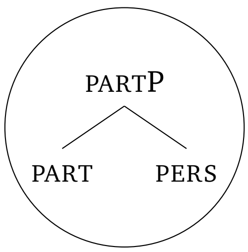
(30)



Immediately after PARTP is formed, it must be spelled out. This means that the highlighted node must be matched against a lexical entry that contains the exact same node. This is the content of (29a).

In all three languages, a matching item is found, and spellout succeeds. For English, the relevant entry is *you* in (15); for Wolof, see the perfectly matching entry (21); and finally for Mandarin, see (again) the perfectly matching (27). The result is that in all of these languages, the structure (30) is spelled out following the 'direct spellout' clause of the algorithm in (29a). Therefore, the only difference among the languages is of the trivial type, such that different languages have different phonology associated to this structure.

(31)

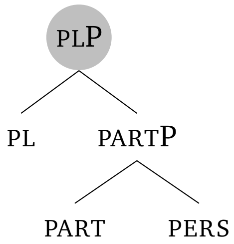


ENG: *you*
WOL: *yow*
MAN: *nǐ*

Since the PARTP node found a match in the lexicon, the derivation converges. At this point, the derivation may either terminate or continue by further Merge. If it terminates, it would be pronounced as indicated in (31). However, if more features are to be added, the derivation is not pronounced as yet, but sent back to Syntax using the feedback loop. Suppose the feature PL is merged on top of (31), producing (32). Once again, after external Merge takes place, spellout takes place, and the node created by external Merge must be matched against a lexical entry. The relevant node is highlighted in (32).

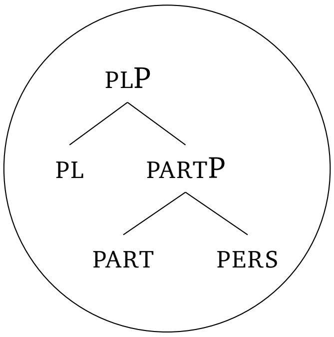
[9] It should be noted that comparing spellout-driven movements to EPP-driven movements does not mean that Nanosyntax relies on EPP features to effect such movements. Rather, Starke's (2014b) paper suggests that spellout movements may in fact serve as a potential replacement for EPP-type of movements, even though it remains to be seen whether this is possible to achieve in the full range of relevant cases.

(32)



At this point, the three languages part ways. English uses the same item as before to spell out this structure, yielding syncretism. Wolof now uses the rule (20) instead of the original winner (21). Despite this difference (syncretism in English, no syncretism in Wolof), the languages pattern alike in that they have one marker for all three features, see (33).[10]

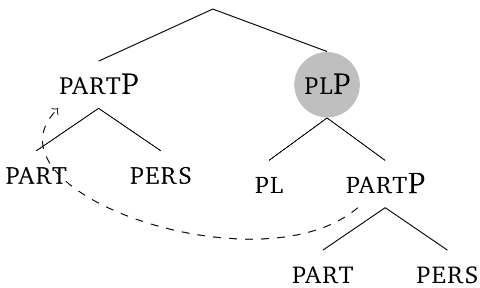
(33)



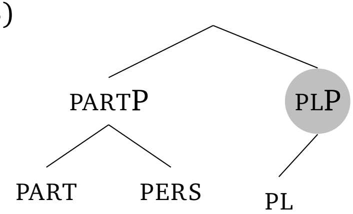
ENG: *you*
WOL: *yeen*
MAN: —

However, Mandarin has no matching item. Therefore, the spellout algorithm has to activate rescue movements. In the algorithm (29), the first rescue movement type is the so-called Spec-movement, see (29b). However, there is no specifier in (32), and this option therefore does not lead to any change. As a result, the spellout option (29c) is activated in Mandarin. Activating this option triggers the movement in (34), which ultimately produces the structure (35) for lexicalisation. The PLP node we need to spell out is highlighted throughout. It is the same node as in (32).

(34)



(35)

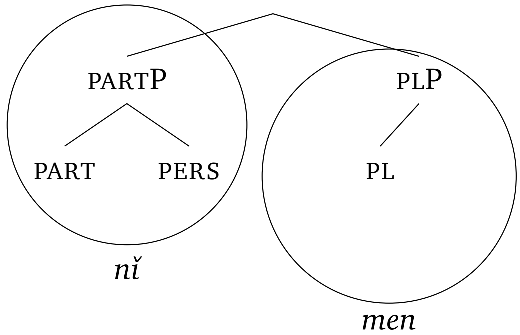


Note that in (35), there is no trace of the moved constituent. This is in accordance with Starke (2018), who assumes that spellout movements either do not leave a trace, or that the trace is ignored by matching. Regardless of which of these options is correct, the PLP node in (35) is matched by the plural marker *men*, recall (28). Therefore, matching is

[10]The previous matches at the level of PARTP are lost at this point, only the topmost match survives. This is called ‘Cyclic Override’ in Nanosyntax. Cyclic override is the consequence of the spellout algorithm (29), which requires that the topmost node is spelled out.

successful, and the relevant markers spell out the nodes circled in (36). This tree correctly captures the agglutinative structure of the second person plural in Mandarin, including the linear order.

(36)



For current work in Nanosyntax that exemplifies (and elaborates on) the spellout algorithm, see Baunaz and Lander (2018a), Bergsma (2019), Blix (2021), Caha (2021, to appear), Caha et al. (2019), De Clercq (2019, 2020), De Clercq and Vanden Wyngaerd (2017, 2018, 2019), Kloudová (2020), Taraldsen (2019), Taraldsen et al. (2018), Türk (2020), Vanden Wyngaerd et al. (2020), Wągiel and Caha (2020), Wiland (2018, 2019), Ziková (2018).

For reasons of space, it is impossible to illustrate here all the various derivational options that such an algorithm offers, and I am also leaving out some derivational options out of the discussion here (namely Backtracking and Spec-formation, for which see the works cited above). The point of discussing Mandarin was rather to show that it is possible to start from language-invariant atoms, combine these atoms in a language-invariant narrow syntax based on binary Merge, and still end up with different structures. Specifically, what we have seen is that Mandarin second person plural pronoun has the structure (36), while English and Wolof have the structure in (33).

Crucially, these difference do not come about as a result of language particular spellout rules: the spellout procedure is the same for Wolof, English and Mandarin, recall (29). The only thing that differs are the lexical items. But since the spellout procedure crucially relies on the lexical entries for matching, the lexical items apparently control how the derivation unfolds. Due to their ability to do this, the lexical items become the triggers of structural parametric differences.

7 Conclusions

The idea that lexical items are the loci of parametric variation has been independently known in the minimalist literature as the so-called Borer-Chomsky Conjecture. Baker (2008:353) states it as follows:

(37) *The Borer-Chomsky conjecture*

> All parameters of variation are attributable to differences in the features of particular items (e.g., the functional heads) in the lexicon.

While attractive on its own, a difficulty related to the Borer-Chomsky conjecture is that it apparently clashes with another attractive idea, namely that the insertion of phonology and conceptual information happens late (recall Section 3). The reason for the clash is the fact Late-Insertion entails a postsyntactic lexicon, but such a lexicon seems to come too late to be able to influence how syntax operates. We are thus left in a paradoxical situation where one set of ideas leads us to propose a postsyntactic (language-particular) lexicon, while another set of ideas leads us to place the (language-particular) lexicon before syntax.

As a result, some Late-Insertion models such as Distributed Morphology actually rely on 'two lexicons' in the sense of two different language-particular lists: they contain both a set of postsyntactic Vocabulary Items that supply phonology and concepts, and they also contain a language-particular presyntactic LEX with language-particular feature bundles. This allows such models to implement both modularity and the Borer-Chomsky conjecture, but it comes at the cost of an apparent doubling of the lexicons. Proposing two lexicons is clearly a way out of the conundrum, yet it seems to institutionalise the paradox rather than resolve it.

In the context of these considerations, the interest of the Nanosyntactic cyclic spellout system as described above is that it makes the Borer-Chomsky conjecture compatible both with Late-Insertion – and with a single postsyntactic lexicon. Moreover, the Nanosyntactic lexicon is constrained in a way that most lexicons are not, namely, it only contains well-formed syntactic structures (recall section 4). These lexical items interact with the spellout procedure (described in Section 6) in a way that different syntactic structures arise in different languages.

As such, the system comes rather close to implementing the Borer-Chomsky conjecture in a way originally envisioned by Borer (1984:2-3): "It is a desirable step forward to try and restrict the class of possible parameters. The strongest claim in this respect would be that there are no language-particular choices with respect to the realization of universal processes and principles. Rather, interlanguage variation would be restricted to the idiosyncratic properties of lexical items. These idiosyncracies, which are clearly learned, would then interact with general principles of UG in a particular way. This interaction would result in vastly different systems."

References

Baker, Mark C. 2008. The macroparameter in a microparametric world. In *The Limits of Syntactic Variation*, edited by Theresa Biberauer, pp. 351–373. John Benjamins, Amsterdam.

Baunaz, Lena and Eric Lander. 2018a. Deconstructing categories syncretic with the nominal complementizer. *Glossa: a journal of general linguistics* 3 1: 31.1–27.

Baunaz, Lena and Eric Lander. 2018b. Syncretisms with the nominal complementizer. *Studia Linguistica* 72 3: 537–570.

Bergsma, Fenna. 2019. Mismatches in free relatives–grafting nanosyntactic trees. *Glossa: a journal of general linguistics* 4 1: 119.1–37.

Blix, Hagen. 2021. Phrasal spellout and partial overwrite: On an alternative to backtracking. Ms., NYU, lingbuzz/005793.

Bobaljik, Jonathan David and Höskuldur Thráinsson. 1998. Two heads aren't always better than one. *Syntax* 1 1: 37–71.

Borer, Hagit. 1984. *Parametric syntax*. Dordrecht: Foris.

Caha, Pavel. 2007. Case movement in PPs. In *Nordlyd, special issue on Space, Motion, and Result*, edited by Monika Bašić, Marina Pantcheva, Minjeong Son, and Peter Svenonius, vol. 34 of *Tromsø Working Papers on Language and Linguistics*, pp. 239–299. CASTL, University of Tromsø, Tromsø.

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

Caha, Pavel. 2010. The parameters of case marking and spell-out driven movement. In *Linguistic Variation Yearbook*, edited by Jeroen Van Craenenbroeck, 10, pp. 32–77. John Benjamins, Amsterdam.

Caha, Pavel. 2013. Explaining the structure of case paradigms by the mechanisms of nanosyntax. *Natural Language & Linguistic Theory* 31: 1015–1066.

Caha, Pavel. 2017. How (not) to derive a *ABA: The case of Blansitt's generalisation. *Glossa: a journal of general linguistics* 2 1: 84.

Caha, Pavel. 2021. Modeling declensions without declension features. The case of Russian. *Acta Linguistica Academica* 67.

Caha, Pavel. to appear. *The nanosyntax of case competition. A study of numeral phrases in Ossetic and Russian*. Language Science Press.

Caha, Pavel, Karen De Clercq, and Guido Vanden Wyngaerd. 2019. The fine structure of the comparative. *Studia Linguistica* 73 3: 470–521.

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. *The Minimalist Program*. Cambridge, Massachusetts: MIT Press.

Chomsky, Noam. 2004. Beyond explanatory adequacy. In *Structures and Beyond*, edited by Adriana Belletti, no. 3 in The cartography of syntactic structures, pp. 104–131. 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.

Collins, Chris and Richard Kayne. 2021. Towards a theory of morphology as syntax. Ms., NYU, lingbuzz/005693.

Culicover, Peter W and Ray Jackendoff. 2005. *Simpler Syntax*. Oxford University Press, Oxford.

Culicover, Peter W and Ray Jackendoff. 2006. The simpler syntax hypothesis. *Trends in cognitive sciences* 10 9: 413–418.

Cysouw, Michael. 2009. *The Paradigmatic Structure of Person Marking.*. Oxford University Press, New York.

De Clercq, Karen. 2013. *A Unified Syntax of Negation*. Phd dissertation, University of Ghent, Ghent.

De Clercq, Karen. 2019. French negation, the superset principle 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. A Nanosyntactic Account*. Mouton de Gruyter.

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 analytic 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.

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.

Fábregas, Antonio. 2009. An argument for phrasal spell-out: indefinites and interrogatives in Spanish. In *Nordlyd 36: Special issue on Nanosyntax*, edited by Peter Svenonius, Gillian Ramchand, Michal Starke, and Knut Tarald Taraldsen, pp. 129–168. CASTL, Tromsø.

Fodor, Jerry A. 1983. *The modularity of mind : an essay on faculty psychology*. MIT Press, Cambridge, Mass.

Giorgi, Alessandra and Fabio Pianesi. 1996. Verb movement in Italian and syncretic categories. *Probus* 8: 137–160.

Harbour, Daniel. 2011. Valence and atomic number. *Linguistic inquiry* 42 4: 561–594.

Harley, Heidi. 2008. When is a syncretism more than a syncretism? In *Phi Theory*, edited by Daniel Harbour, David Adger, and Susana Béjar, pp. 251–294. Oxford University Press, Oxford.

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

Hegarty, Michael. 2005. *A feature-based syntax of functional categories: The structure, ac-*

quisition, and specific impairment of functional systems. Walter de Gruyter, Berlin, New York.

Kayne, Richard. 2005. *Movement and Silence*. Oxford University Press, Oxford.

Kloudová, Veronika. 2020. *The Interaction of Functional Morphemes inside the Nominal Phrase*. LINCOM GmbH, München.

Lander, Eric and Liliane Haegeman. 2016. The nanosyntax of spatial deixis. *Studia Linguistica* pp. 1–66.

Marantz, Alec. 1994. A late note on late insertion. In *Explorations in generative grammar: A festschrift for Dong-Whee Yang*, edited by Young Sun Kim, Byung-Choon Lee, Kyoung-Jae Lee, Kyun-Kwon Yang, and Jong-Kuri Yoon, pp. 396–413. Hankuk, Seoul.

Marantz, Alec. 1996. Cat as a phrasal idiom: consequences of late insertion in Distributed Morphology. Ms., MIT.

Miller, Philip, Geoffrey Pullum, and Arnold Zwicky. 1997. The principle of phonology-free syntax: four apparent counterexamples in French. *Journal of Linguistics* 33: 67–90.

Pantcheva, Marina. 2009. Directional expressions cross-linguistically. Nanosyntax and lexicalization. In *Nordlyd 36: Special issue on Nanosyntax*, edited by Peter Svenonius, Gillian Ramchand, Michal Starke, and Knut Tarald Taraldsen, pp. 7–39. CASTL, Tromsø.

Pantcheva, Marina. 2010. The syntactic structure of locations, goals and sources. *Linguistics* 48 5: 1043–1081.

Pantcheva, Marina. 2011. *Decomposing Path: The Nanosyntax of Directional Expressions*. Ph.D. thesis, University of Tromsø, Tromsø.

Phan, Trang and Nigel Duffield. 2019. A more perfect unification: Exploring a nanosyntactic solution to vietnamese đã. In *Interdisciplinary Perspectives on Vietnamese Linguistics*, edited by Nigel Duffield, Trang Phan, and Tue Trinh, pp. 69–80. John Benjamins, Amsterdam.

Sauerland, Uli, Jan Anderssen, and Kazuko Yatsushiro. 2005. The plural is semantically unmarked. In *Linguistic evidence*, edited by Stephen Kepser and Marga Reis, pp. 413–434. De Gruyter Mouton.

Starke, Michal. 2009. Nanosyntax. A short primer to a new approach to language. In *Nordlyd 36: Special issue on Nanosyntax*, edited by Peter Svenonius, Gillian Ramchand, Michal Starke, and Tarald Taraldsen, pp. 1–6. University of Tromsø, Tromsø.

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

Starke, Michal. 2014b. Towards elegant parameters: Language variation reduces to the size of lexically-stored trees. In *Linguistic Variation in the Minimalist Framework*, edited by M. Carme Picallo, pp. 140–152. Oxford University Press, Oxford.

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.

Taraldsen, Knut Tarald. 2009. Lexicalizing number and gender in Lunigiana. In *Nordlyd*

36: Special issue on Nanosyntax, edited by Peter Svenonius, Gillian Ramchand, Michal Starke, and Knut Tarald Taraldsen, pp. 113–127. CASTL, Tromsø.

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

Taraldsen, Knut Tarald. 2017. A recalcitrant syncretism. *Rivista di Grammatica Generativa — Research in Generative Grammar* 39 1: 3.1–8.

Taraldsen, Knut Tarald. 2019. An introduction to Nanosyntax. *Linguistics Vanguard* 5 1.

Taraldsen, Knut Tarald, Lucie Taraldsen Medová, and David Langa. 2018. Class prefixes as specifiers in Southern Bantu. *Natural Language & Linguistic Theory* pp. 1–56.

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

Türk, Utku. 2020. Tackling the augmentative puzzle in Turkish. In *Proceedings of the Workshop on Turkic and Languages in Contact with Turkic*, vol. 5, pp. 155–162.

Uriagereka, Juan. 1999. Multiple spell-out. In *Working minimalism*, edited by Samuel Epstein and Norbert Hornstein, pp. 251–282. MIT Press, Cambridge, Massachusetts.

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: 23.

Wągiel, Marcin and Pavel Caha. 2020. Universal semantic features and the typology of cardinal numerals. *Catalan Journal of Linguistics* 19: 199–229.

Wiland, Bartosz. 2018. A note on lexicalizing 'what' and 'who' in Russian and in Polish. *Poznan Studies in Contemporary Linguistics* 54 4: 573–604.

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

Ziková, Markéta. 2018. *Licensing of Vowel Length in Czech. The Syntax-Phonology Interface*. Peter Lang, Berlin.

Zwicky, Arnold. 1969. Phonological constraints in syntactic descriptions. *Papers in Linguistics* 1: 411–453.

Zwicky, Arnold and Geoffrey Pullum. 1986. The principle of phonology-free syntax: introductory remarks. *Working papers in Linguistics* 32: 63–91.