Годишњак Филозофског факултета у Новом Саду, Књига XLVII-3 (2022)
Annual Review of the Faculty of Philosophy, Novi Sad, Volume XLVII-3 (2022)

Hayley Ross[*]
Harvard University

UDC: 811.113.4'367
DOI: 10.19090/gff.2022.3.15-41
Originalni naučni rad

IMPLICATIONS OF THE DANISH DEFINITENESS ALTERNATION FOR CONCORD IN NANOSYNTAX[**]

The Danish definiteness alternation presents two challenges for Nanosyntax. First, it displays structural allomorphy of the definiteness marker between a suffix and prenominal article; second, there is concord between the definiteness marker and noun gender. I show that Nanosyntax can address both issues, explaining the suffix-article alternation by virtue of its spellout algorithm and the lexical overlap between suffix and article. This account provides a deeper explanation for the structural allomorphy than the Distributed Morphology analysis proposed by Hankamer & Mikkelsen (2018). The existing proposal for concord in Nanosyntax (Caha, 2019) cannot handle this combination of gender concord and allomorphy, so I propose a simple copying mechanism which handles concord more flexibly. This new proposal, however, is substantially less restrictive than Caha's framework, paving the way for future work to balance restrictiveness with empirical coverage of prefix/suffix alternations and concord across languages.

Keywords: Nanosyntax, Multiple Merge, Danish, definiteness, concord, agreement, morphology

INTRODUCTION

Balancing the restrictiveness and elegance of a formal theory with empirical coverage is a perennial issue for theories of morphology, especially for Nanosyntax (Caha, 2009; Starke, 2010), which aims to have a single, restrictive spellout algorithm to handle all derivations. The Danish Definiteness alternation (Delsing, 1993) poses a particular empirical challenge for the restrictive formulation of Nanosyntax in Caha (2019). Like other Scandinavian languages, Danish has two definiteness markers, a prenominal definite article and a definite suffix. In Danish,

[*] hayleyross@g.harvard.edu
[**] "Presented at SinFonIJA 14. I am grateful to the audiences at AIMM5, SinFonIJA 14, the Brno Nanolab and several anonymous reviewers for their feedback. I am particularly indebted to Pavel Caha for his insightful comments at my Nanolab presentation."

the two are in complementary distribution: the suffix occurs by default, but when modifiers such as adjectives are present, a prenominal article takes its place[1]:

(1)		kant-**en**	
		edge-DEF.SG.C	
		‘the edge’	
(2)		**den**	skarpe kant
		DEF.SG.C	sharp edge
		‘the sharp edge’	
(3)	*	**den**	kant
		DEF.SG.C	edge
		~ ‘the edge’	
(4)	*	skarpe	kant-**en**
		sharp	edge-DEF.SG.C
		~ ‘the sharp edge’	

Further, the definiteness marker shows concord with noun gender. Caha (2019) proposes the principle of Multiple Merge to handle case concord in Russian using Nanosyntax. This elegantly handles the multiple occurrence of case morphemes on noun and number by inserting the case feature both into the main spine of the derivation containing the noun, and also into the number ‘prefix’. I will show, however, that this account cannot extend to gender concord in the Danish definite noun phrase since it is unable to handle the allomorphy involved. Instead, the allomorphy can be captured by a less restrictive formulation of prefix building for Nanosyntax (Starke, 2018). The question then becomes how to handle concord in Danish without Multiple Merge.

Investigating how to handle concord in Nanosyntax sheds light on how agreement (typically thought of as feature copying or sharing) may be implemented in a cartographic theory that insists on a single head per feature, in contrast with theories such as Distributed Morphology (Halle & Marantz, 1993) which use feature bundles and simply inherit agreement from the broader syntax. We will see that a basic feature-copying approach in Nanosyntax can handle the Danish concord data and even extend to adjective agreement, at the expense of being less restrictive. We also gain an explanation of *why* Danish shows this structural allomorphy thanks

[1] In all glosses throughout the paper, I use DEF = definiteness, INDF = indefiniteness, SG = singular, C = common gender and N = neuter gender.

to the spellout algorithm’s preference for suffixes over prefixes; this is an improvement over the DM account proposed by Hankamer & Mikkelsen (2018) which relies largely on a ‘Sisterhood Condition’ describing the phenomenon in words (as opposed to explicit vocabulary insertion rules).

THE DANISH DATA

We saw in the introduction that Danish has two definiteness markers, a prenominal definite article and a definite suffix, in complementary distribution (in contrast with Swedish and Norwegian, where the two may co-occur). The alternation is typically illustrated with the following data, repeated from the introduction (Delsing, 1993; Hankamer & Mikkelsen, 2018):

(1) kant-**en**
edge-DEF.SG.C
‘the edge’

(2) **den** skarpe kant
DEF.SG.C sharp edge
‘the sharp edge’

(3) ***den** kant
DEF.SG.C edge
~ ‘the edge’

(4) *skarpe kant-**en**
sharp edge-DEF.SG.C
~ ‘the sharp edge’

This alternation is not limited to adjectives or to linear intervention between article and noun: Hankamer & Mikkelsen (2018) show that the same phenomenon occurs for restrictive vs. non-restrictive relative clauses, even though these occur to the right of the noun.

(5) **den** stol som jeg sad på
DEF chair that I sat on
‘the chair that I sat on’ [restrictive]

(6)	stol-**en**	som	jeg	sad	på
	chair-DEF	that	I	sat	on

‘the chair, which I sat on’ [non-restrictive][2]

This shows that the allomorphy between suffix and article must be structurally determined: if a modifier intervenes hierarchically in the tree between noun and definiteness marker (regardless of its linear position), the article must be used. My proposal will focus on the adjective case, but the theory developed here extends equally well to any XP inserted in the position proposed for the adjective, including relative clauses. (For the traditional motivations of why restrictive relative clauses are adjoined lower than non-restrictive ones, see Hankamer & Mikkelsen, 2018.)

Further, the forms of the definiteness markers are dependent on noun gender and number. Danish has two noun genders, common and neuter. For common nouns such as *kant* above, definiteness is marked by *-en/den,* while for neuter nouns such as *hus, -et/det* is used[3].

(7)	hus-**et**
	house-DEF.SG.N

‘the house’

(8)	**det**	store	hus
	DEF.SG.N	big	house

‘the big house’

In sum, the Danish data presents two main challenges: one, to handle the structurally motivated alternation between definiteness suffix and prenominal article, and two, to handle the concord between noun and definiteness marker. For Nanosyntax, which insists on one feature per head, expressing a feature such as neuter gender on both the noun and the definiteness marker requires multiple

[2] Hankamer & Mikkelsen (2005) note that for some speakers, (6) also admits a restrictive reading ‘the chair that I sat on’. Like Hankamer & Mikkelsen (2018), I will set aside this possibility of a restrictive reading, assuming that it occurs when the clause adjoins low enough to trigger restrictive semantics but too high to intervene and trigger the definite article.

[3] This paper focuses on the singular forms; see the final section for open questions surrounding the plural.

insertions of that feature. Caha (2019) proposes Multiple Merge to address this (see Section 3.2), but I will show that Multiple Merge cannot handle both the concord and the structural allomorphy.

A third desideratum is to explain the significant overlap between *-en/den* and *-et/det*, ideally by analysing the articles *den* and *det* as *d-en* and *d-et*, i.e. containing the definiteness suffix. (In fact, the final analysis will split them further into *d-e-n* and *d-e-t* in order to account for adjective agreement.) As we will see in Section 3 when discussing prefixes (in Nanosyntax, any material merged on the left), splitting *den/det* in this way will create a 'multi-morpheme prefix' which is only possible in Caha's Nanosyntax under very specific circumstances.

OVERVIEW OF NANOSYNTAX

We begin by reviewing the principles of Nanosyntax[4] as in Caha (2019).

Spellout Algorithm and Fundamental Principles

In Nanosyntax, the lexicon contains not bundles of features mapped to morphemes but rather small syntactic trees which "spell out" (correspond to) a particular morpheme. Further, Nanosyntax follows the principle of one head per feature. For example, a genitive morpheme such as Russian *-i*[5] maps not to a single genitive feature but to the tree [$_{\text{GenP}}$ GEN [$_{\text{AccP}}$ ACC [$_{\text{NomP}}$ NOM]]] containing the lower cases (Caha, 2020). The following two principles govern the *spellout*, i.e. mapping to the syntactic derivation, of these lexical entries:

(9) *The Superset Principle*
A lexically stored tree matches a syntactic node if and only if it contains the syntactic node.

(10) *The Elsewhere Condition*
When two entries can spell out a given node, the more specific entry wins. Under the Superset Principle governed insertion, the more specific entry is the one which has fewer unused features.

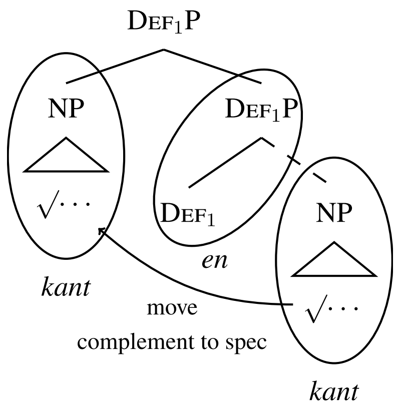
[4] For a conceptual introduction to Nanosyntax and motivations for choosing it over Distributed Morphology, the reader is referred to Baunaz & Lander (2018); for a detailed step-by-step account of the theory, to the excellent first few chapters of Caha (2019) itself.

[5] *-i* is the genitive singular suffix for declensions II and III; see Caha (2020) for the full paradigm. Details of how to handle declensions are omitted in this example.

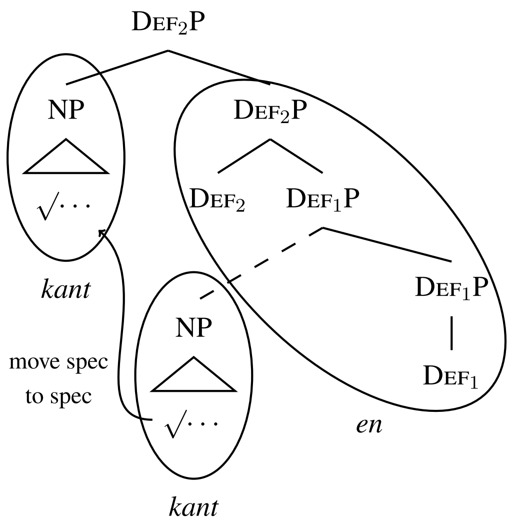
This means that a lexical entry can spell out a derivation at each step as the derivation "builds" the lexical entry, if there is no smaller competitor. Now, given a feature sequence (*fseq*) for a derivation, such as a noun root followed by case features, we merge the features (heads) one by one using the following algorithm (Caha, 2019, Chapter II.6).

(11) *The Spellout Algorithm*
Merge F and
(a) Spell out FP
(b) If (a) fails, move the spec of the complement of F and retry (a)
(c) If (b) fails, move the complement of F and retry (a)
(d) If (c) fails, backtrack to the previous cycle and try the next option for that cycle
(e) If (d) fails, try to spawn a new derivation providing F. Spell out F in the new workspace, then immediately close the new workspace by merging the FP to the main derivation, projecting the feature F to the top.

Steps (b) and (c) spell out F as a suffix, as shown in the following hypothetical examples (F is DEF$_1$ in (12), then DEF$_2$ in (13); suppose that *-en* is spelled out by [DEF$_2$ [DEF$_1$]]).

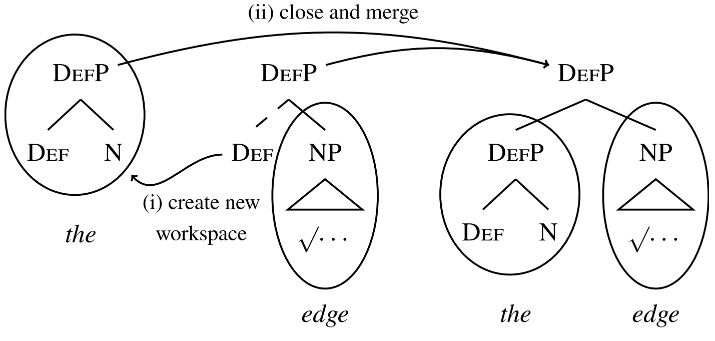
(12)


(13)



The remaining steps cover two important features. Step (d) lets the algorithm backtrack, which notably allows roots which have spelled out features too greedily (features required to spell out other morphemes higher in the tree) to be "shrunk" back down to rescue the derivation (see Caha (2020) for examples with Russian case). Step (e) allows the formation of prefixes, which in Nanosyntax refers to any material adjoined on the left. Caha's spellout algorithm is vague about the exact structure prefixes take; all they need is to have a binary foot [$_{FP}$ F X] ("the identity of X is left open on purpose"). For concreteness I'll follow Caha, De Clercq, & Vanden Wyngaerd (2019) where they take the form [$_{FP}$ F F^{-1}] where F^{-1} is the topmost feature in the main spine. This is shown in the hypothetical example in (14), where F is DEF and F^{-1} is N.

(14)



This encompasses both traditional prefixes as well as left modifiers such as adjectives to nouns or the *more* of *more intelligent* (Caha et al., 2019). Further, prefixes are a “last resort”, becoming an option only after all other movements and backtracking options have failed. This is justified by the high “cost” of opening a new workspace in the derivation and predicts empirically that there should be a preference for suffixation across and within languages.

Under this strict formulation of the algorithm, prefixes are composed only of a single feature in most cases – Caha argues that keeping a second workspace open is so expensive that we should close it immediately. (Having a binary foot does not make the prefix multi-feature; this is just the notation to make it a full constituent.) Prefixes with several heads are possible only if backtracking is triggered and re-opens the relevant prefix workspace. This strictness for prefix size contrasts with the position taken by Caha et al. (2019), where prefixes may be built with multiple heads (the prefix workspace need not be closed) as long as the prefix spells out a single morpheme. It contrasts further with the original proposal for prefixes by Starke (2018) which argues that the prefix workspace should be kept open as long as possible[6]. Starke also appeals to the cost argument: because the second workspace is so expensive, we should get maximal value out of it rather than closing it immediately. I will follow Caha’s approach for now, but we will see the benefits of Starke’s variant in Section 5.1.

Multiple Merge

Caha (2019) extends the spellout algorithm above with Multiple Merge, a principle which adjusts how backtracking interacts with prefix workspaces to permit the copying (multiple merging) of features and thus permit concord.

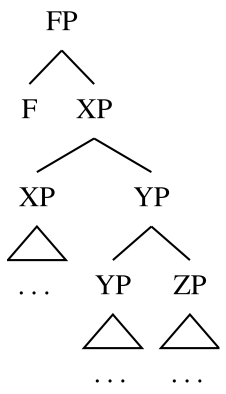
(15) *Multiple Merge*
When backtracking reopens multiple workspaces, merge F in each such workspace.

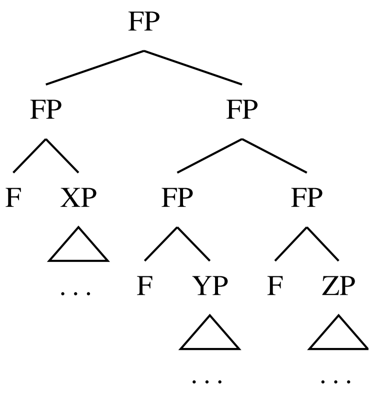
[6] In Starke’s example, “as long as possible” corresponds to the end of a lexical entry. It is unclear whether Starke would permit multiple lexical entries to inhabit the same prefix; this is explored in Section 5.1.

In the previous formulation of the spellout algorithm, if we were unable to merge F and the last step happened to have been prefix formation in an auxiliary workspace, we could backtrack, detach and re-open that workspace, and attempt to merge F with the prefix, creating a multi-feature prefix. Caha argues that there is a second option: once we have detached the auxiliary workspace, we could instead merge F in the main spine (and reattach the prefix afterwards). Multiple Merge proposes that we should try both and adopt whichever succeeds (possibly both). The two workspaces are then recomposed in the same configuration as before. Thus, there are three possible outcomes of Multiple Merge:

1. F is merged only in the prefix; spelling it out in the main spine fails. This is how we derive the configurations produced prior to introducing Multiple Merge, including multi-feature prefixes such as English *more* in Caha et al. (2019), as well as possibly multi-morpheme prefixes.
2. F is merged in both the prefix and the main spine. This is the outcome that yields concord, for example case concord between the German determiner and noun (e.g. *des Kind-es*, ‘the.GEN.SG child-GEN.SG’).
3. F is merged only in the main spine, failing to spell out in the prefix. In this case, features essentially “skip” the prefix. This is useful when features such as case should skip modifiers to the left of the noun such as German numerals (e.g. *den zwei Kinder-n*, ‘the.DAT.PL two children-DAT.PL’).

Further, this process may be recursive if we have a structure with multiple prefixes. Consider the schematic structure in (16) from Caha (2019, p. 204). If F cannot be spelled out by spec or complement movement, Multiple Merge opens not only the prefix XP but also the prefix YP when applied recursively to the main spine. This results in up to three copies of F, if each spellout of [$_{FP}$ F XP], [$_{FP}$ F YP] and [$_{FP}$ F ZP] is successful, as shown in (17).

(16)


(17)


Caha uses Multiple Merge to elegantly derive a particular phenomenon of Russian case marking: when a numeral phrase such as *pjat' stolov* 'five tables' is nominative, there is nominative case marking on the number and genitive on the noun. When the whole phrase is dative, dative case marking appears on both number and noun.

(18)	pjat'	stol-ov
	five.NOM	tables-GEN.PL
	'five tables'	
(19)	pjat'-i	stol-am
	five-DAT.SG	tables-DAT.PL
	'to five tables'	

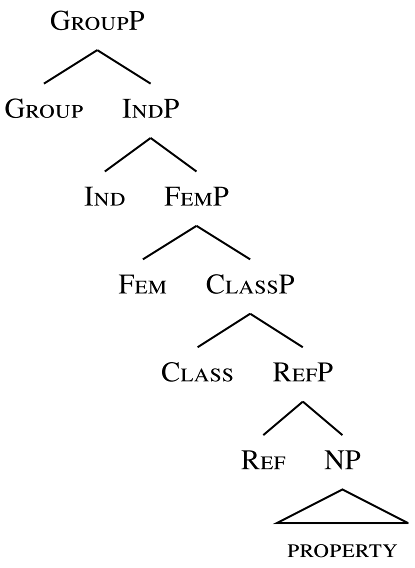
In combination with the hierarchy of cases (dative contains genitive, genitive contains accusative, accusative contains nominative; see Caha (2020) for discussion), Multiple Merge explains why the NOM feature only gets merged in the prefix *five*, since the main spine already contains GEN and we cannot spell out NOM on top of GEN. To spell out DAT, by comparison, we first merge every feature from NOM to GEN. These all spell out only in the number as before. When we add DAT, we may spell it out in both the prefix (number) and the main spine, because now both contain GEN.

Feature sequence for noun phrases

The final ingredient of Nanosyntax we need is an appropriate feature decomposition for noun phrases. I adapt the features for Russian proposed by Caha (2020), inherited from Harley & Ritter (2002), to Danish.

At the bottom of the tree in (20), we have the root for the noun in question[7]. REFP indicates that the noun phrase is referential. CLASSP indicates that it has a noun class, common by default; NEUTP is additionally present if the noun is neuter. INDP denotes that the noun phrase refers to an individual (not e.g. a mass noun) while GROUPP is additionally present if the noun is plural.

(20)



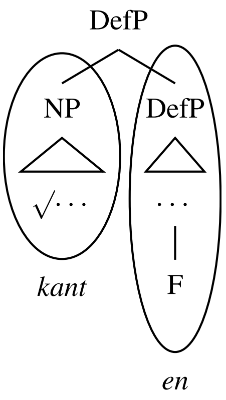
[7] See Caha et al. (2019) for a detailed discussion of the notion of root in Nanosyntax.

ANALYSIS OF DANISH DEFINITENESS WITH MULTIPLE MERGE

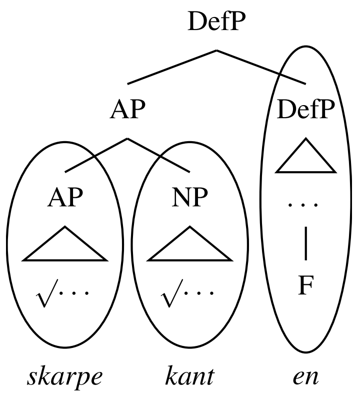
The Danish data clearly calls for an analysis with Multiple Merge: we have concord and we have prefixes (prenominal articles) containing several morphemes, if we take the *d-en/et* decomposition. Unfortunately, Multiple Merge is unable to handle the structural allomorphy where the adjective triggers the alternation between article and suffix, by the very nature of its permitting features to be merged into the main spine even when a prefix is present (outcome 3 above). To see why, suppose that we have a way of deriving the plain definite noun phrase *kant-en* 'the edge'. Suppose that some definiteness feature DEF spells out as *-en*, potentially in combination with other features. Call the feature that *-en* is footed in F. (F could be DEF itself in principle but will need to be a distinct feature for Multiple Merge to trigger.) The derivation of *kant-en* is given in (21); the features needed to derive *kant* itself are abbreviated under NP.

Suppose that we want to add the adjective *skarpe* 'sharp'. We begin with the NP for *kant*. Suppose that we next merge the AP as a prefix on the left. Then, we merge definiteness onto the whole phrase, starting with F. We know that F can spell out as suffix *-en*, because it does so in the derivation of *kant-en*. We add the remaining features and build up to DEF; each time, we can spell out as the suffix *-en*. Ultimately, we end up with *[skarpe kant]-en*, as in (22) – not what we wanted. The correct form is *den skarpe kant*. Moreover, we never triggered Multiple Merge.

(21)



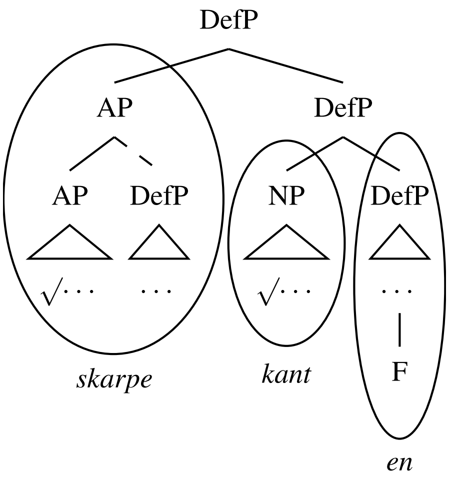
(22)



How do we invoke Multiple Merge? In the previous derivation, we implicitly assumed that F lies after the AP in the feature sequence. Suppose that F is instead merged before the AP (possibly with some other features for -*en*), but DEF is only merged after the AP. Merging F and its neighbours spells out as the suffix -*en*, as before. Next, we merge the AP. Now, when we try to merge DEF after the AP we cannot spell it out as -*en* because the AP intervenes between DEF and its desired foot F. This triggers backtracking – and thus Multiple Merge. We re-open the prefix workspace (the AP) and try to merge DEF there[8]. More importantly, we also open the main spine, which contains the NP and F, and merge DEF there as well. This succeeds: just as in *kant-en*, we have the sequence F through DEF in the main spine and may spell out as -*en*. Unfortunately, this means that when the workspaces are put back together, we still get *skarpe [kant-en]*, just with the new bracketing of (23).

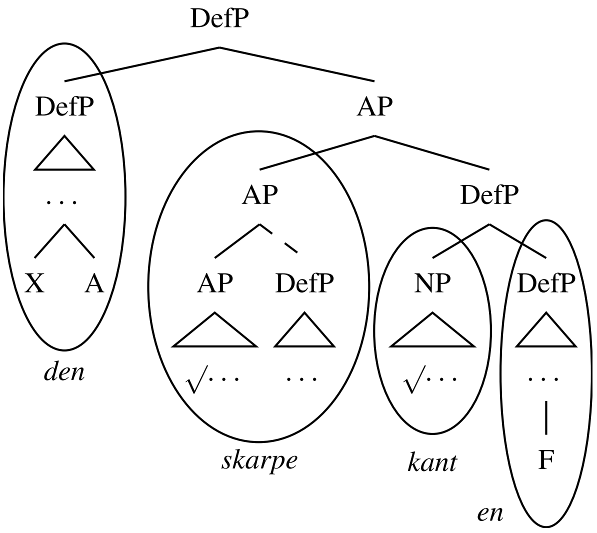
[8] This may succeed, yielding some kind of adjective agreement, or it may not; whether it succeeds does not matter for this argument.

(23)



None of these derivations have produced the article *den*. To host it, we would need an additional prefix in front of the adjective. Perhaps some feature X can be stipulated which is always merged after an adjective and creates this prefix, either spelling out as *den* immediately or phonologically null at first. Then, when DEF is merged, we re-open both prefixes just as in the schema in (17) and merge DEF in each as well as in the main spine. (DEF can't spell out on its own, by construction, so it triggers backtracking every time.) If the tree of X through DEF spells out as *den*, this creates a prenominal article as desired. Unfortunately, Multiple Merge still merges DEF with the main spine as well and spells it out as *-en* as before. So at best, we can derive *den skarpe kant-en*, shown in (24). While this is in fact the correct form for Norwegian and Swedish, it will not do for Danish.

(24)



Multiple Merge's express ability to skip prefixes, perfect for many other languages and situations, is precisely what leads to its downfall in Danish. In Danish, the presence of the adjective "prefix" is precisely triggering the allomorphy and should not be skipped.

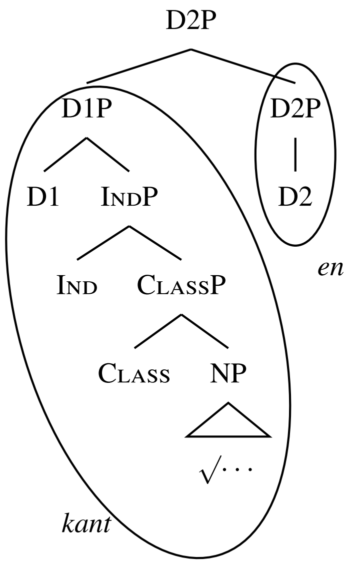
A LESS RESTRICTIVE ANALYSIS

The issues with Multiple Merge stem in part from its highly restrictive formulation. Multiple Merge only triggers when a feature cannot be spelled out by spec or complement movement, and when it does trigger, it always merges F to every workspace where F can be spelled out. While in general it is desirable in Nanosyntax to put the burden on the shape of the lexical entries and keep the spellout algorithm simple, this does not give us enough flexibility to handle the Danish data. I will step back to the original spellout algorithm in (11) (without Multiple Merge) and instead loosen the spellout process in two ways. The first will modify when prefixes are closed, and the second will involve the spellout of "placeholder" heads which initiate the copying necessary for concord without Multiple Merge.

Handling the structural allomorphy with late prefix closure

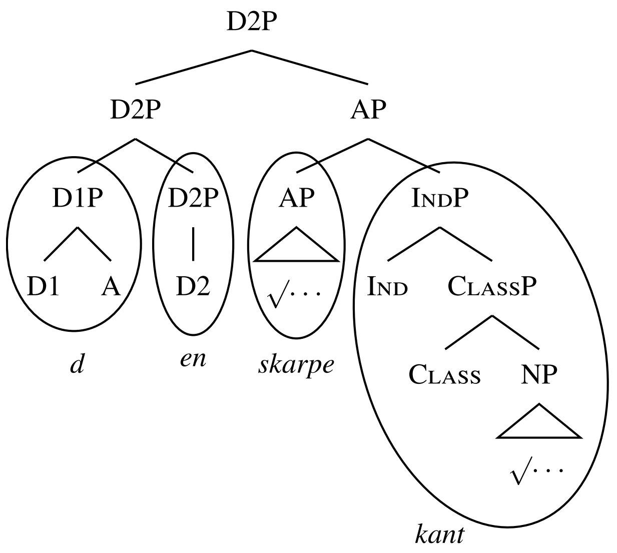
The first problem for the Multiple Merge analysis is building the prefix *den*, provisionally as *d*-*en*, where -*en* is the definite suffix. Under this decomposition, *d* can be viewed as existing to support -*en* in the prefix position, not unlike how *do*-support in English exists to permit tense in the correct position. To achieve this, I split the definiteness head into two parts, provisionally called D1 and D2, such that D2 spells out the suffix -*en*. D1 is normally spelled out as part of the noun, but becomes the prefix *d* and supports -*en* when an adjective is present. By just changing the lexical entries of nouns to contain D1P (optionally, thanks to the Superset Principle), *kant*-*en* is derived straightforwardly as follows:

(25)



Next, I stipulate that the AP merges precisely between IND and D1. This prevents D1 from being spelled out by the noun. Instead, I add a new lexical entry for D1P, a prefix which spells out as *d*. This has the standard prefix shape [$_{\text{D1P}}$ D1 A], as discussed in Section 3.1. We will use A for the binary foot, since D1P adjoins to AP. Now, when we come to spell out D2, there is a prefix workspace containing D1. I depart from the early prefix closure in the spellout algorithm of (11) and instead invoke the late prefix closure option of Starke (2018) which allows us to close prefixes "as late as possible". I will interpret this as allowing us to keep the prefix workspace of D1 open long enough to merge D2 as a suffix to *d*, creating *d-en* as in (26). This may be more generous than Starke intended: Starke's only example happens to close the prefix after spelling out one morpheme. Nonetheless, this proposal does not need workspaces to stay open indefinitely: while the notion of a word is not necessarily well defined by Nanosyntax, the prefix workspace here closes after it spells out a word as understood by Danish speakers; perhaps this is a restriction that can be placed on this process. (See Hankamer & Mikkelsen (2018) for a fascinating discussion of what wordhood might mean in this context.)

(26)



This proposal elegantly handles the structural allomorphy of the Danish definiteness alternation with just a decomposition of definiteness into two heads, plus some straightforward lexical items. Moreover, the way that the structural allomorphy works *explains* why *-en* and *den* overlap so much in form. This is a place where Nanosyntax shines: because all allomorphy in Nanosyntax is structural, it has no trouble handling the Danish allomorphy.

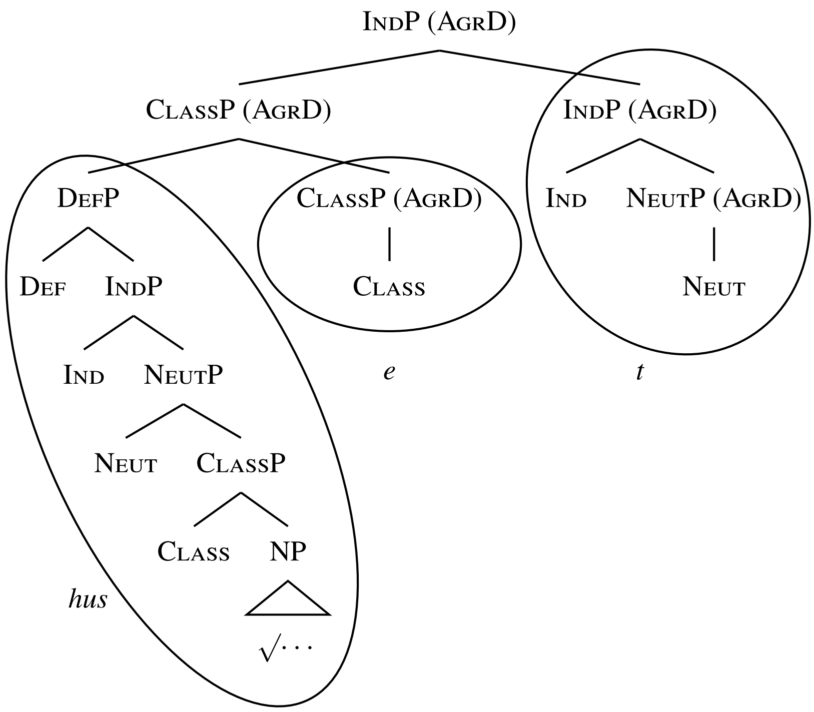
Adding concord

The above derivation only handles common gender: D2 always spells out as *-en*. To derive *hus-et* and *det store hus*, we need D2 to show concord with the noun. Concord in Nanosyntax necessarily means feature multiplicity in one sense or another: if neuter gender is to be expressed both by the noun and on the definiteness marker, then we need two copies. Multiple Merge provides one way to acquire copies by merging the feature in multiple places. This works well for features like case, which are uncontroversially merged after the phrases they attach to. It is less clear how this would work for gender. If we expect gender to be more "core" to the noun than definiteness, which seems intuitive, then gender must merge lower than definiteness; however, if we want Multiple Merge-style copying onto the definiteness marker, then gender must merge higher.

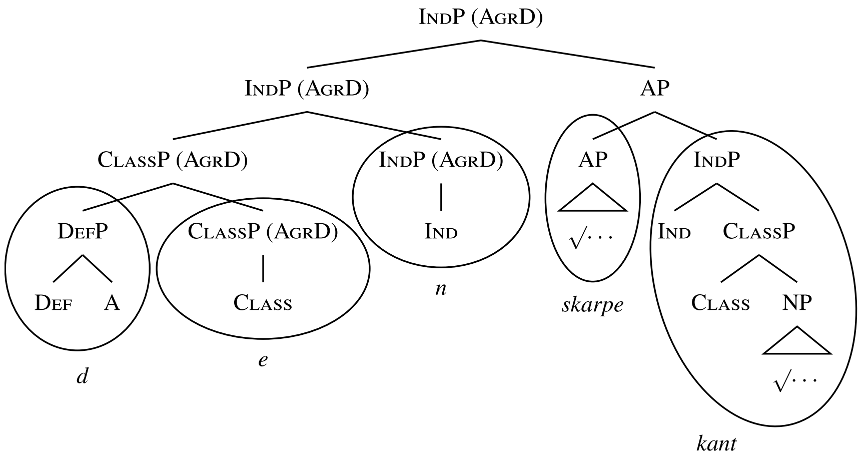
An alternative is to explicitly copy the features from their lower position close to the noun root to the definiteness marker after the definiteness marker is merged. This idea is partially inspired by Taraldsen's analysis of Bantu verbal agreement using Nanosyntax (Taraldsen, 2010; Taraldsen, Taraldsen Medová, & Langa, 2018) (though Taraldsen uses copying only as an analogy, and explicitly does not use it in his analysis proper) and by the implementation of feature assignment via copying for Russian case concord in Pesetsky (2013), though Pesetsky's principle, like Multiple Merge, copies concord features "downwards" onto existing feature bundles when merged. Gender concord is peculiar in appearing to copy "upwards", if we wish to keep the feature hierarchy in (20).

Since features cannot be copied from one bundle to another in Nanosyntax, I propose a new mechanism. Let D2 be a "placeholder" feature AGRD. When merged, it will instead copy all the features between CLASS and IND from the noun and merge them in the place where it was about to be merged. (I will add AGRD in parentheses to nodes which arise by copying heads into the place of AGRD.) This can be viewed as an entry for AGRD in the lexicon (albeit a new kind of lexical entry), or as a rule to be added to the spellout algorithm. For common gender nouns, this will copy exactly CLASS and IND; for neuter nouns, it will copy NEUT as well. Appropriate lexical entries for CLASS, IND and [$_{\text{NEUTP}}$ NEUT INDP] then yield *-en* and *-et*, as shown in (27) and (28). In fact, we decompose the suffix as *-e-n/-e-t*, with a separate entry for CLASS which just spells out *e*. This is mildly motivated by the overlap between *-en* and *-et*, but we will see a better motivation for this when discussing adjective agreement in Section 6. Finally, let D1 be called DEF, suggesting that in the standard case the definiteness is subsumed by the noun and only reflected by the definiteness concord of AGRD, while becoming overt as *d* in the presence of an adjective.

(27)


IndP (AgrD)
ClassP (AgrD)
IndP (AgrD)
DefP
ClassP (AgrD)
Ind
NeutP (AgrD)
Def
IndP
Class
Neut
Ind
NeutP
e
t
Neut
ClassP
Class
NP
hus
√. . .

(28)


IndP (AgrD)
IndP (AgrD)
AP
ClassP (AgrD)
IndP (AgrD)
AP
IndP
DefP
ClassP (AgrD)
Ind
√. . .
Ind
ClassP
Def
A
Class
n
skarpe
Class
NP
d
e
√. . .
kant

Summary

This proposal now handles the full paradigm of data laid out in Section 2. We have gained coverage of the data at the expense of loosening our requirements for prefix closure, adopting late prefix closure from Starke (2018), and at the expense of adding an (in principle) unrestricted copying system via placeholder features, though our use of it was fairly limited. The scope of this system clearly needs to be worked out in more detail by studying concord in other languages if we want to restrict it. I will show in the next section that the proposed copying mechanism can be used 'as is' to derive strong adjective agreement, suggesting that it may not be that stipulative and that it might be possible to restrict it to a narrow domain. Likewise, it may be possible to limit the size of prefix workspaces to at most words, to the extent that Nanosyntax can define wordhood, pending further investigation into complex prefixes.

On the plus side, the account of the structural allomorphy in the Danish definiteness alternation is elegant and can easily be extended to the relative clause examples in (5) and (6) by having the restrictive relative clauses merge in the same place as the AP, while the non-restrictive relative clauses merge higher and do not intervene between the noun and DEF (D1). The alternation is explained by the position of the AP relative to DEF and the availability of prefix and suffix lexical entries for DEF. Nanosyntax's prediction that prefixes are a last resort is borne out in this data and explains why we only get the prefix (prenominal article) when the AP blocks the suffix. This provides a deeper explanation than the Distributed Morphology analysis of Hankamer & Mikkelsen (2018). While Hankamer & Mikkelsen do an excellent job laying out the data and showing that the allomorphy must be structural (contra Embick & Marantz, 2008), their analysis relies entirely on their formulation of the Sisterhood Condition:

(29) *Sisterhood Condition* (Hankamer & Mikkelsen, 2018)
A definite D, D[def], is realized as a suffix if and only if it is a sister to a minimal N. Otherwise D[def] is realized as a free-standing article.

This correctly captures the data and its dependence on structural rather than linear intervention, but since it is phrased in words instead of explicit vocabulary insertion rules, it only provides a minimal DM-internal explanation for why this allomorphy occurs. While the account here likewise stipulates where the modifier intervenes (the position for DEF as a suffix is equivalent to being 'sister to a minimal N'), the spellout algorithm then dictates that definiteness (DEF + AGRD) cannot be spelled out as a suffix, and that the next step is to try a prefix – the prenominal article. (The only "stipulation" needed thereafter is to provide an appropriate lexical entry for *d*.) This motivates why Danish resorts to a prenominal article[9].

Handling Caha's Russian data

As we noted in Section 3.2, Multiple Merge was introduced by Caha (2019) to address case on Russian numeral phrases, repeated here:

(18)	pjat'	stol-ov
	five.NOM	tables-GEN.PL
	'five tables'	
(19)	pjat'-i	stol-am
	five-DAT.SG	tables-DAT.PL
	'to five tables'	

Caha analyses this pattern by having case merge into both the numeral and the noun, with this spelling out successfully on the noun for cases DAT and higher. Moreover, Caha's proposal crucially relies on being able to re-open and modify the main spine below the prefix using Multiple Merge. Without that, we need to propose that a copying feature AGRCASE is merged onto the noun but only "expands" into its copied features after those features have been merged later in the

[9] Another strategy a language could use is to suffix the definiteness marker to the modifier. An anonymous reviewer notes that this is precisely the strategy used in Bulgarian:

(i)	kniga=ta	
book-DEF		
(ii)	nova=ta	kniga
new-DEF	book	

If this clitic may be analysed the same way as suffixes in Nanosyntax, this suggests that in Bulgarian, there is no (overt) DEF head, and so instead of keeping the DEF prefix workspace open and adding the AGRD suffix to that, we are able to keep the highest AP prefix workspace open and suffix AGRD there. (This further predicts why in Bulgarian, DEF only attaches to the highest adjective.) Bulgarian represents a promising avenue for future research for this analysis: if DEF can be null, and definiteness still be expressed, perhaps it was hasty to name this head DEF and the other merely agreement (AGRD).

derivation and suffixed to the numeral. (Recall that when we discussed Multiple Merge for gender, its directionality was a problem; now we see the opposite for copying and case.) We then need to posit that AGRCASE's expansion can be satisfied vacuously if the features that it needs to copy are already present adjacent to it. In sum, while we can handle Caha's Russian numeral phrases with copying, the additional complexity of it suggests that perhaps a Multiple Merge-like account may still be preferable for some cases. I will return to this point in the conclusion.

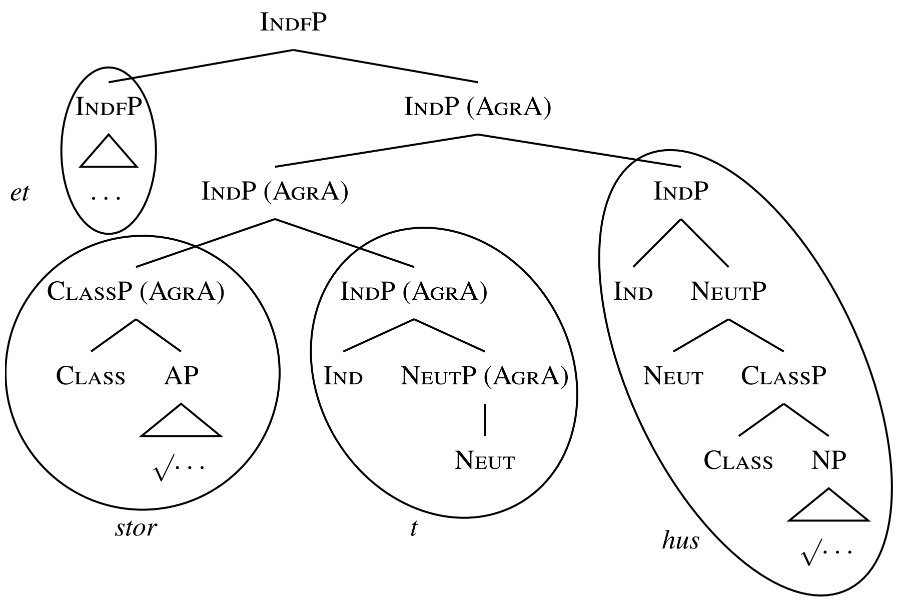
ADJECTIVE AGREEMENT

So far, I have glossed over the fact that Danish adjectives exhibit agreement with the noun. As in many Germanic languages, we see weak agreement (same suffix across genders) with a definite article and strong agreement (distinct suffixes by gender) with an indefinite article. It turns out that the same copying mechanism as before can be used to explain strong adjective agreement, supporting the choice above of which heads to copy. The strong and weak agreement pattern is shown below (using AGR to gloss the gender-unspecific weak agreement and Ø to indicate a null morpheme).

(30)	den	skarp-**e**	kant
	DEF.SG.C	sharp-AGR	edge
	'the sharp edge'		
(31)	det	stor-**e**	hus
	DEF.SG.N	big-AGR	house
	'the big house'		
(32)	en	skarp-**Ø**	kant
	INDEF.SG.C	sharp-C	edge
	'a sharp edge'		
(33)	et	stor-**t**	hus
	INDEF.SG.N	big-N	house
	'a big house'		

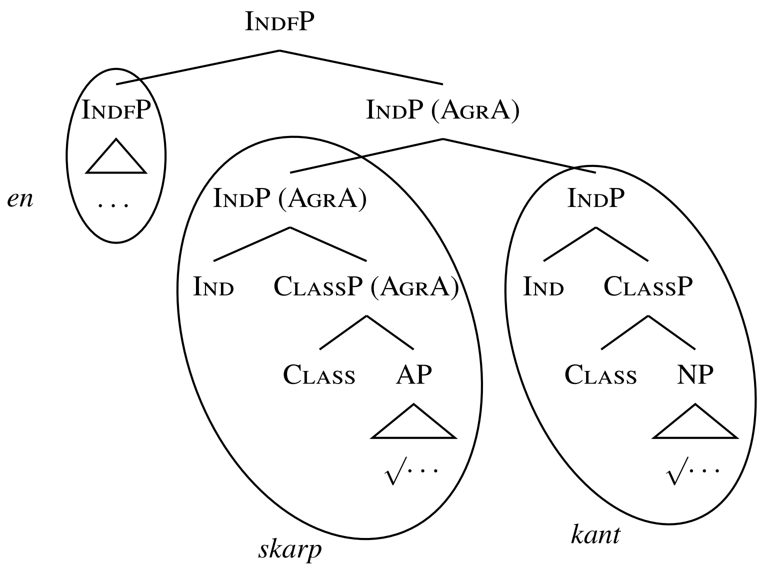
The overlap between the neuter strong agreement -*t* and the neuter definiteness suffix -*et* inspires copying the same material for adjective agreement as for definiteness agreement. Specifically, I introduce a new head AGRA which follows the adjective and which copies the same material as AGRD, namely all heads from IND through CLASS. IND and NEUT give us the desired -*t* suffix, while CLASS is absorbed by the adjective itself. This is shown in (34).

(34)



In fact, the null common gender "suffix" can be derived by allowing the adjective to spell out CLASS and IND but not NEUT; thus the presence of NEUT forces the separate suffix -*t*. The derivation of common gender *en skarp kant* with just CLASS and IND is shown in (35). Theoretically, this implies that adjectives may be 'innately' common gender, but not neuter.

(35)

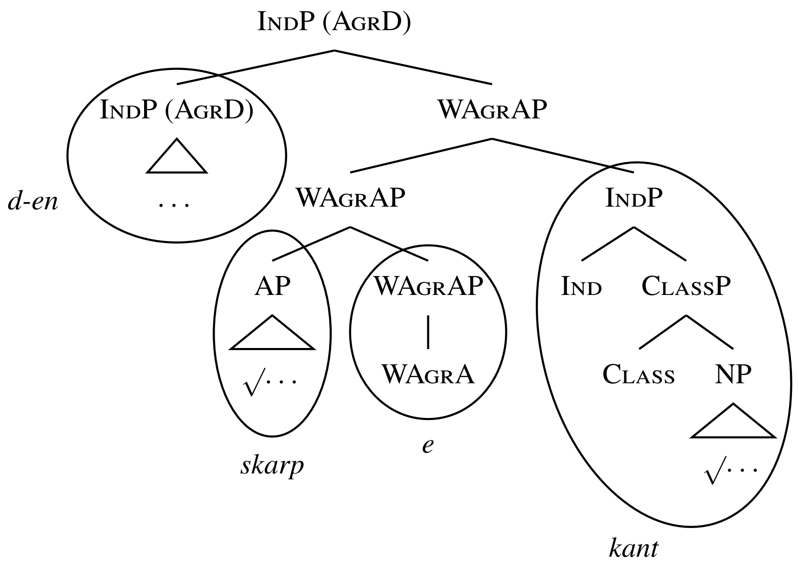


Since the focus of this section is to illustrate how adjective agreement may be derived from the copying mechanism for definiteness, I will not analyse the internal structure of the indefinite article for now (using INDFP as a placeholder) and will set aside its tantalizing overlap in form with *den/det*. This overlap is without question an avenue for future work[10], as is providing a Nanosyntactic explanation for *why* the indefinite article triggers strong agreement while the definite article triggers weak agreement. Here, I will merely assume that if weak agreement is in the feature sequence, then so must be a definite head, and likewise for strong agreement and indefinites. This is not an explanation but may be likened to the stipulation that if NEUT is to be expressed, we must have CLASS, and so forth.

Finally, observe that it is possible to analyse weak agreement in this framework without further theoretical additions, albeit also without adding any interesting insights: simply posit a WAGR head which is lexically defined as -*e* and stipulate, as discussed, that it occurs above the adjective if DEF is present.

[10] One possible analysis which explains the overlap between *en/et* and *den/det* is to posit an INDF head which is always a prefix and which is phonologically null. This behaves exactly like the prefix DEF head and attracts the AGRD suffix *-en/et*, so that the whole indefinite prefix/article gets realised as *Ø-en (en)* and *Ø-et (et)*. This derives the examples above as well as the unmodified phrases *en kant* 'an edge' and *et hus* 'a house'. However, null heads are generally avoided where possible in Nanosyntax and it is not clear from an explanatory perspective why the indefinite should contain a definite agreement morpheme. Without some justification of why the indefinite should be able to be decomposed in this way, explaining *en/et* within Nanosyntax remains an unsolved problem.

(36)



CONCLUSION

Nanosyntax provides an elegant account of the Danish definiteness alternation. Since Nanosyntax treats all allomorphy as structural, it is excellently positioned to account for the Danish data. Moreover, it can explain why the definiteness marker shifts from suffix to prefix, something that the Distributed Morphology account of Hankamer & Mikkelsen (2018) struggles to do using theory-internal reasons. To do so, however, two steps away from the most restrictive formulation of Nanosyntax provided by Caha (2019) are needed. To explain the alternation and the relationship between the articles *den/det* and the suffixes *-en/-et*, I shifted from Caha's immediate prefix workspace closure to closing the prefix "as late as possible" (Starke, 2018). While it need only stay open a "short" time (building a single word), the question of exactly what restrictions we can place on prefix closure remains open for future research. Likewise, the Danish data requires stepping away from Caha's elegant and restrictive principle of Multiple Merge for handling concord. Instead, I proposed an overt copying mechanism to copy the gender features onto the definiteness marker. The scope of what may be copied, when and in what direction remains wide open and will ultimately depend on data from other languages. This proposal is thus not intended as a final solution for concord in Nanosyntax but rather as a first proposal to spark discussion and prompt future research to establish the bounds and limitations of the mechanisms involved. The fact that the Danish plural blocks concord with gender on the definiteness marker (both common and neuter plural share the same definiteness suffix -*ne* and article *de*) suggests that we may not always want to copy blindly from IND to CLASS and invites a more complex criterion to capture this intervention by the plural (GROUP) feature[11]. Further, since gender shows "upward" concord onto items merged after it while case distributes "downward" across previously merged constituents, it remains open whether one or two mechanisms are needed. By proposing a concise account of the structural allomorphy and exposing what needs to be achieved to capture the concord in this case, I hope that this paper may pave the way to a revised proposal of Multiple Merge or a much narrower copying mechanism which can both account for the full Danish paradigm and retain the restrictiveness which so centrally distinguishes Nanosyntax from Distributed Morphology.

REFERENCES

Baunaz, L., & Lander, E. (2018). Nanosyntax. In: L. Baunaz, L. Haegeman, K. De Clercq, & E. Lander (eds.), *Exploring Nanosyntax*. Oxford University Press.

Caha, P. (2009). *The Nanosyntax of case* (PhD Thesis). Universitetet i Tromsø.

Caha, P. (2019). Case competition in Nanosyntax. A study of numeral phrases in Ossetic and Russian. *LingBuzz*. Available at https://ling.auf.net/lingbuzz/004875

Caha, P. (2020). *Modeling declensions without declension features. The case of Russian*. Available at https://ling.auf.net/lingbuzz/005537

Caha, P., De Clercq, K., & Vanden Wyngaerd, G. (2019). The fine structure of the comparative. *Studia Linguistica*, *73*(3), 470–521.

Delsing, L.-O. (1993). *The Internal Structure of Noun Phrases in the Scandinavian Languages* (PhD Thesis). University of Lund.

Embick, D., & Marantz, A. (2008). Architecture and Blocking. *Linguistic Inquiry*, *39(1)*, 1–53.

Halle, M., & Marantz, A. (1993). Distributed morphology and the pieces of inflection. In: *The view from building 20*. The MIT Press. 111–176.

[11] Note that this counts as an intervention in terms of shielding the gender of the noun from "view" for concord, but does not count as an intervention of a modifier between the noun and DEF (definite plurals with no modifier still use a suffix). I am grateful to Pavel Caha and the audience at the Brno Nanolab for drawing this to my attention.

Hankamer, J., & Mikkelsen, L. (2005). When movement must be blocked: A reply to Embick and Noyer. *Linguistic Inquiry*, *36(1)*, 85–125.

Hankamer, J., & Mikkelsen, L. (2018). Structure, Architecture, and Blocking. *Linguistic Inquiry*, *49(1)*, 61–84.

Pesetsky, D. (2013). *Russian case morphology and the syntactic categories*. MIT Press.

Starke, M. (2010). Nanosyntax: A short primer to a new approach to language. *Nordlyd*, *36(1)*, 1–6.

Starke, M. (2018). Complex Left Branches, Spellout, and Prefixes. In: L. Baunaz, L. Haegeman, K. De Clercq, & E. Lander (Eds.), *Exploring Nanosyntax*. Oxford University Press.

Taraldsen, K. T. (2010). The nanosyntax of Nguni noun class prefixes and concords. *Lingua*, *120*(6), 1522–1548.

Taraldsen, K. T., Taraldsen Medová, L., & Langa, D. (2018). Class prefixes as specifiers in Southern Bantu. *Natural Language & Linguistic Theory*, *36*(4), 1339–1394.