Spelling out the Numeration, Part 1: Selection by Itself Accounts for Synthetic-Periphrastic Alternations

Benjamin Bruening (University of Delaware)

rough draft, March 12, 2021; comments welcome

Abstract

The concept of the numeration (Chomsky 1995: chapter 4) has been important in recent syntactic theory, but how it works has never been fully explored. I suggest that spelling out how items are selected from the lexicon and put into the numeration, and how they are taken out of the numeration and merged in the syntax, can explain numerous phenomena in syntax. This paper starts by investigating how items are selected from the lexicon and put into the numeration. It shows that spelling this out in a way that is independently required for things like nominal concord explains synthetic-periphrastic alternations like English *do*-support and the "overflow" pattern of auxiliaries in Kinande (Bjorkman 2011). These fall into a common pattern where a language requires one-to-one matching between inflectional heads and other elements. The proposed account enables us to do without syncategorematic insertion mechanisms; all we need is selection-constrained selection of items for the numeration.

1 Introduction

Chomsky (1995: chapter 4) introduced the notion of a *numeration*. In his conception, the numeration is a selection of elements from the lexicon which a syntactic derivation draws from to build a syntactic structure. The original motivation for this device was to provide a comparison set for calculations of economy: economy conditions compared what the derivation could have done with the same numeration. Subsequent work has often maintained the numeration in some form (with sub-numerations for smaller chunks: the "lexical array" for each *phase* in Chomsky 2000), but its importance has diminished. Most approaches to economy use not global but local calculations of economy, where the numeration plays little role. It is therefore not clear whether the notion of a numeration has any role to play in contemporary models of syntax.

This paper is part of a larger research project exploring whether the concept of a numeration can actually be useful for explaining a variety of syntactic phenomena. If the syntax does include a numeration, then we should explore what constraints might hold of it. We should also explore the process that selects items from the lexicon for the numeration, and the process that selects items from the numeration to use in the derivation. Properties of the numeration and these two input-output processes might help to explain some syntactic phenomena. Here, I suggest that they can, starting with the process that selects items from the lexicon and puts them in the numeration. Other work in this research project will investigate the process that takes items from the numeration and merges them in the syntax.

This paper explores one particular phenomenon where the numeration is directly relevant, namely, synthetic-periphrastic alternations like English *do*-support. These are often analyzed as involving a last-resort insertion mechanism, where the syntax can insert an item that was not previously part of the derivation, or even part of the numeration. I suggest that a better analysis involves selection of items for the numeration, where selectional requirements necessitate selection of the periphrastic item (e.g., *do*) for the numeration. In this analysis, the periphrastic item is present in the numeration from the beginning. I discuss two examples of such alternations: The "overflow" pattern in Kinande described by Bjorkman (2011), and *do*-support in English.

The phenomena discussed here all involve a generalization that shows up in language after language, where we see one thing always requiring another. This commonly occurs with inflectional morphemes, where languages require, for instance, every head of a certain type within a nominal to have an inflectional (concord) morpheme. In the case of verbal morphology, many languages require one verb for every inflectional head. This type of one-to-one matching, I argue, is best accounted for with selectional requirements. As I will show, once we have such selectional requirements driving the selection of items for the numeration, synthetic-periphrastic alternations are easily accounted for, without the need for syncategorematic last resort insertion mechanisms.

I start with background on the conception of the numeration adopted here (section 2), followed by general remarks on synthetic-periphrastic alternations (section 3). Section 4 illustrates the generalization about one-to-one selection by looking at nominal concord, and section 5 formalizes how this works as selection-driven selection of items for the numeration. Sections 6 and 7 apply the analysis to the Kinande "overflow" pattern and English do-support, respectively.

2 Background: Conception of the Numeration

I begin by laying out in broad strokes the model of the grammar that I will be exploring here. Subsequent sections (and companion papers) investigate aspects of this model in greater detail.

First, I assume that there is a numeration for a derivation. Second, following Chomsky (2000) and much subsequent work, I assume that the derivation of a sentence is divided into smaller chunks, *phases*, and that each phase has its own numeration (I will use the term "numeration" and will not adopt the "lexical array" term that Chomsky 2000 uses). I will assume for purposes here that at least CPs and nominals are phases.

The numeration is a set of elements that are selected from the lexicon and placed in a memory buffer. The syntax draws from this memory buffer to build a structure, and is limited to drawing from it; it cannot access the lexicon once the derivation begins. To give a very simple example, the numeration for the nominal phrase *a heavy heart* will include the indefinite determiner *a*, the adjective *heavy*, and the noun *heart* (depending on one's theory, there may be unpronounced elements as well). The syntax will put these three items together to form a nominal. It does not do this in the numeration; there must therefore be a workspace in addition to the numeration where syntactic operations are performed. Let us call this workspace the *workspace*.

In this conception, then, the derivation of each syntactic phase involves a numeration and a workspace. Items are first selected from the lexicon and placed in the numeration. Items are then taken from the numeration and merged in the workspace:

(1) [LEXICON] —L2N→ [NUMERATION] —N2W→ [WORKSPACE]

In this model, there are two mechanisms that interface with the numeration, an input mechanism and an output mechanism. I will call the mechanism that selects items from the lexicon and moves them to the numeration "L2N," for "Lexicon to Numeration selection procedure." The output mechanism I will call "N2W," for "Numeration to Workspace selection procedure." L2N selects items from the lexicon and puts them in the numeration. Once the numeration is complete, N2W selects items from the numeration and moves them to the workspace, where the syntax puts them together using Merge. The derivation cannot access the lexicon at this stage, it can only work with what was selected for the numeration.

How exactly the input mechanism, L2N, works will be the focus of this paper (companion papers investigate N2W). There have to be principles that determine how items are selected from the lexicon and put into the numeration. Subsequent sections investigate what some of these principles might be.

3 Synthetic-Periphrastic Alternations

This paper examines how L2N—the operation that selects items from the lexicon and puts them in the numeration—might work. I propose here that constraints on this operation can help us to understand synthetic-periphrastic alternations like English *do*-support, illustrated below:

(2) a. She studies too much.
    b. Does she study too much?

In English, the main verb can combine synthetically with tense and agreement in a simple declarative (*studie-s*), but in a yes-no question with inversion, a periphrastic construction is used instead. Tense/agreement is borne by a semantically contentless auxiliary (*doe-s*), while the verb is in its uninflected form.

Periphrastic constructions like English *do*-support have generally been analyzed as *last resort* operations. In a simple declarative, there is no reason to do *do*-support, and so it is not allowed; it is only when some constraint is violated—for instance, separating tense and the main verb so that they cannot combine—that a repair operation is triggered, in this case insertion of the item *do* (Chomsky 1957, Bobaljik 1995, among many others). This requires that the syntax be able to insert items in the course of the derivation, even ones that were not part of the numeration from the beginning. This is important here given this paper's focus on the numeration. The whole idea of the numeration was that the syntax cannot access the lexicon directly, it can only work with what was provided to it in the numeration. Syncategorematic operations like *do*-insertion violate the whole spirit of the numeration and are conceptually incompatible with it. Allowing syncategorematic operations also opens the door to massive overgeneration: if the syntax is capable of inserting things that were not in the input it was given, then it should be able to generate any structure from almost any input. This is not a good state of affairs. The whole point of the numeration is to constrain the syntax in what it can do. Even limiting syncategorematic insertion mechanisms to semantically contentless items like *do* and expletives is problematic: as Chomsky (2000) discusses, whether or not the highest NP in a clause moves to Spec-TP in English has to depend on whether there is an expletive in the numeration. If this is the case, then we do not want the syntax bypassing the numeration and inserting semantically contentless elements that were not present in the numeration.

I therefore assume that the syntax is incapable of inserting anything that is not present in the numeration. Syncategorematic last resort insertion operations like *do*-insertion do not exist. This has the consequence that the numeration for a yes-no question like (2b) must include *do* in its numeration. The question now is how that can be ensured.

The basic idea that I will pursue here is that information about items selected for the numeration can be used to select other items for the numeration. For instance, in the English *do*-support case, something will ensure that if the numeration includes a C with the feature that triggers head movement to C, then the numeration also has to have an element of category Aux(iliary)V(erb) in it. What ensures this, I will suggest, is nothing more than selection. As I will show, selectional requirements of this sort are independently necessary, and they are what drive L2N in general. Once we have them, it is a simple matter to account for synthetic-periphrastic alternations without problematic last-resort, syncategorematic insertion operations.

I will discuss two cases of periphrasis from the literature: (1) the "overflow" pattern in Kinande described by Bjorkman (2011); (2) *do*-support in English. First, though, I motivate the idea that selection of items for the numeration is driven by selection, and show that selection is necessary to capture patterns of inflection across the languages of the world.

4 Independent Necessity: Nominal Concord

Many languages have much more agreement than English does. In particular, in many languages, nouns belong to classes based on gender or other features. They also have number features and (often) case features. In many languages, various items show concord with the features of the noun. I will illustrate with Bulgarian. Consider the following two nominal expressions:

(3) a. goljam-a-ta interesn-a knig-a
big-Agr-Def interesting-Agr book-Agr
'the big, interesting book' (Franks 2001: 54, (3c))

b. mnogo xubav-i-te knig-i
very nice-Agr-Def book-Agr
'the very nice books' (Franks 2001: 55, (5a))

Nouns in Bulgarian take different inflectional endings depending on their gender, number, and case. The noun for 'book', *knig*, is feminine. The ending *-a* is feminine singular, nominative case. The ending *-i* is feminine plural, nominative case.

The thing to notice is that various other heads in the nominal, in particular adjectives, also have these endings, and that each adjective has to have one, and only one. The adjectives agree with the head noun in gender, number, and case. It is not just adjectives: possessive pronouns and numerals also bear these agreement suffixes, as in the examples in (4). Adverbs like 'very' do not (3b).

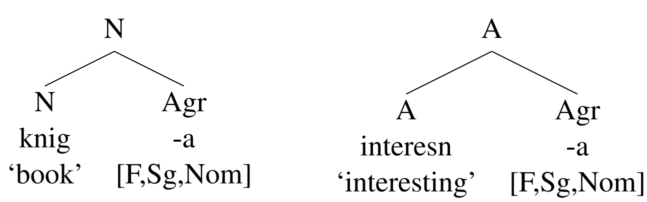
(4) (Embick & Noyer 2001: 568, Harizanov & Gribanova 2014: (2b))

a. moj-a-ta xubav-a knig-a
my-Agr-Def nice-Agr book-Agr

b. tr-i-te nov-i knig-i
three-Agr-Def new-Agr book-Agr

Any analysis of Bulgarian therefore has to account for the fact that certain types of heads (at least Poss, Num, A, and N) must have a single Agr(eement) suffix, one each. I assume that all complex forms are put together by the syntax. That includes concord suffixes being put together with the heads they attach to. This means that the syntax builds forms like the following:

(5)



This in turn means that the numeration must include an Agr head if it includes an N. It must include another Agr head if it also includes an A. In fact, the numeration must include exactly one Agr head for each head of the relevant type that it includes. The only numeration for (3a) that will result in a well-formed output has three Agr heads, one for each A and N:

(6) Numeration for (3a) *goljam-a-ta interesn-a knig-a*, 'the big, interesting book':

a. goljam, 'big', A (requires Agr)

b. a, Agr1

c. ta, Definite Article

d. interesn, ‘interesting’, A (requires Agr)

e. a, Agr2

f. knig ‘book’, N (requires Agr)

g. a, Agr3

For a language like Bulgarian, then, there must be some mechanism that ensures that for every A, N, etc., selected from the lexicon for the numeration by L2N, an Agr head also has to be put into the numeration. For lack of a better term, I will say that all and only the relevant items in Bulgarian have a feature [+N]. Possessive pronouns, adjectives, numerals, and nouns all have the feature [+N]. Adverbs do not. Something must then ensure that the following happens in Bulgarian:

(7) Bulgarian: Each time L2N selects a [+N] element from the lexicon and moves it to the numeration, it must also select an Agr head as the next item to move from the lexicon to the numeration.

This will ensure one-to-one matching between [+N] heads and Agr heads in the numeration.

Any analysis of Bulgarian that includes a numeration and builds all complex forms in the syntax will have to include something that ensures this one-to-one matching. If the syntax can only work with what it has in the numeration, and a well-formed nominal expression in Bulgarian has to have exactly one Agr head per [+N] head, then the numeration will have to be constrained to include exactly as many Agr heads as [+N] heads. This is simply necessary.

The question now is, what ensure this? And, on what basis does L2N work in general? I suggest that the answer to both questions is the same: It operates on the basis of selection.

5 L2N Operates on the Basis of Selection

Consider a CP phase for the moment, and how L2N might proceed to select items for the numeration that will build the CP phase. The obvious place to start is with the head C, since it is the head of the phase. The obvious way to proceed from there is on the basis of selection: C selects T, so the next item selected will be a T. And so on. The more general principle under which L2N works can then be stated as follows (using the notation for selectional features from Bruening 2013):

(8) Each time L2N selects an element X with a selectional feature [S:Y] and moves it from the lexicon to the numeration, it must then select an element Y from the lexicon and move it to the numeration.

Ordering L2N in this way makes conceptual sense. In the clausal domain, it is the higher functional heads that determine the properties of the clause and what items are necessary below them. Starting with the highest functional head and working downward will make L2N’s task easier: L2N will simply follow selectional requirements of the items it has just chosen. So, for instance, in a clausal phase, L2N might start by choosing, say, a declarative C head from the lexicon; a declarative C head selects for a finite tense head, so that is the next item L2N selects and moves to the numeration; and so on. This means that L2N will mostly operate in a fashion that corresponds to top-down in the syntactic tree that will ultimately be built. I will assume that this is indeed the case, and that the selection of items from the lexicon for the numeration proceeds on the basis of selectional requirements of the items previously chosen.

To account for Bulgarian nominal concord, it now suffices to say that all [+N] elements in the language select for Agr:

(9) In Bulgarian, all [+N] elements have the selectional feature [S:Agr].

Now whenever a [+N] element is selected from the lexicon and moved to the numeration, an Agr head must be, too, given (8).

Thus, selection can ensure the presence of one thing in the numeration based on the presence of another thing in the numeration. We can see that this is independently necessary, for things like nominal concord in Bulgarian.

An alternative to selection is to have L2N operate on the basis of a *Hierarchy of Projections* that is either universal or largely so but with some language-particular variations. The idea of a Hierarchy of Projections is widely adopted. It is part of cartographic approaches (e.g., Rizzi 1997; Cinque 1999), but it is also assumed in some form by researchers working within many different approaches. It is spelled out most clearly in Adger (2010), and it is Adger's system that I will focus on here (see also Adger & Svenonius 2011, Cowper 2010).

The Hierarchy of Projections view says that there is a fixed hierarchy of functional projections in each domain, for instance clauses and nominals. The hierarchies proposed by Adger (2010) for English clauses and nominals are the following. The numeral indicates the place on the hierarchy:

(10) (Adger 2010: 198, (39))

a. $\langle$V,1$\rangle$ < $\langle$v,2$\rangle$ < $\langle$Pass,3$\rangle$ < $\langle$Prog,4$\rangle$ < $\langle$Perf,5$\rangle$ < $\langle$Mod,6$\rangle$ < $\langle$Neg,7$\rangle$ < $\langle$T,8$\rangle$ < $\langle$Fin,9$\rangle$ < $\langle$C,10$\rangle$

b. $\langle$N,1$\rangle$ < $\langle$n,2$\rangle$ < $\langle$Poss,3$\rangle$ < $\langle$Num,4$\rangle$ < $\langle$D,5$\rangle$ < $\langle$Q,6$\rangle$

Adger (2010) proposes that there are two modes of combination, Sel(ect)-Merge and HoP-Merge (for "Hierarchy of Projection Merge"). In Sel-Merge, a category selects a feature, say "[C]," and this feature is checked off by Merging it with a category with that feature (e.g., a CP). For instance, a verb like *think* that selects CPs will have an [S:C] selectional feature that is checked off by merging a CP with it as its complement.

HoP-Merge, in contrast, simply combines two categories with no selectional relation between them. The only condition is that it must respect the Hierarchy of Projections. The idea is that it would not be desirable to have each category select a long disjunctive list; rather, Universal Grammar fixes a hierarchy, and Merge is free so long as it respects that hierarchy. For instance, T can freely combine with any of the categories below it (Neg, Mod, Perf, etc.). So we can have T combine directly with a verb, or with a verb plus Pass, verb plus Pass plus Prog, and so on (it is not clear what English elements "v,2" and "Fin,9" are supposed to be, so I ignore them here):

(11) a. The runner tripped. (V,1–T,7)

b. The runner was tripped. (V,1-Pass,3-T,7)

c. The runner was being tripped. (V,1-Pass,3-Prog,4-T,7)

The sequences 1–7, 1–3–7, and 1–3–4–7 all respect the hierarchy, so they are allowed. What is ruled out is merging elements in violation of the hierarchy:

(12) a. * The runner was having tripped. (V,1–Perf,5–Prog,4–T,7)

b. * The runner has should trip. (V,1–Mod,6–Perf,5–T,7)

1–5–4–7 and 1–6–5–7 violate the hierarchy and are ruled out.

Thus, HoP-Merge is supposed to allow all and only the licit orderings of functional elements within the English clause. Similar hierarchies for other languages would presumably have the right result for those languages, as well. In the context of the current paper, we could propose that the way L2N works is on the same basis as HoP-Merge: it starts with something high on hierarchy and proceeds downward, or, it starts with something low and proceeds upward.

Unfortunately, HoP-Merge is too permissive and has to be supplemented with Sel-Merge, as Bruening et al. (2018) point out. HoP-Merge incorrectly permits C to merge directly with V, for instance. This respects the hierarchy of projections, but there is no language that I know of where C can combine directly with the lexical verb, with no functional categories in between. In all languages something is necessary in between, either T or some sort of Asp or Mod. HoP-Merge is then not sufficient, and some sort of selectional feature has to be added to C.

Once we do that, however, we might as well get rid of HoP-Merge, and only have Sel-Merge. As stated above, the idea behind HoP-Merge seems to be that a disjunctive list of possible selectees would be undesirable, but we know that this is necessary for many cases of selection. For instance, many verbs can select multiple categories, but only one at a time. The verb *forget* selects for CPs or NPs, but not both simultaneously; it also does not allow other categories like APs or VPs:

(13) a. She forgot [$_{\text{NP}}$ her didgeridoo].
b. She forgot [$_{\text{CP}}$ that her didgeridoo needed tuning].
c. * She forgot [$_{\text{NP}}$ me] [$_{\text{CP}}$ that her didgeridoo needed tuning].
d. * She forgot [$_{\text{AP}}$ famous].
e. * She forgot [$_{\text{VP}}$ play the didgeridoo].

We need to say that *forget* selects for *either* NP or CP, which is exactly disjunctive selection; there is simply no way around this. Similar selectional patterns are prevalent, and in fact probably the norm: *think* selects CPs or PPs, *become* selects APs or NPs, and so on, ad infinitum. We can notate this with a set:

(14) *forget* has the feature [S:{N, C}].

The [S] feature will then be satisfied by merger of one member of the set.

The effect of HoP-Merge can then be captured very straightforwardly using the same kind of disjunctive selection, as follows (ignoring Adger's v and Fin, since it is not clear what English elements correspond to them):

(15) a. Pass has the feature [S:V].
b. Prog has the feature [S:{V, Pass}].
c. Perf has the feature [S:{V, Pass, Prog}].
d. Mod has the feature [S:{V, Pass, Prog, Perf}].
e. Neg has the feature [S:{V, Pass, Prog, Perf, Mod}].
f. T has the feature [S:{V, Pass, Prog, Perf, Mod, Neg}].
g. C has the feature [S:T].

Prog, Perf, Mod, Neg, and T all select a disjunctive list, but C strictly selects T. Pass also strictly selects V. This correctly captures the effects of HoP-Merge, but does not run afoul of the problem of permitting C to directly combine with lower categories. The selectional statements in (8) also correctly rule out the violations in (12), since Perf is not on the list of things selected by Prog in (12a), and Mod is not on the list of things selected by Perf in (12b).

Disjunctive selection, then, captures the Hierarchy of Projections without a Hierarchy of Projections. It is also independently necessary, for verbs like *forget* and thousands of others. It is entirely unclear why using disjunctive selection would be less desirable than positing an entire second mode of Merge and stipulating an extrinsic hierarchy. Standard metrics of theory comparison clearly favor disjunctive selection over HoP-Merge: It captures everything that HoP-Merge does, but does not overgenerate the way HoP-Merge does; and it is independently necessary. Adding HoP-Merge to Sel-Merge is simply multiplying theoretical devices unnecessarily.

Moreover, HoP-Merge will not help us to capture the Bulgarian nominal concord pattern, where every [+N] head requires an Agr head. There is no sense in which this follows from an extrinsic hierarchy. Just as we actually need selection to capture the hierarchy of projections, so too can selection capture the Bulgarian pattern. All we have to say is that every [+N] head in Bulgarian has to have the feature [S:Agr]. I take all of this to indicate that we do not want L2N to operate on the basis of a universal hierarchy, instead we want it to operate on the basis of selection. I will therefore spell out how L2N operates using selectional features.

I will assume that there are designated starting points for each phase:

(16) *Starting Points for Numerations*
  a. For a CP phase, L2N starts by selecting a C from the lexicon and moving it to the numeration.
  b. For an NP phase, L2N starts by selecting…

Since NP phases are not relevant here, I will leave them aside. CP phases are relevant here, and with them, L2N always starts with a C. L2N then proceeds on the basis of selection (repeated from above):

(17) Each time L2N selects an element X with a selectional feature [S:Y] and moves it from the lexicon to the numeration, it must then select an element Y from the lexicon and move it to the numeration.

C always selects T (15g), so the next item selected will be T. T now selects a disjunctive list (15f). We can assume that when (17) applies to an element with a disjunctive list, it picks one element from the list to move into the numeration. So, for instance, L2N might pick Neg from the list in (15f). Neg then selects its own disjunctive list (15e), so the process will repeat (note that I will modify the analysis of the English clause in section 7).

To summarize so far, the basis for L2N is nothing but selection. Selection of items for the numeration for a phase starts from a designated starting point (the phase head), and then proceeds on the basis of selection. Selection is independently necessary. Once we have it, we can account for patterns like the one-to-one matching that we saw in Bulgarian nominal concord by viewing the Agr requirement of [+N] heads as an instance of selection: [+N] heads select Agr heads (they have the feature [S:Agr]).

Now that we have a simple understanding of how L2N works, we can go on to investigate synthetic-periphrastic alternations. I start with the "overflow" pattern of Kinande.

6 The "Overflow" Pattern: Kinande

Bjorkman (2011) describes several different languages where what she calls an "overflow" pattern occurs with auxiliary verbs. One such language is Kinande. Kinande can mark tense (T) on the main verb (there are four different "distances" of past tense; 3 and 4 are distinguished by the tone pattern):

(18) tw-á-húma
  1Pl-Past3-hit+ToneA
  'We hit (recently).' (Bjorkman 2011: 25, (9c))

Kinande can also mark aspect (Asp) on the main verb, for instance progressive:

(19) tu-nému-húma
  1Pl-Prog-hit
  'We are hitting.' (Bjorkman 2011: 26, (10b))

However, Kinande cannot mark both T and Asp on the V at the same time. Instead an auxiliary must be used:

(20) tw-á-bya i-tu-nému-húma
1Pl-Past3-be+ToneA LNK-1Pl-Prog-hit
'We were (recently, not today) hitting.' (Bjorkman 2011: 26, (11a))

This is what Bjorkman means by "overflow": an auxiliary is only used when there are too many inflectional affixes for a single verb. Tense by itself does not require an auxiliary, nor does aspect by itself; but when they are both present, an auxiliary has to be used. Bjorkman (2011) describes similar patterns in Arabic and Latin, while Pietraszko (2017) describes one in Ndebele.

Bjorkman (2011) analyzes this in terms of last resort. In her analysis, both tense and aspect are Infl heads. A lower Infl head (Asp) gets in the way of a higher Infl head (T). The Infl heads need to Agree with the verb, but when one is in the way of the other, this cannot happen. The higher Infl needs to be realized on a verb, but because of the intervention of the lower Infl head, it cannot be realized on the lexical verb. This forces insertion of a default verb, which Bjorkman argues to be 'be', cross-linguistically.

I pursue a different analysis, for the reasons discussed in section 3. In the analysis I suggest, Kinande is behaving very much like Bulgarian, where there has to be one-to-one matching between [+N] heads and Agr heads. Kinande seems to require one-to-one matching between Infl heads (Asp and T) and verbal heads.[1] If there are two Infl heads, then there must be two verbal heads, as well. Viewed this way, something must ensure that in Kinande, for each Infl head put in the numeration, a head of category V must also be put in the numeration. As explained above, what ensures this is selection. We can state the generalization about Kinande as follows:

(21) In Kinande, all Infl heads have the feature [S:V].

Note the similarity to Bulgarian concord, where all [+N] heads select Agr. Both instances of selection have the result that there is one-to-one matching between certain heads and certain other heads.

Note that the requirement here is on Infl heads, not on V heads. In various languages, auxiliary verbs are not forced to appear in certain environments where inflectional heads seem to be absent (e.g., small clauses). In addition, bare verb stems can sometimes appear in certain environments. So the requirement seems to be a requirement on the Infl heads, and not on the verbal heads. This is also in keeping with selection, which works downward: it is the Infl heads, which are higher than Vs, which select Vs.

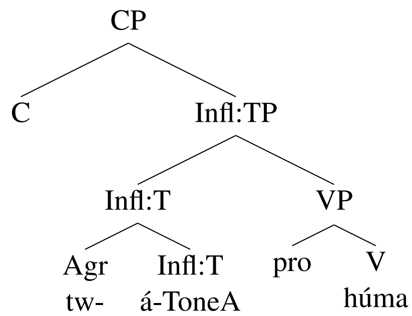
We can now go through possible derivations in Kinande and show how the overflow pattern follows from the selectional pattern in (21). I start with past tense by itself. Consider the following example:

(22) tw-á-húma
1Pl-Past3-hit+ToneA
'We hit (recently).' (Bjorkman 2011: 25, (9c))

Simply to keep things simple, I will not posit the presence of any null structure in an example like this, other than a null subject and a null C head to head the CP phase. The structure that is eventually built by the syntax will be something like the following. T and Asp are both subcategories of Infl (notated Infl:T and Infl:Asp), while the agreement morphology is an Agr head adjoined to Infl:

[1] In a poster presented at the 2020 LSA annual meeting, Arregi and Pietraszko state a generalization according to which many languages only allow one inflectional morpheme per verb. This is exactly the generalization I attempt to capture here. Pietraszko (2017: 89) states this in terms of "feature conflict": "an inflectional category, e.g. past tense, cannot combine with the main verb if the verb already has an inflectional feature (e.g. imperfective aspect)."

(23)



I assume the V moves to Infl and composes with the Agr and Infl:T heads. I will also assume that all NP arguments are arguments of the verb. Here, there is a null subject, which I notate “pro”; it is base-generated in the specifier of the lexical VP. I assume based on the translation that there is no object in this sentence. (Not much would change if we did include other null structure, for instance a null functional head that projects the external argument, as in Kratzer 1996. We would just have an additional head in the numeration, which is not an Infl head for the statement in (21).)

The derivation of this phase starts by selecting the necessary items from the lexicon and putting them in the numeration. As stated above, L2N starts a CP phase with C. The initial step is therefore for L2N to move a C from the lexicon to the numeration:

(24) Numeration for *tw-á-húma*, ‘we hit (recently)’, step 0:
  a. Ø (C[S:Infl])

I assume that in Kinande, C has the feature [S:Infl], as shown. L2N operates on the basis of selection, as specified in (17), so L2N is now forced to select something of category Infl. In this case it selects an Infl:T head, and moves it from the lexicon to the numeration:

(25) Numeration for *tw-á-húma*, ‘we hit (recently)’, step 1:
  a. Ø (C[S:Infl])
  b. a-ToneA (Infl:T[S:V], ‘Past3’)

As stated in (21), Infl heads in Kinande all have the selectional feature [S:V]. This is true of the Infl:T head selected in step 1. This requires that L2N now select something of category V and move it to the numeration. L2N selects the main verb ‘hit’:

(26) Numeration for *tw-á-húma*, ‘we hit (recently)’, step 2:
  a. Ø (C[S:Infl])
  b. a-ToneA (Infl:T[S:V], ‘Past3’)
  c. húma (V, ‘hit’)

Kinande has another requirement, as well. This one requires that there be an Agr head for every Infl head, the same way every [+N] head in Bulgarian has an Agr head. I also state this in terms of selection:

(27) In Kinande, all Infl heads have the feature [S:Agr].

This means that we must go back and revise step 1:

(28) Numeration for *tw-á-húma*, ‘we hit (recently)’, step 1 (revised):
  a. Ø (C[S:Infl])
  b. a-ToneA (Infl:T[S:V,S:Agr], ‘Past3’)

All Infl heads then have two selectional features in Kinande, one which selects a V, and one which selects an Agr. L2N therefore has to select two elements in step 2, so we also need to revise step 2 and add a step 3:

(29) Numeration for *tw-á-húma*, 'we hit (recently)', step 2 (revised):
  a. Ø (C[S:Infl])
  b. a-ToneA (Infl:T[S:V,S:Agr], 'Past3')
  c. húma (V[S:N], 'hit')

(30) Numeration for *tw-á-húma*, 'we hit (recently)', step 3:
  a. Ø (C[S:Infl])
  b. a-ToneA (Infl:T[S:V,S:Agr], 'Past3')
  c. húma (V[S:N], 'hit')
  d. Agr

Note that steps 2 and 3 can be simultaneous; there is no reason that L2N has to move only one item at a time. So we can either think of steps 2 and 3 as simultaneous, or as occurring in sequence (but it does not matter what order they occur in).

I will assume that Agr heads have no form in the numeration, as their form is determined contextually, in the syntax. In the current case, Agr will agree with the subject of the verb, and take its form based on the features it thereby acquires. How this works is unimportant here (any theory of agreement will do).

The verb also selects a subject ([S:N]), so L2N must select something that can satisfy this selectional requirement. In this case it selects an unpronounced pronoun, pro:[2]

(31) Numeration for *tw-á-húma*, 'we hit (recently)', step 4:
  a. Ø (C[S:Infl])
  b. a-ToneA (Infl:T[S:V,S:Agr], 'Past3')
  c. húma (V[S:N], 'hit')
  d. Agr
  e. pro (N, [1Pl])

This numeration is now complete. There is one Infl morpheme (T), and one V and one Agr, as well, as required. The only formal device we needed to ensure this was selection.

The syntax can now build the structure, selecting items from the numeration and merging them in the workspace as required by the output mechanism, N2W (explored in a companion work). No auxiliary verb is required, since the main verb is sufficient to satisfy the requirement of one-to-one matching between Infl heads and verbs (stated as a selectional requirement). Note that if an auxiliary verb were incorrectly selected, it would then require another element of category Infl, since auxiliary verbs have the feature [S:Infl] (see 44 below).

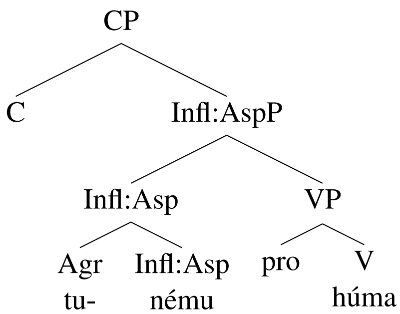
A similar derivation will take place when aspect occurs by itself. Asp is also a subcategory of Infl, Infl:Asp. Consider the following example:

(32) tu-nému-húma
1Pl-Prog-hit
'We are hitting.' (Bjorkman 2011: 26, (10b))

[2]Note that NPs actually constitute their own phase, and therefore have their own numeration; I will ignore this here, and treat the subject pronoun as though it is a single item in the clausal numeration. In the companion papers, I will propose that what is actually selected here is a placeholder item, something with no content but category N, which is at the appropriate point expanded as a phase with its own numeration.

Again eschewing null elements other than C and pro, I assume this has the following structure:

(33)



Selection of items from the lexicon proceeds exactly as in the derivation of the simple past tense. It begins with selection of a C:

(34) Numeration for *tu-námu-húma*, 'we are hitting', step 0:

a. Ø (C[S:Infl])

C selects something of category Infl, so L2N has to select and move to the numeration something of that category. In this case, it selects Infl:Asp:

(35) Numeration for *tu-nému-húma*, 'we are hitting', step 1:

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

In Kinande, all Infl heads have the selectional features [S:V] and [S:Agr]. These two features require that a V and an Agr be selected by L2N in steps 2 and 3 (again, either simultaneous or sequenced, in either order):

(36) Numeration for *tu-nému-húma*, 'we are hitting', step 2:

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

c. húma (V[S:N], 'hit')

(37) Numeration for *tu-nému-húma*, 'we are hitting', step 3:

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

c. húma (V[S:N], 'hit')

d. Agr

Finally, the verb selects for a subject argument, so pro is selected and put into the numeration:

(38) Numeration for *tu-nému-húma*, 'we are hitting', step 4:

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

c. húma (V[S:N], 'hit')

d. Agr

e. pro (N, [1Pl])

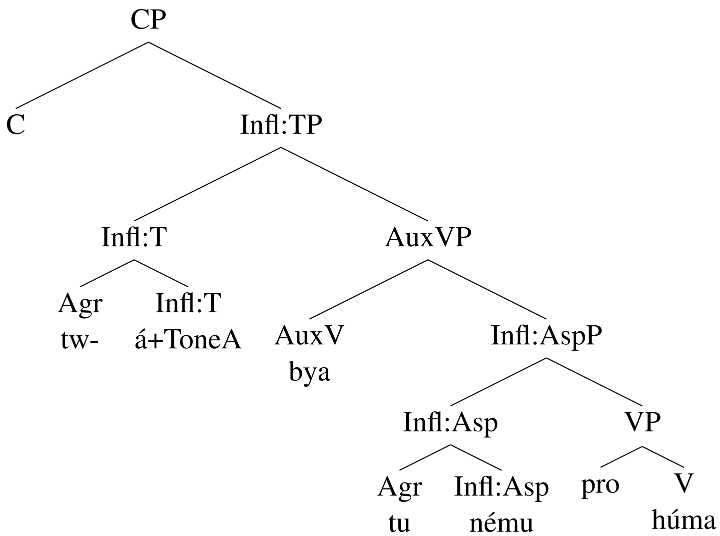
This completes the selection of items from the lexicon. The numeration is complete, as there are no more [S] features. There is one Infl element, one V, and one Agr, as Kinande requires. The syntax can now put the items from the numeration together to create the tree in (33).

Now we come to the case where we see the "overflow," the case of tense and aspect occurring together. Consider the following example:

(39) tw-á-bya i-tu-nému-húma
1Pl-Past3-be+ToneA LNK-1Pl-Prog-hit
'We were (recently, not today) hitting.' (Bjorkman 2011: 26, (11a))

I assume this has the following structure:

(40)



I assume that there is a subcategory of verbs, AuxVs, to which *bya* belongs. I also assume the AuxV *bya* moves to Infl:T and the main V *húma* moves to Infl:Asp. I will ignore the element glossed "LNK," as I have no information about the distribution of this element. What is important here is that each Infl morpheme co-occurs with one verb and one Agr head. This is what the analysis must capture. Whatever the "LNK" is, it should be possible to add it to the structure without changing relevant aspects of the analysis.

Recall that selection of items for the numeration begins with C and operates according to selection. The initial step will therefore be selection of C, which has the feature [S:Infl]:

(41) Numeration for *tw-á-bya i-tu-nému-húma*, 'we were (recently, not today) hitting', step 0:

a. Ø (C[S:Infl])

L2N now has to choose something of category Infl. Both Infl:T and Infl:Asp are of category Infl, and it could in principle choose either one. In fact it does not matter which one it chooses first here. Suppose it chooses Infl:Asp:

(42) Numeration for *tw-á-bya i-tu-nému-húma*, 'we were (recently, not today) hitting', step 1:

a. Ø (C[S:Infl])
b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

The two selectional features now require that a V and an Agr be put into the numeration. Suppose that the V that is chosen is the main (lexical) verb:

(43) Numeration for *tw-á-bya i-tu-némo-húma*, 'we were (recently, not today) hitting', steps 2 and 3 (to be revised):

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

c. húma (V[S:N], 'hit')

d. Agr

This numeration would end after selecting something of category N. This could therefore never be the numeration for a derivation that includes both Infl:T and Infl:Asp. The only way to include both of them is to choose an auxiliary verb in step 2 instead of the lexical verb:

(44) Numeration for *tw-á-bya i-tu-nému-húma*, 'we were (recently, not today) hitting', steps 2 and 3 (revised):

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

c. bya (AuxV[S:Infl])

d. Agr

As an AuxV, *bya* is category V and therefore satisfies the selectional feature of Infl:Asp. In Kinande, AuxVs select Infl, as shown. This is perhaps a defining feature of auxiliary verbs cross-linguistically. This feature now requires that another Infl head be moved to the numeration. In this case, Infl:T is now chosen:

(45) Numeration for *tw-á-bya i-tu-nému-húma*, 'we were (recently, not today) hitting', step 4:

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

c. bya (AuxV[S:Infl])

d. Agr

e. á+ToneA (Infl:T[S:V,S:Agr], 'Past3')

Once again, Infl heads in Kinande have two selectional features, so L2N now has to select a V and an Agr head to be put into the numeration. The lexical verb is selected, along with an Agr head:

(46) Numeration for *tw-á-bya i-tu-nému-húma*, 'we were (recently, not today) hitting', steps 5 and 6:

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

c. bya (AuxV[S:Infl])

d. Agr

e. á+ToneA (Infl:T[S:V,S:Agr], 'Past3')

f. húma (V[S:N], 'hit')

g. Agr

Finally, the lexical verb selects a subject, so L2N selects pro:

(47) Numeration for *tw-á-bya i-tu-nému-húma*, 'we were (recently, not today) hitting', step 7:

a. Ø (C[S:Infl])

b. nému- (Infl:Asp[S:V,S:Agr], 'Prog')

c. bya (AuxV[S:Infl])
d. Agr
e. á+ToneA (Infl:T[S:V,S:Agr], ‘Past3’)
f. húma (V[S:N], ‘hit’)
g. Agr
h. pro (N, [1Pl])

The numeration is now complete, and the derivation can begin. How the derivation works, and how N2W selects items from the numeration to move to the workspace, is not the topic of this paper. However, a few remarks are in order. As can be seen from the above, the order of selection of Infl:T and Infl:Asp for the numeration does not matter. Their order obviously does matter for the ultimate derivation. Something will have to ensure that Infl:T is selected prior to Infl:Asp for merger in the workspace. If selection is also what drives N2W, as would be the default assumption, then Infl:T and Infl:Asp must actually differ in their selectional features. Infl:T must select any type of V, [S:V], while Infl:Asp must select only main verbs, [S:MV]. Then Infl:T can merge with the auxiliary verb, which will then select Infl:Asp, but Infl:Asp cannot merge with the auxiliary verb. This will ensure that the order is Infl:T higher than Infl:Asp. If this is how the selectional features work in Kinande, then the order of selection for the numeration from the lexicon will also have to proceed in that order. Infl:T will have to be selected first, then AuxV, then Infl:Asp. If this is correct, then the above derivation must be revised slightly to incorporate the more nuanced features, and the strict ordering. Otherwise it will proceed as described above.

As can be seen, this analysis correctly derives the overflow pattern in Kinande. The requirement of one-to-one matching between Infl heads and verbs (and Agr heads) is exactly like the requirement of one-to-one matching between [+N] heads and Agr heads in Bulgarian. We see such requirements of one-to-one matching in many different contexts in many different languages. I have proposed that the way to model them is via selection. Selection is also what drives L2N, the procedure that selects items from the lexicon and puts them in the numeration. The basic idea is that each time L2N selects an item with a selectional feature, it must then select an item of of the selected category. This results in one-to-one matching.

It should also be noted that such one-to-one matching holds even in a language like English where the overflow pattern does not obtain. As Bjorkman (2011) discusses, in languages like English it appears that certain inflectional categories are always accompanied by certain auxiliaries. For instance, the progressive always requires *be* and the perfect always requires *have*. But note that even in such languages one-to-one matching obtains: there are as many verbs as there are inflectional categories. There is always a tense, in addition to any other inflectional categories:

(48) English one-to-one matching between inflectional categories and verbs
a. She run-s. (1 Infl: T; 1 verb)
b. She ha-s run. (2 Infl: T, Asp(Perf); 2 verbs)
c. She i-s running. (2 Infl: T, Asp(Prog); 2 verbs)
d. She ha-s been running. (3 Infl: T, Asp(Perf), Asp(Prog); 3 verbs)
e. She could have been running. (4 Infl: T, Mod, Asp(Perf), Asp(Prog); 4 verbs)
f. She might have been being passed. (5 Infl: T, Mod, Asp(Perf), Asp(Prog), Passive; 5 verbs)

English is actually behaving exactly like Kinande, despite initial appearances. One-to-one matching holds in a great many languages, even ones where we do not see an obvious overflow pattern.

It should also be noted that we see one-to-one matching in Kinande not just between Infl heads and verbs, but between Infl heads and Agr heads, too. This is something that the current analysis captures, but

Bjorkman's does not. Her analysis was only designed to capture the one-to-one matching between Infl heads and verbs, and it cannot be extended to the one-to-one matching between Infl heads and Agr heads.

Finally, it was important for the analysis of Kinande that null elements not be present. Consider the progressive aspect by itself again:

(49) tu-nému-húma
1Pl-Prog-hit
'We are hitting.' (Bjorkman 2011: 26, (10b))

It must be the case that present tense in this example is simply the absence of tense. It could not be the case that there is a null Infl:T. If there were, there would be two Infl heads, each of which selects a V, and we would expect an auxiliary to appear. Since there is no auxiliary (and there is only one Agr), there must be only one Infl head in an example like this.

7 English *Do*-Support

I turn now from the overflow pattern in Kinande to the much-discussed phenomenon of *do*-support in English. English *do*-support does not occur in simple declaratives, but does take place in Negative, Inversion, Contrast (or verum focus), and Ellipsis contexts ("NICE" contexts):

(50) *NICE contexts*
a. She studies too much.
b. She doesn't study too much. (Negation)
c. Does she study too much? (Inversion)
d. She DOES study too much. (Contrast or verum focus)
e. They said she would study too much, and she does. (Ellipsis)
They said she would study too much, and study too much she does.

I lump VP fronting with VP ellipsis as cases where the VP is not pronounced in its typical position (50e).

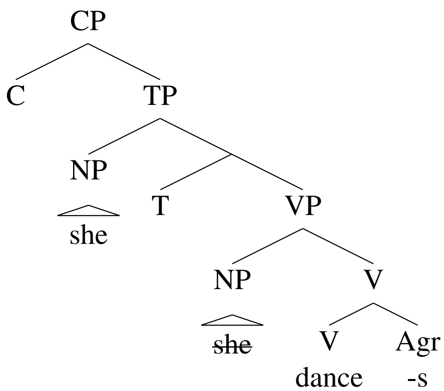
As mentioned above, *do*-support is typically analyzed as a last resort operation, following Chomsky (1957). In the typical analysis (e.g., Bobaljik 1995, Embick & Noyer 2001), the NICE contexts disrupt a required local relation between T(ense) and V (or a light verb v). The semantically contentless auxiliary *do* is inserted as a response (typically, to provide morphological support for the T affix, which, as an affix, cannot stand by itself). As described above, this requires that the syntax be able to insert things that were not present in the numeration. I therefore reject such analyses.

Other reasons have also been given to reject the last-resort analysis of *do*-support. Bruening (2010) argues that certain other grammatical phenomena are sensitive to the exact same NICE contexts, even when no do-support applies (e.g., in nonfinite clauses). This means that there must be a positive specification that all these contexts have in common. Following Baker (1991), Bruening (2010) claims they all have a [Special Purpose] ([SP]) feature. Contexts with a [SP] feature then require an auxiliary verb. A different argument is given by Bjorkman (2011): she argues that the distribution of *do*-support cross-linguistically is unexpected if it applies as a last resort operation to support a stranded T. I assume that the conclusion of these works is correct: *do*-support is not a last resort operation, and we need a different analysis, one where *do* is present in the numeration from the beginning.

I will adopt the [SP] analysis of the NICE contexts proposed by Baker (1991) and Bruening (2010). In this analysis, all of the NICE contexts have a T head that has the feature [SP]. We can then account for *do*-support in these contexts with nothing more than selection.

I will start with a simple declarative without *do*-support, like *She dances*. I will assume that this has the following structure:

(51)



English differs from Kinande in that all elements of category V select an Agr node, rather than elements of category Infl:

(52) In English, all elements of category V have the feature [S:Agr].

The Agr node on each V Agrees with the functional head that immediately c-commands it. In this case, that is T. T Agrees with the subject, and so this Agr node has the features of tense and the subject ([Pres,3Sg]). These features are realized by *-s* in English. Note that the constraint in (52) accounts for the one-to-one matching that was observed in section 6 (48): Every English verb has an Agr, which has to Agree with the functional element above it. Since all functional elements but one (T) are realized by verbs (AuxVs), this results in one verb for every functional element (inflectional heads like T, Mod, Asp, Voice).

I will modify the selectional features of T from (15) above. In particular, I will consider all of Pass, Prog, Perf, Mod to be subcategories of AuxVs. AuxVs and M(ain)Vs are subcategories of V. There are two different T heads in English, and Neg selects only AuxVs:

(53) a. T has the feature [S:V]
b. T[SP] has the feature [S:{Neg,AuxV}]
c. Neg has the feature [S:AuxV]

T without an [SP] feature selects only verbs (but either AuxVs or MVs). T with an [SP] feature selects only Neg or AuxV.

In (51), T is not [SP]. The derivation starts with L2N moving C to the numeration:

(54) Numeration for *she dances*, step 0:
a. Ø (C[S:T])

C selects T, so L2N then selects a T without an [SP] feature:

(55) Numeration for *she dances*, step 1:
a. Ø (C[S:T])
b. Ø (T[S:V])

T selects anything of category V, so either an AuxV or an MV. This clause does not have any semantically contentful auxiliary verbs, so a main verb is chosen and moved to the numeration:

(56) Numeration for *she dances*, step 2:

a. Ø (C[S:T])

b. Ø (T[S:V])

c. dance (MV[S:Agr,S:N])

The verb selects a subject and an Agr, so the final two steps are to move Agr and a subject to the numeration:

(57) Numeration for *she dances*, steps 3 and 4:

a. Ø (C[S:T])

b. Ø (T[S:V])

c. dance (MV[S:Agr,S:N])

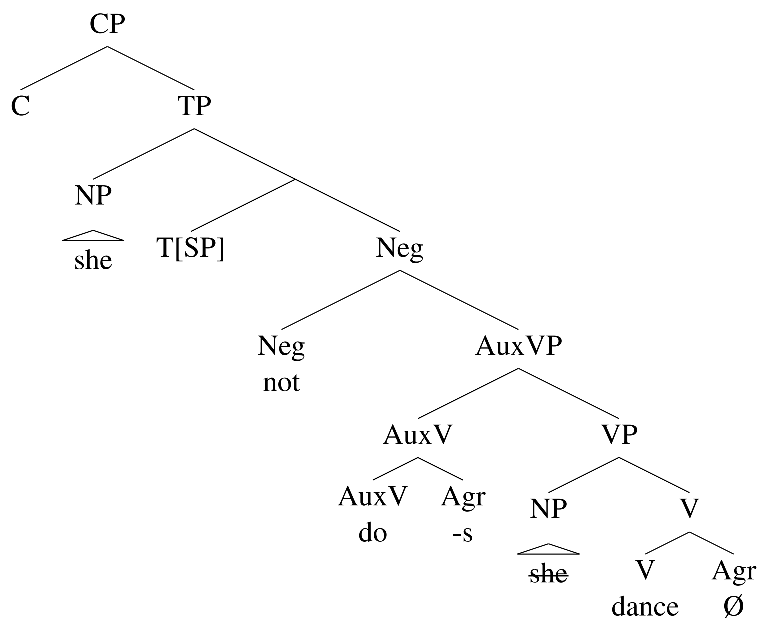
d. Agr

e. she (N)

This derivation is complete, and the syntax can proceed to put the items together in the workspace. No auxiliary verb is required.

Consider now a negative sentence, like *She does not dance*. I assume that this has the following structure:

(58)



As before, all elements of category V have an Agr adjoined to them. In English, the highest AuxV always moves to T, crossing Neg.

Recall that only T[SP] may select Neg (53). A T that does not have the [SP] feature may not select Neg. Neg always selects an AuxV. This will require *do* to be moved to the numeration if no semantically contentful auxiliary is chosen.

As before, selection of elements from the lexicon starts with C:

(59) Numeration for *she does not dance*, step 0:

a. Ø (C[S:T])

C selects for an item of category T, so L2N selects something of that category. If negation is to be selected as well, T[SP] has to be what is selected:

(60) Numeration for *she does not dance*, step 1:

a. Ø (C[S:T])
b. Ø (T[SP;S:{Neg,AuxV}])

T[SP] selects a disjunctive list of either Neg or AuxV. In this case, L2N selects Neg and moves it to the numeration:

(61) Numeration for *she does not dance*, step 2:
a. Ø (C[S:T])
b. Ø (T[SP;S:{Neg,AuxV}])
c. not (Neg[S:AuxV])

In English, Neg selects only AuxV. L2N then has to select an AuxV for the numeration. This clause will not include any semantically contentful AuxVs, so L2N has to select the semantically contentless *do*:

(62) Numeration for *she does not dance*, step 3:
a. Ø (C[S:T])
b. Ø (T[SP;S:{Neg,AuxV}])
c. not (Neg[S:AuxV])
d. do (AuxV[S:MV,S:Agr])

The auxiliary *do* selects only main verbs ([S:MV]). Like all elements of category V, it also selects an Agr. So in steps 4 and 5, an Agr and a main verb are chosen:

(63) Numeration for *she does not dance*, steps 4 and 5:
a. Ø (C[S:T])
b. Ø (T[SP;S:{Neg,AuxV}])
c. not (Neg[S:AuxV])
d. do (AuxV[S:MV,S:Agr])
e. Agr
f. dance (MV[S:Agr,S:N])

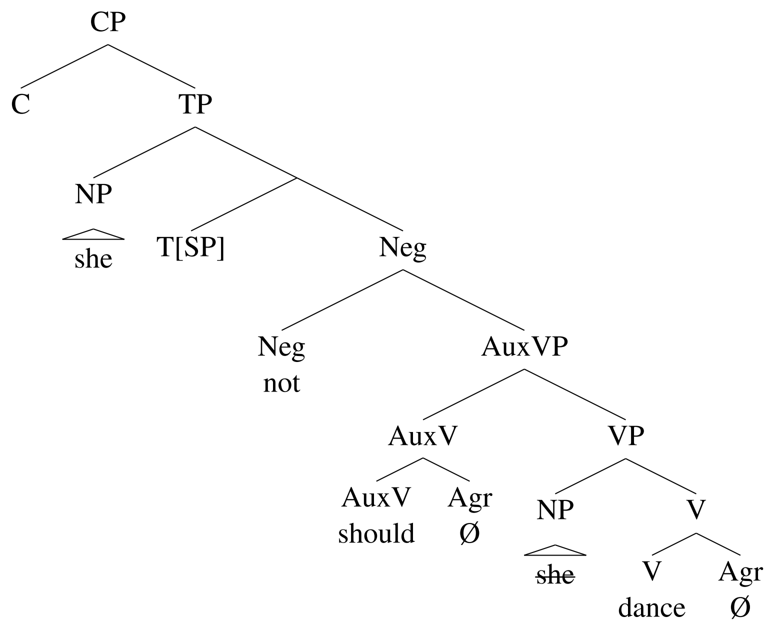
The main verb selects an Agr and a subject, as before, so selection of items from the numeration completes with those two items:

(64) Numeration for *she does not dance*, steps 6 and 7:
a. Ø (C[S:T])
b. Ø (T[SP;S:{Neg,AuxV}])
c. not (Neg[S:AuxV])
d. do (AuxV[S:MV,S:Agr])
e. Agr
f. dance (MV[S:Agr,S:N])
g. Agr
h. she (N)

This numeration can now be put together by the syntax, as (58). The Agr adjoined to *do* Agrees with T, which Agrees with the subject (Neg is inert for Agree). The Agr adjoined to *dance* Agrees with *do*, and so is spelled out as null (since *do* is like the modals in this respect).

Consider an example with a contentful auxiliary like *she should not dance*. This will have the following structure:

(65)



Agr on modals is always null, in English.

Selection of items from the lexicon will proceed exactly as in the previous example, up through step 3. In that step, a modal auxiliary is chosen instead of *do*:

(66) Numeration for *she should not dance*, step 3:

a. Ø (C[S:T])

b. Ø (T[SP;S:{Neg,AuxV}])

c. not (Neg[S:AuxV])

d. should (AuxV[S:{MV,$\text{AuxV}_{\text{Pass}}$,$\text{AuxV}_{\text{Prog}}$,$\text{AuxV}_{\text{Perf}}$},S:Agr])

This satisfies the selectional requirement of *not*, so there is no need for any other auxiliary. Selection of items then proceeds exactly as in the previous example (modals select any of the other AuxVs or an MV, in this example an MV is chosen).

As can be seen from the above, selection by itself accounts for *do*-support in negative clauses and its absence in simple positive declaratives. There is no need at all for a syncategorematic last resort insertion mechanism.

Let me now go on to show how the other NICE contexts are accounted for, starting with subject-auxiliary inversion. In an inversion context like *Does she dance?*, I assume that L2N selects a C head from the lexicon that has the feature that triggers T-to-C movement, call it [SAI]. The C that has this feature also necessarily has an [S:T[SP]] feature, meaning that it obligatorily selects an [SP] T. Since T[SP] has the feature [S:AuxV] and does not have the feature [S:MV] (53b), an auxiliary verb will then have to be selected for the numeration. If no semantically contentful auxiliary is selected, then L2N has to select *do* again and put it in the numeration.

In a context of emphasis or verum focus like *She DOES dance!*, I assume that a verum focus T is also [SP]. (Alternatively, T[SP] may select for some other element that encodes verum focus, for instance the head Σ proposed by Laka Mugarza 1990. This head would then only select for AuxVs, the same way Neg does.) Since T[SP] obligatorily selects an AuxV and not an MV, an auxiliary verb must again be moved to the numeration. If no semantically contentful one is desired, then L2N must again select the contentless AuxV *do*.

VP fronting and VP ellipsis are slightly trickier. The exact analysis of these processes is not so important, so I will only sketch the barest account. Perhaps the simplest account of VP fronting is that VP fronting is triggered by a C with a special feature that triggers fronting of a predicate. This C, like the one that triggers T-to-C movement, also selects T[SP]. Then the numeration will be required to have an AuxV again, since T[SP] selects AuxV and not MV. As for VP ellipsis, we can assume that it is licensed by a special feature on T, something like the [E] feature of Merchant (2001). A T that has an [E] feature to license ellipsis of a phrase within its complement then also has an [SP] feature. Since the T in a numeration that will include ellipsis has an [SP] feature, the numeration must also include an AuxV, since T[SP] selects AuxV and does not select MV. If no AuxV is present for independent reasons, L2N will again be forced to select *do* for the numeration.

This analysis therefore accounts for *do*-support without allowing the syntax to be able to insert elements that were not present in the numeration. *Do* is present in the numeration in this analysis. Note that the analysis still has something of a last resort character, in that selection of *do* is only triggered if no other auxiliary is independently selected. However, it is not triggered by what happens in the syntax, but by selectional features in the process of choosing a numeration. The mechanism that chooses elements for the numeration, L2N, chooses *do* on the same basis as it chooses all elements, namely, on the basis of selectional features.

Note also that the distribution of the [SP] feature that triggers *do*-support in this analysis is language-specific. We expect that other languages could have slightly different contexts that require the presence of an AuxV. The analysis can therefore be extended easily to the other cases of *do*-support discussed in the literature (see Bjorkman 2011 for a survey). In Monnese, for example, *do*-support applies only in matrix interrogatives with T-to-C movement (Benincà & Poletto 2004). In the current analysis, we would say that in this language, only the C that triggers T-to-C movement selects a T with an [SP] feature. Otherwise, things work very similarly to how they do in English (modulo language-particular requirements for Agr and so on). In Breton, a verb-second language, *do*-support applies only when the first constituent is the main verb (Jouitteau 2010). This appears to be the default, discourse-neutral word order in a main clause. The precise analysis of such a state of affairs is probably very tricky and will require much more research, but a possible approach is to say that the V2 C head that triggers T-to-C movement but does *not* have any topic or focus features obligatorily selects an [SP] T. For some reason, the way to satisfy the need for a constituent to the left of C in Breton in such a case is to front the verb. The fact that the T in such a discourse-neutral CP phase has the feature [SP] then requires the Breton *do* in the numeration, since Breton T[SP] is exactly like English in selecting an AuxV and not a MV. As for Danish, Norwegian, and Swedish, the other languages cited by Bjorkman (2011), the distribution of *do*-support described in Houser et al. (2011) does not have exactly the same last-resort character as in the other languages. It is optional with certain other auxiliaries, for one thing. I therefore leave open whether it should be accounted for along the lines proposed here, or in some other way (for instance, the analysis that Houser et al. 2011 propose, where *do* selects a pronominal VP complement).

8 Conclusion and Discussion

Most research on syntax has never bothered to spell out how exactly the syntax selects items from the lexicon. This paper has begun to explore this issue with an investigation of how the input operation L2N selects items from the lexicon and puts them in the numeration. It has only touched on some of the issues involved, but it has shown that spelling out this procedure can shed light on synthetic-periphrastic alternations like *do*-support and the "overflow" pattern of auxiliaries. The mechanism involved is nothing more than selection, and the way selection works in the present case is also necessary for phenomena like nominal concord. An important empirical point is that many languages require one-to-one matching between certain inflectional heads and certain other heads. For instance, in the nominal domain, heads like adjectives, numerals, and nouns all require an Agr head. In the clausal domain, there must be one-to-one matching between inflectional heads and verbal heads in many languages. Many languages also require one-to-one matching between either inflectional heads and Agr heads, or verbs and Agr heads. The current account captures these requirements, and shows how we can explain synthetic-periphrastic alternations without last resort insertion operations.

One final point is that, for the phenomena discussed here, the numeration is actually superfluous. The exact same mechanisms and analyses proposed above would work just as well skipping the numeration and merging items from the lexicon directly into the workspace. However, in companion work on the operation that moves items from the numeration to the workspace (N2W), it will be crucial that there is a numeration and that the workspace cannot access the lexicon directly. For this reason, I have spelled out the analysis here making use of the numeration. Nevertheless, it is worth noting that the numeration itself is not crucial in accounting for synthetic-periphrastic operations, it is the selection mechanism that is crucial. It does not matter whether this selection mechanism puts the selected items into a numeration, or merges them directly in the workspace.

References

Adger, David. 2010. A Minimalist theory of feature structure. In Anna Kibort & Greville G. Corbett (eds.), *Features: Perspectives on a key notion in linguistics*, 185–218. Oxford: Oxford University Press.

Adger, David & Peter Svenonius. 2011. Features in minimalist syntax. In Cedric Boeckx (ed.), *The Oxford handbook of linguistic minimalism*, 27–51. Oxford: Oxford University Press.

Baker, C. L. 1991. The syntax of English *not*: The limits of core grammar. *Linguistic Inquiry* 22. 387–429.

Benincà, Paula & Cecilia Poletto. 2004. A case of *do*-support in Romance. *Natural Language and Linguistic Theory* 22. 51–94.

Bjorkman, Bronwyn Moore. 2011. *BE-ing default: The morphosyntax of auxiliaries*. Massachusetts Institute of Technology dissertation.

Bobaljik, Jonathan David. 1995. *Morphosyntax: The syntax of verbal inflection*. Massachusetts Institute of Technology dissertation. Distributed by MIT Working Papers in Linguistics, Cambridge, Mass.

Bobaljik, Jonathan David. 2012. *Universals in comparative morphology*. Cambridge, MA: MIT Press.

Bruening, Benjamin. 2010. Language-particular syntactic rules and constraints: English locative inversion and *Do*-support. *Language* 86. 43–84.

Bruening, Benjamin. 2013. By-phrases in passives and nominals. *Syntax* 16. 1–41.

Bruening, Benjamin, Xuyen Dinh & Lan Kim. 2018. Selection, idioms, and the structure of nominal phrases with and without classifiers. *Glossa: A Journal of General Linguistics* 3. 1–46. doi:10.5334/gjgl.288.

Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1995. *The minimalist program*. Cambridge, MA: MIT Press.

Chomsky, Noam. 2000. Minimalist inquiries: The framework. In Roger Martin, David Michaels & Juan Uriagereka (eds.), *Step by step: Essays on minimalist syntax in honor of Howard Lasnik*, 89–155. Cambridge, MA: MIT Press.

Cinque, Guglielmo. 1999. *Adverbs and functional heads*. Oxford: Oxford University Press.

Corver, Norbert. 1997. *Much*-support as a last resort. *Linguistic Inquiry* 28. 119–164.

Cowper, Elizabeth. 2010. Where auxiliary verbs come from. In Melinda Heijl (ed.), *Proceedings of the 2010 annual conference of the Canadian Linguistic Association*, http://homes.chass.utoronto.ca/~cla--acl/actes2010/CLA2010_Cowper.pdf. Toronto: University of Toronto.

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

Embick, David & Rolf Noyer. 2001. Movement operations after syntax. *Linguistic Inquiry* 32. 555–595.

Franks, Steven. 2001. The internal structure of Slavic NPs, with special reference to Bulgarian. In Adam Przepiórkowski & Piotr Bański (eds.), *Generative linguistics in Poland: Syntax and morphosyntax*, 53–69. Warszawa: Instytut Podstaw Informatyki PAN.

Halle, Morris & Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In Kenneth Hale & Samuel Jay Keyser (eds.), *The view from building 20: Essays in linguistics in honor of Sylvain Bromberger*, 111–176. Cambridge, MA: MIT Press.

Harizanov, Boris & Vera Gribanova. 2014. Inward-sensitive contextual allomorphy and its conditioning factors. In Hsin-Lun Huang, Ethan Poole & Amanda Rysling (eds.), *Proceedings of the 43rd annual meeting of the north east linguistic society (nels)*, 155–166. Amherst, MA: GLSA.

Houser, Michael J., Line Mikkelsen & Maziar Toosarvandani. 2011. A defective auxiliary in Danish. *Journal of Germanic Linguistics* 23. 245–298.

Jouitteau, Mélanie. 2010. Post-syntactic excorporation in realizational morphology: Evidence from Breton. Ms., available at http://ling.auf.net/lingBuzz/001169.

Kratzer, Angelika. 1996. Severing the external argument from its verb. In John Rooryck & Laurie Zaring (eds.), *Phrase structure and the lexicon*, 109–137. Dordrecht: Kluwer.

Laka Mugarza, Itziar. 1990. *Negation in syntax: On the nature of functional categories and projections*. MIT dissertation. Distributed by MIT Working Papers in Linguistics, Cambridge, MA.

Matushansky, Ora. 2013. More or better: On the derivation of synthetic comparatives and superlatives in English. In Ora Matushansky & Alec Marantz (eds.), *Distributed morphology today: Morphemes for Morris Halle*, 59–78. Cambridge, MA: MIT Press.

Merchant, Jason. 2001. *The syntax of silence: Sluicing, islands, and the theory of ellipsis*. Oxford: Oxford University Press.

Pietraszko, Joanna. 2017. *Inflectional dependencies: A study of complex verbal expressions in Ndebele*. University of Chicago dissertation.

Rizzi, Luigi. 1997. The fine structure of the left periphery. In Liliane Haegeman (ed.), *Elements of grammar*, 281–337. Dordrecht: Kluwer.

Department of Linguistics and Cognitive Science
University of Delaware
Newark, DE 19716
(302) 831-4096
bruening@udel.edu