Synthetic-Periphrastic Alternations without Last Resort: The "Overflow" Pattern of Auxiliaries

Benjamin Bruening (University of Delaware)

rough draft, January 28, 2024; comments welcome

Abstract

Many approaches to synthetic-periphrastic alternations posit last resort insertion of some item, like *do* in English do-support, to repair some syntactic deficiency. I show that in the "overflow" pattern of auxiliaries in Kinande and other languages (Bjorkman 2011, Pietraszko 2017), there is no need for a last resort operation. If we view structure building as being driven largely by selection, and spell out how that might work, we find that the pattern falls out from the structure-building operations that are needed anyway. The same is true for do-support in English and other languages. This means that our model of syntax does not need to countenance a last resort insertion operation that is different from the mechanism that merges items generally.

1 Introduction

There are many examples in many different languages of synthetic-periphrastic alternations. English do-support is a classic example:

(1) 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 verb (*doe-s*), while the main verb is in its uninflected form.

Many approaches to synthetic-periphrastic alternations like do-support have posited *last resort* insertion of an item like English *do* (following Chomsky 1957). The idea is that, 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, using a mechanism different from what builds the rest of the structure. I view this type of approach as undesirable. It would be better if the mechanism that inserted *do* was nothing more than the mechanism that inserted all the other lexical items in the clause, operating in the same way and on the basis of the same type of information.[1]

[1] Last-resort insertion operations are even more problematic if one assumes the concept of the *numeration* (Chomsky 1995), which is a pre-selection of lexical items that the syntax uses to build phrases. The whole idea of the numeration is that the syntax cannot access the lexicon directly, it can only work with what was provided to it in the numeration. Last-resort operations like *do*-insertion violate the whole spirit of the numeration and are conceptually incompatible with it. Allowing them 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. Even limiting last resort 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 that is the case, then we do not want the syntax to be able to bypass the numeration and insert semantically contentless elements that were not present in the numeration.

In this paper, I show that this can indeed be the case. I focus on the "overflow" pattern of auxiliaries described by Bjorkman (2011) in Kinande and other languages. This pattern has the following characterization. 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):

(2) 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:

(3) 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 verb must be used:

(4) 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 verb 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 pattern 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. I first ask what drives selection of items from the lexicon for merger into the syntax in the first place, and suggest that the answer is syntactic selection. I spell out how this might work. It has to work in a top-down fashion, since that is how selection works. In Kinande, the inflectional elements T and Asp both select Vs. If T selects a main verb, the clause then cannot also include Asp, since most main verbs do not select Asp. In order to include Asp as well, T has to instead select an auxiliary verb, which does select Asp. Asp can then select a main verb. In spelling out how this works, I show that it does not require any mechanisms other than what are required anyway. We do not need a last resort insertion operation that differs from other structure-building operations.

I start by motivating syntactic selection as the driver of merge in syntax (section 2). Section 3 presents my analysis of the overflow pattern in Kinande. This section also shows that the analysis works for the other languages, Ndebele, Standard Arabic, and Latin. In section 4, I briefly discuss English do-support, as well as do-support operations in other languages. An analysis of do-support that does not treat it as last resort has already been proposed, by Baker (1991) and Bruening (2010). I extend this analysis to other languages that have been described as having a do-support type operation.

While the cases discussed here are by no means the only instances of synthetic-periphrastic alternations, the success and simplicity of the selection-driven account raises the hope that all such alternations can be analyzed in a similar fashion. If so, we can do without last-resort insertion mechanisms altogether.

2 Syntactic Selection Drives Structure Building

I start with the question of how the syntax selects items from the lexicon and merges them into the syntax.[2] Most researchers agree that syntactic selection plays an important role, particularly for lexical items like verbs. For instance, if the verb is one like *think* that selects for a CP complement, then the syntax will merge something of category C as the complement of the V. If the verb is instead intransitive and does not select a complement, then the syntax will not merge anything. In other words, selectional requirements of lexical items drive merge. Many researchers have formalized this as feature checking, with lexical items having selectional features that are checked off by merging something of the appropriate type (see Svenonius 1994, Collins 2002, Adger 2003, Bruening 2013, Müller 2017, among many others). However, within the functional domain of the clause, it is less clear that this is how the syntax operates. One proposal is that it instead operates on the basis of a *Hierarchy of Projections*.

2.1 Hierarchy-of-Projections Merge?

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 use for explication 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:

(5) (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, an item selects a category 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):

(6) 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)

[2] I do not adopt the notion of the numeration (Chomsky 1995) here, but one could. If one did, then what is described here would be either selection of items from the numeration to be merged into the syntax, or selection of items from the lexicon to be put into the numeration (or both). I do not use the numeration here because it is unnecessary and introduces considerable redundancy.

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:

(7) 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 adopt the view that HoP-Merge is how the syntax builds functional structure: it starts with something high on the 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, Mod, or Voice. 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 or even most cases of selection. For instance, many (most?) verbs can select multiple categories, but only one at a time. The verb *explain* selects for CPs or NPs, but not both simultaneously:

(8) a. She explained [$_{\text{NP}}$ the strange results].
b. She explained [$_{\text{CP}}$ that her new theory was not fully developed yet].
c. * She explained [$_{\text{NP}}$ the strange results] [$_{\text{CP}}$ that her new theory was not fully developed yet].

Explain does not allow other categories like APs or VPs, but it does allow a raising to object or ECM complement:

(9) a. * She explained [$_{\text{AP}}$ new].
b. * She explained [$_{\text{VP}}$ develop a theory].
c. Someone once explained it to be like a pie.

We therefore need to say that *explain* selects for *one* of NP, CP, or whatever category raising to object is, which is exactly disjunctive selection. There is no way around disjunctive selection for verbs. 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, where "Z" is whatever category raising to object is:

(10) *explain* has the feature [S:X$\in${N, C, Z}].

The [S] feature will be satisfied by merger of some X where X is one member of the set.

2.2 Disjunctive Selection Captures the Hierarchy of Projections

The effects of HoP-Merge can then be captured very straightforwardly using the same kind of disjunctive selection. For purposes of explication, we could simply revise Adger's hierarchy to use disjunctive selection (I ignore Adger's v and Fin, since it is not clear what English elements correspond to them):

(11) a. Pass has the feature [S:V].

b. Prog has the feature [S:X∈{V, Pass}].

c. Perf has the feature [S:X∈{V, Pass, Prog}].

d. Mod has the feature [S:X∈{V, Pass, Prog, Perf}].

e. Neg has the feature [S:X∈{V, Pass, Prog, Perf, Mod}].

f. T has the feature [S:X∈{V, Pass, Prog, Perf, Mod, Neg}].

g. C has the feature [S:T].

This would have the desired effect of permitting T to combine with any of the other categories except C, while C would only be able to combine with T.

Because the analysis of auxiliary verbs is important to the goal of this paper, I will not simply transfer Adger's hierarchy to a disjunctive selection analysis, I will revise it. I assume that the categories Pass, Prog, Perf, Mod are borne by auxiliary verbs (AuxVs) in English. AuxVs are a subcategory of verbs. They differ from main verbs in that they are a closed class of grammatical verbs that carry tense, mood, aspect, and voice categories, either inherently or morphologically. In English, the AuxVs have the semantics of these categories inherently. *Have* is Perf, Pass and Prog are both realized by *be* (but with different morphological marking on the complement), and Mod is the set of modal auxiliaries (*can, should*, etc.). I also assume that Neg is not a head in the clausal hierarchy, but an adjunct (see Bruening 2010); it therefore does not belong in this list. I revise the selectional statements as follows:

(12) a. $\text{AuxV}_{\text{Pass}}$ has the feature [S:MainV].

b. $\text{AuxV}_{\text{Prog}}$ has the feature [S:X∈{MainV, $\text{AuxV}_{\text{Pass}}$}].

c. $\text{AuxV}_{\text{Perf}}$ has the feature [S:X∈{MainV, $\text{AuxV}_{\text{Pass}}$, $\text{AuxV}_{\text{Prog}}$}].

d. AuxV_{Mod} has the feature [S:V].

e. T has the feature [S:V].

f. C has the feature [S:T].

$\text{AuxV}_{\text{Prog}}$ and $\text{AuxV}_{\text{Perf}}$ select a disjunctive list, but $\text{AuxV}_{\text{Pass}}$ strictly selects MainV. C strictly selects T. T strictly selects V, but since MainVs and AuxVs are subcategories of V, T can select any main verb or any of the AuxVs. Same for AuxV_{Mod}. If something prevents more than one AuxV_{Mod}, then we do not need to list the other AuxVs for AuxV_{Mod}'s selection.[3]

These statements of selection correctly capture the effects of HoP-Merge, but they do not run afoul of the problem of permitting C to directly combine with lower categories. The selectional statements in (12) also correctly rule out the violations in (7), since $\text{AuxV}_{\text{Perf}}$ is not on the list of things selected by $\text{AuxV}_{\text{Prog}}$ in (7a), and AuxV_{Mod} is not on the list of things selected by $\text{AuxV}_{\text{Perf}}$ in (7b).

Disjunctive selection, then, captures the Hierarchy of Projections without a Hierarchy of Projections. It is also independently necessary, for verbs like *explain* 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 multiplies theoretical devices unnecessarily.

One could complain that the selectional statements in (12) are stipulative, and they are. However, they are no more stipulative than stipulating an extrinsic hierarchy, which is what the Hierarchy of Projections

[3] At this point it is not clear what prevents more than one of a particular category, but it is clear that something does. Whatever it is cannot be categorical, because some dialects of English allow multiple modals. I will have to leave this to future research.

analysis does. The selectional statements basically recreate the Hierarchy of Projections, but without an extrinsic hierarchy, and without a second mode of Merge. In that sense they are doing better. I believe the hope with the Hierarchy of Projections has always been that something will explain why the hierarchy is the way it is. However, no one to my knowledge has proposed any explanation for it, all they have done is describe it (hence the "mapping" metaphor of the Cartography approach). If that hope is ever realized, then whatever the explanation is can probably be couched within the disjunctive selection view just as well as it can be in the Hierarchy of Projections view.

2.3 How Structure Building Proceeds

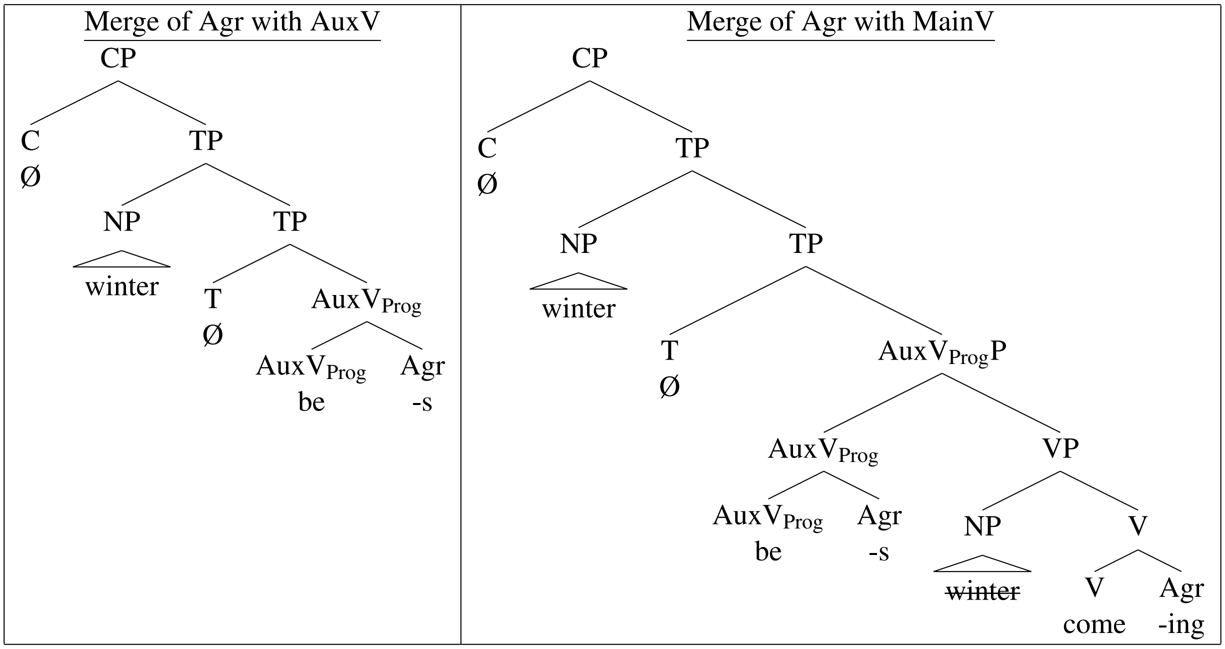
I conclude that the basis on which the syntax chooses items to merge into the structure being built is selection. Since selection typically works top-down, with each head selecting its complement, I will assume that that is how the structure is built, as well, though I do not think this is crucial for the purposes of this paper. Take a simple CP, like the main clause CP below:

(13) Winter is coming.

The syntax has to start somewhere. We can assume that this matrix clause is headed by a null finite declarative C head, and that the syntax always starts by choosing a C. It selects the null finite declarative C head from the lexicon and merges it into a workspace. Finite C selects finite T, so the syntax then selects finite T and merges it with the C. In this case, it chooses a present tense value for T. T selects any member of category V. In this case, $\text{AuxV}_{\text{Prog}}$ is chosen, which in English is realized as the auxiliary verb *be*. $\text{AuxV}_{\text{Prog}}$ then selects one of MainV or $\text{AuxV}_{\text{Pass}}$; in this case a MainV is chosen and merged. Note that, in addition to its complement, T also selects an NP specifier; this had to be merged at the appropriate place, and then copied into VP, since V also selects an NP (V assigns it a semantic interpretation, while T does not).

Note also that every verb in English, both AuxVs and MainVs, has an inflectional suffix on it. Call this Agr. I will capture this by saying that every member of category V also selects an Agr. We need to distinguish this type of selection from complement selection. I will notate it "S_{H}:Agr," indicating that it is selection for a head. An [S_{H}] feature will be checked off on the head that has it by merging a head of the appropriate type with it to form a complex head. [S_{H}] features have to be satisfied before complement selectional features, so in our example an Agr would be merged with $\text{AuxV}_{\text{Prog}}$ before the MainV is, and at the end an Agr would be merged with the MainV:

(14)



I assume that the form of each Agr is determined by the head that takes its host as complement. So the Agr on the MainV is determined to be *-ing* by AuxV$_{Prog}$, and the Agr on AuxV$_{Prog}$ is determined to be *-s* by T (*be-s* is pronounced *is*). We can model this determination as Agree Chomsky (2000), but any other mechanism will work just as well.

There is obviously much more to be said, for instance to incorporate adjuncts into the system, but this will be sufficient to start analyzing the overflow pattern. We need syntactic selection; and syntactic selection is sufficient for merging structure in the clause. The syntax will choose items to merge into the structure being built on the basis of selection (or more accurately, on the basis of selectional features).

3 Analysis of the Overflow Pattern in Kinande

We can now account for the overflow pattern in Kinande with nothing more than syntactic selection. To remind the reader, Kinande can mark T on the main verb:

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

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

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

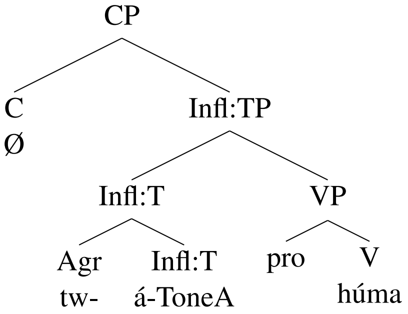
However, Kinande cannot mark both T and Asp on the V at the same time. Instead the auxiliary verb 'be' must be used in addition to the main verb. The auxiliary verb bears T, while the main verb bears Asp:

(17) 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))

3.1 Tense by Itself

I start with past tense by itself, using the example in (15). 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. 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:

(18)



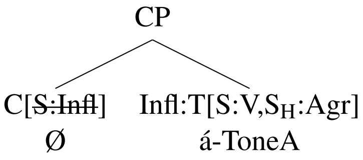
Whether the V moves to Infl to combine with it and Agr as a complex head is not important here; Bjorkman (2011) argues that it does not. I will not include such movement in the derivations. I will also assume for purposes here that all NP arguments are arguments of the main verb. Here, there is a null 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 (see Bruening 2021 on implicit objects of verbs, as in this example the logical object seems to be implicit).[4]

As stated above, the syntax begins the derivation by selecting a root C. In this case it chooses the null matrix declarative finite C. This C does not select a specifier. It is merged into the workspace:

(19) C[S:Infl]
Ø

I assume that in Kinande, C has the feature [S:Infl], as shown. The syntax therefore next selects something of category Infl. In this case it chooses an Infl:T head, and merges it with the C that is already present in the workspace:

(20)

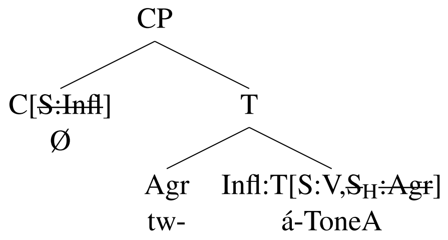


This checks off the selectional feature of C.

Infl:T also has a selectional feature. It selects for something of category V. It is also the case that every Infl head in Kinande has to have an Agr head. As explained above, this is head selection, notated "S_H:Agr." As described above, an [S_H] feature is checked off on the head that has it by merging a head of the appropriate type with it to form a complex head. [S_H] features have to be satisfied before complement selectional features, so the syntax will next merge an Agr head with Infl:T:

[4] It would of course be possible to add a null Voice or v head to introduce the external argument, following e.g. Kratzer 1996. The analysis would have to be complicated slightly: Infl:T would have to have the feature [S:X∈{Voice,AuxV}]. Selecting Voice would lead to a main verb. Infl:Asp would have the feature [S:Voice], which always leads to a main verb. Otherwise, nothing would change.

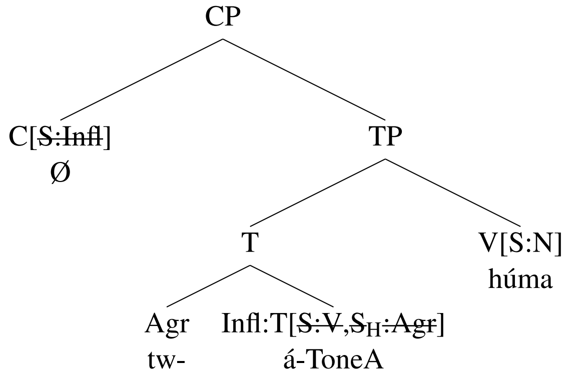
(21)



Agr will agree with the subject of the verb, and take its form based on the features it thereby acquires. Alternatively, Agr comes from the lexicon with certain features, and has to check them against the subject in the syntax (as in, e.g., Chomsky 1993). How this works is unimportant here; any theory of agreement will do.

The syntax will now choose something of category V to merge, because Infl:T has an [S:V] feature. In this case, it chooses the verb meaning 'hit.' This verb is merged with the complex T head as its complement:

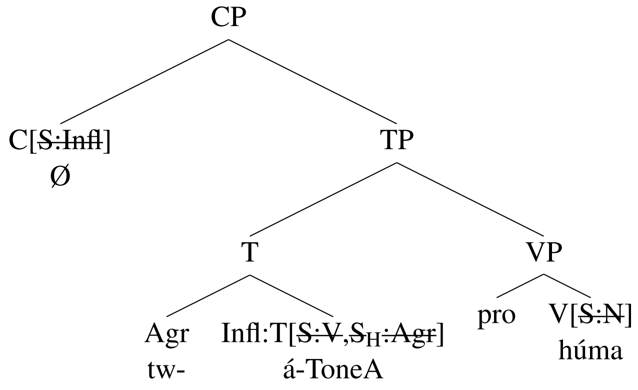
(22)



This checks off the [S:V] feature on Infl:T.

This verb selects a subject ([S:N]), so the syntax must select and merge something that can satisfy this selectional requirement (or build one by selecting multiple things). In this case it selects an unpronounced pronoun, pro. Since the structure of this NP is irrelevant here, I will just assume that it is a single lexical item that the syntax can pick out of the lexicon and merge with the V:

(23)

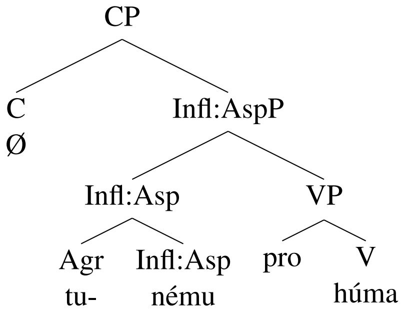


This checks off the verb's selectional feature. There are now no more unchecked selectional features in the structure. The syntax therefore terminates. It has built the correct structure, using nothing but syntactic selection. No auxiliary verb is required, only a main verb, since there is exactly one head (Infl:T) that selects a V.

3.2 Aspect by Itself

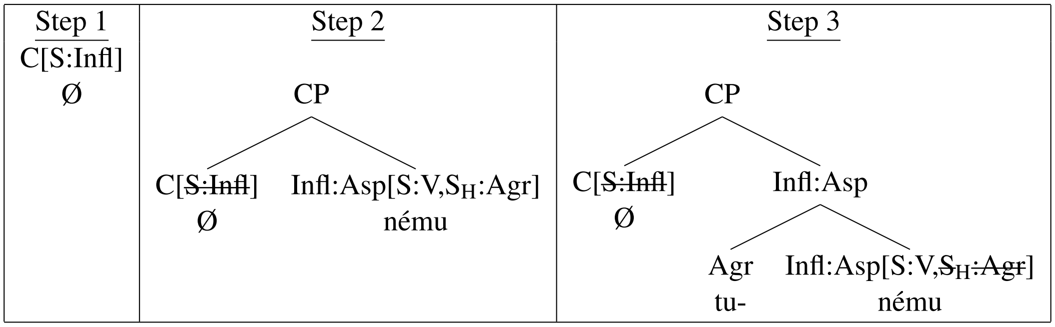
A similar derivation will take place when aspect occurs by itself, as in example (16) above. Again eschewing null elements other than C and pro, I assume that example (16) has the following structure:

(24)



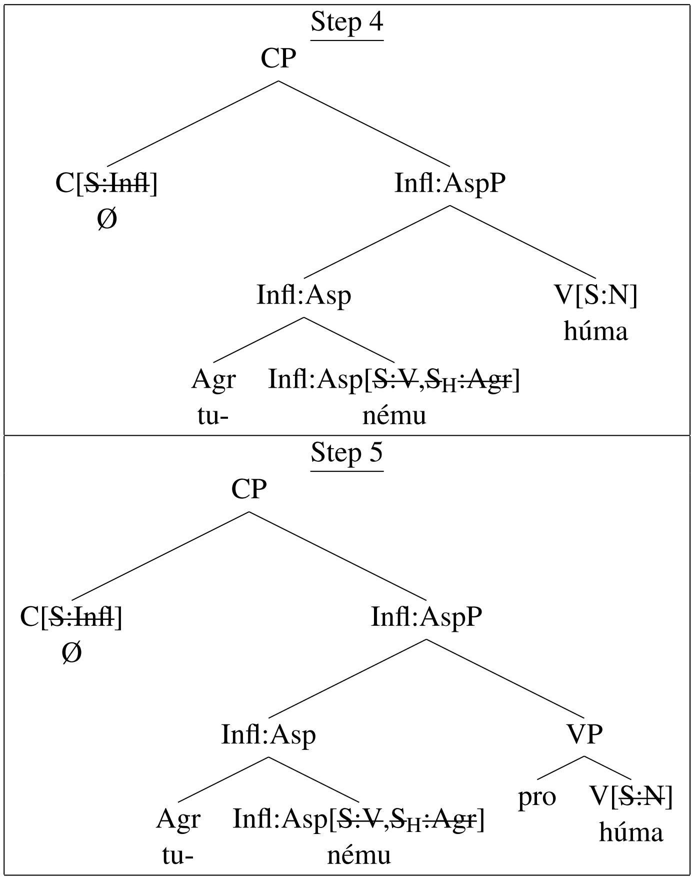
Selection of items for merger into the structure proceeds exactly as in the derivation of the simple past tense. It begins with selection of the null matrix finite declarative C (Step 1):

(25)



C selects something of category Infl, so the syntax then has to select and merge something of that category. In this case, it selects Infl:Asp (Step 2). Infl:Asp, like all Infl heads in Kinande, head-selects an Agr, so the syntax next merges an Agr head (Step 3). Infl:Asp also selects a complement of category V, so the next step is for the syntax to choose a V and merge it (Step 4):

(26)



Finally, V selects an argument of category N, so the syntax must merge something of that category. In this case it merges null pro (Step 5). This checks off the [S:N] feature of the V. There are no more unchecked [S] features in the structure, so the syntax terminates. Once again, no auxiliary verb is required, because there is only one element that selects category V, and it was satisfied by merging the main verb.

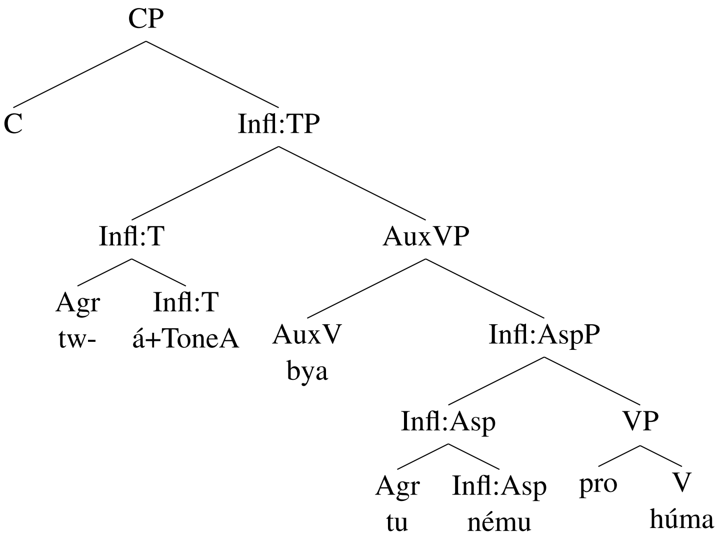
3.3 Tense and Aspect Together (“Overflow”)

Now we come to the case where we see the “overflow,” the case of tense and aspect occurring together. I repeat the example below:

(27) 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:

(28)



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.

3.4 More on Auxiliary Verbs

At this point it is important to say more about AuxVs. As described in section 2, AuxVs are a subcategory of V. They are grammatical verbs whose purpose is to permit the expression of Infl categories like T, Asp, Mod, Voice in a given language. They do that by carrying inflectional categories, either inherently, or morphologically. Recall that in English, the AuxVs bear inflectional categories inherently (so, *have* is Perf). In contrast, in Kinande, the auxiliary 'be' does not inherently express any inflectional category. It enables the expression of Infl categories by bearing an inflectional affix, which is what carries the semantics of the Infl category. The AuxV is necessary because there is a requirement in Kinande that each inflectional affix has to attach to a verb, and moreover, every verb can bear only one inflectional affix. So if there are two Infl categories, there must be two verbs. This means that an auxiliary verb must be used in addition to the main verb.

Since syntactic selection is what drives merger of elements in the syntax, AuxVs have to have the appropriate features to enable the required stacking of Infl categories. Since there seem to be two subcategories of Infl in Kinande, T and Asp, the feature that will enable stacking is [S:Asp]. C can select Infl:T, and Infl:T can then select 'be', which will then select [S:Asp].

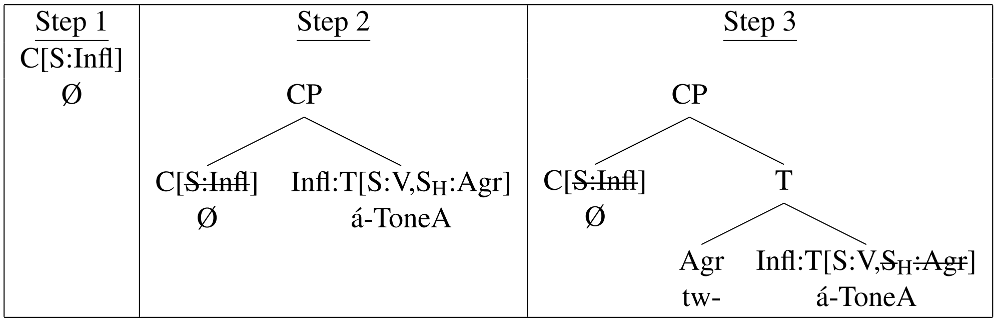
Note that this puts an extrinsic order on T and Asp: T is higher than Asp. We know that this is the case cross-linguistically, and it is the case in Kinande. It is simply stipulated in the Hierarchy of Projections approach. Giving the AuxV the feature [S:Asp] rather than [S:T] also stipulates the hierarchy, but it is doing no worse than cartographic approaches with a Hierarchy of Projections. It may be that there is a semantic explanation for why T must be higher than Asp; but whatever the reason, it can be stated in the current approach as a reason for why the AuxV has the feature [S:Asp] rather than the feature [S:T].

Once we do have this selectional feature, the overflow pattern follows, as I now show.

3.5 Derivation of the Overflow Pattern

The syntax starts with C again, and once again chooses the null finite declarative C, which selects Infl (Step 1):

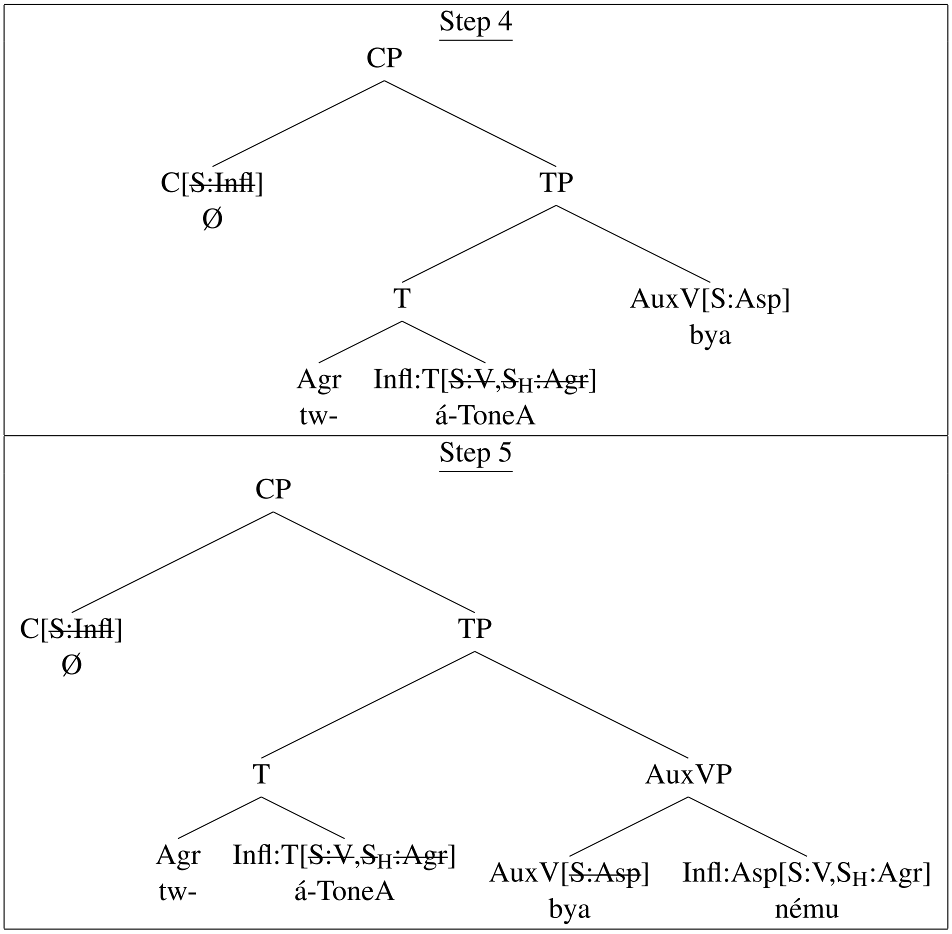
(29)



The syntax now has to merge something of category Infl. Both Infl:T and Infl:Asp are of category Infl, and it could in principle choose either one. If it were to choose Infl:Asp, however, there would be no way to get Infl:T into the clause. Infl:Asp could select the auxiliary verb, but the AuxV in Kinande has the selectional feature [S:Asp], and so only another Infl:Asp could be merged (which I assume is ruled out, either for semantic or syntactic reasons; cf. footnote 3). Infl:T could never be merged. Therefore, if Infl:T is desired in the clause, the syntax has to choose it at this stage. So, in Step 2, the syntax chooses Infl:T and merges it. Infl:T head-selects an Agr, so this is merged next (Step 3).

In the next step, Infl:T selects something of category V. If the syntax were to choose a main verb, this clause could not include Asp, since most main verbs do not select Asp. The syntax therefore has to choose an auxiliary verb (AuxV) instead. There is apparently only one in Kinande. This AuxV has the feature [S:Asp], which will enable the merger of Asp. So the AuxV *bya* is chosen and merged (Step 4):

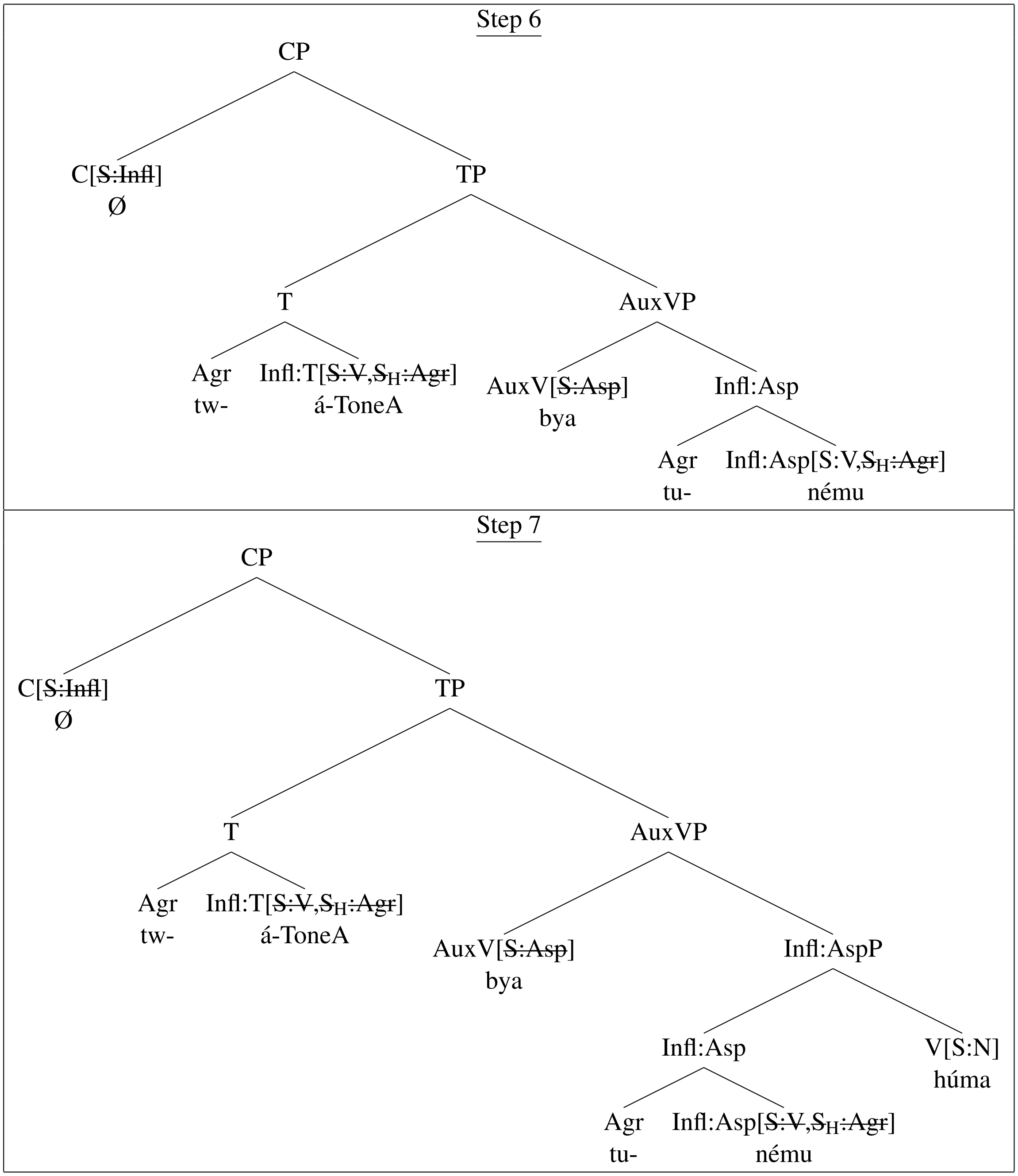
(30)



The AuxV selects Asp, so Infl:Asp is merged next (Step 5).

Infl:Asp head-selects an Agr, so an Agr is merged with it in the next step (Step 6). Infl:Asp also selects a V, so a V has to be merged next. The syntax chooses a main verb (Step 7):

(31)



Finally, pro will be merged with the main verb, checking off its [S:N] feature (Step 8, not shown). This checks off all of the [S] features in the structure, and the syntax terminates. The correct syntax has been built, using nothing more than the selectional features that were needed for tense by itself and aspect by itself, where no auxiliary verb was present.

One thing to note is that it is important for the analysis of Kinande that, when an Infl category does not appear to be present, it really is not. Consider the progressive aspect by itself again:

(32) 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. I assume that, cross-linguistically, Infl categories are typically absent when they are not visibly present. In English, for instance, a simple past tense like *I arrived* will not have null heads corresponding to AuxV$_{\text{Mod}}$, AuxV$_{\text{Perf}}$, AuxV$_{\text{Prog}}$, and AuxV$_{\text{Pass}}$.

The proposed analysis of the overflow pattern in Kinande is maximally simple. It uses only syntactic selection to drive insertion of lexical items into the syntax, which we need anyway. The overflow pattern falls out.

3.6 Overflow in Ndebele, Arabic, and Latin

The analysis will also apply straightforwardly in Arabic and Latin, the other two languages that Bjorkman (2011) claims show an overflow pattern. It will also work for Ndebele, a language that Pietraszko (2017) characterizes as also possessing an overflow pattern. I will start with Ndebele, since it is very similar to Kinande and the analysis of Kinande can apply to it without change. Ndebele is just like Kinande in that the verb can have Asp (progressive in 33a), and the verb can have T (future in 33b), but it cannot bear both at once:

(33) Ndebele (Pietraszko 2017: 88, (6a–c))

a. U-bala ibhuku.
2Sg.S-read 5book
'You are reading a book.'

b. U-za-bala.
2Sg.S-Fut-read
'You will read.'

c. U-za-be u-bala.
2Sg.S-Fut-Aux 2Sg.S-read
'You will be reading.'

Instead an auxiliary verb is used in (33c).

The only difference from Kinande is that the progressive Asp seems to be null, but it must be present because it gives rise to the overflow pattern. Allowing for this, we can give the same analysis as for Kinande. There are two Infl heads, Infl:T and Infl:Asp. C has the feature [S:Infl]. Both Infl:T and Infl:Asp have the feature [S:V]. The auxiliary verb has the feature [S:Asp]. As in Kinande, the only way to get both T and Asp in the same clause is for C to select Infl:T and Infl:T to select the AuxV. See the Kinande derivation above for details.

Standard Arabic is also almost identical to Kinande, assuming that Bjorkman's (2011) characterization is correct. There is a form of the main verb that is variously characterized as past tense or perfective aspect; we can follow Bjorkman and take it to be past tense:

(34) darasa
study.Past.Pfv.3SgM
'He studied.' (Bjorkman 2011: 27, (13a))

There is also a form of the main verb that is imperfective aspect (default present tense):

(35) ya-drusu
3M-Impf.study
'He studies.' (Bjorkman 2011: 28, (13b))

But to combine past tense and imperfective aspect, an auxiliary must be used:[5]

(36) kaana ya-drusu
be.Past.3SgM 3M-Impf.study
'He was studying. / He used to study.' (Bjorkman 2011: 28, (13c))

The analysis proposed for Kinande and Ndebele can be adapted straightforwardly to Standard Arabic. Once again there are two Infl heads, Infl:T and Infl:Asp. C has the feature [S:Infl]. Both Infl:T and Infl:Asp have the feature [S:V]. The auxiliary verb 'be' has the feature [S:Asp]. As in Kinande and Ndebele, the only way to get both T and Asp in the same clause is for C to select Infl:T and Infl:T to select the AuxV 'be'.

Latin is slightly different. Bjorkman (2011) claims that Latin verbs can express up to two inflectional categories at once, and the problem in the overflow pattern is that there are three. The verb can express a present perfect in (37a), and it can express a present passive in (37b), but it cannot express a present perfect passive in (37c). The auxiliary 'be' must be used (see also Embick 2000):

(37) Latin (Bjorkman 2011: 27, (12a–c))

a. Puellae crustulum consumpserunt.
girl-Pl.Nom small.pastry-Acc eat-Pl.Pfv
'The girls ate the little pastry.'

b. Crustulum consumitur.
small.pastry-Nom eat-Pres.Pass
'The little pastry is (being) eaten.'

c. Crustulum consumptum est.
small.pastry-Nom eat-Pass.Ptcp be.3Sg.Pres
'The little pastry {was / has been} eaten.'

There is a very simple selectional analysis that does not rely on there being any morphological limit on how many Infl categories the verb can be marked for in Latin (note that Embick 2000: (11) presents an example that seems to have imperfective, subjunctive, and passive together on one V, along with tense, I assume). All we have to say is that Perf does not select Pass in Latin. It only selects V.[6] In (37a), T selects Perf and Perf selects V. In (37b), T selects Pass and Pass selects V. In (37c), T selects Perf. Perf cannot select Pass, it can only select V. So it selects an AuxV, 'be', which is capable of selecting Pass. Pass then selects V. Since Perf cannot select Pass, this is the only way of combining Perf and Pass. Perf has to select an AuxV.

As can be seen, the analysis extends straightforwardly to the other languages that have been claimed to show an overflow pattern. The overflow pattern only requires selection, which is what drives syntax generally. There is no need for a last resort insertion operation.

[5] Note that there is another possible analysis, where the main verb always inflects for aspect (perfective or imperfective) and the auxiliary 'be' inherently realizes past tense. The default interpretation of perfective aspect would be past tense in this analysis, while the default interpretation of imperfective aspect would be present tense. I consider this a much more likely analysis, but the point in the text is that, if Bjorkman's characterization of this pattern is correct, the analysis as proposed captures it.

[6] The "deponent" verbs in Latin discussed by Embick (2000) must have the Pass head in this analysis, but without the syntactic and semantic effects it usually has.

4 Do-Support in English and Other Languages

As mentioned in the introduction, English do-support is a canonical example of a synthetic-periphrastic alternation that is typically analyzed as involving a last resort insertion operation (following Chomsky 1957). If we wish to do away with such operations, then we have to be able to account for English do-support.

An analysis of English do-support without last resort has already been proposed, by Baker (1991) and Bruening (2010). The gist of this analysis is the claim that, in all of the contexts for do-support, T (or Infl) has an [SP] feature (SP for "special purpose"). T with an [SP] feature strictly selects an AuxV and cannot select a MainV (it has the feature [S:AuxV]). If no semantically contentful AuxV is desired, then the semantically contentless *do* must be merged to satisfy the [S:AuxV] feature on T. So, in a negative clause (38b), T has an [SP] feature and must select an AuxV. In an inversion context (38c), the T that moves to C has an [SP] feature and must select an AuxV as its complement. The AuxV moves to T and then to C. In a verum focus context (38d), T also has an [SP] feature and therefore also an [S:AuxV] feature.

(38) 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. (Verum focus)
e. They said she would study too much, and she does. (VP ellipsis)
f. They said she would study too much, and study too much she does. (VP displacement)

As for VP ellipsis (38e), 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, and this T then only selects an AuxV. Finally, VP displacement also triggers do-support (38f). Perhaps the simplest account of VP fronting is that VP fronting is triggered by a C with a feature that triggers fronting of a predicate. This C also has the feature [SP]. In Bruening (2010), there is feature matching between C and T. If C is [SP], then T must be too, and T then only selects AuxVs.

In contrast, in a simple declarative like (38a), T does not have an [SP] feature. Its selection is then unconstrained, and it can select a MainV as well as an AuxV (recall from section 2.2 that T has the feature [S:V]; this is the T without the [SP] feature).

This analysis does not have a last resort insertion operation. It uses only the mechanism that we need to build structure anyway, namely, selection-driven merge. The one additional component is the [SP] feature on T. However, Bruening (2010) motivates this feature independently, and argues against the typical last resort analysis. The motivation comes from locative inversion, which is ungrammatical in all the environments where do-support applies (negation, inversion, verum focus, VP ellipsis, VP displacement), regardless of whether do-support actually does apply (for instance, in non-finite clauses, where do-support never applies). Bruening (2010) argues that this indicates that all of these contexts have something in common, which he takes to be the [SP] feature (which locative inversion is incompatible with; see Bruening 2010 for details).

Note that the [SP] 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 after merger, but by selectional features in the process of selecting items to merge into the derivation. The mechanism that chooses elements for the derivation 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 T that undergoes T-to-C movement has 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 not straightforward and will require much more research, but a possible approach is to adopt feature sharing between C and T, as Bruening (2010) proposed for English. Then the V2 C head that triggers T-to-C movement but does *not* have any topic or focus features has the [SP] feature and requires 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 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 in these languages that is described in Houser et al. (2011) does not have the same 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).

5 Conclusion and Discussion

This paper has argued that there is no need for a last-resort insertion operation in the analysis of synthetic-periphrastic alternations like do-support and the overflow pattern of auxiliaries. All that is needed is selection-driven merge. The operation that inserts an auxiliary is the same operation that inserts everything else, operating on the basis of the same information. The analyses of the patterns in particular languages are maximally simple, and require nothing more than specifying selectional features on particular items. This is something that is required anyway.

Future work will have to extend the analysis to other examples of synthetic-periphrastic alternations. If this extension is successful, then we can do without last resort insertion mechanisms entirely.

References

Adger, David. 2003. *Core syntax: A minimalist approach*. Oxford: Oxford University Press.

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.

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. 2021. Implicit arguments in English double object constructions. *Natural Language and Linguistic Theory* 39. 1023–1085. doi:10.1007/s11049-020-09498-4.

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. 1993. A minimalist program for linguistic theory. In Kenneth Hale & Samuel Jay Keyser (eds.), *The view from building 20: Essays in linguistics in honor of Sylvain Bromberger*, 1–52. Cambridge, MA: MIT Press.

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.

Collins, Chris. 2002. Eliminating labels. In Samuel David Epstein & T. Daniel Seely (eds.), *Derivation and explanation in the minimalist program*, 42–64. Oxford: Blackwell.

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. 2000. Features, syntax, and categories in the Latin perfect. *Linguistic inquiry* 31(2). 185–230.

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.

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

Müller, Gereon. 2017. Structure removal: An argument for feature-driven Merge. *Glossa: A Journal of General Linguistics* 2(1):28. 1–35. doi:10.5334/gjgl.193.

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.

Svenonius, Peter. 1994. C-selection as feature checking. *Studia Linguistica* 48. 133–155.

Department of Linguistics and Cognitive Science
University of Delaware
Newark, DE 19716
bruening@udel.edu