Spelling out the Numeration, Part 2: A Left-to-Right Derivation Accounts for Elements that Attach to the First Item of a Certain Type

Benjamin Bruening (University of Delaware)

rough draft, March 13, 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 examines how items are taken out of the numeration and merged together in the syntax. Spelling out how this works, I will show, can explain the problematic placement of definite markers in Bulgarian (e.g., Franks 2001, Embick & Noyer 2001) and Amharic (Kramer 2010). These definite markers appear to be placed at least partially on the basis of linear order. They have previously been analyzed as being placed by post-syntactic operations (Embick & Noyer 2001, Kramer 2010). I show here that we can do without post-syntactic processes; all we need is a numeration and a syntax. I argue that the syntax must build structure in a left-to-right rather than bottom-up fashion. Once it does, then the placement of items after the first element of the appropriate type falls out as an automatic consequence.

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.

In this paper and companion work I suggest that 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, by looking at the process that takes items from the numeration and merges them in the syntax.

The phenomenon I investigate is a particular type of word order effect, namely, the positioning of the definite suffix in Bulgarian (Franks 2001, Embick & Noyer 2001, among many others) and the definite suffix in Amharic (Kramer 2010). In Bulgarian, the position of this affix appears to be both hierarchical (because it ignores elements of the wrong syntactic type) and linear (because it attaches to the *first* element of the right type). Previous accounts have analyzed its placement as resulting from a post-syntactic lowering process (Embick & Noyer 2001). I show here that this type of placement (after the first element of the appropriate type) falls out as an automatic consequence of the way elements must be taken out of the numeration and merged in a derivation, if this process builds structure in a left-to-right fashion. I argue further that the syntax must work left-to-right, and so this type of placement is nicely accounted for. The proposal also accounts for the placement of the definite marker in Amharic, another case that has been argued to require post-syntactic mechanisms (Kramer 2010). It also accounts for the placement of English affixal negation (*n't*; Zwicky & Pullum 1983); in fact English affixal negation and the definite suffixes in Bulgarian and Amharic are all instances of the same phenomenon in the current account.

The main advantage of the proposed approach is that it enables us to simplify the model of grammar. We can get rid of post-syntactic reordering operations; in fact we can get rid of post-syntactic levels altogether. The only mechanism that locates elements in a phrase is the syntax itself (that is its job, after all). In the proposed approach, there is only a numeration and the syntax. These put together all complex forms, both morphological and syntactic. The account does without all the devices that other syntactic approaches to morphology have, like Distributed Morphology (Halle & Marantz 1993, Embick & Noyer 2001). The mechanisms proposed here are independently necessary to build all clauses and nominals, as I show, which means we can achieve a paired-down, minimal model of morphosyntax.

I start with a brief discussion of the conception of the numeration that I assume (section 2), followed by a more detailed look at the process the selects items from the numeration and merges them in the syntax (section 3). Sections 4 and 5 examine Bulgarian and Amharic, respectively. Along the way, section 4.5 includes an argument that the syntax must work left-to-right and not bottom-up as in most approaches.

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; I will mostly eschew null elements in this paper.) 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 then 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 these two operations, L2N and N2W, work will be the focus of this and companion papers. There have to be principles that determine how items are selected from the lexicon and put into the numeration. There have to be principles that determine how and in what order items are selected from the numeration and moved to the workspace. Companion papers will investigate how L2N works. This paper explores how N2W works.

3 From the Numeration to the Workspace

In this section I explore the process that selects items from the numeration and moves them to the workspace, N2W. This is a very important process, as it is behind much of syntax. Whatever determines the order of transfer from the numeration to the workspace also determines the order of merge and therefore the form of the syntactic structure. Unfortunately, this process has never received much attention in the literature; researchers have just assumed that there is such a process and it knows what order to operate in. I attempt to spell this process out. By spelling it out, I propose that we can gain a better account of certain phenomena that have been problematic for syntactic analyses, like the case of the Bulgarian definite marker. Of course, the analysis also has to account for more typical syntactic structure building, so I start with attempting to spell out how the procedure works in more typical cases. As we will see, once we spell this out, we do not need anything else in order to account for the apparently problematic cases, they simply fall out from the procedure.

3.1 From the Numeration to the Workspace: A CP Numeration

Companion papers investigate how the the input mechanism, L2N, selects items from the lexicon and puts them in the numeration. In this paper, I assume that it has done this job and that the numeration for any given phase is complete and ready for use in the workspace. It is the output mechanism, N2W, that is the focus of investigation here.

First, the number of unconnected items in the workspace has to be constrained. Imagine that all the items from the numeration could be moved into the workspace at once. Then the syntax would have to have a procedure for figuring out what order to merge them in. That is exactly what I am proposing N2W is: a procedure for determining in what order elements are selected and merged. We do not want two such procedures, so what happens in the workspace must be severely constrained, otherwise N2W would be doing no work. I therefore propose that the workspace can contain *only one item.* This will have the effect that, whenever an item is selected from the numeration and moved to the workspace, it has to be merged with the structure that is already there and cannot be left unconnected to it. We can state this as Maximality:

(2) Maximality: The workspace contains maximally one item.

The result of this principle is what was just stated: If there is already something in the workspace, then whenever an item is selected from the numeration and moved to the workspace, it has to be merged with that item.

As I will show in section 4.5, if the derivation works bottom-up, as most researchers assume, then there is no way to account for the facts discussed here. A bottom-up derivation will require that multiple items remain unconnected in the workspace, but then there is no way to constrain how they ultimately connect. In contrast, a left to right derivation permits Maximality to be observed and fully constrains the order of merge. I therefore adopt a left-to right derivation, rather than a bottom-up one. Note that this is also in keeping with what companion papers propose for L2N: selection of elements from the lexicon begins with functional items at the top of the phase and continues downward, according to selection. I assume the same is true of the procedure that selects elements from the numeration and merges them in the workspace, N2W. (Note that left-to-right and top-down are not always equivalent; left-to-right is what we need here.)

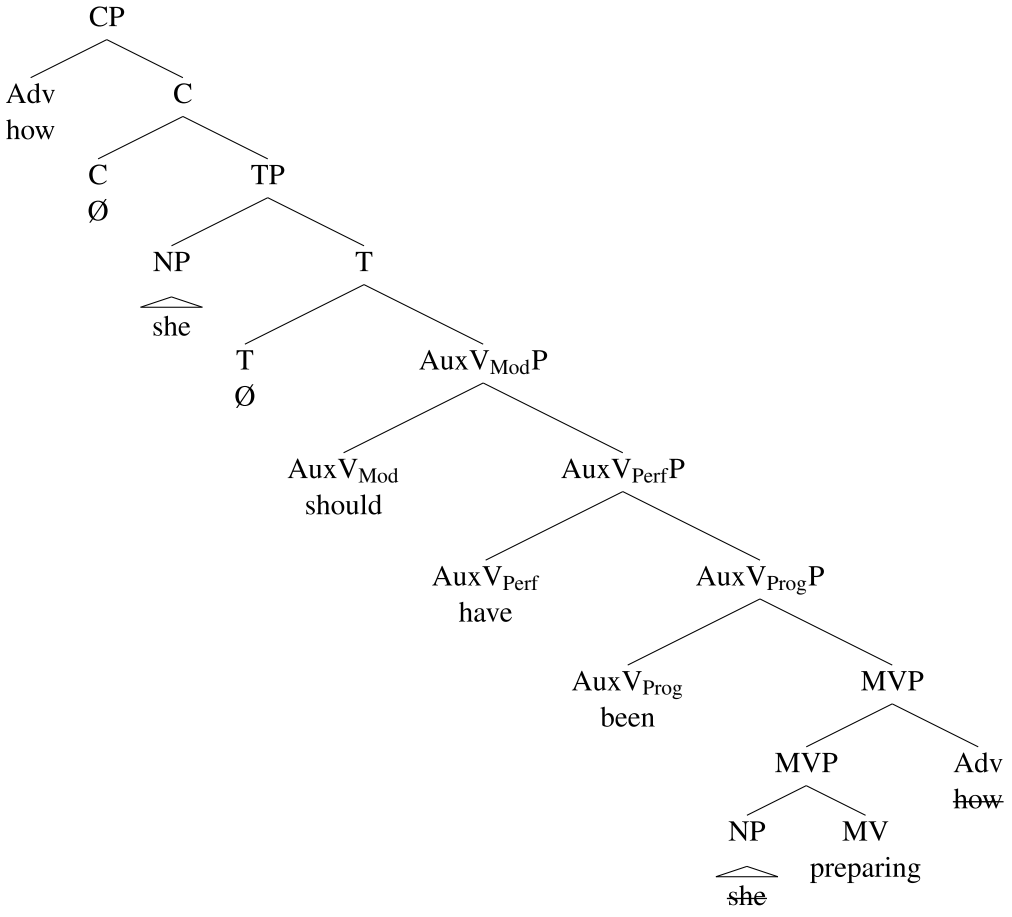
It is important to spell out how the derivation works in a left-to-right (mostly top-down) derivation. In order to spell out the mechanisms in detail, it is necessary to make certain assumptions and choices. Many of these choices are not necessary, and different choices could be made without changing the essentials of the system. The important point here is that spelling out the mechanisms in detail leads to a natural account of previously problematic syntactic placement data.

I will illustrate the system with two examples, one a CP phase and one a nominal phase. I start with a CP phase. Consider the following example:

(3) She is unsure [$_{\text{CP}}$ how she should have been preparing].

I will go through the derivation of the embedded CP, which I assume has the following structure:

(4)



For purposes here, I will assume that there is only one phase in the CP, and the maximal VP is not a phase. I assume that the adverb *how* moves to Spec-CP from a position adjoined to the VP of the main verb ("MVP"), as shown. I assume that subject NPs raise from Spec-VP to Spec-TP in English. I also assume that all auxiliary verbs are category AuxV, which is a subcategory of V (the other subcategory is M(ain)V(erb)). They are divided into further subcategories AuxV$_{\text{Prog}}$, AuxV$_{\text{Perf}}$, AuxV$_{\text{Mod}}$, and AuxV$_{\text{Pass}}$ (not present in this example). The highest AuxV may move to T in a positive declarative clause (and definitely does in a negative one). To keep things simple at the beginning, I also do not discuss the inflectional morphology at this point; inflectional morphology does become important in sections 3.3 and 4.

As mentioned, NPs constitute their own phases and have their own numerations. I propose that a numeration that includes another phase will have a placeholder item that will then be filled in with its own numeration. These are specified only for category. So the NP in the tree above is a placeholder item which I will notate Ph:N, for a phase of category N. The numeration for the above CP will then include the following items:

(5)

CP Numeration: Initial State				
Adv	C	Ph:N	T	AuxV$_{\text{Mod}}$
how	Ø		Ø	should
[WH,S$_{\text{AR}}$:MV]	[S$_{\text{S}}$:WH,S$_{\text{C}}$:T]		[S$_{\text{S}}$:N,S$_{\text{C}}$:V]	[S$_{\text{C}}$:{MV,Pass,Prog,Perf}]

AuxV$_{\text{Perf}}$	AuxV$_{\text{Prog}}$	MV
have	been	preparing
[S$_{\text{C}}$:{MV,Pass,Prog}]	[S$_{\text{C}}$:{MV,Pass}]	[S$_{\text{S}}$:N]

Movement in the syntax is copying, so the numeration includes only one token of each item.

I will assume that much of syntax is feature-driven, as is standard in many approaches to syntax following Chomsky (1993). In particular, merge will be driven by features, for instance selectional features. I assume that all items in the numeration have selectional features, notated "[S:X]" (following Bruening 2013). Selectional features are distinguished into complement selection (S$_{\text{C}}$), specifier selection (S$_{\text{S}}$), and adjunct selection (S$_{\text{A}}$); adjuncts are further distinguished into adjuncts that adjoin on the left (S$_{\text{AL}}$) and adjuncts that adjoin on the right (S$_{\text{AR}}$). I assume that wh-Cs have [WH] specifier selectional features, as shown, and T has an [N] specifier selection feature. Note that labeling can be determined purely on the basis of these selectional features: the item with an S$_{\text{S}}$ or S$_{\text{C}}$ feature projects, whereas an adjunct (S$_{\text{AL/AR}}$) does *not* project, instead what it merges with projects.

I also assume that items can select a disjunctive list of possible categories, notated [S:{X,Y,...}]. This means that the item can select any *one* item of category X, Y, ... For instance, in English, the modal auxiliaries are capable of combining with the perfect auxiliary, the progressive auxiliary, the passive auxiliary, or just a main verb, but only with one of these at a time. See the companion paper on disjunctive selection.

Now that we have this numeration, the question is how the derivation proceeds. Two possibilities are (1) that it proceeds on the basis of selection, or (2) on the basis of a hierarchy of projections (either universal, or language-particular). In the companion paper investigating L2N, it is concluded that selection is necessary while a hierarchy of projections is unnecessary and insufficient and should be done away with. I will therefore pursue the hypothesis that N2W operates on the basis of selectional features.

Since this is a left-to-right derivation, N2W will proceed by identifying an item like C, then checking for that item whether there are any left adjuncts or specifiers, since those have to be merged first. After merging C, it will check whether C selects a complement, then it will repeat the procedure for the complement. Since left adjuncts can themselves have left adjuncts, specifiers, and complements, the procedure needs to iterate for every item it is considering. I therefore propose that the procedure operates on a *stack*: it selects an item for the top of the stack, and considers it. At some point it may need to put another item on top of the stack and consider that one; once that item is removed, it will go back to the item that was below it on the stack.

I formalize the procedure that N2W follows as the following (to be revised in section 3.3):

(6) Procedure for N2W, where α is the top of the stack:

1. Locate the phase head and make it the top of the stack α; go to step 2.
2. Is there an element in the numeration Y with a feature [S$_{\text{AL}}$:α]?
    (a) If no, go to step 3.

    (b) If yes, make Y the top of the stack α and return to step 2.

3. Does α have an unchecked [$\mathrm{S_S}$:X] feature?
    - (a) If no, move α to the workspace; go to step 4.
    - (b) If yes, locate an X in the numeration or the workspace; move X to the workspace or copy X from the workspace to the current location in the tree; check off the [$\mathrm{S_S}$:X] feature on α and return to step 3.
4. Does α have an [$\mathrm{S_C}$:β] feature?
    - (a) If no, go to step 5.
    - (b) If yes, is there a β in the numeration?
        - i. If no, copy a β from elsewhere in the workspace to the current location in the workspace; go to step 5.
        - ii. If yes, remove α from the stack, make β the top of the stack and return to step 2.
5. Is there an element γ with an [$\mathrm{S_{AR}}$:α] feature in the numeration?
    - (a) If no, remove α from the stack and go to the next item on the top of the stack, returning to step 2; if the stack is empty, terminate.
    - (b) If yes, remove α from the stack and make γ the top of the stack; return to step 2.

We also need the following principle:

(7) If at any step more than one item meets the description, choose according to the following preferences:

a. Choose a functional element over a lexical element;

b. Choose according to the case hierarchy Nom > Dat > Acc > Obl;

c. Choose the element with the largest disjunctive set of selectional features.

I now go through the derivation of the CP in (4). The derivation starts with the numeration in (5), which I repeat in (8). At this point the workspace is empty.

(8)

CP Numeration: Initial State				
Adv how [WH,$\mathrm{S_{AR}}$:MV]	C Ø [$\mathrm{S_S}$:WH,$\mathrm{S_C}$:T]	Ph:N	T Ø [$\mathrm{S_S}$:N,$\mathrm{S_C}$:V]	$\mathrm{AuxV_{Mod}}$ should [$\mathrm{S_C}$:{MV,Pass,Prog,Perf}]

$\mathrm{AuxV_{Perf}}$ have [$\mathrm{S_C}$:{MV,Pass,Prog}]	$\mathrm{AuxV_{Prog}}$ been [$\mathrm{S_C}$:{MV,Pass}]	MV preparing [$\mathrm{S_S}$:N]

Workspace:

—*empty*—

Following the procedure in (6), N2W will first locate C in step 1, since it is the phase head. C will be made the top of the stack. In step 2, it scans for an element with an [$\mathrm{S_{AL}}$:C] feature, and does not find one, so it goes to step 3. In step 3, C does have an unchecked [$\mathrm{S_S}$:WH] feature, so N2W looks for something that is WH. It finds the Adv *how*, because it is a wh-phrase with the feature [WH]. N2W will therefore move the Adv to the workspace:[1]

[1] I assume for purposes here that the Adv is a single head. However, wh-phrases can be phrasal; if they are, they are their own phases. The numeration for a CP phase that includes a wh-phrase will then include a placeholder like Ph:N[WH], for instance.

(9)

CP Numeration After First Selection				
	C Ø [~~S_S:WH~~,S_C:T]	Ph:N	T Ø [S_S:N,S_C:V]	AuxV$_{Mod}$ should [S_C:{MV,Pass,Prog,Perf}]

AuxV$_{Perf}$ have [S_C:{MV,Pass,Prog}]	AuxV$_{Prog}$ been [S_C:{MV,Pass}]	MV preparing [S_S:N]

Workspace:
Adv
how
[WH,S_{AR}:MV]

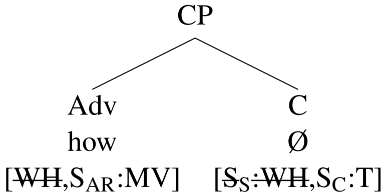
N2W checks off the [S_S:WH] on C and returns to step 3. Since the sole [S_S] feature on C has been checked off, C is then moved to the workspace, where it is merged with the Adv to create a CP (because C selects its specifier):

(10)

CP Numeration After Second Selection				
		Ph:N	T Ø [S_S:N,S_C:V]	AuxV$_{Mod}$ should [S_C:{MV,Pass,Prog,Perf}]

AuxV$_{Perf}$ have [S_C:{MV,Pass,Prog}]	AuxV$_{Prog}$ been [S_C:{MV,Pass}]	MV preparing [S_S:N]

Workspace:



This checks off the [WH] feature of the Adv; the [S_S:WH] feature of C has already been checked off.

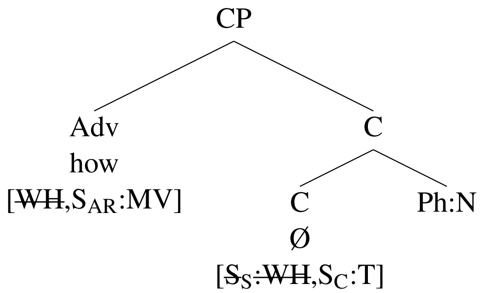
N2W now goes to step 4. C does have an [S_C:T] feature, so N2W scans the numeration for a T. It finds one, so it puts T on the top of the stack as α and returns to step 2. In Step 2, there is no element in the numeration with an [S:$_{AL}$:T] feature, so N2W moves to step 3. In step 3, T has an unchecked [S_S:N] feature, so N2W looks for something of category N in either the numeration or the workspace, and finds one, namely, Ph:N. So Ph:N is moved to the workspace. It merges wth the lowest node in the tree, C:

(11)

CP Numeration After Third Selection				
			T Ø [~~S_S:N~~,S_C:V]	AuxV$_{Mod}$ should [S_C:{MV,Pass,Prog,Perf}]

AuxV$_{Perf}$ have [S_C:{MV,Pass,Prog}]	AuxV$_{Prog}$ been [S_C:{MV,Pass}]	MV preparing [S_S:N]

Workspace:



C projects, since it has an [S_C] feature. N2W checks off the [S_S:N] feature of T and returns to step 3.

First, though, the syntax has to pause the current numeration and select a numeration for the NP phase:

(12) A placeholder Ph:X is expanded as soon as Ph:X is moved to the workspace.

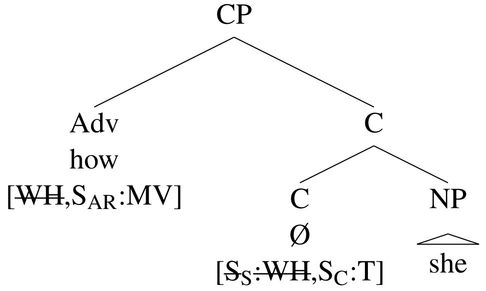
I will spell out an example of this in more detail in section 4.5. For now I will just assume that this takes place, and Ph:N is replaced with a full NP, which I do not spell out here:

(13)

CP Numeration After Completing NP Phase				
			T Ø [~~S_S:N~~,S_C:V]	AuxV$_{Mod}$ should [S_C:{MV,Pass,Prog,Perf}]

AuxV$_{Perf}$ have [S_C:{MV,Pass,Prog}]	AuxV$_{Prog}$ been [S_C:{MV,Pass}]	MV preparing [S_S:N]

Workspace:



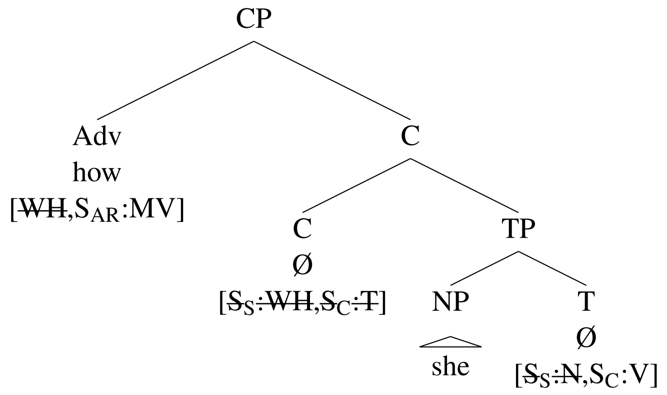
The syntax then returns to the CP numeration. N2W returns to step 3. T no longer has an unchecked [S_S] feature, so T is moved to the numeration. It merges with the NP. Since it selects a category N as specifier, the NP phase is pushed to a left branch of a TP node:

(14)

CP Numeration After Fourth Selection				
				AuxV$_{Mod}$ should [S_C:{MV,Pass,Prog,Perf}]

AuxV$_{Perf}$ have [S_C:{MV,Pass,Prog}]	AuxV$_{Prog}$ been [S_C:{MV,Pass}]	MV preparing [S_S:N]

Workspace:



This checks off the [S_C:T] feature of C. The [S_S:N] feature of T has already been checked off.

N2W now moves to step 4. T does have an [S_C] feature, namely, [S_C:V]. This means that T can select anything of category V, including any subcategory of AuxV or MV. N2W scans the numeration for an item of category V in the numeration. It finds four such elements: AuxV_{Mod}, $\text{AuxV}_{\text{Perf}}$, $\text{AuxV}_{\text{Prog}}$, and MV. This is where the principle in (7) comes into play. This principle tells N2W to choose an AuxV over an MV, but there are still three of those. It also says to choose the element that has the largest set of disjunctive features. In this case, that picks out AuxV_{Mod}. So N2W makes AuxV_{Mod} the top of the stack (α) and returns to step 2.

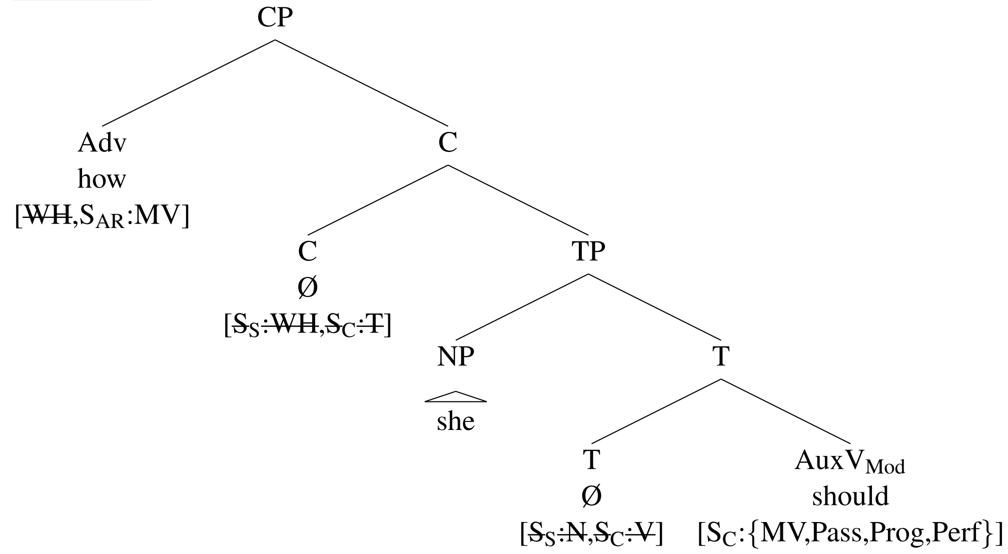
In step 2, there is nothing in the numeration that has an [S_{AL}:AuxV] feature. In step 3, AuxV_{Mod} does not have an unchecked [S_S] feature, so AuxV_{Mod} is moved to the workspace. It merges with T, with T projecting since it selects AuxV as complement:

(15)

CP Numeration After Fifth Selection				

$\text{AuxV}_{\text{Perf}}$ have [S_C:{MV,Pass,Prog}]	$\text{AuxV}_{\text{Prog}}$ been [S_C:{MV,Pass}]	MV preparing [S_S:N]

Workspace:


CP
Adv
how
[~~WH~~,S_{AR}:MV]
C
C
Ø
[~~S_S:WH~~,~~S_C:T~~]
TP
NP
she
T
T
Ø
[~~S_S:N~~,~~S_C:V~~]
AuxV_{Mod}
should
[S_C:{MV,Pass,Prog,Perf}]

This checks off the [S_C:V] feature of T.

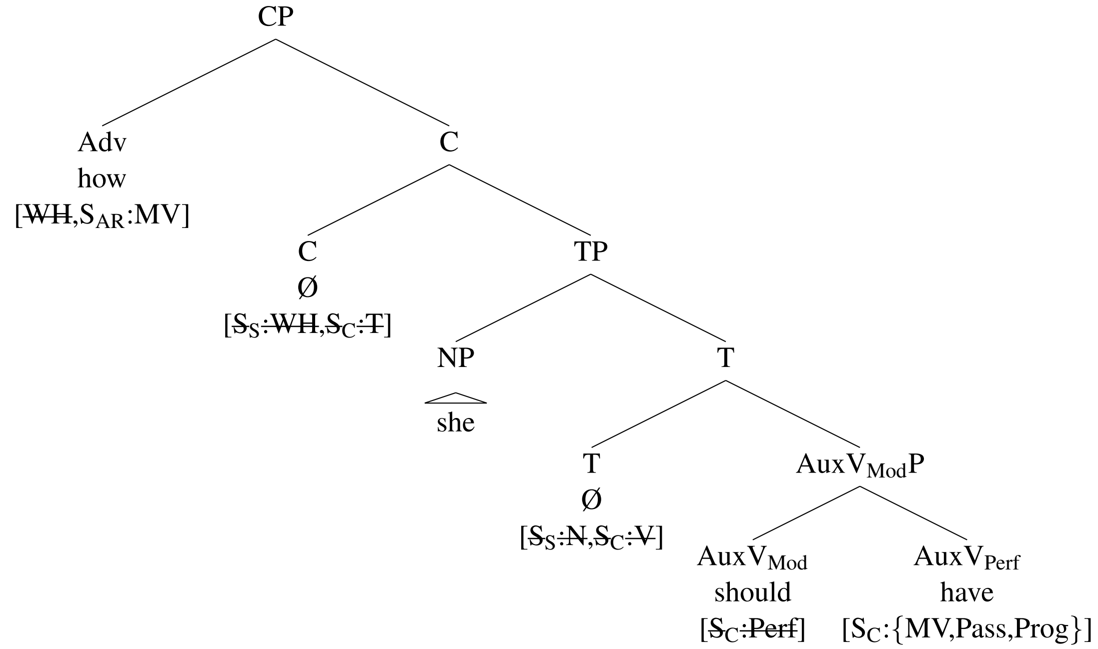
N2W now goes to step 4 in the procedure. AuxV$_{\text{Mod}}$ does have an [S$_{\text{C}}$] feature, so N2W looks for a MV, AuxV$_{\text{Pass}}$, AuxV$_{\text{Prog}}$, or AuxV$_{\text{Perf}}$. It finds three of them. Once again, the principle in (7) says to choose an AuxV over an MV, and it says to choose the AuxV with the longest disjunctive list of selected items. This is AuxV$_{\text{Perf}}$. So N2W makes AuxV$_{\text{Perf}}$ the top of the stack (α) and returns to step 2.

In step 2, there is nothing in the numeration with an [S$_{\text{AL}}$:AuxV] feature. In step 3, AuxV$_{\text{Perf}}$ does not have an unchecked [S$_{\text{S}}$] feature, so AuxV$_{\text{Perf}}$ is moved to the workspace. It merges with AuxV$_{\text{Mod}}$, which projects since it selects AuxV$_{\text{Perf}}$ as its complement (to save space, I reduce disjunctive lists once checked):

(16)

CP Numeration After Sixth Selection				
	AuxV$_{\text{Prog}}$ been [S$_{\text{C}}$:{MV,Pass}]	MV preparing [S$_{\text{S}}$:N]		

Workspace:



CP
Adv
how
[~~WH~~,S$_{\text{AR}}$:MV]
C
C
Ø
[~~S$_{\text{S}}$:WH~~,~~S$_{\text{C}}$:T~~]
TP
NP
she
T
T
Ø
[~~S$_{\text{S}}$:N~~,~~S$_{\text{C}}$:V~~]
AuxV$_{\text{Mod}}$P
AuxV$_{\text{Mod}}$
should
[~~S$_{\text{C}}$:Perf~~]
AuxV$_{\text{Perf}}$
have
[S$_{\text{C}}$:{MV,Pass,Prog}]

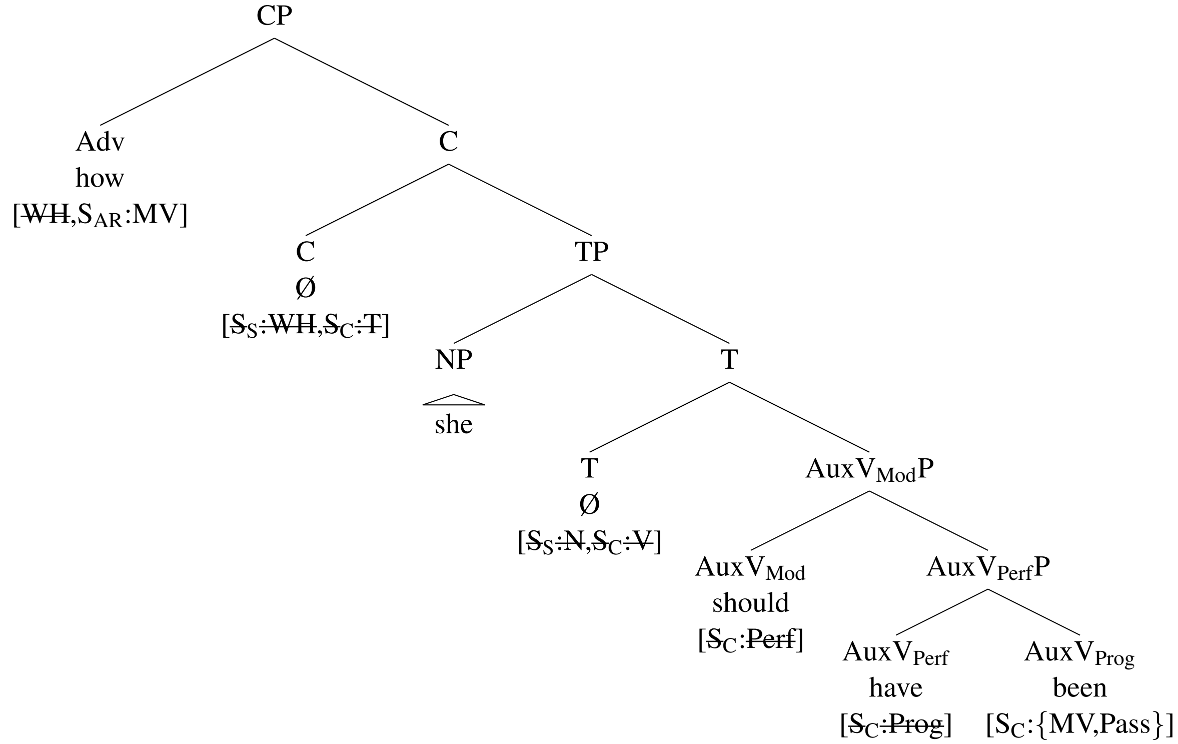
N2W now goes to step 4 in the procedure. AuxV$_{\text{Perf}}$ does have an [S$_{\text{C}}$] feature, so N2W looks for a MV, AuxV$_{\text{Pass}}$, or AuxV$_{\text{Prog}}$. It finds two of them. Once again, the principle in (7) says to choose an AuxV over an MV, and it says to choose the AuxV with the longest disjunctive list of selected items. So AuxV$_{\text{Prog}}$ is selected. N2W makes AuxV$_{\text{Prog}}$ the top of the stack (α) and returns to step 2.

In step 2, there is nothing in the numeration with an [S$_{\text{AL}}$:AuxV] feature. In step 3, AuxV$_{\text{Prog}}$ does not have an unchecked [S$_{\text{S}}$] feature, so AuxV$_{\text{Prog}}$ is moved to the workspace. It merges with AuxV$_{\text{Perf}}$, which projects since it selects AuxV$_{\text{Prog}}$ as its complement:

(17)

CP Numeration After Seventh Selection				
		MV preparing [S_S:N]		

Workspace:


CP
Adv
how
[~~WH~~,S_{AR}:MV]
C
C
Ø
[~~S_S:WH~~,~~S_C:T~~]
TP
NP
she
T
T
Ø
[~~S_S:N~~,~~S_C:V~~]
AuxV_{Mod}P
AuxV_{Mod}
should
[~~S_C~~:~~Perf~~]
$\text{AuxV}_{\text{Perf}}$P
$\text{AuxV}_{\text{Perf}}$
have
[~~S_C:Prog~~]
$\text{AuxV}_{\text{Prog}}$
been
[S_C:{MV,Pass}]

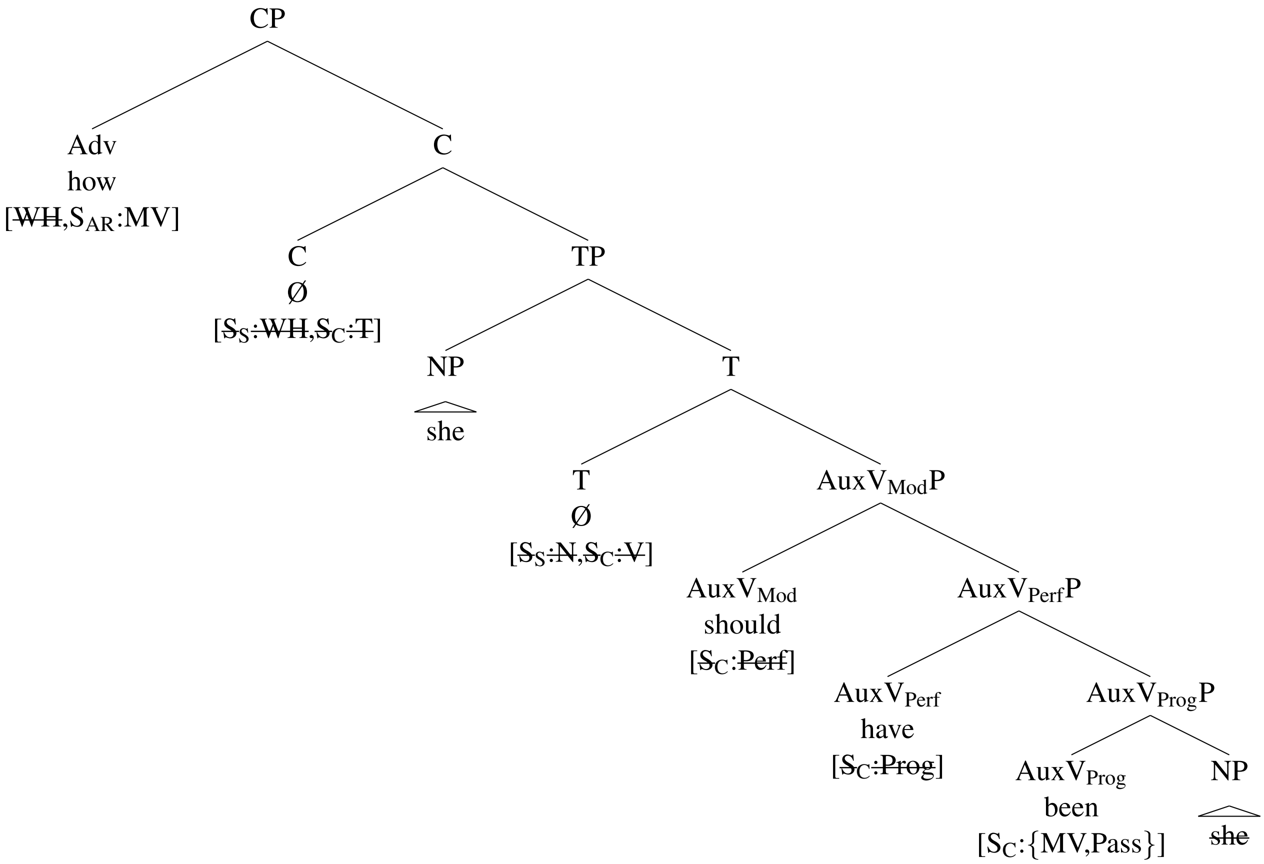
N2W now goes to step 4 in the procedure. $\text{AuxV}_{\text{Prog}}$ does have an [S_C] feature, so N2W looks for an MV or $\text{AuxV}_{\text{Pass}}$. It finds an MV, which is the last item in the numeration. N2W makes MV the top of the stack (α) and returns to step 2.

In step 2, there is nothing in the numeration with an [S_{AL}:MV] feature. In step 3, MV does have an unchecked [S_S] feature, so N2W scans the numeration and the workspace for an N. The only N is the NP *she*, already in the workspace. So this is copied and merged at the bottom of the tree:

(18)

CP Numeration After Copying an N				
		MV preparing [S_S:N]		

Workspace:



AuxV$_{\text{Prog}}$ projects, since it has an [S$_C$] feature.

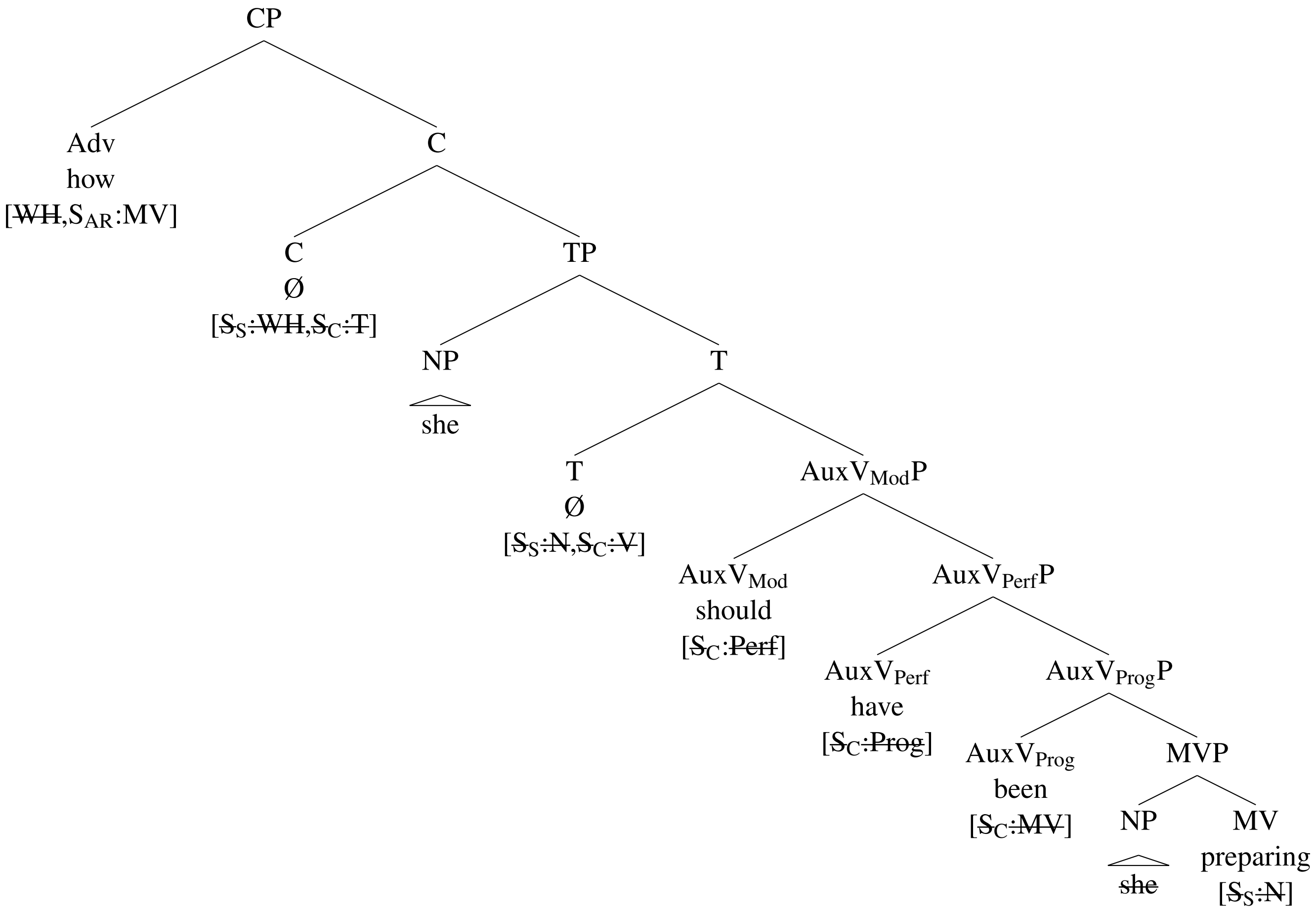
Principles of pronunciation, which are not the concern here, dictate that in general the leftmost copy of any element is the one pronounced, so the lower copy of *she* is not pronounced (represented with strikethrough).

N2W checks off the [S$_S$] feature on MV and returns to step 3. MV does not have any other [S$_S$] features, so MV is moved to the workspace. It merges with the NP which it selects as a specifier, so the NP is pushed to a left branch and MV projects to their mother node:

(19)

CP Numeration After Eighth Selection				

Workspace:



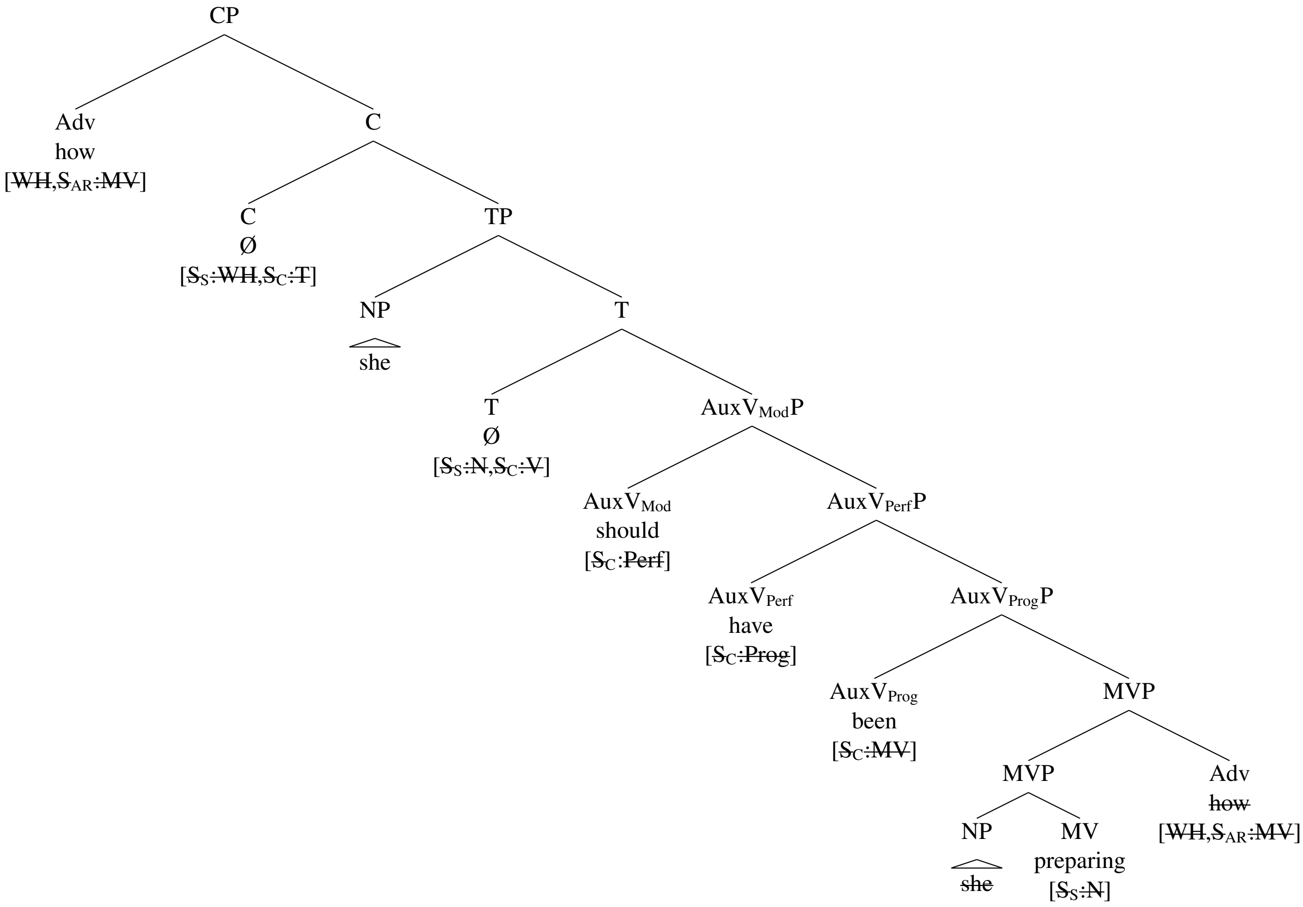
N2W now goes to step 4. MV does not have an [S_C] feature. In step 5, there is no element with an [S_{AR}] feature in the numeration (in fact there is nothing left in the numeration). So N2W terminates. All items have been taken out of the numeration and merged into the workspace.

Inspection of the tree in (43) will reveal that almost all features have been checked. There is exactly one feature that has not been checked, namely, the [S_{AR}:MV] feature of the adverb *how*. In order to check this feature, the syntax will have to follow the following procedure:

(20) After N2W terminates, if there is an X in the workspace with an unchecked [S_A:Y] feature, copy X and merge it with a projection of Y.

This has the effect of copying the Adv and adjoining it to MVP on the right (since Adv has an [S_{AR}] feature and not an [S_{AL}] feature):

(21) Workspace:



Copying Adv and merging it with MVP checks off the [S_{AR}:MV] feature on both copies of the Adv (because they are literally copies of the same item). Principles of pronunciation, which are not the concern here, dictate that the higher copies of Adv and NP are pronounced.

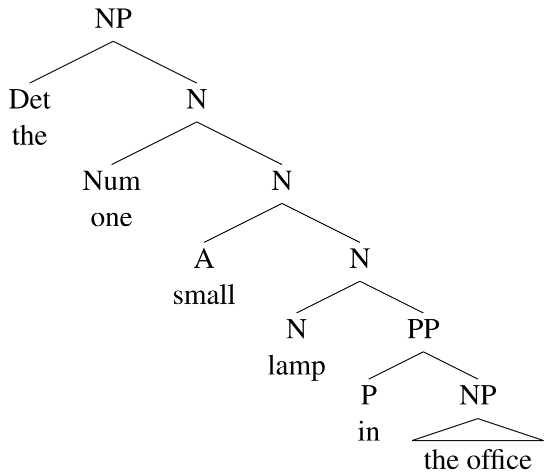
This completes the CP phase. If there are no other phases to build, the syntax terminates, and a well-formed sentence is the result (this particular clause was embedded in another, which in the left-to-right derivation assumed here would have been built first).

3.2 How the System Works: A Nominal Numeration

Now consider a nominal example, like the following:

(22) the one small lamp in the office

I assume that the structure that will be built is something like the following (I do not assume the DP Hypothesis, rather, the functional elements are daughters of N):

(23)


In English, items of category Det always precede items of category Num, which always precede items of category A. I will assume that these categories have [$\mathrm{S_{AL}}$:N] features, which puts them on the left of the head N. I will further assume that when one of these elements merges with a projection of N, their mother node gains a feature from the non-head daughter. So, when an A merges with an N, the result is an AN. When a Num merges with an N, the result is a NumN. And so on. Items of category Det then select for a disjunctive list {N,AN,NumN}; items of category Num select for a disjunctive list {N,AN}; and items of category A select for N. Adnominal PPs do not change the category of N at all, and I assume they can merge at any point in the nominal derivation. They have an [$\mathrm{S_{AR}}$:N] feature.

The numeration for this phase is the following, with the selectional features just discussed. Again there is a placeholder for the lower nominal phase:

(24)

NP Numeration: Initial State					
Det the [$\mathrm{S_{AL}}$:{N,AN,NumN}]	Num one [$\mathrm{S_{AL}}$:{N,AN}]	A small [$\mathrm{S_{AL}}$:N]	N lamp	P in [$\mathrm{S_{AR}}$:N,$\mathrm{S_C}$:N]	Ph:N

Since all of these items select N as adjuncts, the N will project in every instance of merge.

N2W starts with Step 1 and locates the phase head, N, which is made the top of the stack. N2W then goes to step 2. There are multiple elements in the numeration with the feature [$\mathrm{S_{AL}}$:N]: Det, Num, A. In order to decide which to select, it uses the principle in (7). Det and Num are functional elements while A is not. Of Det and Num, Det has the largest disjunctive set of selectional features, so Det is selected and made the top of the stack (α). N2W returns to step 2. There is no element in the numeration with the feature [$\mathrm{S_{AL}}$:Det], so N2W goes to step 3. Det does not have an unchecked [$\mathrm{S_S}$] feature, so Det is moved to the workspace:

(25)

NP Numeration After First Selection					
	Num one [$\mathrm{S_{AL}}$:{N,AN}]	A small [$\mathrm{S_{AL}}$:N]	N lamp	P in [$\mathrm{S_{AR}}$:N,$\mathrm{S_C}$:N]	Ph:N

Workspace:
Det
the
[$\mathrm{S_{AL}}$:{N,AN,NumN}]

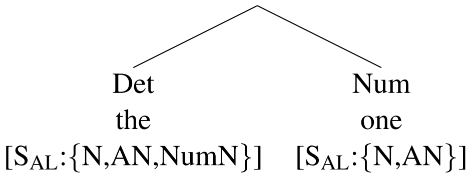
Det does not have an [$\mathrm{S_C}$] feature, so N2W goes to step 5. There is nothing in the numeration with an [$\mathrm{S_{AR}}$:Det] feature, so N2W goes to the next item in the stack, which is N, and returns to step 2.

There are now two elements in the numeration with the feature [S_{AL}:N], Num and A. The principle in (7) picks Num (because it is functional while A is lexical, and its disjunctive list of selectional features is larger). So Num is selected and made the top of the stack (α). N2W returns to step 2. There is no element in the numeration with the feature [S_{AL}:Num], so N2W goes to step 3. Num does not have an unchecked [S_S] feature, so Num is moved to the workspace and merged with Det:

(26)

NP Numeration After Second Selection					
		A small [S_{AL}:N]	N lamp	P in [S_{AR}:N,S_C:N]	Ph:N

Workspace:



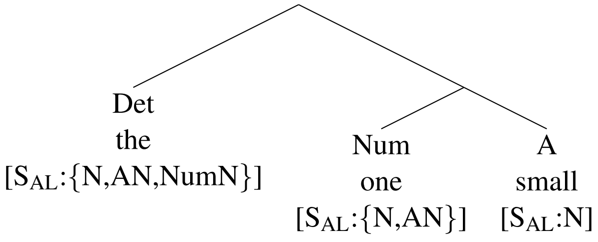
Num does not have an [S_C] feature, so N2W goes to step 5. There is nothing in the numeration with an [S_{AR}:Num] feature, so N2W goes to the next item in the stack, which is N, and returns to step 2.

There is now one element in the numeration with the feature [S_{AL}:N], namely, A. So A is selected and made the top of the stack (α). N2W returns to step 2. There is no element in the numeration with the feature [S_{AL}:A], so N2W goes to step 3. A does not have an unchecked [S_S] feature, so A is moved to the workspace and merged at the bottom of the existing tree:

(27)

NP Numeration After Third Selection					
			N lamp	P in [S_{AR}:N,S_C:N]	Ph:N

Workspace:



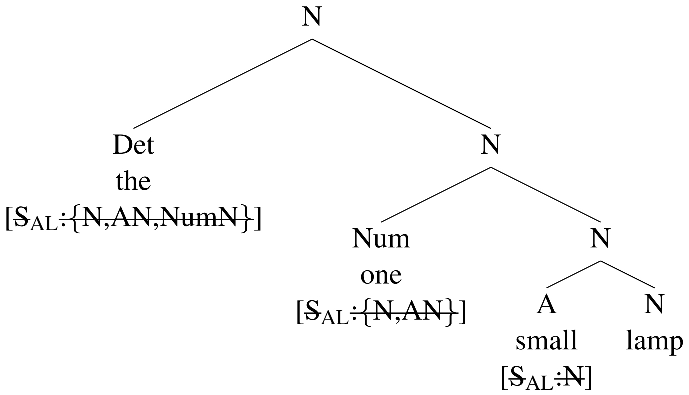
A does not have an [S_C] feature, so N2W goes to step 5. There is nothing in the numeration with an [S_{AR}:A] feature, so N2W goes to the next item in the stack, which is N, and returns to step 2.

There is now nothing in the numeration with the feature [S_{AL}:N], so N2W goes to step 3. N does not have an unchecked [S_S] feature, so N is moved to the workspace and merged at the bottom of the tree:

(28)

NP Numeration After Fourth Selection					
				P in [S_{AR}:N,S_C:N]	Ph:N

Workspace:



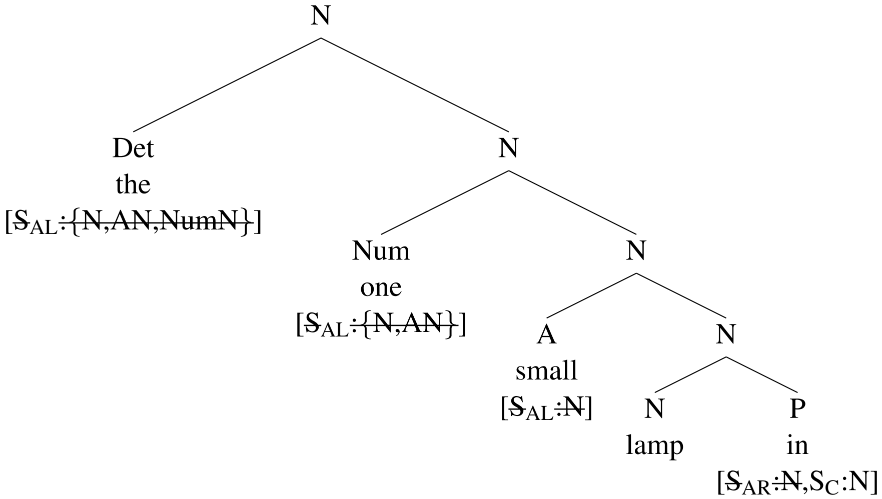
N can now project, as all of the other items in the tree have [S_{AL}] features. All of the selectional features of these previously merged items are also now checked off, since they have been merged with a projection of N.

N2W now goes to step 4. N does not have an [S_C] feature, so N2W goes to step 5. There is an element in the numeration with the feature [S_{AR}], namely, the P. N2W makes P the top of the stack (α) and returns to step 2. There is no element in the numeration with the feature [S_{AL}:P] in step 2. P does not have an unchecked [S_S] feature in step 3, so it is moved to the workspace and merged with a projection of N on the right:

(29)

NP Numeration After Fifth Selection					
					Ph:N

Workspace:



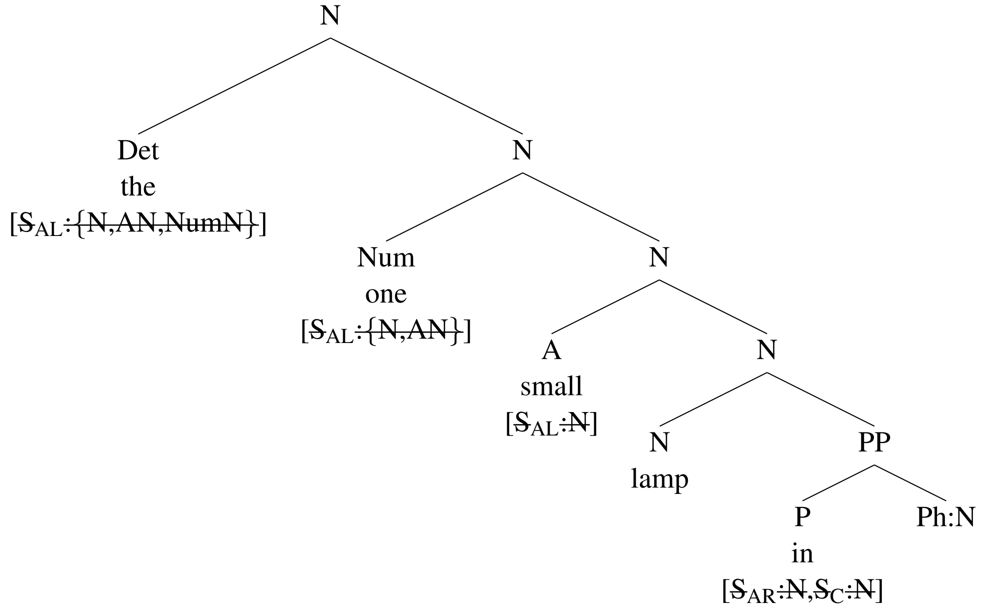
This checks off the [S_{AR}:N] feature of the P.

N2W then goes to step 4. P does have an [S_C:N] feature, and there is an N in the numeration (Ph:N). So N2W makes Ph:N the top of the stack (α) and returns to step 2. In step 2, there is nothing left in the numeration other than Ph:N. Ph:N has no [S_S] feature in step 3, so it is moved to the numeration and merged with the P:

(30)

NP Numeration After Sixth Selection					

Workspace:



This will check off the [S_C:N] feature of the P.

The Ph:N placeholder now needs to be expanded, using its own numeration. I do not go through this here. Once that is complete, N2W moves to step 4. The completed phase does not have an [S_C] feature, so N2W moves to step 5. There is now nothing left in the numeration, and the stack is empty, so N2W terminates. All features are now checked, all items have been moved from the numeration to the workspace, and a well-formed structure has been built.

3.3 Inflection: Agr Heads

Many languages, if not most, also have inflection, which I have not yet discussed. For instance, in English, every verb has an inflectional head (which may be null). In the sentence *He lies*, for instance, the verb has a morpheme *-s* which indicates tense and agreement: present tense, third person singular. To account for these inflectional heads, we need to add an additional type of selection, *head selection*, which I will notate “S_H.” I propose that every verb in English requires an inflectional head which I will notate Agr, for “agreement.” This means that every verb in English head-selects Agr. That is, every verb has the feature [S_H:Agr].

If there is more than one verb, each of them has this feature and requires an Agr morpheme. For instance, in *She has eaten*, the auxiliary verb *have* again has the present tense, third person singular morpheme *-s*, while the main verb has a morpheme *-en* which I assume agrees with *have* (since its form is determined by *have*). In English, the form of the Agr morpheme adjoined to each verb is determined by the functional head immediately above it (T in the case of the highest V, an AuxV in the case of lower Vs). T agrees with the subject, and thereby passes its own tense feature and the features of the subject to the V immediately below it (for purposes here, any theory of agreement will do, for instance the Agree of Chomsky 2000).

In addition, there are heads that head-select another head, but in the manner of an adjunct. For instance, affixal negation in English head-selects something of category AuxV. An example is *She hasn’t eaten*, where the highest AuxV has an *n’t* adjoined to it. This is not simple contraction, as the affixal negation moves along with the verb in subject-auxiliary inversion, which is not possible with contraction of auxiliaries (Zwicky & Pullum 1983):

(31) a. You shouldn't't've given me that.
b. Shouldn't you have given me that?
c. * Shouldn't't've you given me that?

I therefore assume that *n't* is a head that is merged with the AuxV and forms a complex head with it.

Note that it does not seem to be the case that the AuxV head-selects *n't*. Rather, *n't* head-selects an AuxV. It will attach to any AuxV, and whichever one is first. This behavior actually falls out from the current left-to-right system, if we simply give *n't* an [S_{HA}:AuxV] feature, where "HA" is "head adjunct."

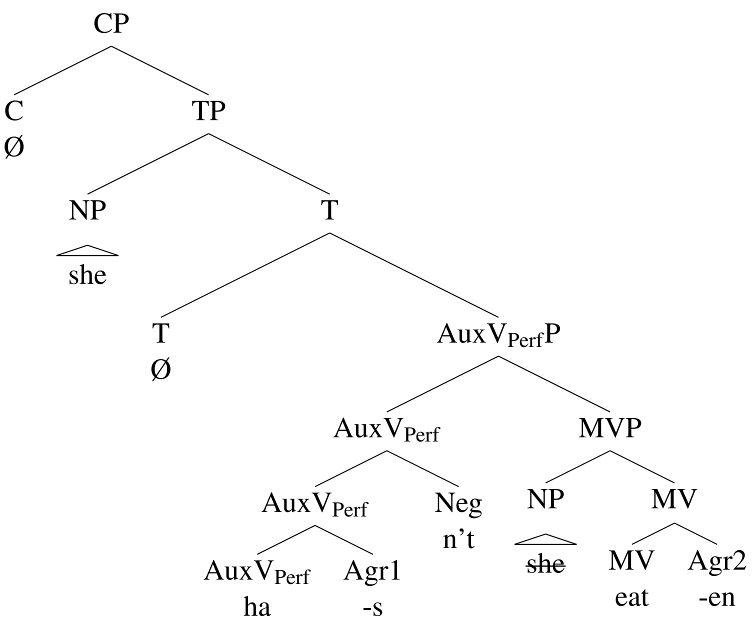
We also need to add two new steps to the N2W procedure, new steps 4 and 5 for head selection:

(32) Procedure for N2W, where α is the top of the stack:

```
1. Locate the phase head and make it the top of the stack $\alpha$; go to step 2.
2. Is there an element in the numeration Y with a feature [$S_{AL}$:$\alpha$]?
   (a) If no, go to step 3.
   (b) If yes, make Y the top of the stack $\alpha$ and return to step 2.
3. Does $\alpha$ have an unchecked [$S_S$:X] feature?
   (a) If no, move $\alpha$ to the workspace; go to step 4.
   (b) If yes, locate an X in the numeration or the workspace; move X to the workspace or copy
       X from the workspace to the current location in the tree; check off the [$S_S$:X] feature on
       $\alpha$ and return to step 3.
4. Does $\alpha$ have an unchecked [$S_H$:$\delta$] feature?
   (a) If no, go to step 5.
   (b) If yes, locate a $\delta$ in the numeration and move it to the workspace; check off the [$S_H$:$\delta$]
       feature on $\alpha$ and return to step 4.
5. Is there an item in the numeration H with an [$S_{HA}$:$\alpha$] feature?
   (a) If no, go to step 6.
   (b) If yes, move H to the workspace and return to step 5.
6. Does $\alpha$ have an [$S_C$:$\beta$] feature?
   (a) If no, go to step 7.
   (b) If yes, is there a $\beta$ in the numeration?
       i. If no, copy a $\beta$ from elsewhere in the workspace to the current location in the workspace;
          go to step 7.
       ii. If yes, remove $\alpha$ from the stack, make $\beta$ the top of the stack and return to step 2.
7. Is there an element $\gamma$ with an [$S_{AR}$:$\alpha$] feature in the numeration?
   (a) If no, remove $\alpha$ from the stack and go to the next item on the top of the stack, returning
       to step 2; if the stack is empty, terminate.
   (b) If yes, remove $\alpha$ from the stack and make $\gamma$ the top of the stack; return to step 2.
```

I will illustrate how the system works with *She hasn't eaten*. I assume this has the following structure:

(33)



Each verb has an Agr morpheme head-adjoined to it, and affixal negation adjoins to the first (highest) AuxV. The highest AuxV probably moves to T.

In a companion paper, I show how it is ensured that the numeration includes one Agr head for each verb. Here, I simply assume that the numeration has been correctly selected. The numeration for this CP phase is the following, where affixal negation (i.e., one with the feature [S_{HA}:AuxV]) has been selected:

(34)

CP Numeration				
C Ø [S_{C}:T]	Ph:N	T Ø [S_{S}:N,S_{C}:V]	AuxV$_{\text{Perf}}$ have [S_{H}:Agr,S_{C}:{MV,Pass,Prog}]	MV eat [S_{S}:N,S_{H}:Agr]

Agr1	Agr2	Neg n't [S_{HA}:AuxV]

Workspace:

—*empty*—

Recall that NPs are their own phases, and in the CP phase are represented with a placeholder, Ph:N. I do not show a form for the Agr morphemes here, as their form is determined in the syntax, through agreement.

Importantly, all elements of category V now have an [S_{H}:Agr] feature in addition to whatever other features they have. The affixal version of Neg in English has an [S_{HA}:AuxV] feature. N2W starts by making the phase head, C, the top of the stack. There is nothing with an [S_{AL}:C] feature in step 2. In step 3, C does not have an unchecked [S_{S}] feature, so C is moved to the workspace:

(35)

CP Numeration after First Selection				
	Ph:N	T Ø [S_S:N,S_C:V]	AuxV$_{Perf}$ have [S_H:Agr,S_C:{MV,Pass,Prog}]	MV eat [S_S:N,S_H:Agr]

Agr1	Agr2	Neg n't [S_{HA}:AuxV]

Workspace:
C
Ø
[S_C:T]

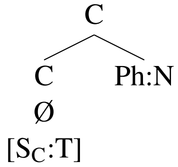
N2W continues to step 4. C does not have an unchecked head feature. In step 5, there is no item in the numeration with an [S_{HA}:C] feature. In step 6, C does have an [S_C:T] feature and there is a T in the numeration, so C is removed from the stack and T is made the top of the stack. N2W returns to step 2. There is nothing in the numeration with an [S_{AL}:T] feature. In step 3, T does have an unchecked [S_S:N] feature, so N2W locates an N. The only one is the placeholder, Ph:N, so this is moved to the workspace:

(36)

CP Numeration after Second Selection				
		T Ø [~~S_S:N~~,S_C:V]	AuxV$_{Perf}$ have [S_H:Agr,S_C:{MV,Pass,Prog}]	MV eat [S_S:N,S_H:Agr]

Agr1	Agr2	Neg n't [S_{HA}:AuxV]

Workspace:



At this point the NP phase needs to be expanded. L2N will choose a new numeration, which will then be moved one by one to the workspace. I assume that this happens, and in subsequent trees I show the full NP.

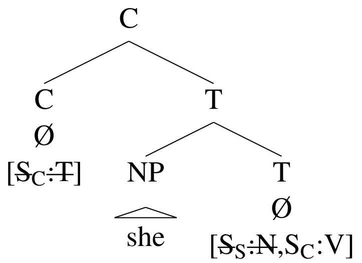
Moving Ph:N to the workspace checks off the selectional feature on T, so when N2W returns to step 3, T no longer has an unchecked [S_S] feature. This means that T is moved to the workspace:

(37)

CP Numeration after Third Selection				
			AuxV$_{Perf}$ have [S_H:Agr,S_C:{MV,Pass,Prog}]	MV eat [S_S:N,S_H:Agr]

Agr1	Agr2	Neg n't [S_{HA}:AuxV]

Workspace:



T projects, since it selects the NP as a specifier; the [S_C:T] feature on C is checked off.

In steps 4 and 5, T does not have an [S_H] feature, nor is there anything in the numeration with an [S_{HA}:T] feature. In step 6, T does have an [S_C:V] feature, so T is removed from the stack. N2W must locate a V and make it the top of the stack. There are two, an AuxV and an MV. The principle in (7) chooses AuxV, so it is made the top of the stack and N2W returns to step 2.

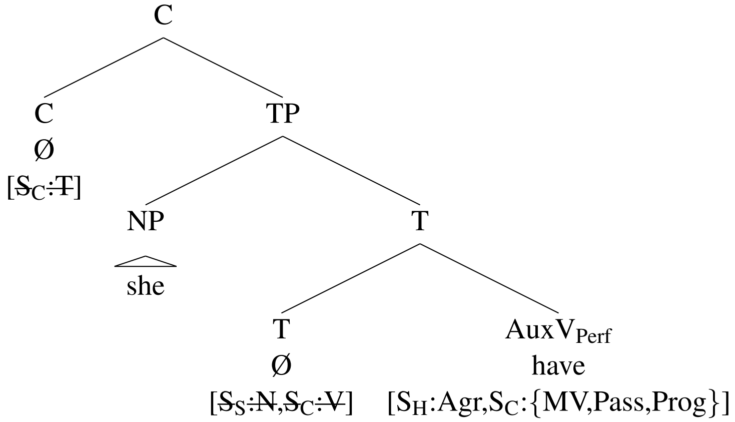
In step 2, there is nothing with an [S_{AL}:AuxV] feature. In step 3, AuxV does not have an [S_S] feature, so it is moved to the workspace:

(38)

CP Numeration after Fourth Selection				
				MV eat [S_S:N,S_H:Agr]

Agr1	Agr2	Neg n't [S_{HA}:AuxV]

Workspace:



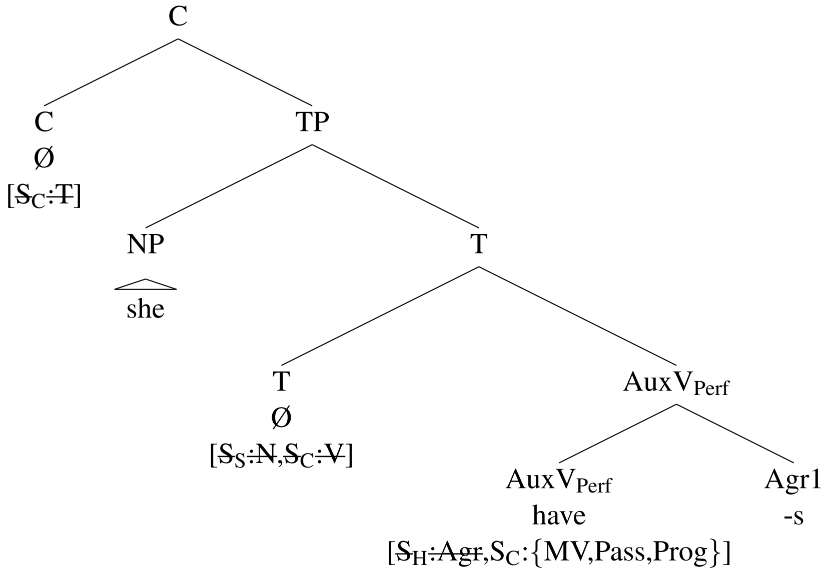
This checks off the [S_C:V] feature on T. N2W goes to step 4. AuxV does have an unchecked [S_H:Agr] feature. N2W therefore locates an Agr in the numeration and moves it to the workspace (it does not matter which one):

(39)

CP Numeration after Fifth Selection				
				MV eat [S_S:N,S_H:Agr]

	Agr2	Neg n't [S_{HA}:AuxV]

Workspace:

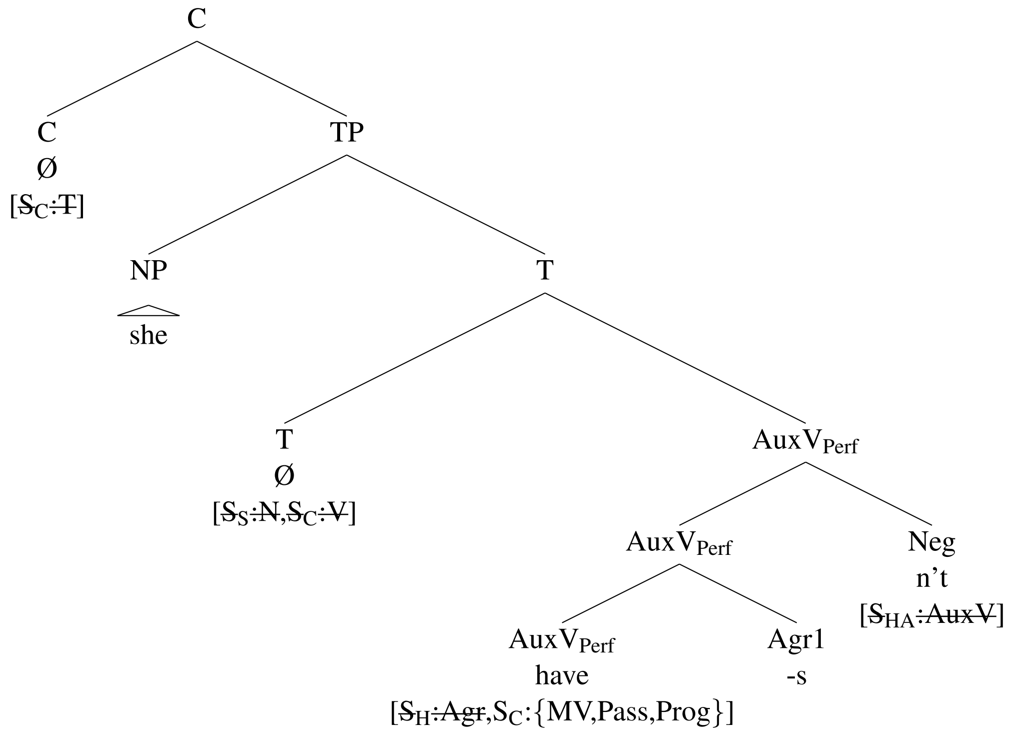


The Agr will be spelled out as *-s* through the process of agreeing with T, not important here. The AuxV will also be pronounced *ha-* rather than *have* in the context of *-s*. Merging Agr checks off the [S_H:Agr] feature on AuxV. N2W returns to step 4, but AuxV now no longer has an unchecked [S_H] feature, so it goes to step 5. There is an item in the numeration with an [S_{HA}:AuxV] feature, namely, Neg. So Neg is moved to the workspace:

(40)

CP Numeration after Sixth Selection				
				MV eat [S_S:N,S_H:Agr]
	Agr2			

Workspace:


C
C
Ø
[~~S_C:T~~]
TP
NP
she
T
T
Ø
[~~S_S:N~~,~~S_C:V~~]
$\text{AuxV}_{\text{Perf}}$
$\text{AuxV}_{\text{Perf}}$
Neg
n't
[~~S_{HA}:AuxV~~]
$\text{AuxV}_{\text{Perf}}$
have
[~~S_H:Agr~~,S_C:{MV,Pass,Prog}]
Agr1
-s

AuxV projects, since Neg head-selects AuxV as an adjunct. Merging Neg checks off the [S_{HA}:AuxV] feature of Neg. N2W returns to step 5, but now there is nothing with that feature in the numeration, so it moves to step 6. In step 6, AuxV does have an [S_C] feature, and there is something in the numeration that satisfies it, namely, MV. So AuxV is removed from the stack and MV is put on top of the stack. N2W returns to step 2.

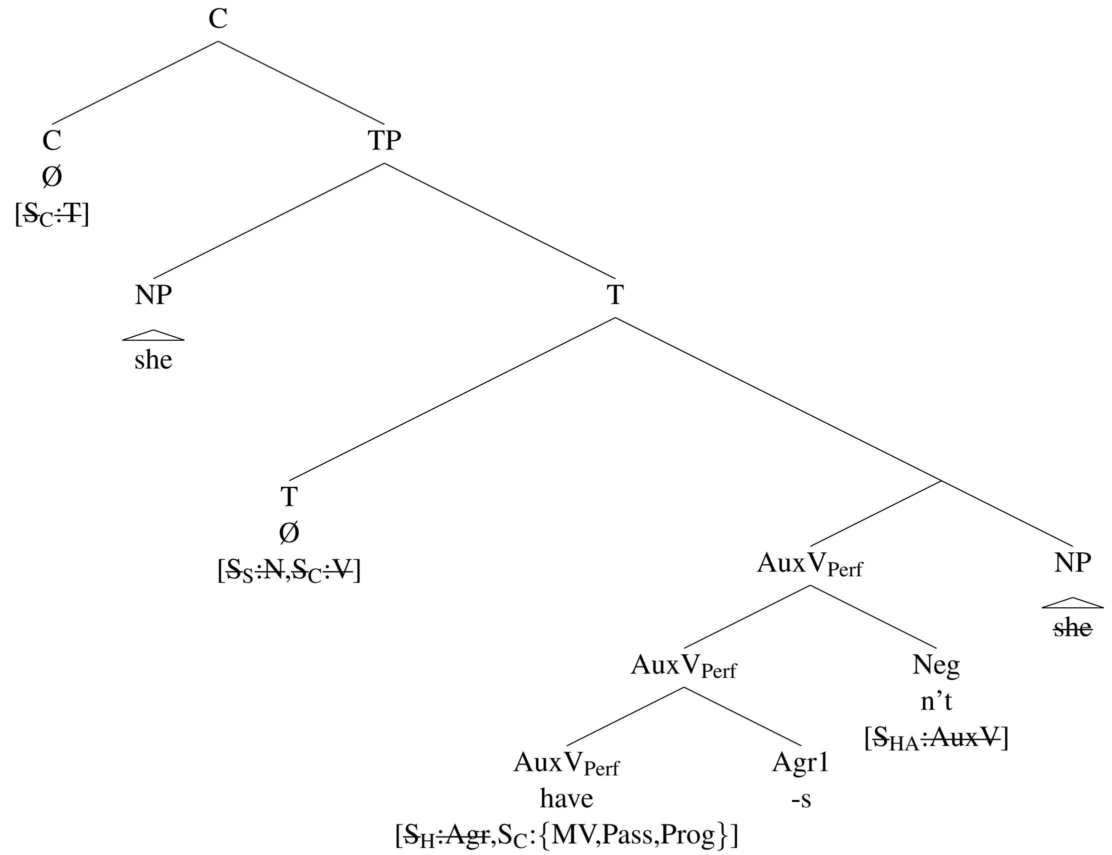
In step 2, there is nothing with an [S_{AL}:(M)V] feature. In step 3, MV does have an unchecked [S_S:N] feature. N2W scans the numeration and workspace for something of category N. The only one is the subject NP, *she*. So this NP is copied to the base of the tree. Now, since Agr1 and Neg adjoined via head features, they are made a complex head with AuxV, so the entire AuxV is shoved to a left branch:

(41)

CP Numeration after Copying the Subject				
				MV eat [~~S_S:N~~,S_H:Agr]

	Agr2	

Workspace:

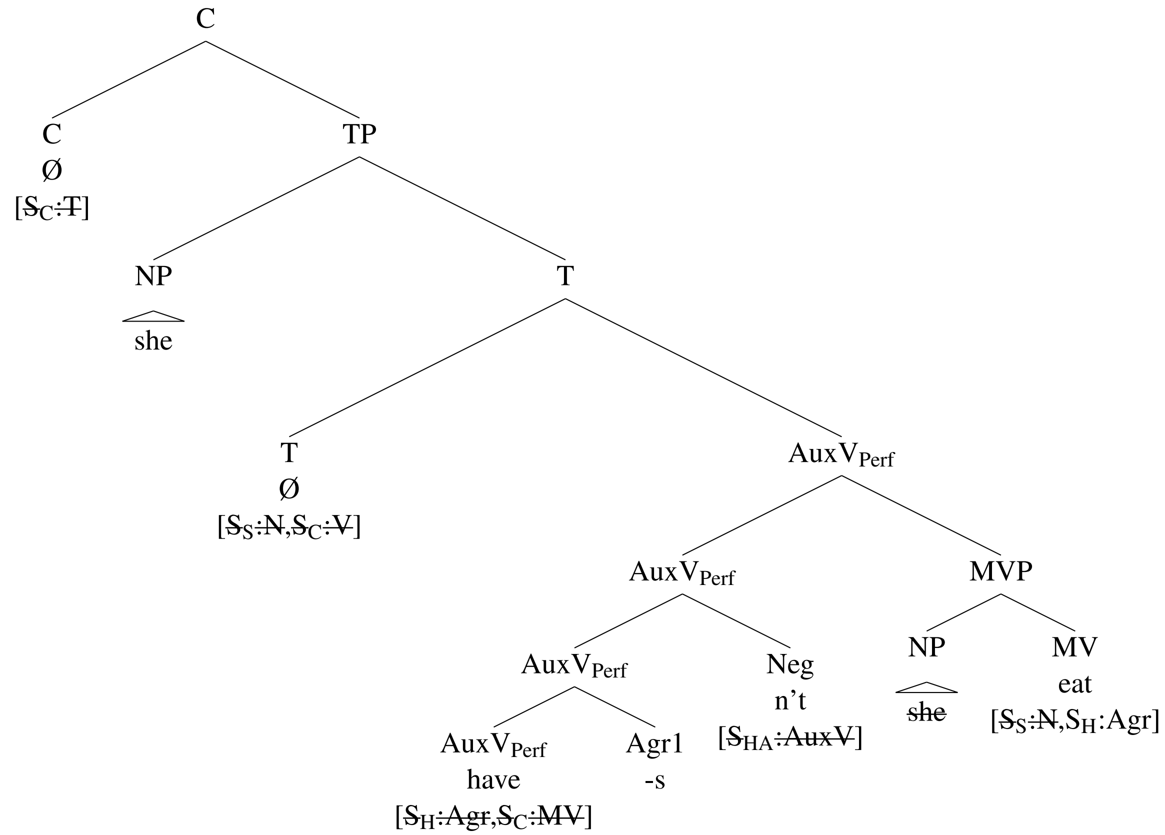

C
C
Ø
[~~S_C:T~~]
TP
NP
she
T
T
Ø
[~~S_S:N~~,~~S_C:V~~]
$\text{AuxV}_{\text{Perf}}$
NP
~~she~~
$\text{AuxV}_{\text{Perf}}$
Neg
n't
[~~S_{HA}:AuxV~~]
$\text{AuxV}_{\text{Perf}}$
have
[~~S_H:Agr~~,S_C:{MV,Pass,Prog}]
Agr1
-s

This checks off the [S_S:N] feature on MV, so when N2W returns to step 3, it no longer has an unchecked [S_S] feature. This results in MV being moved to the workspace:

(42)

CP Numeration after Seventh Selection				
	Agr2			

Workspace:

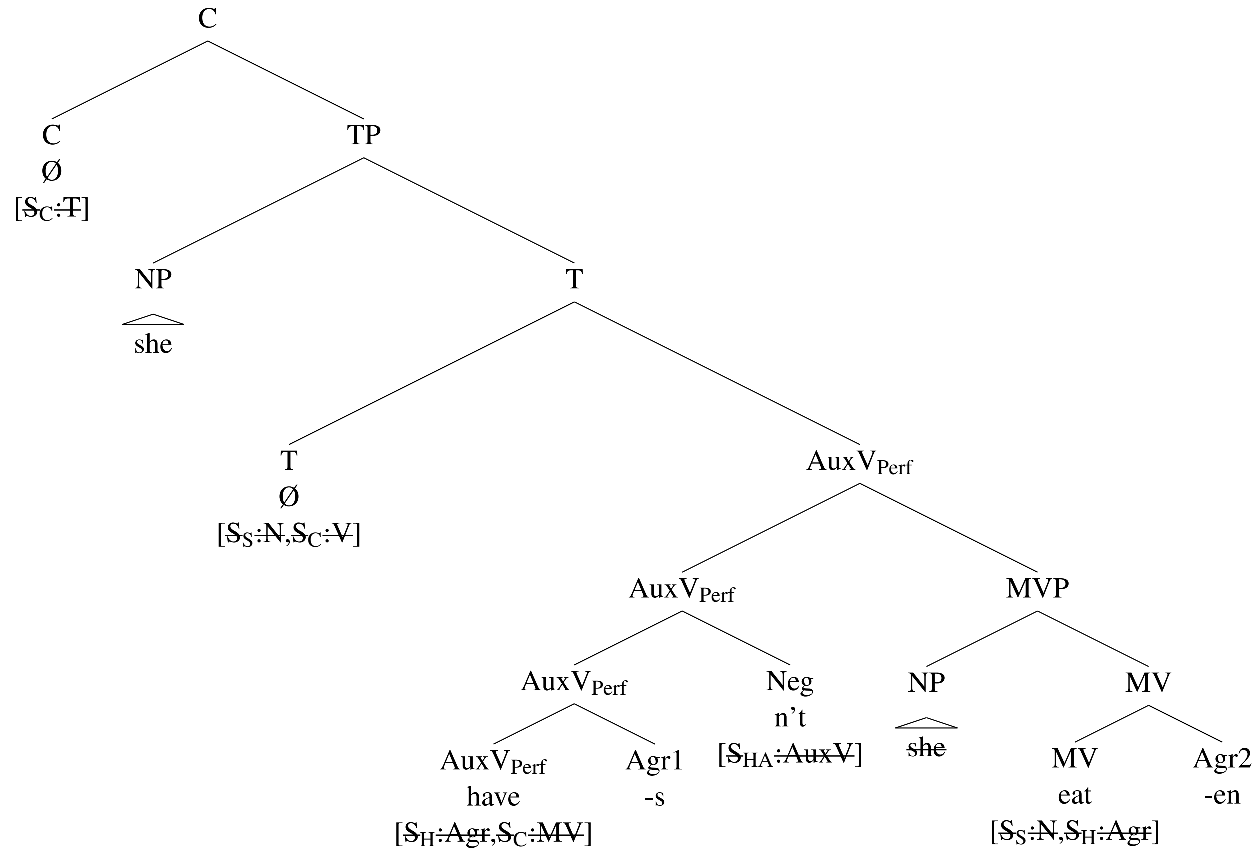

C
C
Ø
[~~S_C:T~~]
TP
NP
she
T
T
Ø
[~~S_S:N~~,~~S_C:V~~]
AuxV_Perf
AuxV_Perf
MVP
AuxV_Perf
Neg
n't
[~~S_HA:AuxV~~]
NP
~~she~~
MV
eat
[~~S_S:N~~,S_H:Agr]
AuxV_Perf
have
[~~S_H:Agr~~,~~S_C:MV~~]
Agr1
-s

This results in checking off the [S_C] feature of AuxV (which I now collapse to save space). N2W goes to step 4. MV does have an unchecked [S_H:Agr] feature, so Agr2 is located in the numeration and moved to the workspace:

(43)

CP Numeration after Eighth Selection				

Workspace:



This checks off the [S_H:Agr] feature of MV. Agr2 will be spelled out as *-en*, by virtue of agreeing with AuxV$_{\text{Perf}}$. N2W returns to step 4, but now MV does not have an unchecked [S_H] feature. In step 5, there is nothing in the numeration anymore. In step 6, MV does not have an [S_C] feature. In step 7, there is nothing in the numeration, so MV is removed from the stack. The stack is now empty, so N2W terminates. There are no unchecked features in the tree, so this derivation is complete.

The procedure for N2W now correctly puts an Agr morpheme on each verb in English. It also correctly locates affixal negation, without having to stipulate which AuxV it attaches to. Affixal negation only requires that it attach to *some* AuxV ([S_{HA}:AuxV]). The fact that the derivation takes place in a linear order from left to right has the result that *n't* attaches to the *first* AuxV that is made the top of the stack. If there were additional AuxVs in the numeration, *n't* would have been removed from the numeration by the time they were made the top of the stack, so *n't* could not attach to them.

This will essentially be the account of the Bulgarian definite marker. We need nothing more than a left-to-right derivation to account for it. Before proceeding to Bulgarian, though, I show that the current system works for coordination, as well.

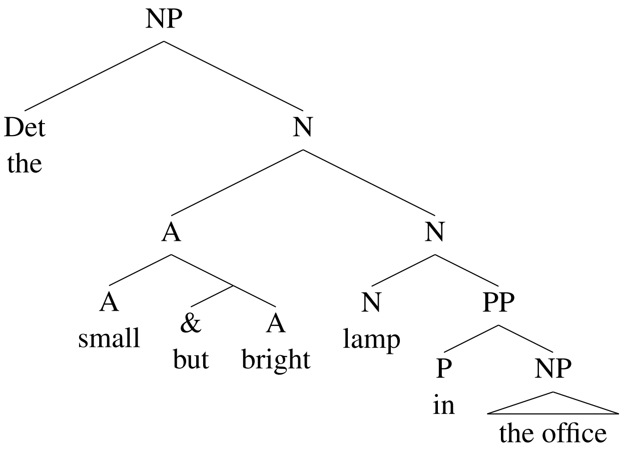
3.4 Coordination

Consider a variant of the nominal phase from above:

(44) the small but bright lamp in the office

I assume that the structure that will be built is something like the following, where coordinators are category "&":

(45)



I assume a non-headed structure for coordination, where the coordinator combines with the two conjuncts to create something that is the same category as the two conjuncts (see, e.g., Moltmann 1992, Chaves 2012, Al Khalaf 2015). See more on this below.

The question is whether the N2W procedure will build this structure as desired, or whether it needs to be modified in any way. Ignoring the PP, the numeration is the following:

(46)

NP Numeration: Initial State				
Det the [S$_{\text{AL}}$:{N,AN,NumN}]	A small [S$_{\text{AL}}$:N]	A bright [S$_{\text{AL}}$:N]	N lamp	&[A] but [S$_{\text{AR}}$:A,S$_{\text{C}}$:A]

I assume that coordinators are elements that require two items of the same category. They combine these two items to create another item of the same category. (I assume that longer sequences of coordination are actually a series of binary coordination; in English, only the final coordinator must be pronounced.) I assume that all coordinators have a feature α which is a variable for the syntactic categories in the languages. When a coordinator is chosen from the lexicon and moved to the numeration by L2N, α has to be specified as one of the syntactic categories. L2N then has to select two elements of that syntactic category and put them in the numeration. In our example, α is specified as A, so & is &[A]. This is shown in the numeration above. The numeration does include two elements of category A (because L2N followed the requirement).

As in the NP phase derivation gone through above, N is identified as the phase head and made the top of the stack. In step 2, there are three elements with the feature [S$_{\text{AL}}$:N], namely, the two As, and Det. The principle in (7) picks Det, since it is functional while the two As are lexical, and Det also has the largest disjunctive set of selectional features. Det is made the top of the stack, and N2W returns to step 2. There are no items in the numeration with the feature [S$_{\text{AL}}$:Det]. In step 3, Det does not have an [S$_{\text{S}}$] feature, so it is moved to the workspace:

(47)

NP Numeration after First Selection				
	A small [S$_{\text{AL}}$:N]	A bright [S$_{\text{AL}}$:N]	N lamp	&[A] but [S$_{\text{AR}}$:A,S$_{\text{C}}$:A]

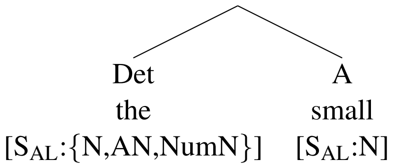
Workspace:
Det
the
[S$_{\text{AL}}$:{N,AN,NumN}]

Steps 4, 5, 6, and 7 come up empty, so Det is removed from the stack. This makes N the top of the stack again, and N2W returns to step 2. In step 2, there are two elements with the feature [S_{AL}:N], namely, the two As. The principle in (7) does not resolve which one to pick, since they are both lexical and they have the same selectional features. So it chooses one at random. In this case it chooses *small*. This A is made the top of the stack and N2W returns to step 2. In step 2, there are no elements with the feature [S_{AL}:A]. A does not have an unchecked [S_S] feature in step 3, so A is moved to the workspace:

(48)

NP Numeration after Second Selection				
		A bright [S_{AL}:N]	N lamp	&[A] but [S_{AR}:A,S_C:A]

Workspace:



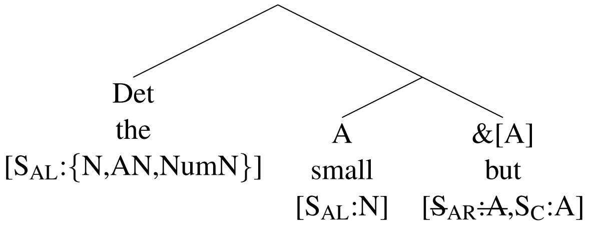
N2W moves on to steps 4, 5, and 6, which all answer 'no'. In step 7, however, there is an element with the feature [S_{AR}:A] in the numeration, namely, the coordinator. This results in A being removed from the top of the stack, and & is made the top of the stack. N2W returns to step 2.

In step 2, there is no element with an [S_{AL}:&] feature, and in step 3, & does not have an [S_S] feature. So & is moved to the workspace, where it merges with A:

(49)

NP Numeration after Third Selection				
		A bright [S_{AL}:N]	N lamp	

Workspace:



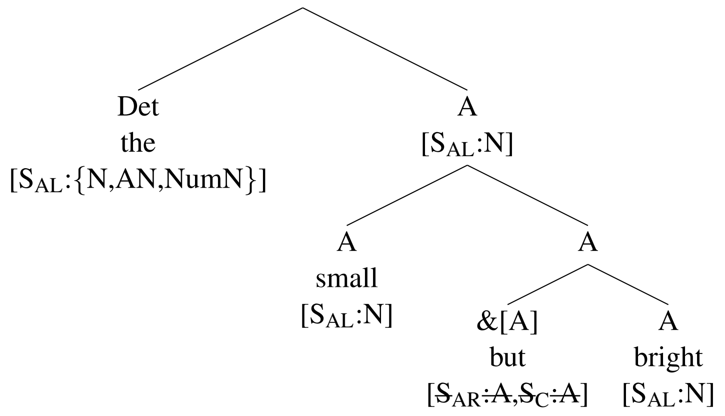
This checks the [S_{AR}] feature of &. There is nothing in steps 4 and 5, but in step 6, & has an [S_C:A] feature. This results in & being removed from the stack, and the next A, *bright*, being put on the top of the stack. N2W returns to step 2.

In step 2, there is no item with an [S_{AL}:A] feature. A does not have an [S_S] feature in step 3, so it is moved to the workspace and merged with &:

(50)

NP Numeration after Fourth Selection				
			N lamp	

Workspace:



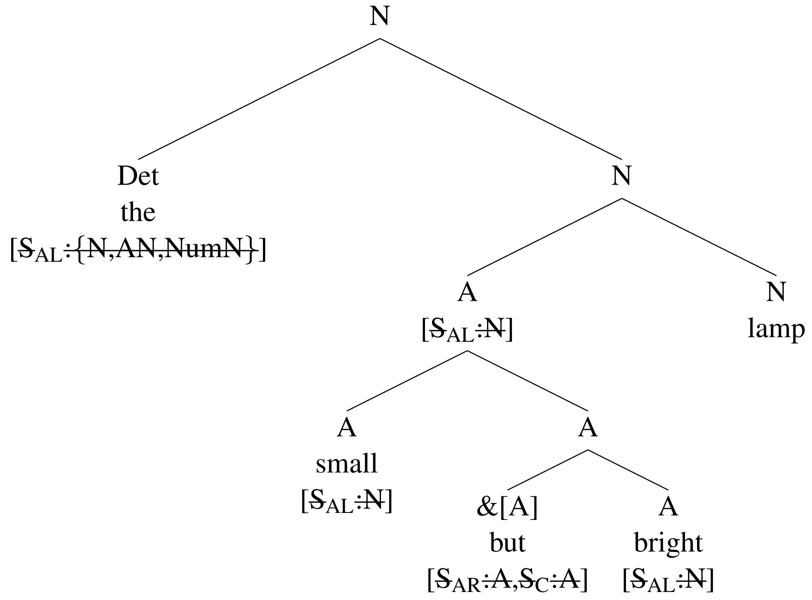
This checks the [S_C:A] feature of &, and the label A can now project to all the nodes combined by &. I assume that the selectional features of the conjuncts also percolate to the topmost node, as shown.

N2W moves to step 4, but comes up empty there and in steps 5, 6, and 7. So A is removed from the stack, which makes N the top of the stack again. N2W returns to step 2. There is now nothing left in the numeration but the N, so in step 3, N is moved to the workspace:

(51)

NP Numeration after Fifth Selection				

Workspace:



Since the topmost A node has the feature [S_{AL}:N], the N merges with it, shoving it onto a left branch, as shown. N now projects. This checks off the [S_{AL}:N] features of all the adjuncts. In the coordination, this checks that feature of the topmost node, and, by percolation, all the features of the conjuncts, too. N2W moves to step 4, but comes up empty there and in steps 5, 6, and 7. N is removed from the stack. The stack is now empty, so N2W terminates. There is nothing in the numeration and all features have been checked. A well-formed derivation has been completed.

As can be seen, the procedure as given accounts for coordination, with nothing needed except the selectional features [S_{AR}:A,S_C:A] on the coordinator. One thing to point out is what would happen if there had been adjuncts or complements to the As. Those would have combined in the left conjunct before the coordinator, and then the right conjunct afterwards, resulting in phrasal coordination. I assume that this is correct, and there is no true head coordination; see Bruening (2018b: 27–29) and Bruening (2018a: e71–e72).

3.5 Summary

In this section, I have spelled out N2W, the procedure that selects items from the numeration and moves them to the workspace. As I have shown, this procedure correctly builds CP and NP phases, and it also correctly builds coordinated phrases. It also puts English affixal negation in the correct place, namely, on the first auxiliary verb, simply because it operates in a left-to-right fashion. As I will now show, nothing more is needed to account for a similar placement for an affixal determiner in Bulgarian.

4 The Bulgarian Definite Marker

The definite marker in Bulgarian appears as a suffix on the first word of a certain type within the NP. The relevant words are those that can bear nominal inflection. These include the head noun itself, adjectives, possessive pronouns, and numerals (see, e.g., Franks 2001). If there is only a head noun, the suffix attaches to that (52a); if there is a prenominal adjective, the suffix attaches to that (52b); if there is a possessive pronoun or a numeral before an adjective, then the definite suffix attaches to that (52c–d):

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

a. kniga-**ta**
book-Def

b. xubava-**ta** kniga
nice-Def book

c. moja-**ta** xubava kniga
my-Def nice book

d. tri-**te** novi knigi
three-Def new books

If an adjective is modified by an adverb, the definite marker attaches to the head adjective, and not to the first element (the adverb):

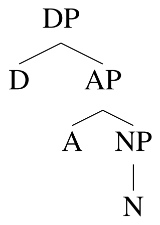
(53) (Embick & Noyer 2001: (23a–b))

a. * mnog-**ət** star teatər
very-Def old theater

b. mnogo starij-**ə** teatər
very old-Def theater

The issue with this placement is that it seems to refer to linear order: the definite marker goes on the *first* element of the appropriate type. Since in many approaches, the syntax deals only in hierarchy and not in linear order, this placement is problematic. Accordingly, Embick & Noyer (2001) propose a post-syntactic *lowering* analysis of the placement of the Bulgarian definite marker. At a level after the syntax, the D head lowers onto the head of its complement. This analysis assumes the DP Hypothesis, where the definite marker is a head D that projects a phrase and takes as its complement a sequence of functional projections terminating in the lexical NP. This analysis also requires that D take A as its complement, in order to lower D onto A in examples like (52b). A then takes N as its complement (as was proposed by Abney 1987):

(54)



A taking N as its complement and projecting is an analysis that has been discredited; see, for example, Hankamer & Mikkelsen (2005). If this is not a viable analysis of adjectives, then this is not a viable analysis of the definite marker.

An empirical problem for the lowering analysis comes from coordinated adjectives. If two adjectives are coordinated, the definite marker appears only on the first one (Harizanov & Gribanova 2014):

(55) prohladna-**ta** i sveža večer
cool-Def and fresh evening
'the cool and fresh evening'

We should compare this to the process that puts tense and agreement on the main verb in English, which Embick & Noyer (2001) also analyze as lowering. This process in English must apply in an across-the-board fashion:

(56) She caught and ate/*eat the fish.

This makes the lowering analysis of Bulgarian suspect, since it does not apply in an across-the-board fashion as it would be expected to. (Note that the English facts follow from the analysis in section 3.3: Every verb head-selects an Agr head.[2])

The generalization concerning the Bulgarian definite marker is that it appears on the *first* element of the right type in the NP. It does not appear on the head of the complement of D, which is how Embick & Noyer (2001) analyze it. This placement appears structural because it skips things of the wrong type (adverbs); but it appears linear because it goes on the *first* element of the right type, even in coordination.

For lack of a better term, I will call the heads of the appropriate type in Bulgarian, namely, those that bear nominal inflection, "[+N] heads." The generalization concerning the placement of the Bulgarian definite marker is then as follows:

(57) Generalization: The Bulgarian definite marker attaches to the first [+N] head in the nominal.

As I will now show, the system developed here for syntax in general accounts for the placement of the Bulgarian definite marker without any additional stipulations.

4.1 Nominal Concord: Agr Heads

Before getting to the proposal, it is important that Bulgarian has nominal concord. Demonstratives, adjectives, numerals, possessive pronouns, and head nouns all bear nominal concord in Bulgarian. Nominal concord typically takes the form of a final vowel, marking number, gender, and case. So example (52d) should be fully glossed as follows (ignoring the specific features):

[2] English affixal negation behaves differently: it can go on either or both of two conjoined AuxVs:

(i) a. She can't and won't reset the machine.
b. She can but won't reset the machine.
c. She won't but could reset the machine.

It seems likely that coordination here involves coordination of larger categories plus ellipsis, so I will leave this aside.

(58) tr-i-te nov-i knig-i
three-Agr-Def new-Agr book-Agr
'the three new books'

Adverbs do not bear nominal concord (Agr) markers, instead they have an invariant *-o* (which I will not gloss and will ignore in the analysis, but a complete analysis would include it):

(59) mnogo star-ij-ə teatər-Ø
very old-Agr-Def theater-Agr

We can therefore restate the generalization as the following:

(60) Generalization (revised): The Bulgarian definite marker attaches to the first [+N] head in the nominal, outside of Agr.

Since I am assuming that all complex forms are put together by the syntax, the numeration of a nominal phase will have to include separate Agr heads for each [+N] head in the nominal. In a companion paper, this is accomplished by the operation L2N, which is not important here. I will simply assume that the correct numeration has been selected for each NP phase under consideration. What is important here is that, in terms of the system developed in section 3.3, all [+N] heads in Bulgarian will have an [S_{H}:Agr] feature:

(61) All [+N] heads in Bulgarian have an [S_{H}:Agr] feature.

We can now go on to explain the placement of the Bulgarian definite article.

4.2 The Bulgarian Definite Article

The system developed here will now correctly place the Bulgarian definite article, with no additional stipulations or constraints. All we need is to say that the definite article, which I will call "Def," has the feature [S_{HA}:+N]. That is, it is like the English affixal negation *n't* in that it requires that it adjoin (like an adjunct) to a head. English affixal negation head-selects an AuxV; Bulgarian Def head-selects a [+N] head.

I show how the system works with the phrase 'the three new books' in (58). The numeration for this phrase is shown below, and the workspace is empty:

(62)

Numeration						
Num	**Def**	**A**	**N**	**Agr1**	**Agr2**	**Agr3**
tr		**nov**	**knig**			
'three'		'new'	'book'			
[+N,S_{H}:Agr,S_{AL}:{N,AN}]	[S_{HA}:+N]	[+N,S_{H}:Agr,S_{AL}:N]	[+N,S_{H}:Agr]			

Workspace:

—empty—

In addition to the lexical items, there are three Agr heads and the definite article (Def). I do not show the phonological form of these items in the numeration, as their forms are determined contextually. In Bulgarian, the order of elements within the nominal is fixed as Dem/Poss > Num > A. As in section 3.2, I account for this by giving those on the left a larger set of disjunctive selectional features than those on the right. In particular, A has the feature [S_{AL}:N], Num has the feature [S_{AL}:{N,AN}], and Dem/Poss has the feature [S_{AL}:{N,AN,NumN}].

The derivation begins with step 1. N2W locates the phase head, N, and puts it on the top of the stack. In step 2, there are two items with the feature [S$_{\mathrm{AL}}$:N], Num and A. Num is functional while A is lexical, and it also has the largest set of disjunctive selectional features, so Num is selected according to (7) and put on the top of the stack. N2W returns to step 2, now considering Num α for the procedure.

In step 2, there is no element with the feature [S$_{\mathrm{AL}}$:Num], so N2W goes to step 3. Num does not have an unchecked [S$_{\mathrm{S}}$] feature, so Num is moved to the workspace:

(63)

Numeration after First Selection						
	Def	**A** **nov** ‘new’ [+N,S$_{\mathrm{H}}$:Agr,S$_{\mathrm{AL}}$:N]	**N** **knig** ‘book’ [+N,S$_{\mathrm{H}}$:Agr]	**Agr1**	**Agr2**	**Agr3**
	[S$_{\mathrm{HA}}$:+N]					

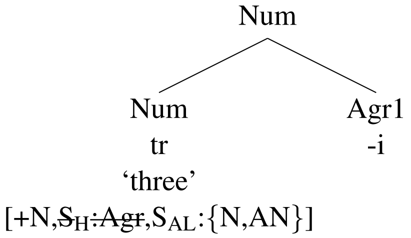
Workspace:
Num
tr
‘three’
[+N,S$_{\mathrm{H}}$:Agr,S$_{\mathrm{AL}}$:{N,AN}]

N2W now goes to step 4. Num does have an unchecked [S$_{\mathrm{H}}$:Agr] feature, so an Agr is moved from the numeration to the workspace and merged with Num (it does not matter which one):

(64)

Numeration after Second Selection						
	Def	**A** **nov** ‘new’ [+N,S$_{\mathrm{H}}$:Agr,S$_{\mathrm{AL}}$:N]	**N** **knig** ‘book’ [+N,S$_{\mathrm{H}}$:Agr]		**Agr2**	**Agr3**
	[S$_{\mathrm{HA}}$:+N]					

Workspace:



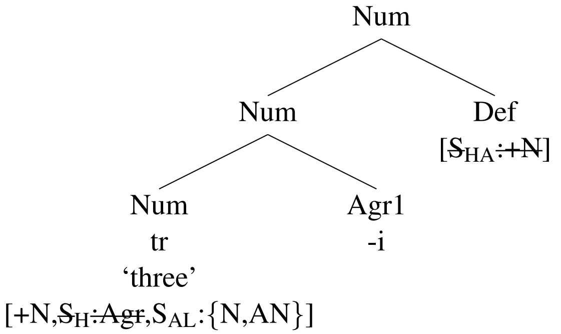
Num projects, because it selects Agr1. Agr1 will receive its features however nominal concord works; this is not the issue here, any theory should work. I will simply show the phonological form of the Agr nodes in the trees here, without showing how that form is determined.

Merging Agr1 checks off the [S$_{\mathrm{H}}$:Agr] feature of Num. N2W returns to step 4, but now Num has no more unchecked [S$_{\mathrm{H}}$] features, so it goes on to step 5. Here is the important part. There is an element in the numeration with an [S$_{\mathrm{HA}}$:+N] feature, namely, Def. So N2W moves Def to the workspace and merges it with the existing tree:

(65)

Numeration after Third Selection						
		A **nov** ‘new’ [+N,S$_{\mathrm{H}}$:Agr,S$_{\mathrm{AL}}$:N]	**N** **knig** ‘book’ [+N,S$_{\mathrm{H}}$:Agr]		**Agr2**	**Agr3**

Workspace:



Num projects again, because Def is merged in step 5, which is like adjunct selection (see section 3.3). By merging with a [+N] element, the [S_{HA}:+N] feature of Def is checked off.

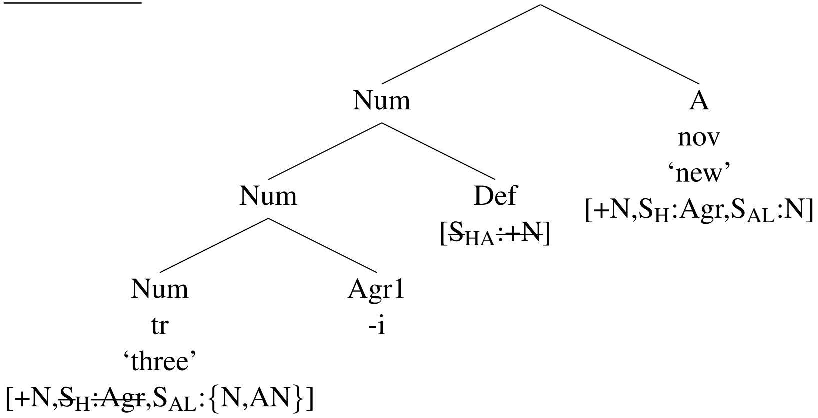
N2W returns to step 5, but now there are no more items in the numeration with an [S_H:Num] or [S_{HA}:+N] feature. So N2W goes to step 6. Num does not have an [S_C] feature. In step 7, there is nothing with an [S_{AR}:Num/+N] feature, so Num is removed from the stack. N is now the top of the stack again, and N2W returns to step 2.

In step 2, there is an item in the numeration with the feature [S_{AL}:N], namely, the A. So the A is made the top of the stack, and N2W returns to step 2. In step 2, there is nothing with an [S_{AL}:A] or [S_{AL}:+N] feature. A does not have an [S_S] feature in step 3, so A is moved to the workspace. Because Agr1 and Def merged with Num via head features, they form a complex head with Num. A then has to merge with the entire complex head:

(66)

Numeration after Fourth Selection						
			N **knig** 'book' [+N,S_H:Agr]		**Agr2**	**Agr3**

Workspace:

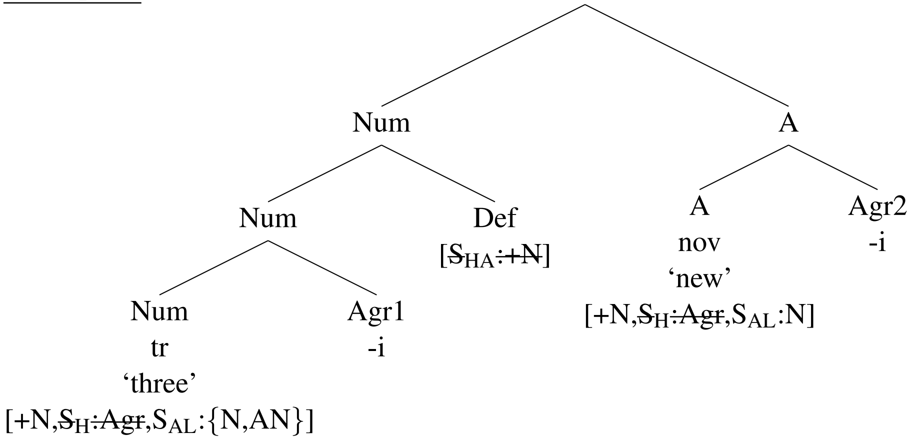


N2W moves to step 4. A does have an unchecked [S_H:Agr] feature, so an Agr is moved from the numeration to the workspace and merged with A:

(67)

Numeration after Fifth Selection						
			N **knig** 'book' [+N,S_H:Agr]			**Agr3**

Workspace:


Num
A
Num
Def
[~~S_{HA}:+N~~]
A
nov
'new'
[+N,~~S_H:Agr~~,S_{AL}:N]
Agr2
-i
Num
tr
'three'
[+N,~~S_H:Agr~~,S_{AL}:{N,AN}]
Agr1
-i

A projects, because it selects Agr. This checks off the feature [S_H:Agr] on A. N2W returns to step 4, but now A does not have an unchecked [S_H] feature, so it goes on to step 5. Def is no longer in the numeration, so the numeration no longer contains anything with an [S_{HA}:+N] feature. N2W therefore moves on to step 6, but A does not have an [S_C] feature. In step 7, there is nothing in the numeration with an [S_{AR}] feature, so A is removed from the stack. This makes N the top of the stack again, and N2W returns to step 2.

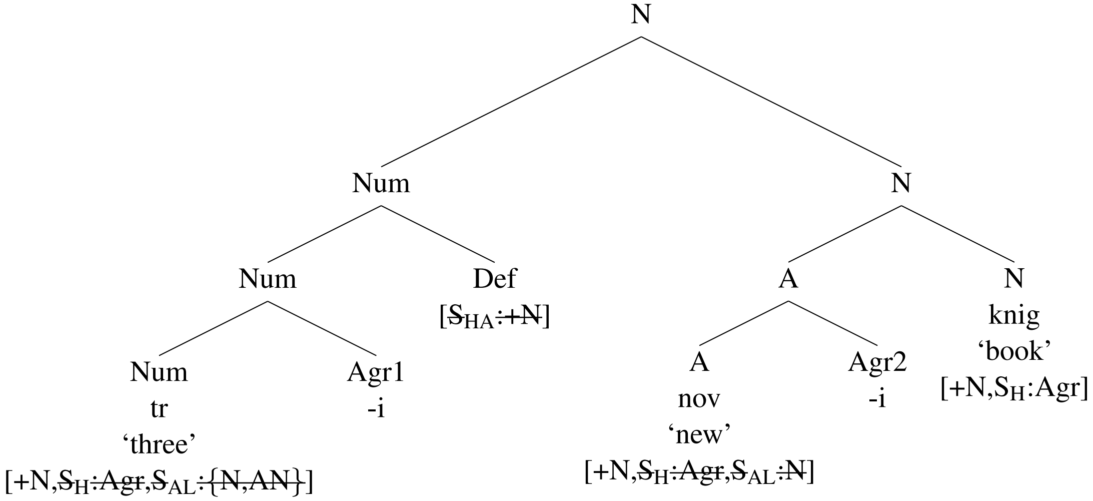
At this point it should be clear why Def was put on the first [+N] element: because step 5 applies whenever it can, it will apply at the first point possible. But then after that, Def will have been moved to the workspace already, so at subsequent instances of step 5, when other [+N] elements are the top of the stack, there will no longer be an element with an [S_{HA}:+N] feature. The placement of Def in Bulgarian then just follows from a left-to-right syntax, the same way the placement of English affixal negation followed.

We still need to complete the derivation. N is now the top of the stack, and N2W has returned to step 2. In step 2, there is no longer any item with an [S_{AL}:N] feature, and in step 3, N does not have an unchecked [S_S] feature. This results in N being moved to the workspace, pushing A to a left branch:

(68)

Numeration after Sixth Selection						
						Agr3

Workspace:

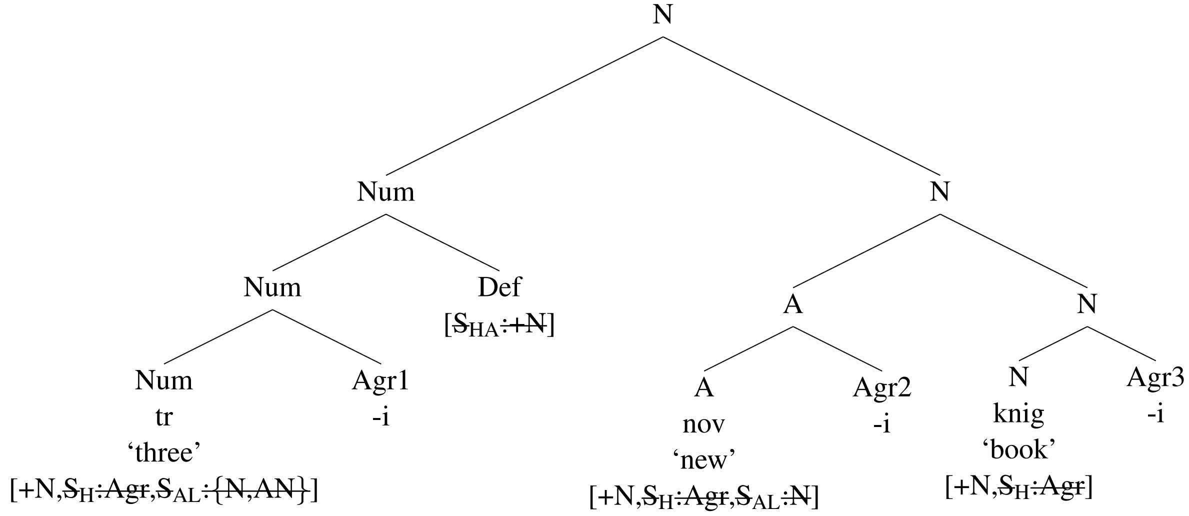


N projects, and this checks off all the [S_{AL}:N] features of the modifiers. N2W now goes on to step 4. N does have an unchecked [S_H:Agr] feature, so the last remaining Agr is moved to the workspace and merged with N:

(69)

Numeration after Seventh Selection						

Workspace:



This checks off the [S_H:Agr] feature of N. N2W returns to step 4, but N has no more unchecked [S_H] features, so it moves on to step 5. There is nothing in the numeration, so it moves to step 6, but N also does not have an [S_C] feature, so it moves to step 7. The numeration is empty, so in step 7, N is removed from the stack. The stack is now empty, so the procedure terminates. All selectional features are checked, and the derivation is well-formed.

As can be seen, the procedure that N2W follows automatically results in the correct placement of the Bulgarian definite article. Absolutely nothing needs to be added to the system. Def will merge when it can, whenever the system gets to step 5 with a [+N] element as the top of the stack. So it will always merge with the *first* [+N] element, in a left-to-right syntax. Once it has merged, it will not be able to merge again, and step 5 will be irrelevant to subsequent [+N] items.

This will also give the correct output when there is nothing in the nominal except a head noun. In this case, Def merges with the head N:

(70) knig-a-ta
book-Agr-Def

In this case, the numeration contains only N, Agr, Def. The only [+N] head is N, so when N2W makes that the top of the stack, Agr and then Def will merge, too, exactly as was gone through in the derivation above.

The correct output is also produced for a nominal that consists of an adjective and a noun:

(71) xubav-a-ta knig-a
nice-Agr-Def book-Agr

In this case, the numeration has two [+N] elements, A and N. A will be merged first, since it has the feature [S_{AL}:N]. When it is at the top of the stack, first an Agr in step 4 and then Def in step 5 will be merged with it. When N is then merged, Def is no longer in the numeration, so only another Agr will merge with the head N.

A case similar to the Num A N case gone through in detail above is Poss A N:

(72) moj-a-ta xubav-a knig-a
my-Agr-Def nice-Agr book-Agr

I assume possessive adjectives are like Num in having at least the disjunctive feature [S_{AL}:{N,AN}]. The numeration then has three [+N] elements. The first one to be selected will be Poss, because it is functional and it has the largest disjunctive selectional feature set. Def will therefore merge with it in step 5, when Poss is the top of the stack, exactly as in the derivation gone through in detail above.

4.3 When the First Element is Not [+N]

Let us now look at the case of an adverb modifying an adjective (53). In this case, Def attaches to the head A, and not to the adverb:

(73) a. * mnog-ət star teatər
very-Def old theater

b. mnogo star-ij-ə teatər-Ø
very old-Agr-Def theater-Agr

This follows in the current account, since adverbs are not [+N] elements. The numeration for (73b) will include the following items:

(74)

NP Numeration: Initial State					
Adv **mnogo** 'very' [S_{AL}:A]	**A** **star** 'old' [+N,S_H:Agr,S_{AL}:N]	**N** **teatər** 'theater' [+N,S_H:Agr]	**Def** [S_{HA}:+N]	**Agr1**	**Agr2**

Once again, I do not show a phonological form for Def and the Agr heads, as their form is determined contextually. The adverb 'very' selects an A as a left adjunct ([S_{AL}:A]).

N2W will proceed by identifying the phase head N and making it the top of the stack. In step 2, there is an element with the feature [S_{AL}:N], namely, the A. So the A is made the top of the stack and N2W returns to step 2. At this point, there is an item with the feature [S_{AL}:A], the adverb *mnogo*. The Adv is made the top of the stack, and N2W returns to step 2. There is nothing with the feature [S_{AL}:Adv], so N2W goes to step 3. The Adv does not have an [S_S] feature in step 3, so the Adv is moved to the workspace:

(75)

NP Numeration after First Selection					
	A **star** 'old' [+N,S_H:Agr,S_{AL}:N]	**N** **teatər** 'theater' [+N,S_H:Agr]	**Def** [S_{HA}:+N]	**Agr1**	**Agr2**

Workspace:
Adv
mnogo
'very'
[S_{AL}:A]

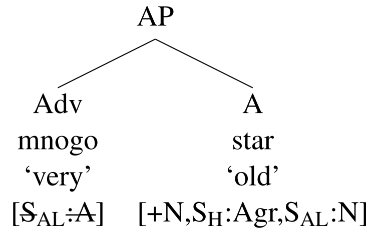
N2W goes on to step 4. The Adv does not have an unchecked [$\mathrm{S_{H}}$] feature in step 4.[3] In step 5, there is nothing in the numeration with an [$\mathrm{S_{HA}}$:Adv] feature. In step 6, the Adv does not have an [$\mathrm{S_{C}}$] feature, so N2W goes to step 7. There is nothing with an [$\mathrm{S_{AR}}$:Adv] feature in the numeration, so the Adv is removed from the stack, making A the top of the stack. N2W returns to step 2. Note that Def has not been merged yet, because the Adv was not [+N].

A is now the top of the stack. In step 2, there is no longer anything in the numeration with an [$\mathrm{S_{AL}}$:A] feature. In step 3, A does not have an [$\mathrm{S_{S}}$] feature, so A is moved to the workspace:

(76)

NP Numeration after Second Selection					
		N **teatər** 'theater' [+N,$\mathrm{S_{H}}$:Agr]	**Def** [$\mathrm{S_{HA}}$:+N]	**Agr1**	**Agr2**

Workspace:


AP
Adv
mnogo
'very'
[~~$\mathrm{S_{AL}}$:A~~]
A
star
'old'
[+N,$\mathrm{S_{H}}$:Agr,$\mathrm{S_{AL}}$:N]

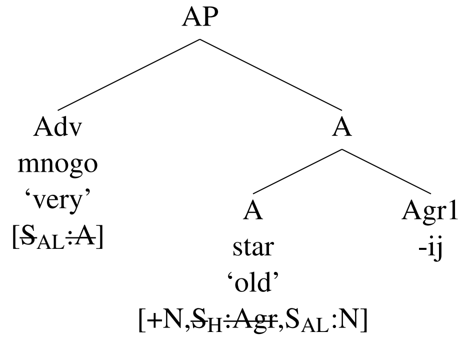
A projects, since the Adv is an adjunct. This checks off the [$\mathrm{S_{AL}}$] feature of the Adv.

N2W goes on to step 4. The A does have an unchecked [$\mathrm{S_{H}}$:Agr] feature, so an Agr is moved to the workspace:

(77)

NP Numeration after Third Selection					
		N **teatər** 'theater' [+N,$\mathrm{S_{H}}$:Agr]	**Def** [$\mathrm{S_{HA}}$:+N]		**Agr2**

Workspace:


AP
Adv
mnogo
'very'
[~~$\mathrm{S_{AL}}$:A~~]
A
A
star
'old'
[+N,~~$\mathrm{S_{H}}$:Agr~~,$\mathrm{S_{AL}}$:N]
Agr1
-ij

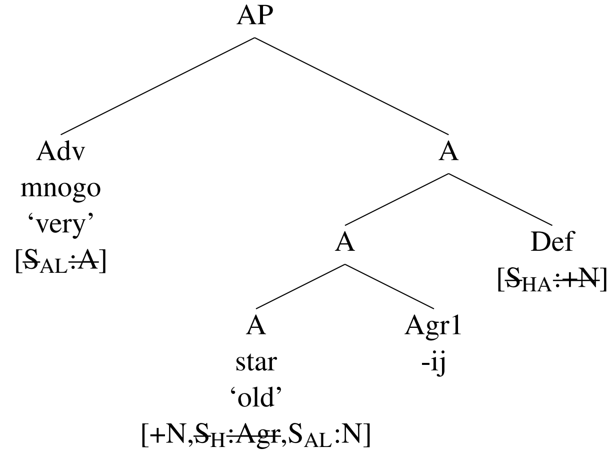
This checks off the [$\mathrm{S_{H}}$:Agr] feature of A. N2W returns to step 4, but now A has no more unchecked [$\mathrm{S_{H}}$] features, so it goes on to step 5. Here there is something in the numeration with an [$\mathrm{S_{HA}}$:+N] feature, namely, Def. Def is moved to the workspace:

[3] If we were to separate out the morpheme *-o* on the Adv, this is where it would merge.

(78)

NP Numeration after Fourth Selection					
		N **teatər** ‘theater’ [+N,S_H:Agr]			**Agr2**

Workspace:



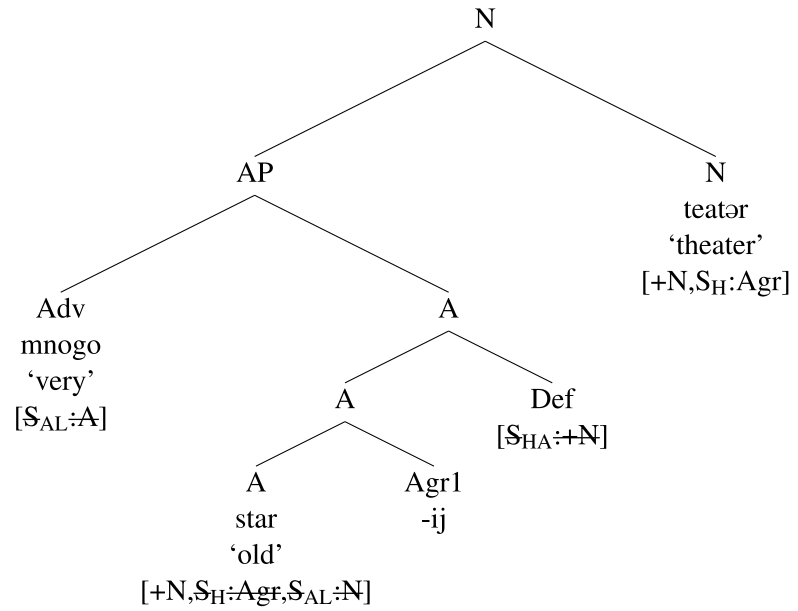
This checks off the selectional feature of Def. N2W returns to step 5, but now the numeration no longer contains anything with an [S_{HA}:+N] feature, so it goes on to step 6. A does not have an [S_C] feature. In step 7, there is nothing with an [S_{AR}:A] feature in the numeration, so A is removed from the stack. This makes N the top of the stack again, and N2W goes back to step 2.

In step 2, there is no longer anything in the numeration with an [S_{AL}:N] feature. In step 3, N does not have an [S_S] feature, so it is moved to the workspace:

(79)

NP Numeration after Fifth Selection					
					Agr2

Workspace:



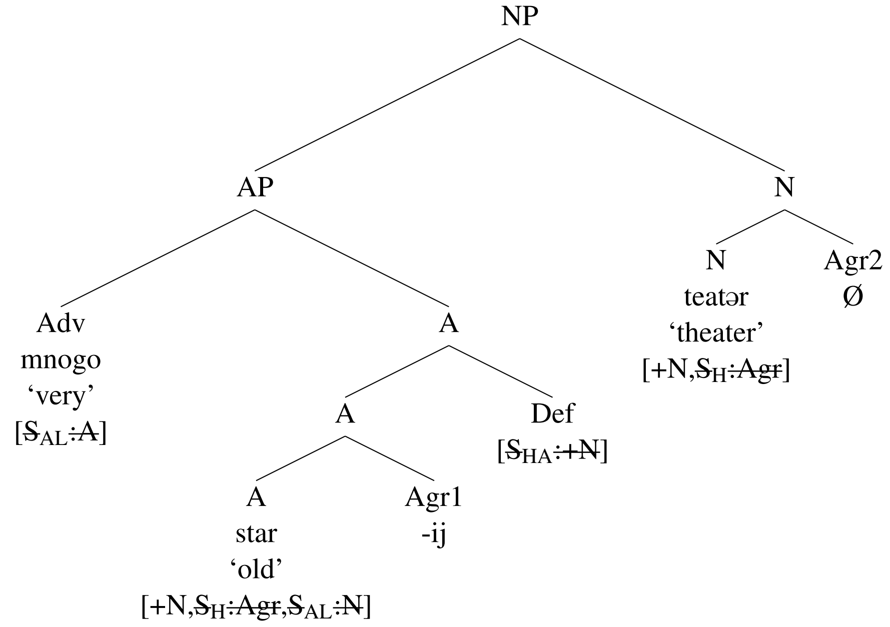
N merges with the topmost projection of A. This checks off the [S_{AL}:N] feature of A.

N2W goes on to step 4. N does have an unchecked [S_H:Agr] head, so the last Agr is moved to the workspace and merged with the N:

(80)

NP Numeration after Sixth Selection					

Workspace:



This checks off the [S_H:Agr] feature of N. N2W returns to step 4, but now N does not have an unchecked [S_H] feature. In step 5, there is nothing in the numeration, and in step 6, N does not have an [S_C] feature. In step 7, there is nothing with an [S_{AR}:N] feature, so N is removed from the stack. The stack is now empty, so the process terminates. All features have been checked, and all items from the numeration have been merged in the workspace. The correct result is achieved, where Def does not attach to the Adv but does attach to the A. Once again, Def is placed correctly by the algorithm that we need for syntax generally.

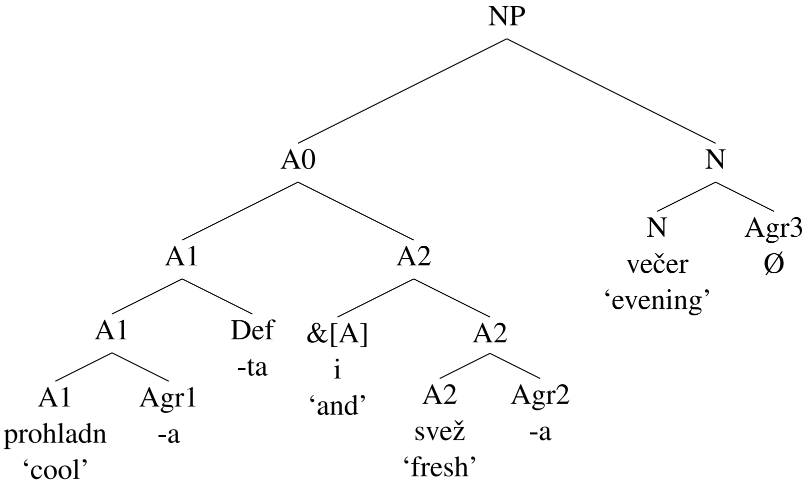
4.4 Coordinated Adjectives

Let us now turn to the case that was problematic for the lowering analysis, coordinated adjectives. As we saw, Def attaches only to the first A:

(81) prohladn-a-**ta** i svež-a večer-Ø
cool-Agr-Def and fresh-Agr evening-Agr
'the cool and fresh evening' (Harizanov & Gribanova 2014: (2d))

As discussed in section 3.4, I assume a non-headed structure for coordination. The structure of the above example will be something like the following:

(82)



Coordinators have a feature that varies over the grammatical categories of the language, and have [S_{AR}] and [S_C] features for the chosen grammatical category. Working from left to right, A1 will be the first [+N] head to be selected, so the current procedure will derive exactly the right result. The derivation begins with a full numeration and an empty workspace:

(83) Initial State

Numeration							
A1 **prohladn** 'cool' [+N,S_H:Agr,S_{AL}:N]	**Def** [S_{HA}:+N]	**A2** **svež** 'fresh' [+N,S_H:Agr,S_{AL}:N]	**N** **večer** 'evening' [+N,S_H:Agr]	**Agr1**	**Agr2**	**Agr3**	**&[A]** **i** 'and' [S_{AR}:A,S_C:A]

Workspace:

—empty—

In the first step, N2W locates the phase head N and makes it the top of the stack. In step 2, there are two elements with [S_{AL}:N] features, namely, the two As. In this case the principle in (7) does not pick one of them out, so N2W chooses one randomly. In this case it chooses ‘cool’. This A is made the top of the stack. Back to step 2, there is nothing with an [S_{AL}:A] feature. In step 3, A does not have an [S_S] feature, so A is moved to the workspace:

(84)

Numeration after First Selection							
	Def [S_{HA}:+N]	**A2** **svež** ‘fresh’ [+N,S_H:Agr,S_{AL}:N]	**N** **večer** ‘evening’ [+N,S_H:Agr]	**Agr1**	**Agr2**	**Agr3**	**&[A]** **i** ‘and’ [S_{AR}:A,S_C:A]

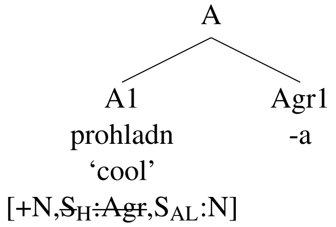
Workspace:
A1
prohladn
‘cool’
[+N,S_H:Agr,S_{AL}:N]

N2W goes on to step 4. A has an [S_H:Agr] feature, so one of the Agr heads is moved to the workspace:

(85)

Numeration after Second Selection							
	Def [S_{HA}:+N]	**A2** **svež** ‘fresh’ [+N,S_H:Agr,S_{AL}:N]	**N** **večer** ‘evening’ [+N,S_H:Agr]		**Agr2**	**Agr3**	**&[A]** **i** ‘and’ [S_{AR}:A,S_C:A]

Workspace:

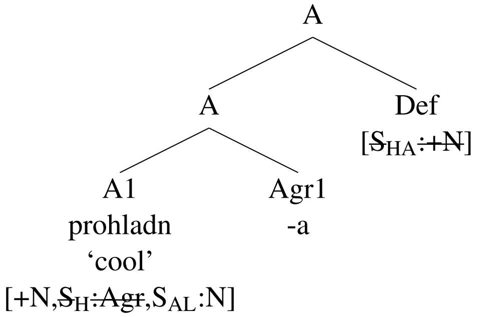


This checks the head feature of A. N2W goes back to step 4, but now A no longer has an unchecked [S_H] feature. In step 5, there is an element with an [S_{HA}:+N] feature, namely, Def. Def is moved to the workspace:

(86)

Numeration after Third Selection							
		A2 **svež** ‘fresh’ [+N,S_H:Agr,S_{AL}:N]	**N** **večer** ‘evening’ [+N,S_H:Agr]		**Agr2**	**Agr3**	**&[A]** **i** ‘and’ [S_{AR}:A,S_C:A]

Workspace:



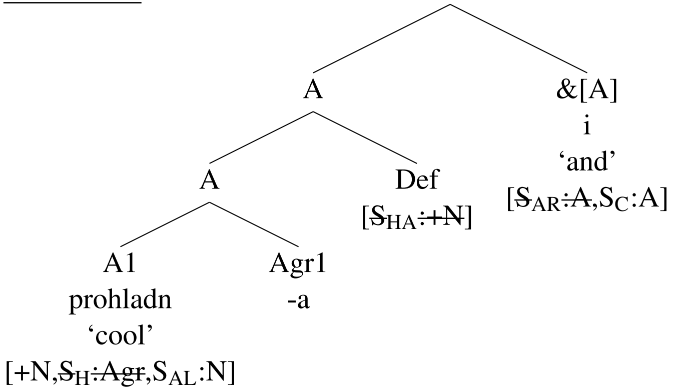
This checks off the [S_{HA}] feature of Def. A projects because Def head-selects as an adjunct.

N2W returns to step 5, but now there is nothing in the numeration with an [S_{HA}:A] or [S_{HA}:+N] feature. In step 7, there is an item with an [S_{AR}:A] feature, namely, &. A is removed from the stack and & is made the top of the stack. Back in step 2, there is nothing with an [S_{AL}:&] feature. In step 3, & does not have an [S_S] feature, so & is moved to the workspace. Agr and Def merged as heads, meaning that Def has to merge with the entire complex head A:

(87)

Numeration after Fourth Selection							
		A2 **svež** ‘fresh’ [+N,S_H:Agr,S_{AL}:N]	**N** **večer** ‘evening’ [+N,S_H:Agr]		**Agr2**	**Agr3**	

Workspace:


A
&[A]
i
‘and’
[~~S~~$_{AR}$~~:A~~,S$_{C}$:A]
A
Def
[~~S~~$_{HA}$~~:+N~~]
A1
prohladn
‘cool’
[+N,~~S~~$_{H}$~~:Agr~~,S$_{AL}$:N]
Agr1
-a

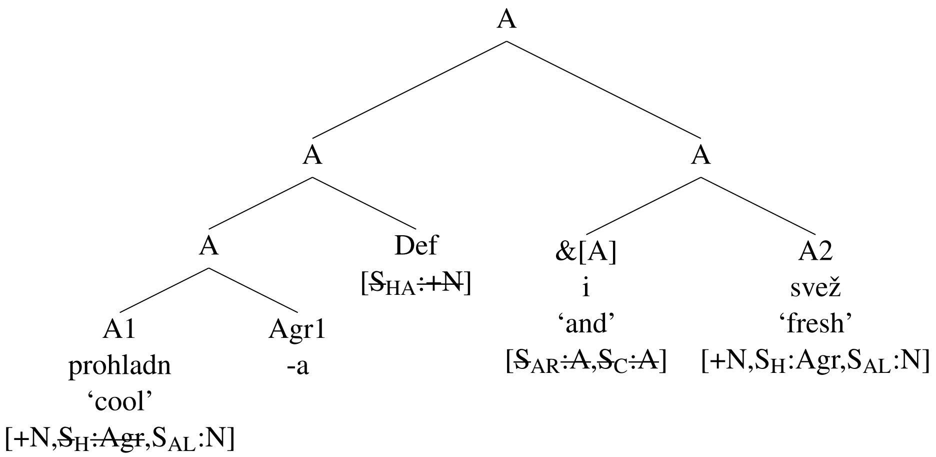
This checks off the [S_{AR}:A] feature of &. In steps 4 and 5, N2W finds nothing, but in step 6, & does have an [S_C:A] feature and there is an A in the numeration. So & is removed from the stack and A2 is made the top of the stack.

Back in step 2, there is nothing with an [S_{AL}:A] feature. In step 3, A2 does not have an [S_S] feature, so A2 is moved to the workspace and merged with &:

(88)

Numeration after Fifth Selection							
			N **večer** ‘evening’ [+N,S_H:Agr]		**Agr2**	**Agr3**	

Workspace:


A
A
A
A
Def
[~~S_{HA}:+N~~]
&[A]
i
'and'
[~~S_{AR}:A~~,~~S_C:A~~]
A2
svež
'fresh'
[+N,S_H:Agr,S_{AL}:N]
A1
prohladn
'cool'
[+N,~~S_H:Agr~~,S_{AL}:N]
Agr1
-a

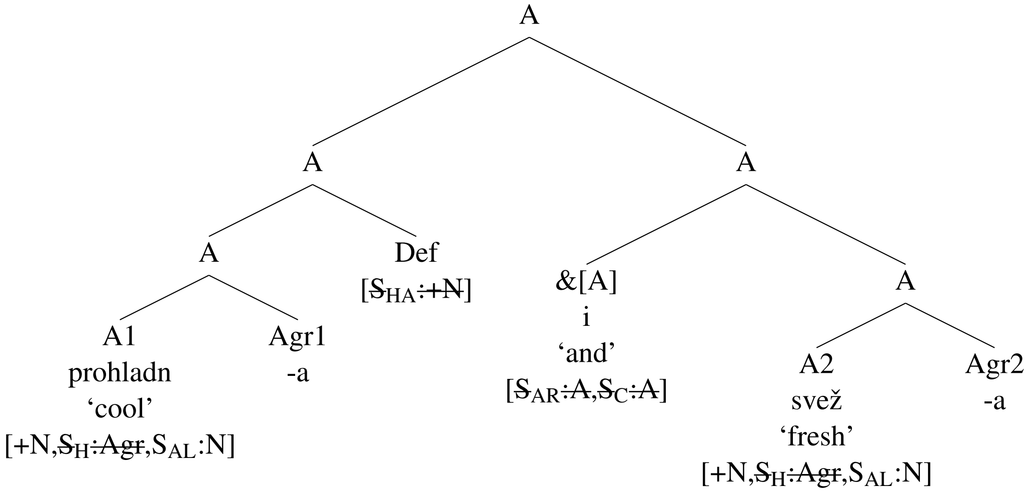
This checks the [S_C:A] feature of &. A projects, as described in section 3.4.

N2W continues to step 4. A2 has an [S_H:Agr] feature, so one of the Agr heads is moved to the workspace and merged with the head A2:

(89)

Numeration after Sixth Selection							
			N **večer** 'evening' [+N,S_H:Agr]			**Agr3**	

Workspace:


A
A
A
A
Def
[~~S_{HA}:+N~~]
&[A]
i
'and'
[~~S_{AR}:A~~,~~S_C:A~~]
A
A1
prohladn
'cool'
[+N,~~S_H:Agr~~,S_{AL}:N]
Agr1
-a
A2
svež
'fresh'
[+N,~~S_H:Agr~~,S_{AL}:N]
Agr2
-a

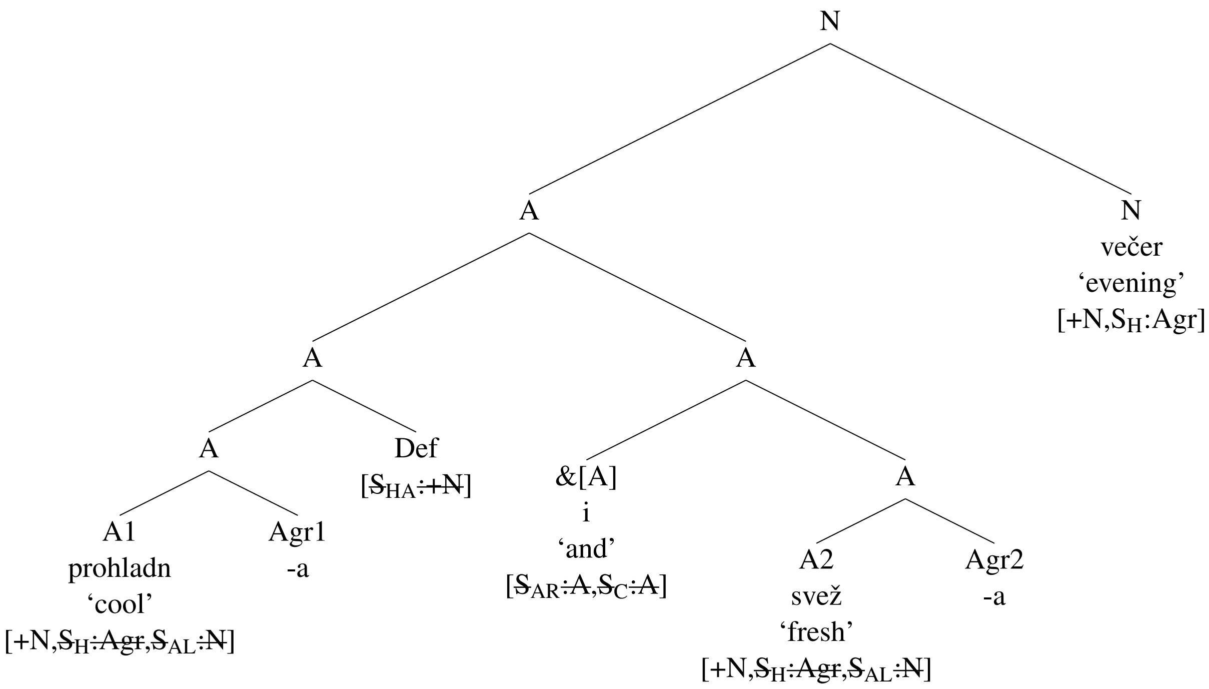
This checks off the [S_H:Agr] feature of A2, so when N2W returns to step 4 it comes up empty. It also does not find anything in steps 5, 6, or 7, so A2 is removed from the stack, making N the top of the stack again. N2W returns to step 2.

In step 2, there is nothing in the numeration with an [S_{AL}:N] feature. In step 3, N does not have an [S_S] feature, so N is moved to the workspace. It has to merge with the entire tree created thus far:

(90)

Numeration after Seventh Selection							
						Agr3	

Workspace:



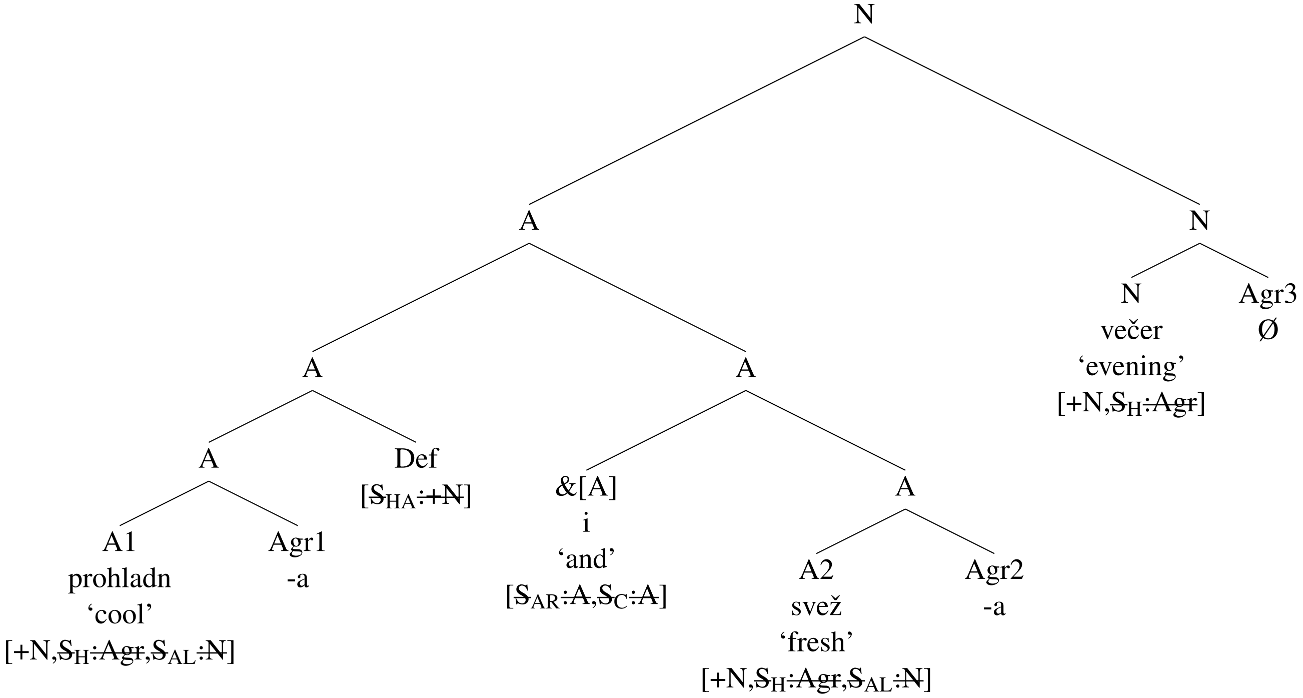
This checks off all the [S_{AL}:N] features of the As. N projects.

N2W goes on to step 4. N has an [S_{H}:Agr] head, so the last Agr is moved to the workspace and merged with it:

(91)

Numeration after Eighth Selection							

Workspace:



This checks off the [S_H:Agr] feature of N, so when N2W returns to step 4, it comes up empty. It also finds nothing in steps 5, 6, and 7, so N is removed from the stack. The stack is now empty, so N2W terminates. All features are checked, and the numeration is empty. The derivation has resulted in a well-formed phrase. Importantly, Def was placed correctly on the first adjective of the coordinated adjectives.

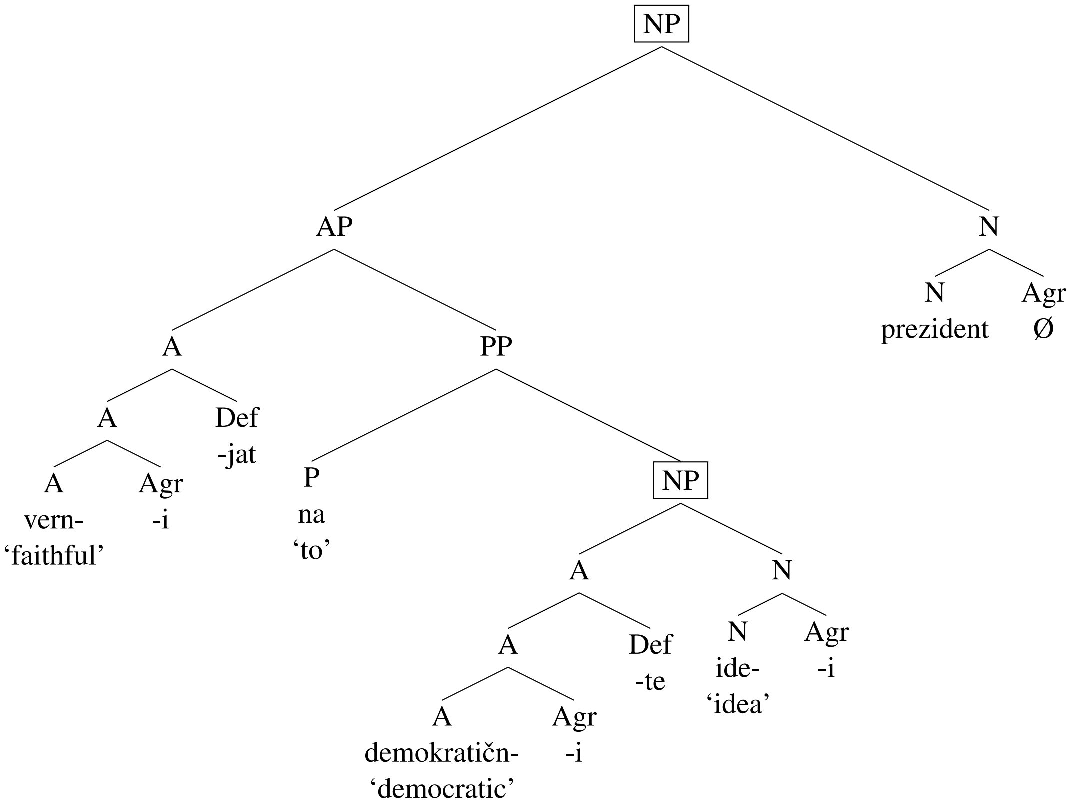
4.5 One NP Phase Inside Another

It should also be noted that one NP can embed another. The following is such an example in Bulgarian:

(92) verni-jat na demokratični-te idei prezident
faithful-Def to democratic-Def ideas president
'the president (who is) faithful to democratic ideas' (Franks 2001: 55, (6a))

In this example, 'the president' embeds another definite NP, 'the democratic ideas'. This has the following structure, with the two NP phase nodes in boxes:

(93)



Since each NP is its own phase, there are two different phases here, with two different numerations. Each numeration has its own Def. I still assume that the tree is built left to right, however. This means that the higher phase will have to be put on hold while the lower phase is built. As discussed in section 3.1, I propose that the higher phase, the one that is begun first, can include a placeholder of category N (Ph:N), to satisfy selectional restrictions, which will then be replaced by a full phase with its own numeration. So, the numeration for the higher phase will be the following:

(94)

Numeration for the Higher Phase						
A	**Def**	**P**	**N**	**Agr1**	**Agr2**	**Ph:N**
vern		**na**	**prezident**			
‘faithful’		‘to’	‘president’			
[+N,S_H:Agr,S_{AL}:N,S_C:P]	[S_{HA}:+N]	[S_C:N]	[+N,S_H:Agr]			

Workspace:

—*empty*—

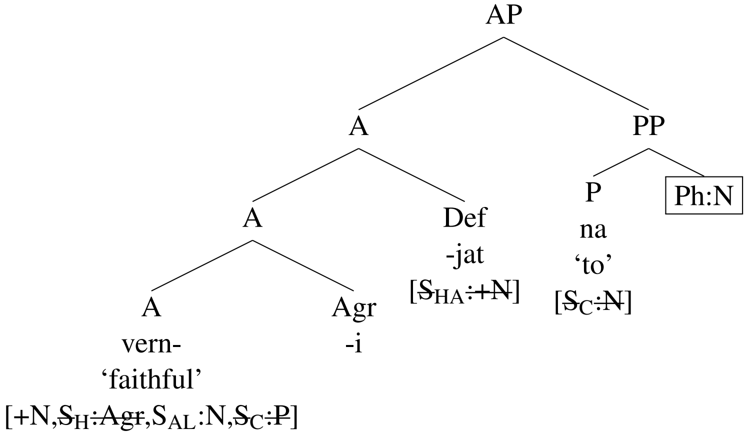
Note that the A meaning ‘faithful’ selects a PP complement.

N2W begins by making the N *prezident* the top of the stack (I assume that a Ph:N placeholder cannot be the head of the current phase). In step 2, there is an item with the feature [S_{AL}:N], namely, the A. The A is made the top of the stack. There is nothing with an [S_{AL}:A] feature in step 2, and in step 3, A does not have an [S_S] feature, so A is moved to the workspace. In step 4, A has an [S_H:Agr] feature, so an Agr is moved next. In step 5, Def has an [S_{HA}:+N] feature, so it is moved next. Agr and Def both merge with A to create a complex head. Next, in step 6, A has an [S_C:P] feature. There is a P in the numeration, so A is removed from the stack and P is made the top of the stack. Steps 2 and 3 come up empty, so P is moved to the workspace. It merges with the entire complex A head. Steps 4 and 5 then find nothing, but in step 6, P has an [S_C:N] feature, and there is an N (Ph:N) in the numeration. P is removed from the stack, and Ph:N is made the top of the stack. Nothing will happen in the new step 2, but in step 3, Ph:N will be moved to the workspace. The following shows the outcome of the above steps:

(95)

Numeration for the Higher Phase after Moving Ph:N						
			N **prezident** 'president' [+N,S_H:Agr]		**Agr2**	

Workspace:



Merging a PP with a projection of A checks off the [S_C:P] feature of A. Merging the placeholder of category N with the P checks off P's [S_C:N] feature. Merger of Agr and Def with A satisfied the head-selection features of A and Def.

At this point, what happens is that the higher numeration is left as-is and another numeration will be selected. How a numeration is selected from the lexicon is the topic of a companion paper. Here, I assume that a numeration is selected correctly. The numeration for the lower phase will consist of the following items:

(96)

Numeration for the Lower Phase				
A **demokratičn** 'democratic' [+N,S_H:Agr,S_{AL}:N]	**Def** [S_{HA}:+N]	**N** **ide** 'idea' [+N,S_H:Agr]	**Agr1**	**Agr2**

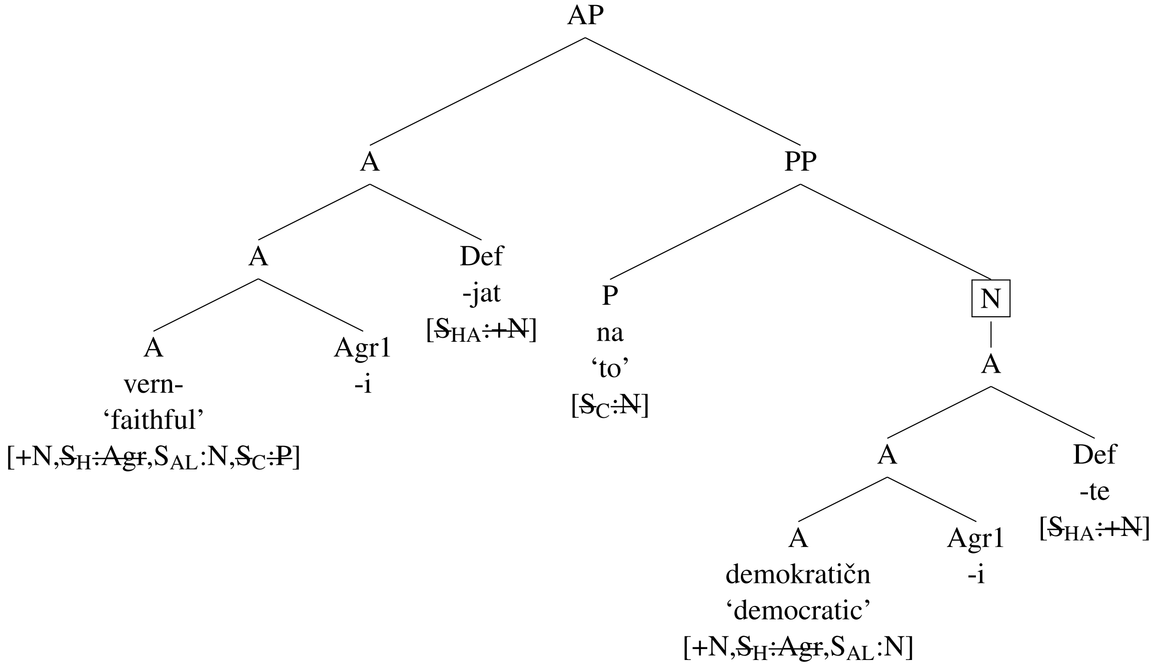
The syntax now begins working on the lower phase. There is only one workspace, so the elements from the lower phase are merged into the tree currently in the workspace. This begins with N2W making the N *ide* the top of the stack. In step 2, there is an item with the feature [S_{AL}:N], namely, the A. The A is made the top of the stack. There is nothing with an [S_{AL}:A] feature in step 2, and in step 3, A does not have an [S_S] feature, so A is moved to the workspace. In step 4, A has an [S_H:Agr] feature, so an Agr is moved next. In step 5, Def has an [S_{HA}:+N] feature, so it is moved next. Agr and Def both merge with A to create a complex head. The following illustrates the outcome of the above steps:

(97)

Numeration for the Higher Phase (on hold)						
			N **prezident** ‘president’ [+N,S_H:Agr]		**Agr2**	

Numeration for the Lower Phase (active)				
		N **ide** ‘idea’ [+N,S_H:Agr]		**Agr2**

Workspace:



Merging Agr1 and Def in the lower phase checks the head-selectional features of A and Def.

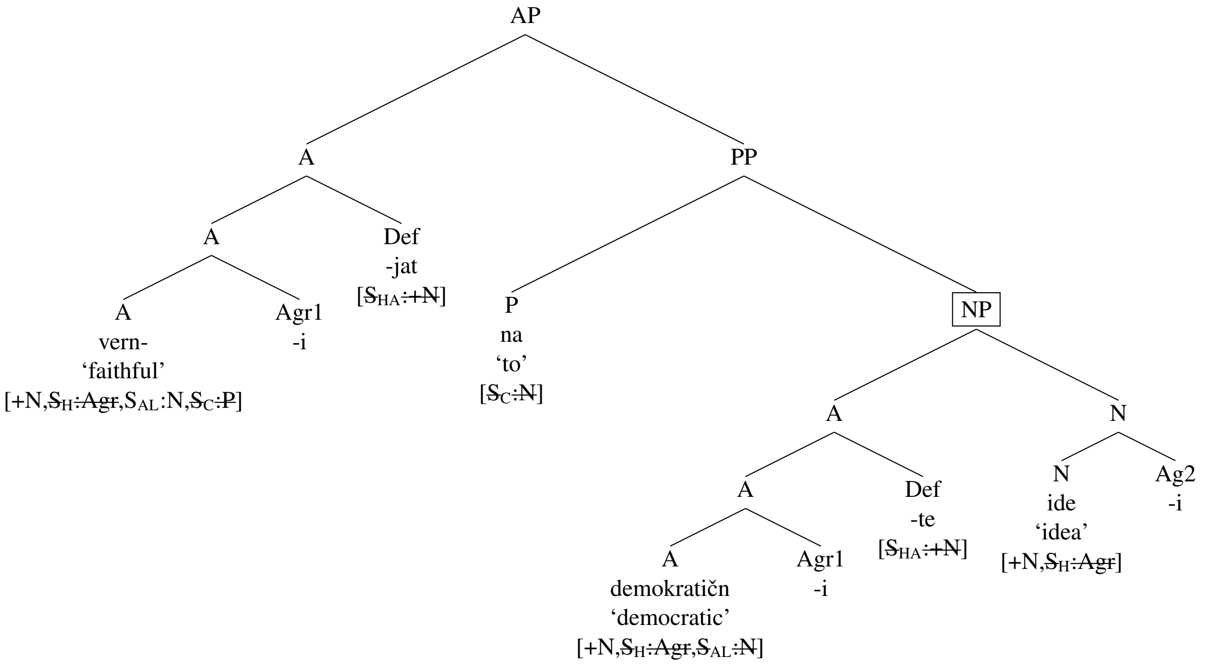
N2W goes on to step 6, but this A does not have an [S_C] feature. In step 7, there is nothing with an [S_{AR}] feature in the current numeration. A is removed from the stack, making N the top of the stack. In step 2, there is nothing left in the numeration with an [S_{AL}:N] feature, and in step 3, N does not have an [S_S] feature, so N is moved to the workspace and merged with the complex head A. In step 4, N has an [S_H:Agr] feature, so Agr2 is merged:

(98)

Numeration for the Higher Phase (on hold)						
			N **prezident** ‘president’ [+N,S_H:Agr]		**Agr2**	

Numeration for the Lower Phase (active)				

Workspace:



This checks the [S_H:Agr] feature of N, and the [S_{AL}:N] feature of A. The numeration is now empty, and N2W will run through steps 5, 6, and 7, where it will terminate. With the lower phase complete, the higher phase becomes active again.

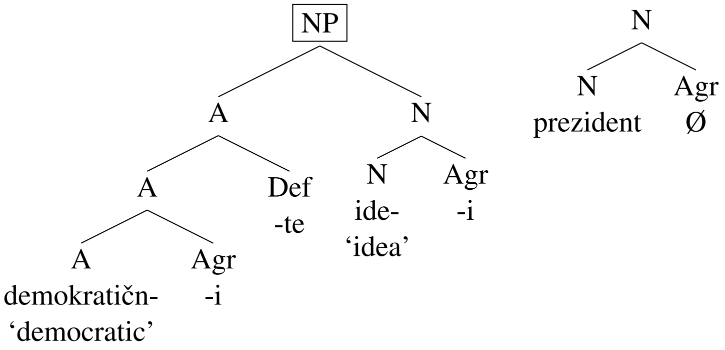
The higher phase had just completed step 3 with Ph:N as the top of the stack. It will run through steps 5, 6, and 7 with Ph:N and come up empty, so Ph:N will be removed from the stack. This will make the N *prezident* the top of the stack again. Steps 2 and 3 come up empty, so the N will be merged into the tree, as a sister of the topmost AP (because that is a left adjunct of N). In step 4, N has an [S_H:Agr] feature, so the final Agr head will be merged with N. N2W will then run through steps 5, 6, and 7, and terminate when N is removed from the stack. The result will be the tree in (93).

As can be seen, the proposed analysis gets the facts exactly right. It puts Def on the first [+N] element of each nominal phase, as it should. It also correctly captures the phase-based nature of the derivation.

At this point it is also worth pointing out that the derivation would not have worked if we had run the derivation bottom-up instead of left-to-right. The first problem is that a bottom-up derivation will not automatically locate Def on the *first* element of the right type. In the current analysis, Def will be merged when it can; this will correctly put it on the first element of the right type, and not any subsequent ones, because it will have been moved out of the numeration at the point where those become active. In a bottom-up derivation, the procedure would somehow have to put Def on the *last* item of the correct type. It is possible to specify such a procedure. One could have an algorithm like the following: (1) each time a [+N] element is merged, scan the numeration; (2) if there is a [+N] element in the numeration, do nothing; (3) if there is no [+N] element in the numeration, move Def to the workspace. This will put Def on the last [+N] element moved from the numeration to the workspace.

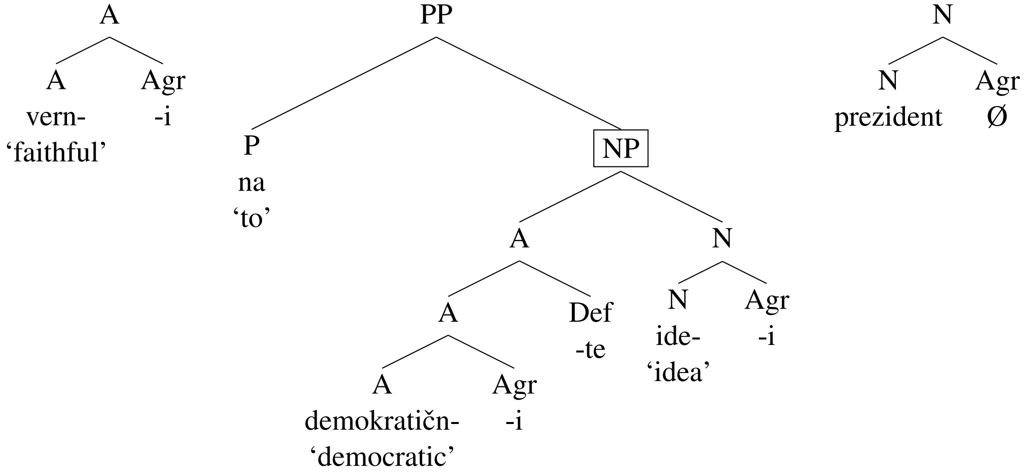
The problem comes from the embedded phase in the example above. In a bottom-up derivation for the structure in (93), the N 'president' and its accompanying Agr node would be merged first. The lower NP phase would have to be built separately, in a separate derivation. Suppose it could be, and then put into the numeration for the higher phase as a single item. It then has to be put into the workspace with 'president', but not yet connected to it. This is two unconnected items:

(99) Workspace:



The P is then merged with the lower NP phase. But now A needs to be merged. The problem is that it cannot be merged with either of the two unconnected items in the workspace. It has to remain on its own until Agr and Def merge with it, then it can be merged with the PP. So there is a stage at which there are three unconnected items in the workspace (this diagram shows the higher Def not having merged yet with 'faithful'):

(100) Workspace:



If this is allowed, however, then there is no way to ensure that the derivations work correctly. Take the case of coordinated adjectives. We could put them into the workspace in the wrong order, because the principle in (7) does not determine which to pick first, but then we could leave them unconnected. The bottom-up Def Algorithm would put Def on whichever A was merged last. But then that one could be merged on the right in the coordinate structure, incorrectly (since coordinates can typically be reversed). This would put Def on the *second* of two coordinated As, which is ungrammatical.

If we allow three or more unconnected items in the workspace, as is necessary for the NP embedded inside an NP in (93) on a bottom-up derivation, then it is impossible to properly constrain N2W and ensure that everything merges in the right order. I conclude that only a left-to-right derivation will work, as only that can be constrained in the way that is necessary. Only a left-to-right derivation will not require leaving structures unconnected in the workspace. Note that in all of the left-to-right derivations gone through here, everything that is moved into the workspace is merged immediately with the existing tree. There is never a point where two things need to remain unconnected.

I conclude that only a left-to-right derivation can work for syntax. We need to reject bottom-up approaches, and adopt a a left-to-right derivation as in the current model.

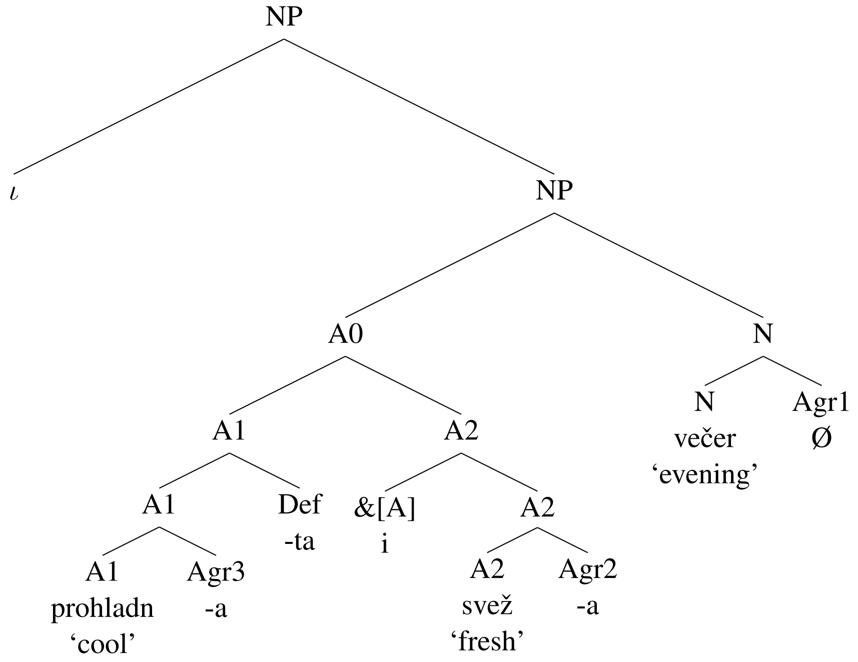
4.6 Semantic Interpretation

One thing to note about the proposed analysis is that it is not compatible with the DP Hypothesis, if Def is identified as the head D. Def cannot project in this proposal, it can only merge with a [+N] element. It is a dependent of the head N in the sense that it is part of the nominal numeration; but in the syntax it may end up as the daughter (and sister) of an A or a Num or Poss, depending on what it merges with. This may appear to pose a problem for compositional semantics. If the head Def is the definite article (an iota operator semantically), then it should combine only after all modifiers have combined with the N. Yet in the proposed structure, Def is often the sister of a modifier. I can see three ways to address the semantic effect Def has on the NP whose numeration it is part of. The first two I will not adopt.

The first alternative is to say that Def is present in an NP numeration when the head N is [+Def]. That is, it is a feature of the head noun that it is definite. A constraint then requires that when the numeration includes a [+Def] N, L2N must move a Def from the lexicon to the numeration. Then it is the feature [+Def] on N, and by extension the whole NP, that is interpreted. For instance, it could induce a semantic typeshift.

The second alternative is to say that a definite NP actually includes an unpronounced iota operator. This null iota operator would adjoin to the highest node in the NP, as shown below for the case of coordinated adjectives:

(101)



The head Def in this alternative would be semantically contentless. Its role would be to mark the presence of the iota operator. We could again analyze this dependency with a constraint on L2N: If a null iota operator is moved from the lexicon to the numeration, then a Def must be too.[4]

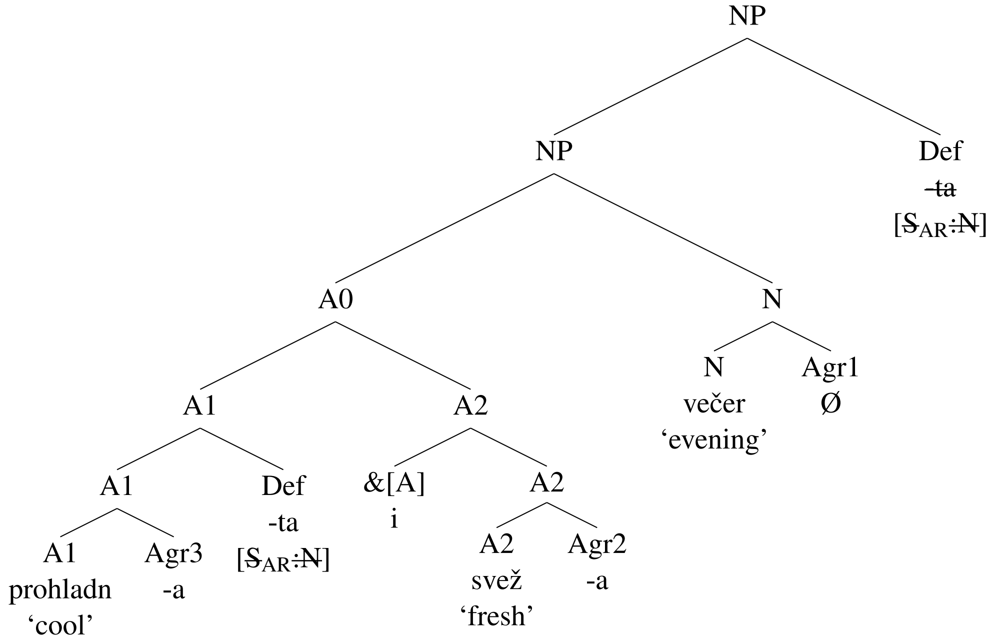
The third alternative is the one that I will adopt. In this alternative, Def is indeed the semantic operator (an iota operator). It needs to take scope over the rest of the NP. I propose that, like the adverb *how* in English (section 3.1), it has an [S_{AR}] feature. This feature selects for NP: [S_{AR}:N]. This feature also does not get checked off in the course of the derivation. At the end of the derivation, this will then trigger the following rule from section 3.1:

[4] It should be noted that this second alternative is compatible with the DP Hypothesis. I have shown the iota operator as a daughter of NP in the structure in (101), but one could also analyze it as a null D heading a DP if one wished. (I do not do so, as the DP Hypothesis is extremely suspect and should be rejected anyway; see Bruening 2009, Bruening et al. 2018, Bruening 2020.)

(102) After N2W terminates, if there is an X in the workspace with an unchecked [S_{A}:Y] feature, copy X and merge it with a projection of Y.

This will have the following result, again illustrating with the case of coordinated adjectives:

(103)



Merging Def with the highest NP checks off the [S_{AR}:N] feature on both copies, as usual. As in the system in general, the leftmost copy of a movement chain is the pronounced one. In the semantics, the pronounced copy, adjoined to A1, is semantically vacuous. Only the higher copy, adjoined to NP, is interpreted (as an iota operator).

This movement does move Def out of a coordinate structure. However, this is allowed precisely because the lower copy is not interpreted. Some recent work has argued that the coordinate structure constraint is not a constraint on movement, but a parallelism constraint on binding (Muadz 2001, Ruys 1992, Fox 2000, Johnson 2009). In the semantics, the lower Def is essentially not there, and so this constraint is not violated.

This third alternative eschews null elements, and makes Def the semantically contentful item, which I view as desirable. It also does not need any new rules or procedures, but follows from a rule we need anyway (the one in 102). I therefore adopt it, and in all of the representations above, Def should be amended to have an [S_{AR}:N] feature that is satisfied by moving rightward at the end of the derivation. (Note that having this feature will not affect any of the derivations gone through above: the [S_{HA}:+N] feature of Def will cause it to be moved to the workspace before the [S_{AR}:N] feature is ever considered.)

4.7 Complications in Bulgarian

As can be seen, the proposed analysis is very successful at capturing the placement of the Bulgarian definite marker. There are of course complications, however. The first complication is that Def does not co-occur with demonstratives (in the standard language):

(104) tazi kniga
this book
(Embick & Noyer 2001: 568, (21))

The most obvious approach to this complementarity is to say that Dem and Def are the same category, and only one element of that category is allowed per NP. The demonstrative instance of this category merges according to a principle different from that of the Def instance (according to the hierarchy Dem > Num > A > N, where the hierarchy is instantiated through size of disjunctive sets that are selected).

The second complication is that the Def head is not the only item in the language that has the distribution that it does. Bulgarian has two forms of possessive pronouns, one shown above (e.g., 72), which Def attaches to if it is the first [+N] element in the NP; but it also has clitic forms of possessive pronouns. These obey the same distribution as Def: they occur after the first [+N] element in the NP, immediately following Def:

(105) (Franks 2001: 59, (23e–f))

a. mnogo-to **ti** novi knigi
many-the your new books
'your many new books'

b. večno mlada-ta **ni** stolica
perpetually young-Def our capital
'our perpetually young capital'

One issue that has been discussed heavily in the literature on this topic in Bulgarian is that Def has all the properties of a canonical affix, like idiosyncratic phonology and arbitrary gaps, but the clitic pronouns do not, they behave like clitics (see, e.g., Franks 2001, Embick & Noyer 2001). The issue is how they can obey the same distribution if one is an affix and the other is not. Embick & Noyer (2001) correctly point out that this is not a problem if the same mechanism puts them in their position. They simply have to be lexically specified as having different prosodic and morphophonological properties. In Embick and Noyer's lowering analysis, Poss first lowers onto the D head, and then D lowers onto its complement. As we saw above, a lowering analysis incorrectly predicts across-the-board lowering in coordination contexts.

In the current analysis, we can give exactly the same analysis for the clitic pronouns as for Def. The clitic pronouns have the same feature that Def does, namely, [S$_{\text{HA}}$:+N]. When a [+N] item is the top of the stack, first an Agr will be merged with it in step 4, then step 5 will locate both Def and the possessive clitic. N2W will need to choose one to start with, and in Bulgarian it has to be Def first. Note that we also have to give Def the properties of an affix, as just discussed, while the possessive clitic is instead a prosodically deficient pronoun. I suggest that this provides the basis for choosing: in addition to the preferences in (7), N2W will also choose something specified as an affix over something that is not. So N2W will choose Def first and move it to the workspace, and then return to step 5 and locate the possessive clitic. The possessive clitic will then be moved to the workspace, and all of Agr, Def, and the clitic will form a complex head with the [+N] element, in that order.

4.8 Summary

The proposed analysis captures the position of the definite marker in Bulgarian without the need for post-syntactic lowering mechanisms or linear displacement rules. It uses only mechanisms that we need for the syntax anyway: the numeration, the workspace, merge, and a procedure that selects items from the numeration and moves them to the workspace (N2W). Spelling out how this procedure works in a left-to-right derivation has the result that the Bulgarian definite marker is placed in the correct location, with no stipulations or additional constraints. In fact the Bulgarian definite marker behaves exactly like English affixal negation: both go on the *first* element of the appropriate type in the phase they are part of. The placement of the definite marker in Bulgarian is therefore not unusual at all, and does not require extra-syntactic operations.

5 The Amharic Definite Marker

The definite marker in Amharic (Kramer 2010) has a distribution very similar to that of Bulgarian, although it differs in some important respects. Most importantly, Amharic is a head-final language for most categories (except apparently P), which results in some of the differences. Otherwise, we can give a very similar analysis for Amharic as for Bulgarian. It is necessary to alter the procedure that N2W follows for head final languages, but in the interests of space I will only say that what has to happen is that step 6, the one that refers to complement selection, will have to precede steps 4 and 5 (head selection). The head will also have to be merged in that step (the new step 4) rather than in step 3. Otherwise the procedure remains the same. It is possible to account for the Amharic facts without needing to go into more detail than this.

As described by Kramer (2010), the definite suffix in Amharic always follows the first full phrase inside the NP. If there is no phrase other than the head noun, it follows the head noun (106a), but if there is a phrase like an AP, it follows the whole AP (106b–d):[5]

(106) (Kramer 2010: (1a), (3a), (32), (6))

a. bet-**u**
house-Def
'the house'

b. tɨllɨk'-**u** bet
big-Def house
'the big house'

c. bät'am tɨllɨk'-**u** bet
very big-Def house
'the very big house'

d. lä-mist-u tammaññ-**u** gäs'ä bahriy
to-wife-his faithful-Def character
'the faithful-to-his-wife character'

As in Bulgarian, adjectives agree with the head noun. Kramer (2010) states that adjectives optionally agree in number with indefinite nouns:

(107) tɨgu(-woʧʧ) tämari-woʧʧ
diligent(-PL) student-PL
'diligent students' (Kramer 2010: 228, (66a))

The first adjective in a sequence obligatorily agrees in case, while subsequent adjectives optionally do (and also in definiteness):

(108) tɨllɨk'-u-n t'ɨk'ur(-u-n) bet
big-Def-Acc black(-Def-Acc) house
'the big black house (Acc)' (Kramer 2010: 228, (66b))

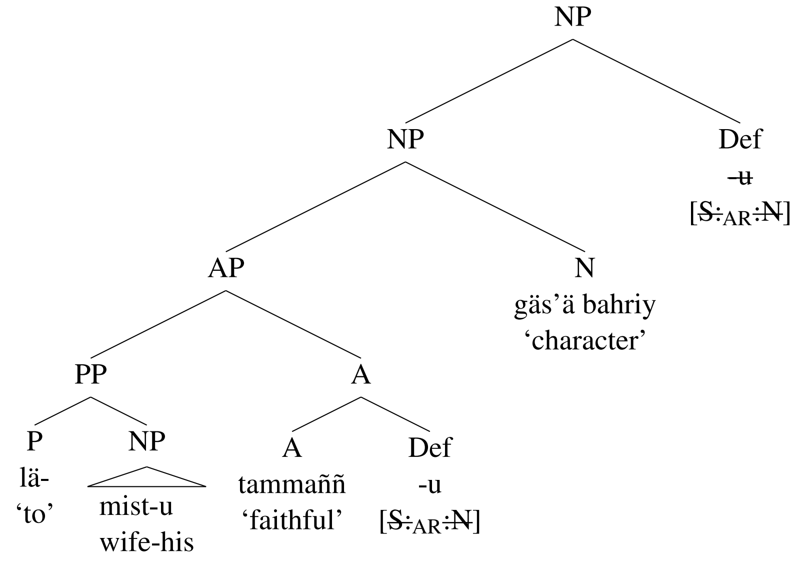
Note that, as in Bulgarian, the definite marker goes on the *first* AP if there is more than one. (Following Kramer, the optional appearance on subsequent adjectives is optional concord; this is then an Agr head and not a Def head. There is only one Def head per nominal numeration.)

[5] All Amharic examples from Kramer (2010). Abbreviations that have not already been introduced: C = complementizer, IMPF= imperfective aspect, M = masculine, PF= perfective aspect.

As in Bulgarian, we can take this to indicate that certain heads in the nominal share a feature, call it [+N] again, and this feature defines the class of elements that can show concord agreement. Head nouns and head adjectives are [+N], while adverbs and prepositions are not. We can then give the Amharic Def essentially the same analysis as in Bulgarian: it has an [S_{HA}:+N] feature. This will put Def on the *first* [+N] head merged into the derivation, since Def merges when it can, and once Def is merged, it is no longer present in the numeration and so cannot be selected when subsequent [+N] heads are merged. In (106a), the only [+N] head is the head noun itself; in step 5, N2W will place Def on the head noun (which was moved to the workspace in head-final step 4). In (106b), there are two [+N] heads, the A and the N. The derivation will start with N as the top of the stack, but then A has an [S_{AL}:N] feature, so it will replace N on the top of the stack and will be moved to the workspace first. Def will then merge with the A in step 5. In (106c), there are again two [+N] heads, an A and an N. Again Def will be placed on the A. The adverb will be ignored, since it is not [+N]. This is exactly like the derivation gone through in detail for Bulgarian. The same is true for the preposition in (106d). It is not [+N]. Its complement includes an N, which is [+N], but this complement is its own phase, with its own numeration; Def is not included in this numeration and so will not be merged when the complement of P is constructed. The head adjective 'faithful' is therefore the first [+N] head merged in the same phase that includes Def in its numeration.

As in Bulgarian, we also need Def to move high and to the right for semantic interpretation (as an iota operator). We can provide the same analysis as we did for Bulgarian: Def has an [S_{AR}:N] feature that requires it to move and adjoin to NP on the right. Only the lower copy is pronounced, but only the higher copy is semantically interpreted. I show the resulting structure for example (106d) below:

(109)



There are two cases in Amharic where things become slightly more complicated. These are relative clauses, and complex numerals. Two relative clauses are shown below. They uniformly have Def attached after the relative clause, apparently to the highest verb (recall that Amharic is head-final and therefore clauses are verb-final):

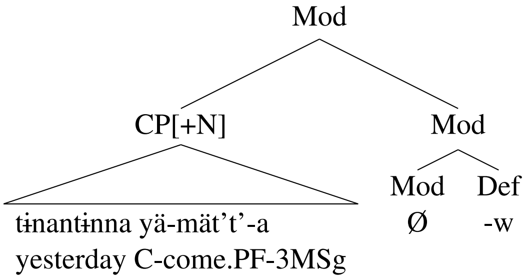
(110) a. tɨnantɨnna yä-mät't'-a-**w** tämari
yesterday C-come.PF-3MSg-Def student
'the student who came yesterday' (Kramer 2010: 199, (8a))

b. lɨʤ-oʧʧ-u-n bähayl yɨ-gärf yä-näbbärä-**w** astämari
child-PL-Def-Acc severely 3MSg-beat.IMPF C-be.Aux-Def teacher
'the teacher who used to beat the children severely' (Kramer 2010: 200, (9))

Note that the complementizer appears to be a prefix on the highest verb inside the relative clause (glossed "C"). I propose that the highest verb undergoes head movement to C, and adjoins on the right of C.

I further propose that relative clauses are the complement of a null head that we can call Mod. CPs are phases, and have their own numeration. In the NP phase, when Mod is selected for a numeration, a placeholder Ph:C is also selected. Mod is an adnominal modifier, like an A, and like an A it is both [+N] and has the selectional feature [S_{AL}:N]. In the two examples in (110), the CP is the first thing inside the NP. This means that N2W will start with the phase head N as the top of the stack, but then it will replace it with Mod, since Mod has the feature [S_{AL}:N]. When the step for complement selection is reached (new step 4 for head-final languages), the Ph:C placeholder will be merged. The NP phase will then be put on hold while the CP is built. Once it is built, the head Mod will be merged. At that point, Def has an [S_{HA}:+N] feature, so it will be moved to the workspace and merged with Mod:

(111)



The nominal derivation will then continue. In (110a), the only thing left is for the head N to be merged.

The second case where things are more complicated is complex numerals. Def always attaches at the end of a complex numeral:

(112) (Kramer 2010: 224, (57a), (59))

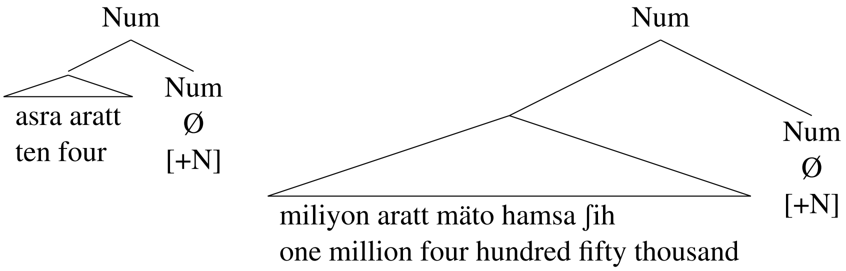
a. asra aratt-**u** tämari-wotʃtʃ
ten four-Def student-PL
'the fourteen students'

b. and miliyon aratt mäto hamsa ʃih-otʃtʃ-**u** wättaddär-otʃtʃ
one million four hundred fifty thousand-Pl-Def soldier-Pl
'1,450,000 soldiers'

In (112a), we can see that the numeral 'four' can host Def, and therefore ought to be a [+N] head. However, in (112b), this head is ignored and Def instead goes on 'thousand'.

I propose that all complex numerals in Amharic involve a null head that takes the overt numerals as complement. The internal structure of the sequence of overt numerals is not important:

(113)


Num
asra aratt
ten four
Num
Ø
[+N]
Num
miliyon aratt mäto hamsa ʃih
one million four hundred fifty thousand
Num
Ø
[+N]

The pronounced numerals are then not [+N], only the null Num head is. This makes Num the first [+N] head merged into the derivation in both examples in (112). Def will then be merged with Num (outside of the plural marker if present), correctly placing it at the end of the sequence of numerals.

This captures the distribution of the definite marker in Amharic, as described by Kramer (2010). There are of course a few complications, but Kramer does not give enough data to decide on an analysis of these. One has already been mentioned: when there is more than one AP, Def is obligatory on the first but optional on subsequent ones. As also mentioned above, we can follow Kramer and analyze markers others than the first as optional nominal concord. They are then Agr heads and not Def heads; any given nominal numeration can have only one Def head. The second complication is that stacked relative clauses have to have Def after each relative clause, not just the first. One possible approach to this is to say that each relative clause is an appositive NP, with a null N head. As Kramer (2010) shows, relative clauses with null N heads have Def in Amharic. The third complication is that coordinated APs or relative clauses have to have a Def on each conjunct. This is very unlike Bulgarian, where only the first coordinated adjective had Def. I can see two possible analyses of this phenomenon. The first is that apparent coordinated APs and relative clauses actually involve NP coordination with ellipsis. Then each NP would have its own Def head. Kramer (2010) does not give enough data to know whether this is a viable analysis. The second possible analysis says that coordination in Amharic has a very strong morphological matching requirement. This matching requirement makes the previously optional nominal concord obligatory in coordinations. Every conjunct then has to have an Agr morpheme that agrees in definiteness. (Something like this is basically Kramer's analysis.)

Assuming that these complications can all be accounted for, the analysis proposed for Bulgarian extends quite naturally to Amharic. Importantly, there is no need for post-syntactic movement operations, as Kramer (2010) proposes. All we need is N2W, the procedure regulating how elements are moved from the workspace and merged into the derivation, which we need in any case.

6 Conclusion

Most research on syntax has never bothered to spell out how exactly the syntax selects items from the numeration and merges them in the workspace. I have tried to spell this out. Spelling it out, I have proposed, requires a left-to-right derivation, and spelling it out as a left-to-right derivation has the consequence that it explains some problematic placement facts, like those of Bulgarian and Amharic. These were previously analyzed with post-syntactic processes, but in the current system, they fall out from the normal procedure for building structure from a numeration. In fact they are placed in the exact same way as English affixal negation, which attaches to the first auxiliary verb in the clause. The proposed system, with left-to-right selection, expects such placement.

One thing to note is that the notion of a numeration is crucial for the placement facts discussed here. It is not possible to select items directly from the lexicon and merge them in the workspace. The reason is that placement after the *first* item of a certain type requires something to merge *when it can*. If the syntax could access the lexicon at each step, then in Bulgarian and Amharic, it would select a Def *every time* it selected a [+N] element. In English, affixal negation would attach to *every* auxiliary verb, not just the first one. We need the notion of a numeration, with just one instance of Def or Neg in it, so that once it is merged, it is no longer available for merger with subsequent items of the appropriate type.

The phenomena discussed here therefore provide strong motivation both for the numeration, and left-to-right structure building.

References

Abney, Steven Paul. 1987. *The English noun phrase in its sentential aspect*. Cambridge, MA: Massachusetts Institute of Technology dissertation.

Al Khalaf, Eman. 2015. *Coordination and linear order*. University of Delaware dissertation.

Bruening, Benjamin. 2009. Selectional asymmetries between CP and DP suggest that the DP Hypothesis is wrong. In Laurel MacKenzie (ed.), *U. Penn working papers in linguistics 15.1: Proceedings of the 32nd annual Penn linguistics colloquium*, 26–35. Philadelphia: University of Pennsylvania Working Papers in Linguistics. Available at http://repository.upenn.edu/pwpl/vol15/iss1/.

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

Bruening, Benjamin. 2018a. Brief response to müller. *Language* 94. e67–e73.

Bruening, Benjamin. 2018b. The lexicalist hypothesis: Both wrong and superfluous. *Language* 94. 1–42.

Bruening, Benjamin. 2020. The head of the nominal is N, not D: N-to-D movement, hybrid agreement, and conventionalized expressions. *Glossa: A Journal of General Linguistics* 5(1). 15. 1?19. doi:10.5334/gjgl.1031.

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.

Chaves, Rui P. 2012. On the grammar of extraction and coordination. *Natural Language and Linguistic Theory* 30. 465–512.

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.

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

Fox, Danny. 2000. *Economy and semantic interpretation*. Cambridge, MA: MIT Press and MIT Working Papers in Linguistics.

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.

Hankamer, Jorge & Line Mikkelsen. 2005. When movement must be blocked: A reply to Embick and Noyer. *Linguistic Inquiry* 36. 85–125.

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.

Johnson, Kyle. 2009. Gapping is not (VP-) ellipsis. *Linguistic Inquiry* 40. 289–328.

Kramer, Ruth. 2010. The Amharic definite marker and the syntax-morphology interface. *Syntax* 13. 196–240.

Moltmann, Friederike. 1992. *Coordination and comparatives*. Massachusetts Institute of Technology dissertation. Distributed by MIT Working Papers in Linguistics, Cambridge, Mass.

Muadz, Husni. 2001. *Coordinate structures: A planar representation*. University of Arizona dissertation.

Ruys, E. G. 1992. *The scope of indefinites*. Utrecht University dissertation.

Zwicky, Arnold M. & Geoffrey K. Pullum. 1983. Cliticization vs. inflection: English *n't*. *Language* 59. 502–513.

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