Realizational Morphology in a Modular Minimalist Grammar

Edward P. Stabler
stabler@ucla.edu

Abstract

A modular minimalist grammar with a realizational morphology is briefly motivated and defined, inspired by recent morphosyntax. A modular grammar identifies and isolates components that are relatively causally independent, making components and their interactions easier to understand, without imprecision or approximation. The modular grammar proposed here assumes a realizational morphology in the sense that the atoms of syntax are not pronounced words, and syntax plays a role in word building. These grammars capture generalizations that previous minimalist grammars, and many other generative grammars, miss.

In Chomsky and Lasnik (1977) and Lasnik (1981), filters are added to a system of generative rules to block certain generated results. One filter blocks dependent affixes from appearing in structures unattached to a free morpheme:

> *Syntactic Structures* makes the claim that there could be another language just like English but where Affix Hopping is optional. The theory we're looking at now . . . makes the claim that there couldn't be any such language.
>
> Affix Hopping and DO-Support . . . describe but don't capture the . . . generalization: *A stranded affix is no good*. (Lasnik, 2000, p.123)

Various versions of this idea have persisted. Bresnan (2000) says "To counter the fact that DO is ungrammatical elsewhere, there must be a constraint that penalizes its presence." Grimshaw (1997) and Sag (2010) suggest that we need to account for the fact that DO is "necessary whenever it is possible." Stabler (2001) offers a DO-support transduction on morpheme sequences, to supplement a minimalist grammar (MG) with head movement and affix hopping.

One recent thread through this work is considered here, with particular attention to auxiliary verbs. Kayne (1993) says that while some heads may select their complements, "There is no auxiliary selection rule." Bjorkman (2011) says "BE is not directly selected for, but is instead inserted to support inflectional material that was unable to combine with a main verb." Olivier (2025) says "HAVE and BE are allomorphs". Kalin and Weisser (2025) say, "Combining all the evidence. . . the most adequate model. . . " is one that is non-lexicalist (syntactic word-building), post-syntactic (syntactic atoms have no phonology), and realizational (phonology 'realizes' features but not in lexical increments). Here, a very simple, preliminary kind of grammar with those three properties is defined.[1]

0 Some varieties of grammars

To situate this project, it will be useful to briefly review some of the formal grammars that have been used for human languages. A context free grammar uses a finite set of rules like 'S → NP VP' to rewrite S repeatedly until there is a string of words. But context free grammars for human languages require large numbers of rules. For example, extracting a context free grammar from the annotations of the Penn Treebank (Marcus et al., 1994), even when done rather carefully, yields a grammar of well over 10,000 rules, not counting the lexical rules. With that many rules, exhaustive parsing of longer sentences from the *Wall Street Journal* becomes infeasible (Charniak et al., 1998). That set of rules is also a very poor grammar, over- and undergenerating badly. Statistics help these grammars to do better in a probabilistic sense, but these grammars are provably unable to capture some patterns found in human languages: crossing dependencies, reduplication, and non-semilinear patterns.[2]

More expressive grammars grammars have been explored, including tree adjoining grammars (Joshi

[1] For a complete, transparent computer implementation and examples: https://github.com/epstabler/mol25.

[2] Shieber (1985); Culy (1985); Michaelis and Kracht (1997); Kobele (2006).

and Schabes, 1997), (parallel) multiple context free grammars (Seki et al., 1991), combinatory categorial grammars (Steedman and Baldridge, 2011), and minimalist grammars (Stabler, 1997).[3] These can in principle be smaller and less ad hoc than context free grammars, since they can define general patterns found in human languages that context free grammars can only approximate by enumerating instances. But, in practice, these more expressive grammars are still large and complex. Take the minimalist grammars proposed by Stabler and Keenan (2003), for example. Those grammars use 5 rules (three cases of 'merge' and two cases of 'move'). But those grammars do not include head movement or affix hopping. To allow those, Stabler (2001) increases the set of rules to 13, and the rules are rather complex. Stanojević (2019) proposes an alternative, more efficient set of 31 rules. But those grammars still do not include rules for coordination, adjunction, agreement, case marking, DO-support, and other things that have been extensively studied in linguistic theory. Adding those things in a principled way is difficult, because each rule gets so complex, and because the set of rules gets larger.

Linguists have proposed, and this paper confirms: refactoring the problem can help. There are relatively independent regularities in the linear order of constituents, in agreement and case, and in patterns of head displacement. But each rule in generative grammars mentioned above defines all of those at once, the way context free grammars do. Those grammars are all 'monostratal' in the sense that each rule application in those grammars builds all aspects of one part of the linguistic structure. An alternative, modular strategy splits apart relatively independent aspects of language structure and defines each aspect separately. Then each piece of structure must satisfy a number of constraints, rather than being defined all at once by a single, complex rule.

Linguists have been exploring grammars one aspect at at time since the beginning. This kind of modularity was the hallmark of Chomsky (1981) and related work. The more recent minimalist tradition, Chomsky (1995) and more recent proposals, has aimed to unify and consolidate some aspects of language previously regarded as distinct, and has separated others as 'post-syntactic'. But those post-syntactic components still cover significant aspects of language that remain of interest. So the minimalist perspective on language remains modular.

Interestingly, the 'minimalist' grammars of Stabler and Keenan (2003) and related work are not modular. They fall squarely in the monostratal tradition mentioned earlier. Each rule application builds all aspects of a piece of structure. That may explain, in large part, why extending early minimalist grammars becomes so complex.

Here, a modular grammar is proposed that generates structures very similar to the early monostratal minimalist grammars, but where the properties of different aspects of each piece of structure are separately defined. A trivially defined infinite set of binary trees is filtered in a sequence of steps, based on relatively distinct aspects of structure, and then transformed by making rather minor adjustment in 'non-syntactic' features. This architecture captures the monostratally defined minimalist structures and more.

Six modules are proposed. Each one is a bottom-up tree transduction, that is, each builds an output tree as it processes the input bottom-up, from the leaves to the root, node-by-node. The first three steps are partial identity transductions, filtering the initial set of all trees that can be built from the atoms. The last 3 steps then make minor adjustments in morphology. After these steps, the range of each modular grammar is very similar to the range of a rule-based, monostratal minimalist grammar from the 1990's. In fact, as explained below, I conjecture the string languages are exactly the string languages of those minimalist grammars. But the modular grammars proposed here capture new structures and new generalizations. Instead of many rules for head movement, for example, there is one simpler function with greater expressive power. It is simpler because it can be stated separately from everything else.[4]

[3] I have left 'constraint based' grammrs out of this list, since they rest on rather different formal foundations (Johnson, 1994; Pullum, 2013). Constraint based grammars have some properties in common with the modular generative grammars that will be the focus here. Our modularization is distinctively Chomskian, and can be formalized either generatively in terms of tree transducers or in terms of constraints (Graf, 2013).

[4] Hornstein (2024, pp.7-8) proposes "All grammatical relations are merge-mediated" as the "Fundamental Principle of Grammar" (FPG). That FPG sounds like the approach formulated here, since our mrg builds/accepts the initial trees and all the following modules either filter or make non-syntactic modifications in them. But Hornstein intends something stricter, suggesting that, in his sense, there can be no modularity in the 'faculty of language', though there may of course be 'interface' operations.

1 A modular grammar

The grammars defined here will each be compositions of 6 functions:

$$g = \text{vi} \circ \text{lin} \circ \text{hm} \circ \text{agr} \circ \text{sel} \circ \text{mrg},$$

where ∘ is standard function composition – often pronounced 'after' – and where: mrg builds/accepts binary trees over a set of syntactic atoms; sel maps those trees to themselves if selection features match appropriately; agr maps those trees to themselves if agreement features match appropriately; hm applies head movement to adjust morphological contents at the leaves (with no change in hierarchical structure); lin maps each tree to its pronounced linear order (with no change in hierarchical structure); and vi, vocabulary insertion, replaces morphological indices with phonologically specified morphs (with no change in hierarchical structure). These 6 functions are defined in the following sections. The composed grammar g is a function from (unpronounceable) binary trees to (hierarchically unchanged but reordered, pronounceable) binary trees. Language variation is attributed to (i) the atoms over which mrg operates and (ii) the rules of vi, as explained below.

2 mrg

Collins (2002), Chomsky (2007, p.8) and others propose an operation merge which forms binary sets over a lexicon. Here, for the definition of composed grammars, it is convenient to let mrg be the tree transducer which accepts (all and only) ordered binary trees with atoms at the leaves, mapping each such tree to itself. The linear ordering of the tree is inessential, but computationally useful.[5] So mrg is a function whose range is the set of binary trees over atoms that associate roots with features.[6]

We assume that the lexicon associates 'roots' with features in a triple, (root, sel-fs, agr-fs), where each root can be thought of as a morphological index that could underlie multiple pronunciations, sel-fs specifies selection features, and agr-fs specifies agreement features. The feature complexes are precisely defined below.

[5] A head-first order is enforced in §3 and used at a number of points, but is just a convenience. §3 shows it could be replaced in a set-based computation, identifying the head as the unique child with unchecked 'negative' features.

[6] This function is trivially definable by a bottom-up tree transducer (Baker, 1979). To keep all derivable syntactic objects in the domain of mrg, we extend it in ways defined below: allowing at the leaves any of finitely many sequences of roots built by head movement, and any of finitely many vocabulary items that can replace the roots.

3 sel

As noted by Collins (2002) and others, lexically specified selection requirements seem unavoidable.[7] The function sel accepts only those elements of the range of mrg that satisfy selection requirements, mapping those trees to themselves. Here, we use a notational variant of the selection checking in Kobele (2021), not too different from Stabler (1997). Selection requirements sel-fs are given by a formula fs⊸fs' where fs and fs' are (possibly empty) feature sequences (non-commutative conjunctions), where the features in the antecedent fs are *negative* and those in the consequent fs' are *positive*.[8] If fs is empty, the antecedent is omitted and fs' is simply written as a dot-separated sequence.

At each internal node, a feature must be 'checked'. Checking is computed without modifying the tree and its features in any way. Selection features are checked bottom-up as follows. At each leaf, the features sel-fs are given by the syntactic atom. Then, moving up through the tree, each internal node has the features (some of which may be checked) at the leaf which is at the end of its left branch, its head. At any internal node with left subtree x and right subtree y, if x has a subtree x' whose first unchecked feature is a positive f, then tree x' must be identical to y (disregarding features checked), those features f in x and x' are checked, and features of y are calculated from x'. This is called *internal merge* or *move*. If x has no subtree x' whose first unchecked feature is a positive f, then the first unchecked feature of y must be positive f, and both of those features are checked, written $\not{f}$. This is *external merge*. Schematically:

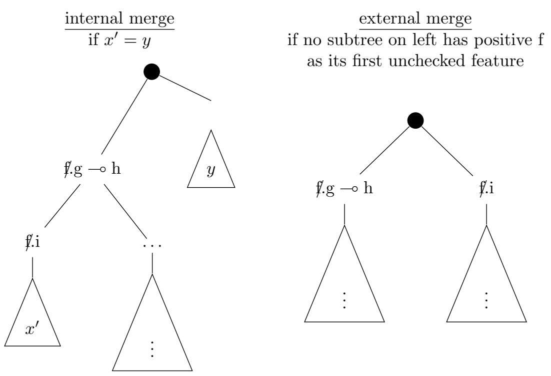


Figure 1 shows an example.

Stabler (1997, 2001) restricts feature checking with a *shortest move constraint*:

[7] But see e.g. Wurmbrand (2014), Kalin and Rolle (2024).

[8] The ⊸ is from linear logic (Girard, 1995, p.15).

(smc) Neither internal nor external merge is possible if it creates a tree in which the first unchecked features of two distinct subtrees are positive and identical (where subtrees related by internal merge are not counted as relevantly distinct).

This blocks analyses in which a single clause has multiple wh phrases all of which are only grammatical in the nearest position, as in

(1) *What does who say [(who) read (what)]?

(2) *Who says what [(who) read (what)]?

The smc is a mathematically simple and well-studied.[9] But evidence supports more complex locality conditions, still poorly understood. There is still controversy about the possible roles of various kinds of 'phases', 'relativized minimality', and 'minimal search'.[10] It is also proposed that internal merge sometimes moves just heads.[11] These and other elaborations can be left aside here.

4 agr

The function agr accepts only those elements of the range of sel that satisfy agreement requirements, mapping those trees to themselves. Following Béjar and Rezac (2009) and others, we assume this check on agreement is *cyclic*, i.e. calculated bottom-up at every step of transducing a tree.

Following Hanson (2025), search is restricted to the first d-commanding element on a specified tier.[12] A head *d-commands* the heads of its externally merged specifiers from highest to lowest, then the head of its complement, and then whatever the complement head d-commands – in that order. If agree is upward, with the probe encountered bottom-up before the goal, then tier d-commanders instead of d-commandees are searched. A *tier* $t = \{s_1, \ldots, s_n\}$ is a set of sets of features. *Head* h *is on tier t* if and only if some s_i is a subset of the features of h, where those include not only its agr features but also its positive selection features. Recall that atoms are triples (root,sel-fs,agr-fs). Let each agr-fs is a set of (tier,f) pairs, where each pair indicates that feature f must agree the tier, where f is feature with a type separated by :, and where underscore indicates unspecified (of a 'probe'). No atom can have two different agreement features of the same type. A probe feature ϕ:_ can be instantiated by goal feature ϕ:3s, becoming ϕ:$\underline{\text{3s}}$.

Unlike sel, agr features can enter into multiple checking relations. For example, ϕ:_ can be instantiated by ϕ:3s and the resulting ϕ:$\underline{\text{3s}}$ can again instantiate ϕ:_. See Figure 2.[13]

5 hm

There are some arguments that (at least some instances of what has been called) head movement are sensitive to adjacency in the linear, pronounced string, as evidenced in the contrast:

(3) Which cat (which cat) chase -s the rat

(4) Which rat do -s the cat chase (which rat)

Some analyses of this contrast propose that tense is unable to unite with the verb in (4), forcing the insertion of DO, because the overt subject *the cat* between the heads T and C somehow disrupts relations among C, T, V, and v, in a way that the moving element (which cat) does not in (3).[14] But even if that is true, it does not follow that linear order is relevant, since moved elements are featurally distinct. In both sel and agr, moved elements are treated as having the features associated with their base positions. So while we will agree with the proposal of Chomsky et al. (2023, p.66) and many others that head movement is post-syntactic, distinct from phrase construction, we reject the idea that this operation must be post-linearization.[15]

Adopting this hypothesis, we formulate a mechanism inspired by Arregi and Pietraszko (2021). Let a *span* be a sequence of $h_1, \ldots, h_n$ $(n \geq 2)$ where (i) for $i < n$, h_i selects the projection of h_{i+1}, (ii) for $i < n$, h_i has a root marked as morphologically *dependent* with a dash, -$\sqrt{}$root, (iii) h_n is not

[9] Graf (2023a); Hunter and Frank (2021); Kanazawa (2016, Ex. 5.8); Stabler (2011); Salvati (2011); Kobele et al. (2007); Michaelis (2001).

[10] Fernández-Serrano (2025); Branan and Erlewine (2025).

[11] See Harizanov (2019) and references cited there.

[12] The hypothesis that locality in agreement and selection derives from search has a long history (Chomsky, 2005; Epstein et al., 2020; Ke, 2024; Branan and Erlewine, 2025). Syntactic search on tiers is more recent (Graf, 2022, 2023b).

[13] Further exploration of agr is beyond the scope of this paper. Note that while Figure 2 puts case in the domain of agr as assumed for example by Bjorkman and Zeijlstra (2019) but not by Hanson, the proper treatment of case is controversial. For different tier-local analyses of case: Vu et al. (2019), Hanson (2023). Deal (2025), Kidwai (2023) argue that case is better treated as a head, not a feature. I conjecture that Ermolaeva and Kobele's (2022) argument that fully instantiated agreement structures are MG-definable can be adapted to our modular grammars with agr, if we assume the smc of §3.

[14] Pollock (1989, §5.5.4), Chomsky (1991, §2.3.1), Bobaljik (1995, §2.1), Sportiche (1998, §5.2.3.1), Embick and Noyer (2001), Bjorkman (2011), Arregi and Pietraszko (2021).

[15] In agreement with, e.g. Branigan (2023, §2.1).

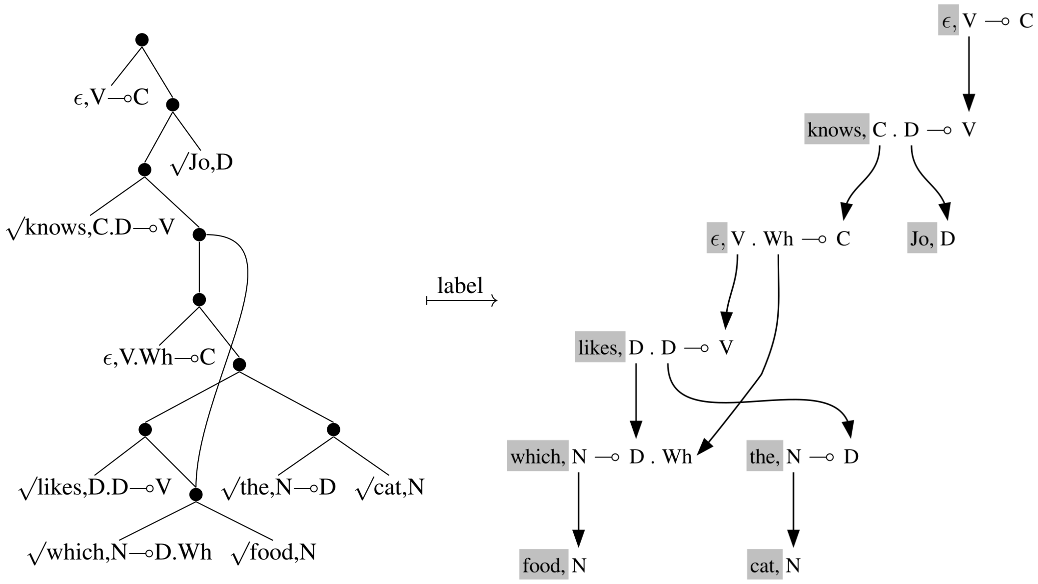


Figure 1: Function sel checks the binary tree on the left for *Jo knows which food the cat likes* (showing roots and sel-fs). At each internal node •, the left child is the head. And to show where internal merge (movement) has applied, the two identical subtrees for *which food* are shown as one node. But our calculations apply to the binary tree, labeling nodes by computing the checking relations shown on the right. Each checking arrow goes from a negative occurrence of a feature to a positive occurrence of that same feature. One feature remains unchecked: C labels the root. The function sel maps the tree on the left to itself. This keeps the computation simple. When checking is needed again, it can be recomputed efficiently.

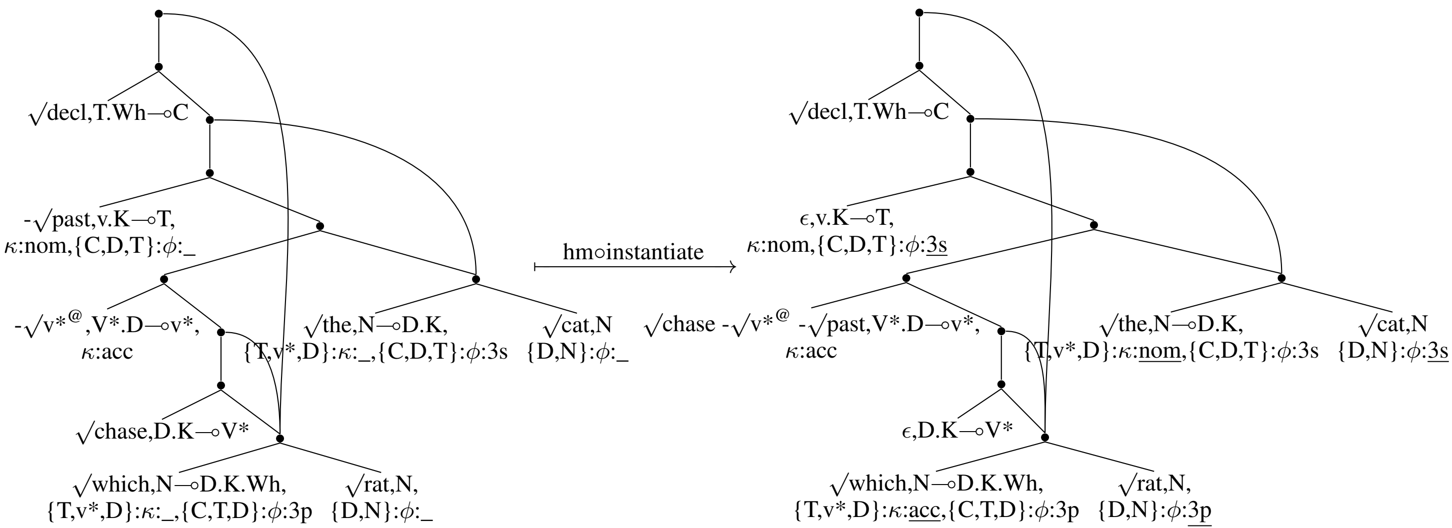


Figure 2: Function hm and agr instantiation act on the embedded clause of *I know [which rat-s the cat chase-s]*. Note that clause structure is elaborated slightly beyond what Figure 1 depicts and atoms are shown with roots, sel-fs and agr-fs. Here, agr instantiates the _ probe features bottom-up with adjacent elements in d-command sequences on the indicated tier, as described in §4. In this example, agr goals agree downward: D values ϕ on N; T values ϕ on D; v* and T value κ on D. We show the tiers only for the probes. Then, as described in §5, hm applies to the -T -$\sqrt{v}^{*@}$ V* chain, putting the result in the strong v* position. Like sel, agr is a check. That is, it makes sure instantiation is possible, but then leaves the structure uninstantiated. This keeps the computation simple, and the instantiation can be recomputed whenever needed. But here we show the instantiated tree to illustrate the mechanism. Function hm, on the other hand, is a transformation. It makes a change only in morphs at the leaves, leaving syntactic (sel and agr) features and hierarchical structure untouched, so that is left in place for lin and vi.

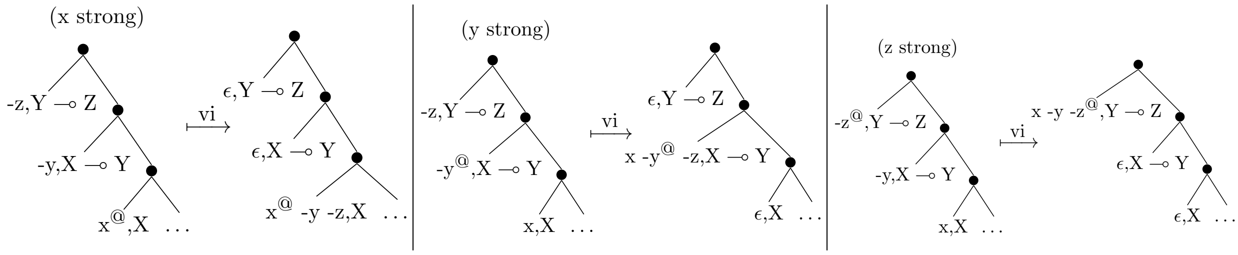


Figure 3: Function hm acts on three cases with span -z, -y, x. The morphs x and -y raise to left-concatenate to -z, in mirror order. That complex is then placed in the highest strong @ position, leaving the other morph positions empty.

marked dependent, and (iv) the sequence is maximal in the sense that h_1 is not the first merge of a higher dependent-marked head. We also require that there is no recursion in spans: no two heads in a span have the same first positive selection feature.

Any head in a span may also be marked *strong*, indicated here by following the root with a @.[16] If no head in the span is marked strong, then h_1 is the strongest. Otherwise, the strongest head is the first in the span that is marked @. In spans, the morphs of dependent heads left-adjoin to h_n, and the complex is pronounced at the strongest position. Representing the left-adjunction of morphs by left-concatenation, this defines the patterns of Figure 3, where the derived order x, y, z of morphs mirrors the syntactic projections Z, Y, X. In the figure, note that in the (x strong) case, both y and x are lowered. In (y strong), x is lowered and z is raised.[17]

Previous minimalist grammars (Stabler, 2001) formalize English subject-auxiliary inversion with a pattern like the (z strong) case of Figure 3, with the complex (verb@ -tense -C@) pronounced in strong question-forming -C@. And English affix-hopping is similar to lowering -tense onto a verb@ with a weak C, like the (x strong) case. See Figure 2. The function vi of §7 allows this analysis to be elaborated in new ways, as in §8.2 below.

6 lin

Adapting an idea from Kayne (1994, 2020), Chomsky (1995), Cinque (2023), linear order is adjusted:

(ord) at any node with daughters x, y where head x itself has daughters, reorder to y, x.

[16] Brody (1997), Svenonius (2016), i.a. also use @. Arregi and Pietraszko (2021) use *.

[17] Cf. Brody (1997, 2000); Adger et al. (2009); Svenonius (2016); Harizanov and Gribanova (2019); Branigan (2023); Giannoula (2025). Kobele (2002) shows how the mirror theory grammars proposed by Brody (1997) are weakly equivalent to MGs. I conjecture that Kobele's result extends to our our similar composed grammars with hm.

And setting the stage for vi, we strip features that are redundant at the interface:[18]

(del) Delete non-final moving elements.

This deletion replaces non-final moving elements, i.e. elements that arguments to an internal merge step, by an empty structure. Then we define: lin = ord ∘ del. See Figure 4.

With del, it is important to remember why internal merge produces structures with identical subconstituents in the first place. Why have those elements in the structure if they are not going to be pronounced? In the current formulation, that question stands out because sel uses an explicit check on whether two subconstituents of unbounded size are identical. And similarly, in tree transducer implementations, transducers with unbounded copying are required (Kobele et al., 2007). With del, why formulate internal merge that way?

Support for internal merge with copying comes from evidence that the copies are really there. There is a large literature, but one kind of supporting argument comes from evidence that, in some constructions, copies are not completely deleted. Yuan (2025) notes, for example, that a number of analyses in the literature claim that deletion of a the copy of a fronted verb is blocked when that would leave a tense affix without a host. She then observes that in Inuktitut we find affixal verbs that require a DP host. And there too, deletion of the DP under identity is blocked when the deletion would leave an affixal verb without a host. As a result, Inuktitut shows a variety of spellout patterns inconsistent with the simple deletion process of del. For the moment, though, allowing del to completely remove copies before pronunciation is a good first approximation. More sophisticated deletion rules are left for future work.

[18] Cf. Chomsky et al. (2023, p26): "An efficiency consideration at [the sensorimotor interface SM] is to pronounce just one (the highest) of a set of copies."

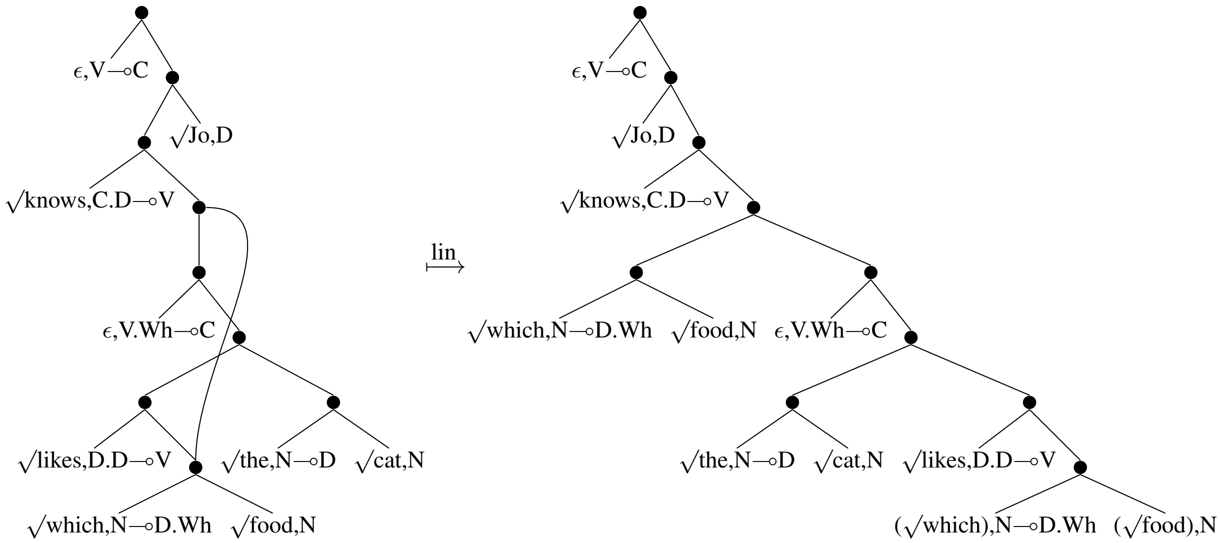


Figure 4: Left, repeated from Figure 1. Right, the result of applying lin. As often done, instead of emptying non-final copies, those copies are shown with their morphs in parentheses.

7 vi

Function vi replaces roots with phonologically instantiated forms, based on syntactic context. For example, the rules

$$\sqrt{\text{cat}} \rightarrow \text{cat}$$
$$\phi\text{:3p} \rightarrow \text{-s}$$

map a leaf of the syntactic tree to the form that underlies its pronunciation, which we will here indicate just with standard orthography. So the root in the syntactic atom is replaced (but other features mentioned in the rules are left in place):

$$(\sqrt{\text{cat}},\text{D},\phi\text{:3p}) \quad \Rightarrow \quad (\text{cat -s},\text{D},\phi\text{:3p}).$$

Irregular plurals are handled with special rules:

$$\sqrt{\text{mouse}}\ \phi\text{:3s} \rightarrow \text{mice}.$$

Special rules like this take precedence over the defaults, 'blocking' them, because they are 'more specific' in the sense that they specify more features (Halle and Marantz, 1993; Embick and Marantz, 2008). With this specificity order defined on our finite set of vi rules, we use the first applicable rule.

The domain of this kind of competition among vi rules was widely held to be a single atom, when atoms were morphemes. But as syntactic principles have entered into word building, features that used to be included among those associated with morphemes are now often split away, treated as syntactic heads. When heads are combined by head movement, these features may be reunited. For example, with the rules

$$\sqrt{\text{chase}} \rightarrow \text{chase}$$
$$\sqrt{\text{past}} \rightarrow \text{-ed,}$$

we derive a pronounced form for this complex head from Figure 2, mapping each element pointwise, but preferring rules with the largest numbers of features:

$$(\sqrt{\text{chase}}\ \sqrt{\text{past}}\ \sqrt{\text{decl}}) \Rightarrow \text{chase -ed}.$$

Adapting ideas from Svenonius (2016) and Haugen and Siddiqi (2016), vi rules can also target spans, where a span is a sequence of heads, each selecting the next, as in hm, but the heads need not be morphologically dependent (marked by -). We replace conditions (ii) and (iii) in the previous definition of *span* in §5 with: (ii') $h_1, \ldots, h_n$ are in the domain of a vi rule. As in hm, we let the highest head be the default strongest. So then, following Haugen and Siddiqi (2016, (7)) we allow this vi rule to derive the Spanish portmanteau *del* from a span that includes a preposition and determiner, indicating the boundaries of syntactic heads with parentheses:

$$(\sqrt{\text{de}})\ (\sqrt{\text{el}}) \quad \rightarrow \quad \text{del}.$$

This rule replaces ($\sqrt{\text{de}}$) with *del* in the preposition, and removes root ($\sqrt{\text{el}}$) from the determiner.

Note that mirror order is not predicted by these vi operations unless operating on a complex formed by hm, and vi rule choice is subject to blocking. We extend these rules slightly in §8.3 below to allow them to test d-commanders on a specified tier, but leave to future work an integration with more developed theories of spellout.

8 Examples

8.1 Nominalization.

In standard presentations of logic, each predicate has an arity, but in human languages, a single pronounced form of of a verb like *capture* allows various numbers of arities and can also be a noun with the same arguments, as in *the capture (of markets) (by oligarchs)*. The morph is associated with a kind of event, allowing its various arguments to be expressed or not in various contexts. The verb *destroy* on the other hand has a distinct a nominalized form *destruction*. Rather than treating such coincidences between nominal and verbal forms as accidental, Chomsky (1970) suggests a common underlying form may be mapped to different pronunciations. Here, vi plays that role using the rules,

$(\sqrt{\text{capture}}) \rightarrow$ capture
$(\sqrt{\text{destroy}}\ \text{V}) \rightarrow$ destroy
$(\sqrt{\text{destroy}}\ \text{N}) \rightarrow$ destruction.

If -$\sqrt{}$-tion is a nominalizing head of some kind, then head movement forms a complex to which a very similar vi rule can apply.[19]

8.2 English auxiliaries.

While that specification of various pronounced forms of $\sqrt{}$destroy could be entirely a lexical matter, agr, hm and vi get involved in determining how other words are built and pronounced. For example, pronouncing *be* requires attention to features:

> Suppose the PF form of a lexical entry is completely unpredictable: the English copula, for example. In this case the lexical coding will provide whatever information the phonological rules need to assign a form to the structure [copula, {F}], where {F} is some set of formal features (tense, person, etc.). (Chomsky, 1995, §4.2.2)

This is exactly our strategy. But unlike Chomsky, we generalize it not just to certain auxiliaries but through the whole vocabulary.

In English, the auxiliary verbs *have* and *be* regularly combine with past and progressive participles, respectively. But in various languages, the appropriate auxiliary for a past participle depends on a number of factors that vary across dialects. Bjorkman (2011) argues that auxiliaries appear not when they are selected but when they are needed to rescue a kind of 'overflow' situation.[20] When a context does not provide a way to express tense on the verb, it can 'spill over' into another form, 'rescuing' the structure. She notes that in Arabic and in the Bantu language Kinande, verbs cannot have both a tense and an aspect marker, so when the main verb has aspect, tense gets expressed on an auxiliary. She argues that English falls into this pattern too, with an analysis that is easily modelled in our grammars, as shown in Figure 5.[21]

8.3 French auxiliaries.

In Standard European French passé composé constructions, some verbs require a HAVE auxiliary, while others require BE. One rough rule for French learners says that when a verb has a direct object, the auxiliary will be HAVE. That is, usually, transitives and unergatives take HAVE, while unaccusatives take BE. Bjorkman (2011) proposes that the presence of an object between a perfective head and the verb blocks agreement between them, yielding HAVE instead BE.

Olivier (2025) observes that this simple rule conflicts with the fact that reflexive verbs with direct objects require HAVE. He suggests instead that an Aux head intervenes between T and v_{Prt}, and that Aux is realized as HAVE if person features on T and Aux are not guaranteed to be identical, and otherwise as BE. This gives the basic unaccusative/transitive contrast if Aux always agrees with the internal argument, while T agrees with the closest element on the {C,T,D} tier. But we do not want the realization rules to have to test whether identities are guaranteed. The alternative is to test for the conditions of the guarantee. Assuming that transitive feature v* is distinct from v, this simple case, the contrast can be decided by testing whether on v* is adjacent on the {T,v*,v} tier:

(Aux) $\rightarrow$ HAVE if v^* is $\{\text{T}, \text{v}^*,\text{v}\}$-adjacent
(Aux) $\rightarrow$ BE

See Figure 6. A fuller integration of Olivier (2025) and similar approaches is left for future work.

9 Conclusions

MG properties adjusted. Previous minimalist grammars have been monostratal, but simplicity

[19] See e.g. Alexiadou and Borer (2020) and references cited there for recent discussion of Chomsky (1970) and recent variants of that kind of proposal.

[20] There are many rescue analyses for auxiliaries (Embick, 2000; Arregi and Klecha, 2015; Fenger, 2019; Cruschina and Calabrese, 2021). An empirical defense of these is beyond the scope of this paper, where we aim only to explore some computational properties. And see e.g. Pietraszko (2023) for a possible non-realizational account of overflow patterns.

[21] A naive DO-support could be implemented with a vi rule (T $\rightarrow$ DO). But Bjorkman (2011) argues that a broader consideration of *do*-like patterns disconfirms rescue analyses. So we leave this for future work.

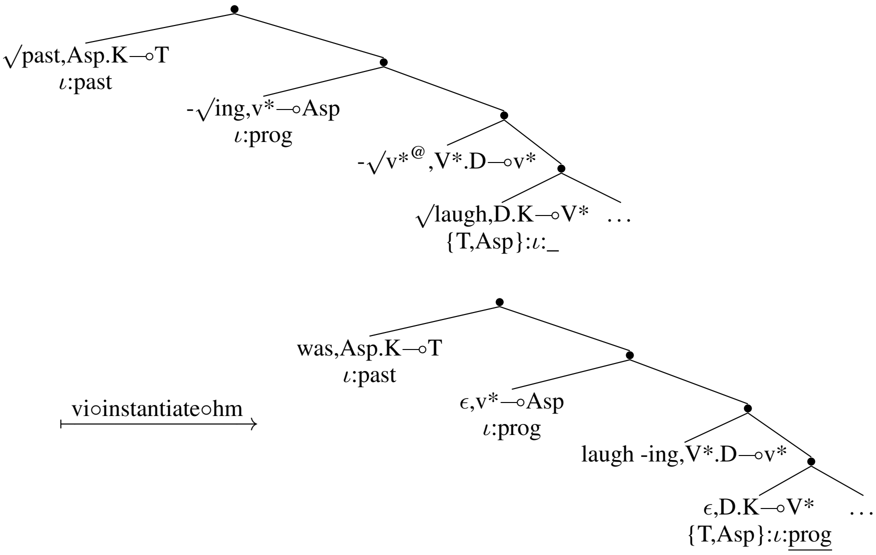


Figure 5: Insertion of English auxiliary BE by vi. An inflectional probe ι:_ on V can be instantiated by ι:past in simple clauses that lack Asp. But here, instantiating bottom-up, ι:prog on Asp values the feature on V, and so valuation by ι:past is not possible. We cannot assume that T is marked here dependent for hm, since the ι values in the resulting complex would conflict, and vi could not realize the structure. The function vi rescues the structure by inserting √BE and mapping √BE+√past to *was*. The complex placed in the strong v* position by head movement, √laugh √v* √ing, is mapped by vi to *laugh-ing*. As usual, vi∘hm∘agr leaves syntactic structure and features untouched, affecting only the morphs inside the leaves. Compare e.g. Bjorkman (2011, §2.3.4.1).

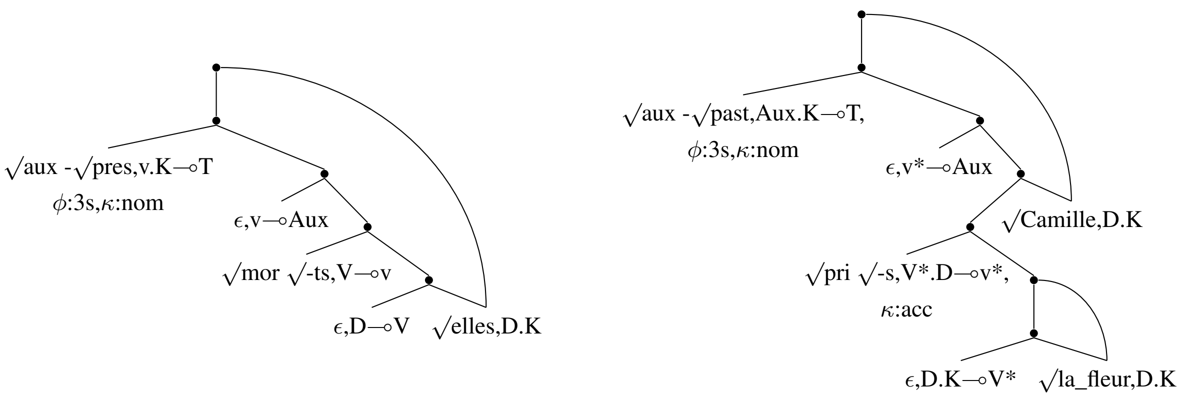
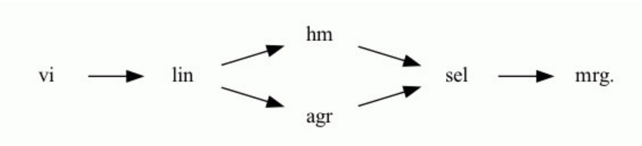


Figure 6: French BE and HAVE as allomorphs. In unaccusative on the left, hm raises V √mor to v √-ts and raises Aux √aux to T √pres. With the simple vi rules in the text, Aux is realized as BE because it is not adjacent to v on the {T,v*,v} tier, and so we get *Elles sont mor-ts* ('they died'). In the the transitive example, right, hm raises V √pri to v √-t and Aux √aux to T √pres. Aux with {T,v*,v}-adjacent v* is realized as HAVE, to yield *Camille a pri-s la fleur* ('Camille took the flower'). Compare Olivier (2025, (37),(38)), Bjorkman (2011, §2.4.4.1,(90),(91)).

and flexibility are enhanced in modular formulation, making understanding and experimentation with alternatives easier. Here, syntactic atoms with unpronounceable roots define the domain of mrg, while agreement, head-movement, linearization, and vocabulary insertion are 'post-syntactic' in the sense of applying after mrg.

Function vi uses a finite set of (language-specific) rules, applied in an order of decreasing specificity. Defaults get listed as the last case. This kind of competition among alternatives amounts to providing defaults with a kind of negative condition – something that rule-based minimalist grammars do not offer. But since these rules are finite in number and scope, the expressive power of the formalism remains restricted.[22]

We can diagram dependencies among the six modules above with a graph of a sort used in software construction, where each module has those it depends on as its descendants:



The function sel relies on the binary structure of mrg. Functions agr and hm rely on the head-complement and internal/external-merge distinctions enforced by sel. The function lin reorders heads and complements, so agr and hm can be slightly simpler if lin applies after them.

Ongoing research is focused on whether we have properly understood these dependencies. In particular, as research cited above shows, considerable efforts can be regarded as focused on uniting sel and agr in a more general labeling theory, and on uniting hm and vi in a general interface theory.

MG properties preserved: Two conjectures. I conjecture that, if we assume the smc of §3 and the del of §6, the string languages definable by our modular minimalist grammars are definable by rule-based, monostratal minimalist grammars and so, as shown by Michaelis (1998), also by multiple context free grammars (MCFGs). Some support for this conjecture is provided by the discussion and notes above.[23]

But is this conjecture interesting, when it depends on smc and del, which are clearly not right? I think the conjecture *is* interesting because, plausibly, well-understood computational approaches to MCFGs will extend easily to the better theories that improve on smc and del. This is my second, more speculative conjecture. One bit of evidence for it, relevant to del, comes from the easy extension of MCFGs to 'parallel' MCFGs with unbounded copying (Seki et al., 1991). The copying noted by Yuan (2025) and others is small, perhaps shaped by performance factors. Structures like that should not impose any big extra burden on parsing and learning. Another piece of evidence, relevant to the smc, comes from the fact that the smc itself is not integrated into the foundations of the grammar. It is a separately stipulated condition. When we find better, empirically well-supported locality conditions, we may be able to impose those in a similar way, without significantly disrupting the basic dependencies that structure our grammars. And on the small structures relevant in human language processing, again, less restrictive locality conditions may be compatible with models of human parsing and generation.

New capture of prominent generalizations. With roots in syntactic atoms, we can model something like Chomsky's (1970) nominalization. With vocabulary insertion, we can model something like the Halle and Marantz (1993) competition-based allomorphy. And we can capture something similar to the 'overflow' analyses of English and French auxiliaries proposed by Bjorkman (2011) and Olivier (2025). Like earlier DO-support analyses, these involve 'last-resort' rules which are, we conjecture, weakly equivalent to rule-based MGs but more succinct. Rule-based MGs miss those generalizations, but can enumerate the relevant cases.

Acknowledgements.

For discussions of earlier versions of this work, thanks to audiences at the Stony Brook University MG+ meetings, especially Meaghan Fowlie, John Hale, and Kristine Yu.

[22]It is interesting to compare negation-as-failure and other extensions to Horn clauses in logic programming and type class definitions (Miller, 2022; Bottu et al., 2017). Kanazawa (2007, 2009, 2017) observes that because MCFGs can be elegantly expressed in definite Horn clauses, they are easily parsed in Datalog. Our modular grammars cannot be directly expressed in definite Horn clauses, but as explained below, we conjecture that the modular grammars are equivalent to MGs and MCFGs, which can be so-expressed.

[23]See fns. 6, 9, 13, 17. A compiler that takes modular grammars (i.e. finite sets of atoms and vi rules) to equivalent MCFGs is also in preparation, adding to the collection of similar compilers for previous mildly context sensitive grammars (Guillaumin, 2004).

References

David Adger, Daniel Harbour, and Laurel Watkins. 2009. *Mirrors and Microparameters*. Cambridge University Press.

Artemis Alexiadou and Hagit Borer. 2020. Introduction. In A. Alexiadou and H. Borer, editors, *Nominalization*, pages 1–23. Oxford University Press.

Karlos Arregi and Peter Klecha. 2015. The morphosemantics of english past tense. *Procs. North Eastern Linguistic Society*, 45:53–66.

Karlos Arregi and Asia Pietraszko. 2021. The ups and downs of head displacement. *Linguistic Inquiry*, 52:241–289.

Brenda Baker. 1979. Composition of top-down and bottom-up tree transductions. *Information and Control*, 41:186–213.

Susana Béjar and Milan Rezac. 2009. Cyclic agree. *Linguistic Inquiry*, 40:35–73.

Bronwyn Bjorkman. 2011. *BE-ing Default: The Morphosyntax of Auxiliaries*. Ph.D. thesis, MIT.

Bronwyn Bjorkman and Hedde Zeijlstra. 2019. Checking up on (ϕ-)agree. *Linguistic Inquiry*, 50:527–569.

Jonathan David Bobaljik. 1995. *Morphosyntax: The syntax of verbal inflection*. Ph.D. thesis, MIT.

Gert-Jan Bottu, Georgios Karachalias, Tom Schrijvers, Bruno C. d. S. Oliveira, and Philip Wadler. 2017. Quantified class constraints. *ACM SIGPLAN Notices*.

Kenyon Branan and Michael Yoshitaka Erlewine. 2025. Locality and (minimal) search. In K.K. Grohmann and E. Leivada, editors, *Cambridge Handbook of the Minimalist Program*. Cambridge University Press.

Phil Branigan. 2023. *The Grammar of Multiple Head Movement*. Oxford University Press.

Joan Bresnan. 2000. Optimal syntax. In J. Dekkers, F. van der Leeuw, and J. van de Weijer, editors, *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford University Press.

Michael Brody. 1997. Mirror theory. *UCL Working Papers in Linguistics*, 9.

Michael Brody. 2000. Mirror theory: Syntactic representation in perfect syntax. *Linguistic Inquiry*, 31:29–56.

Eugene Charniak, Sharon Goldwater, and Mark Johnson. 1998. Edge-based best-first chart parsing. *Procs. Workshop on Very Large Corpora*, 6:127–133.

Noam Chomsky. 1970. Remarks on nominalization. In N. Chomsky, editor, *Studies on Semantics in Generative Grammar*. De Gruyter.

Noam Chomsky. 1981. *Lectures on Government and Binding*. Foris, Dordrecht.

Noam Chomsky. 1991. Some notes on economy of derivation and representation. In R. Freidin, editor, *Principles and Parameters in Comparative Grammar*, page 417–454. MIT Press. Reprinted in Chomsky 1995.

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press.

Noam Chomsky. 2005. Three factors in language design. *Linguistic Inquiry*, 36:1–22.

Noam Chomsky. 2007. Approaching UG from below. In U. Sauerland and H.-M. Gärtner, editors, *Interfaces + Recursion = Language?* de Gruyter.

Noam Chomsky and Howard Lasnik. 1977. Filters and control. *Linguistic Inquiry*, 8:425–504.

Noam Chomsky, T. Daniel Seely, Robert C. Berwick, Sandiway Fong, M.A.C. Huybregts, Hisatsugu Kitahara, Andrew McInnerney, and Yushi Sugimoto. 2023. *Merge and the Strong Minimalist Thesis*. Cambridge University Press.

Guglielmo Cinque. 2023. *On Linearization: Toward a Restrictive Theory*. MIT Press.

Chris Collins. 2002. Eliminating labels. In S.D. Epstein and T.D. Seely, editors, *Derivation and Explanation*. Blackwell.

Silvio Cruschina and Andrea Calabrese. 2021. Fifty shades of morphosyntactic microvariation. In M.-O. Hinzelin and E.-M. Remberger, editors, *Formal Approaches to Romance Morphosyntax*, pages 145–198. de Gruyter.

Christopher Culy. 1985. The complexity of the vocabulary of Bambara. *Linguistics and Philosophy*, 8:345–352.

Amy Rose Deal. 2025. Case sensitivity reflects case structure: Agreement, extraction, and clitics. UC Berkeley.

David Embick. 2000. Features, syntax, and categories in the Latin perfect. *Linguistic Inquiry*, 31:185–230.

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

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

Samuel D. Epstein, Hisatsugu Kitahara, and T. Daniel Seely. 2020. Unifying labeling under minimal search in single- and multiple-specifier configurations. *Coyote Papers: Working Papers in Linguistics, Linguistic Theory at the University of Arizona*.

Marina Ermolaeva and Gregory M. Kobele. 2022. Agreement as information transmission. *Syntax*, 25:466–507.

Paula Fenger. 2019. Size matters: Auxiliary formation in the morpho-syntax and the morpho-phonology. *Procs. North East Linguistic Society*, 49.

Irene Fernández-Serrano. 2025. Phase theory: inception, developments and challenges. In K.K. Grohmann and E. Leivada, editors, *Cambridge Handbook of the Minimalist Program*. Cambridge University Press.

Mina Giannoula. 2025. Deciphering mirror principle violations. *Morphology*.

Jean-Yves Girard. 1995. Linear logic: Its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 1–42. Cambridge University Press.

Thomas Graf. 2013. *Local and Transderivational Constraints in Syntax and Semantics*. Ph.D. thesis, UCLA.

Thomas Graf. 2022. Typological implications of tier-based strictly local movement. *Procs. Society for Computation in Linguistics*, 5:184–193.

Thomas Graf. 2023a. Minimalism and computational linguistics. *LingBuzz*, 005855.

Thomas Graf. 2023b. Subregular tree transductions, movement, copies, traces, and the ban on improper movement. *Procs. Society for Computation in Linguistics*, 6:289–299.

Jane Grimshaw. 1997. Projection, heads, and optimality. *Linguistic Inquiry*, 28:373–422.

Matthieu Guillaumin. 2004. Conversions between mildly sensitive grammars. UCLA and École Normale Supérieure, Paris.

Morris Halle and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In K. Hale and S.J. Keyser, editors, *The View from Building 20*, pages 111–176. MIT Press.

Kenneth Hanson. 2023. A TSL analysis of Japanese case. *Procs. Society for Computation in Linguistics*, 6:15–24.

Kenneth Hanson. 2025. Tier-based strict locality and the typology of agreement. *Journal of Language Modeling*, 13:43–97.

Boris Harizanov. 2019. Head movement to specifier positions. *Glossa*, 4:140.

Boris Harizanov and Vera Gribanova. 2019. Whither head movement? *Natural Language and Linguistic Theory*, 37:461–522.

Jason D. Haugen and Daniel Siddiqi. 2016. Towards a restricted realization theory. In D. Siddiqi and H. Harley, editors, *Morphological Metatheory*. John Benjamins.

Norbert Hornstein. 2024. *The Merge Hypothesis*. Cambridge University Press.

Tim Hunter and Robert Frank. 2021. Comparing methods of tree-construction across mildly context-sensitive formalisms. *Procs. Society for Computation in Linguistics*, 4:355–358.

Mark Johnson. 1994. Two ways of formalizing grammars. *Linguistics and Philosophy*, 17(3):221–248.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Volume 3: Beyond Words*, pages 69–124. Springer.

Laura Kalin and Nicholas Rolle. 2024. Deconstructing subcategorization. *Linguistic Inquiry*, 55:197–218.

Laura Kalin and Philipp Weisser. 2025. Minimalism and morphology. In K.K. Grohmann and E. Leivada, editors, *Cambridge Handbook of the Minimalist Program*. Cambridge University Press.

Makoto Kanazawa. 2007. Parsing and generation as Datalog queries. *Procs. Association for Computational Linguistics*, 45:176–183.

Makoto Kanazawa. 2009. The pumping lemma for well-nested multiple context-free languages. *Developments in Language Theory*, 13:312–325.

Makoto Kanazawa. 2016. Formal grammar: An introduction. Lecture 5: Mildly context-sensitive languages. Lecture notes, Hosei University.

Makoto Kanazawa. 2017. Parsing and generation as Datalog query evaluation. *Journal of Logics and their Applications*, 4.

Richard S. Kayne. 1993. Toward a modular thoery of auxiliary selection. *Studia Linguistica*, 47:1–31.

Richard S. Kayne. 1994. *The Antisymmetry of Syntax*. MIT Press.

Richard S. Kayne. 2020. Antisymmetry and externalization. *LingBuzz*, 005554.

Alan Hezao Ke. 2024. Can agree and labeling be reduced to minimal search? *Linguistic Inquiry*, 55:849–870.

Sana Kidwai. 2023. *Voice, Case and the External Argument: The Perspective from Urdu*. Ph.D. thesis, University of Cambridge.

Gregory M. Kobele. 2002. Formalizing mirror theory. *Grammars*, 5:177–221.

Gregory M. Kobele. 2006. *Generating Copies*. Ph.D. thesis, UCLA.

Gregory M. Kobele. 2021. Minimalist grammars and decomposition. Leipzig University.

Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In *Model Theoretic Syntax at 10, ESSLLI'07*.

Howard Lasnik. 1981. Restricting the theory of transformations: A case study. In N. Hornstein and D. Lightfoot, editors, *Explanation in Linguistics*. Longman.

Howard Lasnik. 2000. *Syntactic Structures Revisited*. MIT Press.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: annotating predicate argument structure. *Procs. Workshop on Human Language Technology*, pages 114–119.

Jens Michaelis. 1998. Derivational minimalism is mildly context-sensitive. *Procs. Logical Aspects of Computational Linguistics*, 3:179–198.

Jens Michaelis. 2001. *On Formal Properties of Minimalist Grammars*. Ph.D. thesis, Universität Potsdam.

Jens Michaelis and Marcus Kracht. 1997. Semilinearity as a syntactic invariant. *Procs. Logical Aspects of Computational Linguistics*, 1:329–345.

Dale Miller. 2022. A survey of the proof-theoretic foundations of logic programming. *Theory and Practice of Logic Programming*, 22:859–904.

Marc Olivier. 2025. A syntactic account of auxiliary selection in French. *Probus*, 31.

Asia Pietraszko. 2023. Cyclic selection: Auxiliaries are merged, not inserted. *Linguistic Inquiry*, 54:350–377.

Jean-Yves Pollock. 1989. Verb movement, universal grammar, and the structure of IP. *Linguistic Inquiry*, 20:365–424.

Geoffrey K. Pullum. 2013. The central question in comparative syntactic metatheory. *Mind and Language*, 28:492–521.

Ivan A. Sag. 2010. Sex, lies, and the English auxiliary system. Stanford University.

Ivan A. Sag, Rui P. Chaves, Anne Abeillé, Bruno Estigarribia, Dan Flickinger, Paul Kay, Laura A. Michaelis, Stefan Müller, Geoffrey K. Pullum, Frank Van Eynde, and Thomas Wasow. 2020. Lessons from the English auxiliary system. *Journal of Linguistics*, 56:87–155.

Sylvain Salvati. 2011. Minimalist grammars in the light of logic. In S. Pogodalla, M. Quatrini, and C. Retoré, editors, *Logic and Grammar*. Springer.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science*, 88:191–229.

Stuart M. Shieber. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–344.

Dominique Sportiche. 1998. *Partitions and Atoms of Clause Structure : Subjects, Agreement, Case and Clitics*. Routledge.

Edward P. Stabler. 1997. Derivational minimalism. *Procs. Logical Aspects of Computational Linguistics*, 1:68–95.

Edward P. Stabler. 2001. Recognizing head movement. *Procs. Logical Aspects of Computational Linguistics*, 4:254–260.

Edward P. Stabler. 2011. Computational perspectives on minimalism. In C. Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–641. Oxford University Press.

Edward P. Stabler and Edward L. Keenan. 2003. Structural similarity within and among languages. *Theoretical Computer Science*, 293:345–363.

Miloš Stanojević. 2019. On the computational complexity of head movement and affix hopping. *Procs. Formal Grammar*, 24:101–116. Springer LNCS 11668.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. In R. Borsley and K. Börjars, editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*. Wiley-Blackwell.

Peter Svenonius. 2016. Spans and words. In H. Harley and D. Siddiqi, editors, *Morphological Metatheory*, page 199–220. John Benjamins, Amsterdam.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax. *Procs. Society for Computation in Linguistics*, 2:267–276.

Susi Wurmbrand. 2014. The merge condition: A syntactic approach to selection. In P. Kosta, S.L. Franks, T. Radeva-Bork, and L. Schürcks, editors, *Minimalism and Beyond*. John Benjamins.

Michelle Yuan. 2025. Morphological conditions on movement chain resolution. *Natural Language and Linguistic Theory*, 43:509–562.