

2018

Grammar Size and Quantitative Restrictions on Movement

Thomas Graf

Stony Brook University, easychair@thomasgraf.net

Follow this and additional works at: <https://scholarworks.umass.edu/scil>

 Part of the [Computational Linguistics Commons](#), and the [Syntax Commons](#)

Recommended Citation

Graf, Thomas (2018) "Grammar Size and Quantitative Restrictions on Movement," *Proceedings of the Society for Computation in Linguistics*: Vol. 1 , Article 4.

DOI: <https://doi.org/10.7275/R5348HJ8>

Available at: <https://scholarworks.umass.edu/scil/vol1/iss1/4>

This Paper is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Proceedings of the Society for Computation in Linguistics by an authorized editor of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

Grammar Size and Quantitative Restrictions on Movement

Thomas Graf

Stony Brook University
Stony Brook, NY 11794-4376
mail@thomasgraf.net

Abstract

Recently it has been proved that every Minimalist grammar can be converted into a strongly equivalent *single movement normal form* such that every phrase moves at most once in every derivation. The normal form conversion greatly simplifies the formalism and reduces the complexity of movement dependencies, but it also runs the risk of greatly increasing the size of the grammar. I show that no such blow-up obtains with linguistically plausible grammars that respect common constraints on movement. This establishes not only the cost-free nature of this normal form for realistic grammars, but also that the known restrictions on movement greatly reduce the range of licit movement configurations relative to what unconstrained Minimalist grammars are capable of. Moreover, this work constitutes a first step towards a quantitatively grounded view of movement.

1 Introduction

One of the defining properties of syntax is that phrases do not surface in their base position, also known as the *displacement property*. For example, the topicalized phrase in (1a) acts as an object and thus its base position is to the right of the verb as in (1b).

- (1) a. This guy_i John really hates_i.
- b. John really hates this guy_i.

While displacement phenomena can be analyzed in numerous ways, movement as a generalization of

Chomskyan transformations is the most commonly chosen option.

The last sixty years of generative research have unearthed numerous properties of movement (Ross 1967; Chomsky 1973; Chomsky 1986; Chomsky 2001; Chomsky 2013; Rizzi 1990; Fox and Pesetsky 2005, a.o.), but one of the most central is that movement produces “punctuated paths” (Abels, 2003). This means that phrases do not immediately move to their target position but may temporarily occupy intermediate landing positions on the way there. The nature of punctuated paths was recently studied in Graf et al. (2016), using *Minimalist grammars* (MGs; Stabler 1997; Stabler 2011a) as a formal model of syntax. Graf et al. (2016) prove a *single movement normal form* (SMNF) theorem for MGs. This theorem describes an encoding of MGs where all movement steps of a phrase are triggered by a single feature at the final target position. SMNF has the advantage of reducing the computational complexity of movement dependencies, making it a particularly parsimonious encoding of MGs. At the same time, it comes at the potential cost of a large blow-up in the size of the grammar. Since parsing and learning algorithms tend to scale badly with large grammars, this endangers the viability of SMNF for practical purposes.

In this paper, I show that such a blow-up is primarily observed with grammars that allow unnatural movement patterns — in realistic grammars, the set of licit movement configurations is greatly limited by locality conditions so that SMNF induces only a minimal blow-up. This is a welcome result for real-world applications, but it also points the direction

towards a novel, quantitative perspective on movement. SMNF can distinguish some natural and unnatural movement patterns on quantitative grounds, and it allows us to evaluate constraints on movement in a more global fashion that considers not only their effect on individual derivations but the grammar as a whole.

The paper is laid out as follows: I first give an intuitive introduction to MGs (2.1), how every MG can be translated into SMNF, and why this may increase the size of the grammar (2.2). I also explain in detail why SMNF is not at odds with current syntactic assumptions and what advantages it may provide on a practical, theoretical, and cognitive level (2.3). I then discuss what kind of movement configurations may increase the size of an SMNF grammar (3.1), and how such configurations are blocked by the ban against improper movement (3.2), the ungrammaticality of superraising (3.3), freezing effects (3.4), and the *wh*-island constraint (3.5). This leaves only a handful of constructions as potentially problematic, foremost multiple *wh*-movement and scrambling (3.6). The relevance of these empirical phenomena greatly depends on the choice of syntactic analysis, though, so that the blow-up with SMNF grammars is still likely to be remarkably limited. I conclude with a brief discussion of the linguistic implications of these findings (3.7).

2 Formal Background

2.1 Minimalist Grammars

MGs are a rigorous formalization of Minimalist syntax (Chomsky, 1995). MGs can be decomposed into two components: a set of well-formed derivation trees, and a mapping from those derivation trees to phrase structure trees. Derivation trees are very similar to phrase structure trees, except that interior nodes are labeled *Merge* or *Move* and, more importantly, moving phrases remain in their base position (Fig. 1). The main role of the mapping to phrase structure trees thus is to move phrases into their surface position and insert traces in all intermediate landing sites.

While this general picture is largely sufficient for this paper, SMNF requires a deeper understanding of the MG feature calculus and how it controls movement. Every MG is a finite set *Lex* (the *lexicon*) of

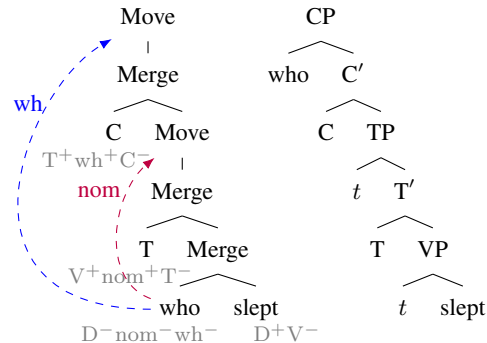


Figure 1: MG derivation tree (left) and corresponding phrase structure tree (right); for the sake of exposition, movement is indicated by arrows

lexical items (LIs), each one of which is annotated with finitely many features. Features come in two polarities, + and -, and they trigger either Merge operations or Move operations. This paper adapts the terminology and notation shown in Tab. 1. An

	Merge	Move
+	selector F^+	licensor f^+
-	category F^-	licensee f^-

Table 1: Terminology and notation for MG feature calculus

MG derivation proceeds by combining LIs until all features have been checked (except the final C^- -feature). Two features f° and g^\bullet ($\circ, \bullet \in \{+, -\}$) can be checked iff $f = g$ and $\circ \neq \bullet$. In addition, the features on LIs have a linear order that encodes the sequence in which they must be checked.

Consider the derivation tree in Fig. 1. It represents a derivation that starts with the LIs $who :: D^- \text{nom}^- \text{wh}^-$ and $slept :: D^+ V^-$. Their first unchecked features are, respectively, D^- and D^+ . Since they have the same feature name but opposite polarity, they can be checked and trigger an application of Merge. The Merge node in the derivation tree is merely a record of this application, the actual result is only seen in the phrase structure tree where *who* and *slept* are combined into a VP.

After feature checking, the remaining feature strings for *who* and *slept* are $\text{nom}^- \text{wh}^-$ and V^- . Since *who* only has licensee features left, it has to wait for matching licensor features to be introduced into the derivation. Until then, *who* is not available for further operations. But *slept* still has a category feature V^- . This feature can be

checked by adding a new LI, the unpronounced T-head $T :: V^+ \text{nom}^+ T^-$. Once again this licenses an application of Merge with a corresponding structure-building step in the phrase structure tree.

All features of *slept* have now been checked, whereas the T-head still carries the feature string $\text{nom}^+ T^-$. Recall that the remaining feature sequence of *who* is $\text{nom}^- \text{wh}^-$. So *who* has been waiting for a licenser feature nom^+ to check its licensee feature nom^- , and this licenser feature is exactly what the T-head provides at this point. Since nom^- and nom^+ are Move features rather than Merge features, their checking triggers an application of Move. Just as with Merge, the derivation tree only contains a record of the Move operation rather than its actual result. In the phrase structure tree, on the other hand, *who* has now been moved from its base position to the specifier of TP. After move, *who* only has the licensee feature wh^- left, and the T-head only has its category feature T^- . At this point, the unpronounced C-head $C :: T^+ \text{wh}^+ C^-$ enters the derivation. The reader is invited to verify for himself that the feature calculus enforces two more feature checking operations that license one instance each of Merge and Move, at which point the only remaining feature is C^- and the derivation can end.

The feature calculus employed in the example above is one of two core mechanisms of all MG derivations. A derivation tree over lexicon *Lex* is well-formed only if all its LIs are members of *Lex* and the features on those LIs can be checked in the manner described above. In addition, the derivation tree must also satisfy the Shortest Move Constraint (SMC): at no point during the derivation are there two distinct LIs with the same licensee feature as their first unchecked feature. Figure 2 shows an example of an SMC-violation. Note that the SMC makes Move a deterministic operation — the feature calculus completely determines which phrase moves where.

The determinism of Move means that the mapping from derivation trees to phrase structure trees is also deterministic. Hence derivation trees already contain all structural information and are MG’s primary data structure for syntax (Kobele et al., 2007; Graf, 2012). The set of well-formed derivation trees, in turn, can be computed in a fully automatic fashion for any given lexicon *Lex* (and therefore any given

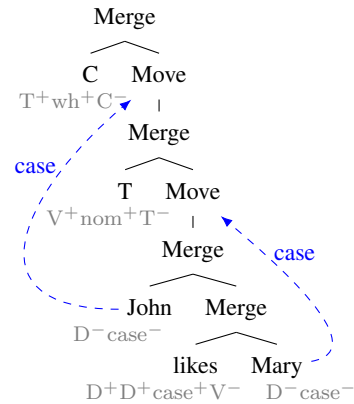


Figure 2: This derivation tree violates the SMC because the case^- features on the subject and the object are both active once the subject has been merged.

MG). In line with this well-established insight, I will also use derivation trees rather than phrase structure trees throughout this paper.

2.2 Single Movement Normal Form

The split between derivation trees and phrase structure trees makes it possible to disentangle movement steps and the feature checking operations that trigger them. In particular, a single feature checking operation can be taken to license multiple movement steps in the phrase structure tree.

This idea was used by Kobele (2006) to solve long-standing problems with successive cyclic movement. Consider a sentence like (2).

- (2) $[_{CP} \text{Who does Bill think } [_{CP} \text{that John thinks } [_{CP} \text{Mary thinks } \dots [_{CP} \text{that Sue likes.}] \dots]]]$

Minimalist syntax posits that the *wh*-phrase *who* starts out as the object of *likes* and then moves through Spec,CP of each embedded clause until it finally reaches its final landing site in Spec,CP of the matrix clause (Chomsky, 1973; McCloskey, 2000; Abels, 2003). This analysis has been criticized because it requires every embedded C-head to carry the feature wh^+ , yet only the matrix clause takes the form of an interrogative. MGs, however, face a much more severe issue: every intermediate landing site requires an extra wh^- to be present on *who*, but this is impossible because an LI can only have finitely many features whereas the number of intermediate landing sites is assumed to be unbounded.

Kobele (2006) addresses both problems by pushing successive cyclic movement out of the feature-controlled derivation trees into the mapping to derivation trees. Under this analysis, *who* carries a single wh^- feature, which is checked by a wh^+ feature on the C-head of the matrix clause. The embedded C-heads do not carry any wh^+ features at all. They are landing sites for the *wh*-phrase not because of some feature checking mechanism, but rather because they occur along the movement path of the *wh*-phrase (Fig. 3).

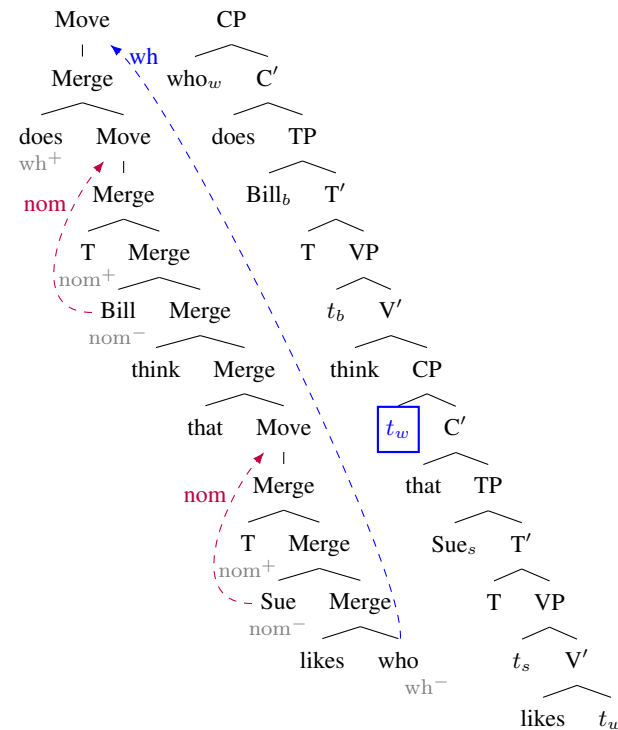


Figure 3: Intermediate landing sites do not need a feature trigger and are instead inserted by the mapping from derivation trees to derived trees. For greater clarity, I indicate only movement features and I highlight the trace that does not directly correspond to any feature-triggered movement operation.

Graf et al. (2016) take this approach and generalize it to a single movement normal form theorem. An MG is in SMNF iff there is no LI in *Lex* with more than one licensee feature. They prove that for every MG G there is an MG G' in SMNF such that G and G' generate the same phrase structure tree language *modulo* intermediate landing sites. In fact, G' can even be coupled with a modified mapping to phrase structure trees that correctly inserts all in-

termediate landing sites, so that G and G' are fully equivalent with respect to the tree languages they generate.

The idea behind SMNF is very simple. As for successive cyclic movement, one deletes all feature triggers for intermediate movement from the derivation. Thus the derivation tree in Fig. 1 becomes the one in Fig. 4.

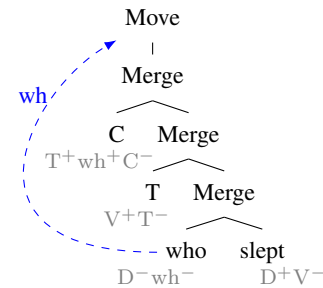


Figure 4: Every phrase moves at most once in an SMNF derivation. Intermediate landing sites can still be inserted by a modified mapping to phrase structure trees.

The SMNF-conversion becomes more complicated, however, if the removal of licensee features would induce an SMC violation. This occurs whenever the derivation contains n LIs ($n \geq 2$) that have the same final licensee feature f^- and whose movement paths overlap at some point in the derivation. In these cases, the only solution is to split f^- into multiple variants $f_1^-, f_2^-, \dots, f_m^-$ ($2 \leq m \leq n$); see Fig. 5 for an abstract example of the relevant configuration.

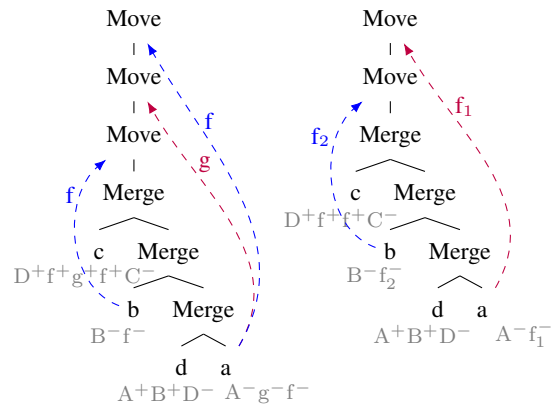


Figure 5: When the derivation tree to the left is brought into SMNF (right), the feature f^- must be split into f_1^- and f_2^- to avoid SMC violations.

Graf et al. (2016) provide an algorithm that keeps

m as small as possible, but the blow-up in Lex may nonetheless be large: for every occurrence of f^- on some LI, m new variants must be added to Lex that only differ in the subscript on f^- . The size of Lex greatly increases as a result. Given an MG G with lexicon Lex , the lexicon size of its SMNF counterpart is linearly bounded by

$$\sum_{l \in Lex} \mu^{\gamma(l)+\delta(l)}$$

where μ is the number of distinct licensee features used by the SMNF grammar, $\gamma(l)$ is the number of licensor features of l , and $\delta(l)$ is 1 if l contains a licensee feature and 0 otherwise.¹

A noticeable blow-up can be observed even with small toy grammars that generate only finite languages:

$$(3) \quad \begin{array}{ll} T :: M^+M^+M^+T^- & C :: T^+C^- \\ a :: M^-a^-f^- & C :: C^+a^+f^+C^- \\ b :: M^-b^-f^- & C :: C^+b^+f^+C^- \\ d :: M^-d^-f^- & C :: C^+d^+f^+C^- \end{array}$$

Conversion to single movement normal form splits each one of a , b , and c into three variants with the feature string $M^-f_i^-$, $1 \leq i \leq 3$. It also replaces the three C -heads with licensor features by C -heads with feature string $C^+f_i^+C^-$. The total size of Lex thus grows from 8 to 14.

The worst-case scenario does not seem to obtain with realistic grammars, however. Multiple LIs in the same derivation rarely have the same f^- as their last feature, and in cases where they do their movement paths do not overlap. Section 3 looks at several cases where attempts to construct such configurations are thwarted by independent constraints on

¹Strictly speaking SMNF induces two kinds of lexical blow-up. The first is due to the refinement of movement features described above. But as described in Graf et al. (2016), the final step of the translation also involves the construction of a bottom-up tree automaton, which is then compiled directly into the category features of the grammar using the algorithm from Graf (2011) and Graf (2013). This final step can induce a blow-up that is polynomial in the size of the automaton. However, one can also incorporate the automaton into the grammar as a constraint definable in monadic second-order logic or simply run it in parallel to the grammar, avoiding the polynomial blow-up of the compilation step. The refinement of movement features, on the other hand, is unavoidable for SMNF MGs. Hence I limit myself to that specific kind of blow-up in this paper.

movement. First, however, a few conceptual remarks on the linguistic plausibility of SMNF are in order.

2.3 Linguistic Plausibility of SMNF

It may seem that SMNF is at best of little relevance to linguistics, and at worst in direct conflict with some of the field's core findings. Numerous empirical arguments have been offered in support of intermediate movement and thus, presumably, against SMNF grammars. However, SMNF does not preclude the existence of intermediate movement, only that such movement is triggered by separate features. To the best of my knowledge there are no widely accepted tests to determine whether certain movement steps are feature-driven, so the status of SMNF is not diminished by current empirical observations.

At the same time, SMNF has several advantages that are of linguistic interest. First of all, the derivation trees of SMNF MG are very similar to dependency graphs, which might be leveraged to develop new learning algorithms for MGs. As dependency parsing has made major inroads in NLP, SMNF is also a first step towards corpus-based MG research.

On a more abstract level, Graf and Heinz (2015) show that SMNF lowers the complexity of Minimalist derivation trees so that they fit into a tree analogue of the subregular string class TSL (Heinz et al., 2011). This class has been found to play a major role in phonology (McMullin, 2016) and morphology (Aks nova et al., 2016; Graf, 2017). SMNF MGs as a model of syntax thus display an unexpected computational parallel to these other language domains, a parallel that disappears with standard MGs where intermediate movement is located in the derivation trees rather than the mapping to phrase structure trees.

There is tentative evidence that SMNF may in fact enjoy some degree of cognitive reality. If one subscribes to the idea that the grammar is not distinct from the parser but both describe the same object at different levels of granularity (cf. Marr 1982 and Neeleman and van de Koot 2010), then one cannot rule out that SMNF MGs are simply yet another level of description, situated between the highly compact grammars with intermediate movement and whatever optimized encoding is used in the

parser. It is interesting in this connection that intermediate movement is handled very differently from final movement in MG parsers (Stabler, 2011b; Stabler, 2013); whereas final movement directly effects the parser’s predictions about the structure of the tree and the order in which it explores these predictions, intermediate movement is just a feature checking step for book keeping purposes. Moreover, recent attempts to use MGs as a model of human sentence processing also see an improvement in empirical coverage when intermediate movement steps are ignored (see e.g. Graf et al. 2017 and Ch. 5 of Zhang 2017). Similarly, Kotek (2017) presents an analysis of wh-intervention effects where at least some instances of intermediate movement must be absent in the syntactic representation.

In sum, SMNF is a useful encoding format of MGs for various practical purposes, and it can also facilitate linguistic research. There are no conclusive empirical arguments at this point to dismiss SMNF at the level of derivation trees, so it is an option worth entertaining. Future work may discover a number of robust arguments for an SMNF-like level of representation, but for the purposes of this paper it suffices that SMNF MGs are linguistically defensible while sporting a few computational advantages.

The major downside of SMNF MGs, on the other hand, is the risk of a blow-up in the number of LIs and hence grammar size. The performance of parsers and learning algorithms depends heavily on grammar size in real-world tasks (because average string length is very low in natural language). Consequently, large SMNF MGs would probably lead to worse performance than their counterparts with intermediate movement. In the next section, however, I argue that large blow-ups are unlikely to arise with realistic grammars because the movement configurations that would cause such a blow-up are disallowed anyways. There is no logical reason as to why languages should be this way, which raises the intriguing possibility that constraints on movement are at least partially motivated by a desire to keep grammars small, compact, and redundancy-free. The paper does not provide a conclusive answer to this larger issue and focuses instead on establishing the low risk of lexical blow-up with SMNF MGs. But in doing so it provides a glimpse of how these linguistic questions could be addressed in follow-up work.

3 Constraints and Grammar Size

3.1 General Observations

As discussed in Sec. 2.2, SMNF increases the size of the lexicon whenever removal of intermediate landing sites would induce SMC violations in some derivations. In this case, a licensee feature has to be split into multiple subscripted variants. This still does not necessarily induce a blow-up, though.

Suppose that the toy grammar in (3) only had a single C-head $C :: T^+ a^+ f^+ b^+ f^+ d^+ f^+ C^-$. In contrast to (3) this now fixes the surface order of a, b, and c with respect to each other. Then the corresponding SMNF grammar would just have a C-head $C :: T^+ f_1^+ f_2^+ f_3^+ C^-$, and the LIs a, b, and c could be replaced by $a :: M^- f_1^-$, $b :: M^- f_2^-$, and $c :: M^- f_3^-$. Since the relative movement configurations of a, b, and c are fixed across all derivations, the feature refinement does not cause a multiplication of LIs.

Therefore the configurations where SMNF necessarily causes a blow-up involve multiple LIs that

1. have the same final licensee feature, and
2. have overlapping movement paths in some derivations, and
3. are flexible in the sense that the arrangement of the overlapping movement paths varies across derivations.

Let us briefly reflect on what kind of movement patterns in natural languages could possibly fit this description.

If one assumes that the different case positions for A-movement are associated with different features (nom, acc, and so on), and that every clause provides exactly one position for each type of case (Spec,TP, Spec,vP, and so on), then the first two conditions above cannot be met unless an LI A-moves to a case position outside its own clause. But this kind of movement is known to be heavily constrained, so overlapping paths are hard to construct.

With A'-movement, movement across clause boundaries is much less restricted, as it witnessed by unbounded wh-movement and topicalization. However, A'-movement is still subject to various principles that penalize overlapping movement paths. This again makes it difficult to design well-formed case of A'-movement that are problematic with SMNF.

The next few sections give concrete examples of how the desired movement configurations violate various well-known constraints on movement.

3.2 Improper movement

One option to create overlapping A-movement paths for LIs l and l' is to have l' first A'-move over the A-mover l , followed by l' undergoing the same kind of A-movement as l . But this requires l' to A-move from an A'-position, which is forbidden by the Ban on Improper Movement, illustrated in (4).

- (4) a. John wonders who_{wh} Bill $_{nom}$ t_{nom} saw t_{wh} . (Proper movement)
 b. * $who_{wh,nom}$ wonders t_{nom} Bill $_{nom}$ t_{nom} saw t_{wh} . (Improper movement)

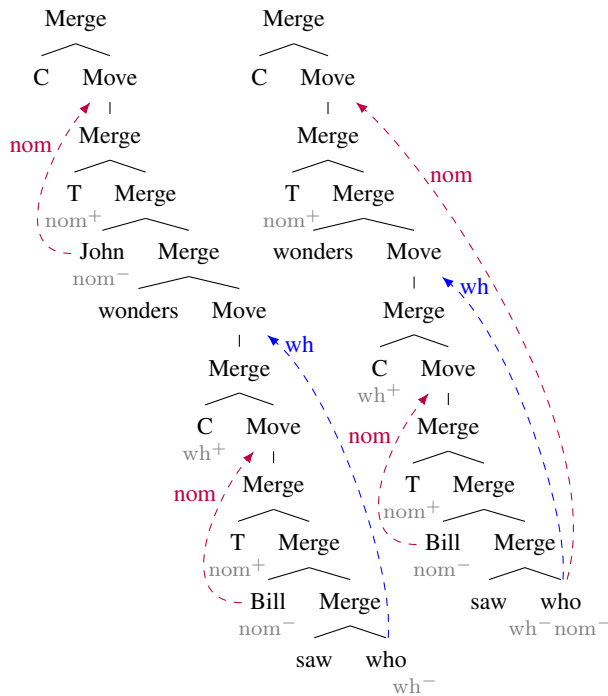


Figure 6: Since standard MGs do not impose any movement constraints besides the SMC, one can construct derivations like the one to the right that violates the ban on improper movement.

A standard MG with intermediate movement has no problem generating both the licit (4a) and the illicit (4b), as is shown in Fig. 6. Suppose for the sake of argument that (4b) were actually well-formed in English, so that the grammar would actually have to allow movement configurations like in Fig. 6. Then in the corresponding SMNF MG, the only licensee

feature of *who* would be nom^- , triggering direct movement from the object position to the subject position of the matrix clause. But since *Bill* also has nom^- as its last licensee feature and the two phrases have overlapping paths, this would trigger an SMC violation. Hence the SMNF MG must refine nom^- into two features nom_1^- and nom_2^- .

- (5) a. T_{nom} wonders T_{nom} Bill $_{nom}$ saw who_{nom} . (SMC violation)
 b. T_{nom2} wonders T_{nom1} Bill $_{nom1}$ saw who_{nom2} . (SMNF derivation tree)
 c. who_{nom2} wonders Bill $_{nom1}$ t_{nom1} saw t_{nom2} . (SMNF phrase structure tree)

Once nom^- has been replaced by nom_1^- and nom_2^- , some LIs need to be duplicated. Every LI with nom^+ now is split into two variants as it may serve as the landing site for a nom_1^- mover (standard subjects) or a nom_2^- mover (improperly moving wh-phrases). In addition, verbs like *wonder* that select a subject and a CP object now also have a variant that only selects an object CP. This is the result of the improper mover assuming the subject role for the clause containing *wonder*.

For the actual movers, however, duplication can be avoided. While the feature nom^- must be refined into nom_1^- and nom_2^- , each type of licensee feature occurs with a specific type of phrase. The licensee feature nom_2^- is limited to improper movers, which are wh-phrases. The licensee feature nom_1^- , on the other hand, occurs on DP that does not move improperly to a subject position. Note that the latter cannot be wh-movers: I) if a DP wh-moves before undergoing subject movement, it violates the ban against improper movement, contrary to our initial assumption; II) if a DP wh-moves after undergoing subject movement, then its final feature is not nom^- so that neither nom_1^- nor nom_2^- would be presented on the DP's counterpart in an SMNF MG.

All these points jointly imply that nom^- is replaced by nom_1^- for non-wh subjects and by nom_2^- for wh-phrases whose final landing site is a subject position. Overall, then, the lexical blow-up is not as large as one would initially suspect. But the size of an SMNF grammar still increases by $2 \times |T| + 2 \times |V|$, where $|T|$ is the number of LIs that carry nom^+ and $|V|$ indicates the number of LIs that select both a subject and a CP object.

3.3 Superraising

Another option to A-move a DP out of a finite clause is superraising, which looks very similar to improper movement from the perspective of SMNF MGs.

Prototypical cases of superraising are irrelevant for SMNF because they only involve a single mover.

- (6) a. John seems [TP *t* to [VP *t* like Mary.]]
 b. * John seems [TP *t* likes [VP Mary.]]

However, analogous cases with two movers can be imagined.

- (7) a. It seems Bill believes John likes Mary.
 b. * John seems [CP Bill believes *t* likes Mary.] (Superraising)

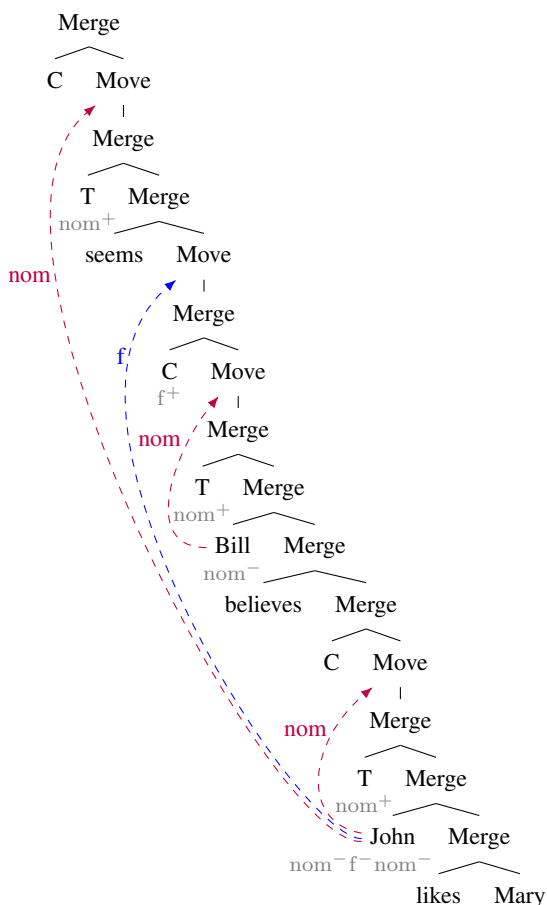


Figure 7: Standard MGs also allow for superraising configurations, the derivations of which look very similar to the improper movement derivations.

Again an MG analysis of (7b) is readily available. Due to the SMC, *John* cannot directly undergo

nom-movement to the matrix clause because *Bill* is also undergoing nom-movement. Instead, some feature f^- first moves *John* over *Bill* into Spec,CP of the highest embedded clause, from where *John* then moves on to the matrix subject position via nom^- . The role of f^- is almost identical to wh^- in the improper movement case in Fig .6, except that f^- has no independent motivation beyond avoiding an SMC violation.

In a SMNF grammar, *John* would only possess the licensee feature nom^- . But *Bill* carries the same feature, so that once again nom^- would have to be split into multiple features. As before, though, we can tie specific variants of nom^- to specific LIs: nom_1^- is chosen for normal subject movers, whereas nom_2^- is for superraising subject movers, identified by their peculiar feature string $nom^-f^-nom^-$. We also preserve the usual split into nom_1^+ and nom_2^+ for LIs that carry nom^+ , so that the SMNF-induced blow-up is $2 \times |T|$.

However, banning superraising configurations not only avoids this size increase, but shrinks the size of the lexicon even in comparison to the original MG. Where the original MG contained the LIs *John* :: D^-nom^- and *John* :: $D^-nom^-f^-nom^-$, the SMNF MG contains *John* :: $D^-nom_1^-$ and *John* :: $D^-nom_2^-$, but the SMNF MG without superraising only needs *John* :: D^-nom^- . In other words, banning superraising cuts the number of subject movers listed in the lexicon in half.

3.4 Freezing Effects

One more attempt at using A-movement to induce lexical blow-up with SMNF centers around A-moving a phrase out of another A-moving phrase.

- (8) a. It seems your comment about John annoys Sue.
 b. * John seems your comment about *t* annoys Sue. (Freezing effect)

Here *your comment about John* undergoes nom-movement before *John* nom-moves to the matrix subject position (presumably preceded by some kind of *f*-movement to avoid an SMC violation). The ungrammaticality of such configurations is known as *freezing effects*, but as far as the SMNF conversion is concerned the logic works exactly as for superraising (and hence improper movement).

That said, freezing effects are more general in that they also pertain to A'-movement as in the example below.

- (9) * Who don't you know [which pictures of *t*] Mary bought. (Freezing effect)

Given standard Minimalist assumptions the sentence above would be generated by *which pictures of who* moving from the object position to Spec,CP, at which point *who* undergoes wh-movement to Spec,CP of the matrix clause. For MGs, the need to avoid an SMC violation would complicate the derivation somewhat in that *who* first *f*-moves to Spec,CP and then wh-moves to the next higher specifier. Nonetheless we would once again see SMNF induce a split for wh-movers into wh_1^- and wh_2^- , with one for normal wh-movement and one for wh-movement out of a freezing configuration. As for superraising, then, a SMNF MG that obeys freezing effects not only avoids a blow-up in grammar size but even sees a decrease compared to standard MGs.

3.5 Wh-Islands

Freezing effects make it impossible to create overlapping wh-movement paths by extracting a wh-mover from a wh-mover. But even if the wh-movers are independent of each other, overlapping movement paths are difficult to produce, due to the wh-island constraint. Consider first the simple paradigm below.

- (10) a. What_{wh} did John say Mary gave *t_{wh}* to Bill?
 b. * What_{wh0} did John say who_{wh1} Mary gave *t_{wh0}* to *t_{wh1}*? (Wh-island violation)

As before, sentences like (10b) would require refinement of the licensee and licensor features and thus increase grammar size. But in contrast to the previous constructions, wh-island violations easily allow for much more elaborate patterns that require even more indices. With each index, the SMNF grammar would gain more and more LIs.

- (11) a. * What_{wh0} did Bill think which_{wh1} man *t_{wh1}* says who_{wh1} Mary gave *t_{wh0}* to *t_{wh1}*?

- b. * What_{wh0} did Sue claim who_{wh1} Bill thinks which_{wh2} man *t_{wh2}* says Mary gave *t_{wh0}* to *t_{wh1}*?

3.6 Potential Cases of Blow-Up

The list of examples here is not exhaustive, but a wider sampling of the literature still confirms the general tendency that potential sources of lexical blow-up are blocked for independent reasons. One notable exception is the availability of superraising in some languages like Standard Arabic, but it is unclear whether this involves multiple movers of the same type (Ura, 2007).

Besides that, there is one more class of constructions that seem to be problematic. Like the toy grammar in (3), they involve configurations where many movers are drawn to the same target position. This is the case for multiple wh-movement, quantifier raising under certain movement-based analyses (May 1977; May 1985), and scrambling. However, all of them are already known to be problematic for standard MGs anyways, and alternative analyses have been proposed at least for the former two (Gärtner and Michaelis, 2010). With these analyses, SMNF is entirely unproblematic. Scrambling, on the other hand, is known to differ from other movement types in several respects and is also computationally challenging (Becker et al., 1996; Joshi et al., 2000). Therefore a movement-based account may not be the best choice (see also Frey and Gärtner 2002). Overall, then, the few empirical cases that are problematic with SMNF MGs may not be adequately analyzed in terms of standard movement anyways.

Of course this does not guarantee that a wide-coverage MG can be safely translated into SMNF without a significant increase in grammar size. The constructions surveyed for this paper are noteworthy in that they only looked at movement with a clear function — wh-movement and case movement. These features are distributed in a principled manner and are usually tied to specific functional heads. But MGs, just like the Minimalist literature they are modeled after, also posit more abstract features like *f*, *g*, *h* whose only purpose is to produce the observed surface order. These features were completely ignored in this paper because it seems unlikely that manual analysis can reveal much about them. Instead, it seems more promising to run sim-

ulations where realistic MGs are automatically converted to SMNF. While such realistic MGs are currently being worked on (Torr, 2017), they are still in a highly preliminary stage. Hopefully some simulations will be feasible in the near future. If these features should turn out to be problematic for SMNF, this might suggest that there is a real difference between “functionally grounded” kinds of movement and the more stipulative word-order movement.

3.7 Linguistic Evaluation

The central claim of this paper is that SMNF is relatively safe for realistic grammars because most of the configurations that might induce a large blow-up in grammar size are ungrammatical anyways. It is tempting to couple this descriptive observation with a more speculative linguistic proposal: constraints on movement are (at least partially) motivated by the desire to keep grammars small and compact. This would be similar to a minimum-description-length approach and would provide a third-factor explanation (Chomsky, 2005) that could tie together seemingly unrelated phenomena such as freezing effects, wh-islands, superiority conditions, and the ban against improper movement. As we have already seen, however, these constraints are usually more general than necessary for grammar succinctness. Improper movement is a problem for SMNF only if the A-movement is of a type that was crossed by the A'-movement, superraising is fine unless one subject-raises over another subject, and so on. Hence one must not be too eager, at this point, to posit a causal link between grammar compactness and movement constraints. That said, the possibility is certainly intriguing, and the SMNF-perspective will be useful in exploring the idea to its fullest.

4 Conclusion

Single movement normal form is a useful encoding of MGs that greatly simplifies mathematical proofs and has potential applications in the design of new parsers and learning algorithms for MGs. While it comes with the risk of greatly increasing the size of the grammar, this seems to be less of an issue with natural languages because movement has to obey numerous constraints that greatly limit the set of possible movement configurations. In particular, it

is very difficult for two lexical items to have overlapping movement paths while also sharing final landing sites of the same type. From an applied perspective, this result increases the confidence in SMNF as a useful MG encoding. But there is also linguistic potential: studying movement configurations with respect to their effect on grammar size may unearth entirely new generalizations about natural language.

References

- Klaus Abels. 2003. *Successive Cyclicity, Anti-locality, and Adposition Stranding*. Ph.D. thesis, University of Connecticut.
- Alëna Aksënova, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 121–130.
- Tilman Becker, Owen Rambow, and Michael Niv. 1996. The derivational generative power of formal systems, or scrambling is beyond LCFRS. Technical Report IRCS 92-38, Institute for Research in Cognitive Science, University of Pennsylvania.
- Noam Chomsky. 1973. Conditions on transformations. In Stephen Anderson and Paul Kiparsky, editors, *A Festschrift for Morris Halle*, pages 232–286. Holt, Rinehart, and Winston, New York.
- Noam Chomsky. 1986. *Barriers*. MIT Press, Cambridge, MA.
- Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.
- Noam Chomsky. 2001. Derivation by phase. In Michael J. Kenstowicz, editor, *Ken Hale: A Life in Language*, pages 1–52. MIT Press, Cambridge, MA.
- Noam Chomsky. 2005. Three factors in language design. *Linguistic Inquiry*, 36(1):1–22.
- Noam Chomsky. 2013. Problems of projection. *Lingua*, 130:33–49.
- Danny Fox and David Pesetsky. 2005. Cyclic linearization of syntactic structure. *Theoretical Linguistics*, 31:1–45.
- Werner Frey and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in Minimalist grammars. In Gerhard Jäger, Paola Monachesi, Gerald Penn, and Shuly Wintner, editors, *Proceedings of the Conference on Formal Grammar*, pages 41–52.
- Hans-Martin Gärtner and Jens Michaelis. 2010. On the treatment of multiple-wh-interrogatives in Minimalist grammars. In Thomas Hanneforth and Gisbert Fanselow, editors, *Language and Logos*, pages 339–366. Akademie Verlag, Berlin.

- Thomas Graf and Jeffrey Heinz. 2015. Commonality in disparity: The computational view of syntax and phonology. Slides of a talk given at GLOW 2015, April 18, Paris, France.
- Thomas Graf, Alëna Aksënova, and Aniello De Santo. 2016. A single movement normal form for Minimalist grammars. In Annie Foret, Glyn Morrill, Reinhard Muskens, Rainer Osswald, and Sylvain Pogodalla, editors, *Formal Grammar : 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016*, pages 200–215, Berlin, Heidelberg, Springer.
- Thomas Graf, James Monette, and Chong Zhang. 2017. Relative clauses as a benchmark for Minimalist parsing. *Journal of Language Modelling*, 5:57–106.
- Thomas Graf. 2011. Closure properties of Minimalist derivation tree languages. In Sylvain Pogodalla and Jean-Philippe Prost, editors, *LACL 2011*, volume 6736 of *Lecture Notes in Artificial Intelligence*, pages 96–111, Heidelberg, Springer.
- Thomas Graf. 2012. Locality and the complexity of Minimalist derivation tree languages. In Philippe de Groot and Mark-Jan Nederhof, editors, *Formal Grammar 2010/2011*, volume 7395 of *Lecture Notes in Computer Science*, pages 208–227, Heidelberg, Springer.
- Thomas Graf. 2013. *Local and Transderivational Constraints in Syntax and Semantics*. Ph.D. thesis, UCLA.
- Thomas Graf. 2017. Graph transductions and typological gaps in morphological paradigms. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 114–126.
- Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.
- Aravind Joshi, Tilman Becker, and Owen Rambow. 2000. Complexity of scrambling: A new twist to the competence-performance distinction. In Anne Abeill and Owen Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing*, pages 167–181. CSLI, Stanford.
- Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to Minimalism. In James Rogers and Stephan Kepser, editors, *Model Theoretic Syntax at 10*, pages 71–80.
- Gregory M. Kobele. 2006. *Generating Copies: An Investigation into Structural Identity in Language and Grammar*. Ph.D. thesis, UCLA.
- Hadas Kotek. 2017. Movement and alternatives don't mix: A new look at wh-intervention effects. Handout for an invited talk presented on October 20 at Stony Brook University, Stony Brook, NY.
- David Marr. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman, New York. Reprinted in 2010 by MIT Press.
- Robert May. 1977. *The Grammar of Quantification*. Ph.D. thesis, MIT.
- Robert May. 1985. *Logical Form: Its Structure and Derivation*. MIT Press, Cambridge, MA.
- James McCloskey. 2000. Quantifier float and wh-movement in an Irish English. *Linguistic Inquiry*, 31:57–84.
- Kevin McMullin. 2016. *Tier-Based Locality in Long-Distance Phonotactics: Learnability and Typology*. Ph.D. thesis, University of British Columbia.
- Ad Neeleman and Hans van de Koot. 2010. Theoretical validity and psychological reality of grammatical code. In Martin Everaert, Tom Lentz, Hannah de Mulder, Oystein Nilsen, and Arjen Zondervan, editors, *The Linguistic Enterprise: From Knowledge of Language to Knowledge of Linguistics*, pages 183–212. John Benjamins, Amsterdam.
- Luigi Rizzi. 1990. *Relativized Minimality*. MIT Press, Cambridge, MA.
- John R. Ross. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT.
- Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.
- Edward P. Stabler. 2011a. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.
- Edward P. Stabler. 2011b. Top-down recognizers for MCFGs and MGs. In *Proceedings of the 2011 Workshop on Cognitive Modeling and Computational Linguistics*. to appear.
- Edward P. Stabler. 2013. Two models of minimalist, incremental syntactic analysis. *Topics in Cognitive Science*, 5:611–633.
- John Torr. 2017. Autobank: a semi-automatic annotation tool for developing deep Minimalist grammar treebanks. In *Proceedings of the Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 81–86.
- Hiroyuki Ura. 2007. Returning to superraising. *Departmental Bulletin Paper*, 56:59–74.
- Chong Zhang. 2017. *Stacked Relatives: Their Structure, Processing, and Computation*. Ph.D. thesis, Stony Brook University.