 Theoretical Linguistics 2022; 48(3–4): 245–278

Thomas Graf*

Diving deeper into subregular syntax

https://doi.org/10.1515/tl-2022-2043

1 Introduction

In my target paper *Subregular linguistics: Bridging theoretical linguistics and formal grammar* I argued that theoretical linguistics and formal grammar, despite major differences in methodology, have a lot to offer to each other, and that the recently emerged subfield of *subregular linguistics* provides a conduit for knowledge transfer between the two. As concrete examples of this, I discussed prior findings that suggest a surprising degree of computational parallelism between phonology and syntax. This computational/cognitive parallelism may be stated in two ways. First, phonology, morphology, and syntax (*modulo* conditions at the syntax-semantics interface) share a robust upper bound on complexity in the sense that all their constraints and operations can be defined in first-order logic. Second, the overwhelming majority of phenomena within each one of those domains seems to fall into the highly restricted classes of *strictly local* (SL) and *tier-based strictly local* (TSL) dependencies.

As part of this discussion, I illustrated how movement can be conceived of as a local dependency over tree tiers and that this also provides a new perspective on islands as blockers on a movement tier. From a computational perspective, this makes the existence of islands unsurprising: if movement already involves mechanisms of complexity *C*, and adding islands does not push us beyond *C*, then the fact that a cognitive system with the mental resources for movement would also exhibit island effects is no more surprising than the fact that the object on the cover of a LEGO box is just one of many different objects that can be built with those bricks. If the cognitive resources allow for phenomenon *P*, it is the cross-linguistic absence of *P* that requires an explanation, not its existence. The cognitive parallelism hypothesis and this perspective on islands constitute two specific cases where subregular complexity provides surprising new insights into the nature of language, but the prior accomplishments and future potential of subregular linguistics extend far beyond that.

Three of the five commentaries substantiate this point. Chandlee broadens my abridged presentation of subregular phonology and subregular morphology into a highly accessible, empirically grounded overview of the major findings in these two areas of subregular linguistics, emphasizing in particular the relevance of mappings

***Corresponding author: Thomas Graf**, Department of Linguistics, Stony Brook University, Stony Brook, NY, USA, E-mail: mail@thomasgraf.net

from underlying representations to surface forms and recent work on the effects of representational choices. Avcu & Rhodes discuss how formal grammar in general and subregular linguistics in particular enable researchers to connect the findings of theoretical linguistics to questions of psycho- and neurolinguistics, for example via artificial language learning experiments. Himmelreich uses the tier-based view of syntax to explore what focus marking in Likpakpaanl might tell us about the syntactic status of functional elements, in particular with respect to adjuncts. This is an impressive demonstration of the analytic potential of subregular linguistics for empirical work in linguistics. It is a testament to subregular linguistics that in less than 15 years it has accumulated a body of knowledge and techniques that is too vast to even mention all relevant facets in a single paper, and I thank the authors for greatly broadening the scope of my initial contribution in support of subregular linguistics.

The commentaries by Brody and Chaves & Putnam take a more critical perspective, and while they approach subregular linguistics from very different angles, the general thrust of their arguments is very similar. Both question the value of formal restrictions on syntactic machinery, and both are unconvinced by the argument that TSL provides new insights on island effects, in particular adjunct island effects. Brody also contends that TSL syntax still has to stipulate tree structures for syntax when a purely string-based approach built on path languages might suffice. While I do not agree with these remarks, I am grateful for the opportunity to clarify some important points about the status of subregular claims and why this makes many of the concerns expressed by Brody and Chaves & Putnam inapplicable.

I thank all the authors for their thought-provoking commentaries that greatly enrich the discussion that I hoped to start with the initial paper. A single reply cannot do justice to the many points that have been raised, and hence I will focus on some shared themes revolving around subregular syntax: the role of mappings (Section 2), with Irish wh-agreement as a concrete example; the challenges of linearization (Section 3), with a new *Generalized Ban on Improper Movement* derived from subregular limits of linearization; the importance of careful empirical analysis (Section 4), including a brief discussion of existing work on gradient TSL and its empirical applications; and finally, what exactly it means to say that a specific phenomenon falls within a subregular class like TSL, and why this is orthogonal to the issue of formalisms and metalanguages (Section 5). The Appendix expands on this final point by discussing an alternative specification of TSL over trees that does not use trees at all.

2 Mappings

Mappings feature prominently in Chandlee's commentary, which gives an accessible overview of the many advances subregular linguistics has made in the domain of phonology and morphology over the last 15 years. In phonology and morphology, it is commonly assumed that each surface form is obtained from some underlying representation (UR)—either via SPE-style rewrite rules, or via a generate-and-filter approach as in OT. Mathematically, this is modeled as a sequence of mappings from an input form to some output form, and as Chandlee explains in detail (pp. 206–212), there is ample evidence that these mappings are subject to strong computational limitations. So far, there are no comparable results for subregular syntax. Chandlee thus asks (p. 215):

> Results from the study of string-to-string mappings have spanned a wider range of the hierarchy compared to phonotactics, from ISL/OSL […], to TSL […], to subsequential […], to non-deterministic regular functions […]. Are there likewise applications of these classes to syntactic phenomena and if so should we expect to see a similar range of results?

Whether the cognitive parallelism between phonology and syntax extends to mappings is an interesting issue, and one that ongoing work in subregular syntax has some light to shed on.

In the following, I will briefly sketch why extraction morphology, e.g. wh-agreement in Irish, can be regarded as an analogue of harmony processes in phonology (Section 2.1). But syntax also differs from phonology in a crucial respect: because syntactic URs are the output of a computational system, many problems that look like mapping problems can be insightfully studied as constraints on syntactic URs (Section 2.2). As a result syntax can be studied almost completely without mappings, and hence it is not clear how much of the subregular work on mappings in phonology we should expect to carry over to syntax.

However, it is indisputable that there is at least one mapping that applies to syntactic URs, and that is the linearization mapping that translates syntactic derivations into pronounceable, fully linearized strings, and as I will argue later in Section 3, this mapping seems to be subject to subregular constraints that derive (a generalized version of) the Ban on Improper Movement.

2.1 Irish wh-agreement as a TSL mapping

Extraction morphology (which I take to include wh-agreement, cf. Zentz 2015, fn. 3) refers to cases where the morphological make-up of a lexical item depends on whether it occurs along a movement path. The best-known instance of this is wh-agreement in Irish, where complementizers along a wh-movement path have a special form that is distinct from standard complementizers (cf. McCloskey 2001, p. 94, example (ii)):

(1)	Cé	a/*go	dúradh	léithi	a/*go	cheannódh	é?
	who	C-wh/C	was-said	with-her	C-wh/C	would-buy	it
	‘Who was she told would buy it?’						

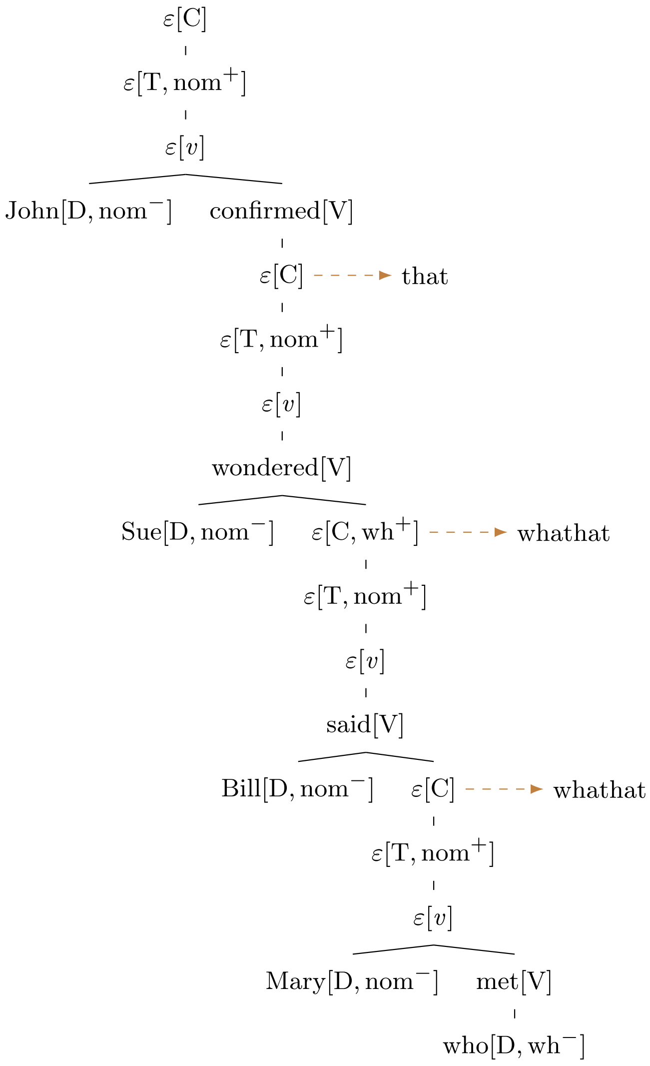
One may reasonably construe this as a syntactic mapping phenomenon: the output of every C-head shows a principled alternation that is contingent on whether the C-head occurs along a wh-movement path.

In order to keep the discussion focused, let us make two simplifying assumptions. First, we will assume a fictitious dialect of English that works like Irish in that a C-head is spelled out as the special wh-agreement form *whathat* iff the C-head occurs along a wh-movement path. This will allow us to focus just on the phenomenon of extraction morphology without getting bogged down by the syntactic details of Irish. In this fictitious dialect of English, the following acceptability judgments would hold.

(2) John confirmed that/*whathat Sue wondered who *that/whathat Bill said *that/whathat Mary met.

The dependency tree for this sentence is given in (3) with dashed arrows indicating how each embedded C-head is to be spelled out.

(3)



Now let us simplify things even further by taking a hint from Brody (2019) and consider only the ancestor string of the wh-mover *who*. This is the string of nodes in the dependency tree that reflexively dominate *who*.

(4) **Ancestor string of *who* in (3)**
who[D, wh$^-$] met[V] ε[v] ε[T, nom$^+$] ε[C] said[V] ε[v] ε[T, nom$^+$] ε[C, wh$^+$] wondered[V] ε[v] ε[T, nom$^+$] ε[C] confirmed[V] ε[v] ε[T, nom$^+$] ε[C]

These two simplifications have no bearing on the ensuing discussion as the general point holds just as well with the trees one might posit for Irish, except that the syntactic analysis and the definitions of the relevant tree mappings are more involved.

Given these simplifying assumption, let us consider the problem of determining the correct spell-out forms for the embedded C-heads. The string has to be rewritten so that each C-head between *who* and its landing site, i.e. the closest node with wh$^+$, is spelled out as *whathat*. This can be done with the kind of TSL function (Burness et al. 2021, and references therein) that Chandlee mentions in her discussion of Karajá ATR harmony (p. 207). While processing the string from left to right, we project all nodes carrying wh$^-$ or wh$^+$. If we encounter an embedded C-head while the most recent symbol on the tier carries wh$^-$, we rewrite the embedded C-head as the wh-agreeing *whathat*, and as the non-agreeing *that* otherwise.

(5) **TSL rewriting of the ancestor string in (4)**

Tier	who[D, wh$^-$]					
Input	who[D, wh$^-$]	met[V]	ε[v]	ε[T, nom$^+$]	ε[C]	said[V]
Output	who[D, wh$^-$]	met[V]	ε[v]	ε[T, nom$^+$]	whathat[C]	said[V]

Tier [cont.]			ε[C, wh$^+$]		
Input [cont.]	ε[v]	ε[T, nom$^+$]	ε[C, wh$^+$]	wondered[V]	ε[v]
Output [cont.]	ε[v]	ε[T, nom$^+$]	whathat[C, wh$^+$]	wondered[V]	ε[v]

Tier [cont.]						
Input [cont.]	ε[T, nom$^+$]	ε[C]	confirmed[V]	ε[v]	ε[T, nom$^+$]	ε[C]
Output [cont.]	ε[T, nom$^+$]	that[C]	confirmed[V]	ε[v]	ε[T, nom$^+$]	ε[C]

Extraction morphology phenomena such as Irish wh-agreement, which have attracted a lot of interest from syntacticians, thus are closely related to the phonological phenomena that Burness et al. (2021) analyze with TSL functions, e.g. Turkish backness vowel harmony, parasitic harmony in the Kachin dialect of

Khakass, Khalkha Mongolian rounding harmony with blocking, and Slovenian sibilant harmony with blocking.

Moreover, the final landing site with wh^+ corresponds to *icy targets* in phonology (Jurgec 2011). Icy targets are segments that participate in a given process but also block further application of said process—just like the head with wh^+ undergoes wh-agreement while also blocking further spreading of the wh-agreement to higher complementizers. Note that this special behavior of the final landing site is a property of the specific TSL function we defined, other TSL functions can handle things differently. In particular, the C-head that provides the final landing site for wh-movement differs from all intervening C-heads in that it carries wh^+, and as a result it can be treated differently by TSL functions. Georgi (2017) surveys extraction morphology phenomena across a number of typologically diverse languages, and she notes that there are four attested patterns based on two parameters: whether the intermediate targets (i.e. C-heads without wh^+ in our analysis) display extraction agreement, and whether the final landing site displays extraction agreement. This four-way split in the typology is perfectly expected under the TSL mapping analysis.

In sum, extraction morphology may be construed as a mapping from underspecified forms to fully inflected ones. This mapping is a TSL function in the sense of Burness et al. (2021) (see also Burness and McMullin 2019; Hao and Andersson 2019; Hao and Bowers 2019). The discussion above simplified things by defining the mapping over an ancestor string instead of directly over the tree, but nothing hinges on that. One can define tree-analogues of TSL functions, and extraction morphology could equally be modeled by such tree-TSL functions. Hence there are at least some aspects of syntax that can be studied through the lens of mappings, and they turn out to be very similar to well-attested phenomena in phonology.

2.2 Irish wh-agreement as a TSL constraint

The mapping analysis above presupposes that the spell-out of a complementizer is determined in some post-syntactic step, mirroring proposals in Distributed Morphology and Nanosyntax. But early Minimalism (Chomsky 1995) was very explicit that lexical items are assumed to have fully inflected output forms in syntax, and presumably this includes inflections due to extraction morphology. If we take the stance that all lexical items enter syntax fully inflected, then extraction morphology reduces from a mapping problem to a constraint on the distribution of fully inflected lexical items.

In the concrete case of wh-agreement, this means that our fictitious dialect of English would have a fully inflected C-head *whathat* that may only appear along movement paths. This is a very simple TSL condition over ancestor strings: we still project all lexical items carrying wh^- and wh^+, but in addition we also project all embedded C-heads (or alternatively, all pronounced embedded C-heads if the dialect allows for empty C-heads along wh-movement paths).

(6) **Wh-agreement as a constraint on inflected ancestor strings**

Tier	who[D, wh^-]				whathat[C]	
	|				|	
Input	who[D, wh^-]	met[V]	$\varepsilon[v]$	ε[T, nom^+]	whathat[C]	said[V]

Tier [cont.]			whathat[C, wh^+]		
			|		
Input [cont.]	$\varepsilon[v]$	ε[T, nom^+]	whathat[C, wh^+]	wondered[V]	$\varepsilon[v]$

Tier [cont.]		that[C]				
		|				
Input [cont.]	ε[T, nom^+]	that[C]	confirmed[V]	$\varepsilon[v]$	ε[T, nom^+]	ε[C]

In order for a tier to be well-formed, it may not contain any of the following forbidden bigrams.

(7) a. $\rtimes$*whathat* (no wh-agreement before movement starts), and
b. wh^+*whathat* (no wh-agreement after movement is complete), and
c. wh^-*that* (no non-agreeing C-heads after wh-movement has started), and
d. *that whathat* and *whathat that* (no alternation of agreeing and non-agreeing C-heads).

The well-formed string in (6) satisfies all these conditions, whereas replacing one of the embedded C-heads with another pronounced form would immediately yield an illicit string.

Again the typological variation observed by Georgi (2017) is easy to capture. Whether the final landing site displays wh-agreement hinges on whether C-heads with wh^+ are specified with an agreeing or a non-agreeing form in the lexicon. Whether other C-heads along the movement path display agreement is contingent on whether they are projected onto the tier. And once again extraction morphology looks very similar to attested harmony processes when they are construed as phonotactic constraints.

At least with respect to extraction morphology, then, it is difficult to say whether syntactic phenomena are better analyzed in terms of constraints or mappings. Note that this is not an accidental consequence of our use of strings instead of trees. Graf (2022) gives an analysis of wh-agreement as a constraint on tree tiers, and the central ideas are the same as in the simplified string-based version above.

To some extent, it is not surprising that one can freely switch between mappings and constraints because syntacticians have been wrestling with this very issue for a long time, in particular with respect to binding and morphological agreement. When syntactic binding is construed as a set of constraints on the distribution of morphological forms of pronominals, one can formalize this either as a mapping that spells out an underlying *pro* as a pronoun or a reflexive, or as a set of constraints that regulate where forms like *her* and *herself* may occur in the syntactic structure. Similarly, morphosyntax can be taken to regulate the distribution of fully inflected forms, or to describe how underspecified forms are spelled out as fully inflected ones. Even the *that*-trace constraint can be reanalyzed as a spell-out rule that must leave C-heads unpronounced in certain structural contexts.

To the extent that there is any general consensus among syntacticians how the work should be split between constraints and mappings, it tends to be short-lived. As mentioned before, early Minimalism as defined in Chomsky (1995) assumed fully inflected lexical items, whereas the introduction of Agree in Chomsky (1998) prompted a shift to underspecified lexical items. But usually this is limited to ϕ-features and only a few proposals such as Heinat (2006) have argued to extend underspecification to binding. With so much uncertainty surrounding the role of constraints and mappings, it make sense that subregular linguistics so far has taken the path of least resistance, favoring constraints to avoid the more challenging mathematics of mappings.

Still, I concur with Chandlee regarding the importance of studying syntactic mappings. Having two formal characterizations of the same phenomenon provides a deeper understanding than clinging to just one. By studying both perspectives, we may discover computational reasons to favor constraints over mappings or the other way round. For cases where syntactic mappings deviate greatly from what we find in phonology, a constraint-based analysis might be preferable. Subregular syntax thus has the potential to shed light on issues that are at the very core of Distributed Morphology and related approaches.

3 Linearization as a mapping

Even though it is not always clear in syntax how the workload should be distributed between mappings and constraints, one aspect of syntax is intrinsically tied to mappings, and that is linearization. As pointed out by Avcu & Rhodes (p. 190), linearization is at the heart of many cognitive issues surrounding syntax. And while there is still a lot to be learned about linearization from a subregular perspective, it seems that subregular complexity once again plays an important role.

In the following, I will briefly sketch in Section 3.1 why linearization without movement is an input strictly local (ISL) mapping (see also Chandlee this volume,

p. 209f, 215), and how we can enrich ISL with tier-local information to associate each mover to all its landing sites (Section 3.2). This on its own is not enough for linearization because tiers do not allow us to determine whether a given landing site is a mover's final landing site, i.e. the position where it must be pronounced. But as I will argue in Section 3.3, movement seems to be restricted in exactly the right manner so that this distinction can be drawn in another, tier-local manner. Overall, then, the linearization mapping is not an outlier with respect to subregular complexity, and studying it from this perspective provides a new perspective on key properties of movement, one that is compatible with findings in the experimental literature (Section 3.4).

3.1 Linearization without movement

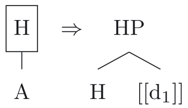
Let us first clarify what exactly is meant by the term *linearization* here. Linearization can be taken to refer to the process of computing the string yield of a tree. But it may also be construed as the process of converting a tree structure into another tree structure where the leaf nodes appear in the correct linear order, e.g. a PF structure or an X′-tree prior to Kayne (1994). I will refer to the former as *full linearization* and the latter as *tree linearization*. Tree linearization is one way to achieve full linearization: one first linearizes the syntactic structure into a phrase structure tree and then performs a recursive descent traversal of the linearized tree. Since the recursive descent traversal is fairly simple, this two-step decomposition of full linearization allows us to study the interesting parts of linearization as a tree-to-tree rewriting system, the mathematics of which are better understood than tree-to-string rewriting systems.

Suppose, then, that our syntactic representations are once again feature-annotated dependency trees as in the target paper. Even though these trees have linear order, this order has nothing to do with tree linearization itself and merely encodes the difference between the first argument and the second argument. If it were not for movement, though, these trees would be easy to transform into linearly ordered phrase structure trees, at which point the string yield is obtained by reading leaves from left to right. This requires only three types of rewrite rules depending on whether a head H has no arguments, exactly one argument A, or exactly two arguments A and B.

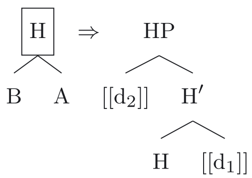
(8) **Three rule templates for rewriting dependency trees without movement into phrase structure trees**

a. [H] $\Rightarrow$ H

b.



c.



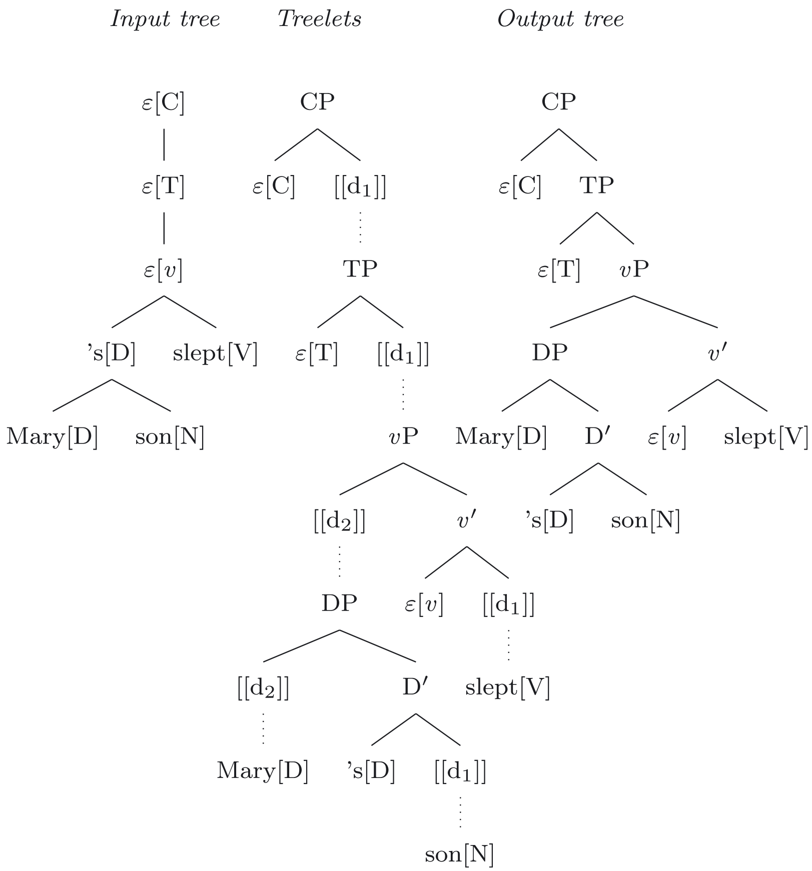
The rule templates above build on the generalization of ISL from strings to trees in Graf (2020).[1] Each one defines a way of rewriting nodes in an input tree into partial trees, called *treelets*, from which the final output tree will be assembled. In the case at hand, the input trees are dependency trees, and the treelets correspond to phrasal projections with placeholders for arguments.

Consider the template in (8a). It states that a head H without any daughters (i.e. without any arguments) should be rewritten as itself. This template will have many instantiations, e.g. one where H is the noun *car*, another one where it is *he*, yet another one where it is *slept*, and so on. But they all have in common that each head H without any arguments is rewritten as a treelet that consists only of a single node, which is H itself. The template in (8b), on the other hand, tells us that if we have a head H with exactly one argument A, we replace H with a treelet that consists of a node HP with two daughters. The left daughter is H, whereas the right daughter must be filled by the treelet that the first argument is rewritten as. This is formally expressed as [[d_1]]. Expressions between double square brackets like [[d_1]] are called *ports* as they provide the means to link together the treelets produced from nodes in the input tree. Each port is associated with a constraint that describes what material the position of the port is to be filled with. In this case, [[d_1]] means "fill this position with the treelet produced from the unique node x such that x is the first daughter (from the right) of the node that is currently being rewritten". And [[d_2]] means "fill this position with the treelet produced from the unique node x such that x is the second daughter (from the right) of the node that is currently being rewritten". We see

1 The version of ISL in Graf (2020) is distinct from the version of ISL in Ji and Heinz (2020), which is mentioned by Chandlee (p. 215). This is because there are multiple definitions of ISL that are identical when considering string-to-string functions yet diverge when taking the step from strings to trees. Ji and Heinz (2020) take as their vantage point the presentation of ISL in terms of finite-state transducers in Chandlee (2017). Graf (2020), on the other hand, starts with the context-based definition used in Chandlee and Heinz (2018).

both of these ports in the third rule template. It tells us that a head H with two arguments will be rewritten as a full HP with the specifier being [[d_2]] (i.e. the treelet produced from B) and the complement being [[d_1]] (i.e. the treelet produced from A).[2] Given these templates, a dependency tree is converted into a linearized phrase structure tree by first replacing every node with the corresponding treelet and then stitching together these treelets in the manner prescribed by the ports.

(9) **ISL rewriting of a movement-free dependency tree into a phrase structure tree**



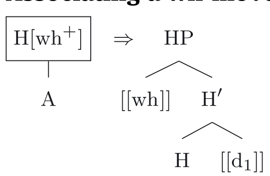
2 Of course one could change the shape of the treelets in order to accommodate headedness differences between languages. The ISL view of linearization thus is perfectly compatible with a Merge-based approach to Universal 20 as advocated in Abels and Neeleman (2009). As far as subregular complexity is concerned, there seems to be no reason to prefer a movement-based account via the LCA (Kayne 1994).

Crucially, the system above only works if there is no movement. That is because our only ports are [[d_1]] and [[d_2]]. These ports express constraints that only use mother–daughter relations in the input tree, and this is insufficient to handle movement dependencies that can span arbitrary distances. But if we also allow ports that use mother–daughter relations on movement tiers, we gain the ability to attach movers to their landing sites.

3.2 Associating movers with their landing sites

The rule in (10) uses a new kind of port [[wh]], which requires the position to be filled with the treelet produced from the unique node *x* such that i) *x* carries wh^- and ii) the node currently being rewritten is the mother of *x* on the wh-movement tier. Given this interpretation of [[wh]], the rule states that the head H carrying wh^+ should be rewritten as an HP such that the specifier is filled by the treelet produced from the closest wh-mover. Note that in a well-formed derivation, this closest wh-mover is guaranteed to exist and to be unique.[3]

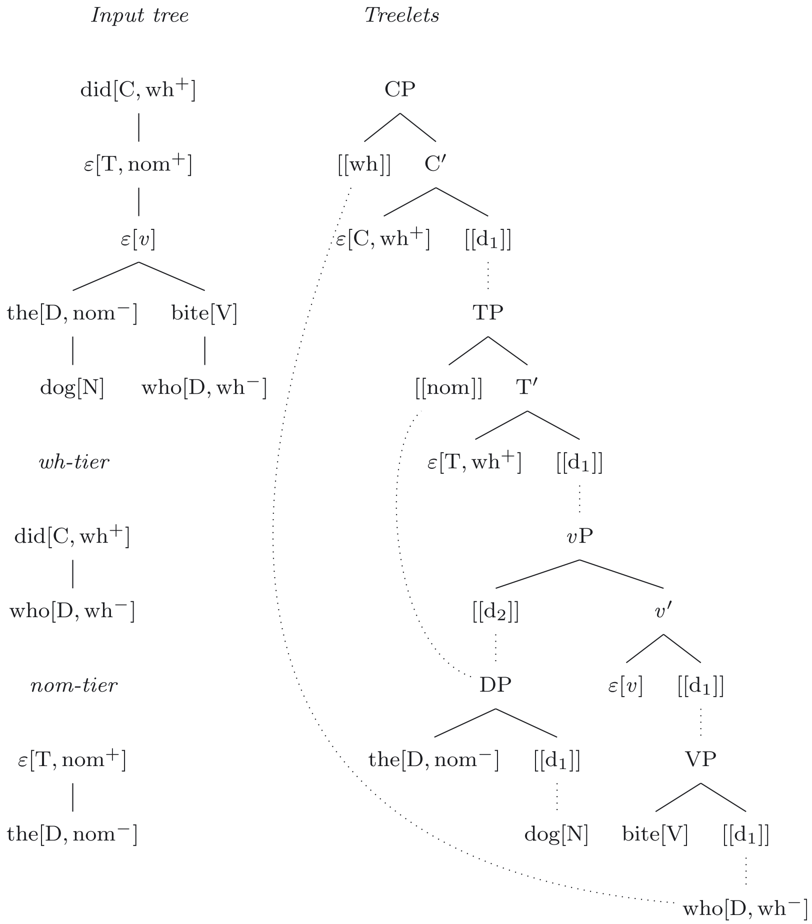
(10) **Associating a wh-mover with its landing site**



This template is sufficient to connect movers to their landing sites, as (11) illustrates for *who did the dog bite*.

3 As pointed out by Brody (p. 200), Graf and Kostyszyn (2021) show that one can relax the conditions on movement such that a node with wh^+ may have more than one daughter with wh^-, giving rise to multiple wh-movement. In this case, [[wh]] no longer picks out a unique node. When our intended outputs are multi-dominance trees, this is no problem as we can add an edge to every node *x* that satisfies the constraints imposed by the port, associating all of them with the same specifier position. For output trees with traces, however, it is currently unclear how a single port could be filled with a possibly unbounded number of movers, all of them in the correct linear order.

(11) **Dependency tree with movement and corresponding treelet configuration**



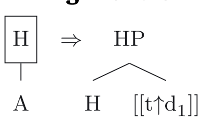
Depending on whether we take the mapping to produce graphs or trees, the output is either a multi-dominance tree or a phrase structure tree with copies. The multi-dominance tree is in a sense the more fundamental one of the two as this is the configuration we obtain from the rewrite rules before the treelets are stitched together to form the output structure. Either way, though, we arrive at a structure where every mover is associated with all its landing sites, and we were able to do so using only mother-daughter relations over the input tree and its movement tiers.

3.3 Delinking movers from non-final positions

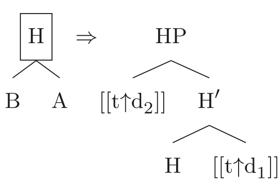
Unfortunately, connecting movers to their landing site is insufficient for tree linearization. Rather, we want to connect each mover to its final landing site and delink it from all other positions, including its base position. Let us first consider the case of delinking from the base position. In this case, we use a slightly more elaborate constraint format for ports that can enforce conditions such as "fill this position with a trace if the first daughter is a mover, and with the output of the first daughter otherwise" (written more succinctly as $[[t{\uparrow}d_1]]$).[4] This gives us two revised versions of (8b) and (8c).

(12) **Delinking movers from their base positions**

a.



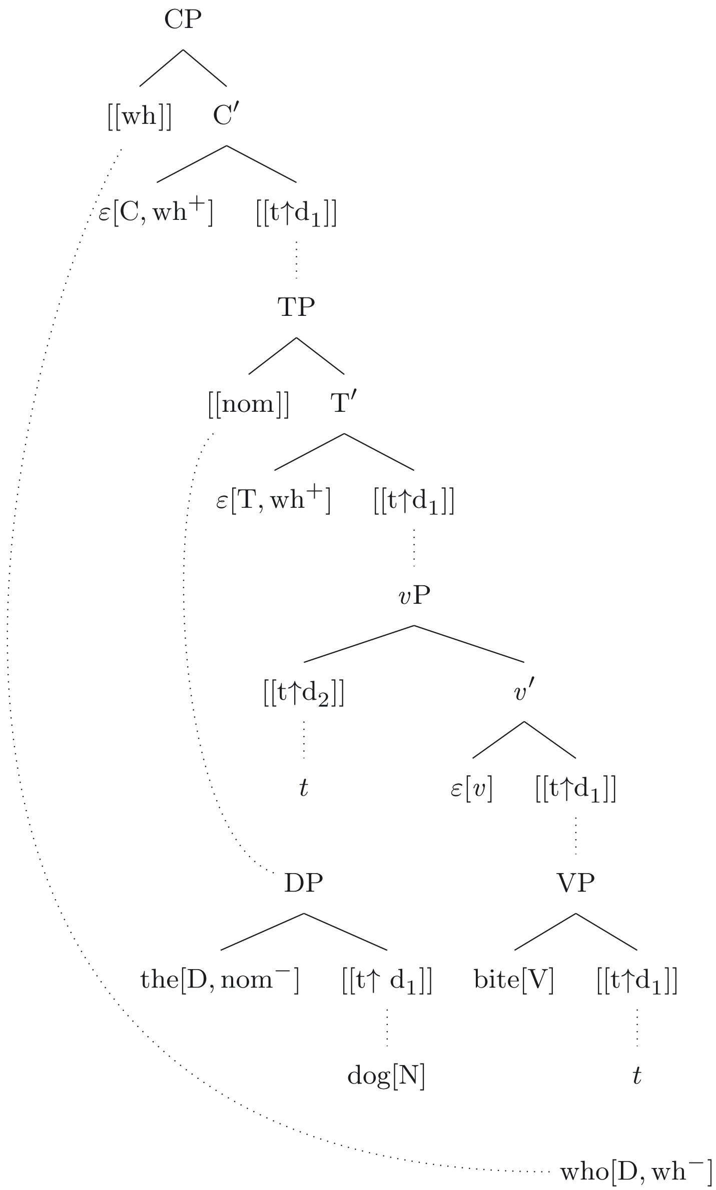
b.



With these updated rules to account for base position delinking, the treelet configuration from (11) is no longer multi-dominant.

4 The notation is inspired by *ternary if* in various programming languages and can be read as follows: "*t* if movement (↑), else d_1".

(13) **Treelet configuration with movers delinked from base positions**



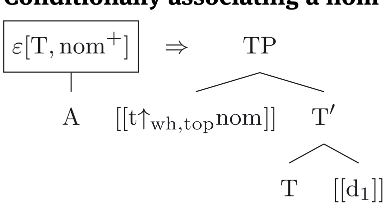
While this correctly handles the delinking at base positions, we still run into problems with intermediate landing sites. Consider the example sentence in (14), with copies in pointy brackets.

(14) Who does John think that Mary said ⟨who⟩ might ⟨who⟩ have left.

Here *who* first undergoes movement to the embedded subject position (Spec,TP) before wh-moving to the left edge (Spec,CP) of the matrix clause. In our feature calculus, this means that *who* carries both nom^- and wh^-. Hence the rule in (12a) will correctly delink *who* from its base position, replacing it with an unpronounced trace. But the rewrite rules for movement in (10) will leave *who* attached to both Spec,TP and Spec,CP, and the sentence would be incorrectly linearized as *who does John think that Mary said who might have left*. The rewrite rules fail to differentiate an intermediate landing site from a final landing site. This information cannot be gleaned from the nom-tier or the wh-tier in isolation. Nor can we use a combined tier for both subject movement and wh-movement because all the instances of subject movement in higher clauses would make the relation between *who* and its wh-landing site non-local on said tier. Tiers on their own, then, do not provide the necessary information to clearly distinguish final landing sites from intermediate landing sites, which makes it impossible to correctly linearize movement in those cases where a phrase undergoes multiple movement steps.

However, it seems that syntax is conspicuously designed to avoid exactly this problematic scenario. Consider once more the example in (14). It is commonly claimed that subject movement must always precede wh-movement. This is expressed via the *Ban on Improper Movement*, which does not allow any phrase to undergo A-movement steps like subject movement after it has already undergone an A′-movement step such as wh-movement. Suppose then that we have a port for the Spec,TP position that combines the constraint we used for movement with the one we used for delinking at the base position: "fill this position with a trace if the nom-tier daughter with nom^- also carries an A′-movement feature (e.g. wh^-), and with the output of this tier daughter otherwise". The notation for that is $[[\text{t}{\uparrow}_{f_1,\ldots,f_n}\text{nom}]]$, where $f_1,\ldots,f_n$ is a list of A′-movement features.

(15) **Conditionally associating a nom-mover with Spec,TP**



With this revised port, a subject mover will be inserted in Spec,TP only if the mover isn't also undergoing wh-movement or topicalization. This rule works as intended because we can infer from the Ban on Improper Movement that wh-movement or topicalization must follow the subject movement in a well-formed derivation, and consequently the subject position in Spec,TP cannot be the mover's final landing site if it carries wh$^-$ or top$^-$.

The tree linearization system described above is subregular. It takes as its vantage point Graf's (2020) generalization of ISL string-to-string mappings to ISL tree-to-tree mappings. As discussed by Chandlee (p. 209f, 215), ISL mappings are very common across phonology and morphology. For syntax, ISL tree-to-tree mappings are sufficient to handle linearization without movement as well as cases where spell-out of a node is conditioned by the local tree context.[5] Movement, however, requires ports that can consider not only local information but also tier-local information, as outlined above. Keep in mind that these tier-based port constraints are sufficient to connect movers to all their landing sites no matter what the grammar looks like. But once linearization enters the picture and we have to distinguish final landing sites from intermediate landing sites, tier-based ports are sufficient only if there are ways to infer directly from the mover which of its landing sites is the last one.

This provides a third-factor explanation in the sense of Chomsky (2005) for the Ban on Improper Movement: the reason this constraint exists is because tiers on their own are not sufficient to handle linearization otherwise. In fact, this computational perspective derives a more general version of the Ban on Improper Movement. Whenever a lexical item carries more than one movement

5 For example, an ISL tree-to-tree transduction can replace the root $\sqrt{destroy}$ with the verb *destroy* if it appears under the functional head *v* and with the noun *destruction* if it appears under a functional head *n*. ISL mappings can also take spans of functional heads (or even individual features as in Nanosyntax) and spell them out as a single lexical item or affix.

However, ISL transductions cannot handle phenomena that are not local in the tree, even if they are local in the tree's string yield. Examples of that include the *a*/*an* alternation in English and the contraction of *zu dem* to *zum* in German. The phrases below illustrate that these are non-local processes over trees.

(1) a*(n) [[[[extraordinarily rarely] observed] and [hardly common]] phrase]

(2) zu [[[[dem Künstler] seiner Mutter] ihrem Freund] seiner Gallerie]
to [[[[the artist] his mother] her friend] his gallery]
'to the artist's mother's friend's gallery'

These alternations are ISL over strings but require a lot more power over trees, although their precise complexity remains unknown. All one can say for now is that they can be captured with first-order logic over trees, sensing top-down tree transducers (Graf and De Santo 2019), and bottom-up tree transducers, but these are loose upper bounds on complexity.

feature, it must be possible to predict just from its feature make-up the final movement step it will take in every well-formed derivation.

(16) **Generalized Ban on Improper Movement**
Let d_1 and d_2 be two well-formed derivations of some language. Then there must be no lexical item l with movement features f^- and g^- such that l occurs in both d_1 and d_2 and the last movement step of l is f-movement in d_1 but g-movement in d_2.

This condition seems to hold. An analysis of the MG treebank created by Torr (2017) reveals no counterexamples, and it actually supports a slightly stronger version of the Generalized Ban on Improper Movement: there are no two lexical items that have the same feature make-up but differ in the order of the movement steps they undergo (including intermediate movement steps). While corpus data constitutes but a finite sample of an infinite problem space, it is encouraging that considerations of subregular complexity give rise to a generalization that meshes well with existing syntactic principles and also passes initial tests based on available data.

3.4 Cognitive implications

While the subregular picture of tree linearization has the appeal of maintaining the parallels to phonology and morphology, it seems to be at odds with experimental findings discussed by Avcu & Rhodes (p. 190):

> [T]rees cannot be pronounced. They must be sent to Spell-Out where a linearization algorithm renders them as strings.
>
> We have neurobiological evidence for this division of labor (and division of complexity). [...] Matchin and Hickok (2020) propose a division of labor where hierarchical structures are built-in pSTS, then sent to LIFG for linearization during speech production. This proposal gives us two systems—one which builds trees and (by Graf's account) is likely limited to subregular computations; and one which converts trees into strings and may need more computational power or greater memory resources.

The two perspectives are not incompatible, though. Of course the usual disclaimers apply regarding the disconnect between competence/specification and performance/implementation. But invoking this distinction to inoculate formal work against experimental criticism is far less productive than assuming that there is a tight coupling between competence and performance (cf. Kobele et al. 2013; Graf et al. 2017; Pasternak and Graf 2021) and that any apparent discrepancies require a more principled explanation.

In the case at hand, it is important to keep in mind that tree linearization is just a small part of converting trees into pronounceable output structures. At the very least, full linearization also involves a recursive descent traversal of the linearized tree in order to obtain the actual string yield. But even that is too simplified a picture as full linearization still misses crucial aspects of post-syntactic computation. PF structures must be enriched with prosodic information, and as part of this we may see restructuring and post-syntactic movement (cf. Richards 2016). There are also lexical alternations and contractions that are better handled over the linearized string rather than the tree itself (see also fn. 5). The findings mentioned by Avcu & Rhodes do not argue against tree linearization being subregular, they argue against subregular tree linearization being all there is to the output pipeline from syntactic derivations to pronounced sentences.[6]

Summarizing the discussion in the preceding two sections, we see that subregular syntax is not limited to constraints and can be readily extended to mappings, be it for the spell-out of individual heads, attaching movers to their landing sites, or generating a linearized tree structure. In all these cases, we find that the computational mechanisms are once again local over a suitable notion of tier. These are just the first forays into subregular mappings for syntax, but the results are promising and have already yielded new empirical predictions such as the Generalized Ban on Improper Movement.

6 Another complicating factor here is that it can be difficult to predict the aggregate complexity of a sequence of rewrite steps from the complexity of the individual rewrite steps. For example, just because an SPE-style grammar consists of two rewrite rules such that each one is an ISL string-to-string mapping, this does not guarantee that the grammar itself is an ISL string-to-string mapping. This is irrelevant for the analysis of movement given above because it involves only a single mapping from dependency trees to linearized phrase structure trees. But the conversion of a dependency tree into the actual pronounced string presumably involves many steps: tree linearization, PF adjustments, tree traversal, string rewriting rules from phonology and morphology, and perhaps additional steps in-between or at the end. Even if each one of these steps is very simple, the full pipeline could still be quite complex.

Fortunately, it seems that first-order logic once again provides a reasonable upper bound on complexity. Each individual step should be definable in first-order logic, and since first-order transductions are closed under composition, applying multiple first-order transductions in sequence still does not push us beyond the threshold of first-order logic.

4 Subregular linguistics for empirical analysis

The previous two sections show that subregular linguistics is a much more empirical enterprise than is common for work rooted in formal grammar. In contrast to upper bounds like definability in first-order logic, the bounds of SL and TSL are so tight that every phenomenon must be considered carefully. A lot of the work so far has focused on outlining how one could even begin to reanalyze well-known data through the lens of syntactic tiers. For example, Graf (2022) shows how tiers can be used to capture the core properties of Irish wh-agreement, the *that*-trace effect, and the anti-*that*-trace effect, among others. This initial work, by necessity, cannot hope to give an exhaustive analysis that captures every detail that has been noted in the linguistic literature. But it can show that further exploration of these phenomena from a subregular perspective is viable because TSL can at least capture their central properties. The analyses are a first step, not the final story, and this is exactly why theoretical linguists with their analytical expertise have a lot to contribute to the program of subregular linguistics.

In light of this, I greatly appreciate the empirical discussions in the commentaries by Himmelreich and Chaves & Putnam, and I will briefly comment on each one. My discussion of Himmelreich's analysis in Section 4.1 will focus on how her argument goes even deeper than the argument-adjunct distinction, telling us something about the structural arrangement of specific heads. My reply to Chaves & Putnam (Section 4.2), on the other hand, revolves largely around the notion of gradience and whether it is incompatible with the TSL-view of islands.

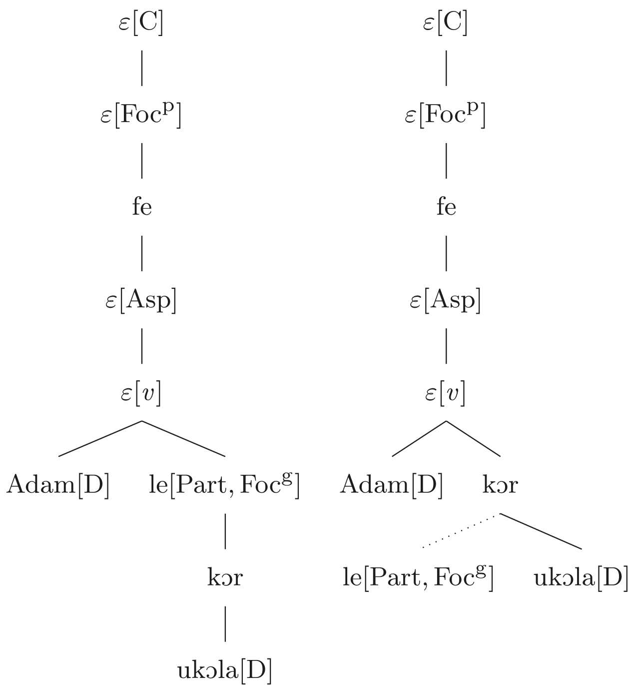
4.1 Himmelreich's analysis of focus particles

I take Himmelreich's analysis to establish two important points. First, the combination of feature-annotated dependency trees with tree tiers is flexible enough to be readily applied to a phenomenon that is very much unlike any others previously considered in subregular syntax. This phenomenon is focus marking in Likpakpaanl, which turns out to have some unexpected formal connections to island constraints. Second, restrictions that are inherent to TSL phenomena allow us to make inferences about the syntactic status of specific elements.

Himmelreich phrases the latter point in terms of a split between selection and adjunction, but as we will see soon the contrast runs even deeper than that. As she points out, focus marking in Likpakpaanl always targets phrases, not individual heads. Hence it is impossible to focus mark just the verb and the focus marker *le* has to attach to the whole VP instead. This is expected if the focus marker is a head that takes the verb as an argument, whereas it is surprising if the focus marker can be a dependent of the verb by adjoining to it.

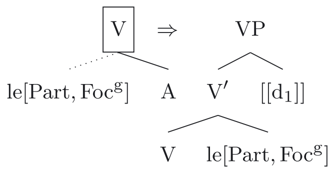
(17) **Head and adjunct analysis of *le* (Himmelreich (22) & (23))**

Head analysis *Adjunct analysis*

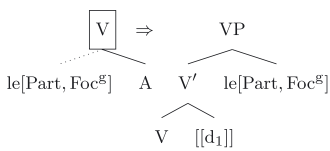


This line of reasoning is corroborated by our discussion of ISL tree-to-tree mappings in Section 3.1. The rules for a head with a single daughter will linearize the head either to the left of the treelet produced from the daughter node, or to its right. It is impossible to put the head inside of this treelet. If the focus marker is a head that selects the verb, this immediately entails that the focus marker must be to the left or to the right of the treelet built from the verb, which is the whole VP. If, on the other hand, the focus marker is a daughter of the verb, it could be inserted immediately after the verb as in (18a) or after the other argument as in (18b).

(18) a.



b.

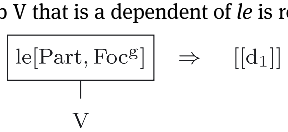


Just based on how tree linearization works, in particular that heads cannot appear inside the treelets produced from their daughters, it follows immediately that the head analysis of the focus marker *le* forces it to be linearized with respect to the whole VP, not just the head V.[7]

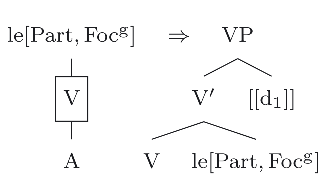
Note that the discussion above does not mention whether the focus marker is an adjunct or not. That is because Himmelreich's insight hinges on the position of the focus marker relative to the verb, not whether that position belongs to an adjunct or an argument. If we adopted the treatment of adjunction as category preserving selection in Graf (2022), then the focus marker could be an adjunct and still be unable to appear VP-internally.

7 There is a way to use ISL rewrite rules to place *le* next to the verb even under the head analysis of *le*. In this case, the focus marker produces no output of its own when it takes a verb as its argument, and a verb V that is a dependent of *le* is rewritten as V-*le*.

(1) a.

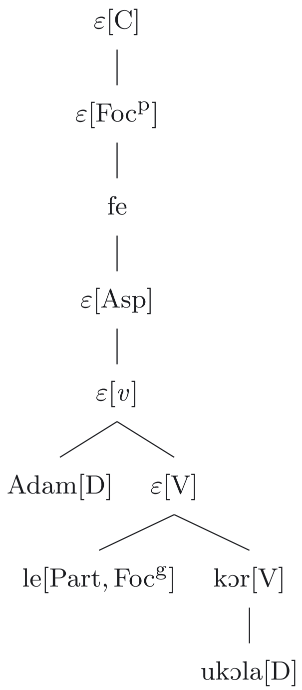


b.



But note that this comes at the cost of an increased locality domain, because the rewrite rule for the verb now needs to consider both its mother and its daughters. The larger the locality domain, the higher the complexity of the ISL mapping. If languages show a general bias towards ISL mappings of lower complexity, then these rules are more marked and hence should be less common typologically.

(19) **Adjunct analysis with category-preserving selection**



Here adjunction is implemented via an unpronounced V-adjunctivizer, i.e. a V-head that takes the VP as its first argument and the focus marker as its second argument. Even though this implementation reduces adjunction to category-preserving selection and thus has no split between arguments and adjuncts, the tree linearization mapping we have been using still cannot linearize the focus marker VP-internally. Hence, even if the grammar has no separate adjunction operation at all, Himmelreich's analysis still tells us something about the structural location of the focus marker (or alternatively, about the complexity of tree linearization, see fn. 7). This gives rise to two different lines of research that could be pursued, with one focusing on additional evidence for the marker's structural location, and the other exploring whether there are independent reasons to posit a more powerful tree linearization mapping. This is an instance of exactly the kind of interplay between theory and data that drives inquiry in theoretical linguistics.

4.2 Islands and gradience

Just as Himmelreich's remarks set the state for future work, so do the observations by Chaves & Putnam regarding the behavior of adjuncts. They conclude (p. 222):

> [The] TSL account of Adjunct Islands and the CNPC is not only stipulative, it is incompatible with the empirical facts as it wholesale rules out all such extractions. It is unclear how SL can predict the nuanced observed empirical patterns (gradience, satiation, construction-specificity, contextualization effects, etc.) from independently motivated factors, and explain why things should be as they are.

I thank them for clearly outlining their empirical benchmark for what a modern approach to islands has to be able to account for. I am confident that the program of subregular syntax is able to handle these facts, and that we will gain a better understanding of both islands and subregular syntax by doing so.

Let me first restate the central point I hoped to establish regarding adjuncts, this time with particular attention to the difference between islands as classically construed in generative syntax and islands as a phenomenon of experimental linguistics: The class of TSL dependencies includes not only unrestricted movement dependencies, but also movement dependencies that are subject to blocking effects, with the latter corresponding to the classical notion of islands. As I pointed out, this insight does not explain why every natural language exhibits island effects, nor why island effects tend to be very systematic. As such it clearly cannot hope to be a theory of islands, but that does not preclude that a theory of islands could be built on this TSL foundation, and there are at least two ways to do so.

One is to adopt the position that is still commonly entertained in generative syntax, namely that islands are a multi-causal phenomenon and that the job of a syntactic theory is just to capture the aspects of islands that arise in syntax. It seems that this stance is still implicitly assumed in a lot of contemporary work on islands. Take the widely cited papers by Stepanov (2001, 2007), and let us contrast his account against the TSL view. Stepanov derives the islandhood of adjuncts from the assumption that they are all late-merged, which is a major change to the Minimalist architecture. The TSL view highlights that island effects can already arise in a system without such architectural changes, simply because they are computable with the cognitive resources that are needed for movement—if something can happen, it will happen eventually in some language. Of course this approach of "free variation within the limits of computation" means that more needs to be said about why island effects are very principled, but the positive flip side is that exceptions to island effects are easier to accommodate than in Stepanov's approach. If *while* selecting finite T induces island effects, for example, but *while* selecting infinitival T does not, then that can be captured by projecting the former but not the latter as the two differ in their feature make-up and are thus distinguishable for the purposes of tier projection. Hence the tier-based view of islands is easily amenable to lexical variability as observed by Bondevik et al. (2021).

Chaves & Putnam might still object that this locks us into a categorical view of islands that is incompatible with the gradient data, but that is a fallacy. Modeling island constraints in a categorical manner does not limit us to a categorical view of islands. Categorical constraints already give rise to gradience because each structure (e.g. a dependency tree) is associated with a particular violation profile (cf. Pullum and Scholz 2005, pp. 490–492) — what constraints were violated, and in what configurations?[8] The criticism of Chaves & Putnam implicitly conflates the nature of constraints with the nature of the whole grammar, and these two things must be kept separate. In phonology, this point has been made even more forcefully by Gorman (2013) and Durvasula (2020), who argue at length that switching from a categorical grammar formalism to a gradient one is not only unnecessary to account for the observed data, it actively obscures the relevant generalizations. While phonology may well be different from syntax in this respect, the onus is on Chaves & Putnam to support their assertion that gradient data is incompatible with a categorical model of syntax. At this point, then, it is far from obvious that island effects are not TSL.

But conceptual banter can only take us so far, let us focus on the real issue here: is there any practical route forward for a TSL-based view of islands to an empirically adequate model of the experimental data on island effects? As I already mentioned above, I am very optimistic that the answer is Yes. Consider once more the issue of gradience. The difference between categorical and gradient grammars is much smaller than commonly assumed—in mathematical terms, we are replacing the grammar's Boolean monoid with a probabilistic one but keep the same method of computing an aggregate value for the whole structure from the values of its local substructures (cf. Goodman 1999). In the concrete case of TSL over strings, Mayer (2021) shows how the tier projection can be made probabilistic, and he uses this to model gradient blocking and distance-based decay in Hungarian and Uyghur. The very same technique can be applied, unaltered, to TSL grammars over trees, although it is of course an empirical question whether that is sufficient to handle island effects. Perhaps the tier projection also needs to consider the structural context of specific lexical items. This would correspond to the more powerful class ITSL (De Santo and Graf 2019), which was originally proposed to handle some phonological phenomena that are not TSL. Notably absent from the range of properties Chaves & Putnam list for islands are exactly those that Graf (2022) identifies as very likely not to be TSL, e.g. cowardly islands ("XP is an island iff there are at least *n* XPs in the same clause") or discerning

8 Note that one does not need to keep track of the number of violations unless one wants infinitely many distinct violation profiles and hence infinite degrees of gradience. None of the data in the literature suggests that such infinite gradience is needed for language relative to a system with, say, 10 degrees of gradience.

islands (“XP is an island only for movers that contain a YP”). My conjecture, then, is that island effects stay within gradient counterparts of TSL or ITSL.

That said, if a step up in power turns out to be absolutely necessary, then so be it. The goal of subregular linguistics is not to shove a square peg into a round hole in a mindless effort to analyze everything as TSL dependencies; the goal is to determine through empirical analysis the properties of the computations that underpin specific phenomena, and to leverage these properties for typological predictions, learning algorithms, and new bridges to neighboring areas of cognitive science.[9] I will use the final section of this reply to sharpen this important point.

5 Formalisms, again

It is perhaps fitting that this debate, which started with my lengthy discussion of the role of formalisms in formal grammar and theoretical linguistics, should wrap up with another batch of methodological musings. I return to this issue in an effort to explain why I am sympathetic to many points raised by Brody and Chaves & Putnam while at the same time rejecting their relevance to subregular linguistics.

Let me reiterate a point I already made in the target paper: subregular linguistics, be it subregular phonology, subregular morphology, or subregular syntax, is not in the business of prescribing formalisms or meta-languages. As I wrote on p. 152 regarding the focus of formal grammar on the formalism:

> Subregular linguistics is a noteworthy departure in this respect. Rather than the linguistic formalism, it is the linguistic analyses that matter most to the mathematical inquiry. […] [S]ubregular syntax is not a formalization of GB, Minimalism, or TAG, but a mathematical investigation of syntax that builds directly on the insights that have grown out of these syntactic theories as well as what formal grammarians have learned about them.

9 If it were the goal to make syntactic dependencies TSL, the goal would be trivially easy to meet as we could just take a hint from GPSG and encode all dependencies via abstract features that are never morphologically realized and are passed around in a local manner via SLASH-style feature percolation. As long as all the syntactic dependencies are regular, the resulting system would be SL and thus would not need any tiers at all. Graf (2017) explains in detail how this works, and why the problem is primarily with Merge features (category and selector features) rather than movement features. Graf (2020) presents a principled method to rein in this kind of Merge-based feature coding. The central idea is that Merge features must be locally inferable: if all we are given is a dependency tree without feature annotations, there must be an ISL tree-to-tree transduction that assigns the correct category and selector features to each lexical item. This restriction makes it impossible to encode long-distance dependencies via Merge features. Graf (2020) conjectures that all natural languages have ISL-inferable category systems, and a preliminary analysis of the MG treebank (Torr 2017) bears this out.

Chaves & Putnam miss this key aspect of subregular linguistics when they claim (p. 227) that "[t]he research program that [subregular syntax] builds on assumes that the ideal grammar formalism should impose restrictive expressiveness on the theory." Subregular linguistics is not about grammar formalisms or how linguists choose to codify their findings. The remarks in Section 3.1 of Chaves & Putnam regarding "naive grammatical realism" (p. 222) and the difference between formalisms and theories (which echoes a lot of Section 2.2 of my target paper) thus do not apply to subregular linguistics.

Subregular linguistics is about the complexity of the patterns we find in natural languages, not how we should talk about these patterns. When we say that a phenomenon is TSL, the claim is not that speakers are literally projecting tiers in their mind (see the Appendix for an alternative, tree-less way of thinking about TSL over trees). Tree tiers are a very intuitive way to think about TSL, but it is just one of many ways the computational class TSL can be conceptualized, just like there are dozens of ways to think about what it means for a dependency to be regular (recognized by a finite-state automaton, definable as a regular expression, definable in monadic second-order logic, having a Myhill–Nerode congruence relation of finite index, and much more). Subregular linguistics is in the business of making empirical claims about the complexity of phenomena given some initial assumptions about how these phenomena ought to be analyzed, e.g. over strings or autosegmental structures in phonology (cf. Chandlee p. 214), or over specific tree structures in syntax that are inspired by current linguistic proposals.

At the very heart of subregular linguistics as an empirical enterprise at the intersection of theoretical linguistics and formal grammar are two claims.

(20) a. **The empirical challenge**
It is surprising that the majority of linguistic dependencies and processes exhibit very low complexity, in particular because we seem to find the same kind of dependencies across phonology, morphology, and syntax. This is in need of explanation.

b. **The theoretical opportunity**
The fact that the patterns we find are very restricted opens up new research opportunities.

My target paper, the commentaries by Avcu & Rhodes, Chandlee, and Himmelreich, as well as the preceding sections of this paper have given numerous examples of such research opportunities. Below is a short summary, with particular emphasis on subregular syntax.

(21) a. new perspectives on well-known phenomena: movement, islands, extraction morphology, linearization, focus marking
b. unexpected connections between seemingly unrelated phenomena: movement and culminativity, islands and blocking, extraction morphology and harmony
c. novel generalizations: the Generalized Ban on Improper Movement, ISL-inferability as a universal of category systems (see fn. 9)
d. bridges to cognitive science: minimum requirements for cognitive inference mechanisms and memory (see also Pullum and Rogers 2006; Rogers and Pullum 2011), artificial language learning experiments

Another major payoff comes in the form of learning algorithms, and this is also what provides an answer to the empirical challenge. Lambert et al. (2021) prove that there is a close connection between subregular classes like SL and TSL on the one hand and classes of efficiently learning string languages on the other hand. That is to say, a bias towards efficient learning is a bias towards simple subregular classes. Subregular linguistics thus provides an answer to its own empirical challenge.

Competing approaches, on the other hand, still have to provide an answer to the empirical challenge posed by subregular linguistics. It cannot be handwaved away. Brody states (p. 201) that "there is no reason to think that the simplest, most explanatory theory of syntactic competence that generates all clearly necessary structures and is therefore descriptively adequate as part of the grammar, will not generate additional structures that the other interacting components ignore or map to various deviant outputs." This is a methodologically sound point, but in the absence of a concrete proposal of what these other components are, and without proof that they all conspire to limit us to only the restricted subregular patterns that we see, this stance is empirically toothless.

One might object that subregular syntax is in an equally problematic situation because the results of Lambert et al. (2021) concern phonology and thus do not offer an explanation for the TSL nature of syntax. But as Lambert et al. (2021) emphasize themselves, their learnability claims are not tied to strings, they generalize to trees assuming that the learner is given trees as input. For the specific trees I have assumed, this might be close to true. While the dependency trees are an abstract representation of the syntactic derivation, their basic structure is determined by head-argument relations. Head-argument relations are a fundamental part of a sentence's meaning. To the extent that a child learner can infer the meaning of a sentence, they can infer head-argument relations. Syntactic learning, then, could be idealized as the problem of being given a dependency tree and a string and to infer from that what features must be assigned to each lexical item. Once the features are in place, a variation of the existing learning algorithms for string classes can be applied to infer tiers and identify tier constraints. But how hard would it be to learn those features? Category features might be inferable from the local context (see fn. 9), and some information about movement features is provided by the discrepancy between the string and how the dependency tree would have been linearized if there were no movement. Whether that is sufficient to correctly infer all movement features remains to be seen. Even if that should turn out to be true, it is still a long way from a learning algorithm to a realistic model of acquisition. Subregular linguistics does not have all the answers yet, but crucially it provides a concrete plan of action for syntactic learning, with clearly identified subproblems that can be solved incrementally as our understanding of subregular learning improves.

A lot more could be said here about specific methodological points: why using the term *stipulative* (Chaves & Putnam p. 222) is a category mistake when talking about formal complexity results, how subregular linguistics engages with I-language and not E-language (Chomsky 1986), how formal grammar has ways to quantify the trade-off between expressive adequacy and descriptive elegance (Savitch 1993), or why subregular linguistics would be an insightful enterprise even if some patterns may not be subregular at all. The paper would gain little, though, from such philosophical pontificating. The commentaries by Brody and Chaves & Putnam have given me an opportunity to clarify some crucial points about the nature of subregular results, and it is these results that really matter. As Brody says (p. 202), subregular linguistics in general and subregular syntax in particular "needs to be judged, like any other theoretical move, not on its prior plausibility, but on its actual explanatory merits." The target paper, the commentaries by Avcu & Rhodes, Chandlee, and Himmelreich, as well as this response paper have provided a smorgasbord of results, generalizations, and predictions. I hope that this sampling will convince many readers that subregular linguistics has something of value to offer to them.

6 Conclusion

Subregular syntax is part of the larger enterprise of subregular linguistics, which takes as its vantage point the observation that a surprising number of linguistic phenomena are remarkably simple from the perspective of formal grammar. Far from merely a mathematical curiosity, this empirical observation opens up new strategies to explore learnability, typology, and cognition. Subregular linguistics provides a unified vision across language modules that connects seemingly unrelated phenomena—for example, wh-agreement and phonological harmony with icy targets, as we saw in this paper. Subregular complexity considerations derive key aspects of language, such as a version of the Ban on Improper Movement. At the same time, the mathematical foundation of subregular linguistics makes seemingly major changes in architecture such as the move from categorical to gradient notions of acceptability a matter of minor, almost trivial modifications. These are but a few examples of what subregular linguistics brings to the table for linguists. While it remains to be seen just how much subregular linguistics in general and subregular syntax in particular can deepen our understanding of language, the last 10 years have marked a very promising start.

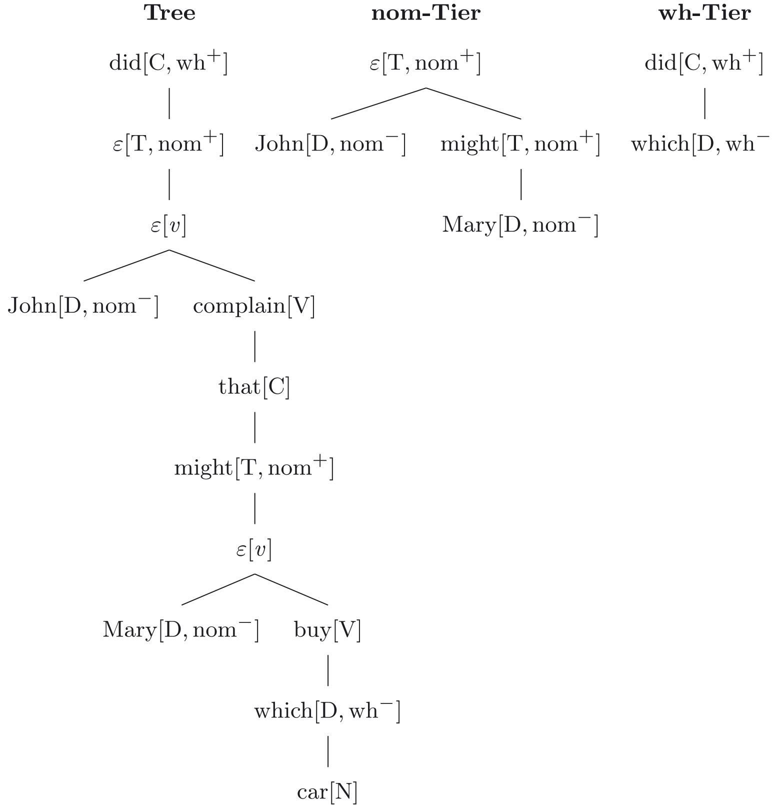
Research funding: The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344.

Appendix: TSL over trees without trees

This appendix illustrates that subregular classes like SL and TSL represent very abstract concepts that can be specified in numerous ways. In particular, it is even possible to have a tree-less view of TSL over trees. This addresses the remark of Brody (p. 199f) that "Graf talks here about daughter strings of a given node, and this way of talking contains the concept of 'string'. But it crucially presupposes the tree structure: a node with a set of daughters." It also illustrates that subregular linguistics does not put any restrictions on formalisms or metalanguages because it is the complexity of the underlying computation that matters, not how that computation is specified.

TSL over trees can be regarded as a system that associates every node in a dependency tree with one or more strings of other nodes in the tree. We visualize this process as the projection of tree tiers, and on each one of those tiers we then check that each node has a licit string of daughters. But mathematically this is only a convenient metaphor for associating a node with a string of other nodes. Each tier corresponds to a specific association between nodes and strings of nodes (in the case where a node is not projected onto the tier, it is associated with the empty string, whereas a node that is projected on the tier but has no tier daughters is associated with the special string $\ltimes$). Instead of tiers, we can use a table to represent these associations, which is exemplified below.

(22) a. **Node-string associations visualized via tiers**



b. **Node-string associations as a table**

Node	**nom-string**	**wh-string**
did[C, wh$^+$]	ε	**which**[D, wh$^-$]
ε[T, nom$^+$]	**John**[D, nom$^-$] might[T, nom$^+$]	ε
ε[*v*]	ε	ε
John[D, nom$^-$]	⋉	ε
complain[V]	ε	ε
that[C]	ε	ε
might[T, nom$^+$]	**Mary**[D, nom$^-$]	ε
ε[*v*]	ε	ε
Mary[D, nom$^-$]	⋉	ε
buy[V]	ε	ε
which[D, wh$^-$]	ε	⋉
car[N]	ε	ε

Suppose then that we have a function *parse* that assigns every string a dependency tree with tree tiers, and a function *stringify* that takes as its input a dependency tree with tiers and converts it to a table of the form above. Given a string s, the output of *stringify*(*parse*(s)) is a table that encodes associations between each lexical item and some other lexical items in the sentence, and these associations are then checked by the grammar in the same manner that we employed over tiers. The mathematical core of TSL over trees then is not the projection of tiers, but rather how tier projection allows us to associate nodes in a dependency tree, which are just lexical items, with strings of other nodes, i.e. strings of other lexical items.

This might seem like a pointless mathematical exercise because we are still invoking tree structure in order to get our function *stringify* to work. But given two functions f and g, it is always possible to compose those two functions into a single function $f \circ g$ that directly maps every x to the output of $f(g(x))$. Crucially, this need not involve the intermediary output produced by g and fed into f. For example, if $f(x) = x - 1$ and $g(x) = x + 1$, then we can simply define $f \circ g$ as mapping every x to x — there is no need to first increment x by 1 as is done by g only to decrement it by 1 immediately afterwards in line with f. The possibility of expressing TSL dependencies over strings rather than over trees by feeding the output of *parse* into *stringify* tells us that there is some function *stringify* ∘ *parse* that can take as its input a string and immediately, in one fell swoop, tell us what strings of lexical items each lexical item is associated with.

We linguists have not figured out yet what this mystery function is or how we could define it without invoking tree structure at some point, and as a result this tree-free perspective on TSL over trees is not very useful. It may also be completely different from how TSL computations are actually represented in the human mind, either because they are stated over trees after all, or because the human mind uses yet another specification that subregular researchers have not discovered yet. Be that as it may, this alternative view of TSL establishes that there are many different ways of thinking of the formal class TSL, some of which do not invoke trees at all.

References

Abels, Klaus & Ad Neeleman. 2009. Universal 20 without the LCA. In José M. Brucart, Anna Gavarró & Jaume Solà (eds.), *Merging features: Computation, interpretation, and acquisition*, 60–79. Oxford: Oxford University Press.

Bondevik, Ingrid, Dave Kush & Terje Lohndal. 2021. Variation in adjunct islands: The case of Norwegian. *Nordic Journal of Linguistics* 44. 223–254.

Brody, Michael. 2019. Some biolinguistic remarks. *Acta Linguistica Academica* 66. 335–348.

Burness, Phillip & Kevin McMullin. 2019. Efficient learning of output tier-based strictly 2-local functions. *MoL* 16. 78–90.
Burness, Phillip, Kevin McMullin & Jane Chandlee. 2021. Long-distance phonological processes as tier-based strictly local functions. *Glossa* 6. 1–37.
Chandlee, Jane. 2017. Computational locality in morphological maps. *Morphology* 27. 599–641.
Chandlee, Jane & Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry* 49. 23–60.
Chomsky, Noam. 1986. *Knowledge of language: Its nature, origin, and use*. New York: Praeger.
Chomsky, Noam. 1995. *The Minimalist program*. Cambridge, MA: MIT Press.
Chomsky, Noam. 1998. *Minimalist inquiries: The framework*, vol. 15 of *MIT occasional papers in linguistics*. Cambridge, MA: MIT Press.
Chomsky, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36. 1–22.
De Santo, Aniello & Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In Raffaella Bernardi, Gregory Kobele & Sylvain Pogodalla (eds.), *Formal grammar*, 35–50. Heidelberg: Springer.
Durvasula, Karthik. 2020. *O gradience, whence do you come?* Slides of a talk given at AMP 2020, September 20, 2020. Department of Linguistics: UC Santa Cruz.
Georgi, Doreen. 2017. Patterns of movement reflexes as the result of the order of merge and agree. *Linguistic Inquiry* 48. 585–626.
Goodman, Joshua. 1999. Semiring parsing. *Computational Linguistics* 25. 573–605.
Gorman, Kyle. 2013. *Generative phonotactics*. University of Pennsylvania Doctoral Dissertation.
Graf, Thomas. 2017. A computational guide to the dichotomy of features and constraints. *Glossa* 2. 1–36.
Graf, Thomas. 2020. Curbing feature coding: Strictly local feature assignment. *SCiL* 3. 362–371.
Graf, Thomas. 2022. Typological implications of tier-based strictly local movement. *SCiL* 5. 184–193.
Graf, Thomas & Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. *MoL* 16. 12–26. Available at: https://www.aclweb.org/anthology/W19-5702.
Graf, Thomas & Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist grammars. *SCiL* 4. 275–285.
Graf, Thomas, Monette James & Chong Zhang. 2017. Relative clauses as a benchmark for Minimalist parsing. *Journal of Language Modelling* 5. 57–106.
Hao, Yiding & Samuel Andersson. 2019. Unbounded stress in subregular phonology. *SIGMORPHON* 16. 135–143.
Hao, Yiding & Dustin Bowers. 2019. Action-sensitive phonological dependencies. *SIGMORPHON* 16. 218–228.
Heinat, Fredrik. 2006. *Probes, pronouns and binding in the Minimalist program*. University of Lund Doctoral Dissertation.
Ji, Jing & Jeffrey Heinz. 2020. Input strictly local tree transducers. In *Language and automata theory and applications: 14th International conference, LATA 2020, Milan, Italy*, vol. 12038 of LNCS, 369–381.
Jurgec, Peter. 2011. *Feature spreading 2.0: A unified theory of assimilation*. University of Tromsø Doctoral Dissertation.
Kayne, Richard S. 1994. *The antisymmetry of syntax*. Cambridge, MA: MIT Press.
Kobele, Gregory M., Sabrina Gerth & John T. Hale. 2013. Memory resource allocation in top-down Minimalist parsing. In Glyn Morrill & Mark-Jan Nederhof (eds.), *Formal grammar: 17th and*

18th International conferences, FG 2012, Opole, Poland, August 2012, Revised Selected Papers, FG 2013, Düsseldorf, Germany, August 2013, 32–51. Heidelberg: Springer.

Lambert, Dakotah, Jonathan Rawski & Jeffrey Heinz. 2021. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling* 9. 151–194.

Mayer, Connor. 2021. Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars. *SCiL* 4. 39–50.

McCloskey, James. 2001. The morphosyntax of wh-extraction in Irish. *Journal of Linguistics* 37. 67–100.

Pasternak, Robert & Thomas Graf. 2021. Cyclic scope and processing difficulty in a Minimalist parser. *Glossa* 6. 1–34.

Pullum, Geoffrey K. & James Rogers. 2006. *Animal pattern-learning experiments: Some mathematical background*. Radcliffe Institute for Advanced Study, Harvard University Ms.

Pullum, Geoffrey K. & Barbara C. Scholz. 2005. Contrasting applications of logic in natural language syntactic description. In Petr Hájek, Luis Valdés-Villanueva & Dag Westerståhl (eds.), *Logic, methodology and philosophy of science*, 481–503. London: College Publications Department of Computer Science, KCL, The Strand.

Richards, Norvin. 2016. *Contiguity theory*. Cambridge, MA: MIT Press.

Rogers, James & Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information* 20. 329–342.

Savitch, Walter J. 1993. Why it might pay to assume that languages are infinite. *Annals of Mathematics and Artificial Intelligence* 8. 17–25.

Stepanov, Arthur. 2001. Late adjunction and Minimalist phrase structure. *Syntax* 4. 94–125.

Stepanov, Arthur. 2007. The end of CED? Minimalism and extraction domains. *Syntax* 10. 80–126.

Torr, John. 2017. Autobank: A semi-automatic annotation tool for developing deep Minimalist grammar treebanks. *EACL* 15. 81–86.

Zentz, Jason. 2015. Bantu *Wh*-agreement and the case against probe impoverishment. *ACAL* 44. 290–301.