Typological Implications of Tier-Based Strictly Local Movement

Thomas Graf
Stony Brook University
Department of Linguistics
100 Nicolls Road, Stony Brook, NY 11794, USA
mail@thomasgraf.net

Abstract

Earlier work has shown that movement, which forms the backbone of Minimalist syntax, belongs in the subregular class of TSL-2 dependencies over trees. The central idea is that movement, albeit unbounded, boils down to local mother-daughter dependencies on a specific substructure called a tree tier. This reveals interesting parallels between syntax and phonology, but it also looks very different from the standard view of movement. One may wonder, then, whether the TSL-2 characterization is linguistically natural. I argue that this is indeed the case because TSL-2 furnishes a unified analysis of a variety of phenomena: multiple wh-movement, expletive constructions, the *that*-trace effect and the anti-*that*-trace effect, islands, and wh-agreement. In addition, TSL-2 explains the absence of many logically feasible yet unattested phenomena. Far from a mere mathematical curiosity, TSL-2 is a conceptually pleasing and empirically fertile characterization of movement.

1 Introduction

A number of recent works (Graf 2018; Graf and De Santo 2019; Vu et al. 2019; Shafiei and Graf 2020; Graf and Kostyszyn 2021, a.o.) have investigated the complexity of syntax from a subregular perspective. One of the central findings is that movement as formalized in Minimalist grammars (Stabler, 1997, 2011) is *tier-based strictly 2-local* (TSL-2). This means that one can determine whether a movement step in a syntactic derivation is well-formed by I) constructing a tree tier that only contains material relevant to this kind of movement, and ii) checking mother-daughter configurations over this tree tier. But the specific system for movement is just one among many options that could be expressed in TSL-2. This raises questions about the empirical status of those other options, and whether they ever occur in language. In this paper, I argue that TSL-2 provides a broad typology of movement in the sense that every architectural option it provides is actually used with some movement-related phenomenon: multiple wh-movement, expletive constructions, the *that*-trace effect and the anti-*that*-trace effect, islands, and wh-agreement in Irish.

All of these phenomena, many of which are puzzling under the standard conception of Minimalist movement, fall out naturally from the TSL-perspective. The central argument is that if a cognitive system must be TSL-2 to handle movement, then we should expect to see these TSL-2 resources be used in a variety of ways. For instance, if the complexity of a system with movement and island constraints is not higher than that of the movement system without island constraints, additional explanations would be needed if no language ever exhibited island effects. Island constraints would inevitably be part of a linguistic ecosystem of free variation that is limited only by the available cognitive resources. Free variation limited to TSL-2 thus carves out a space within which we find some of the most surprising movement phenomena.

The paper is primarily a progression of case studies. The necessary background of TSL-2 movement is covered in §2. I then summarize earlier arguments by Graf and Kostyszyn (2021) that multiple wh-movement and expletive constructions are also TSL-2 (§3.1) before I turn to a new TSL-2 analysis of the *that*-trace effect (§3.2) that, among other things, hinges on the ability to put non-movers on movement tiers. I subsequently generalize this technique to also handle island effects (§4.1) and even wh-agreement (§4.2). All of this establishes that the space of TSL-2 dependencies includes a large variety of movement phenomena. But as discussed in §5, there is still overgeneration within this space, and some movement phenomena do seem to fall outside TSL-2. This should not prove to be an insur-

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 184-193.
Held on-line February 7-9, 2022

mountable challenge, though, and I propose several ways this could be addressed in future research.

2 Tier-based strictly local movement

The TSL-view of syntax builds on Minimalist grammars (MGs; Stabler, 1997, 2011), which are a formalization of Minimalist syntax. Every lexical item (LI) is annotated with features that determine its syntactic behavior. At the very least, each LI has some *category feature* $\mathtt{F}^-$, for instance in the noun $\mathtt{party} :: \mathtt{N}^-$. An LI may also have a string of *selector features* $\mathtt{F}_1^+ \cdots \mathtt{F}_m^+$ that determine which arguments it takes. An example would be the ditransitive verb $\mathtt{introduce} :: \mathtt{P}^+\mathtt{D}^+\mathtt{D}^+\mathtt{V}^-$ as in *John introduced Mary to Sue*. In addition, an LI may carry *licensor features* $\mathtt{f}_1^+ \cdots \mathtt{f}_n^+$, which provide a landing site for movement. In this paper, no LI will ever have more than one licensor feature — consider for instance the empty topicalization head $\varepsilon :: \mathtt{T}^+\mathtt{top}^+\mathtt{C}^-$, with a single licensor feature $\mathtt{top}^+$ that attracts a topicalized phrase. Finally, an LI may carry a set $\{\mathtt{f}_1^-, \ldots, \mathtt{f}_n^-\}$ of unordered *licensee features* (standard MGs assume that licensee features are also linearly ordered, but this is incompatible with the TSL-view of syntax; as is already implicit in Graf et al. 2016, the use of unordered licensee features does not alter the weak or strong generative capacity of MGs). Each licensee feature $\mathtt{f}^-$ on LI l indicates that the phrase headed by l moves to the closest landing site provided by an LI with $\mathtt{f}^+$. Each LI thus has a feature annotation of the form $\gamma\mathtt{F}^-\delta$, where γ is a (possibly empty) string of selector and licensor features, $\mathtt{F}^-$ is some category feature, and δ is either the empty string or a set of licensee features.

The syntactic derivations driven by those features can be succinctly represented in the form of a dependency tree as shown in Fig. 1. Movement in this formalism is *tier-based strictly 2-local* (TSL-2). A full definition of TSL-2 over trees is given in Graf and Kostyszyn (2021), but an intuitive discussion suffices for the purposes of this paper. I will first discuss TSL-2 over strings and then explain how this idea is generalized to trees.

TSL-2 over strings was first defined in Heinz et al. (2011) and is a generalization of the class *strictly 2-local* (SL-2). A stringset L is SL-2 iff there is a finite (and possibly empty) set G of forbidden bigrams such that L contains all strings s, and only those, such that $\rtimes s \ltimes$ does not contain any of G's forbidden bigrams. Here $\rtimes$ and $\ltimes$ are distinguished symbols that mark the beginning and end of the string, respectively. A well-known SL-2 stringset is $(ab)^+$, which contains ab, $abab$, and so on. It is SL-2 because it can be described by 5 forbidden bigrams (assuming that the alphabet is already limited to just a and b, otherwise additional bigrams are needed):

(1) a. $\rtimes\ltimes$: the string must contain at least one symbol
  b. $\rtimes b$: the string must not start with b
  c. aa: a must not be followed by a
  d. bb: b must not be followed by b
  e. $a\ltimes$: the string must not end with a

As another example, suppose that we only consider strings over the symbol a. Then all of the following stringsets are SL-2:

(2) a. the set of all strings over a ($G := \emptyset$)
  b. the set of all strings with no a (G contains at least $\rtimes a$ or $a\ltimes$)
  c. the set of all strings with at least one a ($G := \{\rtimes\ltimes\}$)
  d. the set of all strings with at most one a ($G := \{aa\}$)
  e. the set of all strings with exactly one a ($G := \{\rtimes\ltimes, aa\}$)

Intuitively, SL-2 models string dependencies that can be expressed as a finite number of constraints where one symbol restricts what other symbols may immediately occur to its right. TSL-2 over strings enriches SL-2 with a tier projection mechanism to allow for limited types of long-distance dependencies. Formally, tier projection is expressed as a function E_T that takes a string s as its input and deletes all symbols in s that do not belong to T. For example, $E_{\{a,b\}}$ would map $caccbac$ to aba. A stringset L is TSL-2 iff there is some finite tier alphabet T such that the image of L under E_T is SL-2. For instance, the set of strings over a and b that contain exactly one a is not SL-2, but it is TSL-2: we set $T := \{a\}$ and $G := \{\rtimes\ltimes, aa\}$. Then the well-formed $babbb$ has the well-formed tier a (or $\rtimes a \ltimes$ with explicit edge markers), whereas the illicit $babab$ has the ill-formed tier aa. TSL-2 thus captures the notion that long-distance dependencies are still local when irrelevant material is ignored.

TSL-2 over trees follows a very similar system of combining an SL-mechanism with a tier projection. Given a finite set T of tier symbols, one

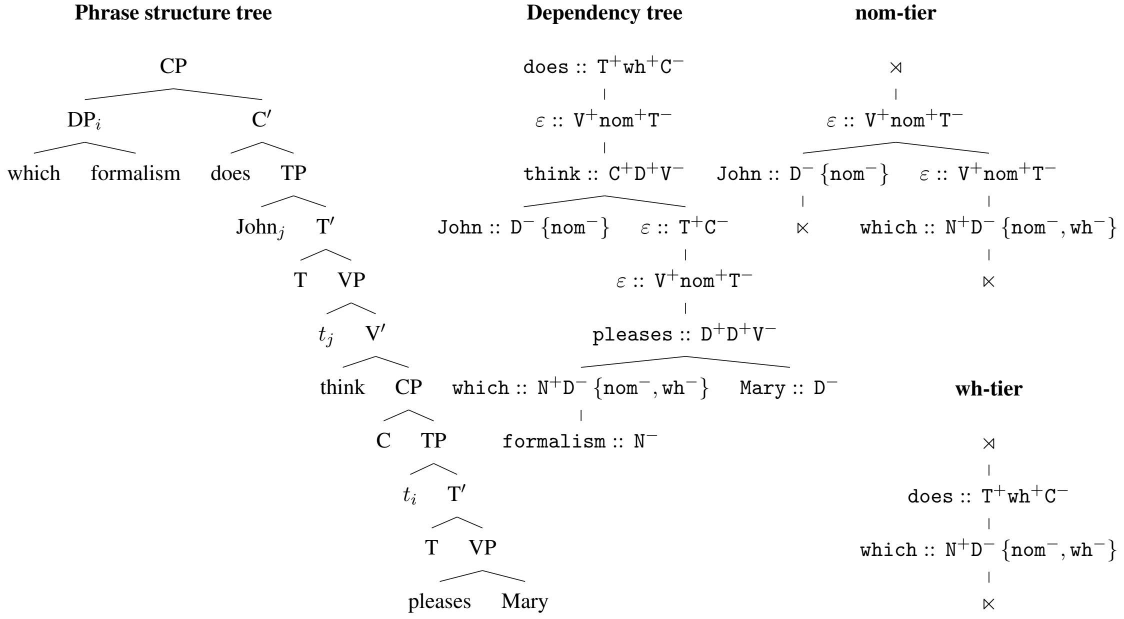

Figure 1: Phrase structure tree (left) with corresponding annotated derivation tree (middle) and two well-formed movement tiers (right), each one containing exactly one LI with $\mathtt{f}^-$ among the daughters of each LI with $\mathtt{f}^+$; note that intermediate movement of *which formalism* to Spec,CP of the embedded clause is not encoded via features; $\rtimes$ and $\ltimes$ on tiers will be omitted for the rest of the paper

removes from the tree all nodes whose labels do not belong to T, while preserving dominance relations between the remaining nodes. On the tree tier, each mother may restrict the shape of its daughters, similar to how in SL-2 over strings a symbol may restrict the shape of the symbol immediately following it. Formally, each tier symbol σ in T is associated with a stringset L_σ, and if a node on the tier is labeled σ, then its daughters on the tier must form a string that belongs to L_σ.

MG movement fits into this general system as follows: For each movement type `f` (`nom`, `wh`, and so on) one removes all nodes from the dependency tree that do not carry at least one of $\mathtt{f}^+$ and $\mathtt{f}^-$. The result is the tree tier for `f` (cf. Fig. 1). In analogy to the string case, the tier also has a distinguished root $\rtimes$, and each leaf is made a mother of $\ltimes$. On the tier, each tier symbol σ is associated with a particular daughter stringset L_σ that is TSL-2: If n is a node on an `f`-tier and n has a label that includes $\mathtt{f}^+$, then the daughter string of n must contain exactly one node whose label includes $\mathtt{f}^-$. If n is labeled $\rtimes$, instead, then its daughter string must not contain any $\mathtt{f}^-$. This results in a system where both of the following hold for each `f`-tier: I) every $\mathtt{f}^+$-node has exactly one $\mathtt{f}^-$-daughter, and II) every $\mathtt{f}^-$-daughter has a $\mathtt{f}^+$-mother. That is exactly how movement behaves in MGs, making it "doubly TSL-2": it is TSL-2 over trees, and on each movement tier it holds for every node that its set of well-formed daughter strings is TSL-2.

But this TSL-2 view of movement allows for several alternatives of the same formal complexity. As previously illustrated in (2), TSL-2 can perform limited counting, distinguishing between 0, "at least 1", "at most 1", and "exactly 1". In standard MGs, the daughter string of an LI with $\mathtt{f}^+$ must contain exactly one LI with $\mathtt{f}^-$, but from the view of TSL-2 one could just as well require at least one $\mathtt{f}^-$, at most one, or none at all. In addition, $\mathtt{f}^+$ and $\mathtt{f}^-$ are meaningless symbols from the perspective of TSL-2, and thus there is no inherent reason why only LIs with those features should be present on a tier. And once these LIs appear on a tier, they could behave like LIs with $\mathtt{f}^+$ in that they put constraints on their daughters, or like LIs with $\mathtt{f}^-$ in that they can satisfy those constraints. The rest of this paper explores this typology of grammatical options carved out by TSL-2. I will show how varying these TSL-2 parameters yields various phenomena related to movement, which suggests that the TSL-2 characterization of movement isn't just a mathematical coincidence but touches on fundamental properties of movement.

3 Varying the number of dependents

I first consider the configurations that arise if one changes how many $\texttt{f}^{-}$ have to occur in the daughter string. I argue that this yields multiple wh-movement, optional movement, and the *that*-trace effect in English (including exceptions brought about by adjuncts). The first two were already discussed in Graf and Kostyszyn (2021), so I will sketch them only briefly.

3.1 Multiple wh-movement and optional movement

Multiple wh-movement refers to the phenomenon where multiple wh-phrases move to the left edge of the clause

(3) *Multiple wh-movement in Bulgarian* (Bošković, 2002, p.353)

[Ko$_i$ koga$_j$ [t_i voli t_j]]?
who whom loves

'Who loves whom?'

In terms of TSL, this can be analyzed as a relaxation of movement where the matrix C-head $\varepsilon :: \texttt{T}^{+}\texttt{wh}^{+}\texttt{C}^{-}$ still carries only one instance of $\texttt{wh}^{+}$, but its string of daughters on the wh-tier may contain any number of wh-movers with $\texttt{f}^{-}$, as long as it contains at least one (see Fig. 2). Since this is a weakening of the standard constraint ("exactly one" is equivalent "at least one and at most one"), the TSL-2 account of movement tells us that multiple wh-movement is unremarkable in the sense that a system that can require the presence of exactly one mover can also enforce the presence of at least one.

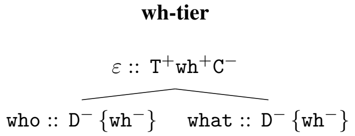

Figure 2: Example of wh-tier with multiple wh-movement

If, on the other hand, the requirement is loosened to "at most one", one gets a landing site that does not need a mover but can accommodate one if necessary — in other words, optional movement. Graf and Kostyszyn (2021) argue that this provides an alternative explanation of expletive constructions.

(4) a. A man is in the garden.
b. There is a man in the garden.

In (4a), the T-head $\varepsilon :: \texttt{V}^{+}\texttt{nom}^{+}\texttt{T}^{-}$ has a matching $\texttt{nom}$-tier daughter $\texttt{a} :: \texttt{N}^{+}\texttt{D}^{-}\ \{\texttt{nom}^{-}\}$, and movement takes place as usual. If *a* loses its licensee feature, one gets (4b) instead, where the T-head has no suitable daughter on the $\texttt{nom}$-tier, causing the unmatched $\texttt{nom}^{+}$ to be spelled out as the expletive *there*. Again a well-known movement phenomenon has a natural place in the TSL-2 formalism.

3.2 The *that*-trace effect

The *that*-trace effect refers to the phenomenon that even though English allows for long-distance extraction from an embedded clause, subjects may not be extracted if the complementizer is *that*. Curiously, this effect disappears if *that* is followed by an adverb (cf. Browning, 1996, p.238).

(5) a. Who$_i$ do you think (that) John should have met t_i?
b. Who$_i$ do you think (*that) t_i should have t_i met John?
c. Who$_i$ do you think (that) under normal circumstances t_i should have t_i met John?

This can be analyzed in various ways, e.g. as a string constraint against *that* t. But TSL can accommodate this phenomenon without additional machinery.

Let us ignore the effect of adverbs for now. Suppose that we construct a wh-tier in the usual manner to verify that there is a match between wh-mover and wh-landing site. But in addition, we also construct another tier whose job it is to further restrict the behavior of subjects, thus giving rise to the *that*-trace effect. This *that-trace tier* (TTT) contains all of the following: I) every LI with $\texttt{wh}^{+}$, II) every LI with both $\texttt{nom}^{-}$ and $\texttt{wh}^{-}$, and III) every C-head, including $\texttt{that} :: \texttt{T}^{+}\texttt{C}^{-}$. Only one constraint is active on TTT, namely that the complementizer *that* must not have any LI among its daughters that carries $\texttt{nom}^{-}$.

As shown in Fig. 3, this system correctly rules out the illicit *Who do you think that met John* while still allowing for well-formed counterparts that do not involve extraction of a subject wh-phrase. This account works thanks to the interaction of three factors. First, we can correctly pick out wh-subjects by their features $\texttt{nom}^{-}$ and $\texttt{wh}^{-}$, so that only subjects (but not objects) are projected onto TTT. Second, by also projecting $\texttt{wh}^{+}$ nodes we introduce a safety buffer on TTT that pushes wh-subjects out of the

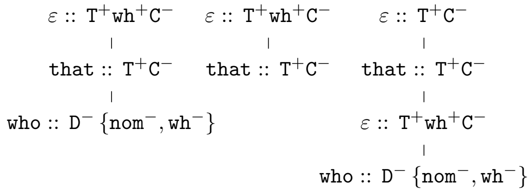

Figure 3: Ill-formed TTT for illicit *Who do you think that met John* (left) and well-formed TTTs for licit *Who do you think that John met* and *I know that Mary wondered who met Bill* (middle and right)

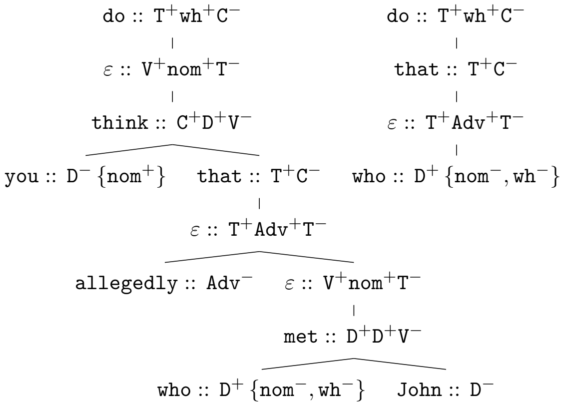

Figure 4: In the dependency tree for *who$_i$ do you think that allegedly t_i met John* (left), projecting $\sim$ creates a buffer between *that* and *who* (right).

daughter string of *that* if their wh-movement does not actually cross the complementizer. Finally, by projecting every C-head, including empty ones, we allow subject-wh phrases to cross *that* as long as their immediately containing clause has a different complementizer. This allows for well-formed examples such as the one below.

(6) Who$_i$ do you think that Mary said that John believes [C t_i met Bill]?

As the reader might have already noticed, the ameliorating effect of adverbs could be captured by projecting them onto TTT so that they separate subject wh-phrases from *that*. The big puzzle is how one wants to represent adverbs, which are adjuncts, in dependency trees. While the MG literature furnishes many different implementations of adjunction (see Frey and Gärtner 2002, Graf 2014, and Hunter 2015, a.o.), the easiest option in this case is *category-preserving selection*. That is to say, adjunction of some YP to XP is expressed as selection by an empty head $\varepsilon :: \texttt{X}^{+}\texttt{Y}^{+}\texttt{X}^{-}$ that projects another XP. This is illustrated in Fig. 4. Since no other empty heads ever seem to display the particular feature pattern $\texttt{T}^{+}\texttt{X}^{+}\texttt{T}^{-}$, the projection for TTT can correctly single out these TP-adjunction heads. But projecting TP-adjunction heads onto TTT can push the wh-subject out of the daughter string of *that*, and in this case TTT will be well-formed.

The reader may object that this is a highly stipulative proposal, but quite the opposite is the case. No stipulations are involved at all. TSL-2 carves out a space of options, and what this section shows is that both the *that*-trace effect and its exceptions are already part of this space. Individual points within the space may look highly peculiar, but the whole space itself is very natural.

Overall, then, the existence of the *that*-trace effect is unsurprising in the sense that it requires no additional machinery, assumptions, or stipulations beyond what is already furnished by TSL-2. It admittedly requires a very particular choice of tier projections and constraints on daughter strings, but this is simply one among myriads of possible combinations of tier projections and constraints. The very marked nature of TTT might actually serve as an explanation for why the *that*-trace effect is attested in very few languages.

In addition, the TSL-2 view also makes it less surprising that we find anti-*that*-trace effects with other movement types (Douglas, 2017):

(7) I met [the woman]$_i$ *(that/who) t_i saw John.

TSL-2 can treat this as a simple variation of the *that*-trace effect such that i) we now operate on a TTT-like variant of the `rel`-tier, where `rel` is the movement feature that extracts head nouns from their relative clause, and ii) it is the unpronounced complementizer, not the pronounced one, that bans wh-subjects in its string of tier daughters. A long-standing puzzle reduces to accidental variation across tiers.

4 Opaque and transparent tier buffers

The account of the *that*-trace effect uses two tricks. By setting the number of allowed elements of a specific type to 0, we enforce the absence of those elements in specific daughter strings. But at the same time, additional elements are projected to act as a kind of *tier buffer* that blocks the constraint from applying in specific circumstances. In this section, we will see two additional uses of buffers. Buffers that interrupt licensing conditions give rise to islands (§4.1). Buffers that daisychain licensing conditions give rise to wh-agreement (§4.2).

4.1 Islands: opaque tier buffers

Islands are constituents that are opaque to (certain types of) extraction. A phrase contained within an island may not move to positions outside that island. Some common examples of islands in English are shown in (8).

(8) a. *Well-formed extraction without island*
What$_i$ did John complain that Mary brought t_i to the party?

b. *Adjunct island*
*What$_i$ did John complain because Mary brought t_i to the party?

c. *Whether island*
*What$_i$ did John wonder whether Mary brought t_i to the party?

d. *Complex NP island*
*What$_i$ did John complain about the fact that Mary brought t_i to the party?

e. *Relative clause island*
*What$_i$ did John complain about the person that brought t_i to the party?

The specific configurations that induce island effects vary across languages and even speakers, and so does what types of movement are subject to island effects (see Szabolcsi and Lohndal 2017 and references therein). Hence any good theory of movement must solve multiple puzzles: I) why do island effects exist in the first place, II) why aren't all movement types subject to the same island effects, and III) why aren't island effects uniform across languages and speakers?

The TSL view of movement provides natural answers to all those questions, and it does so without any extra stipulations. Quite simply, islands arise when a tier contains elements that cannot satisfy the need of nodes with $\texttt{f}^{+}$ for a daughter with $\texttt{f}^{-}$. Just like seemingly irrelevant nodes on a tier prevent a constraint violation with the *that*-trace effect, with islands such nodes prevent constraint satisfaction.

Consider the dependency tree for sentence (8d) with a complex NP island, as depicted in Fig. 5. The observed island effect is unexpected under the standard tier projection for wh-tiers, which projects all LIs, and only those, that carry $\texttt{wh}^{+}$ and $\texttt{wh}^{-}$. As can be seen in Fig. 5 (middle), the resulting tier is well-formed. With the default tier projection, then, the complex NP island effect is entirely unexpected. But there is nothing that prevents English from using a different tier projection where the wh-tier contains not just LIs that carry $\texttt{wh}^{+}$ or $\texttt{wh}^{-}$. The wh-tier could just as well contain complex NPs, which are exactly those LIs whose feature annotation starts with $\texttt{C}^{+}\texttt{N}^{-}$. The resulting tier, depicted in Fig. 5 (right), now has the two movement nodes separated by `fact` :: $\texttt{C}^{+}\texttt{N}^{-}$. Since this LI carries no movement features at all, $\texttt{f}^{+}$ is missing a matching $\texttt{f}^{-}$ among its daughters. This renders the tier ill-formed, and a single ill-formed tier is sufficient to rule out the entire derivation.

Other island constraints similarly reduce to the projection of specific LIs that interrupt licensing relations. Adjunct islands arise whenever adjuncts are projected (in contrast to the *that*-trace effect, here one has to project the adjunct itself instead of the empty adjunction head as extraction from the adjoinee is still permitted). This also includes relative clause islands, which can be analyzed as NP and DP adjuncts. Similarly, *whether* islands are the result of projecting the LI `whether` :: $\texttt{T}^{+}\texttt{C}^{-}$, which once again poses no computational challenges. The same strategy even accounts for subject islands.

(9) *Subject island constraint*
[Which student]$_i$ did [the advisor of t_i] study island constraints?

As long as all subjects carry some $\texttt{nom}^{-}$ that enforces (overt or covert) subject movement, and as long as $\texttt{nom}^{-}$ can only occur on subjects, the subject island constraint is the result of projecting every LI with $\texttt{nom}^{-}$ on every tier. We see, then, that TSL readily accommodates island effects because there is no *a priori* ban against projecting specific LIs onto movement tiers, including those with no movement features at all.

The TSL account also explains why island effects can vary across movement types, and why they aren't universal across languages and speakers. Since every movement tier uses its own tier projection, there is no reason why all tier projections should project the same LIs. By extension, there is also no reason why all languages have to have exactly the same tier projections for every movement type. Note that this even includes exceptions to island constraints, e.g. *Truswell sentences* (Truswell, 2007).

(10) a. * [Which car]$_i$ did John drive Mary crazy while he tried to fix t_i?

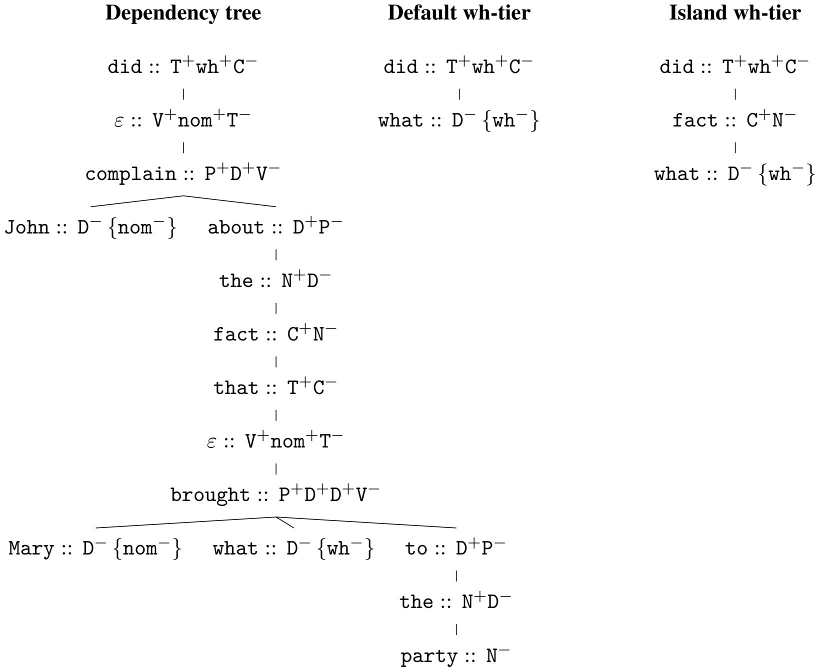

Figure 5: Dependency tree for the complex NP island in (8d) and two choices of wh-tier

b. [Which car]$_i$ did John drive Mary crazy while trying to fix t_i?

Under the plausible assumption that the category feature $\texttt{T}^-$ should actually be split into $\texttt{T}^-_{\mathit{fin}}$ and $\texttt{T}^-_{\mathit{inf}}$ for finite and infinitival TPs, respectively, this split boils down the fact that `while` :: $\texttt{V}^+\texttt{T}^-_{\mathit{fin}}$ projects onto movement tiers whereas `while` :: $\texttt{V}^+\texttt{T}^-_{\mathit{inf}}$ does not. For TSL-2, Truswell sentences are no more remarkable than the fact that *whether* induces islands while *if* does not.

(11) a. * What$_i$ did John wonder whether Mary brought t_i to the party?

b. What$_i$ did John wonder if Mary brought t_i to the party?

Without additional restrictions, TSL allows for free variation in tier projections, and this explains the variability we find across movement types, languages, and speakers.

What more, TSL provides a natural upper bound on the complexity of islands. All of the following are logically feasible island constraints, yet none of them are attested:

(12) a. *Gang-up island effects*
A mover can escape n islands, but not $n+1$.

b. *Configurational island effects*
XP is an island iff it is inside an embedded clause.

c. *Cowardly island effects*
XP is an island iff there are at least n XPs in the same clause.

d. *Narcissist island effects*
XP is an island iff there are no other XPs in the same clause.

e. *Rationed island effects*
At most n phrases per clause can be an island.

f. *Discerning islands*
XP is an island only for movers that contain a PP.

What they all have in common is that the TSL tier projection, which only considers individual nodes/LIs and never their structural context, cannot project nodes in a manner that would match these island effects. A cognitive device that is limited to TSL-2 is simply incapable of expressing such constraints on movement.

4.2 Wh-agreement: Transparent tier buffers

We just saw that islands arise from tier nodes that lack both $\texttt{f}^+$ and $\texttt{f}^-$ and thus interrupt all licensing relations related to those features. But one could

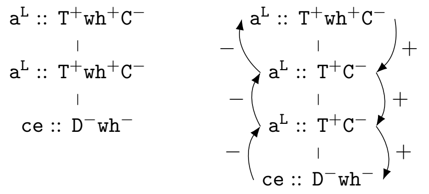

Figure 6: Left: ill-formed tier for (13) where the embedded complementizer carries `wh`$^+$; Right: tier with licensing relations if *aL* acts as if it had both `wh`$^+$ and `wh`$^-$

also imagine the opposite: a node that lacks both features yet acts as if it had both. More than just a technical curiosity, this allows for a novel analysis of wh-agreement in Irish (McCloskey, 1979, 2001) and Chamorro (Chung, 1998), among others.

The example below (McCloskey, 2001, p.94) shows how complementizers in Irish change their phonetic exponent from *go* to *a* or *aL* if a wh-phrase moves across them (the phenomenon also happens with other kinds of movement, but the proposed TSL-2 analysis generalizes to those, too).

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

Crucially, this happens to all complementizers along the movement path, no matter how many there are.

The alternation in the first complementizer is easily captured by having two separate lexical entries `a`$^{\mathtt{L}}$ :: `T`$^+$`wh`$^+$`C`$^-$ and `go` :: `T`$^+$`C`$^-$ that differ in the presence of `wh`$^+$. But the complementizer of the embedded clause cannot carry `wh`$^+$ — if it did, the wh-tier would be ill-formed (see Fig. 6, left). How, then, can TSL possibly capture the movement-sensitive distribution of *aL* and *go*?

As with the *that*-trace effect and islands, the answer is that projection onto an `f`-tier need not be limited to LIs with `f`$^+$ or `f`$^-$. Suppose that both `a`$^{\mathtt{L}}$ :: `T`$^+$`C`$^-$ and `go` :: `T`$^+$`C`$^-$ project onto the wh-tier, but exhibit very different types of behavior on this tier. The default complementizer *go* acts like an island for wh-movement: if a clause is headed by *go*, no phrase can wh-move out of it. Hence *go* can never occur along a wh-movement path.

The agreeing complementizer *aL*, on the other hand, behaves as if it carried both `wh`$^+$ and `wh`$^-$. Because *aL* acts as if it carried `wh`$^+$, it requires a negative daughter with `wh`$^-$. But since *aL* also acts as if it carried `wh`$^-$, the daughter can be just another instance of *aL*. Eventually, though, the lowest element must be a wh-mover that only carries `wh`$^+$ and thus puts no requirements on its daughter string. At the same time, the fact that *aL* behaves as if it carried `wh`$^-$ also means that it must have a mother with `wh`$^+$. Again this can be another instance of *aL* because *aL* also acts like `wh`$^+$. But eventually there has to be a node at the very top that only carries `wh`$^+$ and no `wh`$^-$ — in other words, a wh-landing site with `wh`$^+$. Putting all of this together, a sequence of one or more instances of *aL* can only occur sandwiched between `wh`$^+$ and `wh`$^-$, i.e. along a wh-movement path.

The TSL-2 account of Irish thus posits a complementizer *go*, which can never occur along a wh-path, and a separate complementizer *aL*, which can occur only along a wh-path. What is usually analyzed as a single complementizer agreeing with a successive-cyclic wh-mover is actually two distinct complementizers that are in complementary distribution due to how they differ in their behavior on the wh-tier.

5 Discussion

We have seen that the TSL-2 characterization not only captures movement in a simple manner, it also accounts for a number of seemingly unrelated phenomena that arise with movement: multiple wh-movement (§3.1), optional movement and expletive constructions (§3.1), *that*-trace effects and anti-*that*-trace effects (§3.2), adjunct islands, complex NP islands, *whether* islands, relative clause islands, subject islands (§4.1), and finally wh-agreement (§4.2). Most importantly, these phenomena require no additional machinery or assumptions. A cognitive system that can handle the TSL-2 dependencies of standard movement has all the computational resources to also handle these phenomena. If we assume free variation in the lexicon and the tier projections, each one of these phenomena is bound to eventually show up in some language. But this is also the shortcoming of the current TSL-2 perspective: languages are much more principled and systematic than the free variation account predicts.

If tier projections vary freely across tiers, languages, and speakers, why then do we find no languages that completely lack the adjunct island constraint? Why do even those languages where relative clauses do not induce island effects still show processing effects that suggest that they are islands (Tutunjian et al., 2017)? Why isn't there a language where the facts for Truswell sentences are exactly the other way around, with infinitival T opaque to extraction whereas finite T allows for it? And why isn't there an analogue of the *that*-trace effect that targets objects instead of subjects? While TSL-2 rules out many unnatural kinds of movement dependencies (cf. (12)), it still allows for any kind of unnatural phenomenon that can be expressed as the projection of a finite subset of the lexicon, no matter how idiosyncratic that subset.

This shows that TSL-2 in its current form still overgenerates and is too lax a restriction on the typology of island constraints. However, the TSL tier projection also provides a natural locus for addressing this overgeneration. What TSL-2 needs is a theory of tier projections. This could come in the form of substantive universals, perhaps coupled with abstract notions like monotonicity (Graf, 2019, 2020; Moradi, 2019, 2020, 2021). Alternatively, there may be restrictions on the relation of tiers to each other, akin to the constraints on harmony tiers identified by Aksënova and Deshmukh (2018). The key point is that while the issue is still open, TSL already furnishes a path towards its solution — in contrast to other analyses of islands, which usually have to add on new machinery to account for unexpected variation rather than pruning down the already predicted typology.

That said, TSL-2 isn't a uniform account of all attested movement constraints, either. As far as I can tell, some conditions on movement simply are beyond the purview of TSL-2, e.g. freezing effects and the Coordinate Structure Constraint. Whether this is an isufficiency of TSL-2 or my own analytical abilities remains to be seen, and it may still be possible to come up with, say, a TSL-3 account of freezing effects. In addition, there are alternative models of subregular dependencies in syntax, foremost constraints on string representations obtained from dependency trees (Graf and Shafiei, 2019; Shafiei and Graf, 2020) and the class of constraints recognizable by sensing tree automata (Graf and De Santo, 2019). Even though these were developed for constraints that do not directly regulate movement, for instance Principle A of binding theory, there is no obvious reason why well-attested conditions on movement cannot come from this class instead. Again the logic is that if these computational resources are already available to handle phenomena like Principle A, it would be surprising if this machinery were never applied to movement. Perhaps, then, TSL-2 covers a large portion of movement, but not the full space, with other subregular classes picking up the slack. Overall, TSL is far from the final word on movement, but it provides a surprisingly versatile starting point that can be refined in various ways (tier projection, going beyond TSL-2) to improve its empirical adequacy.

Conclusion

I have argued that the TSL-2 characterization of Minimalist movement is not a purely mathematical curiosity but an empirically fertile perspective that readily accommodates a large variety of phenomena related to movement. This is a unique conceptual advantage of TSL-2. Whereas other syntactic proposals require additional machinery to go from the basic mechanism of movement to multiple wh-movement, island effects, *that*-trace effects, and wh-agreement, all of them come for free with TSL-2. Any cognitive system capable of movement also has the computational resources to handle these phenomena. Similarly, TSL-2 also predicts that we should never see unnatural things like the *gang-up islands* from (12) because they are not TSL-2, whereas the non-existence of such islands is puzzling under standard Minimalist accounts. Despite all these advantages, TSL-2 is not the final word on movement because it predicts too much variation across movement types, languages, and speakers. Future work should strive to identify abstract properties of tier projections that separate natural from unnatural movement phenomena.

Acknowledgments

The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344. I thank the three anonymous reviewers for their detailed comments. Reviewer 3 in particular provided numerous thought-provoking observations and suggestions, many of which will find their way into follow-up work with less severe space constraints.

References

Alëna Aksënova and Sanket Deshmukh. 2018. Formal restrictions on multiple tiers. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 64–73.

Željko Bošković. 2002. On multiple wh-fronting. *Linguistic Inquiry*, 33:351–383.

Marguerite Browning. 1996. CP recursion and *that*-t effects. *Linguistic Inquiry*, 27:237–255.

Sandra Chung. 1998. *The Design of Agreement: Evidence from Chamorro*. Chicago University Press, Chicago.

Jamie Douglas. 2017. Unifying the *that*-trace and anti-*that*-trace effects. *Glossa*, 2:1–28.

Werner Frey and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in Minimalist grammars. In *Proceedings of the Conference on Formal Grammar*, pages 41–52.

Thomas Graf. 2014. Models of adjunction in Minimalist grammars. In *Formal Grammar 2014*, volume 8612 of *Lecture Notes in Computer Science*, pages 52–68, Heidelberg. Springer.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, pages 117–136.

Thomas Graf. 2019. Monotonicity as an effective theory of morphosyntactic variation. *Journal of Language Modelling*, 7:3–47.

Thomas Graf. 2020. Monotonicity in syntax. In *Monotonicity in Logic and Language*, volume 12564 of *Lecture Notes in Computer Science*, pages 35–53, Berlin, Heidelberg. Springer.

Thomas Graf, Alëna Aksënova, and Aniello De Santo. 2016. A single movement normal form for Minimalist grammars. In *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 and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 12–26, Toronto, Canada. Association for Computational Linguistics.

Thomas Graf and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist grammars. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*, pages 275–285.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 205–215.

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.

Tim Hunter. 2015. Deconstructing merge and move to make room for adjunction. *Syntax*, 18:266–319.

James McCloskey. 1979. *Transformational Syntax and Model Theoretic Semantics: A Case Study in Modern Irish*. Reidel, Dordrecht.

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

Sedigheh Moradi. 2019. *ABA generalizes to monotonicity. In *Proceedings of NELS 49*, volume 2.

Sedigheh Moradi. 2020. Morphosyntactic patterns follow monotonic mappings. In *Monotonicity in Logic and Language*, pages 147–165, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sedigheh Moradi. 2021. A formal restriction on gender resolution. In *All Things Morphology: Its Independence and Its Interfaces*, pages 41–54. John Benjamins, Amsterdam.

Nazila Shafiei and Thomas Graf. 2020. The subregular complexity of syntactic islands. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*, pages 272–281.

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. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.

Anna Szabolcsi and Terje Lohndal. 2017. Strong vs. weak islands. In Martin Everaert and H.C. Riemsdijk, editors, *The Wiley Blackwell Companion to Syntax, Second Edition*, pages 479–532.

Robert Truswell. 2007. Extraction from adjuncts and the structure of events. *Lingua*, 117:1355–1377.

Damon Tutunjian, Fredrik Heinat, Eva Klingvall, and Anna-Lena Wiklund. 2017. Processing relative clause extractions in Swedish. *Frontiers in Psychology*, 8:2118.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 267–276.