The Subregular Complexity of Syntactic Islands

Nazila Shafiei and **Thomas Graf**
Department of Linguistics
Stony Brook University
Stony Brook, NY 11794, USA
nazila.shafiei@stonybrook.edu

Abstract

We provide a formal framework for analyzing syntactic island effects from a subregular perspective. Key aspects of the syntactic representation are encoded as strings where precedence represents containment. Island effects then are expressed as constraints on the shape of these strings. The constraints fit in the class IBSP (Interval-Based Strictly Piecewise), which has been previously explored in subregular phonology. Consequently, the characterization of islands in terms of IBSP string constraints not only provides a computational upper bound on the inventory of feasible island effects, but also establishes a surprising link between syntax on the one hand and phonology on the other.

1 Introduction

The subregular program is concerned with analyzing the complexity of linguistic dependencies that are at most regular. The program has found great success in computational phonology (see Heinz 2018 and references therein), where it has resulted in a computational typology of phonological patterns and corresponding learning algorithms. Syntax, by virtue of being mildly context-sensitive, may seem far beyond the purview of the subregular program. But syntax is also subregular once one considers more suitable representations. Two routes have been explored: lifting subregular classes from strings to trees (Graf, 2018b; Vu et al., 2019), and putting string constraints on particular path languages of syntactic trees (Graf and Shafiei, 2019). Whereas the former has been mostly used in the analysis of structure building operations, the latter has been applied to syntactic constraints such as NPI-licensing.

This paper focuses on an area where these two aspects of syntax meet: island constraints. Island constraints impose additional restrictions on displacement, which in the tradition of Transformational grammar is equated with the operation *Move*. The shape of islands is narrowly circumscribed, indicating that they are very limited from a computational perspective. In this paper, we confirm this intuition. Island constraints are expressed as constraints over a path language where linear precedence in the string encodes (a specific notion of) containment. Given such a string representation, island constraints fall into the subregular class Interval-Based Strictly Piecewise (IBSP), which has been argued to play a central role in phonology (Graf, 2017, 2018a). At the same time, IBSP is sufficiently weak to rule out many unattested island constraints. Our paper thus makes several contributions: it deepens our understanding of subregular syntax, establishes parallels to phonology, and provides linguists with a computational theory of islands.

Due to space constraints, we focus largely on strong islands, and only on the canonical cases for most of them. We also investigate the *that*-trace constraint and the coordinate structure constraint, and we show that they cannot be handled in the system proposed here. This paper thus marks but the first step towards a fully articulated, empirically grounded theory of islands.

The discussion proceeds as follows: the preliminaries section (§2) discusses Minimalist grammars (§2.1), our string representation format (§2.2), and the subregular class IBSP (§2.3). Section 3 presents the central result that a number of (strong) island constraints follow a uniform IBSP pattern of very low complexity. We start with the adjunct island constraint (§3.1) and then generalize the analysis to wh-islands, the complex np constraints, the subject condition, and freezing effects (§3.2). Section 4 then explores the limits of IBSP over a-strings. On the one hand this allows us to correctly rule out many unattested island constraints, but it also means that the approach cannot handle all aspects of the *that*-trace constraint and the coordinate structure constraint. In addition, our approach currently lacks any notion of linguistic naturalness, which allows for some very odd (albeit computationally simple) island constraints (§5).

2 Preliminaries

The paper rests on several research traditions, which are briefly sketched in this section: Minimalist grammars as a formal model of syntax (§2.1), string representations for syntax (§2.2), and the subregular class of IBSP string languages (§2.3).

2.1 Minimalist Grammars

Since island constraints have mostly been studied in the generative tradition, we adopt Minimalist grammars (MGs; Stabler, 1997) as a formal model of syntax. MGs are a derivational grammar formalism for building tree structures by combining feature-annotated lexical items via the operations Merge and Move. Figure 1 gives a concrete example of this process. Only a few key aspects of MGs matter for this paper, in particular their feature system (see Stabler 2011 for a full discussion).

Each lexical item consists of a phonetic exponent and a string of features. There are four distinct types of features. *Category features* (X^-) and *selector features* (X^+) establish head argument relations via *Merge*. The other two feature types drive the operation *Move*. A *licensee feature* f^- indicate that the phrase headed by the lexical item undergoes f-movement, and the matching *licensor feature* f^+ indicates the landing site of f-movement. As in Minimalist syntax, movement is a mechanism for displacing subtrees of an already assembled tree, and movement always targets the closest available landing site (encoded in MGs via licensor features).

Given the special role of adjuncts in island constraints, we also adopt the adjunction mechanism of (Frey and Gärtner, 2002). Instead of a category feature, a lexical item l may carry an *adjunction feature* $\mathrm{X}^\sim$ which allows it to adjoin to an XP.

An MG's structure building process is usually represented as a derivation tree like the one in Fig. 1. But we will frequently represent derivation trees with the more compact format of dependency trees. The rightmost tree in Fig. 1 presents a concrete example.

2.2 String Representations for Syntax

Our investigation of island constraints will not operate directly over trees, but rather over strings that represent specific aspects of the tree structure. This follows recent work by Graf and Shafiei (2019), who analyze syntactic constraints such as NPI-licensing and Principle A as operating over strings that encode asymmetric c-command relations. A tree is well-formed iff it holds for every node n in the tree that the relevant string representation for n is well-formed with respect to the syntactic string constraints.

Graf and Shafiei (2019) choose a string representation that encodes both containment and a limited form of c-command (cf. Frank and Vijay-Shanker, 2001). These *augmented command strings* (or simply *c-strings*) can be defined in various ways, but the easiest option uses MG dependency trees. We adopt this definition but simplify it so that the resulting string representation only keeps track of containment. For this reason, we call these strings *ancestor strings* (or simple *a-strings*).

Definition 1 (A-strings). Let t be an MG dependency tree. If n is the root of t, then $\mathrm{as}(n) := n$. If n has mother m, then $\mathrm{as}(n) := n\ \mathrm{as}(m)$. ⌟

Example. In Fig. 1, $\mathrm{as}(\text{Mary} :: \mathrm{D}^-\mathrm{nom}^-) = \text{Mary} :: \mathrm{D}^-\mathrm{nom}^-\ \text{buy} :: \mathrm{D}^+\mathrm{D}^+\mathrm{V}^- \uparrow \varepsilon :: \mathrm{V}^+\mathrm{nom}^+\mathrm{T}^- \uparrow \text{did} :: \mathrm{T}^+\mathrm{wh}^+\mathrm{C}^-$. For increased readability, we may omit features and replace empty heads by their category. Then $\mathrm{as}(\textit{Mary}) =$ *Mary buy T did*. ⌟

The use of strings is a matter of mathematical convenience. The results obtained this way can be backported to subregular machinery that operates directly on dependency trees or derivation trees (Graf and De Santo, 2019). This will be discussed further in §5.

2.3 Subregular Complexity

Formal language theory has a rich tradition of studying proper subclasses of the regular string languages (McNaughton and Papert, 1971; Pin, 1997; Yli-Jyrä, 2005, a.o.). More recently, this line of work has been picked up and extended by computational phonologists (see Heinz 2018 and references therein). The class Interval-Based Strictly Piecewise (IBSP) was proposed as a linguistically natural unification of previously pro-

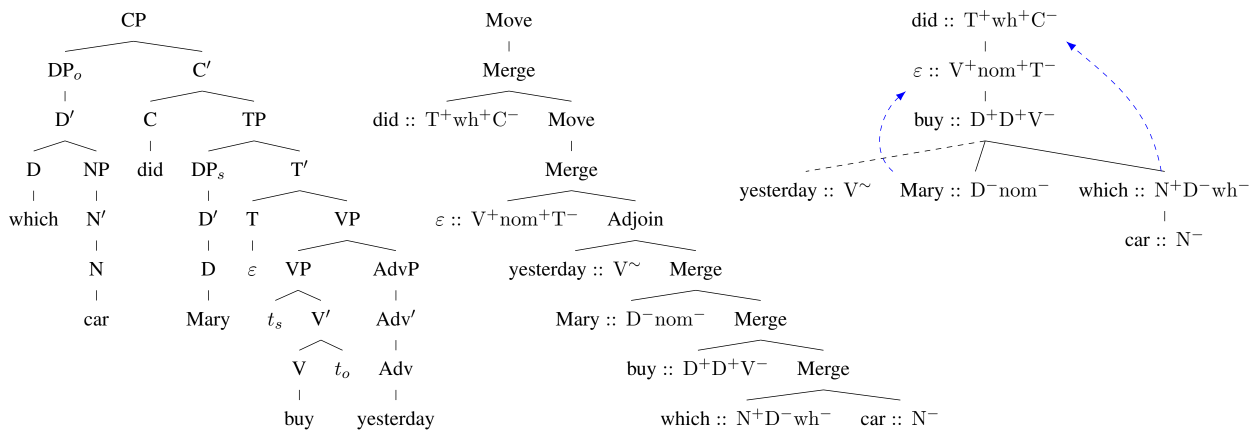


Figure 1: X′-tree, MG derivation tree, and equivalent dependency tree for *Which car did Mary buy yesterday*

posed classes for subregular phonology (Graf, 2017, 2018a). IBSP constitutes an approximate upper bound on string dependencies in phonology.

Definition 2 (*k*-val). A *segmented k-interval* ($k \geq 0$) over alphabet Σ, or simply *segmented k-val*, is a tuple $\langle L, R, F_i \rangle_{0 \leq i \leq k}$ such that

- $L, R \subseteq \Sigma \cup \{\varepsilon\}$ specify the *left edge* and *right edge*, respectively, and
- $F_i \subseteq \Sigma$ specifies the i-th *filler* slot. ⌟

Definition 3 (IBSP-*k*). Let Σ be some fixed alphabet and $\rtimes, \ltimes \notin \Sigma$ two distinguished symbols. An IBSP-k grammar over Σ ($k \geq 0$) is a pair $G := \langle i, S \rangle$, where i is a segmented k-val over $\Sigma \cup \{\rtimes, \ltimes\}$ and $S \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$ is a set of forbidden k-grams. A string $s \in \Sigma^*$ is generated by G iff there is no k-gram $u_1 \cdots u_k \in S$ such that $\rtimes^k s \ltimes^k$ is a member of the language

$$(\Sigma \cup \{\rtimes, \ltimes\})^* \cdot L \cdot F_0^* \cdot \{u_1\} \cdot F_1^* \cdot \{u_2\} \cdot \ldots \cdot F_{k-1}^* \cdot \{u_k\} \cdot F_k^* \cdot R \cdot (\Sigma \cup \{\rtimes, \ltimes\})^*$$

The language $L(G)$ is the set of all $s \in \Sigma^*$ that are generated by G. A stringset L is IBSP-k iff $L = L(G)$ for some IBSP-k grammar G. ⌟

In the definition above, * represents the usual Kleene closure. The symbol $\cdot$ denotes string concatenation, lifted to sets: $A \cdot B := \{ab \mid a \in A, b \in B\}$.

Following Graf and Shafiei (2019), we can use IBSP grammars over strings to regulate the shape of trees.

Definition 4 (IBSP over trees). Let G be an IBSP grammar, t an MG dependency tree, and f_t a total function from nodes of t to strings. Then t is well-formed with respect to G iff it holds for all lexical items l in t that $f_t(l)$ is generated by G. ⌟

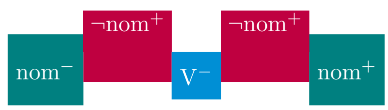
Example. Suppose that nom-movement is forbidden out of VPs. Over a-strings, this corresponds to the requirement that no V-head may occur between a node with nom$^-$ and the closest node with nom$^+$ (because movement in MGs always targets the closest head with a matching feature). This can be expressed as the following IBSP-1 grammar:

$$\begin{aligned} L &:= \{l \mid l \text{ carries nom}^-\} \\ F_0 &:= \{l \mid l \text{ does not carry nom}^+\} \\ F_1 &:= \{l \mid l \text{ does not carry nom}^+\} \\ R &:= \{l \mid l \text{ carries nom}^+\} \\ S &:= \{l \mid l \text{ carries V}^-\} \end{aligned}$$

Then as(*Mary*) $=$ *Mary buy T did* will be deemed illicit because it matches a forbidden pattern with $L :=$ *Mary*, $F_0^* := \varepsilon$, $F_1^* := \varepsilon$, $R :=$ *T*, and *buy* $\in S$. Consequently, the dependency tree is not well-formed, either. ⌟

It is often convenient to represent IBSP grammars in a more visual format. The example grammar above corresponds to the diagram below.

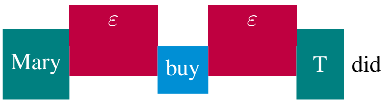
(1) *Graphical representation of an IBSP grammar*



The outermost vertical boxes represent the left and right edge, respectively. The square in the middle represents a position of the forbidden k-grams

— since the example grammar uses forbidden unigrams, there is only one such square. The vertically offset boxes represent the fillers, in this case F_0 and F_1. We use features to as a shorthand for the set of lexical items that carry this feature. For instance, nom^- denotes the set of all lexical items carrying nom^-. The expression $\neg\mathrm{nom}^+$ denotes the of all lexical items that do not carry the relevant feature, in this case nom^+. This visual format can also be used to show that a string is ill-formed.

Example. Recall that $\mathrm{as}(\textit{Mary}) = \textit{Mary buy T did}$ is illicit. We can show this by giving a specific instantiation of the interval and the k-gram in the string.



This disagram conveys the same information as the formal description in the previous example. ⌟

In the next section, we use this machinery to analyze syntactic island effects from a subregular perspective. We show that strong islands follow a fixed IBSP pattern over a-strings that is exceedingly simple.

3 Strong Islands over A-Strings

The notion of syntactic islands originates from Transformational Grammar (Ross, 1967). From the perspective of MGs, a constituent C is an island iff no phrase contained by C may have a licensee feature checked by a matching licensor feature outside C. A distinction is commonly made between *strong islands* and *weak islands*. Strong islands limit movement irrespective of whether the mover is an argument or an adjunct. Weak islands, on the other hand, limit adjunct movement but not argument movement. We will focus mostly on strong islands in this paper. We first analyze the adjunct island constraint (§3.1) as an IBSP perspective over a-strings, and then show how the same template can be used for several other strong island effects (§3.2).

3.1 Adjunct Island Constraint

The adjunct island constraint is arguably the most robust case of a strong island. It is illustrated by the contrast in (2).

(2) *Adjunct island constraint*

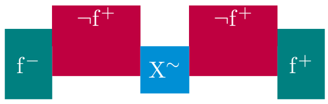
a. Which car did John complain [$_{\mathrm{CP}}$ that he can't fix _]?

b. * Which car did John complain [$_{\mathrm{CP}}$ because he can't fix _]?

In both (2a) and (2b) the wh-phrase *which car* moves out of an embedded clause. But in (2a) the embedded clause is an argument of the verb *complain*, whereas it is an adjunct in (2b). That movement is allowed out of the argument clause but not the adjunct clause is referred to as the adjunct island constraint.

The adjunct island constraint can be easily expressed as an IBSP constraint on a-strings. In fact, it uses the template we already encountered in §2.3, except that the set of forbidden unigrams consists of all adjuncts rather than all verbs. As explained in §2.1, we adopt the proposal of Frey and Gärtner (2002) that every adjunct carries an adjunct feature $\mathrm{X}^\sim$ that allows it to adjoin to XPs. The IBSP grammar for the adjunct island constraint thus corresponds to the following template.

(3) *IBSP-1 grammar for adjunct islands*



Note that this template actually represents multiple IBSP grammars as f must be correctly instantiated for each movement feature: nom for subject movement, wh for wh-movement, top for topicalization, and so on. If all of those were put inside a single IBSP grammar, then one lose the fact that the left edge and the right edge must be opposite polarities of the same feature — a nom^- for the left edge could be paired up with an wh^+ as the right edge. Since IBSP lacks a direct means of coordinating left and right edges like this, we instead have to posit a separate grammar for each movement feature in order to correctly enforce the adjunct island constraint for that specific movement type.

Example. Figure 2 shows the MG dependency tree for *which topic did you leave because Mary talked about* _. This sentence contains illicit wh-movement out of an adjunct. Now consider the a-string for *which*, with the relevant features indicated in square brackets: $\mathrm{as}(\textit{which}) = \textit{which}[\mathrm{wh}^-]$ *talked about T because*$[\mathrm{V}^\sim]$ *leave T did*$[\mathrm{wh}^+]$. As shown by the diagram below, this a-string is ill-formed with respect to the IBSP grammar in (3) (assuming $\mathrm{f} := \mathrm{wh}$).

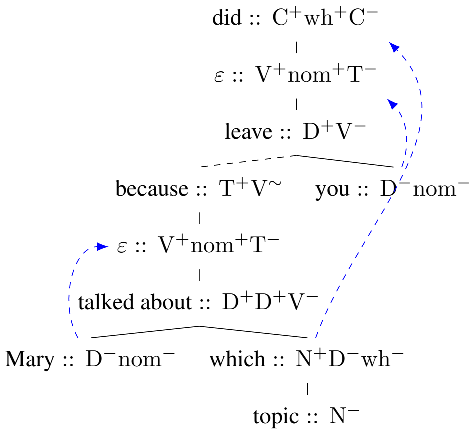
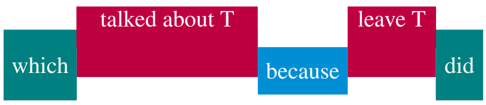


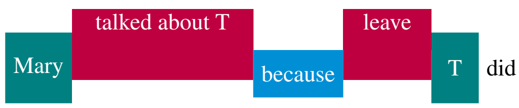
Figure 2: Adjunct island violation



As the a-string is illicit, the whole sentence is ill-formed. ⌟

The reader may wonder why the template explicitly forbids f^+ as fillers. This ensures that the right edge is always the closest f^+, which is the one targeted for movement by the lexical item with f^- in the left edge. Without this restriction, the IBSP grammar would incorrectly rule out well-formed movement patterns.

Example. Consider once more the example sentence *which topic did you leave because Mary talked about _* as depicted in Fig. 2. This sentence contains two instances of nom-movement, both of which are well-formed. But now consider the IBSP grammar regulating nom-movement. Suppose that this grammar allows for lexical items with nom^+ to appear in the filler slots. Then this grammar would incorrectly rule out $\mathrm{as}(Mary) =$ *Mary*$[\mathrm{nom}^-]$ *talked about T*$[\mathrm{nom}^+]$ *because*$[\mathrm{V}^\sim]$ *leave T*$[\mathrm{nom}^+]$ *did*.



⌟

The reader should also keep in mind that the use of $\mathrm{X}^\sim$ is just a notational shorthand for specifying a list of lexical items. One can remove some items from this set to allow for exceptions to the adjunct island constraint, such as the ones noted by Truswell (2007).

(4) a. * Which car did John drive Mary crazy [while he tried to fix _]?

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

Assuming a distinction between finite T-heads (T^-) and other T-heads (T^-_{inf}), we can account for this by excluding while :: $\mathrm{T}^+_{inf}\mathrm{V}^\sim$ from the list of forbidden lexical items.

In sum, the adjunct island constraint can be handled by a very simple and intuitive IBSP grammar (or rather, a collection of such grammars, one for each movement type). From a formal perspective, this IBSP grammar looks very similar to the IBSP treatment of blocking effects in phonology. In phonology, an intervening consonant cluster may block long-distance harmony. In syntax, an intervening head with an adjunction feature interrupts the dependency between an f^- and an f^+. The existence of the adjunct island constraint thus becomes a bit less mysterious: it is very simple from a computational perspective, and it employs a general blocking mechanism that also seems to be active in other parts of language.

3.2 Other Strong Islands

Besides the adjunct island constraint, the class of strong islands also includes wh-islands, complex NPs, and subjects. The corresponding constraints are illustrated below.

(5) *Wh-island constraint*

a. Which movie did John say that Mary liked _?

b. * Which movie did John wonder whether Mary liked _?

(6) *Complex NP constraint*

a. What did you say [that John bought _]?

b. * What did you hear rumors [that John bought _]?

(7) *Subject condition*

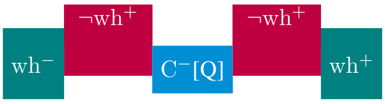
a. Who did John write [a story about _]?

b. * Who was [a story about _] written by John?

These all use minor variations of the template for the adjunct island constraint.

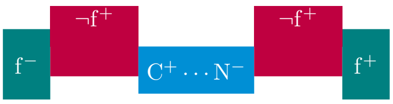
Let us start with the wh-island constraint. Here it suffices to make two changes. Since most types of movements, e.g. topicalization, are not affected by this constraint, we limit the possible instantiations for f^- and f^+ to just wh^- and wh^+, respectively. Then the list of blockers is changed from adjuncts to all elements that induce wh-islands. These are commonly taken to be all C-heads that have some kind of question semantics, including *whether*, *how*, and *if*. We denote this set $\mathrm{C}^-[\mathrm{Q}]$.

(8) *IBSP-1 grammar for wh-islands*



Next we turn to the complex NP constraint. This one, too, uses the basic template of the adjunct island constraint, but we once again have to change the list of blockers. In the complex NP constraint, the blocking is not done by an adjunct, but by a more complex structural configuration: movement out of a CP is illicit if the CP is the argument of a noun. Thanks to the MG feature calculus, we can rephrase this as a ban against moving out of an NP that selects a CP,[1] which means that the set of blockers contains all lexical items, and only those, that contain a selector feature C^+ and a category feature N^-. We denote this set of lexical items by $\mathrm{C}^+ \cdots \mathrm{N}^-$.

(9) *IBSP-1 grammar for complex NP constraint*



The reader is invited to verify that this grammar correctly rules out the sentence *what did you hear rumors that John bought*, which is depicted in Fig. 3.

This leaves us with the subject condition, which can actually be regarded as an instance of what is known as *freezing effects*. This describes the phenomenon that once a phase XP has undergone movement, it becomes opaque to extraction. Any mover inside XP has to move out of the phrase before it starts moving. From the perspective of MGs, this can be rephrased as a constraint on the distribution of movement features. Let f_1^- and g_1^-

[1]Our feature-based interpretation of the complex NP constraint is actually stronger than the original version. Suppose that the NP selects a CP as its complement and some XP as its specifier. The complex NP constraint as originally stated would allow the XP to be extracted, whereas our version does not. As far as we have been able to determine, there are no nouns that take two arguments in this configuration, let alone one where the XP then is allowed to undergo movement.

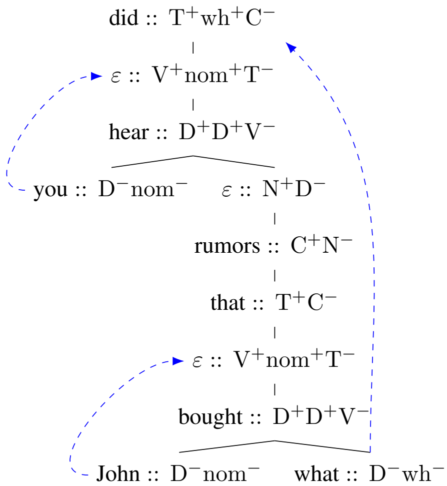
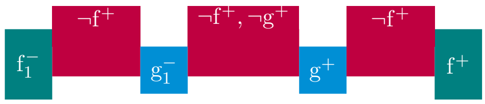


Figure 3: Violation of the complex NP constraint

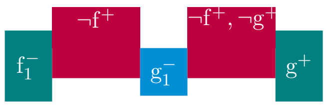
denote lexical items whose first movement feature is f^- and g^-, respectively. If the phrase headed by g_1^- contains f_1^-, then the target of f_1^- must be contained by the target of g_1^-. We can capture this generalization by moving from an IBSP-1 grammar to an IBSP-2 grammar (or rather, a collection of such grammar for every possible choice of f^- and g^-).

(10) *IBSP-2 grammar for freezing effects*

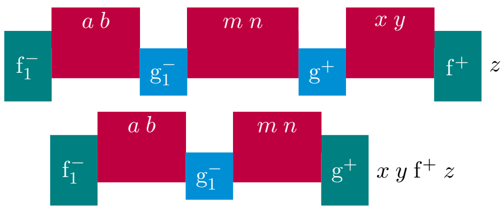


The step up from IBSP-1 to IBSP-2 makes freezing effects appear more complex. But it is actually possible to get the same effect just with an IBSP-1 grammar. The trick is to make g^+ the right edge of the k-val rather than f^+.

(11) *IBSP-1 grammar for freezing effects*



Example. Consider the abstract a-string $\mathrm{f}_1^-\ a\ b\ \mathrm{g}_1^-\ m\ n\ \mathrm{g}^+ x\ y\ \mathrm{f}^+\ z$. Both grammars correctly rule it out as illicit.



Similarly, both grammars agree that the minimally different $\mathrm{f}_1^-\ a\ b\ \mathrm{g}_1^-\ m\ n\ \mathrm{f}^+ x\ y\ \mathrm{g}^+\ z$ is well-formed. ⌟

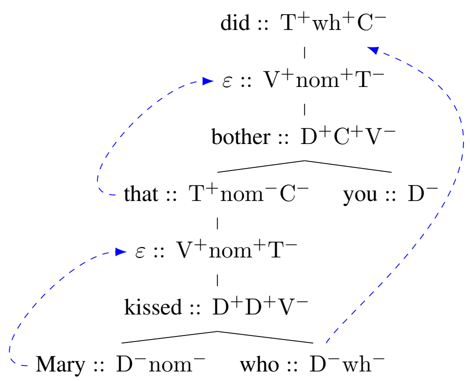
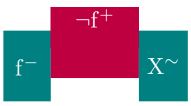


Figure 4: Violation of the subject condition

Both grammars also agree that the tree in Fig. 4 is illicit because of the ill-formed a-string of *who*.

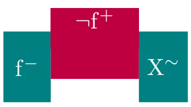
Note that we can apply the same kind of truncation strategy to the IBSP grammars for the other island constraints. This effectively reduces their complexity of IBSP-1 to IBSP-0. As laid out in Def. 3, an IBSP-0 grammar consists only of the left edge L, the right edge R, and a single filler F_0 inbetween. The set of forbidden k-grams is immaterial as every string is ruled out that matches $(\Sigma \cup \{\rtimes, \ltimes\})^* \cdot L \cdot F^* \cdot R \cdot (\Sigma \cup \{\rtimes, \ltimes\})^*$.

(12) *IBSP-0 grammar for adjunct islands*



(13) *IBSP-0 grammar for wh-islands*

(14) *IBSP-0 grammar for complex NP constraint*



In sum, all four island constraints can be captured with very simple IBSP-1 grammars (or even IBSP-0 grammars) over a-strings. Adjunct islands, wh-islands, and the complex NP constraint all follow the very same pattern. Subject islands, as a specific subcase of freezing effects, have a slightly higher complexity in that they are either IBSP-2 or IBSP-1. This depends on whether one requires the left and right edge of the k-val to be tied to the same feature f. Since freezing effects are widely considered to be more complex than standard island constraints and depend on the interaction of multiple movements, it is unsurprising that their IBSP complexity should be slightly higher. Nonetheless the IBSP approach with a-string provides a unified perspective of several movement restrictions that highlights their computational simplicity and treats them as a natural syntactic counterpart of blocking effects in phonology.

4 The Limits of A-Strings

The previous section has argued that IBSP grammars over a-strings provide an insightful perspective on movement constraints that highlights their simplicity and their formal parallels to blocking effects in phonology.

It is also noteworthy just how limited the machinery is. For instance, it is now unsurprising that no language has island constraints such as "you may move out of as many adjuncts as you have movement features". This simply cannot be expressed with IBSP-1 or IBSP-0. Similarly, we correctly predict that no language has complex structural conditions like "an adjunct is an island iff it is c-commanded by another adjunct". Not only would this require a larger k-val than IBSP-1 and IBSP-0 provide, the use of a-strings makes it completely impossible to refer to c-commanders. By adopting a string representation that only keeps track of containment, c-command conditions become inexpressible. While every Y in $\text{as}(X)$ c-commands X, not every c-commander of X appears in $\text{as}(X)$ — only those that are heads of phrases containing X do so. The absence of some c-commanders in a-strings thus makes them unsuitable to express c-command conditions.

The limits of a-strings with respect to c-command is both a curse and a blessing. As just discussed, it has the advantage of greatly limiting the predicted typology of island constraints. At the same time, it also means that the current approach is entirely incapable of handling some well-known restrictions on movement: the *that*-trace effect, and the coordinate structure constraint.

Let us first consider the *that*-trace effect, the core cases of which are illustrated below:

(15) a. Who do you think [Mary will leave _]?

b. Who do you think [_ will leave Mary]?

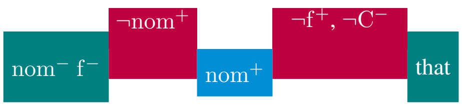
c. Who do you think [that Mary will leave _]?

d. * Who do you think [that _ will leave Mary]?

The *that*-trace filter forbids a subject to move across the head of the smallest containing sentential CP if that head is empty. This adds several new complications, but these can all be handled with IBSP.

The restriction to subjects amounts to the requirement that the left edge of the k-val must be a mover whose first movement feature is nom^-, followed by some f^-. Similarly, the limitation to sentential CPs can be expressed in terms of the MG feature calculus. The complementizer in the examples above has the feature make-up $\mathrm{T}^+\mathrm{C}^-$, whereas the complementizer of a relative clause, for instance, would have $\mathrm{T}^+\mathrm{N}^\sim$ (under an analysis of relative clauses as NP-adjuncts; other analyses require different features, but it will never be $\mathrm{T}^+\mathrm{C}^-$). So this aspect of the *that*-trace effect does not challenge the IBSP perspective either. Finally, the requirement that the constraint only applies to the closest such complementizer can be captured by restricting the appropriate filler. Overall, the typical instances of the *that*-trace constraint can be handled by an IBSP-1 grammar that uses the same truncation trick as our IBSP-1 treatment of freezing effects in (11).

(16) *IBSP-1 grammar for the* that-*trace effect*



For the core cases, then, the *that*-trace effect exceeds the strong island constraints in complexity, but is comparable to freezing effects.

However, there are cases where *that*-trace violations are repaired, and these cannot be handled in our approach. For instance, the *that*-trace effect does not apply when the gap is c-commanded by additional material.

(17) Who do you think [that [under no circumstances] _ will leave Mary]?

Here *under no circumstances* is an adjunct that attaches to TP or some other position below the complementizer and above the subject gap. This adjunct does not contain the gap, it only c-commands it. As a result, it is not present in the relevant a-strings, which makes it impossible for us to suspend the *that*-trace constraint. In order to handle this case, one needs a representation that encodes both c-command and containment, e.g. the c-strings of Graf and Shafiei (2019).

But the addition of c-command actually undermines the whole approach because it becomes impossible to determine a mover's landing site. Recall that our grammars block f^+ from occurring in the fillers so that we can correctly pick out the landing site for f-movement, i.e. the closest containing head with f^+. Crucially, heads that c-command the mover but do not contain it are not viable landing sites. For instance, if we are looking at the c-string of some f-mover that is the complement of some head H, the specifier of H may be some lexical item carrying f^+. This specifier should be allowed to go into a filler slot. At the same time, a c-commander that both carries f^+ and contains the f-mover should not be allowed to go into a filler slot. Since fillers are specified as lists of lexical items, there is no way to distinguish in their specification between containing c-commanders and all other c-commanders. Either we allow both in the filler or neither, and in each case we end up with an unsuitable grammar. IBSP is too weak to make the relevant distinctions with a representation format that encodes both c-command and containment.

While *that*-trace repair points out a limitation of IBSP, the coordinate structure constraint challenges the very notion of string-based representations for movement. This constraint forbids extraction from a conjunct, except if movement takes place across-the-board from all conjuncts.

(18) a. * Which wine did [Ed brew beer and Greg drink _]?

b. Which wine did [Ed brew _ and Greg drink _]?

Since there is no c-command or containment relation between the gaps in (18b), neither one appears in the other's c-string or a-string. Consequently, the c-strings for the object of *drink* do not differ at all between the two sentences, which makes it impossible to give a c-string account of this island constraint irrespective of how powerful one's computational apparatus is.

These two constraints show that IBSP over a-strings does not provide a fully exhaustive theory of islands or movement constraints. But the IBSP approach does highlight the structural uniformity of many islands, their computational simplicity, and their parallels to blocking effects in phonology. While our findings are still preliminary and need to be vetted by detailed analysis of a much wider range of constraints across many languages, it is encouraging that they closely mirror previous findings in phonology and yield rigorous claims about the possible shapes of islands.

5 Linguistic Naturalness

The previous section focused on some shortcomings of our approach with respect to expressivity, but there is also the issue of linguistic naturalness. First, the choice of string representations is unusual. Second, the reliance on lists of lexical items for specifying the components of an IBSP grammar means that there is no notion of naturalness. We acknowledge both issues, but we think that they can be insightfully addressed in future work.

As was briefly mentioned in §2.2, a-strings are just a convenient abstraction and the findings of this paper can be restated in terms of formal machinery that operates over trees instead of strings. This includes the tree tiers of Graf (2018b) and the sensing tree automata of Graf and De Santo (2019). But in both cases the necessary math is more likely to obfuscate the simplicity of the underlying principles, and the use of tree structures hides that the simple notion of containment is already enough to state many conditions on movement. We thus maintain that a-strings are methodologically useful even if they may not be cognitively real.

This leaves the lack of natural classes. It is true that our current approach is still too lenient a characterization of the class of possible island constraints. For instance, one can easily write an IBSP grammar over a-strings that does not allow topicalization across a ditransitive verb. Similarly, the ability to account for some exceptions such as (4) also allows us to specify ludicrous exceptions, for instance that the head of an adjunct induces an island unless it is a palindrome. These are clearly undesirable options, but they are typical of computational work. Our primary goal was to analyze island constraints from a subregular perspective to more accurately pinpoint their overall complexity. This allows us to put an upper bound on what island constraint may look like, but this is still a very generous bound. The formal restrictions must be paired with a theory of linguistic substance to accurately circumscribe the class of possible island constraints (see e.g. Graf 2013 for one such account for the adjunct island constraint).

6 Conclusion

We have argued that the most common cases of strong islands can be expressed as IBSP-1 (or IBSP-0) constraints on string representations that encode only containment. This formal characterization establishes new parallels to phonology and tightens the linguistic typology by excluding logically conceivable yet unattested island constraints. While a lot of empirical modeling work remains to be done, we are confident that this novel perspective on islands will prove very fertile.

Acknowledgments

The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344. We would like to thank the three anonymous reviewers for their detailed comments, which prompted a major rewrite of the paper.

References

Robert Frank and K Vijay-Shanker. 2001. Primitive c-command. *Syntax*, 4(3):164–204.

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. 2013. The syntactic algebra of adjuncts. In *Proceedings of CLS 49*. To appear.

Thomas Graf. 2017. The power of locality domains in phonology. *Phonology*, 34:385–405.

Thomas Graf. 2018a. Locality domains and phonological c-command over strings. In *NELS 48: Proceedings of the Forty-Eighth Annual Meeting of the North East Linguistic Society*, volume 1, pages 257–270, Amherst, MA. GLSA.

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

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 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. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frank Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. Mouton De Gruyter.

Robert McNaughton and Seymour Papert. 1971. *Counter-Free Automata*. MIT Press, Cambridge, MA.

Jean-Eric Pin. 1997. Syntactic semigroups. In *Handbook of Language Theory*, pages 679–764. Springer, Berlin.

John R. Ross. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT.

Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.

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

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

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.

Anssi Yli-Jyrä. 2005. *Contributions to the Theory of Finite-State Based Grammars*. Ph.D. thesis, University of Helsinki.