UCLA Working Papers in Linguistics, Papers in Mathematical Linguistics 1
Volume 15, Article 2: 1–26, 2010

For Game Settings, Press SELECT

Natasha Abner

Robust empirical support has been found for the idea that certain properties of the grammar are naturally non-symmetrical, as evidenced by the fact that certain logically possible word orders remain unattested in cross-linguistic inventories. It is proposed here that these linear non-symmetries arise as a result of the selection relation that drives syntactic structure building. A theory-independent definition of selection is offered and is shown to derive linear order non-symmetries with minimal assumptions about the properties of the grammar. The generative mechanisms of a number of diverse grammatical frameworks are evaluated and are shown to each instantiate at least one operation that satisfies the selection definition provided here.

Keywords syntax, selection, linearization, typology, (Combinatory) Categorial Grammar, Principles & Parameters, Tree Adjoining Grammar

Introduction

Empirical evidence, both within and across languages, repeatedly homes in on the observation that certain properties of the grammar are non-symmetrical. These non-symmetries of human language present themselves across a number of typological domains and have been explored within a variety of theoretical frameworks. For example, licit and illicit orderings of verbal clusters in West Germanic have received analysis in both the transformational framework (Haegeman and van Riemsdijk 1986) as well as the generative Tree Adjoining Grammars (Kroch and Santorini 1991). Likewise, both Combinatory Categorial Grammar (Steedman 1996) and Head Driven Phrase Structure Grammar (Pollard and Sag 1992) provide a means of accounting for the possible and impossible binding relations between the nominal elements in an expression of a language. Given that each of these frameworks endeavors to provide a description of the grammars of human language, their mutual success across a number of domains is unsurprising. Nevertheless, the existence of diverse and successful grammatical frameworks raises the question of whether or not there are any properties that hold across these distinct frameworks and, moreover, whether such properties may be responsible for observed non-symmetries in language. I propose that the obligatory presence of selection functions in grammars of human language is a candidate for just such a property.

Following a brief introduction to the Bare Grammar framework of Keenan and

© 2010 Natasha Renee Abner
This is an open-access article distributed under the terms of a Creative Commons Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/).

Stabler (2003) — a framework compatible with all of those mentioned above — I provide a theory-independent definition of what I refer to as a selection function: a function that establishes an immutable local dependency between elements of the language. Using linear order patterns within and across languages as a testing ground, I show how the presence of these selection functions can account for documented non-symmetries with only minimal assumptions about the grammar. Having established the potential for selection functions in grammar to account for non-symmetries of language, a variety of grammatical frameworks are explored and are each shown to contain selection functions as defined here.

1 Bare Grammar Framework

The claim made here is that non-symmetric properties of human languages arise as the consequence of the obligatory presence of selection relationships between elements of the language. The verifiability of such a claim, however, obviously hinges on the existence of a definition of selection that is applicable across a variety of generative frameworks. In order to provide a formal definition of selection without being overly dependent on framework-specific properties of the grammar, I make use of the Bare Grammar (BG) framework developed in Keenan and Stabler (2003). The minimal restrictions imposed by the BG model on the grammar allow this framework to be compatible with diverse theoretical frameworks. Thus, a BG definition of selection, as will be discussed in Section 4, can be employed across frameworks relatively unhindered by specific properties of a formalism.

In the BG framework, the licit expressions of a given language L are derived by the four-tuple grammars formalized below, though we allow that the grammar-identifying subscript be omitted when it is clear from context.

Definition 1 (Bare Grammar)**.** A bare grammar G is defined as $\langle V_G, Cat_G, Rule_G, Lex_G \rangle$, where:

V_G = vocabulary items (strings)
Cat_G = categories
Lex_G = lexical items, a subset of $V_G \times Cat_G$
$Rule_G$ = partial functions from $(V_G^* \times Cat_G)^j$ into $V_G^* \times Cat_G$ for any j.

Thus, the BG formulation of the generative grammar for a given language will require the specification of: the strings of the language (V_G), the categories of the language (Cat_G), the possible pairings of strings and categories of the language (Lex_G), and the generative mechanisms by which fixed-length sequences of lexical items of the language can be successively combined into larger structures through the structure building functions of G ($Rule_G$). The language generated by such a grammar is formally defined as the set of expressions that are either in the lexicon of the language (Lex_G) or are outputs of the rules of the language applied to those elements:

Definition 2 (Language)**.** For any grammar G let $L_G = \bigcup_n Lex_n$, where $Lex_0 = Lex_G$ and for all $n \geq 0$, $Lex_{n+1} = Lex_n \cup \left\{ F(t) \mid F \in Rule_G,\ t \in Lex_n^* \cap \text{domain}(F) \right\}$.

At several points in the discussion, the string component str and category cat of an expression play a role; these are straightforwardly defined as below.

Definition 3 (String Component & Category)**.** For any $x \in \langle V^*, Cat\rangle \in L_G$, $\mathrm{str}(x) = V^*$ and $\mathrm{cat}(x) = Cat$.

A benefit of using the BG framework is that it provides a means of identifying structural similarities across expressions of the language without making a commitment to the structural properties of the grammar itself — that is, BG allows one to identify expressions as having the same structure without identifying what that structure is. The framework is designed such that the precise structure of any particular expressions of a language is an intrinsic result of the generative mechanisms of the language — expressions that are derived by the same rules are analyzed as sharing the structural characteristics engendered by those rules. If one takes an expression of the language and substitute its elements piecewise, the structure of that expression will remain identical so long as the generative rules that derived that expression are maintained. Properties that an expression has are said to be structural properties if all structurally identical expressions also have those properties and, analogously, relations between expressions are said to be structural relations if all other sets of structurally identical expressions also bear those relations to each other. Such piecewise substitutions are formalized as rule-preserving automorphisms, as defined below.

Definition 4 (Automorphism)**.** A function h from L_G to L_G is an automorphism of G *iff* h is a bijection and h is rule-preserving in the sense that $h(F) = F$ for all $F \in Rule_G$. For any grammar G, Aut_G represents the set of such automorphisms.

The notion that such automorphisms are rule-preserving — that is, that $h(F) = F$ for all $F \in Rule_G$ — simply means that if F is considered as a set of pairs $\{\langle K, J\rangle \mid F(K) = J\}$, where K is a sequence of expressions in L_G in the domain of F, then this set is identical to the set $\{\langle h(K), h(J)\rangle \mid F(h(K)) = h(J)\}$, letting $h(K)$ denote the pointwise application of the automorphism h to the elements in the sequence K. Conventionally, then, this notion is simply the requirement that the function h commute with the structure building functions of L_G.

The structural properties that remain constant under such automorphisms are termed the invariants of the language.

Definition 5 (Invariant)**.** The invariants of a grammar G are the fixed points of the automorphisms of G.

The intuitive idea captured here is that the invariants of a grammar G are those things that must be held constant under substitution — that is, the things that cannot be changed without affecting the structure of an expression. Thus, the property *is a subject* will be an invariant property of a language if for any expression u that has that property, then in any automorphism h of the language, $h(u)$ — an expression whose derivation mirrors that of u in the sense that all of the same rules are applied and are applied in the same way — also has the property *is a subject*. It is a trivial but nevertheless welcome truth, then, that the structure building functions themselves are invariant properties of grammars. Non-trivial invariants in certain grammars are the 'functional'/'grammatical' elements such as voice and case markers. Crucially, in identifying these invariants, it is only necessary to specify that the rules be preserved, no additional restrictions on the details of the rules themselves need be imposed.

Within the BG framework overviewed here, Keenan and Stabler explore additional restrictions that may be placed on the grammars of human languages. The aspects of BG just discussed, however, are sufficient to provide a definition of selection that is provably invariant, has consequences for linear order relation, and can be applied across grammatical frameworks.

2 Selection

Providing a definition of selection that allows it to define invariant relationships across a variety of grammatical frameworks is not a straightforward task, not least of all because selection, in name, at least, is not explicitly incorporated into all of the frameworks mentioned here. Even within the Principles and Parameters framework, wherein categorial and semantic selection are frequently mentioned, the selection relationship goes without formal characterization. Nevertheless, the generative mechanisms across all of these frameworks do generate syntactic and semantic relationships between elements of the grammar using what I call selection functions, a formal definition of which is provided at the end of this section.

In each of the frameworks evaluated here, these selection functions put expressions of the language together in a fixed way, where this fixedness can be understood as a resulting from the properties of the expressions themselves. These expressions go together in a fixed way because they bear a certain local relation to each other: the selection (selector-selectee) relation. This selection relation is a byproduct of the category membership of the expressions — it is not a single lexical item that acts as a selector or selectee in relation to another lexical item, but rather a category of lexical items that act as a selector or selectee in relation to another category of lexical items. Moreover, this selection relationship is responsible for encoding the semantic dependency that arises in a local domain between expressions of a language. Each of the conditions outlined below captures the intuition that the selection relationship is established in a local, fixed way between categories of expressions in a language.

2.1 Selection: The Conditions

In this section, I further explore the notion of selection that was just discussed and is summarized informally below.

Definition 6 (Selection Function, Informally)**.** A selection function in a grammar G is a function that takes elements of the language and puts them together in a fixed way, as determined by the category membership of those elements.

I posit three specific conditions that must hold if a rule of a given grammar is to be considered a selection function. This discussion and definition of selection functions will allow me to defend the hypothesis below.

Hypothesis 1. (Selection). *Grammars of human languages are selection grammars.*

Definition 7 (Selection Grammar)**.** A grammar G is a selection grammar if and only if it contains at least one selection function.

In the course of discussing the conditions on selection functions, I will outline generative systems that Hypothesis 1 rules out as possible grammars of human languages. A formal definition of selection, including a formalization of each of these conditions, is then provided.

2.1.1 *Condition (i): Sequence Length*

The most obvious condition to be imposed on the selection functions in a grammar stems from the fact that these selection functions "put elements of the language together" — they must operate over more than a single item of the language. The selection functions of a grammar are, thus, only those with an arity of at least two. Grammars which generate expressions using only unary operations, such as the admittedly undergenerating approach to Dutch below, are ruled out by Condition (i).

Example (Unary Grammar). Let $G_{Un} = \langle V_{Un}, Cat_{Un}, Rule_{Un}, Lex_{Un} \rangle$ with
$V_{Un} = \{dee, hond, eet\}$
$Cat_{Un} = \{\mathrm{N}, \mathrm{D}, \mathrm{Det}, \mathrm{V}, \mathrm{S}\}$
$Lex_{Un} = \{\langle dee, \mathrm{Det}\rangle, \langle hond, \mathrm{N}\rangle, \langle eet, \mathrm{V}\rangle\}$
$Rule_{Un} = \{U_{\mathrm{N}}, U_{\mathrm{D}}\}$
where $U_1, U_2 \in Rule_{G_{Un}}$ are defined as:

$$\begin{array}{lcl} U_1(\langle hond, \mathrm{N}\rangle) & \longmapsto & \langle dee\ hond, \mathrm{D}\rangle \\ U_2(\langle dee\ hond, \mathrm{D}\rangle & \longmapsto & \langle dee\ hond\ eet, \mathrm{S}\rangle \end{array}$$

Condition (i) will also have the effect of ensuring that selection functions of a given grammar do not vacuously satisfy Conditions (ii–iii), discussed in the following sections, as Condition (i) requires these functions to have an arity greater than one.

2.1.2 *Condition (ii): Category Closure*

The condition of *category closure* formalizes the idea that the selection relations that hold between elements in the grammar hold due to the categorial status of those elements — that selection is a property of categories, not individual expressions of the language. Thus, Condition (ii) requires that for all selection functions in the grammar, if an element α is in a sequence in the domain of a selection function, then all items that are of the same category as α can stand in the place of α in that sequence. The domain of a selection function is, then, closed under replacement by elements in the same category, rendering the applicability of the selection function determinable solely by the category of the elements in a given sequence.

While category closure can be imposed independent of the categories used within a grammar, it forces grammars of human languages to be those that make natural and reasonable generalizations over categories. Grammars that fail to make generalizations over the categories, like the model grammar for feminine DP formation in French below, are thus ruled out as possible grammars for human languages.

Example (Non-Category Closed Grammar). Let $G_{CO} = \langle V_{CO}, Cat_{CO}, Rule_{CO}, Lex_{CO} \rangle$, with $a_1 \ldots a_n \in Rule_{CO}$ defined as:

$$
\begin{array}{lcl}
a_1(\langle la, \mathrm{D_{fem}} \rangle, \langle abbaye, \mathrm{NP_{fem}} \rangle) & \longmapsto & \langle l'abbaye, \mathrm{DP_{fem}} \rangle \\
a_2(\langle la, \mathrm{D_{fem}} \rangle, \langle abeille, \mathrm{NP_{fem}} \rangle) & \longmapsto & \langle l'abeille, \mathrm{DP_{fem}} \rangle \\
a_3(\langle la, \mathrm{D_{fem}} \rangle, \langle abondance, \mathrm{NP_{fem}} \rangle) & \longmapsto & \langle l'abondance, \mathrm{DP_{fem}} \rangle \\
 & \vdots & \\
a_n(\langle la, \mathrm{D_{fem}} \rangle, \langle zoologie, \mathrm{NP_{fem}} \rangle) & \longmapsto & \langle la\ zoologie, \mathrm{DP_{fem}} \rangle
\end{array}
$$

Rules like those in $a_1, \ldots, a_n$ are not selection functions because they fail to allow the interchangeability of the elements that are of category $\mathrm{NP_{fem}}$ and, thus, are not category closed. Grammars with only rules like these, while perhaps adequate in terms of generative capacity, are not possible grammars of human languages. Grammars of human languages must include rules — selection functions — that make use of the category system in determining which sequences of elements are in the domain of those rules. Such a restriction is empirically motivated given the task that learners of a language face and the competence that speakers of a language exhibit in certain linguistic domains. When the speaker of French encounters a new feminine noun, the speaker knows immediately how to combine that feminine noun with the definite determiner, because the speaker's knowledge of language involves generalizations across categories. Likewise, the learnability of human language grammars seems contingent upon the learner being licensed to draw broad generalizations from limited input.

In addition to supporting generalizations such as these — which, in fact, amounts to accounting for the natural generalizations that a speaker's knowledge of a language includes — Condition (ii) will also enforce a certain level of fine-grained detail in the category structure of a language. Specifically, if selection functions by definition are closed under replacement by elements in the same category, then two expressions of the language must be of a different category if they behave differently with respect to the selection functions in a language. That is, if two expressions of a language are superficially quite similar but nevertheless fail to be in the domains of the same sets of selection functions, then they must be categorically distinct. This is illustrated by looking at the complete gender system of a language like French where, while nouns may be similar with respect to their denotation and their ability to host number marking, they must be divided into at least two classes, feminine and masculine, if a function that combines them with a determiner is to be considered a selection function. Thus, Condition (ii) captures the fact that the assignment of categories to base and derived expressions in human languages is not fully arbitrary but, rather, is used to determine how those expressions behave with respect to certain generative mechanisms of the language: the selection functions.[1]

[1] In terms of the phonological aspects of language, Condition (ii) has the effect of preventing selection functions which are sensitive to the string component of the elements in their domain (e.g. notions such as 'heaviness').

2.1.3 Condition (iii): Constancy Under Permutation & Sub-composition

Condition (iii) is responsible for capturing the fact that selection functions, as described informally above, put together elements of the grammar in a fixed way. The thrust of this idea is that, just as the sequences in the domain of a selection function will be determined by the elements in the sequence — here, their categories, the classes of which may encode both syntactic and semantic information — so too will the output of the selection function. That is, the category of an element determines what it selects or what it is selected by and, moreover, its category also determines the end result once that selection relationship has been established by the rules of the grammar.[2] Thus, if a sequence of elements is in the domain of a selection function, then any rule of the grammar that combines those elements in any order must combine them such that the output of the rule is identical to that of the selection function.

Given, then, any sequence of elements in the domain of a selection function, there are two logically possible means of designing an alternative rule for combining those elements. The first of these is to design a rule that takes as its domain a permutation of the sequence of elements in the domain of the original selection function. Condition (iii) will allow that such alternative rules exist in the grammar of the language but, as just noted, will require that the output of those rules match the original selection function. Thus, Condition (iii) will allow grammars such as that below, which includes two functions that combine one place predicates with their arguments, as the output of the selection rule is matched by that of the alternative rule.

Example (Permissible Permutation Grammar). Let $G_{PP} = \langle V_{PP}, Cat_{PP}, Rule_{PP}, Lex_{PP} \rangle$ with $R \in Rule_{PP}$ defined as:

$$\begin{array}{lcl} f(\langle John, \mathrm{DP}\rangle, \langle fell, \mathrm{P1}\rangle) & \longmapsto & \langle John\ fell, \mathrm{P0}\rangle \\ g(\langle fell, \mathrm{P1}\rangle, \langle John, \mathrm{DP}\rangle) & \longmapsto & \langle John\ fell, \mathrm{P0}\rangle \\ h(\langle John, \mathrm{DP}\rangle, \langle fell, \mathrm{P1}\rangle) & \longmapsto & \langle John\ fell, \mathrm{P0}\rangle \end{array}$$

However, grammar such as the following, which, like that above, includes only three functions, are not possible grammars of human language, as Condition (iii) is not met.

Example (Impermissible Permutation Grammar). Let $G_{IP} = \langle V_{IP}, Cat_{IP}, Rule_{IP}, Lex_{IP} \rangle$ with $R \in Rule_{IP}$ defined as:

$$\begin{array}{lcl} f(\langle John, \mathrm{DP}\rangle, \langle fell, \mathrm{P1}\rangle) & \longmapsto & \langle John\ fell, \mathrm{P0}\rangle \\ g(\langle John, \mathrm{DP}\rangle, \langle fell, \mathrm{P1}\rangle) & \longmapsto & \langle fell\ John, \mathrm{P0}\rangle \\ h(\langle fell, \mathrm{P1}\rangle, \langle John, \mathrm{DP}\rangle) & \longmapsto & \langle John\ fell, \mathrm{P3}\rangle \end{array}$$

[2] The constancy under permutation enforced by Condition (iii) is similar to the notion of Category Functionality, defined in (i).

(i) Category Functional. For any grammar G, a function $f^n \in Rule_G$ is category functional *iff* there is a function g from $(Cat_G)^n$ into Cat_G such that for all n-tuples σ in domain(f), $\mathrm{cat}(f(\sigma)) = g(\mathrm{cat}(\sigma_i), \ldots, \mathrm{cat}(\sigma_n))$.

Condition (iii), however, places restrictions that are stronger than those of category functionality, as it enforces identity of both string and category components and, moreover, requires that any such category selection function — i.e. g in (i) — be indifferent to the order of elements in the tuples in its domain.

The second logically possible design for an alternative rule that combines the elements in a selection function is to combine subsets of those elements. In this case, if selection functions combine elements of the grammar in a fixed way, then rules that combine subsets of those elements in a way different than the selection function should be disallowed, as exhibited in the grammar below.

Example (Impermissible Sub-composition Grammar). Let $G_{IS} = \langle V_{IS}, Cat_{IS}, Rule_{IS}, Lex_{IS} \rangle$ with $R \in Rule_{IS}$ defined as:

$$\begin{array}{lcl} f(\langle John, \mathrm{DP_{nom}} \rangle, \langle cake, \mathrm{DP_{acc}} \rangle, \langle ate, \mathrm{P2} \rangle) & \longmapsto & \langle John\ ate\ cake, \mathrm{P0} \rangle \\ g(\langle cake, \mathrm{DP_{acc}} \rangle, \langle ate, \mathrm{P2} \rangle) & \longmapsto & \langle cake\ ate, \mathrm{P1} \rangle \\ h(\langle John, \mathrm{DP_{nom}} \rangle, \langle cake\ ate, \mathrm{P1} \rangle) & \longmapsto & \langle John\ cake\ ate, \mathrm{P0} \rangle \end{array}$$

Given that empirical evidence such as constituency and prosody may necessitate the presence of such 'sub-composition' rules in the grammars of human languages, an appropriate definition of a selection function should allow that they exist. Condition (iii) will allow for such sub-composition functions only if the grammar also includes a function that can compose with the sub-composition function and produce an output identical to that of the selection function. That is, so long as the sub-composition function combines the subset of elements in the same way that they were combined in the full selection function and, thus, can be composed so as to match the output of the selection function. This allows for grammars of the type below.

Example (Permissible Sub-composition Grammar). Let $G_{PS} = \langle V_{PS}, Cat_{PS}, Rule_{PS}, Lex_{PS} \rangle$ with $R \in Rule_{PS}$ defined as:

$$\begin{array}{lcl} f(\langle John, \mathrm{DP_{nom}} \rangle, \langle cake, \mathrm{DP_{acc}} \rangle, \langle ate, \mathrm{P2} \rangle) & \longmapsto & \langle John\ ate\ cake, \mathrm{P0} \rangle \\ g(\langle cake, \mathrm{DP_{acc}} \rangle, \langle ate, \mathrm{P2} \rangle) & \longmapsto & \langle ate\ cake, \mathrm{P1} \rangle \\ h(\langle John, \mathrm{DP_{nom}} \rangle, \langle ate\ cake, \mathrm{P1} \rangle) & \longmapsto & \langle John\ ate\ cake, \mathrm{P0} \rangle \end{array}$$

Therefore, though Condition (iii) allows for both permutation and sub-composition of the elements in the domain of a selection function, two characteristic properties of selection functions are nevertheless maintained. First, it is the elements themselves that define the functional output of their combination, as the end result of the permutation or sub-composition functions will always match that of the original selection function. Second, the relationship established between the elements in the domain of a selection function is an obligatorily local relationship. While permutation and sub-composition are permissible, neither of these deviations from the original selection function will disrupt the locality of this relationship.

2.2 *Selection: Defined*

Conditions (i) through (iii) were posited in order to capture the aspects of the informal definition of selection given above. Having carefully explored the implications of these conditions on the functions that satisfy them and the grammars that contain such functions — hypothesized here to be all grammars of human languages — a formal definition of a selection function can now be provided. In order to accurately capture the second aspect of Condition (iii) — constancy under sub-composition, a condition that will only be invoked if the grammar has functions that exceed an arity of two — the definitions of i-Composition and $Rule_\circ$ below are used.

Definition 8 (i-Composition)**.** For any functions f of arity j and g of arity k and for some i such that $1 \leq i \leq j$:

$$\text{domain}(f \circ_i g) = \Big\{ \langle s_1 \ldots s_{i-1} t_1 \ldots t_k s_{i+1} \ldots s_j \rangle \mid (t_1 \ldots t_k) \in \text{domain}(g) \text{ and } \langle s_1 \ldots s_{i-1} g(t_1 \ldots t_k) s_{i+1} \ldots s_j \rangle \in \text{domain}(f) \Big\}$$

$$f \circ_i g(s_1 \ldots s_{i-1} t_1 \ldots t_k s_{i+1} \ldots s_j) \stackrel{\text{def}}{=} f(s_1 \ldots s_{i-1} g(t_1 \ldots t_k) s_{i+1} \ldots s_j)$$

Definition 9 ($Rule_\circ$)**.** For any $G = \langle V, Cat, Lex, Rule \rangle$, $Rule_\circ$ is the closure of $Rule$ under i-Composition:

$Rule_\circ \stackrel{\text{def}}{=} \text{closure}(Rule, \{\circ_i \mid i \in \mathbb{N}\})$.

Finally, along with the three conditions discussed above, each of which are intended to characterize the nature of a selection relationship, an additional restriction characterizing the usefulness of the selection relationship is imposed. This restriction requires that the domain of a selection function in a grammar not be empty — that is, not only must grammars of human languages contain selection functions as defined by Conditions (i)–(iii), they must make use of these selection functions in the generation of expressions of the language.

Definition 10 (Selection Function)**.** For any function f in a grammar G, f is a selection function in G *iff* $\text{domain}(f) \neq \emptyset$ and for all n-tuples $\sigma \in \text{domain}(f)$, the following conditions hold:

Condition (i): Sequence Length. $\text{length}(\sigma) > 1$.

Condition (ii): Category Closure. For any $x \in \sigma$, for any σ' such that σ' is the result of replacing x with $y \neq x$, $\sigma' \in \text{domain}(f)$ if $\text{cat}(x) = \text{cat}(y)$.

Condition (iii): Constancy Under Permutation & Sub-composition. For any $\sigma' \in \text{domain}(g)$ for $g \in Rule_G$, if $|\{x \mid x \in \sigma\} \cap \{y \mid y \in \sigma'\}| > 1$, then either:

a. σ' is a permutation of σ and $g(\sigma') = f(\sigma)$ or
b. $\exists k \in Rule_\circ$ of G such that $\sigma = s_1 \ldots s_{i-1} \sigma' s_{i+1} \ldots s_j \in \text{domain}(k \circ_i g)$ and $k \circ_i g(s_1 \ldots s_{i-1} \sigma' s_{i+1} \ldots s_j) = f(\sigma)$.

The additional restriction placed on the cardinality of the set intersection in Condition (iii) captures the fact that human languages do seem to allow a certain amount of malleability in the selection relationships. The perfect auxiliary, *have*, for example, may combine directly with a verbal element, *have gone*, or with the progressive, *have been going*. Likewise, nominal elements are found in selection relationships with both verbs and prepositions. Condition (iii) allows for this malleability by evaluating the fixed output of a given selection function only if there exists another rule in the grammar that puts together more than one of the elements in the sequence of the selection function. If such a rule exists, then this rule must, as discussed above, combine the elements in a way that matches the output of the selection function. In the case of non-complete overlap between the alternative rule and the original selection function, the second clause of Condition (iii) requires that the grammar contain rules that can be *i*-composed with the alternative rule so as to match the output of the selection function. In the case of complete overlap between the alternative rule and the original selection function, the first clause of Condition (i) requires that the output of the alternative rule match that of the selection function. In the latter case, the language generated by the grammar remains unaffected if this rule is removed.

Theorem 11. *For all selection functions f in any grammar $G = \langle V, Cat, Lex, Rule\rangle$ and for any $g \in Rule_G$ such that σ', a permutation of σ in* $\mathrm{domain}(f)$*, is in* $\mathrm{domain}(g)$*, let $G' = \left\langle V, Cat, Lex, Rule'\right\rangle$, where $Rule' = (Rule - \{g\}) \cup \{g - \langle \sigma', g(\sigma')\rangle\}$. Then $L_G = L_{G'}$.*

Proof. From the definition of G, G', it follows that $(Lex_G)_0 = (Lex_{G'})_0$. Supposing that $(Lex_G)_n = (Lex_{G'})_n$, it can be shown that $(Lex_G)_{n+1} = (Lex_{G'})_{n+1}$.

> $(Lex_{G'})_{n+1} \subseteq (Lex_G)_{n+1}$: Also trivial.
>
> $(Lex_G)_{n+1} \subseteq (Lex_{G'})_{n+1}$: Let $x \in (Lex_G)_{n+1}$. Then either (i) $x \in (Lex_G)_n$ or (ii) $\exists h \in Rule_G$, $\exists \alpha \in (Lex_G)^*_n$ such that $h(\alpha) = x$.
>
> If (i) then $x \in (Lex_{G'})_{n+1}$.
> If (ii) where $h = g$ and $\alpha = \sigma'$,
> then $\sigma' \in (Lex_{G'})^*_n$ and $g(\sigma') = f(\sigma) \in (Lex_{G'})_{n+1}$.
> If (ii) where $h \neq g$ then $h(\alpha) \in (Lex_{G'})_{n+1}$.
> If (ii) where $h = g$ and $\alpha \neq \sigma'$ then $g(\alpha) \in (Lex_{G'})_{n+1}$ □

This definition of a selection function can be straightforwardly used to provide a definition of a selection relationship between elements of the language.

Definition 12 (Selection Relationship)**.** For all $u, v \in L_G$, there is a selection relationship between u and v *iff* $u, v \in \sigma$ for some $\sigma \in \mathrm{domain}(f)$ for f a selection function in G.

Given that the selection relationship is defined in terms of the generative mechanisms of the grammar, it is unsurprising that this relationship is provably invariant.

Theorem 13. *The selection relationship in a grammar G is invariant.*

Proof. Let $u, v \in L_G$ such that there is a selection relationship between u, v. Then $u, v \in \sigma$ for some $\sigma \in \mathrm{domain}(f)$ for f a selection function in G and for any $h \in \mathrm{Aut}_G$, because h preserves $Rule_G$, it must be the case that $h(u), h(v) \in \sigma$ for some $\sigma \in \mathrm{domain}(f)$ for f a selection function in G, thus there is a selection relationship between $h(u)$ and $h(v)$. □

Note, finally, that grammars satisfying Hypothesis 1 need only contain a single selection function. While there is a natural intuition that if the presence of selection functions is one of the defining characteristics of human language grammars, then such selection functions will do much of the generative work of the grammar, there is no requirement that they do all of the generative work of the grammar. Though this stronger stance may turn out to be empirically motivated, it is not the one pursued and evaluated here. In the following section, I illustrate how even this weaker stance — that grammars contain a single selection function — can be used in a BG framework to explain the linear non-symmetries that are found within and across languages.

3 Linear Order as a Consequence of Selection

Typological investigations frequently center on the licit linear orderings of words and morphemes in language. Such investigations repeatedly converge on several facts about linear order in human languages: (i) linear order of elements within a given language is never truly unrestricted (Legate 2002), (ii) certain linear order patterns occur frequently in language while others remain unattested (Cinque 2005) and (iii) the linear order of elements in one domain of a language frequently correlate with the order of elements in other domains (Greenberg 1966). Given this convergence, it becomes obvious that the linear order of elements, both within and across languages, is one of the non-symmetric properties alluded to earlier. An explanation of this empirical convergence, however, will be dependent upon how linear order relations are defined and established in a given grammatical framework.

In what follows, I propose a means of explaining the non-symmetry of linear order within the BG framework outlined above. Given that generative operations within the BG framework are defined such that they always yield a string output, the linear order of elements within a BG-defined language can be established as a byproduct of the rules of the grammar. If an additional restriction is imposed requiring that the string components in the domain of a function be fully and uniquely reflected in the string output of the function — a restriction termed here *string fixing* — the linear order of elements is provably invariant. The invariant linear order between expressions of the language is, moreover, a non-symmetric relation if the obligatory selection functions are subject to an additional restriction over the legitimate sequences of categories and strings in their domain. Thus, if certain categories are assumed to be instantiated across languages then so too are the selection relationships that those categories enter into to. Given that these selection relationships result in the non-symmetry of linear order within a language, adopting this assumption provides a means of accounting for the restricted word order patterns found across languages.

3.1 Defining Linear Precedence

The restricted patterns of linear order possibilities within and across languages suggest that the linear order relations in language are structurally derived. That is, the generative operations found in grammars of human languages should themselves derive the linear order properties of expressions of the grammar and, furthermore, that restrictions on these generative operations should also yield restrictions on linear order possibilities. Given that string outputs are an obligatory component of the rules in a BG-defined grammar, the BG framework provides a straightforward means of connecting the generative operations of the grammar to the linear order of elements within the expressions that grammar generates. The relation that will be used here to explicitly make this connection is that of local precedence, a linear relationship that is established as the consequence of a single generative step, defined below.

Definition 14 (Local Precedence (PRE))**.** For all $u, v \in L_G$, $u \,\mathrm{PRE}\, v$ *iff* $\exists f \in Rule_G$ and strings $t_1, t_2, t_3, t_4, t_5 \in V^*$ such that for some $\sigma \in \mathrm{domain}(f)$, $u, v \in \sigma$, $\mathrm{str}(f(\sigma)) = t_1 t_2 t_3 t_4 t_5$ and $\mathrm{str}(u) = t_2$, $\mathrm{str}(v) = t_4$.

In order that a local precedence relation be established between two expression of a language, the above definition requires that (a) there be a rule of the grammar that directly combines those expressions and (b) that the string components of each expression be represented in the string output of said rule.[3] Should both of these requirements be satisfied, the local precedence relation will then be determined by whichever string component occurs first in the string output of the function that combines the two expressions. Taking the Permissible Sub-composition Grammar from Section 2.1.3 as an example, this definition will yield the linear precedence relations given below.

Example (Permissible Sub-composition Grammar, Redux). Let $G_{PS} = \langle V_{PS}, Cat_{PS}, Rule_{PS}, Lex_{PS} \rangle$ with $R \in Rule_{PS}$ defined as:

$$\begin{array}{lcl} f(\langle John, \mathrm{DP_{nom}} \rangle, \langle cake, \mathrm{DP_{acc}} \rangle, \langle ate, \mathrm{P2} \rangle) & \longmapsto & \langle John\ ate\ cake, \mathrm{P0} \rangle \\ g(\langle cake, \mathrm{DP_{acc}} \rangle, \langle ate, \mathrm{P2} \rangle) & \longmapsto & \langle ate\ cake, \mathrm{P1} \rangle \\ h(\langle John, \mathrm{DP_{nom}} \rangle, \langle ate\ cake, \mathrm{P1} \rangle) & \longmapsto & \langle John\ ate\ cake, \mathrm{P0} \rangle \end{array}$$

PRE *Relations:*
$\langle John, \mathrm{DP_{nom}} \rangle$ PRE $\langle ate, \mathrm{P2} \rangle$, $\langle John, \mathrm{DP_{nom}} \rangle$ PRE $\langle cake, \mathrm{DP_{acc}} \rangle$ (by f)
$\langle ate, \mathrm{P2} \rangle$ PRE $\langle cake, \mathrm{DP_{acc}} \rangle$ (by f, g)
$\langle John, \mathrm{DP_{nom}} \rangle$ PRE $\langle ate\ cake, \mathrm{P1} \rangle$ (by h)

With regard to this example, note that if the function f were removed from the rules of G_{PS}, then the definition of local precedence would not establish a precedence relation between $\langle John, \mathrm{DP_{nom}} \rangle$ and either $\langle ate, \mathrm{P2} \rangle$ or $\langle cake, \mathrm{DP_{acc}} \rangle$. The linear order that does arise between $\langle John, \mathrm{DP_{nom}} \rangle$ and these two expressions, then, only does so due to the local precedence relation between $\langle John, \mathrm{DP_{nom}} \rangle$ and $\langle ate\ cake, \mathrm{P1} \rangle$.

3.2 *Defining the String Operations of a Grammar*

Though the precedence relation can easily be defined as a consequence of the generative operations of the grammar, this definition will fail to provide insight into the structural properties of the precedence relation lest the string operations of the

[3] Here and throughout I abstract away from morphophonological processes that may operate so as to alter the direct correspondence between string inputs and outputs.

grammar also be restricted. In the model grammar below, for example, though precedence is defined as a rule-based notion, it nevertheless fails to be invariant under automorphism.

Example (Non-Invariant Precedence Grammar). Let $G_{NIP} = \langle V_{NIP}, Cat_{NIP}, Lex_{NIP}, Rule_{NIP} \rangle$ with $R \in Rule_{NIP}$ defined as:

$$\begin{array}{lcl} f(\langle a,A\rangle, \langle b,B\rangle) & \longmapsto & \langle ab, C\rangle \\ f(\langle d,D\rangle, \langle e,E\rangle) & \longmapsto & \langle f, F\rangle \end{array}$$

PRE *Relations*: $\langle a,A\rangle$ PRE $\langle b,B\rangle$

Let h be a bijection on G_{NIP} such that

$$h(\langle a,A\rangle) = \langle d,D\rangle \quad h(\langle b,B\rangle) = \langle e,E\rangle \quad h(\langle ab,C\rangle) = \langle f,F\rangle$$
$$h(\langle d,D\rangle) = \langle a,A\rangle \quad h(\langle e,E\rangle) = \langle b,B\rangle \quad h(\langle f,F\rangle) = \langle ab,C\rangle$$
$$h(f(\langle a,A\rangle, \langle b,B\rangle)) = h(\langle ab,C\rangle) = \langle f,F\rangle$$
$$f(h(\langle a,A\rangle), h(\langle b,B\rangle)) = f(\langle d,D\rangle, \langle e,E\rangle) = \langle f,F\rangle$$

Therefore, h commutes with f and is an automorphism on G_{NIP}. Since $\langle d,D\rangle$ PRE $\langle e,E\rangle$ does not hold, $h(\langle a,A\rangle)\,\mathrm{PRE}\,h(\langle b,B\rangle)$ does not hold, either. Therefore, PRE is not invariant in G_{NIP}.

The precedence relation in the above grammar fails to be invariant because the function f does not map string inputs to string outputs in a fixed, predictable manner. Thus, though precedence relations may be established between elements in one sequence in the domain of f, these elements are interchangeable under automorphism with elements that do not stand in the precedence relation.

To establish precedence as an invariant relation within a grammar, it will be necessary to restrict the string operations of the grammar such that there is a fixed mapping between string inputs and string outputs. With such a restriction in place, the string of an element and that of the element that an automorphism interchanges it with are guaranteed to be treated the same way by the generating functions of the grammar and, thus, to enter into the same set of precedence relations. One means of restricting the string operations of a function is to require that they bear a fixed and transparent relation to their string inputs, a property that I term *string fixing* and define below.

Definition 15 (String Fixed). A function f^n is string fixed *iff* there exists a unique permutation π of $\{1,\ldots,n\}$ and a unique set of strings $\alpha_0,\ldots,\alpha_{n+1} \in V^*$ such that:

$$\forall \langle \langle s_1, C_1\rangle, \ldots, \langle s_n, C_n\rangle \rangle \in \mathrm{domain}(f),$$
$$\mathrm{str}(f^n(\langle \langle s_1, C_1\rangle, \ldots, \langle s_n, C_n\rangle \rangle)) = \alpha_0 s_{\pi(1)} \alpha_1 \ldots \alpha_n s_{\pi(n)} \alpha_{n+1}$$

String fixed functions can produce as their output any permutation of the string components of the expressions in their input, provided that the permutation is fixed across the entire domain of the function — that is, the function must permute each sequence of strings in the same manner for all sequences in its domain. Moreover, such functions may insert fixed string constants into their string output, provided, again, that such insertions be fixed across the domain of the functions. Finally, to enforce that the string outputs of the functions be transparently related to their string inputs, the definition requires that each string component of their input be fully and uniquely reflected in their string outputs. Given this, functions cannot copy, delete or interleave string components of the input in their string output. The local precedence relations in a grammar containing only functions with string operations restricted in this manner is invariant.

Theorem 16. *For any grammar G such that all $f \in Rule_G$ are string fixed, local precedence is invariant.*

Proof. Let $u, v \in L_G$ such that $u\,\mathrm{PRE}\,v$. Then $\exists f \in Rule_G$, strings $t_1, t_2, t_3, t_4, t_5 \in V^*$ and σ such that $u, v \in \sigma \in \mathrm{domain}(f)$ and $\mathrm{str}(f(\sigma)) = t_1t_2t_3t_4t_5$ and $\mathrm{str}(u) = t_2$, $\mathrm{str}(v) = t_4$. Let $\langle s_1, \ldots, s_n \rangle = \sigma$ with $u = s_k$, $v = s_l$ for $1 \leq k, l \leq n$ such that $\mathrm{str}(f(\langle s_1, \ldots, s_n \rangle)) = t_1\mathrm{str}(s_k)t_3\mathrm{str}(s_l)t_5$. For any $h \in \mathrm{Aut}_G$, $\langle h(s_1), \ldots, h(s_n) \rangle \in \mathrm{domain}(f)$. But G is string fixed, so $\mathrm{str}(f(\langle h(s_1), \ldots, h(s_n) \rangle) = t_1\mathrm{str}(h(s_k))t_3\mathrm{str}(h(s_l))t_5$. Thus, $\exists f \in Rule_G$ such that for $\sigma \in \mathrm{domain}(f)$, $h(s_k), h(s_l) \in \sigma$ and $\mathrm{str}(f(\sigma)) = t_1t_2t_3t_4t_5$ and $\mathrm{str}(h(s_k)) = t_2$, $\mathrm{str}(h(s_l)) = t_4$ so $h(s_k)\,\mathrm{PRE}\,h(s_l)$ and since $u = s_k$, $v = s_l$, $h(u)\,\mathrm{PRE}\,h(v)$. □

3.3 Linear Precedence in Human Language

Given that the human language learner is faced with the task of acquiring grammatical rules from their string outputs in the primary linguistic data, a reasonable claim is that these string outputs should be related to their inputs in a relatively fixed manner. This will have the positive consequence of facilitating the learner's acquisition of the domain elements of a given function based solely on the range of the function — i.e. based solely on the positive, overt evidence the learner receives. Thus, it is natural to propose that rules in grammars of human languages are string fixed in the sense above.

Within contemporary generative analysis of human language, the research goals are twofold. First, analysis seeks to find an explanation for the constrained amount of variation found across languages. Second, analysis seeks to explain how human infants learn the grammar of their ambient languages in an unsupervised learning environment based only on surface-apparent properties of the input. The string fixed restriction proposed above can provide a partial explanation for how human language learners are successful under these conditions. Namely, if functions of the grammar are restricted to string operations that are both fixed and transparent, then the learner exposed to expressions that are the output of those functions can more easily identify both the input sequences of the functions and the operations of the functions themselves. Thus, I propose string fixity of all functions with an arity that is greater than or equal to two as a second hypothesis regarding the class of possible grammars of human languages, leaving open the possibility that such grammars may contain unary functions with copying, deletion or reordering of string components.

Hypothesis 2. String Fixity. *For any function f in a grammar of a human language, if* $\text{arity}(f) > 1$*, f is string fixed*.

Imposing this restriction not only makes headway into the learning problem of human languages, but moreover, given the results of the above section, has the consequence that local precedence in human languages will be invariant, as all functions with an arity that is greater than or equal to two — that is, all functions that can establish a local precedence relation — will be string fixed functions.

Restricting human language grammars to those that are string fixed, in light of the selection functions made obligatory by Hypothesis 1, suggests that the sequence of categories possible in the selection functions of human language grammars are subject to an additional restriction.

Example (Category Uniqueness). Let $G_{Fr} = \langle V_{Fr}, Cat_{Fr}, Lex_{Fr}, Rule_{Fr} \rangle$ with a selection function $f \in Rule_{Fr}$ defined as:

$$f\left(\langle \textit{le chien}, \mathrm{DP}\rangle, \langle \textit{chasse}, \mathrm{V}\rangle, \langle \textit{le chat}, \mathrm{DP}\rangle\right) \quad \longmapsto \quad \langle \textit{le chien chasse le chat}, \mathrm{S}\rangle$$

By Condition (ii) of Definition 10, $(\langle \textit{le chat}, \mathrm{DP}\rangle, \langle \textit{chasse}, V\rangle, \langle \textit{le chien}, \mathrm{DP}\rangle \in \text{domain}(f)$. By Hypothesis 2, $f(\langle \textit{le chat}, \mathrm{DP}\rangle \langle \textit{chasse}, \mathrm{V}\rangle, \langle \textit{le chien}, \mathrm{DP}\rangle) \in \text{domain}(f)$ is mapped to $\langle \textit{le chat chasse le chien}, \mathrm{S}\rangle$. It follows that $f(\langle \textit{le chien}, \mathrm{DP}\rangle, \langle \textit{chasse}, \mathrm{V}\rangle \langle \textit{le chat}, \mathrm{DP}\rangle)$ and $f(\langle \textit{le chat}, \mathrm{DP}\rangle, \langle \textit{chasse}, \mathrm{V}\rangle, \langle \textit{le chien}, \mathrm{DP}\rangle)$ are distinct. Therefore, Condition (ii) of Definition 10 and Hypothesis 2 lead to a violation of Condition (iii) of Definition 10.

The interaction of string fixity with the category closure imposed by Condition (ii) of selection, thus, suggests that selection functions are barred from combining multiple elements of the same category, lest a violation of Condition (iii) ensue.

Hypothesis 3. Category Uniqueness. For any $x \in \sigma \in \text{domain}(f)$ of f a selection function in G, $\neg\exists y \in \sigma$ such that $\text{cat}(x) = \text{cat}(y)$.

Category uniqueness is motivated not only by the interaction of Hypotheses 1 and 2 but by empirical evidence found across languages. Specifically, domains wherein it seems reasonable to propose a function that combines a selector with two arguments of the same category, such as the two-place predicate example above, frequently contain evidence that such a function is not empirically adequate. This evidence may come in the form of case or agreement marking distinctions between the two arguments, the distributional restrictions on anaphors, or the 'extractability' of certain argument positions, all of which suggest the presence of a more fine-grained categorial system. Alternately, constituency and dominance relations in the derived expression can be used as evidence that a more complex generative sequence is necessary to produce an empirically adequate structure for the expression.

The invariant linear order relation in human language grammars can be further strengthened to a non-symmetric relation if a limit on the amount of homophony permissible in the domain of a selection function is imposed. This limit on homophony is here formalized as the requirement that selection functions in human language grammars contain at least one string unique pair, leaving open the empirical and theoretical question of how homophony is bounded in natural languages. The necessity of this restriction in proving the non-symmetry of local precedence is interesting given the invariant — that is, structural — nature of local precedence and the inherently non-structural nature of the strings components of expressions of the language — that is, the traditional observation that string components are paired with their syntactic and semantic forms in an arbitrary manner. Nevertheless, given that hypotheses about the nature of human language grammars are intrinsically hypotheses about the grammars of human language learners and that homophony negatively affects learnability, this restriction receives independent motivation.

Definition 17 (String Unique Pair)**.** For any $x, y \in L_G$ such that $x, y \in \sigma \in \text{domain}(f)$ for $f \in Rule_G$, $\langle x, y \rangle$ is a string unique pair *iff* $\text{str}(x) \neq \text{str}(y)$ and $\neg\exists z \in \sigma$ such that $\text{str}(z) = \text{str}(x)$ or $\text{str}(z) = \text{str}(y)$.

Hypothesis 4. String Uniqueness. For any grammar G of a human language, there must exist a pair $\langle x, y \rangle \in \sigma \in \text{domain}(f)$ for f a selection function in G such that $\langle x, y \rangle$ is a string unique pair.

Theorem 18. *If a grammar G satisfies Hypotheses 1–4, local precedence is non-symmetric in G.*

Proof. Let $\sigma = \langle (s_1, C_1), \ldots, (s_n, C_n) \rangle \in \text{domain}(f)$ for a selection function $f \in Rule_{G_{HS}}$. Then there are $u, v \in \sigma$ such that $u = (s_k, C_k)$, $v = (s_l, C_l)$ with $s_k \neq s_l$ for $1 \leq k, l \leq n$ and:

$(u\,\text{PRE}\,v \vee v\,\text{PRE}\,u)$: Since $u, v \in \sigma \in \text{domain}(f)$, $\text{arity}(f) \geq 2$ so f is string fixed, so there is some permutation, π, of $\{1, \ldots, n\}$ and strings $\alpha_0, \ldots, \alpha_{n+1} \in V^*$ such that $\text{str}(\sigma) = \alpha_0 s_{\pi(1)} \alpha_1 \ldots \alpha_n s_{\pi(n)} \alpha_{n+1}$ with s_k and s_l as proper substrings. Let $\alpha_0 s_{\pi(1)} \alpha_1 \ldots \alpha_n s_{\pi(n)} \alpha_{n+1} = t_1 t_2 t_3 t_4 t_5$. Then either $s_k = t_2$ and $s_l = t_4$ or vice versa, so $u\,\text{PRE}\,v$ or $v\,\text{PRE}\,u$.

$\neg(u\,\text{PRE}\,v \wedge v\,\text{PRE}\,u)$: Let $u\,\text{PRE}\,v$. By Condition (iii) of Definition 10, for any $\sigma' \in \text{domain}(g)$ for $g \in Rule_{G_{HS}}$ such that $u, v \in \sigma'$, either

σ' is a permutation of σ: Then $\text{str}(g(\sigma')) = \text{str}(f(\sigma)) = t_1 \text{str}(u) t_3 \text{str}(v) t_5$. By Definition 10, u and v are string distinct in σ, σ', so $\text{str}(g(\sigma')) \neq t_1 \text{str}(v) t_3 \text{str}(u) t_5$.

or σ' is not a permutation of σ: Since $|\{x \mid x \in \sigma\} \cap \{y \mid y \in \sigma'\}| > 1$, $\exists j \in Rule_{G_{HS}}\ j \circ_i g(\ldots \sigma' \ldots) = f(\sigma)$. By Hypothesis 2, f is string fixed so $\text{str}(g(\sigma'))$ must be a substring of $\text{str}(f(\sigma))$. By Definition 10, u and v are string distinct in σ, so there can be no substring t_{sub} of $\text{str}(f(\sigma))$ such that $t_{sub} = t_1 \text{str}(v) t_3 \text{str}(u) t_5$, so $\text{str}(g(\sigma')) \neq t_1 \text{str}(v) t_3 \text{str}(u) t_5$.

Thus, there is no $\sigma' \in \text{domain}(g)$ for $g \in Rule_{G_{HS}}$ such that $u, v \in \sigma'$ and $\text{str}(g(\sigma')) = t_1 \text{str}(v) t_3 \text{str}(u) t_5$, so $\neg v\,\text{PRE}\,u$. □

The presence of selection functions in grammars of human languages, given that those grammars are string fixed and that the domains of the selection functions are subject to category uniqueness, has the consequence that local precedence relations in language will be non-symmetric relations. This extends straightforwardly to a second fact about local precedence relations in grammar: it will be an asymmetric relation for any string distinct elements in the domain of a selection function

Theorem 19. *For any grammar G the local precedence relation between any string unique pair $u, v \in L_G$ is asymmetric if $u, v \in \sigma \in \mathrm{domain}(f)$ for f a string fixed selection function in G.*

Proof. Follows from the proof of Theorem 18 above. □

3.3.1 *Linear Non-Symmetry Within a Language*

Within a single human language, the linear non-symmetries that are relevant to the argument made here are those that give rise to typological classification in terms of word order properties. For example, within a given language, do objects and verbs come in object-verb (OV) or verb-object (VO) order. If it is assumed that such pairs enter into selection relationships and are by and large non-homophonous, then the linear order restrictions found within a given language will be a consequence of the selection functions that hold in that language.

3.3.2 *Linear Non-Symmetry Across Human Languages*

Across the class of human languages, the relevant linear non-symmetries are the patterns of attested and unattested word order possibilities, such as those noted by Greenberg (1966). If the word order typology of a given language can be deduced from the selection relationships that hold in that language, then assuming a similarity of categories across human languages yields the conclusion that the selection relationships will also be similar across languages. The similarity of these selection relationships will have the effect of forcing a local relationship to hold between certain categories of expressions across languages. As Cinque (2005), Abels and Neeleman (2009) and Stabler (to appear) have shown, if local relationships between expressions are held constant across languages, then only a proper subset of the logically possibly word orders in a given domain can be generated, even if languages are permitted string reordering (movement) operations and local precedence relationships are allowed to vary across languages. Finally, the correlations of linear order that are found across languages, such as that between subject-object-verb order and postpositional adpositions, can be related to the similarities in the string operations that functions perform.

4 Evidence for Selection Across Grammatical Frameworks

The definition of a selection function provided above is designed to be compatible across a number of grammatical frameworks, thus allowing a verifiability of the hypothesis of the universality of selection in human languages across theories and implementation. To illustrate the wide-ranging applicability of the definition of selection that I have provided, I now discuss it in the context of a number of generative frameworks that have been developed to account for the grammars of human languages. As will become clear in the course of this discussion, not only do each of these frameworks provide generative mechanisms that can be evaluated with respect to the conditions for selection outlined above, but in each of them we find rules that meet all of the conditions. Thus, given that the evaluation undertaken here is, in most cases, that of the grammatical formalism, not its implementation with respect to a specific language, assuming that the domain of these functions is non-empty and that Hypotheses (2)–(4) are true, the results in the previous sections will hold across these grammatical frameworks.

4.1 *Bare Grammars of Typologically Diverse Languages*

In outlining the properties of the BG framework, Keenan and Stabler (2003) develop a number of BG grammars that generate typologically diverse languages. As discussed above, the presence of selection functions in human language grammars can provide an explanation for the empirical non-symmetries found across languages. Though Keenan and Stabler do not impose the selection on the grammars they develop, it is nevertheless the case that each of the grammars they develop obey this proposed restriction on human languages.

The Toba grammar, provided by Keenan and Stabler as a model of the effects of voice marking in Toba Batak, is presented below as an illustration of the universal presence of selection functions in the Keenan and Stabler grammars.

Example. Toba (Keenan and Stabler 2003: 67-68)

Lex: $V \times Cat$

mang-	V_{af}
di-	V_{pf}
laughed, cried, sneezed	P1n
praised, criticized, saw	P2
John, Bill, Sam	NP
self	NP_{refl}
and, or	CONJ

Rule: Verb Mark (VM), Predicate-Argument (PA), Coordination (Coord)

VM

Domain			Value
mang-	t	$\longmapsto$	*mang-*$\frown t$
V_{af}	P2		P2a
di-	t	$\longmapsto$	*di-*$\frown t$
V_{pf}	P2		P2n

PA

Domain			Value	Conditions
s	t	$\longmapsto$	$s\frown t$	$x \neq y \in \{n, a\}$
P2x	NP		P1y	
s	t	$\longmapsto$	$s\frown t$	$x \in \{n, a\}$
P1x	NP		P0	
s	t	$\longmapsto$	$s\frown t$	
P2a	NP_{refl}		P1n	
s	t	$\longmapsto$	$s\frown t$	
P1a	NP_{refl}		P1n	

$$
\begin{array}{ccccccc}
 & & & \text{Coord} & & & \\
 & \text{Domain} & & & \text{Value} & & \text{Conditions} \\
\textit{and} & s & t & \longmapsto & \textit{both}^\frown s^\frown \textit{and}^\frown t & & C \in Cat - \{V_{af}, V_{pf}, \text{CONJ}, \text{P2}\} \\
\text{CONJ} & \text{C} & \text{C} & & \text{C} & & \\
\textit{or} & s & t & \longmapsto & \textit{either}^\frown s^\frown \textit{or}^\frown t & & C \in Cat - \{V_{af}, V_{pf}, CONJ, \text{P2}\} \\
\text{CONJ} & \text{C} & \text{C} & & \text{C} & & \\
\textit{and} & s & t & \longmapsto & \textit{both}^\frown s^\frown \textit{and}^\frown t & & C \neq C' \in \{\text{NP}, \text{NP}_{refl}\} \\
\text{CONJ} & \text{C} & \text{C}' & & \text{NP}_{refl} & & \\
\textit{or} & s & t & \longmapsto & \textit{either}^\frown s^\frown \textit{or}^\frown t & & C \neq C' \in \{\text{NP}, \text{NP}_{refl}\} \\
\text{CONJ} & \text{C} & \text{C}' & & \text{NP}_{refl} & &
\end{array}
$$

Though the reader is referred to Keenan and Stabler (2003) for a full exposition of these facts, the Coord rule above endows Toba with infinite generative capacity, while VM and PA together account for the fact that argument ordering and the binding of reflexives is in Toba Batak, as in other similar Austronesian languages, dependent upon the voice marking prefix found on the verb. With respect to the matter at issue here, however, both of these rules, VM and PA, can be shown to satisfy each of the conditions necessary to be a selection function, thus making Toba a selection grammar.

First, each of the sequences in the domain of both VM and PA are binary, satisfying Condition (i) of selection. In the case of PA, sequences in the domain are defined by variables over categories, a definition that automatically yields category closure. This is also true for the second element in the sequences in the domain of VM — anything in the P2 category can be verb marked — but the first element of the sequence is defined as a single vocabulary item: $\langle \textit{mang-}, V_{af} \rangle$ and $\langle \textit{di-}, V_{pf} \rangle$. In this case, however, Lex_{Toba} contains only a single lexical item of both the V_{af} and V_{pf} categories. Thus, PA is also category closed. Finally, sequences in the domain of both VM and PA are uniquely in the domain of these rules — no rules of Toba operate over either permutations or subsequences of these elements — so Condition (iii) is vacuously satisfied. Since it is clear that the domain of VM and PA are non-empty, Toba qualifies as a selection grammar. Furthermore, given that Toba is a BG formalism for a specific language, the grammar can be evaluated with respect to Hypotheses (2)–(4) as well, all of which it satisfies. Thus, local precedence in Toba will be an invariant, non-symmetric relation that is asymmetric for all string unique pairs in the domain of both VM and PA, which, for the lexicon provided, includes all sequences in the domain of both rules.

The Toba grammar may also be used as an illustration of an additional fact mentioned earlier: Hypothesis 1 requires only that grammars of human languages contain selection functions, not that they contain only selection functions. Though Toba qualifies as a selection grammar, it contains a function, Coord, which is not a selection function. As defined, Coord takes as two of its three arguments two expressions of the same category, with the string output of Coord dependent upon the order in which it takes these two arguments. As was discussed in relation to Hypothesis 2 and 3 above, this will allow Coord to produce string-distinct results when applied to permutations of the same sequence, violating Condition (iii) of selection.

Example. Coord, G_{Toba}

$$\text{Coord}(\langle \textit{and}, \text{CONJ}\rangle, \langle \textit{mang-praised}, \text{P2a}\rangle, \langle \textit{mang-criticized}, \text{P2a}\rangle) \longmapsto \langle \textit{both}^\frown\textit{mang-praise}^\frown\textit{and}^\frown\textit{mang-criticized}, \text{P2a}\rangle$$

$$\text{Coord}(\langle \textit{and}, \text{CONJ}\rangle\ \langle \textit{mang-criticized}, \text{P2a}\rangle, \langle \textit{mang-praised}, \text{P2a}\rangle) \longmapsto \langle \textit{both}^\frown\textit{mang-criticized}^\frown\textit{and}^\frown\textit{mang-praised}, \text{P2a}\rangle$$

The presence of Coord, however, does not affect the status of Toba as a selection grammar, given the presence of VM and PA.

4.2 *Categorial Grammar, 'Pure' & 'Classic'*

The categorial system and functions of traditional categorial grammar are outlined below.

Example. Categorial Grammar.

Basic Categories, *BCat***:**	$\{x_0, \ldots, x_n\}$
Categories, *Cat***:**	$x, y \in BCat$
	x/y for $x, y \in BCat$
	$x\backslash y$ for $x, y \in BCat$
Function Application (FA):	$\langle s, x/y\rangle\ \langle t, y\rangle \longmapsto \langle s^\frown t, x\rangle$ for $x, y \in Cat$
	$\langle s, y\rangle\ \langle t, y\backslash x\rangle \longmapsto \langle s^\frown t, x\rangle$ for $x, y \in Cat$

The FA rule — the only rule defined in traditional categorial grammar — is obligatorily binary, thus satisfying Condition (i) of selection and, as with VM and PA in Toba, rendering Condition (iii) vacuously satisfied with respect to sub-composition. With respect to the permutation clause of Condition (iii), the left or right cancellation of FA will simply fail to apply to non-identical permutations of the pairs in its domain. Furthermore, since the domain of FA is defined by variables over category types, FA satisfies the category closure property of Condition (ii).

Thus, traditional categorial grammars not only contain functions that satisfy the conditions of selection, but contain only functions of this type. Assuming that grammars defined in this formalism will contain expressions that make the domain of FA non-empty, traditional categorial grammars will satisfy Hypothesis 1. Moreover, since FA always applies to two distinct categories and the string component of the output is always concatenation of the string components of the input, Hypotheses (2) and (3) are also satisfied in traditional categorial grammars, with the satisfaction of Hypothesis (2) rendering local precedence invariant. The non-symmetry of local precedence in grammars defined in this formalism, however, is dependent upon the homophony bound of the lexicon — that is, the satisfaction of Hypothesis 4 — an evaluation which cannot be undertaken abstractly.

4.3 *Combinatory Categorial Grammar*

Though classic categorial grammars of the type discussed above are easily shown to be selection grammars, their generative capacity is context free and, thus, insufficient to account for the grammars of human language. This limitation has led to a number of reformulations categorial grammar, such as that of Combinatory

Categorial Grammar (CCG), which maintains the inductive definition of category types as in classic categorial grammar but extends the rules of the grammar beyond function application. The example below provides a definition of some the rule additions proposed for CCGs in Steedman (2000), omitting crossed composition and coordination, the latter of which is much like that of Toba.

Example. Combinatory Categorial Grammar.

Basic Categories, ***BCat*****:**	$\{x_0, \ldots, x_n\}$
Categories, ***Cat*****:**	$x, y \in BCat$
	x/y for $x, y \in BCat$
	$x\backslash y$ for $x, y \in BCat$
Function Application, Forward:	$\langle s, x/y\rangle \langle t, y\rangle \longmapsto \langle s^\frown t, x\rangle$
Function Application, Backward:	$\langle s, y\rangle \langle t, y\backslash x\rangle \longmapsto \langle s^\frown t, x\rangle$
Type Raising 1:	$\langle s, x\rangle \longmapsto \langle s, y/(y\backslash x)\rangle$
Type Raising 2:	$\langle s, x\rangle \longmapsto \langle s, y\backslash(y/x)\rangle$
Forward Composition:	$\langle s, x/y\rangle \langle t, y/z\rangle \longmapsto \langle s^\frown t, x/z\rangle$
Backward Composition:	$\langle s, y\backslash z\rangle \langle t, x\backslash y\rangle \longmapsto \langle s^\frown t, x\backslash z\rangle$
Backward Crossed Substitution:	$\langle s, y/z\rangle \langle t, (x\backslash y)/z\rangle \longmapsto \langle s^\frown t, x/z\rangle$
	$\forall x, y, z \in Cat$

As the traditional function application rules of classic categorial grammar remain in the CCG system, it is worthwhile to explore whether any of the additional rules affect whether or not function application satisfies the criteria of a selection function.

Conditions (i) and (ii), which are determined only on the definition of the function itself, are still satisfied for function application in CCG, as the domain of function application remains binary and category closed, as it remains defined over category variables. With regard to Condition (iii), though type-raising can, in a sense, reverse the function-argument relation, it nevertheless remains the case that the role of elements cannot be reversed without this intermediate step, which is a unary operation. Thus, the sequence of elements in the domain of function application are not in the domain of any other function the grammar, satisfying the permutation clause of Condition (iii). Moreover, since all rules of the language, save the ternary coordination rule not mentioned here, are unary or binary, the sub-composition clause of Condition (iii) is vacuously satisfied as well. Therefore, even with the addition of these other rules to the categorial grammar system, the function application still satisfy the conditions of selection and remain selection functions provided that their domain is non-empty in a given CCG grammar. Finally, as with traditional categorial grammar above, the CCG rules also satisfy the string fixity and category uniqueness (save coordination) of Hypotheses (2) and (3), leaving the non-symmetry of local precedence determinable by empirical question as to the extent of homophony in a given grammar, as per Hypothesis (4).

4.4 Principles & Parameters, Minimalism

Though a variety of approaches exist within the minimalist framework, the presence of the Merge operation is a unifying similarity across these approaches. To explore concretely the selective nature of Merge, the definitions of Merge here will be based on the discussion in Chomsky (2001).

Definition 20 (*Merge*)**.** For any two elements α, β,

$$\mathrm{Merge}_{\mathrm{set}}(\alpha,\beta) = \{\alpha,\beta\}$$
$$\mathrm{Merge}_{\mathrm{label}}(\alpha,\beta) = \{L(\{\alpha,\beta\}),\{\alpha,\beta\}\}$$

where L is a function identifying the label of $\{\alpha,\beta\}$.

As is clear in the definition, $\mathrm{Merge}_{\mathrm{set}}$ represents label-free set formation, thus generating the bare phrase structure that Chomsky (1995) assumes to be the most minimal assumption, whereas $\mathrm{Merge}_{\mathrm{label}}$ generates both a set from the two merged elements as well as a label for that set. Crucially, following Chomsky, there exists a function — here, L — responsible for identifying the choice of the label for $\{\alpha,\beta\}$. Given the controversial decision between $\mathrm{Merge}_{\mathrm{set}}$ and $\mathrm{Merge}_{\mathrm{label}}$, I will evaluate both with regard to the selection criteria outlined above, using Merge to refer to both operations when the presence of the label does not make a difference.

As Merge is an obligatorily binary operation independent of the generation of a label, Condition (i) is satisfied. If Merge is assumed to be a completely free operation, with its output filtered only at the level of the interface and not in the narrow syntax (the grammar, as construed here), then it is trivially closed over categories, independent of what one decides is the appropriate categorial system. Thus, Condition (ii) is satisfied.

Because the output of Merge is always in part set formation, for which only membership is necessary to evaluate identity, the set formed from Merge(α,β) will be identical to that formed from any permutation of this pair: $\{\alpha,\beta\}$. In the case of $\mathrm{Merge}_{\mathrm{set}}$, then, the output for any permutation will be identical as the set is the only output generated. With regard to $\mathrm{Merge}_{\mathrm{label}}$, given that L is a function that takes as input the merged set and that this set, as just noted, is always identical under permutation, then $\mathrm{Merge}_{\mathrm{label}}$, too, is identical under permutation. Interestingly, unlike the issue that arose with Coord in the BG grammar Toba, the interaction of category closure and permutation will not cause either Merge operation to fail to meet the selection criteria, as Merge does not itself generate a linear order, only a set of elements and, for $\mathrm{Merge}_{\mathrm{label}}$, a label. Finally, because Merge is obligatorily binary, no Merge operations can put together subsequences of greater than length one, thus both clauses of Condition (iii) is satisfied. Therefore, looking only at the output of Merge operations, it is clear that Conditions each of the conditions of selection are met and that grammars in this framework will be selection grammars provided that Merge has a non-empty domain.

Less clear, however, is whether any of the additional operations that have been proposed to exist in minimalist grammars can take as input the pairs in the domain of Merge — subsequences, due to the binarity of Merge, are again irrelevant — and produce as output something distinct from Merge applied to those two elements. External Merge (Move) clearly will not cause a problem here, as it is simply the special case of Merge in which an element of a set merges with the set itself. However, certain relations have been proposed to be established *in situ*, such as the probe-goal relation established under Agree. Because Agree causes a featural change of some kind, regardless of whether it is checking, valuation, sharing or deletion, the output of the Agree operation applied to any pair of elements is distinct from the application of Merge to that pair of elements. The fact that such a scenario challenges the selection grammar status of minimalism could in and of itself be used to motivate two theoretical proposals: (a) such *in situ* Agree relations do not hold (Koopman 2006) and (b) Agree itself is a subcomponent of Merge, both external and internal. With such modifications in place, Merge will fail to be a completely free operation, but will nevertheless satisfy the category closure of Condition (ii) if categories are defined by the feature matrices of the expressions of the language. Thus, the presence of Agree$\langle \alpha, \beta \rangle$ in the rules of the language will not interfere with the selectional nature of Merge, as Merge will either definitionally contain the Agree operation or will only apply to the output of Agree, not to the original $\langle \alpha, \beta \rangle$ pair.

4.5 Tree Adjoining Grammars

As a final illustration of the presence of selection functions across grammatical formalisms that have been posited for human language grammars, consider the Tree Adjoining Grammars (TAG) wherein the functions of the language operate directly over tree structures. Such grammars are discussed informally here and the reader is referred to Kallmeyer (1996) for a formal characterization.[4] The vocabulary of such grammars can be defined as the leaf yield of the set of initial and auxiliary trees in the grammar, with the categories provided by the actual structures of the trees. The generating rules in TAG are those of tree adjunction and tree substitution, though I restrict the evaluation here to adjunction given the theorem below.

Theorem 21 (Strong Equivalence of TAGs Without Substitution)**.** *For any TAG G defined as above, there is a strongly equivalent TAG G' that uses adjunction only.*

Proof. Cf. Kallmeyer (1996). □

Thus, the lexicon of TAGs can be defined as the leaf yields and tree structures closed under the adjunction operation.

The adjunction operation in TAG operates over initial trees and foot-marked trees and is illustrated in Fig. 1 on the following page. Given that this operation, like those evaluated in each of the grammatical frameworks examined thus far, is obligatorily binary, Condition (i) of selection is satisfied by adjunction in TAG. Given that the categories of TAG can be identified by the tree structures and that adjunction is defined by the nodes of the trees and the constraints that apply at those nodes, the adjunction operation is one that is category closed, satisfying Condition (ii). The binarity of TAG adjunction will, as in other rules evaluated, render the sub-composition clause of Condition (iii) vacuously satisfied. Finally, given that adjunction is defined only for pairs of initial and foot-marked trees, there will be no permutation of such pairs that has a distinct output in the domain of adjunction. Since TAG can be defined by adjunction only, this means that no permutation of

[4] The reader is also referred to work along the lines of Kasper et al. (1995) as evidence that Head Driven Phrase Structure Grammars also successfully meet the criteria of selection.

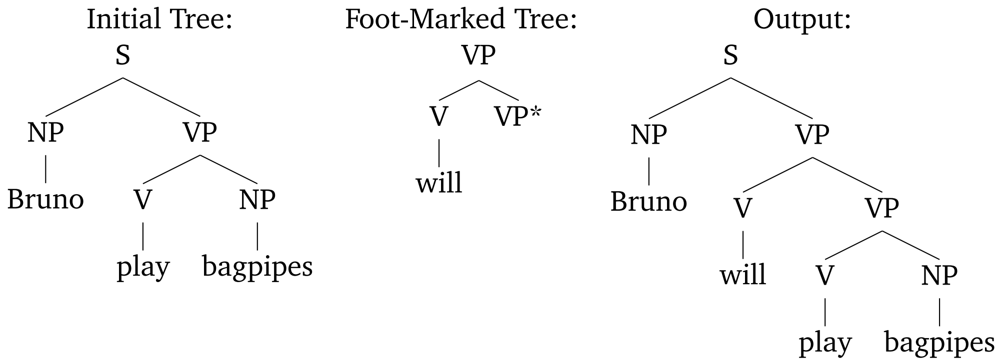

Figure 1: TAG adjunction

such pairs with a distinct output will be in the generative mechanisms of these grammars. Thus, if it is additionally assumed that adjunction has a non-empty domain, a reasonable assumption given that, like Merge, it is the only generative operation, grammars defined in TAG are selection grammars. Moreover, given that adjunction, as just noted, is defined only between pairs of initial and foot-marked trees and that its string output is defined by leaf yield, it will satisfy both string fixity and category closure. Thus, as in other cases, local precedence in TAG will be an invariant relation, with non-symmetry of this relation dependent upon the presence of string unique pairs — that is, the satisfaction of Hypothesis 4.

5 Concluding Remarks

Research across grammatical frameworks endeavors to identify the properties that characterize human language grammars and facilitate their acquisition by human language learners. One such natural property that is shown here to be pervasive across diverse frameworks is the obligatory presence of local dependencies between categories and the fixed structure of these dependencies, as defined by the selection functions of the grammar. Given an adequate set of restrictions over the string operations of the language, it has been illustrated that linear non-symmetries within and across languages can be related to this single local dependency, suggesting that other non-symmetries of language can also be considered as a result of the local selection relationships established by the grammar. This result suggests that the selection relationship established between expressions of a language may also be the underlying force behind many of empirical phenomena in linguistic research.

Acknowledgements

This work has benefited immeasurably from many discussions with many people, especially Benjamin George, Thomas Graf, Edward Keenan and Edward Stabler. Any issues that persist in this work or have been introduced into the lives of the aforementioned people are the responsibility of the author.

References

Abels, Klaus, and Ad Neeleman. 2009. Universal 20 without the LCA. In *Merging features: Computation, interpretation, and acquisition*, ed. Jose M. Brucart, Anna Gavarro, and Jaume Sola, 60–79. Oxford: Oxford University Press.

Chomsky, Noam. 1995. *The minimalist program*. Cambridge, Mass: MIT Press.

Chomsky, Noam. 2001. Beyond explanatory adequacy. In *MIT occasional papers in linguistics 20*. MITWPL.

Cinque, Guglielmo. 2005. Deriving Greenberg's universal 20 and its exceptions. *Linguistic Inquiry* 36:315–332.

Greenberg, Joseph H. 1966. Some universals of grammar with particular reference to the order of meaningful elements. In *Universals of language*, ed. J.H. Greenberg, 73–113. MIT Press.

Haegeman, Liliane, and Henk van Riemsdijk. 1986. Verb projection raising, scope, and the typology of rules affecting verbs. *Linguistic Inquiry* 17:417–466.

Kallmeyer, Laura. 1996. Tree Description Grammars and underspecified representation. Doctoral Dissertation, University of Tübingen.

Kasper, Robert, Bernd Kiefer, Klaus Netter, and K. Vijay-Shanker. 1995. Compilation of HPSG to TAG. In *Proceedings of ACL 95*, 92–99. Cambridge, Mass.

Keenan, Edward, and Edward Stabler. 2003. *Bare grammar: Lectures on linguistic invariants*. Stanford, CA: Center for the Study of Language and Information.

Koopman, Hilda. 2006. Agreement configurations: In defense of "spec head". In *Agreement Systems*, ed. Cedric Boeckx, 159–199. Philadelphia, Penn: John Benjamins.

Kroch, Anthony, and Beatrice Santorini. 1991. The derived constituent structure of the West Germanic verb-raising construction. In *Principles and parameters in comparative grammar*, ed. Robert Freidin, 269–338. Cambridge, Mass.: MIT Press.

Legate, Julie. 2002. Warlpiri: Theoretical implications. Doctoral Dissertation, MIT.

Pollard, Carl, and Ivan Sag. 1992. Anaphors in English and the scope of binding theory. *Linguistic Inquiry* 23:261–303.

Stabler, Edward. to appear. Computational perspectives on minimalism. In *Oxford Handbook of Linguistic Minimalism*, ed. Cedric Boeckx.

Steedman, Mark. 1996. *Surface structure and interpretation*. Cambridge, Mass.: MIT Press.

Steedman, Mark. 2000. *The syntactic process*. Cambridge, Mass.: MIT Press.

Affiliation

Natasha Abner
Department of Linguistics
University of California, Los Angeles
nabner@ucla.edu