UCLA Working Papers in Linguistics, Papers in Mathematical Linguistics 1
Volume 15, Article 1: 1–17, 2010

On Language Variation and Linguistic Invariants[1]

Edward L. Keenan and Edward P. Stabler

Keywords linguistic universals, linguistic invariants, language variation

We present a brief, informal overview of a formal approach to grammatical variation developed in K&S (Keenan and Stabler 2003), to which we refer the reader for proofs and other formal statements. Our purpose here is to show how we can formulate structural universals of grammar given grammars that are structurally distinct in relevant respects.

1 The One and the Many

A central quest in syntactic theory is to reconcile the audible diversity of natural languages (NLs) with the claim that they have a common, biologically determined, form.

1.1 The One

Mainstream syntactic theory (MST) from Chomsky (1957) to Bošković and Lasnik (2007) attempts this reconciliation by building it into the form of individual expressions which must satisfy general constraints on rules/derivations and representations. For a *core* expression X of L, MST asks "What is the structure of X?" The initial response, now, is often a binary branching tree in Spec-[Head-Complement] order, using language independent category symbols and structure building functions (Merge, Move). Regarding variation, some, perhaps much, is relegated to the *periphery* beyond *narrow* syntax and ignored; some is acknowledged in parameters with small ranges (e.g. question words remain in situ or front); and some lies in feature variation forcing slightly different patterns of movement/copying. Overt morphology is language specific, not determined by UG and not structurally autonomous (Bobaljik 2002) but a "reflection" of hierarchical constituent structure.[2]

[1] This paper is a slightly augmented version of an LSA plenary session presentation by the first author on January 5, 2007.

[2] Borer (2005a,b) is something of an exception to this claim. And in general imputation of properties to MST are not intended to hold exceptionlessly for those who contribute to that tradition. Note too that features called morphological can be checked without requiring morphology to be overt.

So MST focuses on the unity pole of the unity-diversity continuum, treating different languages as syntactically and semantically similar.

1.2 The Many

K&S in contrast, proposes a reconciliation that focuses on diversity. It provides a conceptual notion of structural invariant that can be satisfied by non-isomorphic structures. It formalizes the early Chomkyan desideratum that linguistic operations are *structure dependent* (Chomsky 1965: 55–56; 1975: 30–35; Radford 1997: 11–15). Often invariants are not present in the grammar as conditions on rules or representations, and are not instantiated by single expressions. Given an expression X, we ask "What expressions have the *same* structure as X?" (not "What is *the* structure of X?"). Perhaps some feeling for our relational approach to linguistic invariants can be given with the following, very imperfect, anthropometric analogy: The absolute height of human is certainly not invariant: it varies from about 2 to 8 feet. But the ratio of arm span to height is much closer to being invariant (the same). Also, in presenting our perspective we alert the reader to one difference in character between it and MST:

MST focuses on the form of syntactic theory.

Minimalist principles such as inclusiveness and economy are notation based — they constrain how we derive expressions, not directly what can be derived. Kayne (1994) opens with the central role of notation (emphasis ours): "It is difficult to attain a restrictive theory of syntax. One way. . . is to restrict the space of available syntactic *representations*, for example, by imposing a binary branching requirement,. . . The present monograph proposes further severe limitations on the range of syntactic *representations*. . . ". But restricting notation can be detrimental. Different notations may suggest different questions and generalize differently to new phenomena. For example, in the early 1960's context free grammars and categorial grammars were shown to define the same class of languages (Bar-Hillel, Gaifman, and Shamir 1960). But it was categorial grammar that most naturally captured function-argument structure (Lewis 1970; Montague 1973), effectively creating the field of formal semantics as we know it today. The *significant* properties of objects are invariant under changes of descriptively comparable notation, notational artifacts aren't. The truth of *x is hotter than y* does not vary according as we measure temperature in Fahrenheit or Celsius. The truth of *x is twice as hot as y* does. So let us focus on regularities of linguistic nature, not their notational expression, adopting the slogan

If you can't say something two ways you can't say it.

1.3 Back to the Many

Here first is a semantic generalization in propositional logic based on notationally distinct objects. (1a) uses standard infix notation for conjunction and disjunction, (1b) uses prefix (Polish) notation. $\models$ is "entails".

(1) a. $((P \vee Q) \wedge \neg P) \models Q$
b. $\wedge \vee PQ\neg P \models Q$

The notational variation is non-trivial: prefix notation uses no parentheses and has no structural ambiguities. Eliminating all parentheses from infix notation yields structural and semantic ambiguity: $((P \wedge Q) \vee R) \not\equiv (P \wedge (Q \vee R))$. On the other side, students of logic tend to find infix notation easier to understand, especially in long formulas. But the entailment fact in (1a,b) is *the same*: a disjunction of formulas conjoined with the negation of one entails the other. The entailments expressible in the two notations are the same. The synonymy of $(P \wedge Q)$ and $\wedge PQ$ is determined by independent compositional interpretation. We have no need to derive them from a common "deep structure" or map them to a common "LF".

2 On Structure

We treat a language as a compositionally interpreted set of expressions defined by a grammar $\mathrm{G} = (Lex_\mathrm{G}, Rule_\mathrm{G})$, where Lex_G, the *lexicon* of G, is a finite set of expressions called *lexical items* and $Rule_\mathrm{G}$ is a set of *structure building functions* called *rules*, which iteratively map tuples of expressions to expressions. The *language* L_G *generated* by G is the set of expressions derived from Lex_G by finite iteration of the rules. So L_G is the closure of Lex_G under $Rule_\mathrm{G}$. A *descriptively adequate* G for a NL L is *sound* (everything it generates is judged by competent speakers to be in L) and *complete* (everything speakers accept is generated).

A grammar G is *isomorphic* to a grammar $\mathrm{G}', G \cong G'$, iff there is a bijection $h : L_\mathrm{G} \to L_{G'}$ matching the functions F in $Rule_\mathrm{G}$ with the F' in $Rule_{G'}$ so that when F derives an expression z from some expressions $x_1, \ldots, x_n$ then F' derives $h(z)$ from $h(x_1), \ldots, h(x_n)$, and conversely. An isomorphism h from G to itself is called an *automorphism* (or *symmetry*) of G. They are the ways of substituting one expression for another within a language which do not change how expressions are built. So they preserve structure: F builds z from $(x_1, \ldots, x_n)$ iff F builds $h(z)$ from $(h(x_1), \ldots, h(x_n))$, all automorphisms h. The set Aut_G of automorphisms of G represents the "structure" of G. It contains the identity map, and it is closed under function composition and inverses, so is a group, the *automorphism* (or *symmetry*) *group* of G, noted Aut_G or Sym_G.[3]

2.1 Complexity

In cases of interest (natural languages) L_G is infinite. Since automorphisms map an infinite set to itself one might think that there are massively many of them, and thus hard to study and characterize in any given language. But in fact in cases of interest the number of automorphisms of G is finite. The reason is that the value an automorphism h assigns to a derived expression $F(x_1, \ldots, x_n)$ is uniquely determined by the values it assigns to the expressions it is built from, namely $x_1, \ldots, x_n$. So

[3] Rules are partial functions, so we define $g \circ f$ as that map with domain $\{a \in \mathrm{Dom}\, f \mid f(a) \in \mathrm{Dom}\, g\}$ given by: $(g \circ f)(a) = g(f(a))$.

once we have given the values of an automorphism on the lexical items — finite in number in cases of interest — we have determined its values at all expressions: $h(F(x_1,\ldots,x_n)) = F(h(x_1),\ldots,h(x_n))$. ($\mathrm{Aut}_G$ is *forced* to be finite if Lex_G is finite and none of its elements is also a derived expression).

2.2 *Invariants*

An expression s has the same structure as an expression t iff $h(s) = t$ for some automorphism h. We do not mention here "the structure" of an expression — an epistemological plus, as different syntactic theories agree more readily that *John sang* and *Bill danced* have the same structure, than they do about what the structure of *John sang* is. We define:

Definition 1. A relation R on L_G is **invariant** iff $h(R) = R$, all automorphisms h. That is, replacing all tuples $(s_1,\ldots,s_n) \in R$ by $(h(s_1),\ldots,h(s_n))$ leaves R unchanged. So $w \in L_G$ is **invariant** iff $h(w) = w$, all automorphisms h.

Thesis. $w \in L_G$ is a *grammatical formative* ("function word") iff $w \in Lex_G$ and w is invariant.

So function words are items that are isomorphic only to themselves. Replacing them with something else changes structure (usually destroying it yielding ungrammaticality). In the models of grammars in K&S, reflexive pronouns, case, voice, and agreement markers are provably invariant. We expect that lexical invariants correspond to heads of functional projections in more usual terminology. What is new here is a characterization of what counts as *functional* — namely, being a linguistic object (including bound morphemes) which can only be mapped to itself by the automorphisms, the structure preserving functions, of the grammar).

As an item that a function f maps to itself is called a *fixed point* of f, we characterize the functional expressions in a grammar as the fixed points of the syntactic automorphisms. We turn now to some generalizations built on this notion of invariant.

2.3 *Degrees of Invariance, Language Change*

Invariance generalizes to a scalar by:

Definition 2. a. $Inv(x)$, the *degree of invariance* of a linguistic object (expression, property, relation) x is the proportion of automorphisms that fix x, that is, that map x it itself. So if x is invariant per Definition 1, then $Inv(x) = 1$, as all automorphisms map x to itself.

b. x is *more invariant than* x' iff $Inv(x) > Inv(x')$.

These definitions provide a rigorous way to say that conjunctions and prepositions are more grammaticized than intransitive verbs. And we can use this notion to represent the grammaticization (Hopper and Traugott 1993) of an expression w by saying that $Inv(w)$ increases over time. If $Inv(x)$ reaches 1 then x is fully grammaticized.

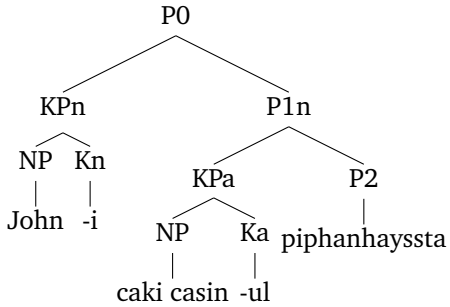
2.4 *Relation Invariants*

K&S prove that for all G (not just G for NLs), *is a constituent of* and *c-commands* are invariant relations. These notions require a more general than usual definition since we have not limited ourselves to rules whose action is modeled by standard labeled trees. Such functions basically just derive expressions by concatenation, not, for example, substitution (widely used to derive consequences from premisses in logical deductive systems for example). Here are some examples of the generalized definitions:

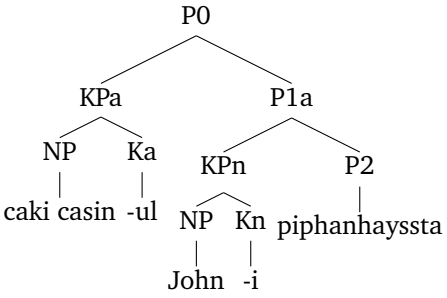
Definition 3. u is an *immediate constituent* of v iff v is the value of a rule F at a tuple $(s_1,\ldots,s_n)$ and u is one of the s_i. u is a *constituent* of v iff u is v or u is an immediate constituent of a constituent of v. s is a sister of t in v iff $s \neq t$ and for some constituent u of v, s and t are immediate constituents of u. s *c-commands* t in u iff s is a constituent of a sister of t in u.

All these relations are provably invariant in all G. In contrast, the property of being an *anaphor* and the relation *x is a possible antecedent of an anaphor y in z* are not invariant in all G, but they are for all G we have constructed to model NLs. This illustrates invariants among non-isomorphic structures. K&S's grammar for minimal clause structure in Korean generates (2b) where anaphors like *self-acc* asymmetrically c-command their antecedents. To see that *self-acc* is an anaphor one must check its semantic interpretation in K&S (see below): it maps a binary relation R to the property $\{a|aRa\}$).

(2)



a. John-nom self-acc criticized
'John criticized himself'



b. self-acc John-nom criticized
'John criticized himself'

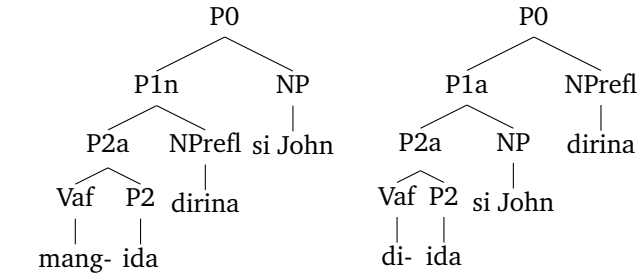
Each of these Ss is interpreted compositionally, and they receive the same interpretation: True iff $(j,j) \in$ criticize, j the denotation of *John*. This fact is not different in kind from the fact that $(P \wedge Q)$ is logically equivalent to the structurally distinct $\neg(\neg P \vee \neg Q)$ in propositional logic.

Lovers of trees Beware! (2a,b) are isomorphic qua ordered labeled trees (same branching structure, each has distinct labels just where the other does). But the expressions are not isomorphic in our grammar: if an automorphism h mapped *john* to *self* and *-nom* to *-acc* it could map *John -nom laughed* to *self -acc laughed*, which is not in our model language, contradicting that h is an automorphism. See K&S p.50.

(3) is from Toba Batak (W. Austronesian; N. Sumatra). See (Schachter 1984; Cole and Hermon 2008). The distribution of anaphors in Toba, as in other W. Austronesian

languages (Tagalog, Malagasy, Balinese) is conditioned by verb voice, the reflexive often occurring as what we thought was a "subject". And as in the Korean case, lexical items occurring in both Ss have identical interpretations. So again anaphors may asymmetrically c-command their antecedents.

(3)



a. mang- see self John
'John sees himself'

b. di- see John self
'John sees himself'

Theorem 4. *The case markers* -nom *and* -acc *and voice markers* mang- *and* di- *are invariant in their grammars in K&S (pp.49 and 70, respectively).*

Thus is morphology "structural" in the *same* sense as constituent structure: both are *preserved by all the automorphisms*.

Theorem 5. *a. The property of being an anaphor is invariant in K&S's model grammars for English, Korean and Toba Batak (pp.36, 50 and 70, respectively)*

b. The Anaphor-Antecedent relation is invariant in K&S's models (pp.53 and 71, respectively), as it is in their model of English, not illustrated here, in which the usual c-command conditions hold.

Theorem 2 builds on two properties which distinguish our approach to anaphora from more standard ones:

P1. *anaphor* is semantically defined, so we are not free to stipulate that himself in English, *caki casin* in Korean, and *dirina* in Toba Batak are anaphors. Rather, the anaphoric expressions in a language are those whose semantic interpretation satisfies our definition (the same for all languages). Often the anaphors in a language cannot be listed as there are infinitely many (all but finitely many being syntactically complex).

P2. The structural means used to identify anaphors and their antecedents are not structurally the same across languages. Nonetheless the anaphor-antecedent relation is, we claim, invariant in each NL grammar (and so universally invariant).

We explain P1 and P2 and illustrate with examples, then formulate P2 as a universal claim U1.

Referentially Autonomous Expressions (RAEs) such as *Zelda*, *most men who Zelda dates*, etc. combine with $n+1$-ary predicates (P_{n+1}s), to form n-ary ones (P_ns). For simplicity we limit ourselves here to P_ns, $0 \leq n \leq 2$. Semantically P_ns denote

n-ary relations over whatever domain E is under consideration.[4] RAEs denote *Type-1* functions — they map $n+1$-ary relations to n-ary ones, thus reducing arity by 1. But not just any Type-1 function is a possible RAE denotation. The value such a function assigns to a binary ($n+1$-ary) relation is determined by the values it assigns to the unary relations (subsets of E). Thus *Dana praised every student* is true iff Dana is in the denotation of *praised every student*, the set of objects that praised every student, that is, the set of objects b such that *(every student)* holds of the set of things that b praised. For R a binary relation write bR for $\{d|\ (b,d) \in R\}$. So bR is the set of objects d that b stands in the relation R to. Then the possible RAE denotations F are those Type-1 functions satisfying (4):

(4) For R a binary relation, $F(R) = \{b|F(bR) = 1\}$.[5]

Theorem 6. *The Type-1 F satisfying (4) are just those which satisfy the AEC (Keenan 1988):*

> **Accusative Extensions Condition (AEC)** *For all $x, y \in E$, all binary relations R, S, if $xR = yS$ then $x \in F(R)$ iff $y \in F(S)$.*

So *most men who Zelda dates* satisfies the AEC since, for example, whenever Sue distrusts just the people that Ann likes then *Sue distrusts most men Zelda dates* and *Ann likes most men Zelda dates* must have the same truth value (both true, or both false).

Now observe that expressions like *himself, herself*, etc. as they occur in *No model hates herself* fail the AEC. If Sue distrusts just Jean, Robyn, Amy, Ann, Mary, and Pat and those are exactly the people that Ann likes then *Sue distrusts herself* is false and *Ann likes herself* is true. Such expressions do however satisfy a weaker invariance condition (Keenan 1988):

Accusative Anaphor Condition (AAC) For all $x \in E$, all binary relations R, S, if $xR = xS$ then $x \in F(R)$ iff $x \in F(S)$.

So if x bears R to the same things that x bears S to then x bears R to himself/herself iff x bears S to himself/herself. And we may, on first pass, define anaphoric expression by:

Definition 7. An expression α in L_G is an *anaphor* iff the interpretation of α satisfies the AAC in all models and fails the AEC in some.[6]

One checks by example that the underlined complex expressions in (5) are anaphors:

(5) a. John criticized every student but himself /no student but himself

[4] Note that P_0s denote subsets of $E^0 = \{\emptyset\}$, so a P_0 denotes an element of $\{\emptyset, \{\emptyset\}\} = \{0, 1\}$, our usual representation of the set of two truth values.

[5] Interpreting b as an n-tuple and bR as $\{d|\ (b,d) \in R\}$, the equation in (4) is the general condition for F a map from P_{n+1}s to P_ns.

[6] This definition must be generalized to account for anaphors in a wider range of contexts: *Mary protected John from himself, every worker's criticism of himself*, etc.

b. John criticized only himself and the teacher /neither himself nor the teacher
c. John knows many people smarter than himself /no one as verbose as himself
d. He nominated someone other than himself /He wouldn't nominate anyone other than himself

3 Structural Universals

We consider some candidates U1 — U5 as structural universals of human language. Our purpose is to illustrate how such claims can be formulated in our framework. We think they are plausible, but much empirical investigation is needed.

U1. The property of being an anaphor is a structural invariant of human language.

U1 says that anaphors are always mapped to anaphors by the syntactic automorphisms. Note that even if each lexical anaphor is mapped to itself, each complex anaphor typically will not be. An automorphism of English might map *himself* to *himself* but map *no doctor but himself* to *no lawyer but himself*. Nonetheless, U1 says that in each L_G anaphors have some syntactically distinctive properties, ones which may fail to be comparable across languages. In nuclear clauses in Batak for example they concern distributional constraints with respect to *mang-* vs. *di-* prefixed verbal roots. In Korean they concern case marking, and in English it is Principle A that distinguishes the distribution of anaphors: *John laughed* is fine, *Himself laughed* is not. These same observations support U2:

U2. The relation *x is a possible antecedent of an anaphor y in z* is a linguistic invariant.

3.1 Theory Design

Merely representing a NL as a pair $(Lex_G, Rule_G)$ is not a theory — at best it is a common denominator of theories such as Minimalism, HPSG, LFG, RG,... But two features of our approach do have a liberating effect on theory design: (i) the requirement of a compositional semantics, and (ii) the theorem that morphology may be structural. From (ii), in designing a grammar we are free to condition the distribution of anaphors (semantically defined) directly in terms of case or voice, it is not necessary to derive them from, or reconstruct them to, forms c-commanded by their antecedents.

From (i), semantic representations — on our view compositionally interpreted audible structures — differ from language to language, whereas in MST (Higginbotham 1985, Chomsky 1986:156, Hornstein 1995:§1) LFs for different languages are assumed or argued to be roughly the same. They claim that primary semantic data do not suffice for the child to infer the structure of LF parameters (in distinction to primary phonetic data which do permit the inference of diverse phonologies).

Since we do learn to use language meaningfully the semantic module must be innate, hence essentially the same across speakers of different languages.

The argument does not convince.

No characterization of primary semantic data is given, nor are reasons why they are insufficient to set "LF parameters". In fact children learn to use language meaningfully as they learn to pronounce it. Simple situations in which they follow commands, make requests, answer questions, disagree,... are easily seen to contribute to learning the meaning of expressions — reference (count and mass), basic argument structure and theta roles, modification (*Hand me the red crayon. — No, no, not the* blue *one, the* red *one!*), etc. And of course learners of different languages are interpreting different expressions, an unproblematic fact as our example with $((P \vee Q) \wedge \neg P)$ and $\wedge \vee PQ\neg P$ shows. (6) and (7) below exhibit pairs of logically equivalent sentences with different internal structures — showing that different structures may determine the same (logical) meaning.

(6) a. Between a third and two-thirds of Americans brush their teeth regularly
b. Between a third and two-thirds of Americans don't brush their teeth regularly

(7) a. All but two students read at least as many poems as plays
b. Exactly two students read more plays than poems

(6) is surprising as they differ just in that the predicate in (6b) negates that in (6a). The pattern is general as long as the fractions sum to 1 (Keenan 2004). We should note that LF is not designed to represent meaning in general. It doesn't even support a definition of entailment (Chomsky 1986:67n11) and several of the semantic notions it does represent are esoteric and likely learned later than the notions we mentioned above. For example, relative scope of quantifiers only arises with two quantified arguments of a given predicate. Languages provide many means for distinguishing the arguments of transitive verbs (word order, agreement, case marking, chain of being orders) but do not systematically disambiguate scope (Keenan 1988). T. Lee (1986) supports experimentally that both English and Chinese children understand some basic quantifiers in intransitive Ss by age 4, but even by age 8 do not have adult competence on their relative scope judgments in transitive Ss. Even in formal logic it was only with Henkin (1961) that we learned to construct formulas with universal and existential quantifiers lacking scope dependencies.

3.2 Property Invariants

Is the property of being a lexical item invariant in all G for NL? That is, do automorphisms always map lexical items to lexical items? This is a possible empirical truth (not a theorem) even though the lexicons of different NLs differ. Similarly, is the property of having a given grammatical category invariant? That is, do all automorphisms map each phrase of category C to a phrase of category C? Taking this as axiomatic would provide a universal structural role for categories. But K&S show

this fails: for example, it is possible to design a grammar in which an automorphism exchanges the masculine and feminine nouns and adjectives.

Another idea is that a category C is an equivalence class of expressions defined by a coarsest congruence, where a *congruence* is an equivalence with the property that for every rule F that applies to expressions $(s_1,\ldots,s_n)$, if $(t_1,\ldots,t_n)$ is such that s_i is equivalent to t_i, all $1 \leq i \leq n$, then F applies to $(t_1,\ldots,t_n)$ and $F(s_1,\ldots,s_n)$ is equivalent to $F(t_1,\ldots,t_n)$. But this fails too for some reasonable grammars. It rules out, for example, grammars with certain kinds of identity conditions. For example, a rule that allowed coordination of any two *distinct* NPs — *both John and Bill* but not *both John and John* — would treat *John* and *Bill* as the same category even though one cannot generally replace the other in the domain of coordination.

K&S argue instead for the weaker U3. U3 provides a universal structural role for categories which allows grammars of different languages to have different categories. Embarrassingly the field has no purely syntactic definition of category that allows us to infer that category C in language L and category C' in language L' are the same. See Baker (2003) for some discussion.

U3. For all stable automorphisms h, all expressions x, if x has category C so does $h(x)$.

Definition 8. a. An automorphism of a grammar G is *stable* iff it extends to an automorphism of each lexical extension of G.

b. G_n is a *lexical extension* of G iff there is a sequence $(G_1,\ldots,G_n)$ of grammars such that each G_{i+1} differs from G_i just by the addition of a single lexical item isomorphic to one in Lex_{G_i}. (The set of stable automorphisms in Aut_G is always a subgroup of Aut_G).

K&S's model of Spanish for example has two gender classes of nouns, Nm and Nf, and overt agreement of adjectives and dets with Nouns. There are automorphisms h which interchange the lexical Nms and Nfs, provided these two sets have the same number of members. If we add one new member to just one of the classes we can no longer interchange them by an automorphism, so such h are *unstable*: their status as automorphisms is not preserved under trivial additions to the lexicon. A fourth candidate for a structural universal is:

U4. Theta role assignment is invariant.

So if x bears theta role τ to y in z then $h(x)$ bears τ to $h(y)$ in $h(z)$, h any automorphism. That is, theta role assignment Θ is a function of structure. If *Ed* has different theta roles in *Ed ran* and *Ed arrived* then these Ss must be non-isomorphic (per current theories). But *Ed* may occur in structurally distinct environments and still be assigned the same theta role. U4 is strictly weaker than UTAH, which requires that Θ be (structurally) one to one: so same theta role $\Rightarrow$ isomorphic sources. But functions may take the same values at different arguments: $(2+3)=(1+4)$. So active subjects and their corresponding passive agent phrases may originate in non-isomorphic configurations.

3.3 *Greenberg Duality*

We close with a much more contentious candidate universal, U5 below. Two languages are word order *duals* if the expressions of one are the mirror images of those of the other. Lexical items are self dual. A rigid SXOV L, like Turkish, is (isomorphic to) the dual of a rigid VOXS language, like Malagasy. Formally, the dual v^d of a sequence $v = (v_1, v_2, \ldots, v_n)$ of lexical items is just its mirror image, $(v_n, \ldots, v_2, v_1)$. The dual K^d of a set K of expressions is the set of w^d for $w \in K$. If F is in $Rule_{\mathrm{G}}$ then its dual F^d is that function with domain $\mathrm{Dom}(F)^d$, that maps $(w_1^d, \ldots, w_n^d)$ to the dual of $F(w_1, \ldots, w_n)$. We define G^d to be that grammar with the same lexical items as G and whose rule set is the set of duals of rules of G. We have:

Theorem 9. *a.* $(L_{\mathrm{G}})^d = L_{\mathrm{G}^d}$

b. $\mathrm{G} \cong \mathrm{G}^d$, *the map sending each* $w \in L(G)$ *to* w^d *is an isomorphism.*

Theorem 9a just says that the dual of the language is the language generated by the dual grammar, so we defined G^d right. Theorem 9b says that G and G^d are isomorphic.

U5. The set PHG of possible human grammars is closed under isomorphism.

U5 just says that if $G \in \mathrm{PH}_{\mathrm{G}}$ and $\mathrm{G} \cong \mathrm{G}'$ then $G' \in \mathrm{PH}_{\mathrm{G}}$. Our justification for U5 is that UG only selects for structure not content and thus cannot distinguish between isomorphic variants (though other constraints, say ones on possible phonological systems, might rule out some isomorphic images as being non-natural on other grounds).

Corollary (Duality): PH_{G} is closed under duals. (From U5 and Theorem 9b).

The Duality Corollary makes us hesitant to accept Kayne's Antisymmetry axiom, which forces right branching structures. If only such grammars were acceptable then PH_{G} would not be closed under duals. But since there are left branching Ls (Toba Batak, Malagasy) the Antisymmetry axiom must be weakened.

But U5 is also problematic. The Duality Corollary suggests an equal distribution of word order types and their duals, which is not the case. Right branching (SXOV) Ls are the most common across areal and genetic groupings whereas VOXS Ls are a clear minority, but include Malagasy (Keenan 1976) and several other Austronesian languages, and Tzotzil (Aissen 1987) and several other Mayan languages. Worse, the OVS duals of SVO languages have (to our knowledge) just Hixkaryana (Carib; Brazil) (Derbyshire 1977) as a well attested exemplar, while the OSV duals of VSO Ls are just barely attested: *Ethnologue* (Gordon and Grimes 2005) cites Jamamadi, an Arawakan language in Brazil.

But immediate rejection of U5 would be short sighted. Much work in the physical sciences, esthetics, mathematics, and the philosophy of science supports both the fundamental role of symmetry in the phenomena under study and also the presence of asymmetries and spontaneous symmetry breaking. Acknowledging and studying these asymmetries and symmetry failures has been significant stimulus to deeper understanding. "The study of anomalies now plays an important role in our search for the symmetries of nature" (Zee 1986: 300).

We can hardly summarize here the basic, and at times dazzling and provocative, work in this area. We just note a few highlights that have influenced our thinking and point the reader to several accessible and enlightening introductions to symmetry and symmetry breaking. Weyl (1952) is a classic, discussing symmetries, and asymmetries, in physics, biology, and art. Bunch (1989) and Gardner (2005) are more recent and very informative. Darvas (2007) focuses more on symmetry in art, and Zee (1986) on symmetry in physics. On the mathematical side we note Stewart and Golubitsky (1999) plus any basic textbook on group theory, the language of symmetry and invariance, for example Rotman (1999: §1–3). On the more philosophical and epistemological side we have found van Fraassen (1989) and Wigner (1979) enlightening.

Concerning the conceptually fundamental nature of symmetry and invariants (what remains unchanged under the action of the symmetries) the first author must acknowledge his awe at Felix Klein's Erlangen dissertation (Klein 1893) in which he stood Euclidean geometry on its head. The objects of study became whatever was invariant under the action of translations, rotations, and reflections. Other geometries are invariants of other transformations. Later topology, which grew out of geometry, became the study of continuous transformations and topological invariants those objects, properties, . . . which remain invariant under these transformations. In 1918 Emmy Noether (Bunch 1989:95; Brewer and Smith 1981; Cole 1997:183), at Erlangen, proves that symmetry principles in physics (including relativity theory) imply conservation laws. Indeed Hermann Weyl (cited in Bunch 1989:144) claims "The entire theory of relativity. . . is but another aspect of symmetry". Gardner 2005: 337 quotes Einstein to the effect that invariant theory would have been a better name for his achievement than relativity. More recently symmetries have been used to predict new elementary particles (Sternberg 1994). And symmetry breaking is the exclusive subject of a recent monograph (Strocchi 2005). Still, "Why", asks Feynman, "is nature so nearly symmetrical?" (Bunch 1989: 189).

Cotton (1990), a standard textbook, exemplifies the utility of symmetry in chemistry by using group theory to classify molecules by structure: ". . . the number and kinds of energy levels that an atom or molecule may have are rigorously and precisely determined by the symmetry of the molecule or of the environment of that atom." Cotton 1990: 3. In mathematical domains group theory, as noted, the mathematics of symmetry and invariants, has become a major subfield in mathematics from its initial impetus by Galois (1811–1832). In logic, Tarski (1986) presents informally the idea that "logical operations" are simply the most general ones (in distinction to translations, etc. which must obey constraints in addition to being permutations of the domain. Keenan (2001) studies this idea more formally). And of interest Roman Jakobson (1963) pushed the notion of linguistic invariant in the famous 1963 Universals conference: "Naive attempts to deal with variations without attacking the problem of invariants are condemned to failure" (Jakobson 1963: 272).

We return now to the unequal distribution of word order duals — which, being isomorphic but distinct, we might have expected to be equally distributed (perhaps assuming a Leibnizian "Sufficient Reason" basis for Nature). But as we have seen they are not. Nor is right and left spiraling DNA. As far as we know, all animals

are built from right-handed DNA, though it seems that a little left-handed DNA has been found in nature, and it can be synthesized. So what accounts for the statistical disparity? For that matter, why do righthanders outnumber lefthanders? In some cases at least we feel the choice was arbitrary, but once established it self-perpetuated. Driving on the left or the right is an arbitrary convention, entailing building cars with the steering wheel on the left or the right. But as one came to dominate in continental Europe Sweden was pushed to "walk in step", that is, drive on the right. Britain is still holding out. (Japan, Australia, and New Zealand continue to drive on the left).

Now, to return to properly linguistic considerations, how good is the comparison between linguistic symmetries and invariants and those in physical or mathematical fields? Here the comparison is very good indeed. In both cases we study what is preserved under the action of classes of structural functions. So we are not simply making some analogy here, we are using linguistic science as another case where symmetries and their invariants can be studied. But will this approach lead to enlightening results, as it has in the fields discussed above? And here of course we don't know how enlightening this approach will be until we pursue it detail to see where it leads. There are however some grounds for a preliminary positive assessment.

In the first place, we have claimed a methodology that permits the description of structural regularities across structurally non-isomorphic languages. This is already a strong reason to pursue our approach. And secondly, we can benefit from the massive amount of work that has gone into the study of symmetries and the mathematical apparatus (group theory) needed to describe them. Minimally we can ask if there is something distinctive about the automorphism groups (symmetry groups) for grammars of human languages. Can we distinguish human languages from other formal systems by the structure of their symmetry groups? To pursue an answer to such a big question we need to know massively more about the group structure of well studied grammars.

Do any of the following traditionally structural properties of grammars force some distinctive property on the automorphisms of G: paradigm, inflectional morphology, allomorph, subcategory, extraction and copy rules (Kobele 2006), category changing operations, case and voice marking? Are locality constraints or cyclic domains (or phases) characterizable in terms of automorphisms? We don't know. Some simple classes of objects have characteristic automorphism groups. For example regular polygons (n-gons) have dihedral groups (n rotations, n reflections). Do the Aut_G for natural languages have any characteristic structure? We simply don't know.

But merely modeling the simplest type of agreement phenomena we learned that some automorphisms are unstable, and may fail to extend to automorphisms of grammars that trivially augment the original just by adding a new lexical item isomorphic to an old one. So unstable auts may indicate basic linguistic symmetries that can be overridden by default forms, as in coordination. How much other linguistic information will be coded in the automorphisms? Can we for example reconstruct the grammatical category distinctions among lexical items given the automorphisms? For example we hypothesize that for a given lexical item d, the

other lexical items of the same category as d are just those in Orbit(d) under the stable automorphisms, where Orbit(d) is the set of expressions that a stable automorphism can map d to.

Finally, the Corollary invites a deeper, less speculative comparison with MST: eliminating redundancy can be overridden. Among the virtues of axiom systems mathematicians include independence (non-redundancy) — no axiom is to follow from the others. But:

Symmetry trumps redundancy.

For example, common axiomatizations for Boolean algebra contain the two distributivity laws:

(8) a. $(x \wedge (y \vee z)) = (x \wedge y) \vee (x \wedge z)$
b. $(x \vee (y \wedge z)) = (x \vee y) \wedge (x \vee z)$

In (8a) meet $\wedge$ distributes over join $\vee$ and in (8b) join distributes over meet. If (8a,b) are both removed then neither is entailed by the remaining axioms. But either one is eliminable and provable from the remaining ones, so including both is redundant. The reason for the redundant inclusion is symmetry: There is no basis for choosing among (8a) and (8b) — each is derivable from the other by duality. Choosing just one as an axiom would imply that it was basic and the other "derived", creating an asymmetry where there is none. So here symmetry conflicts with redundancy and symmetry wins.

Lest the reader think Boolean algebra is atypical, here is a more fundamental example. A group is set G with an associative binary relation $\bullet$ satisfying two additional axioms:

(9) a. Identities: There is an $e \in$ G such that for all $x \in$ G, $e \bullet x = x$ and $x \bullet e = x$.
b. Inverses: For all $x \in$ G there is a $y \in$ G such that $y \bullet x = e$ and $x \bullet y = e$.

Now the second conjuncts of (9a) and (9b) can be simultaneously eliminated and proven from the remaining axioms. But doing that implicates that having a left identity element and a left inverse is more basic than having a right identity and a right inverse — their existence being a "mere" theorem, not axiomatic. Again these implicatures introduce an unwarranted asymmetry. We can in fact keep both right hand conjuncts in (9a,b) and eliminate the two left hand ones, deriving them as theorems. Again symmetry trumps redundancy.

4 A Goal of Descriptive Linguistics: Classify Human Grammars by Their Symmetry Groups

How many ways are there to build a Predicate-Argument system? A Modifier system? These questions suggest lower bounds on the expressivity of NLs. We just hint at how to flesh them out. First, for an arbitrary domain E, write Pn for $P(E^n)$, the set of n-ary relations over E (the subsets of En). Write $[A \to B]$ for the set of functions from A into B. Then a minimal Predicate-Argument system is the set of Pn, $0 \leq n \leq 3$ with argument algebras $[Pn+1 \to Pn], 0 \leq n \leq 2$. A modifer system

contains $[Pn \to Pn]$ for $n = 1, 2$ at least, where the functions are restricting: $F(p) \subseteq p$. See Keenan (1981). Obviously we need much more in terms of boundary conditions on denotable objects: quantifiers deriving arguments from $P1$'s, nominalizers of various sorts deriving arguments from Pn's, boolean and binding operators, etc.

References

Aissen, Judith. 1987. *Tzotzil clause structure*. Dordrecht: Reidel.

Baker, Mark C. 2003. *Lexical categories: Verbs, nouns, and adjectives*. NY: Cambridge University Press.

Bar-Hillel, Y., C. Gaifman, and E. Shamir. 1960. On categorial and phrase structure grammars. *Bulletin of the Research Council of Israel* F9:155–166. Reprinted in Y. Bar-Hillel, *Language and Information: Selected Essays on their Theory and Application*. NY: Addison-Wesley, 1964.

Bobaljik, Jonathan David. 2002. A-chains at the interfaces: Copies, agreement and covert movement. *Natural Language and Linguistic Theory* 20:197–267.

Borer, Hagit. 2005a. *In name only*, volume I of *Structuring Sense*. Oxford: Oxford University Press.

Borer, Hagit. 2005b. *The nominal course of events*, volume II of *Structuring Sense*. Oxford: Oxford University Press.

Bošković, Željko, and Howard Lasnik. 2007. *Minimalist syntax: The essential readings*. Oxford: Blackwell.

Brewer, J. W., and M. K. Smith, ed. 1981. *Emmy noether: A tribute to her life and work*. NY: Marcel Dekker.

Bunch, Bryan. 1989. *Reality's mirror*. NY: Wiley.

Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, Massachusetts: MIT Press.

Chomsky, Noam. 1975. *Reflections on language*. NY: Pantheon.

Chomsky, Noam. 1986. *Knowledge of language*. NY: Praeger.

Cole, K. C. 1997. *The universe and the teacup*. NY: Harcourt Brace.

Cole, Peter, and Gabriella Hermon. 2008. VP raising in a VOS language. *Syntax* 11:144–197.

Cotton, Albert F., ed. 1990. *Chemical applications of group theory 3rd edition*. NY: Wiley.

Darvas, G. 2007. *Symmetry*. Boston: Birkhäuser.

Derbyshire, Desmond C. 1977. Word order universals and the existence of ovs languages. *Linguistic Inquiry* 8:590–599.

Gardner, Martin. 2005. *The new ambidextrous universe 3rd revised edition*. NY: Dover.

Gordon, Raymond G., and Barbara F. Grimes, ed. 2005. *Ethnologue: Languages of the world, 15th edition*. Dallas: SIL.

Henkin, Leon. 1961. Some remarks on infinitely long formulas. In *Proceedings of the 1959 Warsaw Symposium on Foundations of Mathematics: Infinitistic Methods*.

Higginbotham, James. 1985. On semantics. *Linguistic Inquiry* 16:547–593.

Hopper, Paul J., and Elizabeth Closs Traugott. 1993. *Grammaticalization*. NY: Cambridge University Press.

Hornstein, Norbert. 1995. *Logical form: From gb to minimalism*. Oxford: Blackwell.

Jakobson, Roman. 1963. Implications of language universals for linguistics. In *Universals of language: report of a conference held at dobbs ferry, new york, april 13-15, 1961*, ed. Joseph Greenberg. Cambridge, Massachusetts: MIT Press.

Kayne, Richard S. 1994. *The antisymmetry of syntax*. Cambridge, Massachusetts: MIT Press.

Keenan, Edward L. 1976. Remarkable subjects in Malagasy. In *Subject and topic*, ed. Charles N. Li. NY: Academic Press.

Keenan, Edward L. 1981. A boolean approach to semantics. In *Formal methods in the study of language*, ed. J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof. Amsterdam: Mathematisch Centrum.

Keenan, Edward L. 1988. On semantics and the binding theory. In *Explaining language universals*, ed. J. Hawkins, 105–145. Oxford: Blackwell.

Keenan, Edward L. 2001. Logical objects. In *Logic, meaning and computation: Essays in memory of alonzo church*, ed. C.A. Anderson and M. Zeleny, 149–180. Boston: Kluwer.

Keenan, Edward L. 2004. Further excursions in natural logic: The mid-point theorems. In *Explaining language universals*, ed. Fritz Hamm and Stephan Kepser. Oxford: Oxford University Press.

Keenan, Edward L., and Edward P. Stabler. 2003. *Bare grammar: Lectures on linguistic invariants*. Stanford, California: CSLI Publications.

Klein, Felix. 1893. A comparative review of recent researches in geometry: Programme on entering the philosophical faculty and the senate of the university of Erlangen in 1872. *Bulletin of the New York Mathematical Society* 2:215–249. Translation by M.W. Haskell of the original October 1872 publication, with a prefatory note by the author.

Kobele, Gregory M. 2006. Generating copies: An investigation into structural identity in language and grammar. Doctoral Dissertation, UCLA.

Lee, Thomas Hun Tak. 1986. Acquisition of quantificational scope in Mandarin Chinese. In *Papers and Reports on Child Language Development, No. 25*. Standford University.

Lewis, David K. 1970. General semantics. *Synthese* 22:18–67. Reprinted in D. Davidson and G. Harman, eds., *Semantics of Natural Language*. Dordrecht: Reidel, 1972.

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In *Approaches to natural language*, ed. J. Hintikka, J.M.E. Moravcsik, and P. Suppes. Dordrecht: Reidel. Reprinted in R.H. Thomason, editor, *Formal Philosophy: Selected Papers of Richard Montague.* New Haven: Yale University Press, §8.

Radford, Andrew. 1997. *Syntactic theory and the structure of English: A minimalist approach*. Cambridge Textbooks in Linguistics. Cambridge: Cambridge University Press.

Rotman, Joseph. 1999. *An introduction to the theory of groups*. NY: Springer.

Schachter, Paul. 1984. Studies in the structure of Toba Batak. Technical Report UCLA Occasional Papers in Linguistics, Number 5, UCLA, Los Angeles.

Sternberg, S. 1994. *Group theory and physics*. NY: Cambridge University Press.

Stewart, Ian, and Martin Golubitsky. 1999. *Symmetry*. Oxford: Blackwell.

Strocchi, Franco. 2005. *Symmetry breaking*. Lecture Notes in Physics 732. NY: Springer.

Tarski, Alfred. 1986. What are logical notions? *History and Philosophy of Logic* 7:143–154.

van Fraassen, Bas C. 1989. *Laws and symmetry*. Oxford: Clarendon.

Weyl, Hermann. 1952. *Symmetry*. Princeton, New Jersey: Princeton University Press.

Wigner, Eugene P. 1979. *Symmetries and reflections: Scientific essays*. Woodbridge, Connecticut: Oxbow Press.

Zee, A. 1986. *Fearful symmetry*. NY: Macmillan.

Affiliation

Edward L. Keenan
Department of Linguistics
University of California, Los Angeles
keenan@humnet.ucla.edu

Edward P. Stabler
Department of Linguistics
University of California, Los Angeles
stabler@ucla.edu

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

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 $x \in \langle V^*, Cat\rangle \in L_G$, $\text{str}(x) = V^*$ and $\text{cat}(x) = Cat$.

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:

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

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:

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

[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:

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

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:

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

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 $1 \leq i \leq j$:

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

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

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 $\sigma = s_1 \dots s_{i-1} \sigma' s_{i+1} \dots s_j \in \text{domain}(k \circ_i g)$ and $k \circ_i g(s_1 \dots s_{i-1} \sigma' s_{i+1} \dots s_j) = f(\sigma)$.

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* domain(f)*, is in* 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 $\sigma \in$ 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 $\sigma \in$ domain(f) for f a selection function in G and for any $h \in$ Aut$_G$, because h preserves $Rule_G$, it must be the case that $h(u), h(v) \in \sigma$ for some $\sigma \in$ 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 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 $\sigma \in \text{domain}(f)$, $u, v \in \sigma$, $\text{str}(f(\sigma)) = t_1 t_2 t_3 t_4 t_5$ and $\text{str}(u) = t_2$, $\text{str}(v) = t_4$.

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:

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

PRE *Relations:*
$\langle John, \text{DP}_{\text{nom}} \rangle$ PRE $\langle ate, \text{P2} \rangle$, $\langle John, \text{DP}_{\text{nom}} \rangle$ PRE $\langle cake, \text{DP}_{\text{acc}} \rangle$ (by f)
$\langle ate, \text{P2} \rangle$ PRE $\langle cake, \text{DP}_{\text{acc}} \rangle$ (by f, g)
$\langle John, \text{DP}_{\text{nom}} \rangle$ PRE $\langle ate\ cake, \text{P1} \rangle$ (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 $\langle John, \text{DP}_{\text{nom}} \rangle$ and either $\langle ate, \text{P2} \rangle$ or $\langle cake, \text{DP}_{\text{acc}} \rangle$. The linear order that does arise between $\langle John, \text{DP}_{\text{nom}} \rangle$ and these two expressions, then, only does so due to the local precedence relation between $\langle John, \text{DP}_{\text{nom}} \rangle$ and $\langle ate\ cake, \text{P1} \rangle$.

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, $h(\langle a,A\rangle)$ 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:

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

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 $\mathrm{str}(f(\sigma)) = t_1 t_2 t_3 t_4 t_5$ 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 $\mathrm{str}(f(\sigma)) = t_1 t_2 t_3 t_4 t_5$ 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* 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:

$$f(\langle le\ chien, \text{DP}\rangle, \langle chasse, \text{V}\rangle, \langle le\ chat, \text{DP}\rangle) \longmapsto \langle le\ chien\ chasse\ le\ chat, \text{S}\rangle$$

By Condition (ii) of Definition 10, $(\langle le\ chat, \text{DP}\rangle, \langle chasse, V\rangle, \langle le\ chien, \text{DP}\rangle \in \text{domain}(f)$. By Hypothesis 2, $f(\langle le\ chat, \text{DP}\rangle \langle chasse, \text{V}\rangle, \langle le\ chien, \text{DP}\rangle) \in \text{domain}(f)$ is mapped to $\langle le\ chat\ chasse\ le\ chien, \text{S}\rangle$. It follows that $f(\langle le\ chien, \text{DP}\rangle, \langle chasse, \text{V}\rangle \langle le\ chat, \text{DP}\rangle)$ and $f(\langle le\ chat, \text{DP}\rangle, \langle chasse, \text{V}\rangle, \langle le\ chien, \text{DP}\rangle)$ 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 $u, v \in \sigma \in \text{domain}(f)$ 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	

Coord

Domain				Value	Conditions
and	*s*	*t*	$\longmapsto$	$both^\frown s^\frown and^\frown t$	$\mathrm{C} \in Cat - \{\mathrm{V}_{af}, V_{pf}, \mathrm{CONJ}, \mathrm{P2}\}$
CONJ	C	C		C	
or	*s*	*t*	$\longmapsto$	$either^\frown s^\frown or^\frown t$	$\mathrm{C} \in Cat - \{\mathrm{V}_{af}, V_{pf}, CONJ, \mathrm{P2}\}$
CONJ	C	C		C	
and	*s*	*t*	$\longmapsto$	$both^\frown s^\frown and^\frown t$	$\mathrm{C} \neq \mathrm{C}' \in \{\mathrm{NP}, \mathrm{NP}_{refl}\}$
CONJ	C	C′		NP_{refl}	
or	*s*	*t*	$\longmapsto$	$either^\frown s^\frown or^\frown t$	$\mathrm{C} \neq \mathrm{C}' \in \{\mathrm{NP}, \mathrm{NP}_{refl}\}$
CONJ	C	C′		NP_{refl}	

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: $\langle mang\text{-}, \mathrm{V}_{af} \rangle$ and $\langle di\text{-}, \mathrm{V}_{pf} \rangle$. 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,

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

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

As is clear in the definition, $\text{Merge}_{\text{set}}$ represents label-free set formation, thus generating the bare phrase structure that Chomsky (1995) assumes to be the most minimal assumption, whereas $\text{Merge}_{\text{label}}$ 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 $\text{Merge}_{\text{set}}$ and $\text{Merge}_{\text{label}}$, 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 $\text{Merge}_{\text{set}}$, then, the output for any permutation will be identical as the set is the only output generated. With regard to $\text{Merge}_{\text{label}}$, 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 $\text{Merge}_{\text{label}}$, 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 $\text{Merge}_{\text{label}}$, 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.

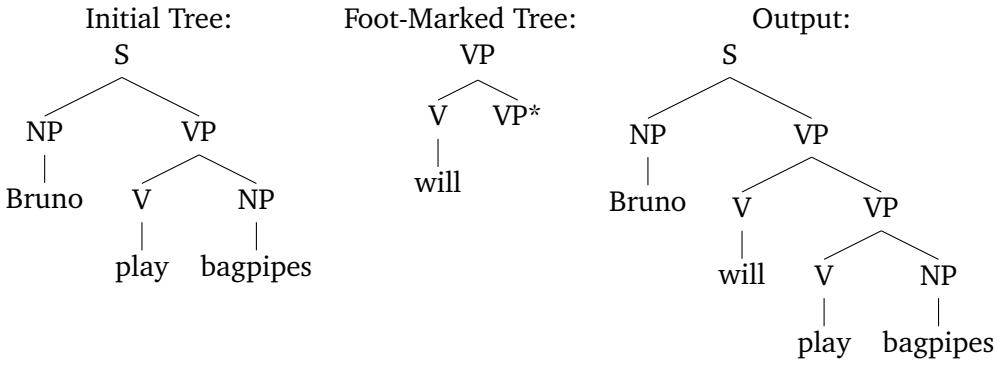


Figure 1: TAG adjunction

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

UCLA Working Papers in Linguistics
Volume 15, Article 3: 1–18, 2010

Little Tagalog

Free Word Order in Bare Grammar

Meaghan Fowlie

Bare Grammars are a simple and straight-forward model for syntax. Normally, the lexicon in such a grammar is kept very minimal: each lexical item is an ordered pair consisting of the string and its category. However, the simplicity of such a lexicon can make formulating rules that are succinct and that capture grammatical regularities impossible. In this paper I propose that adding more information to the lexicon can make it possible to write rules that are succinct and capture more generalisations. The second part of this paper makes use of this additional information in the lexicon to formulate a grammar for free word order. I propose that it can be accounted for in bare grammars if we allow rules to form not only concatenated strings (which by definition have order) but also sets (which by definition do not).

Keywords free word order, Tagalog, Bare Grammar, sets, multisets

Introduction

This work attempts two feats:

1. To "reify" bare grammar subscripts and other notations that mnemonically relate categories
2. To account for genuinely free word order without appealing to multiple derivations

Bare grammars (Keenan and Stabler 2003) typically define the Lexicon as a set of ordered pairs (string, cat), where *string* is the phonological form of the word and *cat* it the category of the word.

Example. (banana, N)

Sometimes it is convenient to give two separate categories similar names to help the linguist remember that the categories have something important in common. For example, a language like Spanish, which distinguishes masculine and feminine nouns, might have two categories named Nm and Nf, for Noun (masculine) and Noun (feminine). However, these categories will be treated by the derivation as being as different as any other two categories. This means that if we want a rule to apply to both masculine and feminine nouns, we will have to write two separate rules.

Often these separate rules are written as one rule, for readability. In these cases the rules are written as if it is possible, for example, to let a variable range over {m, f}. However, this is really just notational shorthand for two rules.

Merge (notational shorthand): $((s, \mathrm{Ax}), (t, \mathrm{Nx})) \longmapsto (s^\frown t, \mathrm{Nx}),\ x \in \{\mathrm{m}, \mathrm{f}\}$

Merge (full version): $\begin{cases} (\langle s, \mathrm{Am}\rangle, \langle t, \mathrm{Nm}\rangle) \longmapsto \langle s^\frown t, \mathrm{Nm}\rangle \\ (\langle s, \mathrm{Af}\rangle, \langle t, \mathrm{Nf}\rangle) \longmapsto \langle s^\frown t, \mathrm{Nf}\rangle \end{cases}$

This paper proposes that these "fake" variable, like x above, be made real, so that rules that refer to just the N part of Nf and Nm can be written.

The second part of this paper proposes a syntax for free word order phenomena within a Bare Grammar framework. I will apply the first proposal, that subscripts be reified, to this problem. The grammar will consist of a finite vocabulary V, a set *Rule* of functions for combining elements of the vocabulary, a set *Cat* of categories linked to the vocabulary items, and a set θ of theta roles, also linked to the vocabulary items.

We will use a toy grammar modelled on aspects of Tagalog as an example of free word order. Vocabulary items will be linked not only to a category as is done in Keenan and Stabler (2003), but also with a third element that encodes further information about the word, regarding the theta role it plays or requires in the sentence. It is this last set, θ, that makes use of the first proposal. The inclusion of θ is a way of keeping track of the number arguments that will later be linked to the verb.

1 A New Element for the Grammar

1.1 A Problem: Subtypes are not Subtypes

In X-bar and related theories, lexical categories, such as N, are related to the phrases that contain them, such as NP, by projection. When two elements merge, one of the two projects, meaning among other things that the enclosing phrase behaves much like the projecting category in terms of selection. A verb looking for a noun can in fact look for an NP.

However, in bare grammars, projecting categories are not usually related to the lexical categories except through the rules that make phrases from lexical categories. With only two dimensions on which to describe lexical items – string and category – lexical and projecting categories are only related if they are the same; that is, if adding an adjective or article to a noun leaves the category as N, rather than something new, like an N' or an NP.

When, say, a verb looks for a noun phrase, the fact that the noun phrase was built from a noun is obscured. We also have no way to relate nouns that differ in morphological marking, such as case or gender. Once the rule that morphologically marks the nouns has applied, the fact that they are indeed nouns is normally hidden. This means that a verb that can take either a masculine or feminine noun needs two separate rules, forcing conceptually redundant rules and again obscuring the relationship between the gender-marked nouns.

For example, in Little Spanish, given as a model of gender marking in Keenan and Stabler (2003), *Cat* includes Nf (Noun-feminine) and Nm (Noun-masculine), as well as A, Af, and Am (Adjective, Adjective-feminine, Adjective-masculine). Adjective Modification (**AM**) is written with a notational shorthand which refers to the m and f, but m and f are not really part of the grammar. The A in Am and Af has no real meaning; it only serves to remind us, the linguists, which familiar categories we are dealing with.

AM (notational shorthand): $((s,\mathrm{Ax}),(t,\mathrm{Nx})) \longmapsto (s^\frown t,\mathrm{Nx}),\ x \in \{\mathrm{m},\mathrm{f}\}$

AM (full version): $\begin{cases} (\langle s,\mathrm{Am}\rangle,\langle t,\mathrm{Nm}\rangle) \longmapsto \langle s^\frown t,\mathrm{Nm}\rangle \\ (\langle s,\mathrm{Af}\rangle,\langle t,\mathrm{Nf}\rangle) \longmapsto \langle s^\frown t,\mathrm{Nf}\rangle \end{cases}$

If we remove the mnemonic names, and let $\mathrm{Am}=\alpha, \mathrm{Af}=\beta, \mathrm{Nm}=\gamma, \mathrm{Nf}=\kappa$, Merge looks like this:

AM: $\begin{cases} (\langle s,\alpha\rangle,\langle t,\gamma\rangle) \longmapsto \langle s^\frown t,\gamma\rangle \\ (\langle s,\beta\rangle,\langle t,\kappa\rangle) \longmapsto \langle s^\frown t,\kappa\rangle \end{cases}$

Now these four categories are clearly related only by what rules they undergo. While this is, in a sense, exactly what categories are, it does not capture the intuition that the same kinds of things happen to both categories when they undergo the same rule. In these rules, in both cases the first argument's string precedes the second argument's string, and the category of the image is the category of the second argument. The semantics will also be the same: the meaning of both will be whatever the meaning of a modified noun is.

Moreover, despite the intuition that masculine and feminine nouns are both nouns, the lexicon does not relate them at all. The idea is that there are relationships between categories, both intuitive and formal, that are not captured by a lexicon that is a subset of a cross product of only two sets.

Additionally, it is worthwhile to consider what kinds of rules are possible under this system, as well as what kinds of rules might be built in parallel or on analogy with existing rules. As we have it, there are almost no restriction on the possible rules, but one could examine the patterns of existing rules and create restrictions. Simpler here would be to create a rule on analogy with an existing rule, under the supposition that if it is similar enough it will fall under most restrictions one might want to place on *Rule*.

For example, Little Spanish's adjective modification is meant to capture a relationship between two categories, adjectives and nouns, restricted by agreement requirements. However, since there is no such category as "noun" and the adjectives that combine with Nfs and Nms are also no longer adjectives but Afs and Ams, in fact adjective modification combines two unrelated pairs or categories. Why not, then, add a third unrelated set of categories, as would be necessary if there were a third gender? Why not, say, V and A? Such a move would certainly be in keeping with the style of rule, but it would not capture a related modification.

This example also illustrates how unsuccinct this style of rule can be: what if the language has a great number of noun classes, all of which behave the same under adjective modification, i.e. the noun and ajdective must match in gender? There will have to be a great number of subcases of the adjective modification rule.

In order to better capture intuitions about category, and to allow for more succinct rules, I propose that such categories be related in the lexicon. Let us consider what happens if we allow *Lex* to include ordered *triples* as well as ordered pairs. The third element essentially represents useful subscripts, such as case (a, n), gender (f, m), or predicate arity (0, 1, 2).

In the example of gender, not only can the rule now have only one case, but there is a very simple way to describe the gender-agreement: the third element of the noun must be identical with the third element of the adjective.

AM (modified): $(\langle s, \mathrm{A}, x\rangle, \langle t, \mathrm{N}, x\rangle) \longmapsto \langle s^\frown t, \mathrm{N}, x\rangle, \forall x \in \{f, m\}$

Notice this is almost exactly the notational shorthand for the rule given in Keenan and Stabler (2003).

Suppose, by way of illustration, we let this third element simply be a number representing case or predicate arity. Then we can match the verbs with the right number of arguments with the right case-marking. Consider the following simplified model.

1.1.1 Verbs

Verbs come inherently intransitive, transitive, or ditranstitive. Rather than calling these P1, P2 and P3, I propose separating the number from the P, allowing math to be done on it, for example $\langle sneezed, \mathrm{P}, 1\rangle$, $\langle admired, \mathrm{P}, 2\rangle$, and $\langle gave, \mathrm{P}, 3\rangle$.

Now the arity of the predicate – the number of arguments required by the predicate – can be accessed formally. We will see below, for example, that the arity of the verb will be reduced as arguments are merged, and the case of the arguments will be selected accurately because the case number will match the gradually lowering arity number, which is exactly the intuition to be captured.

1.1.2 Case

Nouns are generally listed in Lex_0 as ordered pairs, and when case-marked also take a number as a third element. Suppose case-markers in *Lex* look like this: $\langle$-NOM, K, 1$\rangle$, $\langle$-ACC, K, 2$\rangle$ and Case Marking is a rule that takes a K and an N and yields a case-marked noun.[1]

CM: $(\langle s, \mathrm{N}\rangle, \langle t, \mathrm{K}, n\rangle) \longmapsto \langle s^\frown t, \mathrm{N}, n\rangle$

Example. $\langle man, \mathrm{N}\rangle, \langle$-NOM, K, 1$\rangle \longmapsto \langle man$-NOM, N, 1$\rangle$

With the case type *1* separated from the categories K and N, each part of the image under CM is taken directly from a part of the preimage. The string is concatenated

[1] This is not the only possible way to relate un-case-marked and case-marked nouns, of course, but this works, and does have a certain logic: both are nouns in that they have a second element *N*, but they differ in that case-marked nouns carry more information, instantiated in their having a third element.

from the strings of the input, the category is one of the categories, and the case number is the third element of one of the pairs in the input. Nothing new need be introduced in the output: everything was already present in the input. This means that the grammar is manipulating only what it is given. Nothing is being pulled out of the aether.

CM: $(V^* \times Cat) \times (V^* \times Cat \times \mathbb{N}) \to (V^* \times Cat \times \mathbb{N})$,
$\text{CM}(A,B) = (\langle [A]_1, [B]_1 \rangle, [A]_2, [B]_3)$

That is, the image of the pair (A,B) under CM is the ordered triple consisting of 1) the sequence of the first elements of A and B, 2) the second element of A, and 3) the third element of B.

This is, if nothing else, more intuitive than a grammar in which new things are introduced in the output. Although the definition of a function is not constrained at all by such considerations, such functions are arguably more elegant and more intuitive. A possible restriction on *Rule* presents itself. Just as the rules of Little Spanish never have a string in the output that was not present in the input, perhaps bare grammars should not have rules with categories in the output that were not present in the input. Such a restriction would not be possible without separating case or gender from noun.

By way of comparison, suppose we defined an operation CM2 without the ordered triples. Let further $\text{N} = \alpha$, $N(\textsc{nom}) = \beta$, $N(\textsc{acc}) = \gamma$, Case marker($\textsc{nom}$) $= \kappa$, Case marker($\textsc{acc}$) $= \delta$ to make the separateness of the categories clear.

CM2: $\begin{cases} (\langle s, \alpha \rangle, \langle t, \kappa \rangle) \longmapsto \langle s^\frown t, \beta \rangle \\ (\langle s, \alpha \rangle, \langle t, \delta \rangle) \longmapsto \langle s^\frown t, \gamma \rangle \end{cases}$

Not only are there now two quite different lines to our definition of CM, but new elements are introduced in the image that were not present in any part of the preimage. Only the first element, the sequence of the first elements of the input, can be copied from the input.

CM2: $(V^* \times Cat) \times (V^* \times Cat) \to (V^* \times Cat)$,
$$\text{CM2}(A,B) = \begin{cases} \langle [A]_1, [B]_1, \beta \rangle \text{ if } [B]_2 = \kappa \\ \langle [A]_1, [B]_1, \gamma \rangle \text{ if } [B]_2 = \delta \end{cases}$$

1.1.3 Verbs and Arguments

Consider a Little English with such elements as $\langle admired, \text{P}, 2 \rangle$, $\langle him, \text{N}, 2 \rangle$, and $\langle she, \text{N}, 1 \rangle$. Then we can write a Predicate-Argument function (**PA**) like this:

PA: $(\langle x, \text{P}, n \rangle, \langle y, \text{N}, n \rangle) \longmapsto \langle x^\frown y, \text{P}, n-1 \rangle \qquad n \in \mathbb{N}$

Notice that the third elements of the domain pair must match, and the image's third element, while not copied from the preimage, is calculated from the third elements in the preimage using simple natural number arithmetic. The match between the predicate arity and the case marker is the reason I defined both in numbers. Each argument reduces the arity of the predicate until it is a P0, a sentence. We want the

object to have the same type as a transitive verb because it will combine with the verb to form a P1, which is looking only for a nominative. A nominative is of type *1*.

Now we can see in the simple case that a P1 takes as argument a nominative-marked noun phrase. We have defined NP-NOM as an ordered triple $\langle s, \mathrm{N}, 1\rangle$.

Example. $(\langle \textit{sneezed}, \mathrm{P}, 1\rangle, \langle \textit{he}, \mathrm{N}, 1\rangle) \longmapsto \langle \textit{he sneezed}, \mathrm{P}, 0\rangle$

Similarly, *She admired him* can be built up using this one formulation of the PA rule, modulo word order.[2]

(1) a. $(\langle \textit{admired}, \mathrm{P}, 2\rangle, \langle \textit{him}, \mathrm{N}, 2\rangle) \longmapsto \langle \textit{admired him}, \mathrm{P}, 1\rangle$
b. $(\langle \textit{admired him}, \mathrm{P}, 1\rangle, \langle \textit{she}, \mathrm{N}, 1\rangle \longmapsto \langle \textit{She admired him}, \mathrm{P}, 0\rangle$

Adding a third element to the Lexicon makes *Rule* much more succinctly and mathematically definable, and captures intuitions not otherwise captured.

2 Tagalog

Tagalog has free word order in a number of constituents. I will be using multisets in the free word order proposal. Following is a brief introduction thereto.

2.1 Multisets

Intuitively, a multiset is a set in which elements can be repeated. In a regular set, $\{1,2,3\} = \{1,1,1,1,1,2,2,3\}$. If these were multisets, they would not be the same set.

Since sets are defined by their elements, technically a multiset cannot be thought of as a set per se, although we will treat them as though they are for simplicity of notation. Instead, a multiset is perhaps best thought of as a function.

Definition 1 (Multiset)**.** A *multiset of a set* S is a map $m\colon S \to \mathbb{N}$ from S to $\mathbb{N} = \{0,1,2,\ldots\}$

The intuition is that x is "in" $m(x)$ iff $m(x) > 0$. Moreover, the number of x's in the multiset is the value of m at x.

I will notate multisets just as I would real sets, with the understanding that if an element is repeated in the list notation, the number of times it appears is its image under m. In other words, I will notate multisets intuitively.

Here are some definitions of set theoretic notions for multisets. For any multisets M, P of S:

Membership: $s \in M$ iff $M(s) > 0$

Equality: $M = P$ iff $\forall s \in S, M(s) = P(s)$

Union: $(M \cup P)\colon S \to \mathbb{N}$ s.t. $(M \cup P)(s) = M(s) + P(s)$

[2] The word order problem I don't care about right now, as I'm working on Tagalog! But certainly it is an issue.

Intersection: $(M \cap P)\colon S \to \mathbb{N}$ s.t. $(M \cap P)(s) = M(s) \wedge P(s)$

Difference: $(M - P)\colon S \to \mathbb{N}$ s.t. $(M - P)(s) = M(s) - (M(s) \wedge P(s))$

Subset: $M \subseteq P$ iff $\forall s \in S, M(s) \leq P(s)$

Let us now turn to Tagalog.

2.2 Tagalog Data

Tagalog shows free word order in most of the sentence, and does not seem to have any preferred order or change in meaning. Generally, V comes first, followed by everything else, in any order (data from Kroeger 1993).

(2) *Nagbigay ng-libro sa-babae ang-lalaki*
gave GEN-book DAT-woman NOM-man
'The man gave the woman a book'
Nagbigay ng-libro ang-lalaki sa-babae
Nagbigay sa-babae ng-libro ang-lalaki
Nagbigay sa-babae ang-lalaki ng-libro
Nagbigay ang-lalaki sa-babae ng-libro
Nagbigay ang-lalaki ng-libro sa-babae

When an adjective is added, the case marker is consistently in first position within the DP, but the adjective and noun can be in either order with the same meaning.

(3) a. *ng libro-ng malaki*
GEN book-LK big
'the big book'
b. *ng malaki-ng libro*
GEN big-LK book

When the DP *the big book* is ordered as in (3-a), the three DPs yield the usual six orders as in (2). Similarly, when *the big book* is ordered as in (3-b), these DPs can be arranged in six possible orders, yielding a total of twelve possible word-orders.

2.3 Proposal: Two Combining Operations

I propose that in Tagalog there are two ways to form expressions: concatenation and multiset formation. When word-order matters, elements are concatenated. When it is free, they form multisets. (They must be multisets rather than regular sets given that we don't want two identical strings to be treated as one: *Some rabbit killed some rabbit* ≠ *Some rabbit killed*.)

Because sets, unlike sequences (i.e. concatenation), have no inherent order defined on them, I propose that free word order arises when merge forms sets rather than concatenating. The unordered set elements can appear in any order, but only one derivation will be required for all possible orders.

In order for these multisets to be concatenated with other elements, including other multisets, Tagalog's version of concatenation must be *multiset* concatenation. Since concatenation is just sequence formation, we need only say that the elements of the sequence are multisets. Since both multiset concatenation and multiset formation need to be able to apply to any element of the language, everything must be multiset formation in some sense. Therefore multiset concatenation rules will have as output a *multiset* containing a sequence of multisets. This may not always be made explicit as we go, as the brackets get quite messy, but keep in mind that every element of the language is a multiset.

Ordered: multiset concatenation

Free: multiset formation

I also claim that Tagalog Vocabulary consists of unit multisets of words rather than words. This allows a single formulation of each $f \in Rule$, without having to reformulate it when the arguments are outputs of other rules. We will see later on why this is desireable.

3 Verbs and Arguments in Tagalog

3.1 V-Initial

Before we go on to the actual grammar and examples, we must look at the V-initial nature of Tagalog. Normally, one would probably say that V starts lower in the sentence, and moves up to C or T, leaving behind its arguments to scramble. Since we lack a theory of movment, I reformulated it under the assumption that the verb really does combine late. Therefore rather than the usual cumulative gathering of arguments by the verb, I propose that the arguments gather themselves into a multiset. Call the Rule *Argument Linking* (AL).

Now, the required noun phrases of the verb are determined by a combination of case marking and theta role. There is not a one-to-one relationship between case marking and theta roles in Tagalog, which presents a difficult and interesting problem.

3.2 The ang *NP*

Every sentence of Little Tagalog, and indeed, nearly every sentence of real Tagalog, has one NP which is marked with the case marker *ang*. There is debate about what exactly the *ang*-marked element is (whether it be a topic, "subject", or something else entirely (see Kroeger 1993 for discussion), but whatever it is, it can be any NP, argument or adjunct, agent or object. The verb is voice-marked to indicate the theta-role of the *ang*-marked NP. The remaining NPs will be marked with *sa*, which Kroeger calls genitive, or *ng*, which he calls dative. The case markers are divided up by theta role. Some sentences have NPs marked with the same case.

I propose that the verb starts with a theta grid, which determines the case of the NP(s) it needs. Moving away from numbers, I propose that the theta roles are

grammatical primitives, i.e. $\theta :=$ {actor, theme, goal, recipient, locative, instrumental, benefactive, possessor}[3], which I abbreviate as $\theta := \{a, t, g, r, l, i, b, p\}$. These theta roles will need to be grouped into sets, so let $\mathbb{K} = \wp(\theta)$. Now every verb can be given not only an arity but a theta grid, for example $\langle gave, \mathrm{P}, \{a, t, r\}\rangle$, $\langle cooked, \mathrm{P}, \{a, t\}\rangle$, $\langle sneezed, \mathrm{P}, \{t\}\rangle$.

Voice-marking the verb maps a theta-role to a case-marker, specifically to whatever *ang* is. Following Kroeger I will call it NOM. Any theta role can be nominative, but the other two case markers partition θ. Let $\textsc{gen} = \{a, t, p, i, b\}$, $\textsc{dat} = \{l, g, r\}$, and $\textsc{nom} = \theta$.

Definition 2 (f_θ)**.** Let $f_\theta \colon \wp(\theta) \to \{\textsc{gen}, \textsc{dat}, \emptyset\}$ be a partial function defined as follows:

$$f_\theta(X) = \begin{cases} \emptyset & \text{iff } X = \emptyset \\ Y & \text{iff } X \subseteq Y \text{ and } X \neq \emptyset \end{cases}$$

.

That is, f_θ takes sets of theta roles and maps them to their associated case.

We lift f_θ to g_θ which maps sets of theta-roles to multisets of their associated case.

Definition 3 (g_θ)**.**

$$g_\theta(X) = \{f_\theta(\{x\}) | \{x\} \subseteq X\}$$

Schematically, f_θ and g_θ are as follows:

$$f_\theta(\{\theta_1, ..., \theta_n\}) = Y \iff \{\theta_1, ..., \theta_n\} \subseteq Y$$

$$g_\theta(\{\theta_1, ..., \theta_n\}) = \{f_\theta(\{\theta_1\}), ..., f_\theta(\{\theta_n\})\}$$

Note that f_θ and g_θ are single-valued (and therefore functions) since GEN and DAT partition $\wp(\theta)$. Note also that no set containing $\emptyset$ is in the range of g_θ since $\emptyset = f_\theta(\emptyset)$, and $\emptyset$ is not a singleton set as required by g_θ.

3.3 Voice-Marking

I now add to the grammar category Voi and lexical items voice-marking affixes.[4] The third coordinate is the theta role (or set of theta roles) that will be *ang*-marked.

$$\langle \mathrm{AV}, \mathrm{Voi}, \{a\}\rangle \quad \langle \mathrm{TV}, \mathrm{Voi}, \{t\}\rangle \quad \langle \mathrm{IV}, \mathrm{Voi}, \{i\}\rangle \quad \langle \mathrm{BV}, \mathrm{Voi}, \{b\}\rangle \quad \langle \mathrm{DV}, \mathrm{Voi}, \textsc{dat}\rangle$$

The verb can now be instructed to look for appropriately case-marked nouns to take as arguments. Here is the rule *Voice Mark* (VM), with its subfunction $v_S(T)$ defined below.

[3] Simplification: real Tagalog allows themes to be -DAT or -GEN, interpreting one as definite and the other as indefinite.

[4] In real Tagalog, these are prefixes, suffixes and infixes. To simplify, I am treating them as suffixes here, as they need to stick to the verb.

VM: $(\langle x, \text{P}, T\rangle, \langle y, \text{Voi}, S\rangle) \longmapsto \langle x^\frown y, \text{P}, v_S(T)\rangle$

Since voice-marking already maps a theta role to case, let us define it to map all the verb's theta roles to their appropriate case. For this, we must define a function, v_S for each $S \in \mathbb{K}$.

$$v_S(T) = (g_\theta(T) - \{f_\theta(T \cap S)\}) \cup \{\text{NOM}\}$$

The function v_S maps the set of theta roles of the verb to the multiset of their associated cases, if necessary subtracts the case associated with any argument theta role it is voice-marked for, and adds NOM (*ang*). If the voice-marker marks the verb for an argument of the verb, $T \cap S \neq \emptyset$. Then the difference operator removes that case from the set the verb is looking for and replaces it with NOM.

If the voice marker is for what would normally be an adjunct, $T \cap S = \emptyset$ so it is just added into the set of required NPs for the verb as a NOM.

Looking more closely at v_S, consider the case in which the verb is voice-marked for an *argument*. Recall that T is the theta grid of the verb and S the set of theta roles associated with the voice marker.

1. The set of theta roles T in the theta grid of the verb are mapped to their corresponding cases by g_θ.
2. $T \cap S$ is the set of theta-roles the verb and voice-marker have in common. Normally this is a singleton set, unless the voice marker is dative, in which case it will be some subset of DAT.
3. $f_\theta(T \cap S)$ is the case-marker associated with the voice-marker, since $T \cap S \subseteq S$ and f_θ is defined interms of subsets.
4. The case for the voice marker is subtracted from the set of cases the verb is seeking.
5. The subtracted case is replaced by NOM.

Now the verb is looking for the same number of arguments, but it knows which cases it needs, including one NOM, which replaced one of the original cases the un-voice-marked verb was seeking.

Now consider the case when the verb is voice-marked for a *non-argument* of the verb. $T \cap S = \emptyset$ so v_S just maps the theta grid to the corresponding cases, and adds NOM.

1. The set of theta roles in the theta grid of the verb are mapped to their corresponding cases by g_θ.
2. $\{f_\theta(\emptyset)\} = \{\emptyset\}$. $\emptyset$ is never a member of $g_\theta(T)$ so the difference operation changes nothing.
3. NOM is added.

Result of v_S: a multiset of the cases of the nouns the verb will look for.

Example. Below we see an instrumental-marked verb. The instrumental theta role is not part of the theta grid of the verb. The usual theta roles are retained, but the verb is now looking for four, not three, NPs, as one has been added by the voice marker.

$$\text{VM}(\langle \textit{gave}, \text{P}, \{a, t, r\}\rangle, \langle \text{IV}, \text{Voi}, \{i\}\rangle) = \langle \textit{gave-IV}, \text{P}, \{\text{GEN}, \text{NOM}, \text{GEN}, \text{DAT}\}\rangle$$

Here $v_{\{i\}}(\{a,t,r\})$ is calculated as follows. We have $\{i\} \cap \{a,t,r\} = \emptyset$, so

$$\begin{aligned} v_{\{i\}}(\{a,t,r\}) &= (g_\theta(\{a,t,r\}) - f_\theta(\emptyset)) \cup \{\textsc{nom}\} \\ &= g_\theta(\{a,t,r\} \cup \{\textsc{nom}\} \\ &= \{\textsc{gen}, \textsc{gen}, \textsc{dat}\} \cup \{\textsc{nom}\} \\ &= \{\textsc{gen}, \textsc{gen}, \textsc{dat}, \textsc{nom}\} \end{aligned}$$

Example. The verb below is dative-marked. The theta role which would normally be dative is now nominative. This is acheived by subtracting DAT from the set of cases sought and replacing it with NOM.

$$\text{VM}(\langle \textit{gave}, \text{P}, \{a,t,r\} \rangle, \langle \text{DV}, \text{Voi}, \textsc{dat} \rangle) = \langle \textit{gave-DV}, \text{P}, \{\textsc{gen}, \textsc{nom}, \textsc{gen}\} \rangle$$

Recall that $\textsc{dat} = \{l, g, r\}$.

$$\begin{aligned} v_{\textsc{dat}}(\{a,t,r\}) &= (g_\theta(\{a,t,r\}) - f_\theta(\{l,g,r\} \cap \{a,t,r\})) \cup \{\textsc{nom}\} \\ &= (g_\theta(\{a,t,r\}) - f_\theta(\{r\})) \cup \{\textsc{nom}\} \\ &= (\{\textsc{gen}, \textsc{gen}, \textsc{dat}\} - \{\textsc{dat}\}) \cup \{\textsc{nom}\} \\ &= (\{\textsc{gen}, \textsc{gen}\}) \cup \{\textsc{nom}\} \\ &= \{\textsc{gen}, \textsc{gen}, \textsc{nom}\} \end{aligned}$$

Note that the difference between a voice-marked verb and an un-voice-marked verb is the third coordinate. Before a verb is voice-marked, its third coordinate is an element of $\mathbb{K}$. A voice-marked verb's third coordinate is an element of $\wp(\mathbb{K})$. The verb is now set up to seek appropriately case-marked NPs. Let us now turn to the nouns.

3.4 Nouns

Nouns are listed in the lexicon as simple ordered pairs $\langle s, N \rangle$. Case markers are ordered triples which have as their third coordinate a subset of $\mathbb{K}$. We add three items to the lexicon: $\langle \textit{ang}, \text{K}, \textsc{nom} \rangle$, $\langle \textit{sa}, \text{K}, \textsc{gen} \rangle$, $\langle \textit{ng}, \text{K}, \textsc{dat} \rangle$. And we add CM as defined in Sec. 1.1.2 to *Rule*.

As mentioned in section 2.3 above, in order for the free-order operation (multiset formation) and our ordered operation (multiset concatenation) to work together, everything must be multisets. Lexical items must in fact be singleton multisets. Multi-set concatenation must form multisets consisting of sequences of multisets. This is of particular importance in the next rule, *Argument Linking*.

Since Little Tagalog is verb-initial, the set of nouns must be linked together before the verb selects them. This means that the verb will select the whole group of nouns together, rather than picking them up one at a time. Most of the constraints on this process will actually be on the result of the rule PA that links the nouns to the verb. Nouns will be allowed to combine quite freely, but only certain sets of them will be able to be taken as an argument set by the verb. The only restriction here is that there may not be more than one nominative (*ang*-marked) NP.

AL: $(\langle x, \text{N}, k \rangle, \langle y, \text{N}, l \rangle) \longmapsto \langle x \cup y, \text{N}, k \cup l \rangle$ if $(k \cup l)(\textsc{nom}) \leq 1$

AL creates a multiset of nouns, accompanied by a multiset of their case-markers.

3.5 *Putting it all Together: Predicate-Argument Operation*

Finally, the voice-marked V must select its arguments.

PA: $(\langle x, \mathrm{P}, K\rangle, \langle y, \mathrm{N}, L\rangle) \longmapsto \langle x^\frown y, \mathrm{P}, K - L\rangle$ for $K, L \in \wp(\mathbb{K})$

The idea is that if L contains all the cases the verb is looking for, the result is a P0. Otherwise it is not a fully saturated verb, and cannot be a licit sentence. Any non-nominative adjuncts in L are ignored, since the difference operator does nothing with elements of L not also in K.

Example. Figures 3–6 depict how the sentence meaning *The man gave the woman the book* with the agent *man* as the *ang*-phrase – i.e. example (2) on page 7 – is built from the argument set {{ng} ⌢{woman}, {sa} ⌢{book}, {ang} ⌢{man}}, N, {NOM, GEN, DAT}⟩.

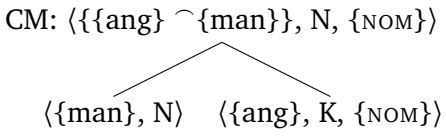


Figure 1: Case-marking *man*

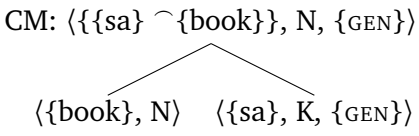


Figure 2: Case-marking *book*

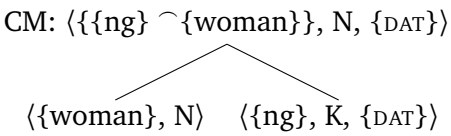


Figure 3: Case-marking *woman*

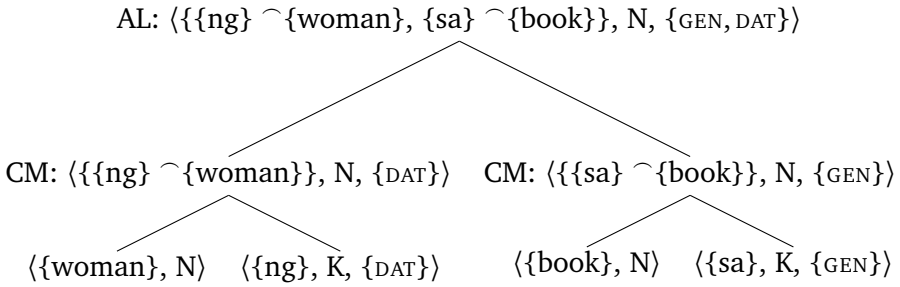


Figure 4: Linking the arguments *woman, book*

(Continued in figures 5 and 6.)

3.6 *Adjectives*

Recall that Tagalog can freely order adjectives with nouns, but adjectives and nouns must stay together. Please ignore the linker *-ng*, which will be excluded from Little Tagalog (data from Raphael Marcado, p.c. 2007).

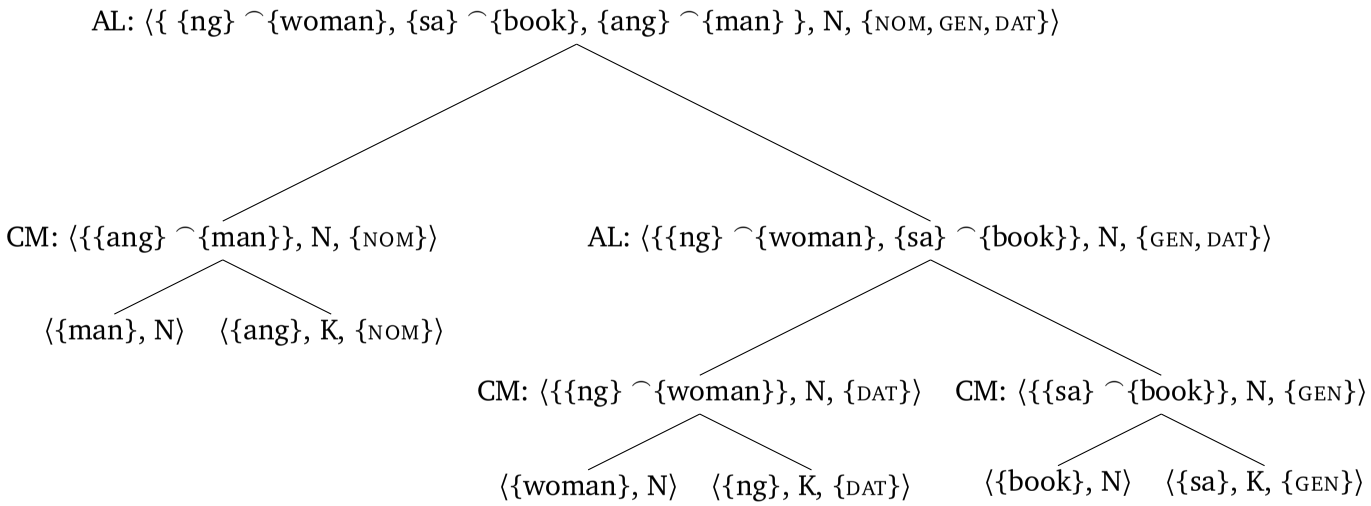


Figure 5: *man, woman, book* argument-linked

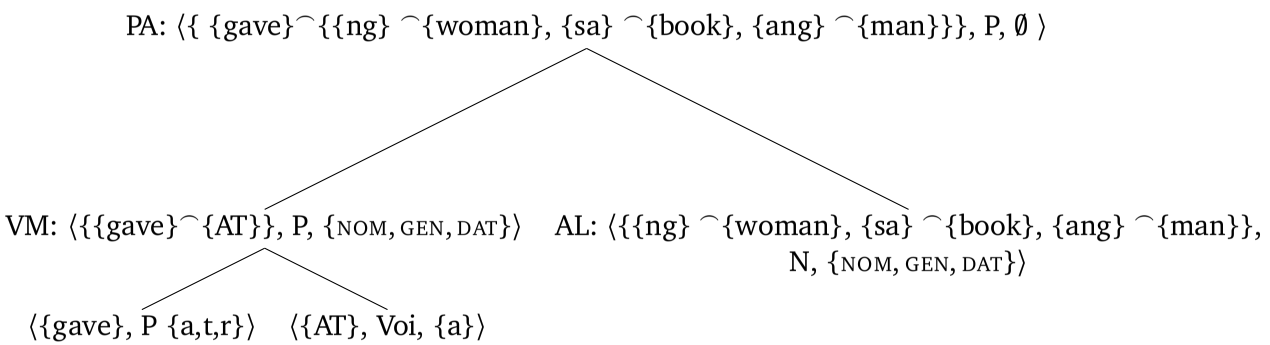


Figure 6: Adding the verb to build the full sentence *the man gave the woman the book*

(4) a. *ng libro-ng malaki*
GEN book-LK big
'the big book'
b. *ng malaki-ng libro*
GEN big-LK book

I propose a rule of Adjective Modification, thus:

AM: $(\langle x, \mathrm{N}\rangle, \langle y, \mathrm{A}\rangle) \longmapsto \langle x \cup y, \mathrm{N}\rangle$

The words are combined with multiset formation, so they are freely ordered. Notice that the category is the same as for the noun, so it can still be combined with the case marker. Because CM combines using multiset concatenation, we get the case-marker followed by the noun and adjective, in either order. An example is given in Fig. 7.

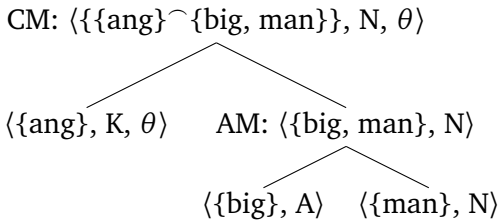


Figure 7: Assembling *ang big man*

3.7 Coordination

$$\textbf{COORD:} \begin{cases} (\langle and, \mathrm{Cj}\rangle, \langle x, \mathrm{C}\rangle, \langle y, \mathrm{C}\rangle) \longmapsto \langle x^\frown and^\frown y, \mathrm{C}\rangle \\ (\langle and, \mathrm{Cj}\rangle, \langle x, \mathrm{C}, X\rangle, \langle y, \mathrm{C}, X\rangle) \longmapsto \langle x^\frown and^\frown y, \mathrm{C}, X\rangle \end{cases}$$
$$\forall C \in Cat, X \in \mathbb{K} \cup \wp(\mathbb{K})$$

Let us see how COORD interacts with AM and CM. Suppose we want to coordinate nouns. COORD predicts that we can do so before or after CM.

(5) a. COORD
$(\langle and, \mathrm{Cj}\rangle, \langle man, \mathrm{N}\rangle, \langle woman, \mathrm{N}\rangle) \longmapsto \langle \{man^\frown and^\frown woman\}, \mathrm{N}\rangle$
CM
$(\langle \{man^\frown and^\frown woman\}, \mathrm{N}\rangle, \langle \{ang\}, \mathrm{K}, \theta\rangle) \longmapsto$
$\langle \{ang\}^\frown \{man^\frown and^\frown woman\}, \mathrm{N}, \theta\rangle$

b. CM
$(\langle \{man\}, \mathrm{N}\rangle, \langle \{ang\}, \mathrm{K}, \theta\rangle) \longmapsto \langle \{\{ang^\frown man\}\}\rangle$
$(\langle \{woman\}, \mathrm{N}\rangle, \langle \{ang\}, \mathrm{K}, \theta\rangle) \longmapsto \langle \{\{ang^\frown woman\}\}\rangle$
COORD
$(\langle and, \mathrm{Cj}\rangle, \langle \{\{ang\}^\frown \{man\}\}\rangle, \langle \{\{ang\}^\frown \{woman\}\}\rangle)$
$\longmapsto \langle \{\{ang\}^\frown \{man\}\}^\frown \{and\}^\frown \{\{ang\}^\frown \{woman\}\}, \mathrm{N}, \theta\rangle$

Tagalog does in fact allow coordination of both case-marked and un-case-marked nouns, so Little Tagalog generates correctly here. More interesting is that Little Tagalog generates correctly for modified nouns.

(6) COORD
$(\langle and, \text{Cj}\rangle, \langle man, \text{N}\rangle, \langle woman, \text{N}\rangle) \longmapsto \langle \{man^\frown and^\frown woman\}, \text{N}\rangle$
AM
$(\langle \{man^\frown and^\frown woman\}, \text{N}\rangle, \langle \{big\}, \text{A}\rangle) \longmapsto \langle \{big, man^\frown and^\frown woman\}, \text{N}\rangle$
CM
$(\langle \{big, man^\frown and^\frown woman\}, \text{N}\rangle, \langle \{ang\}, \text{K}, \theta\rangle$
$\longmapsto \langle \{\{ang\}^\frown \{big, man^\frown and^\frown woman\}\}, \text{N}, \theta\rangle$

The word orders here are as follows (data from Nerissa Black, p.c. 2009).

(7) a. ang big man and woman
b. ang man and woman big

The string in example (7-a) can also mean that the man, but not the woman, is big. (7-b) can also mean that the woman but not the man is big. This would be generated by modifying only one of the nouns instead of the coordination of the nouns:

(8) a. [ang [[big man] and woman]]
b. [ang [man and [woman big]]]

Both can mean that both the man and the woman are big, though to be unambiguous one could modify both with the adjective. Clearly, our grammar generates these too.

(9) a. [ang [[big man] and [big woman]]]
b. [ang [[man big] and [woman big]]]

I don't yet know whether *ang big man and woman big* and *ang man big and big woman* are grammatical in real Tagalog. I predict them to be.

To see the ambiguity of (7-a), consider the trees in Fig. 8 and 9. The tree in Fig. 8 is the tree for the derivation decribed above. The tree in Fig. 9 is a different derivation, wherein only *man* is modified by the adjective. The word-orders for this tree are:

(10) a. ang big man and woman
b. ang man big and woman

Despite the different derivation tree, (10-a) here is identical to (7-a), explaining the ambiguity.

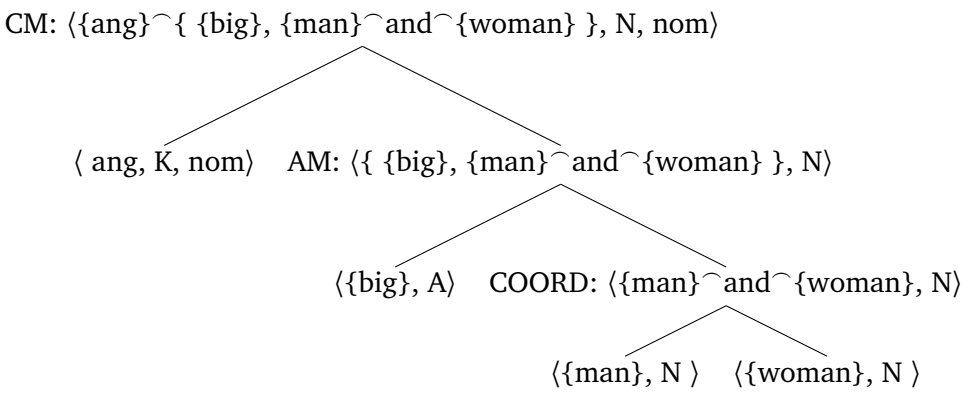


Figure 8: Tree with *man and woman* modified by the adjective

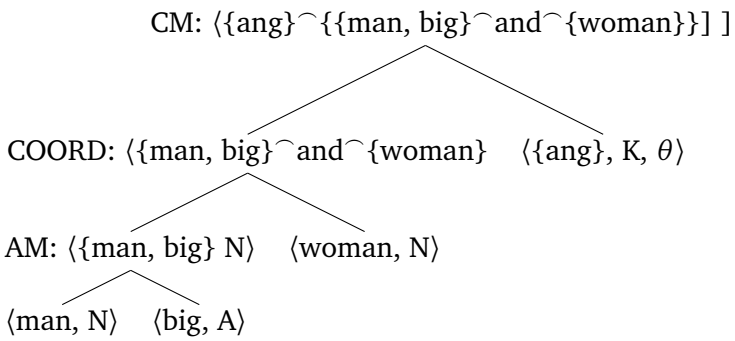


Figure 9: Tree with only *man* modified by the adjective

I also predict correctly that *ang man big and woman* can only mean the man is big. This is because in the case where both the man and woman are big, *man and woman* must form a concatenated constituent by the action of COORD. Then *big* merges with *man and woman* and therefore cannot appear between them in this case.

Conclusion

By adding more elements to the tuples in the lexicon, the grammar can be expressed more succinctly and can capture more thoroughly the relationships between categories. We have seen how a third element can be used to calculate case marking and the fulfillment of theta grids.

Truly free word order can be accounted for if we allow the derivation of concatenated (multi-)sets, rather than just concatenated words. The elements of a multiset are unordered, and arbitrary orders can therefore be defined from a single derivation.

Appendix: Little Tagalog Grammar

$L_T = \langle V, Cat, Rule, Lex, \theta \rangle$

- $Cat := \{\text{P}, \text{N}, \text{A}, \text{Voi}, \text{CJ}\}$

- $\theta := \{a, t, g, r, l, i, b, p\}$
- $\mathbb{K} = \wp\theta$
- $Lex \subset (V^* \times Cat) \cup (V^* \times Cat \times \mathbb{K}) \cup (V* \times Cat \times \wp(\mathbb{K}))$:

$\langle gave, \text{P}, \{a,t,r\}\rangle$	$\langle ang, \text{K}, \text{NOM}\rangle$	$\langle man, \text{N}\rangle$
$\langle cooked, \text{P}, \{a,t\}\rangle$	$\langle sa, \text{K}, \text{GEN}\rangle$	$\langle woman, \text{N}\rangle$
$\langle sneezed, \text{P}, \{a\}\rangle$	$\langle ng, \text{K}, \text{DAT}\rangle$	$\langle book, \text{N}\rangle$
$\langle \text{AV}, \text{Voi}, \{a\}\rangle$	$\langle big, \text{A}\rangle$	$\langle and, \text{CJ}\rangle$
$\langle \text{TV}, \text{Voi}, \{t\}\rangle$		
$\langle \text{IV}, \text{Voi}, \{i\}\rangle$		
$\langle \text{BV}, \text{Voi}, \{b\}\rangle$		
$\langle \text{DV}, \text{Voi}, \text{DAT}\rangle$		

- *Rule*

VM $((x, \text{P}, T), (y, \text{Voi}, S)) \longmapsto (x^\frown y, \text{P}, v_S(T))$

CM $(\langle x, \text{N}\rangle, \langle y, \text{K}, k\rangle) \longmapsto \langle y^\frown x, \text{N}, k\rangle$

AL $(\langle x, \text{N}, k\rangle, \langle y, \text{N}, l\rangle) \longmapsto \langle x \cup y, \text{N}, k \cup l\rangle$ if $(k \cup l)(\text{NOM} \leq 1)$

PA $(\langle x, \text{P}, K\rangle, \langle y, \text{N}, L\rangle) \longmapsto \langle x^\frown y, \text{P}, \emptyset\rangle$ where $K, L \in \wp(\mathbb{K})$.

AM $(\langle x, \text{N}\rangle, \langle y, \text{A}\rangle) \longmapsto \langle x \cup y, \text{N}\rangle$

Coord
$$\begin{cases} (\langle and, \text{Cj}\rangle, \langle x, \text{C}\rangle, \langle y, \text{C}\rangle) \longmapsto \langle x^\frown and^\frown y, \text{C}\rangle \\ (\langle and, \text{Cj}\rangle, \langle x, \text{C}, X\rangle, \langle y, \text{C}, X\rangle) \longmapsto \langle x^\frown and^\frown y, \text{C}, X\rangle \end{cases}$$

Acknowledgements

This paper was written as part of a Fall 2009 course at UCLA taught by Ed Stabler and Ed Keenan. I would also like to thank Raphael Mercado and Nerissa Black for their help as Tagalog consultants and Lisa deMena Travis for her guidance in formulating my initial analysis of Tagalog.

References

Keenan, Edward L., and Edward P. Stabler. 2003. *Bare grammar*. Stanford: CSLI Publications.

Kroeger, Paul. 1993. *Phrase structure and grammatical relations in Tagalog*. Stanford: CSLI Publications.

Affiliation

Meaghan Fowlie
University of California, Los Angeles
mfowlie@ucla.edu

UCLA Working Papers in Linguistics, Papers in Mathematical Linguistics 1
Volume 15, Article 4: 1–53, 2010

A Tree Transducer Model of Reference-Set Computation

Thomas Graf

Reference-set constraints are a special class of constraints used in Minimalist syntax. They extend the notion of well-formedness beyond the level of single trees: When presented with some phrase structure tree, they compute its set of competing output candidates and determine the optimal output(s) according to some economy metric. Doubts have frequently been raised in the literature whether such constraints are computationally tractable (Johnson and Lappin 1999). I define a subclass of Optimality Systems (OSs) that is sufficiently powerful to accommodate a wide range of reference-set constraints and show that these OSs are globally optimal in the sense of Jäger (2002), a prerequisite for them being computable by linear tree transducers. As regular and linear context-free tree languages are closed under linear tree transductions, this marks an important step towards showing that the expressivity of various syntactic formalisms is not increased by adding reference-set constraints. In the second half of the paper, I demonstrate the feasibility of the OS-based transducer approach by exhibiting the transducers for three prominent reference-set constraints, Focus Economy (Reinhart 2006), Merge-over-Move (Chomsky 1995b), and Fewest Steps (Chomsky 1991, 1995b). My approach sheds new light on the internal mechanics of these constraints and suggests that their advantages over standard well-formedness conditions have not been sufficiently exploited in the empirical literature.

Keywords reference-set constraints, transderivationality, Optimality Systems, tree transducers, modelling, finite-state methods, Focus Economy, Merge-over-Move, Fewest Steps

Introduction

Out of all the items in a syntactician's toolbox, reference-set constraints are probably the most peculiar one. When handed some syntactic tree, a reference-set constraint does not determine its well-formedness from inspection of the tree itself. Instead, it constructs a *reference set* — a set containing a number of trees competing against each other — and chooses the optimal candidate from said set.

Consider *Fewest Steps*, also known as the *Shortest Derivation Principle* (Chomsky 1991, 1995a). The reference set that this constraint constructs for any given tree t consists of t itself and all the trees that were assembled from the same lexical items as t. All the trees in the reference set are then ranked by the number of movement

steps that occurred during their assembly, and the tree(s) with the fewest instances of movement is (are) chosen as the winner. All other trees are flagged as ungrammatical, including t if it did not emerge as a winner.

Another reference-set constraint is *Focus Economy* (Szendrői 2001; Reinhart 2006), which accounts for the empirical fact that neutral stress is compatible with more discourse situations than shifted stress. Take a look at the utterances in (1), where main stress is indicated by **bold face**. Example (1a) can serve as an answer to various questions, among others "What's going on?" and "What did your neighbor buy?". Yet the virtually identical (1b), in which the main stress falls on the subject rather than the object, is compatible only with the question "Who bought a book?". These contrasts indicate a difference as to which constituents may be *focused*, i.e. can be interpreted as providing new information.

(1) a. My neighbor bought a **book**.
b. My **neighbor** bought a book.

Focus Economy derives the relevant contrast by stipulating that first, any constituent containing the node carrying the sentential main stress can be focused, and second, in a tree in which stress was shifted from the neutral position a constituent may be focused only if it cannot be focused in the original tree with unshifted stress. In (1a), the object, the VP and the entire sentence can be focused, since these are the constituents containing the main stress carrier. In (1b), the main stress is contained by the subject and the entire sentence, however, only the former may be focused because focusing of the the latter is already a licit option in the neutral stress counterpart (1a).

The application domain of reference-set constraints includes narrow syntax as well as the interfaces. In syntax, one finds Fewest Steps (Chomsky 1995b), Merge-over-Move (Chomsky 1995b, 2000), Pronouns as Last Resort (Hornstein 2001), resumption in Lebanese Arabic (Aoun, Choueiri, and Hornstein 2001), phrase structure projection (Toivonen 2001), the Person Case Constraint (Rezac 2007), Chain Uniformization (Nunes 2004), object extraction in Bantu (Nakamura 1997), and many others. The most prominent interface constraints are Rule I (Grodzinsky and Reinhart 1993; Heim 1998; Reinhart 2006; Heim 2009), Scope Economy (Fox 1995, 2000), and the previously mentioned Focus Economy, but there are also more recent proposals such as Situation Economy (Keshet 2010).

The somewhat esoteric behavior of reference-set constraints coupled with a distinct lack of formal work on their properties has provoked many researches to explicitly reject them (Sternefeld 1996; Gärtner 2002; Potts 2002) and led to various conjectures that they are computationally intractable (Collins 1996; Johnson and Lappin 1999). In this paper, I refute the latter by showing how reference-set constraints can be emulated by a new variant of Optimality Systems (OSs), and I contend that this route paves the way for reference-set constraints to be implemented as finite-state devices; linear bottom-up tree transducers (lbutts), to be precise. Lbutts are of interest for theoretical as well as practical purposes because both regular and linear context-free tree languages are known to be closed under linear transductions, so applying a linear transducer to a regular/linear context-free tree language yields a regular/linear context-free tree language again. On a theoretical level, this provides

us with new insights into the nature of reference-set constraints, while on a practical level, it ensures that adding reference-set constraints to a grammar does not jeopardize its computability. I support my claim by exhibiting lbutts that provide formal models for Focus Economy, Merge-over-Move and the Shortest Derivation Principle. My results shed new light on reference-set computation as well as on OSs and should be of interest to readers from various formal backgrounds, foremost computational phonology and Minimalist Grammars; moreover, when viewed as tree transducers, reference-set constraints also exhibit some previously overlooked connections to synchronous TAG (Shieber and Schabes 1990; Shieber 2004, 2006), the exploration of which I have to leave to future research, unfortunately.

The paper is laid out as follows: After the preliminaries section, which due to space restrictions has to be shorter than is befitting, I give a brief introduction to OSs before defining my own variant, controlled OSs, in Sec. 3. The mathematical core results of this section are a new characterization of the important property of global optimality and a simplification of Jäger's theorem (Jäger 2002) regarding the properties of an OS which jointly ensure that it does not exceed the power of linear tree transducers. In the last three sections, I show how to model Focus Economy, Merge-over-Move and the Shortest Derivation Principle as such restricted OSs, and I discuss the similarities between the tree transducers corresponding to these constraints.

1 Preliminaries and Notation

Let me introduce some notational conventions first. For any two sets A and B, $A \setminus B$ denotes their relative complement and $A \times B$ their cartesian product. The *diagonal* of A is $id(A) := \{\langle a, a\rangle \mid a \in A\}$. Given a relation $R \subseteq A \times B$, its *domain* is denoted by $\mathrm{dom}(R)$, its *range* by $\mathrm{ran}(R)$. For any $a \in \mathrm{dom}(R)$, we let $aR := \{b \mid \langle a, b\rangle \in R\}$, unless R is a function, in which case $aR = R(a)$. The inverse of R is signified by R^{-1}. The *composition* of two relations R and S is $R \circ S := \{\langle a, c\rangle \mid \langle a, b\rangle \in R, \langle b, c\rangle \in S\}$.

Tree languages and tree transductions form an integral part of this paper, however, the technical machinery is mostly hidden behind the optimality-theoretic front-end so that only a cursory familiarity with the subject matter is required. Nevertheless the reader is advised to consult Gécseg and Steinby (1984) and Kepser and Mönnich (2006) for further details. I also assume that the reader is knowledgeable about string languages and generalized sequential machines (see Hopcroft and Ullman 1979).

Definition 1. A *context-free tree grammar* (CFTG) is a 4-tuple $\mathscr{G} := \langle \Sigma, F, S, \Delta\rangle$, where Σ and F are disjoint, finite, ranked alphabets of terminals and non-terminals, respectively, $S \in F$ is the start symbol, and Δ is a finite set of productions of the form $F(x_1, \ldots, x_n) \to t$, where F is of rank n, and t is a tree with the node labels drawn from $\Sigma \cup F \cup \{x_1, \ldots, x_n\}$.

A production is linear if each variable in its left-hand side occurs at most once in its right-hand side. A CFTG is *linear* if each production is linear. A CFTG is a *regular* tree grammar (RTG) if all non-terminals are of rank 0. A tree language is *regular* iff it

is generated by an RTG, and every regular tree language has a context-free language as its string yield.

Definition 2. A *bottom-up tree transducer* is a 5-tuple $\mathscr{A} := \langle \Sigma, \Omega, Q, Q', \Delta \rangle$, where Σ and Ω are finite ranked alphabets, Q is a finite set of states, $Q' \subseteq Q$ the set of final states, and Δ is a set of productions of the form $f(q_1(x_1), \ldots, q_n(x_n)) \rightarrow q(t(x_1, \ldots, x_n))$, where $f \in \Sigma$ is of rank n, $q_1, \ldots, q_n, q \in Q$, and $t(x_1, \ldots, x_n)$ is a tree with the node labels drawn from $\Omega \cup \{x_1, \ldots, x_n\}$.

Example. It is easy to write a transducer for very simple cases of wh-movement. First, let the input alphabet Σ consist of all the labels that appear in trees generated by some input grammar. In a GB setting, for instance, Σ contains all lexical items and X′-annotated category labels. The output alphabet contains every element of Σ and the indexed trace t_{wh}. There are only two states, q_* (which we might call the *identity* state) and q_{wh} (which we might call the *I have previously encountered a wh-word*-state). The final state is q_*. The number of productions of the transducer can be rather high ($2 + 3 * |\Sigma - 1|$), but they can be compressed into 5 production schemes by abstracting away from the specific lexical items. Thus, in the following, σ denotes any element of Σ except the wh-phrase *what*.

(1) $\sigma \rightarrow q_*(\sigma)$

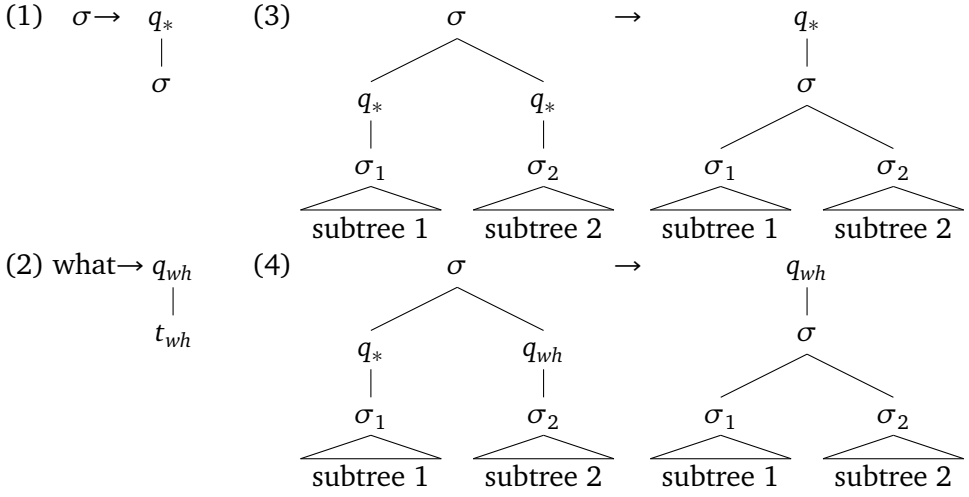
(2) $what \rightarrow q_{wh}(t_{wh})$

(3) $\sigma(q_*(x), q_*(y)) \rightarrow q_*(\sigma(x, y))$

(4) $\sigma(q_*(x), q_{wh}(y)) \rightarrow q_{wh}(\sigma(x, y))$

(5) $\mathrm{TP}(q_i(x), q_{wh}(y)) \rightarrow q_*(\mathrm{CP}(\mathrm{what}, \mathrm{C}'(\mathrm{does}, \mathrm{TP}(x, y))))$

These rules can also be written in tree notation.



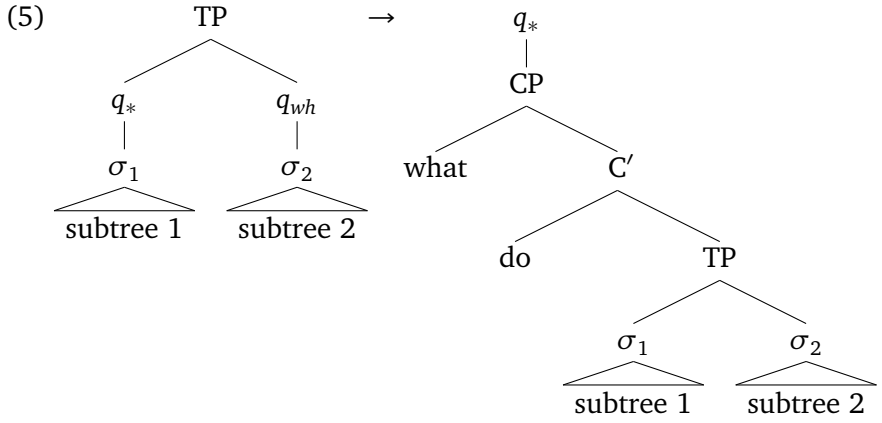


Figure 1 on the following page shows the phrase structure tree for *the men like what* and how it is transformed by the transducer into the tree for *what do the men like*.

Definition 3. A *top-down tree transducer* is 5-tuple $\mathscr{A} := \langle \Sigma, \Omega, Q, Q', \Delta \rangle$, where Σ, Ω and Q are as before, $Q' \subseteq Q$ is the set of initial states, and all productions in Δ are of the form $q(f(x_1, \ldots, x_n)) \to t$, where $f \in \Sigma$ is of rank n, $q \in Q$, and t is a tree with the node labels drawn from $\Omega \cup \{q(x) \mid q \in Q, x \in \{x_1, \ldots, x_n\}\}$.

For the sake of succinctness (but to the detriment of readability), I adopt the following notational conventions for tree transducer productions:

- $\alpha_{\{x,y\}}$ is to be read as "α_x or α_y".
- $\alpha_{a\ldots z}(\beta_{a'\ldots z'}, \ldots, \zeta_{a'',\ldots,z''})$ is to be read as "$\alpha_a(\beta_{a'}, \ldots, \zeta_{a''})$ or ... or $\alpha_z(\beta_{z'}, \ldots, \zeta_{z''})$".

Example. The production $\sigma(q_{ij\{a,b\}}(x), q_{jkc}(y)) \to q_{\{a,c\}}(\sigma(x,y))$ is a schema defining eight productions:

$$\begin{aligned}
&\sigma(q_i(x), q_j(y)) \to q_a(\sigma(x,y)) && \sigma(q_i(x), q_j(y)) \to q_c(\sigma(x,y)) \\
&\sigma(q_j(x), q_k(y)) \to q_a(\sigma(x,y)) && \sigma(q_j(x), q_k(y)) \to q_c(\sigma(x,y)) \\
&\sigma(q_a(x), q_c(y)) \to q_a(\sigma(x,y)) && \sigma(q_a(x), q_c(y)) \to q_c(\sigma(x,y)) \\
&\sigma(q_b(x), q_c(y)) \to q_a(\sigma(x,y)) && \sigma(q_b(x), q_c(y)) \to q_c(\sigma(x,y))
\end{aligned}$$

As with CFTGs, a production is linear if each variable in its left-hand side occurs at most once in its right-hand side. A transducer is *linear* if each production is linear. I denote a linear bottom-up/top-down tree transducer by lbutt/ltdtt. The class of transductions realized by ltdtts is properly contained in the class of transductions realized by lbutts, which in turn is closed under union and composition. The domain and the range of an lbutt are both recognizable, i.e. regular tree languages. The relation τ induced by a (linear) tree transducer is called a (linear) *tree transduction*. For a bottom-up tree transducer, the graph of τ consists of pairs $\langle s, t \rangle$ such that s and t are Σ- and Ω-labeled trees, respectively, and for some $q \in Q'$, $q(t)$ can be obtained from s by finitely many applications of productions $\delta \in \Delta$. The definition is almost unchanged for top-down tree transducers, except that we require that t can

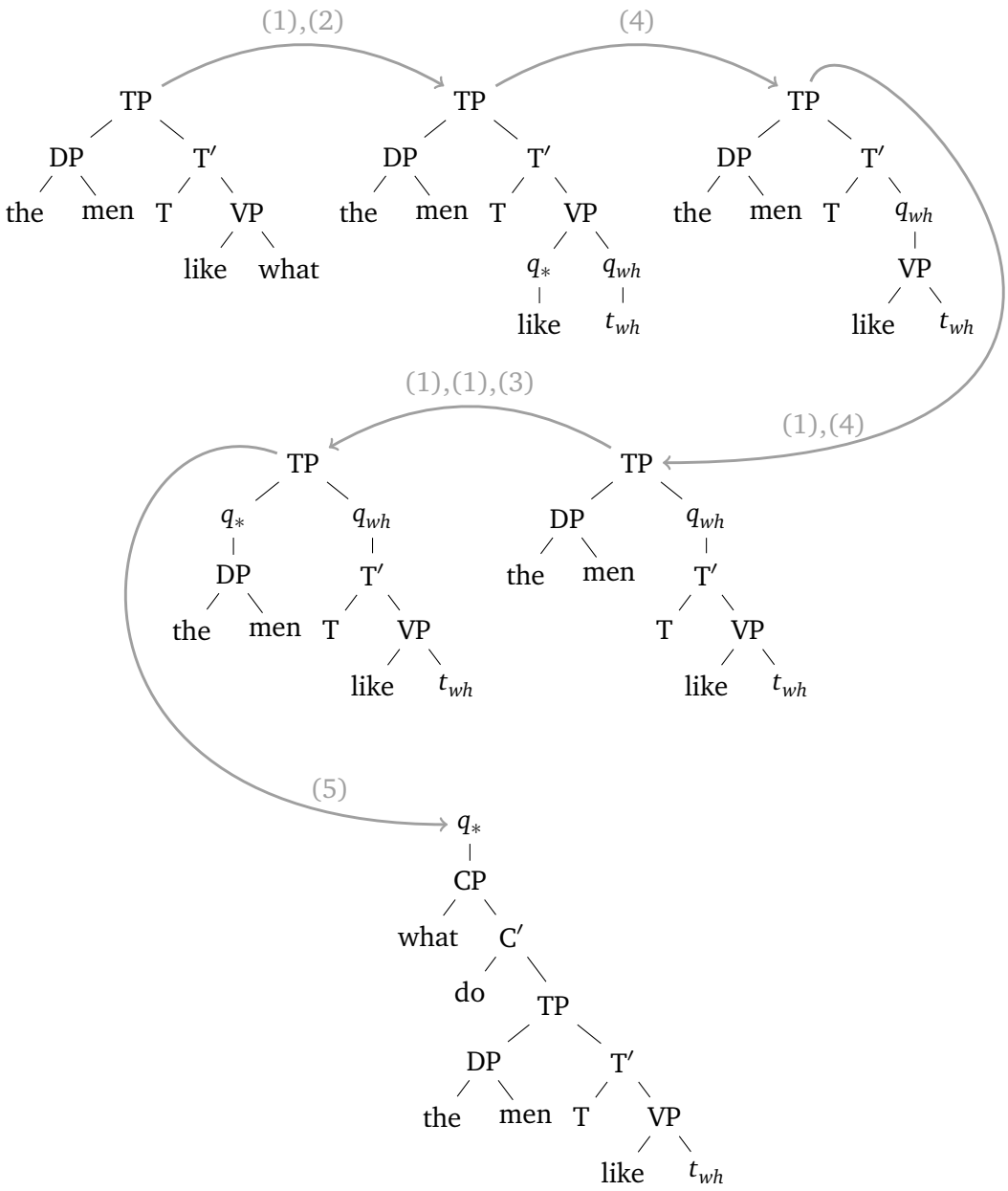


Figure 1: Example of transduction for simple wh-movement

be obtained from $q(s)$. In a slight abuse of terminology, I call a relation *rational* iff it is a finite-state string transduction or a linear tree transduction. For any recognizable tree language L, $id(A)$ is a rational relation. Furthermore, both regular string/tree languages and linear context-free tree languages are closed under rational relations.

Since reference-set constraints originate from Minimalist syntax, I will often assume that the input language to some transduction is given by a Minimalist grammar (MG).

Definition 4. A *Minimalist grammar* is a 5-tuple $\mathscr{G} := \langle \Sigma, F, Types, Lex, O \rangle$, where

- $\Sigma \neq \emptyset$ is the alphabet,
- F is the union of a non-empty set *base* of *basic features* and its prefixed variants $\{=f \mid f \in base\}$, $\{+f \mid f \in base\}$, $\{-f \mid f \in base\}$ of *selector*, *licensor*, and *licensee* features, respectively,
- $Types := \{::, :\}$ serves in distinguishing *lexical* from *derived* expressions,
- the lexicon *Lex* is a finite subset of $\Sigma^* \times \{::\} \times F^+$,
- and O is the set of generating functions to be defined below.

We define the set C of *chains* $\Sigma^* \times Types \times F^*$ (whence $Lex \subset C$) and refer to non-empty sequences of chains as *expressions*, the set of which we call E.

The set O of generating functions consists of the operations *merge* and *move*. The operation *merge*: $(E \times E) \to E$ is the union of the following three functions, for $s, t \in \Sigma^*$, $\cdot \in Types$, $f \in base$, $\gamma \in F^*$, $\delta \in F^+$, and chains $\alpha_1, \ldots, \alpha_k$, $\iota_1, \ldots, \iota_k$, $0 \leq k, l$:

$$\frac{s :: =f\,\gamma \qquad t \cdot f, \iota_1, \ldots, \iota_k}{st : \gamma, \iota_1, \ldots, \iota_k}\ merge1$$

$$\frac{s : =f\,\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f, \iota_1, \ldots, \iota_l}{ts : \gamma, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\ merge2$$

$$\frac{s \cdot =f\,\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f\,\delta, \iota_1, \ldots, \iota_l}{s : \gamma, \alpha_1, \ldots, \alpha_k, t : \delta, \iota_1, \ldots, \iota_l}\ merge3$$

As the domains of all three functions are disjoint, their union is a function, too.

The operation *move*: $E \to E$ is the union of the two functions below, with the notation as above and the further assumption that all chains satisfy the Shortest Move Constraint (SMC), according to which no two chains in the domain of *move* display the same licensee feature $-f$ as their first feature.

$$\frac{s : +f\,\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f, \alpha_{i+1}, \ldots, \alpha_k}{ts : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \alpha_k}\ move1$$

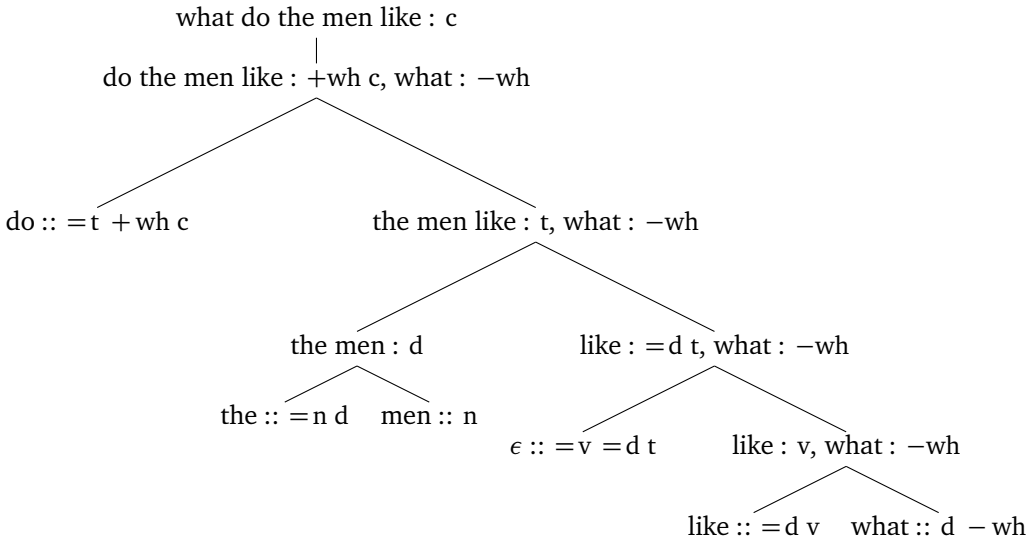
$$\frac{s : +f\,\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f\,\delta, \alpha_{i+1}, \ldots, \alpha_k}{s : \gamma, \alpha_1, \ldots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \ldots, \alpha_k}\ move2$$

The language $L(\mathcal{G})$ generated by $\mathcal{G}$ is the closure of the lexicon under the generating functions.

Example. The following MG assigns the question *what do the men like* the same phrase structure tree as the transducer from the previous example.

what :: d $-$wh	like :: =d v
men :: n	ϵ :: =v =d t
the :: =n d	do :: =t +wh c

The corresponding derivation is depicted below, with binary branching nodes indicating instances of Merge and unary ones instances of Move.



The string language derived by an MG is mildly context-sensitive in the sense of Joshi (1985). In particular, for every MG there exists a strongly equivalent multiple-context free grammar (Michaelis 1998, 2001). This implies that a tree language that can be derived by an MG may not be linear context-free. However, the set of derivation trees of an MG is a regular tree language and there is an effective procedure for obtaining the derived trees from their derivation trees — this holds even of the strictly more powerful class of MGs with unbounded copying (Kobele 2006; Kobele, Retoré, and Salvati 2007).[1]

At the end of Sec. 5.2 and 6.3, I make good use of $\mathcal{L}^2_{K,P}$ (Rogers 1998), an incarnation of monadic second-order logic (MSO) specifically designed for linguistic purposes. MSO is the extension of first-order logic with monadic second-order variables and predicates as well as quantification over them such that the first-order variables represent nodes in the tree and the second-order variables and predicates sets of nodes. A set of finite strings/trees is definable in MSO iff it is regular. Specifics of $\mathcal{L}^2_{K,P}$ will be briefly introduced in the relevant sections. See Thomas (1997) for further background material on MSO and Rogers (1997, 1998) for $\mathcal{L}^2_{K,P}$ in particular.

[1] An alternative regular representation of MGs is given in Kolb, Michaelis, Mönnich, and Morawietz (2003). The method of Kobele et al. is a better choice for this project, though, as many reference-set constraints in the literature operate on derivation trees.

2 Traditional Perspective on Optimality Systems

OSs were introduced independently by Frank and Satta (1998) and Karttunen (1998) as a formalization of Optimality Theory (OT) (Prince and Smolensky 2004), which has been the dominant theory of phonology in mainstream linguistics for the last fifteen years. In OT, well-formed expressions are no longer derived from underlying representations through iterated applications of string rewrite rules, as was the case with SPE (Chomsky and Halle 1968). Instead, underlying representations — which are usually referred to as *inputs* — are assigned a set of *output candidates* by a relation called *generator*, abbreviated GEN. This set is subsequently narrowed down by a sequence of constraints $c_1, \ldots, c_n$ until only the *optimal* output candidates remain. This narrowing-down process proceeds in a fashion such that only the candidates that incurred the least number of violations of constraint c_i are taken into account for the evaluation of c_{i+1}. Thus every constraint acts as a filter on the set of output candidates, with the important addendum that the order in which the filters are applied is crucial in determining optimality.

Consider the example in Fig. 2, which depicts an OT evaluation of output candidates using the tableau notation. Here some input i is assigned three output candidates o_1, o_2 and o_3. The OT grammar uses only three constraints c_1, c_2 and c_3, with each c_i preceding c_{i+1}, $1 \leq i < 2$. Constraint c_1 is applied first. Candidates o_2 and o_3 each violate it once, however, o_1 violates it twice and there are no other output candidates. Thus o_2 and o_3 are the output candidates incurring the least number of violations of the constraint and are allowed to proceed to the next round of the evaluation. Candidate o_1, on the other hand, is ruled out and does not participate in further evaluations. Neither o_2 nor o_3 violate c_2 (nor does o_1, but this is immaterial since it has been previously ruled out), so neither is filtered out. In the third round, o_2 and o_3 are evaluated with respect to c_3. Each of them violates the constraint once, but since there is no candidate that fares better than them (again, o_1 is not taken into consideration anymore), they also survive this round of the evaluation. Thus, o_2 and o_3 are the optimal output candidates for i. If c_3 had been applied before c_1, on the other hand, o_2 and o_3 would lose out against o_1.

i	c_1	c_2	c_3
o_1	2	0	0
o_2	1	0	1
o_3	1	0	1

Figure 2: Example of an OT evaluation in tableau notation

With this intuitive understanding of OT grammars under our belt, the formal definitions of OSs and their *output language* (not to be confused with the *candidate language* ran(GEN)) are straightforward.

Definition 5. An *optimality system* over languages L, L' is a pair $\mathcal{O} := \langle \text{GEN}, C \rangle$ with $\text{GEN} \subseteq L \times L'$ and $C := \langle c_1, \ldots, c_n \rangle$ a linearly ordered sequence of functions $c_i : \text{GEN} \to \mathbb{N}$. For $a, b \in \text{GEN}$, $a <_{\mathcal{O}} b$ iff there is an $1 \leq k \leq n$ such that $c_k(a) < c_k(b)$ and for all $j < k$, $c_j(a) = c_j(b)$.

Definition 6. Given an optimality system $\mathcal{O} := \langle \textsc{Gen}, C \rangle$, $\langle i, o \rangle$ is *optimal* with respect to $\mathcal{O}$ iff both $\langle i, o \rangle \in \textsc{Gen}$ and there is no o' such that $\langle i, o' \rangle \in \textsc{Gen}$ and $\langle i, o' \rangle <_{\mathcal{O}} \langle i, o \rangle$. The transduction induced by $\mathcal{O}$ is given by $\tau := \{\langle i, o \rangle \mid \langle i, o \rangle$ is optimal with respect to $\mathcal{O}\}$. The output language of $\mathcal{O}$ is $\mathrm{ran}(\tau)$.

The important insight of Frank and Satta (1998) as well as Karttunen (1998), which was later improved upon by Wartena (2000), Jäger (2002) and Kepser and Mönnich (2006), is that an OS as defined above can be understood to define a transduction from a set of inputs to its set of optimal output candidates. Moreover, if the OS is suitably restricted, it is guaranteed to define a rational relation, which implies its efficient computability.

Theorem 7. *Let $\mathcal{O} := \langle \textsc{Gen}, C \rangle$ be an OS such that*

- $\mathrm{dom}(\textsc{Gen})$ *is a regular string language, or a regular/linear context-free tree language, and*
- Gen *is a rational relation, and*
- *all $c \in C$ are output-markedness constraints, and*
- *each $c \in C$ defines a regular tree language.*

Then the transduction τ induced by the OS is a rational relation and $\mathrm{ran}(\tau)$ *belongs to the same formal language class as* $\mathrm{dom}(\tau)$.

The theorem makes reference to a term the reader is presumably familiar with from the OT literature, output-markedness, which is easily defined in formal terms.

Definition 8. Given an OS $\mathcal{O} := \langle \textsc{Gen}, C \rangle$, $c \in C$ is an *output-markedness constraint* iff $c(\langle i, o \rangle) = c(\langle i', o \rangle)$ for all $\langle i, o \rangle, \langle i', o \rangle \in \textsc{Gen}$.

In the case of regular string and tree languages, the theorem can be generalized significantly, as was shown by Jäger (2002).

Theorem 9. *Let $\mathcal{O} := \langle \textsc{Gen}, C \rangle$ be an OS such that*

- $\mathrm{dom}(\textsc{Gen})$ *is regular string/tree language, and*
- Gen *is a rational relation, and*
- *all $c \in C$ are output-markedness constraints, and*
- *each $c \in C$ defines a rational relation on* $\mathrm{ran}(\textsc{Gen})$*, and*
- *$\mathcal{O}$ is globally optimal.*

Then the transduction τ induced by the OS is a rational relation and $\mathrm{ran}(\tau)$ *belongs to the same formal language class as* $\mathrm{dom}(\tau)$.

Global optimality is a rather technical notion that requires a lot of finite-state machinery to be in place before it can be made formally precise. Intuitively, an OS is globally optimal iff for every optimal output candidate o it holds that there is no input i such that o is an output candidate for i but not an optimal one. It is worth going through the formal definition, though, as this will also make it clear why the more general theorem does not carry over to linear context-free tree languages.

We start out with two definitions that reimplement the constraints of an OS, *plus* its filtering procedure in relational terms.

Definition 10. Given some $R \subseteq \textsc{Gen}$, we associate with every $c_i \in C$ a relation $rel_i^R := \{\langle o, o' \rangle \mid c_i(o) < c_i(o')\} \cap (R^{-1} \circ R)$, the *ranking of c_i relativized to R*.

The relativization with respect to R is achieved by intersecting the relation representing the ranking the constraint induces over the entire candidate language with $(R^{-1} \circ R)$, which is the relation that holds between two candidates o and o' iff they are competing output candidates for some input i, i.e. iff both $\langle i, o \rangle$ and $\langle i, o' \rangle$ belong to $\textsc{Gen}$. In structural terms, rel_i^R is the substructure obtained from the structure defined by c_i by removing all branches between output candidates that never compete against each other. Note that since the class of rational string relations is closed under intersection, composition, and taking inverse, rel_i^R will be a rational relation iff R is rational and there is a rational relation S such that $S \cap (R^{-1} \circ R) = \{\langle o, o' \rangle \mid c_i(o) < c_i(o')\} \cap (R^{-1} \circ R)$. Linear tree transductions, on the other hand, are not closed under inverse, so when talking about rational tree relations it has to be ensured that rel_i^R itself is rational.

Definition 11. Let $R \subseteq \textsc{Gen}$ and $c_i \in C$. Then $R \downharpoonright c_i := R \circ id(\mathrm{ran}(R) \setminus \mathrm{ran}(R \circ rel_i^R))$ is called the *generalized lenient composition of R with c_i*.

The generalized lenient composition is the OS-model of the OT-filtering procedure. It looks rather scary but is actually easy to master. Let us proceed from the inside to the outside. By composing R with rel_i^R, we obtain the subset of R in which the output candidates are suboptimal. To see this, suppose that o is an optimal output candidate with respect to rel_i^R, so it is a minimal element in the structure defined by rel_i^R. Now suppose that there is an output candidate o' that is worse than o, i.e. $\langle o, o' \rangle \in rel_i^R$. Assuming that R contains the pair $\langle i, o \rangle$, composing R and rel_i^R means composing the pairs $\langle i, o \rangle$ and $\langle o, o' \rangle$, which yields $\langle i, o' \rangle$. The optimal candidate o has been "hidden" by the composition. With this short explanation the reader should now be able to verify that the range of $R \circ rel_i^R$ contains all suboptimal output candidates. Removing those from the range of R and taking the diagonal over this new set thus yields the identity function over the set of optimal output candidates, and composing R with this function hence means filtering out all suboptimal candidates (as a helpful exercise, the reader may verify that we get the right result even if all competing output candidates incur the same number of violations with respect to c_i).

To see that the generalized lenient composition is a finite-state procedure, it suffices to know (i) that the range of a rational relation is a regular language if its domain is regular, (ii) that the class of regular languages is closed under relative complement, and (iii) that the diagonal of a regular set is a rational relation. However,

the class of linear context-free languages is not closed under relative complement, and this is why we need stronger conditions for OSs over linear context-free languages.

The finite-state perspective on OSs also allows for a redefinition of the transductions they induce. As Jäger (2002) shows, transductions can be defined purely in terms of generalized lenient composition.

Theorem 12. *Given an OS* $\mathcal{O} := \langle \textsc{Gen}, \langle c_1, \ldots, c_n \rangle \rangle$ *satisfying the conditions above,* $\tau := \textsc{Gen} \mid c_1 \mid \ldots \mid c_n$.

But we set out to give a formal definition of global optimality, and now we have all the required machinery.

Definition 13. Let R and S be relations. *Optimality is global with respect to R and S* iff

$$\forall i, o[(iRo \wedge \neg\exists p(iRp \wedge pSo)) \rightarrow \neg\exists p(p \in \mathrm{ran}(R) \wedge pSo)]$$

Now let $\mathcal{O} := \langle \textsc{Gen}, C \rangle$, $C := \langle c_1, \ldots, c_n \rangle$ be an optimality system. Then we say that $\mathcal{O}$ is *globally optimal* or *satisfies global optimality* iff optimality is global with respect to $\textsc{Gen} \mid c_1^{i-1} := \textsc{Gen} \mid c_1 \mid \ldots \mid c_{i-1}$ and the ranking of c_i relativized to $\textsc{Gen} \mid c_1^{i-1}$ for all $1 \leq i \leq n$.

The important thing to keep in mind here is that S does not correspond to the absolute ranking induced by a constraint c_i, but its relativized ranking (i.e. the one with "holes" in it). Therefore global optimality does not demand that if o is optimal for i, there is no output p that is more optimal than o for i, but rather that no output p that competes against o with respect to some input j is more optimal than o. In less technical terms, if o is an optimal output candidate for some input i, then there is no input j for which o is an output candidate but not an optimal one.

In the next section, I introduce a notational variant of OS and use this variant to show that global optimality is always satisfied for reference-set constraints, thereby highlighting them as a very restricted subclass of OS. For the sake of simplicity, I will implicitly assume that the OSs consist of only one constraint, as this does away with a lot of tedious induction steps. Fortunately, this has no negative consequences for the relevance of the results. First, when modelling reference-set constraints we do not need more than one constraint. Second, every OS $\mathcal{O} := \langle \textsc{Gen}, \langle c_1, \ldots, c_n \rangle \rangle$ can be decomposed into a cascade of n OSs $\mathcal{O}_k := \langle \textsc{Gen}_k, \langle c_k \rangle \rangle$ such that $\textsc{Gen}_k$ is given by $\textsc{Gen}$ for $k = 1$ and $\{\langle i, o \rangle \in \textsc{Gen}_{k-1} \mid \langle i, o \rangle$ is optimal with respect to $c_{k-1}\}$ otherwise (for finite-state OS, the latter is equivalent to $\textsc{Gen}_{k-1} \mid c_{k-1}$).

3 Controlled Optimality Systems

OSs are perfectly capable of modelling reference-set constraints. The reference set for any input i is defined by $i\textsc{Gen}$, and the evaluation metric can straightforwardly be implemented as a sequence of constraints. Fewest Steps, for instance, can be viewed as an OS in which a tree t is related by $\textsc{Gen}$ to all the trees that were constructed from the same lexical items as t, including t itself. Besides that, the OS has only one

constraint *Trace, which punishes traces. As a consequence, only the trees with the least number of traces will be preserved, and these are the optimal output candidates for t. Focus Economy is slightly more involved and allows for different models. In one of them (which isn't the one I will formalize in Sec. 5.2), Gen relates a tree t without stress annotation but a focus-marked constituent to its analogs with neutral and shifted stress. The OS then uses two constraints, the first of which weeds out trees in which the focus-marked constituent does not contain the main stress carrier, while the second penalizes derivation length (the assumption here is that derivations for trees with shifted stress are longer than those for trees with neutral stress). Then a tree derived by stress shift will be optimal just in case the equivalent tree with neutral stress has already been ruled out by the first constraint.

While these two examples show that OSs certainly can get the job done, the way output candidates are specified for reference-set constraints actually relies on additional structure — the reference sets — that is only indirectly represented by Gen. In the following I introduce controlled OSs as a variant of standard OSs that is closer to reference-set computation by making reference sets prime citizens of OSs and demoting Gen to an ancillary relation that is directly computed from them. Note, though, that in the case of a constraint like Focus Economy, where input language and candidate language are disjoint, distinct inputs might be assigned the same reference set. In such a configuration, it makes sense to map entire sets of inputs to reference sets, rather than the individual inputs themselves. Let us call such a set of inputs a *reference type*. An OS can then be defined by reference types and a function mapping them to reference sets:

Definition 14. An *$\mathscr{F}$-controlled optimality system* over languages L, L' is a 4-tuple $\mathcal{O}[\mathscr{F}] := \langle \textsc{Gen}, C, \mathscr{F}, \gamma \rangle$, where

- Gen and C are defined as usual,
- $\mathscr{F}$ is a family of non-empty subsets of L, each of which we call a *reference type*,
- the *control map* $\gamma : \mathscr{F} \to \wp(L') \setminus \{\emptyset\}$ associates every reference type with a *reference set*, i.e. a set of output candidates,
- the following conditions are satisfied
  - *exhaustivity*: $\bigcup_{X \in \mathscr{F}} X = L$
  - *bootstrapping*: $x\textsc{Gen} = \bigcup_{x \in X \in \mathscr{F}} X\gamma$

Every controlled OS can be translated into a canonical OS by discarding its third and fourth component (i.e. $\mathscr{F}$ and the control map γ). In the other direction, a controlled OS can be obtained from every canonical OS by setting $\mathscr{F} := \{\{i\} \mid i \in L\}$ and $\gamma : \{i\} \mapsto i\textsc{Gen}$. So the only difference between the two is that controlled OSs modularize Gen by specifying it through reference types and the control map.

Now that OSs operate (at least partially) at the level of sets, it will often be interesting to talk about the set of optimal output candidates assigned to some reference type, rather than one particular input. But whereas the set of optimal

output candidates is always well-defined for inputs — recall that for any input i this set is given by $i\tau$ — we have to be more careful when lifting it to reference types, because distinct inputs that belong to the same reference type may not necessarily be assigned the same optimal output candidates. Such a situation may arise in OSs with faithfulness or input-markedness constraints, which are sensitive to properties of the input, or when two inputs i and j are of reference type X, but in addition j is also of reference type Y. Given this ambiguity, one has to distinguish between the set of output candidates that are optimal for at least one member of reference type X, and the set of output candidates that are optimal for all members of reference type X. The former is the *up-transduction* $X\tau^{\uparrow} := \bigcup_{x\in X} x\tau$, the latter the *down-transduction* $X\tau^{\downarrow} := \bigcap_{x\in X} x\tau$.

At this point it might be worthwhile to work through a simple example. Fig. 3 on the facing page depicts a controlled OS and the distinct steps of its computation. We are given a collection of reference types consisting of $\textsc{Red} := \{i_1, i_2, i_3, i_4, i_5, i_6\}$, $\textsc{Sienna} := \{i_4\}$, $\textsc{Teal} := \{i_5, i_6, i_7\}$, $\textsc{Purple} := \{i_8\}$, and $\textsc{Lime} := \{i_7, i_9, i_{10}\}$. The reference sets are $\textsc{Blue} := \{o_1, o_2, o_3\}$, $\textsc{Orange} := \{o_3, o_4, o_5, o_6, o_7\}$, $\textsc{Green} := \{o_6, o_7\}$, and $\textsc{Brown} := \{o_8, o_9\}$. Finally, the graph of γ consists of the pairs $\langle \textsc{Red}, \textsc{Blue}\rangle$, $\langle \textsc{Sienna}, \textsc{Brown}\rangle$, $\langle \textsc{Teal}, \textsc{Green}\rangle$, $\langle \textsc{Purple}, \textsc{Brown}\rangle$, and $\langle \textsc{Lime}, \textsc{Orange}\rangle$. Note that a reference type may overlap with another reference type or even be a proper subset of it, and the same holds for reference sets. This means that an input can belong to several reference types at once. Consequently, $x\textsc{Gen}$ may be a superset of $X\gamma$ for every reference type X that contains x, as is the case for i_4, say, but not for i_7, even though both are assigned exactly two reference types. Input i_4 is related by $\textsc{Gen}$ to all outputs contained in $\textsc{Red}\gamma \cup \textsc{Sienna}\gamma = \textsc{Blue} \cup \textsc{Brown} = \{o_1, o_2, o_3, o_8, o_9\}$, whereas i_7 is related to $\textsc{Lime}\gamma \cup \textsc{Teal}\gamma = \textsc{Orange} \cup \textsc{Green} = \textsc{Orange} = \{o_3, o_4, o_5, o_6, o_7\}$. As soon as $\textsc{Gen}$ has been determined from the reference types and the control map, the computation proceeds as usual with the constraints of the OS filtering out all suboptimal candidates.

Interestingly, almost all reference-set constraints fall into two classes with respect to how reference types and reference sets are distributed (see Fig. 4 on the next page). In the case of Fewest Steps, where the input language is also the candidate language, each reference type is mapped to itself, that is to say, there is no distinction between reference types and reference sets. A constraint like Focus Economy, on the other hand, requires not only the input language and the candidate language to be disjoint, but also all reference sets and reference types.

The Fewest Steps-type class of OSs is best captured by the notion of *endocentricity*.

Definition 15. An $\mathscr{F}$-controlled OS is *endocentric* iff $X\gamma = X$ for all $X \in \mathscr{F}$.

Constraints like Focus Economy, on the other hand, are *output-partitioned* and *output-segregated* in the following senses:

Definition 16. An $\mathscr{F}$-controlled OS is

- *output-partitioned* iff for all distinct $X, Y \in \mathscr{F}$, $X\gamma \neq Y\gamma$ implies $X\gamma \cap Y\gamma = \emptyset$.
- *output-segregated* iff for all distinct $X, Y \in \mathscr{F}$, neither $X\gamma \subseteq Y\gamma$ nor $Y\gamma \subseteq X\gamma$.

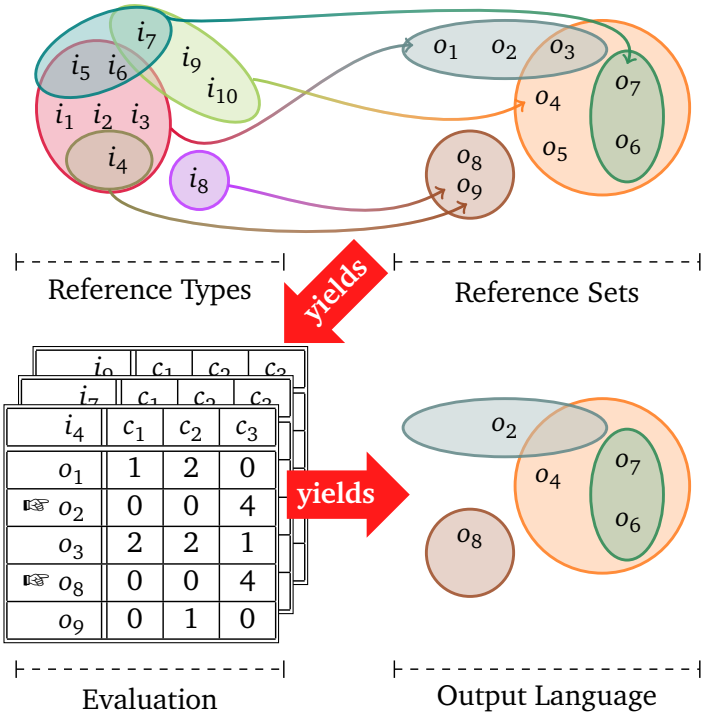


Figure 3: Example of a controlled OS; GEN is defined in a modular fashion using reference types, reference sets, and the control map γ from reference types to reference sets

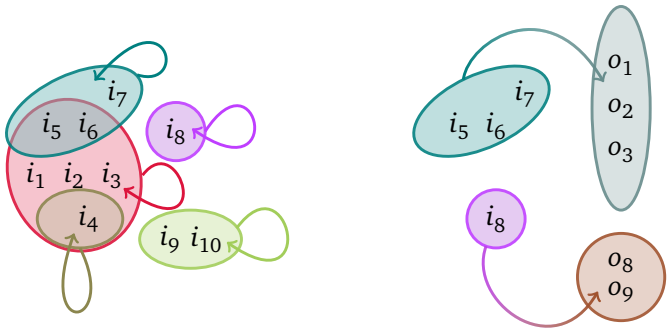


Figure 4: Almost all instances of reference-set computation in the literature use one of the two configurations above, both of which are output joint preserving

While this class certainly fits Focus Economy and its ilk pretty well, it does not extend naturally to endocentric OSs, which means that we would have to study the two classes independently from each other. In particular, endocentric OSs aren't necessarily output-partitioned, as they do allow for overlapping reference-sets. Nor is output-segregation a meaningful restriction on endocentric OSs: Given some $\mathscr{F}$-controlled OS, assume $x \in X$, $y \in Y$, $X \subseteq Y$. Then x is also a member of Y, whence both $\langle x, y \rangle$ and $\langle y, x \rangle$ are in (the graph of) GEN. The reference type X is immaterial for computing GEN and can be removed from $\mathscr{F}$ without consequences. Given these discrepancies, we should not be too sure that results pertaining to one class can easily be carried over to the other. As it turns out, though, a natural unification of the two subclasses is available in the form of *output joint preservation*.

Definition 17. An $\mathscr{F}$-controlled optimality system is *output joint preserving* iff for all distinct $X, Y \in \mathscr{F}$, $X\gamma \cap Y\gamma \neq \emptyset \rightarrow X \cap Y \neq \emptyset$.

The OS depicted in Fig. 3 on the preceding page fails output joint preservation. It is clearly violated by SIENNA and PURPLE, which are disjoint yet mapped to the same reference set, BROWN. It isn't respected by RED and LIME, either, which are mapped to BLUE and ORANGE, respectively, the intersection of which is non-empty even though RED and LIME are disjoint. Moving on to Fig. 4 on the previous page, we observe that every endocentric OS is output joint preserving. In addition, all output-partitioned OSs trivially satisfy output joint preservation, too, because there are no joints to preserve in these OSs. These inclusions tell us that output joint preservation is certainly general enough a property to encompass the kinds of controlled OSs we are interested in. In the following, I show that in conjunction with another property it is also sufficiently restrictive to establish a link to global optimality.

Definition 18. An $\mathscr{F}$-controlled OS is *type-level optimal* iff $X\tau^{\uparrow} \restriction X\gamma = X\tau^{\downarrow} \restriction X\gamma$ for all $X \in \mathscr{F}$.

Type-level optimality is essentially the restriction of global optimality to reference types: If o is an optimal output candidate for some $x \in X$, then there is no $y \in X$ such that o isn't optimal for y. Against this backdrop, the following lemma is hardly surprising.

Lemma 19. *Let $\mathscr{O}[\mathscr{F}]$ an $\mathscr{F}$-controlled OS. Then $\mathscr{O}[\mathscr{F}]$ is type-level optimal if it is globally optimal.*

Proof. We prove the contrapositive. If $\mathscr{O}[\mathscr{F}]$ is not type-level optimal, then it holds for some $X \in \mathscr{F}$ that $X\tau^{\uparrow} \restriction X\gamma \neq X\tau^{\downarrow} \restriction X\gamma$. But this implies that there are $x, y \in X$ and $z \in X\gamma$ such that $x\tau \ni z \notin y\tau$, which is an unequivocal violation of global optimality. □

It should also be pointed out that type-level optimality is trivially satisfied if all reference types are singleton. This setting also provides numerous examples that show that the converse of the lemma does not hold. For instance, let $\mathscr{O}[\mathscr{F}]$ consist only of the reference types $\{i\}$ and $\{j\}$, which are both mapped to the reference set $\{o, p\}$, but $i\tau = \{o\}$ whereas $j\tau = \{p\}$. Then $\mathscr{O}[\mathscr{F}]$ is type-level optimal but not

globally optimal. The problem is that type-level optimality leaves a loop-hole for OSs, as different reference types may have overlapping reference sets but disagree on which candidates in the intersection they deem optimal. This hole is patched by output joint preservation.

Theorem 20. *Every output joint preserving OS is type-level optimal iff it is globally optimal.*

Proof. The right-to-left direction follows from Lem. 19. We prove the contrapositive of the other direction. If $\mathcal{O}[\mathscr{F}]$ fails global optimality, then there are $x, y \in L$ and $z \in L'$ such that $\langle x, z\rangle, \langle y, z\rangle \in \textsc{Gen}$ yet $x\tau \ni z \notin y\tau$. W.l.o.g. let $x \in X$ and $y \in Y$, $X, Y \in \mathscr{F}$, whence $z \in X\gamma \cap Y\gamma$. As $\mathcal{O}[\mathscr{F}]$ is output joint preserving, $X\gamma \cap Y\gamma \neq \emptyset$ entails $X \cap Y \neq \emptyset$. Pick some $p \in X \cap Y$. Now assume towards a contradiction that $\mathcal{O}[\mathscr{F}]$ is type-level optimal. Then it holds that $X\tau^{\uparrow} \restriction X\gamma = X\tau^{\downarrow} \restriction X\gamma$ and $Y\tau^{\uparrow} \restriction Y\gamma = Y\tau^{\downarrow} \restriction Y\gamma$, so $z \in x\tau$ implies $z \in p\tau$, whereas $z \notin y\tau$ implies $z \notin p\tau$. Contradiction. It follows that $\mathcal{O}[\mathscr{F}]$ is not type-level optimal. □

To reiterate, type-level optimality ensures that optimality is fixed for entire reference types, so the individual inputs can be ignored for determining optimality. However, it is too weak a restriction to rule out disagreement between reference types that are mapped to overlapping reference sets, so output joint preservation has to step in; it guarantees that if two reference types X and Y share at least one output candidate, there exists some input p belonging to both X and Y that will be faced with conflicting requirements if X and Y disagree with respect to which candidates in $X\gamma \cap Y\gamma$ they deem optimal (since the OS is type-level optimal, optimality can be specified for entire reference types rather than their members). It is clear, then, that the conditions jointly imply global optimality.

Given our interest in using controlled OS to investigate the computability of reference-set constraints, it would be advantageous if we could read off the constraints right away whether they interfere with type-level optimality. This is indeed feasible thanks to the following entailment.

Lemma 21. *Let $\mathcal{O}[\mathscr{F}] := \langle \textsc{Gen}, C, \mathscr{F}, \gamma\rangle$ an $\mathscr{F}$-controlled OS such that every $c \in C$ is an output-markedness constraint. Then $\mathcal{O}[\mathscr{F}]$ is type-level optimal.*

Proof. Assume the opposite. Then for some $X \in \mathscr{F}$, $X\tau^{\uparrow} \restriction X\gamma \neq X\tau^{\downarrow} \restriction X\gamma$, whence there are $x, y \in X$ and $z \in X\gamma$ with $x\tau \ni z \notin y\tau$. But this is the case only if there is some $c \in C$ such that $c(\langle x, z\rangle) \neq c(\langle y, z\rangle)$, i.e. c isn't an output-markedness constraint. □

Corollary 22. *Let $\mathcal{O}[\mathscr{F}] := \langle \textsc{Gen}, C, \mathscr{F}, \gamma\rangle$ an output joint preserving OS such that every $c \in C$ is an output-markedness constraint. Then $\mathcal{O}[\mathscr{F}]$ is globally optimal.*

Combining these results, we arrive at the equivalent of Thm. 7 for $\mathscr{F}$-controlled OSs.

Corollary 23. *Let $\mathcal{O}[\mathscr{F}] := \langle \textsc{Gen}, C, \mathscr{F}, \gamma\rangle$ an $\mathscr{F}$-controlled OS such that*

- $\mathrm{dom}(\textsc{Gen})$ *is a regular string language, or a regular/linear context-free tree language, and*

- GEN *is a rational relation, and*
- *all* $c \in C$ *are output-markedness constraints, and*
- *each* $c \in C$ *defines a rational relation on* ran(GEN)*/a rational tree language, and*
- $\mathcal{O}[\mathscr{F}]$ *is output joint preserving.*

Then the transduction τ *induced by the OS is a rational relation and* ran(τ) *belongs to the same formal language class as* dom(τ)*.*

In sum, then, not only do output joint preserving OSs look like a solid base for modelling reference-set constraints, they also have the neat property that the global optimality check is redundant, thanks to Lem. 21. As it is pretty easy to determine for any given reference-set constraint whether it can be modeled by output-markedness constraints alone, the only stumbling block in the implementation of constraints that can be modeled by output joint preserving OSs is the transducers for the constraints and GEN. If those transducers each define a rational relation, so does the entire optimality system.

It is not difficult to see, however, that Thm. 20 leaves ample room for generalization. After all, the only role of output joint preservation is to ensure — in a rather round-about way — that reference types with overlapping reference sets agree on which candidates in the intersection they consider optimal. If this criterion can be expressed directly, output joint preservation is redundant.

Definition 24. An $\mathscr{F}$-controlled OS $\mathcal{O}[\mathscr{F}]$ satisfies *synchronized optimality* or is *in sync* iff it is type-level optimal and satisfies the following condition:

(*) for all $X, Y \in \mathscr{F}$ with $X\gamma \cap Y\gamma \neq \emptyset$, if $z \in X\gamma \cap Y\gamma$ is an optimal output candidate for X, it is also optimal for Y.

Theorem 25. *An* $\mathscr{F}$*-controlled OS* $\mathcal{O}[\mathscr{F}]$ *satisfies synchronized optimality if and only if it is globally optimal.*

Proof. In proving the contrapositive of the left-to-right direction, we distinguish two cases. If it holds for all $X, Y \in \mathscr{F}$ that $X\gamma \cap Y\gamma = \emptyset$, then (*) is vacuously satisfied but type-level optimality does not hold since $\mathcal{O}[\mathscr{F}]$ fails global optimality, so there has to be a reference type $X \in \mathscr{F}$ whose inputs disagree on the optimality of some $o \in X\gamma$. So assume that $\mathcal{O}[\mathscr{F}]$ is type-level optimal but not output-partitioned. Then the lack of global optimality entails that types $X, Y \in \mathscr{F}$ disagree on the optimality of some $o \in X\gamma \cap Y\gamma$, whence (*) is not satisfied.

It only remains for us to show the implication in the other direction. We prove the contrapositive. If $\mathcal{O}[\mathscr{F}]$ is not in sync, then it fails type-level optimality or violates (*). In the former case, Lem. 19 tells us immediately that the OS does not satisfy global optimality. Assume then w.l.o.g. that $\mathcal{O}[\mathscr{F}]$ is type-level optimal but (*) does not hold. Then there are $X, Y \in \mathscr{F}$ and $z \in X\gamma \cap Y\gamma$ such that $x\text{GEN} \ni z \in y\text{GEN}$ for some $x \in X$ and some $y \in Y$, yet $x\tau \ni z \notin y\tau$. Thus $\mathcal{O}[\mathscr{F}]$ is not globally optimal. □

While Thm. 25 is more general than Thm. 20, its use is also more limited for practical purposes. This is because synchronized optimality does not follow from restricting the OS to output-markedness constraints. Just consider a case where $X\gamma$ is a proper subset of $Y\gamma$. Then it cannot be precluded that even though $x \in X\gamma$ is optimal for X, there is some $y \in Y\gamma \setminus X\gamma$ that it loses out against, which means that x is not optimal for Y. In particular, if $X\gamma$ is singleton, x will always be optimal for X, no matter how bad a candidate it is. Without the entailment from output-markedness constraints, we are forced to manually check for synchronized optimality, which might be a laborious process. As output joint preservation seems to be a robust property of reference-set constraints and makes it a lot easier to check for global optimality, Thm. 20 is of greater importance.

4 Transduction Preserving Operations

One argument that could be leveled against approaching global optimality by means of the subclass of output joint preserving OSs rather than through synchronized optimality is that output-joint preservation does not extend to several configurations that seem very natural and may in principle give rise to globally optimal OSs. Consider for instance an $\mathscr{F}$-controlled OS where $\mathscr{F}$ consists only of two disjoint reference types X and Y, which are mapped to the same reference set. Such an OS fails output joint preservation, yet it might be globally optimal if it is type-level optimal.

One reply to this concern is to emphasize once more that such configurations do not seem to occur with reference-set constraints. However, such a rebuttal would be rather unappealing on a theoretical level, especially because the solution to the problem above is simple: if we take the union of the reference types X and Y, what we get is an OS that defines the same generator relation, and thus the same transduction, yet satisfies output joint preservation. So it seems that output joint preservation is not as restrictive a property as one might think, provided that we allow ourselves to meddle with the makeup of $\mathscr{F}$. The question we have to answer, then, is in which ways $\mathscr{F}$ may be manipulated without affecting the transduction induced by an OS.

Note that the malleability of $\mathscr{F}$ is interesting for practical purposes, too, as it might allow us to deal with peculiar cases where some reference types taken by themselves are considerably more complex than their union. To give an example, let P be the set of strings over some alphabet Σ whose length is prime, and Q its complement. Now P does not define a regular language, not even a context-free one (it is context-sensitive), but the union of P and Q is Σ^*, which is regular (in fact, it is strictly 2-local, i.e. it belongs to one of the weakest classes of string languages that are commonly studied).

So let us see what kind of operations can be applied to $\mathscr{F}$ without altering the transduction τ induced by the OS. In our short discussion above, it was already mentioned that if an operation has no effect on GEN, it has no effect on τ either. In the case at hand, the operation was to take unions of reference types that are mapped to the same reference set. Or speaking in functional terms, we took the

union of all reference types that have the same image under the control map γ. So we only turned γ from a many-to-one into a one-to-one function.

Theorem 26. *Let $\mathcal{O}[\mathscr{F}] := \langle \textsc{Gen}, C, \mathscr{F}, \gamma \rangle$ be an $\mathscr{F}$-controlled OS over language L, L'. Then there is an $\mathscr{F}'$-controlled OS $\mathcal{O}[\mathscr{F}'] := \langle \textsc{Gen}, C, \mathscr{F}', \gamma' \rangle$ over the same languages such that γ' is one-to-one (and $\mathscr{F}'$ is obtained from $\mathscr{F}$ by taking the union of all $X, Y \in \mathscr{F}$ with $X\gamma = Y\gamma$).*

Thanks to this theorem, only OSs whose control map is one-to-one need to be considered in the remainder of this section. In these cases taking unions of reference types also requires taking union of reference sets. So if $X\gamma = A$ and $Y\gamma = B$, we want $(X \cup Y)\gamma$ to be $A \cup B$. The same holds in the other direction, whence it does not make a real difference whether we talk about unions of reference types or reference sets. Soon though it will become evident that the conditions one may want to impose are best expressed over reference sets; consequently, "taking unions" should be read as "taking unions of reference sets" in the following.

Many examples are readily at hand that establish that transductions are not preserved under arbitrary union. Consider an OS with $\mathscr{F} := \{\{a, b\}, \{b, c\}\}$, $\{a, b\}\gamma = \{d\}$, $\{b, c\}\gamma = \{e\}$, $a\tau = \{d\}$, $c\tau = \{e\}$, and $b\tau = \{d, e\}$. If we take the union of $\{d\}$ and $\{e\}$ (and consequently also of $\{a, b\}$ and $\{b, c\}$), there is no guarantee that the transduction will remain the same. In fact, it is very unlikely, since the constraints of the OS would have to be sensitive to the input in such a way that d and e are equally optimal for b, but d is a better output for a and at the same time worse for c. This is a feasible out-turn, but hardly a common one. In particular, there is no way such a result could obtain if all constraints were output-markedness constraints, which is an important restriction for us. It also implies that we lost global optimality by unionizing the reference sets. In the light of this outcome it seems prudent to focus our attention on globally optimal OSs that use only output-markedness constraints; after all, we are mostly interested in tinkering with computable OSs and extending the applicability of output joint preservation, so these restrictions have no negative repercussions for our endeavor.

For globally optimal OSs with output-markedness constraints only, then, arbitrary union will in general induce a change in the transduction. As indicated by the example above, this is almost inevitable if the reference sets are disjoint, since it is very unlikely that all optimal output candidates of reference set A incur the same number of violations with every constraint as all optimal output candidates of reference set B. So what about taking unions of overlapping reference sets? Can we find restrictions for this case that will ensure that the transduction itself remains untouched?

At first sight overlapping reference sets appear to be just as problematic. If we add an element f to the reference sets in the previous example, we run into the same problems nonetheless. The source of all our troubles isn't the disjointness of the reference sets, but that by unionizing the reference sets, the optimal output candidates of reference set A are suddenly confronted with new contenders, the optimal output candidates of reference set B; unless all optimal output candidates of A and B are evenly matched, a change in the transduction is inevitable. Figuratively speaking, taking unions is a little bit like reshuffling the pack. For disjoint reference

sets, the reshuffling always creates an unpredictable situation, but with overlapping reference sets, there is a case where we can still predict the outcome after the reshuffling: if all optimal output candidates already had to compete against each other in the original OS.

Definition 27. Given an OS $\mathcal{O}[\mathscr{F}] := \langle \textsc{Gen}, \langle c_1, \ldots, c_n \rangle, \mathscr{F}, \gamma \rangle$ over L, L', $z \in L'$ is a *finalist* iff it is in the range of $\textsc{Gen} \restriction c_1 \restriction \ldots \restriction c_{n-1}$. Two sets $X, Y \in \mathscr{F}$ with $X\gamma \cap Y\gamma \neq \emptyset$ are *finalist intersective* iff $X\gamma \cap Y\gamma \supseteq \{z \in X\gamma \cup Y\gamma \mid z \text{ is a finalist}\}$. An optimality system is *finalist intersective* iff all $X, Y \in \mathscr{F}$ with $X\gamma \cap Y\gamma \neq \emptyset$ are finalist intersective.

The notion of being finalist intersective is rather indirect, ideally there would be a particular arrangement of reference sets that can be linked to it in a systematic way. Now observe that no optimality system $\mathcal{O}[\mathscr{F}]$ with distinct $X, Y, Z \in \mathscr{F}$ such that $X\gamma \cap Y\gamma \neq \emptyset$, $X\gamma \cap Z\gamma \neq \emptyset$ and $X\gamma \cap Y\gamma \cap Z\gamma = \emptyset$ is finalist intersective. Otherwise, all finalists of X would have to be contained in $X\gamma \cap Y\gamma$ and $X\gamma \cap Z\gamma$, i.e. $(X\gamma \cap Y\gamma) \cap (X\gamma \cap Z\gamma) = X\gamma \cap Y\gamma \cap Z\gamma$, which is the empty set. In fact it can be shown that all finalist intersective OSs are built from one basic pattern, which I call a blossom.

Definition 28. Given an $\mathscr{F}$-controlled OS $\mathcal{O}[\mathscr{F}]$ with $n \geq 1$ distinct $X_1, \ldots, X_n \in \mathscr{F}$ such that $\bigcap_{i=1}^{n} X_i\gamma$ is non-empty, we call $B := \{X_1\gamma, \ldots, X_n\gamma\}$ a *blossom* of $\mathcal{O}[\mathscr{F}]$, each $X_i\gamma$ a *petal* of B and $\bigcap_{i=1}^{n} X_i\gamma$ the *stem* of B. We say that B is maximal iff there is no distinct $X_{n+1} \in \mathscr{F}$ with $X_{n+1}\gamma \cap X_i\gamma \neq \emptyset$.

From the definition it follows immediately that all distinct maximal blossoms are disjoint.

Lemma 29. *Let $\mathcal{O}[\mathscr{F}]$ be an $\mathscr{F}$-controlled OS with $B := \{X_1, \ldots, X_n\} \subseteq \mathscr{F}$, $n \geq 1$. If $X_1\gamma, \ldots, X_n\gamma$ are pairwise finalist intersective then B is a blossom, the stem of which contains the finalists of each X_i, $1 \leq i \leq n$.*

Proof. If $\mathcal{O}[\mathscr{F}]$ is output partitioned, this is trivially true. In all other cases, we have to show that $\bigcap_{i=1}^{n} X_i$ is non-empty and that it contains every finalist. This follows from a simple proof by induction: W.l.o.g. put all members of B in a total order that is reflected by their index. Now consider X_1 and X_2 in B. Since they are finalist intersective, $X_1 \cap X_2 \neq \emptyset$ and contains all finalists of X_1 and X_2. Now suppose that M is the intersection of $X_1, \ldots, X_m \in B$, $2 < m < n$; by assumption it contains the finalists of each X_i. By virtue of pairwise finalist intersectivity, $X_{m+1} \cap X_i$ has to contain all the finalists of X_{m+1} as well as X_i for every $1 \leq i \leq m$, so $X_{m+1} \cap M$ is non-empty and contains all the finalists of $X_1, \ldots, X_{m+1}$. □

Theorem 30. *Let $\mathcal{O}[\mathscr{F}]$ be an OS that satisfies global optimality and uses only output-markedness constraints. Then global optimality of $\mathcal{O}[\mathscr{F}]$ is preserved under union of finalist intersective reference sets.*

Proof. Assume w.l.o.g. that the reference sets $X_1, \ldots, X_n$ are finalist intersective. By the previous lemma, all finalists of each X_i, $1 \leq i \leq n$, belong to $\bigcap_{i=1}^{n} X_i$. So none of them are outranked by any member of $X_j \setminus \bigcap_{i=1}^{n} X_i$ for all $1 \leq j \leq n$ (an easy proof by contradiction is sufficient to establish this). Thus no member of $\bigcup_{i=1}^{n} X_i \setminus \bigcap_{i=1}^{n} X_i$ can be an optimal output candidate for any $x \in \bigcup_{i=1}^{n} X_i$, whence $x\tau$ is unaffected by extending $x\textsc{Gen}$ to $\bigcup_{i=1}^{n} X_i$ and global optimality is preserved. □

Since maximal blossoms are disjoint, it follows that if an OS is finalist intersective, an equivalent output-partitioned OS can be obtained by taking unions of finalist intersective reference sets. Adding to this the observation that γ can be assumed to be one-to-one, we derive that every finalist intersective OS has an equivalent output joint preserving OS.

Corollary 31. *For every finalist intersective OS* $\mathcal{O}[\mathscr{F}] := \langle \textsc{Gen}, C, \mathscr{F}, \gamma \rangle$ *there exists an output joint preserving OS* $\mathcal{O}[\mathscr{F}]' := \langle \textsc{Gen}, C, \mathscr{F}', \gamma' \rangle$ *that defines the same transduction.*

5 Focus Economy

Judging from the general results about OSs it seems very likely that many reference-set constraints can be implemented by linear transducers and hence are efficiently computable. We already established that output joint preservation is usually satisfied, and the same goes for type-level optimality. The crucial question, then, is whether Gen and the constraints can be computed by linear transducers. In the remainder of this paper, I demonstrate that this is true for three popular constraints, starting with Focus Economy.

5.1 Focus Economy Explained

Focus Economy (Szendrői 2001; Reinhart 2006) was briefly discussed in the introduction. It is invoked in order to account for the fact that sentences such as (2a), (2b) and (2c) below differ with respect to what is given and what is new information. Once again main stress is marked by **boldface**.

(2) a. My friend Paul bought a new **car**.
b. My friend Paul **bought** a new car.
c. My friend Paul bought a **new** car.

That these utterances are associated to different information structures is witnessed by the following (in)felicity judgments. For the reader's convenience, the focus, i.e. the new discourse material introduced by each answer, is put in square brackets.

(3) What happened?
a. [$_F$My friend Paul bought a red **car**.]
b. # [$_F$My friend Paul **bought** a red car.]
c. # [$_f$My friend Paul bought a **red** car.]

(4) What did your friend Paul do?
a. He [$_F$ bought a red **car**].
b. # He [$_F$ **bought** a red car].
c. # He [$_F$ bought a **red** car].

(5) What did your friend Paul buy?
a. He bought [$_F$ a red **car**].

b. # He **bought** [$_F$ a red car].
c. # He bought [$_F$ a **red** car].

(6) Did your friend Paul sell a red car?
a. # No, he [$_F$ bought] a red **car**.
b. No, he [$_F$ **bought**] a red car.
c. # No, he [$_F$ bought] a **red** car.

(7) Did your friend Paul buy a green car?
a. # No, he bought a [$_F$ red] **car**.
b. # No, he **bought** a [$_F$ red] car.
c. He bought a [$_F$ **red**] car.

Restricting our attention to the a-sentences only, we might conclude that a constituent can be focused just in case one of its subconstituents carries sentential main stress. A short glimpse at the b- and c-utterances falsifies this conjecture, though. Perhaps, then, main stress has to fall on the subconstituent at the right edge of the focused constituent? This is easily shown to be wrong, too. In (8) below, the stressed constituent isn't located at either edge of the focused constituent.

(8) a. What happened to Mary?
b. [$_F$ John **killed** her.]

The full-blown Focus Economy system (rather than the simplified sketch given in the introduction) accounts for the data as follows. First, the *Main Stress Rule* demands that in every pair of sister nodes, the "syntactically more embedded" node (Reinhart 2006:p.133) is assigned strong stress, its sister weak stress (marked in the phrase structure tree by subscripted S and W, respectively). If a node has no sister, it is always assigned strong stress (in Minimalist syntax, this will be the case only for the root node, as all Minimalist trees are strictly binary branching). Main stress then falls on the unique leaf node that is connected to the root node by a path of nodes that have an S-subscript. See Fig. 5 for an example.

The notion of being syntactically more embedded isn't explicitly defined in the literature. It is stated in passing, though, that "... main stress falls on the most embedded constituent on the recursive side of the tree" (Reinhart 2006:p.133). While this is rather vague, it is presumably meant to convey that — at least for English, in which complements follow the heads they are introduced by — the right sister node is assigned strong stress as long as it isn't an adjunct. This interpretation seems to be in line with the empirical facts.

The second integral part of the proposal is the operation *Stress Shift*, which shifts the main stress to some leaf node n by assigning all nodes on the path from n to the root strong stress and demoting the sisters of these nodes to weakly stressed nodes. For instance, the tree for "My **friend** Paul bought a new red car" is obtained from the tree in Fig. 5 by changing friend$_W$ and Paul$_S$ to friend$_S$ and Paul$_W$, respectively, and DP$_W$ and T$'_S$ to DP$_S$ and T$'_W$, respectively.

While Stress Shift could be invoked to move stress from anaphoric elements to their left sister as in (8), this burden is put on a separate rule, for independent

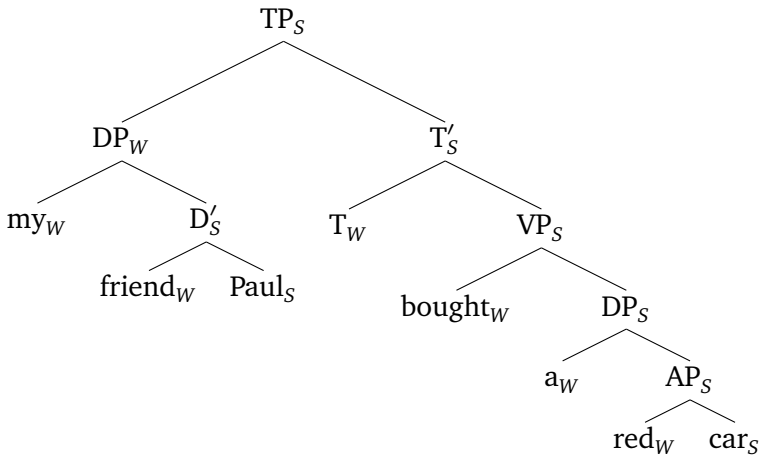


Figure 5: The stress-annotated phrase structure tree for (2a)

reasons. The rule in question is called *Anaphoric Destressing* and obligatorily assigns weak stress to anaphoric nodes, where a node is anaphoric iff it is "...D[iscourse]-linked to an accessible discourse entity" (Reinhart 2006:p.147). Thus Anaphoric Destressing not only accounts for the unstressed anaphor in (8), but also for the special behavior of stress in cases of parallelism.

(9) First Paul bought a red **car**.
   a. Then **John** bought one.
   b. * Then John bought **one**.

The overall system now works as follows. Given a phrase structure tree that has not been annotated for stress yet, one first applies Anaphoric Destressing to make sure that all d-linked constituents are assigned weak stress and thus cannot carry main stress. Next the Main Stress Rule is invoked to assign every node in the tree either W or S. Note that the Main Stress Rule cannot overwrite previously assigned labels, so if some node n has been labeled W by Anaphoric Destressing, the Main Stress Rule has to assign S to the sister of n. Now that the tree is fully annotated, we compute its *focus set*, the set of constituents that may be focused.

(10) *Focus Projection*
Given some stress-annotated tree t, its focus set is the set of nodes reflexively dominating the node carrying main stress.

The focus set of "Paul bought a red **car**", for instance, contains [car], [.AP red car], [.DP a red car], [.VP bought a red car] and [.TP Paul bought a red car] (equivalently, we could associate every node in the tree with a unique address and simply use these addresses in the focus set). For "Then **John** bought one", on the other hand, it consists only of [John] and [.TP Then John bought one].

At this point, Stress Shift may optionally take place. After the main stress has been shifted, however, the focus set has to be computed all over again, and this time the procedure involves reference-set computation.

(11) *Focus Projection Redux*
Given some stress-annotated tree t' that was obtained from tree t by Stress Shift, the focus set of t' contains all the nodes reflexively dominating the node carrying main stress which aren't already contained in the focus set of t.

So if "Then **John** bought one" had been obtained by Stress Shift from [.TP Then John bought one] rather than Anaphoric Destressing, its focus set would have contained only [John], because [.TP Then John bought one] already belongs to the focus set of "Then John bought **one**". As an easy exercise, the reader may want to draw annotated trees for the examples in (2) and compute their focus sets.

5.2 A Model of Focus Economy

After this general overview, we may attempt to formalize Focus Economy. In order to precisely model Focus Economy, though, I have to make some simplifying assumptions, for reasons that are entirely independent from the restrictions of OSs. First, I stipulate that adjuncts are explicitly marked as such by a subscript A on their label. This is simply a matter of convenience, as it reduces the complexity of the transducers and makes my model independent from the theoretical status of adjuncts in syntax.

Second, I decided to take movement out of the picture, because the interaction of focus and movement is not touched upon in Reinhart (2006), so there is no original material to formalize. Incidentally, movement seems to introduce several interesting complications, as illustrated by sentences involving topicalization, where no other assignment of focus and main stress is grammatical.

(12) a. [$_F$ **John**$_i$] Paul likes t$_i$.
b. * John$_i$ [$_F$**Paul**] likes t$_i$.
c. * John$_i$ [$_F$ Paul **likes** t$_i$].

At the end of the section I argue that the model can be extended to capture theories involving movement.

The last simplification concerns Anaphoric Destressing itself. While the core of Anaphoric Destressing, the destressing of pronominal (and possibly covert) elements, is easy to accommodate, the more general notion of d-linking is impossible to capture in any model that operates on isolated syntactic trees. Devising a working model of discourse structure vastly exceeds the scope of this contribution. Also, the role of d-linking in anaphoric destressing is of little importance to this paper, which focuses on the reference-set computational aspects of Focus Economy. Thus my implementation will allow almost any constituent to be anaphorically destressed and leave the task of matching trees to appropriate discourse contexts to an external theory of d-linking that remains to be specified.

With these provisions made explicit, the formalization of Focus Economy as a controlled OS can commence. The input language is supposedly derived by some movement-free MG $\mathscr{E}$ for English (Stabler and Keenan 2003) in which interior nodes

are given explicit category labels (once more for the sake of convenience). As MGs without remnant movement generate regular tree languages (Kobele 2010), it is safe to assume that MGs without any kind of movement do so, too.

Next I define GEN as the composition of four linear transducers corresponding to Anaphoric Destressing, the Main Stress Rule, Stress Shift, and Focus Projection, respectively. Given a tree t derived by $\mathscr{E}$, the transducer cascade computes all logically possible variants of t with respect to stress assignment and then computes the focus in a local way. This means that GEN actually overgenerates with respect to focus, a problem that we have to take care of at a later step.

Anaphoric Destressing is modeled by a non-deterministic ltdtt that may randomly add a subscript D to a node's label in order to mark it as anaphoric. The only condition is that if a node is labeled as anaphoric, all the nodes it properly dominates must be marked as such, too.

Definition 32. Let $\Sigma := \Sigma_L \cup \Sigma_A$ be the vocabulary of the MG $\mathscr{E}$ that generated the input language, where Σ_L contains all lexical items and category labels and Σ_A their counterparts explicitly labeled as adjuncts. *Anaphoric Destressing* is the ltdtt $\mathscr{D}$ where $\Sigma_{\mathscr{D}} := \Sigma$, $\Omega_{\mathscr{D}}$ is the union of Σ and $\Sigma_D := \{\sigma_D \mid \sigma \in \Sigma\}$, $Q := \{q_i, q_d\}$, $Q' := \{q_i\}$, and $\Delta_{\mathscr{D}}$ contains the rules below, with $\sigma \in \Sigma$ and $\sigma_D \in \Sigma_D$:

$$q_i(\sigma(x,y)) \to \sigma(q_i(x), q_i(y)) \qquad q_i(\sigma) \to \sigma$$
$$q_{\{i,d\}}(\sigma(x,y)) \to \sigma_D(q_d(x), q_d(y)) \qquad q_{\{i,d\}}(\sigma) \to \sigma_D$$

The transducer for the Main Stress Rule is non-deterministic, too, but it proceeds in a bottom-up manner. It does not alter nodes subscripted by A or D, but if it encounters a leaf node without a subscript, it randomly adds the subscript S or W to its label. However, W is allowed to occur only on left sisters, whereas S is mostly restricted to right sisters and may surface on a left sister just in case the right sister is already marked by A or D. Note that we could easily define a different stress pattern, maybe even parametrized with respect to category labels, to incorporate stress assignment rules from other languages.

Definition 33. *Main Stress* is the lbutt $\mathscr{M}$ where $\Sigma_{\mathscr{M}} := \Omega_{\mathscr{D}}$, $\Omega_{\mathscr{M}}$ is the union of Σ, Σ_D and $\Sigma_* := \{\sigma_S, \sigma_W \mid \sigma \in \Sigma\}$, $Q := \{q_s, q_u, q_w\}$, $Q' := \{q_s\}$ and $\Delta_{\mathscr{M}}$ contains the following rules, with $\sigma \in \Sigma$, $\sigma_A \in \Sigma_A$, $\sigma_x \in \{\sigma_x \mid \sigma \in \Sigma\}$ for $x \in \{D, S, W\}$:

$$\sigma_A \to q_u(\sigma_A) \qquad \sigma_A(q_u(x), q_u(y)) \to q_u(\sigma_A(x,y))$$
$$\sigma_D \to q_u(\sigma_D) \qquad \sigma_D(q_u(x), q_u(y)) \to q_u(\sigma_D(x,y))$$
$$\sigma \to q_{sw}(\sigma_{SW}) \qquad \sigma(q_{\{u,w\}}(x), q_s(y)) \to q_{sw}(\sigma_{SW}(x,y))$$
$$\sigma(q_s(x), q_u(y)) \to q_{sw}(\sigma_{SW}(x,y))$$

Stress Shift is best implemented as a non-deterministic ltdtt that may randomly switch the subscripts of two S/W-annotated sisters.

Definition 34. *Stress Shift* is the ltdtt $\mathscr{S}$ where $\Sigma_{\mathscr{S}} = \Omega_{\mathscr{S}} = \Omega_{\mathscr{M}}$, $Q := \{q_i, q_s, q_w\}$,

$Q' := \{q_s\}$, and $\Delta_{\mathscr{S}}$ contains the rules below, with $\sigma \in \Sigma_{\mathscr{S}}$ and $\sigma_* \in \Sigma_*$:

$$\begin{aligned} q_s(\sigma_*(x,y)) &\to \sigma_{SSS}(q_{isw}(x), q_{iws}(y)) & \qquad q_s(\sigma_*) &\to \sigma_S \\ q_w(\sigma_*(x,y)) &\to \sigma_W(q_i(x), q_i(y)) & \qquad q_w(\sigma_*) &\to \sigma_W \\ q_i(\sigma(x,y)) &\to \sigma(q_i(x), q_i(y)) & \qquad q_i(\sigma) &\to \sigma \end{aligned}$$

The last component is Focus Projection, which is formalized as a non-deterministic ltdtt with two states, q_f and q_g. The transducer starts at the root in q_f. Whenever a node n is subscripted by W, the transducer switches into q_g at this node and stays in the state for all nodes dominated by n. As long as the transducer is in q_f, it may randomly add a superscript F to a label to indicate that it is focused. Right afterward, it changes into q_g and never leaves this state again. Rather than associating a stress-annotated tree with a set of constituents that can be focused, Focus Projection now generates multiple trees that differ only with respect to which constituent along the path of S-labeled nodes is focus-marked.

Definition 35. *Focus Projection* is the ltdtt $\mathscr{F}$, where $\Sigma_{\mathscr{F}} = \Omega_{\mathscr{S}}$, $\Omega_{\mathscr{F}}$ is the union of $\Omega_{\mathscr{S}}$ and $\Omega^F_{\mathscr{S}} := \{\omega^F \mid \omega \in \Omega_{\mathscr{S}}\}$, $Q := \{q_f, q_g\}$, $Q' := \{q_f\}$, and $\Delta_{\mathscr{F}}$ contains the rules below, with $\sigma \in \Sigma_{\mathscr{F}}$ and $\sigma_{\overline{S}} \in \Sigma_{\mathscr{F}} \setminus \{\sigma_S \mid \sigma \in \Sigma\}$:

$$\begin{aligned} q_f(\sigma_S(x,y)) &\to \sigma_S(q_f(x), q_f(y)) \\ q_f(\sigma_S(x,y)) &\to \sigma^F_S(q_g(x), q_g(x)) & \qquad q_f(\sigma_S) &\to \sigma^F_S \\ q_f(\sigma_{\overline{S}}(x,y)) &\to \sigma_{\overline{S}}(q_g(x), q_g(x)) & \qquad q_f(\sigma_{\overline{S}}) &\to \sigma_{\overline{S}} \\ q_g(\sigma(x,y)) &\to \sigma(q_g(x), q_g(y)) & \qquad q_g(\sigma) &\to \sigma \end{aligned}$$

All four transducers are linear, so they can be composed into a single lbutt modelling Gen (see Engelfriet 1975 for a constructive proof that every ltdtt can be converted into an lbutt defining the same transduction). Expanding on what was said above about the inner workings of Gen, we now see that for any tree t in the input language, tGen is the set of stress-annotated trees in which, first, some subtrees may be marked as adjuncts or anaphorical material (or both) and thus do not carry stress information, second, there is exactly one path from the root to some leaf such that every node in the path is labeled by S, and third, exactly one node belonging to this path is marked as focused. The reader should have no problem verifying that in terms of controlled OSs, all reference types are singleton and their reference-sets do not overlap, i.e. output joint preservation and type-level optimality are satisfied.

Now it only remains for us to implement *Focus Projection Redux*. In the original account, Focus Projection Redux applied directly to the output of Stress Shift, i.e. trees without focus information, and the task at hand was to assign the correct focus. In my system, on the other hand, every tree is fed into Focus Projection and marked accordingly for focus. This leads to overgeneration for trees in which Stress Shift has taken place — a node may carry focus even if it could also do so in the tree without shifted main stress. Consequently, the focus set of "**John** died", for instance, turns out to contain both [John] and [.TP John died] rather than just the former. Under my proposal, then, Focus Projection Redux is faced with the burden of filtering out focus information instead of assigning it. In other words, Focus Projection Redux is a constraint.

This is accomplished by defining a regular tree language L_c such that when GEN is composed with the diagonal of L_c (which is guaranteed to be a linear transduction), only trees with licit focus marking are preserved. Said regular language is easily specified in the monadic second-order logic $\mathscr{L}^2_{K,P}$ (Rogers 1998). First one defines two predicates, *StressPath* and *FocusPath*. The former picks out the path from the root to the leaf carrying main stress, whereas the latter refers to the path from the root to the leaf that would carry main stress in the absence of stress shift. This implies that *FocusPath* replicates some of the information that is already encoded in the Main Stress transducer. Note that in the definitions below, $A(x)$, $D(x)$ and $S(x)$ are predicates picking out all nodes with subscript A, D, S, respectively, $x \triangleleft y$ denotes "x is the parent of y", $x \prec y$ "x is the left sibling of y", and $\triangleleft^*$ the reflexive transitive closure of $\triangleleft$.

$$\text{Path}(X) \longleftrightarrow \exists x \Big[X(x) \wedge \neg \exists y [y \triangleleft x] \Big] \wedge \exists! x \Big[X(x) \wedge \neg \exists y [x \triangleleft y] \Big] \wedge$$
$$\forall x, y, z \Big[\big(X(x) \wedge X(y) \rightarrow x \triangleleft^* y \vee y \triangleleft^* x \big) \wedge \big(X(x) \wedge \neg X(z) \rightarrow \neg (z \triangleleft^* x) \big) \Big]$$

$$\text{StressPath}(X) \longleftrightarrow \text{Path}(X) \wedge \forall x [X(x) \rightarrow S(x)]$$

$$\text{FocusPath}(X) \longleftrightarrow \text{Path}(X) \wedge \forall x, y, z \Big[X(x) \wedge x \triangleleft y \wedge x \triangleleft z \rightarrow$$
$$\big((A(y) \vee D(y)) \rightarrow X(z) \big) \wedge \big(\neg A(y) \wedge \neg D(y) \wedge y \prec z \rightarrow X(z) \big) \Big]$$

In a tree where no stress shift has taken place, StressPath and FocusPath are true of the same subsets and any node contained by them may be focused. After an application of the Stress Shift rule, however, the two paths are no longer identical, although their intersection is never empty (it has to contain at least the root node). In this case, then, the only valid targets for focus are those nodes of the StressPath that aren't contained in the FocusPath. This is formally expressed by the $\mathscr{L}^2_{K,P}$ sentence ϕ below. Just like $A(x)$, $D(x)$ and $S(x)$ before, $F(x)$ is a predicate defining a particular set of nodes, this time the set of nodes labeled by some $\omega^F \in \Omega^F_{\mathscr{S}}$. I furthermore use $X \approx Y$ as a shorthand for $\forall x [X(x) \longleftrightarrow Y(x)]$.

$$\phi := \forall x, X, Y [F(x) \wedge X(x) \wedge \text{StressPath}(X) \wedge \text{FocusPath}(Y) \rightarrow (Y(x) \rightarrow X \approx Y)]$$

Note that ϕ by itself does not properly restrict the distribution of focus. First of all, there is no requirement that exactly one node must be focused. Second, nodes outside StressPath may carry focus, in which case no restrictions apply to them at all. Finally, StressPath and FocusPath may be empty, because we have not made any assumptions about the distribution of labels. Crucially, though, ϕ behaves as expected over the trees in the candidate language. Thus taking the diagonal of the language licensed by ϕ and composing it with GEN filters out all illicit foci, and only those. Since the diagonal over a regular language is a linear transduction, the transduction obtained by the composition is too. This establishes the computational feasibility of Focus Economy when the input language is a regular tree language. That is, Focus Economy preserves the regularity of the input language.

So far I have left open the question, though, how movement fits into the picture. First of all, it cannot be ruled out *a priori* that the interaction of movement and focus are so intricate on a linguistic level that significant modifications have to be made to the original version of Focus Economy. On a formal level, this would mean that the transduction itself would have to be changed. In this case, it makes little sense to speculate how my model could be extended to accommodate movement, so let us instead assume that Focus Economy can remain virtually unaltered and it is only the input language that has to be modified. In my model, the input language is a regular tree language by virtue of being generated by an MG without movement. But note that MGs with movement generate regular tree languages, too, in the presence of a ban against more exotic kinds of movement such as remnant movement or head movement (Kobele 2010). Now keep in mind that regular tree languages yield context-free string languages, which are generally assumed to be powerful enough for the greatest part of natural language syntax. Thus the restriction to regular tree languages itself does not preclude us from accommodating most instances of movement.

If we want the full expressive power of MGs, then the best strategy is to express Focus Economy as a constraint over derivation trees, since for every MG the set of derivation trees it licenses forms a regular language that fully determines the tree yield of the grammar (Kobele et al. 2007). The only difference between Minimalist derivation trees and movement-free phrase structure trees as derived above is that the latter are unordered. Hence, if we require that linear order (which can be easily determined from the labels of the leaves) is directly reflected in the derivation trees, the formalization above carries over unaltered to derivation trees and may be extended as desired to deal with instances of movement. One possible obstacle for taking this route though is that even though regular languages are closed under linear transductions, it is still an open problem whether the derivation tree languages of an MG are, too. If they weren't, then applying a linear transduction would still yield a regular language, but not necessarily a well-formed derivation language.

At least for Focus Economy, though, closure under linear transductions may be more than we actually need. First, notice that the transducers the composition of which makes up GEN are very simple non-deterministic finite-state relabelings. Now we shouldn't expect the derivation tree languages of MGs to be closed under finite-state relabelings, despite their simplicity (the IO-context-free tree languages, for instance, aren't closed under these relabelings). However, when we consider the form of the trees in the range of GEN, it seems fairly unlikely that it couldn't be obtained directly by an MG. The only conditions are that there is a unique path of S-labeled nodes and that one node in this path is also marked for focus. Things are complicated slightly by the special status of adjuncts and anaphorically destressed nodes, but overall we are dealing with very simple conditions that a MG should have no problem with. The truly problematic question, then, is whether the tree languages of MGs are closed under intersection with a regular language, i.e. whether we can apply the filtering step to tame the overgeneration inherent to GEN. To my knowledge, this is an open problem, too, although the answer might already be implicit in the automata-theoretic perspective on MGs of Kobele et al. (2007) or the

two-step approach of Kolb et al. (2003).

6 Merge-over-Move

Another well-known reference-set constraint is Chomsky's Merge-over-Move condition (MOM; Chomsky 1995b, 2000), which is the subject of inquiry in this section. After a short discussion of the mechanics of the constraint and its empirical motivation, I turn to the formal aspects of implementing MOM. In spite of the fact that MOM is what we might now — using the terminology previously introduced for OSs — call an endocentric, i.e. Fewest Steps-like constraint, the procedure for devising a MOM transducer exhibits many parallels to Focus Economy. I take this as further support of my earlier claim that both kinds of constraints can be naturally studied in the framework of OSs and tree transducers.

6.1 Merge-over-Move Explained

In comparison to Focus Economy, modelling MOM is slightly more intricate, because there are multiple versions of the constraint, which are seldom carefully teased apart in the literature. Naturally they all share the core idea of MOM: if at some point in a derivation we are allowed to choose between Merge and Move as the next step of the derivation, Merge is preferable to Move. This idea can be used to account for some puzzling contrasts involving expletives (if not indicated otherwise, all examples are taken from Castillo, Drury, and Grohmann 2009).

(13) a. There seems to be a man in the garden.
b. * There seems a man to be in the garden.
c. A man seems to be in the garden.

Recall that in an MG in Chomsky's sense, we start out with a multiset of lexical items — the *numeration* — that are enriched with interpretable and uninterpretable features, the latter of which have to be erased by the operation of feature checking. Under such a conception, (13a) and (13c) are easy to derive. Let us look at (13c) first. It starts out with the numeration {seems, to, be, a, man, in, the, garden}. Multiple applications of Merge yield the phrase [$_{\text{TP}}$ to be a man in the garden]]. At this point, the Extended Projection Principle (EPP) demands that the specifier of the infinitival TP be filled by some phrase. The only item left in the numeration is *seems*, which cannot be merged in SpecTP. Hence we are stuck with moving the DP *a man* into SpecTP, yielding [$_{\text{TP}}$ a man to be t_{DP} in the garden]. Afterwards, the TP is merged with *seems* and the DP is once again moved, this time into the specifier of *seems* to check the case feature of the DP and satisfy the EPP.

For (13a), however, things are slightly different. Here the numeration initially consists of {there, seems, to, be, a, man, in, the, garden}. Once again we start out by merging items from the numeration until we arrive at [$_{\text{TP}}$ to be [$_{\text{DP}}$ a man in the garden]]. But now we have two options: Merger of *there*, which is later followed by moving *there* into the specifier of *seems*, thus yielding the grammatical (13a), or first

moving *a man* into the specifier of *to be* and subsequently merging *there* with *seems a man to be in the garden*, which incorrectly produces the ungrammatical (13b). MOM rectifies this overgeneration problem by barring movement of *a man* into the specifier of *to be*, as the more economical route of merging *there* in this position is available to us. At the same time, MOM does not block (13c) because we aren't given a choice between Merge and Move at any point of its derivation.

Different versions of MOM emerge depending on the setting of two binary parameters:

P1 Reference set algorithm: *indiscriminate*/*cautious*
Indiscriminate versions of MOM (iMOM) pick the most economical derivation even if it derives an ungrammatical phrase structure tree — such derivations are said to *crash*. Cautious versions of MOM (cMOM), on the other hand, picks the most economical derivation that yields a well-formed tree.[2]

P2 Mode of application: *sequential*/*output*
Sequential versions of MOM (sMOM) check for MOM violations after every step of the derivation. Thus early violations of MOM carry a significantly greater penalty than later ones.[3] MOM applied to the output (oMOM), however, is sensitive only to the total number of violations, not their timing. So if derivation d incurs only one violation of MOM, which occurs at step 4 in the derivation, while derivation d' incurs seven, starting at step 5, then d will win against d' under an output filter interpretation of MOM and lose under a sequential one.

Combining the parameters in all logically possible ways (*modulo* underspecification) yields the four variants isMOM, csMOM, ioMOM and coMOM. All four of them supposedly use the *Identity of Numerations Condition* (INC) for computing reference sets, according to which the reference set of a derivation d contains all the derivations that can be built from the same numeration as d.[4] "Supposedly", because only the sMOM variants have been discussed at length in the literature. The original proposal by Chomsky (1995b) is what I call csMOM. But the global flavor of csMOM (if MOM is evaluated at every step of the derivation, i.e. before the derivation

[2]If we include crashing derivations in the reference set, however, we face the problem that the empty derivation, or a derivation that never moves anything, will always be the most grammatical option. The condition would have to be strengthened such that one may only consider derivations that obey all syntactic principles up to the first choice between Merge and Move, at which point they may later run into irreparable problems. The technical details are presumably much more complicated than that, but fortunately they are of little importance given my objectives. I will simply adopt the tentative assumption in the literature that only "reasonable" alternatives are in the reference set.

[3]This raises several thorny issues concerning the notion of derivational earliness, as derivations are usually partial rather than total strict orders. Fortunately these complications do not surface in the cases MOM was designed to account for, so I will happily ignore them. But in grammars with sidewards-movement this issue needs to be properly addressed (Drummond 2010).

[4]The astute reader may rightfully point out that the INC is both too weak and too strong. On the one hand, it erroneously allows *John likes Mary* to compete against *Mary likes John*, yet on the other hand it seems to block competition between candidates that differ only in their feature make-up, even if only marginally so. This observation is correct and highlights a problem in the specification of MOM's reference-set algorithm that has frequently been lamented in the literature (Sternefeld 1996). As it turns out, though, these formal problems, as well as certain empirical quandaries I will discuss later on, do not arise in a tree transducer model of MOM.

is completed, how does it know which competing derivations will crash later on and thus may be discarded for the comparison?) prompted the creation of isMOM as a strictly local alternative. Indeed isMOM can be argued to contain not even a modicum of reference-set computation, as it simply states that if there is a choice between Merge and Move, pick Merge. Whether such a choice exists can always be checked locally.

For simple cases like (13), where we only have to choose once between Merge and Move, all MOM variants produce the same results (although evaluation of iMOM variants is complicated by the open question which degree of deviancy one wants to allow for competing derivations; see my remarks in fn. 2). But as soon as we encounter examples involving embedded clauses, the predictions diverge (which was already noted as early as 1997 by Wilder and Gärtner).

(14) a. There was [a rumor [that a man was t_{DP} in the room]] in the air.
b. [A rumor [that there was a man in the room]] was t_{DP} in the air.

Both oMOM-versions get the right result: Each sentence prefers Move over Merge exactly once, so assuming that there are no (grammatical) competing derivations that start from the same numeration and incur fewer violations, (14a) and (14b) should both be grammatical. The sMOM variants, on the other hand, struggle with this data. The sentences are built up from the same numeration, so (14b) should block (14a), since the former violates MOM at a later derivational step than the latter. In order to account for such cases, Chomsky (2000) stratifies numerations into subnumerations such that each CP has its own numeration (which is extended in Chomsky 2001 to the contemporary phase system). In the case at hand, (14a) is built from the numeration {{there, was, a, rumor, in, the, air}, {that, was, a, man, in, the, room}}, and (14b) from the minimally different {{was, a, rumor, in, the, air}, {that, there, was, a, man, in, the, room}}. By the INC, then, derivations built from the former do not belong to the same reference set as derivations built from the latter.

So now we have a third parameter to take into account. Even though it isn't directly related to the makeup of MOM, I will indicate it as a prefix as before.

P3 Application domain: *restricted*/*unbounded*
Restricted versions of MOM (rMOM) are parts of a grammar where every CP has its own numeration. Unbounded versions (uMOM) belong to grammars with one big numeration.

Taking stock, we have csuMOM as the version of MOM introduced in Chomsky (1995b), isuMOM as its local counterpart, and csrMOM as the modification put forward in Chomsky (2000). Somewhat surprisingly, no oMOM variants are entertained in the literature, despite the small empirical advantage they have displayed so far. For this reason, I shall mostly restrict myself to sMOM variants in the following.

6.2 *Properties of Merge-over-Move*

Let us step back for a second to take in the architecture of MOM on a broad scale, in terms of OSs. If one abstracts away from the peculiarities of the application

mode, MOM merely acts as a filter on the tree language derived by a grammar; the trees don't have to be manipulated at all, as was the case with Focus Economy. It follows that MOM is endocentric and thus output joint preserving. Moreover, reference sets presumably do not overlap, at least not if we take the identity of numerations requirement literally, whence MOM is also output partitioned. Finally, the evaluation metric is independent of the input — for oMOM, that is. With sMOM we run into a problem. Whether a step was taken earlier in comparison to other candidates cannot simply be read off the total number of violations. Instead, the evaluation itself has to proceed in a bottom up fashion, weeding out candidates with unnecessary instances of Move after every derivational step. It seems, then, that the metric involves genuine reference-set computation which requires us to consider multiple tree structures at once. However, remember that we faced a similar problem with Focus Economy, where the constraint as it was stated in the literature relied on comparing two focus-annotated trees to determine licit foci. Our answer to this problem was to represent the competing trees within a single tree (using the FocusPath and StressPath predicates) and thus emulate the comparative procedures by well-formedness constraints on this one underlying tree. If we could find a similar way of representing competing derivations within one derivation tree, a major hurdle would be out of the way.

But there is yet another problem, and this one pertains to sMOM as well as oMOM: the INC; it is impossible for a linear tree transducer to define the corresponding partition over the input language. The major culprit here is the restriction to finite memory, which entails that we can only distinguish between a bounded number of occurrences of lexical items. For some suitably large n, the multiset M' obtained from $M := \{\text{John}_n, \text{thinks}_n, \text{that}_n, \text{Mary died}\}$ by adding one more occurrence of *John*, *thinks*, and *that* will be indistinguishable from M for the transducer. Thus the challenges surrounding definability by transducers extend from the evaluation metric directly to GEN. And so does the solution.

6.3 A Model of sMOM

The INC is both too powerful and too weak. Consider (13) again, repeated here for the reader's convenience.

(15) a. There seems to be a man in the garden.
b. * There seems a man to be in the garden.
c. A man seems to be in the garden.

MOM's objective is to explain why (15b) is ungrammatical, and it does so by using a metric that makes it lose out against (15a). The grammaticality of (15c), on the other hand, follows from the fact that it isn't a member of the same reference set, due to the INC. But identity of numerations is a rather indirect encoding of the relationship that holds between (15a) and (15b). A formally simpler condition emerges when we look at their derivation trees (cf. Fig. 6 on page 35). Ignoring the feature specifications of the lexical items, we see that the only difference between the respective derivation trees is the timing of move. Rather than a transducer

modelling the INC, then, all we need is a transducer that will produce (at least) the derivation trees for (15a) and (15b) when given either as an input. This involves merely changing the position of the unary branch, which is an easy task for a linear transducer. But now compare these derivations to the one for (15c) in Fig. 7 on page 36. The derivation trees of (15a) and (15b) are essentially the result of non-deterministically replacing one instance of move in the derivation tree of (15c) by merger with expletive *there*. Strikingly, though, rewriting the lower occurrence of O yields the grammatical (15a), whereas rewriting the structurally higher occurrence gives rise to the ungrammatical (15b). Now if we design the transducer such that it won't rewrite an O as a *there*-merger after it has already passed up on an opportunity to do so earlier in the derivation, (15b) cannot be generated from the derivation tree of (15c). In linguistic parlance, this is tantamount to treating MOM as a well-formedness condition on derivation trees (note the similarity to the Focus Economy strategy).

The idea just outlined is captured as follows: First, we take as our input language I the set of derivation trees of an MG that generates the intended derivation trees. Then, we use a transducer α to map this language to a set U of underspecified derivation trees. The transducer strips away all features from the lexical items, deletes expletive *there* and rewrites O as O/there in the TP-domain. The underspecified representation of (15a)–(15c), for instance, is almost identical to the derivation tree of (15c) except that the two O nodes are now labeled O/there (and the lexical items are devoid of any features). These underspecified representations are then turned into fully specified representations again by the transducer β. It reinstantiates the features on the lexical items and non-deterministically rewrites O/there as O or Merger of *there*, but with the added condition that once an O/there node has been replaced by an O, all remaining instance of O/there in the same CP have to be rewritten as O. I call the output language of the second transduction J. The name is meant to be a shorthand for *junk*, as J will contain a lot thereof, for two independent reasons. First, the non-deterministic rewriting of O/there allows for two occurrences of O/there to be rewritten as *there*, which yields the (derivation tree of the) ungrammatical *there seems there to be a man in the garden*. Second, the reinstantiation of the features is a one-to-many map that will produce a plethora of illicit derivation trees as some lexical items may not be able to get all their features checked. This overgeneration problem is taken care of by intersecting J with I. The overall structure of the computation, in comparison to Focus economy and the under specification strategy in general, is depicted in Fig. 8 on page 36. Note that in terms of OSs, the actual content of MOM is now squeezed into GEN and the only constraint of the OS is the input language itself.

Just as with Focus Economy, there is some reason for concern as it is still an open question whether Minimalist derivation tree languages are closed under intersection with regular languages (as regular sets are closed under linear transductions, J is guaranteed to be regular, so we only need to know whether $I \cap J$ is a derivation tree language for some MG). For sMOM itself I expect that $I \cap J$ is the derivation tree language of some MG, as it only removes a few derivation trees from I, which could also be achieved by judicious use of the feature calculus.

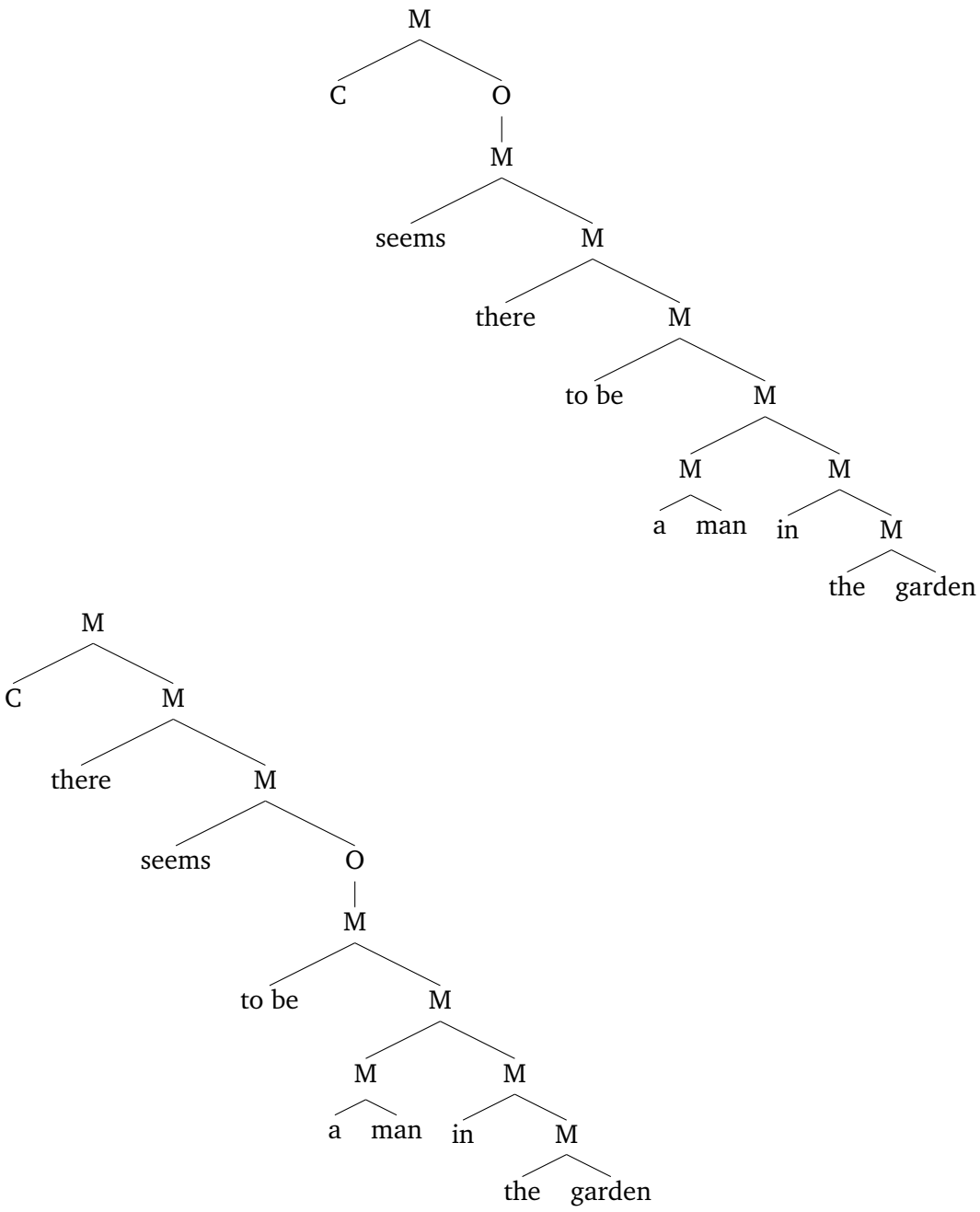


Figure 6: The derivation trees of (15a) and (15b) differ only in the position of the unary branch

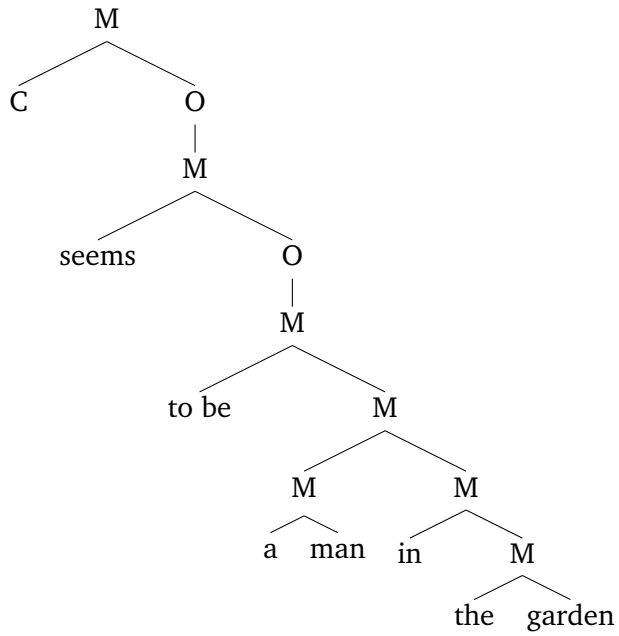


Figure 7: The derivation tree of (15c) can be taken as a basis for the previous two

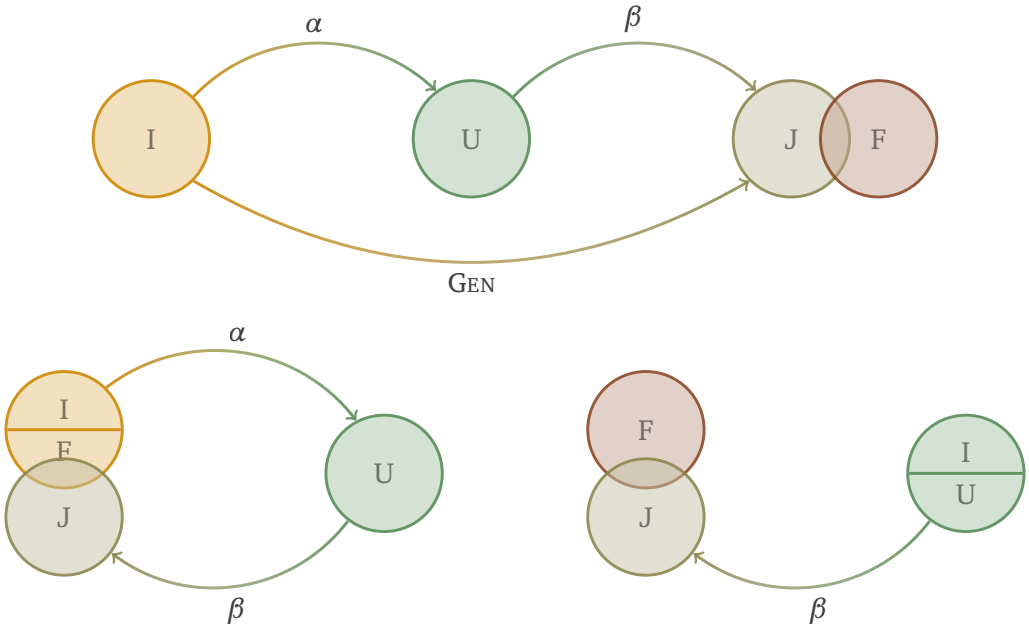


Figure 8: Architecture of the underspecification-and-filtration strategy in general (top), sMOM (left) and Focus Economy (right) in comparison

After these important remarks, let us get started on the low-level implementation of MOM. As mentioned before, I assume that I is given by some MG $\mathscr{E} := \langle \Sigma_{\mathscr{E}}, F_{\mathscr{E}}, Types, Lex_{\mathscr{E}}, O \rangle$. The specifics of $\mathscr{E}$ will be elaborated in the next section, for now I only have to assume that the derivation trees are fully labeled such that leave nodes are decorated by items drawn from $Lex_{\mathscr{E}}$ and unary and binary branching nodes, respectively, by M and O (short for *merge* and *move*). The transduction α is obtained from composing the two transducers *Remove Features* and *Underspecify*.

Definition 36. *Remove Features* is the deterministic (one-state) relabeling that maps each $l := \langle \sigma :: f_1, \ldots, f_{base}, \ldots, f_n \rangle \in Lex_{\mathscr{E}}$ to $l' := \sigma_{f_{base}}$, where f_{base} is the base feature of l. The set of these simplified lexical items is denoted by Λ.

Even though the definition of an MG in Sec. 1 allows for a lexical item to have several base features or none at all, neither option is ever exploited for real-life grammars, so *Remove Features* is well-defined. If for some reason multiple base features (or their absence) are indispensable, the transduction can be extended such that each lexical item is subscripted by the n-tuple of its n base features. In either case, the map defined by *Remove Features* is many-to-one, so Λ is finite by virtue of the finiteness of $Lex_{\mathscr{E}}$.

Definition 37. *Underspecify* is the lbutt $\mathscr{U}$, where $\Sigma_{\mathscr{U}} := \Lambda \cup \{M, O\}$, $\Omega_{\mathscr{U}} := \Sigma_{\mathscr{U}} \cup \{\text{O/there}\}$, $Q := \{q_*, q_c, q_i, q_t\}$, $Q' := \{q_*\}$, and $\Delta_{\mathscr{U}}$ consists of the rules below, where I use the following notational conventions:

- σ_I (σ_C) denotes any lexical item $l \in \Lambda$ whose base feature is I (C),
- the symbol "there" refers to any expletive $l \in \Lambda$ involved in MOM (usually just *there*, but possibly also *it*),
- σ_l denotes any lexical item which doesn't fall into (at least) one of the categories described above,
- as derivation trees aren't linearly ordered, rules for binary branching nodes are given for only one of the two possible orders (namely the one that reflects the linear order in the derived structure).

$$\sigma_l \to q_*(\sigma_l) \qquad \text{M}(q_{c*}(x), q_{i*}(y)) \to q_*(\text{M}(x, y))$$
$$\sigma_I \to q_i(\sigma_I) \qquad \text{M}(q_i(x), q_{\{i,*\}}(y)) \to q_i(\text{M}(x, y))$$
$$\text{there} \to q_t(\text{there}) \qquad \text{M}(q_t(x), q_{\{i,*\}}(y)) \to q_i(\text{O/there}(y))$$
$$\sigma_C \to q_c(\sigma_C) \qquad \text{O}(q_*(x)) \to q_*(\text{O}(x))$$
$$\text{O}(q_i(x)) \to q_i(\text{O/there}(x))$$

The underspecified derivation have to be turned back into fully specified ones by the transduction β, which is the composition of *Path Condition* and the inverse of *Remove Features*.

Definition 38. *Path Condition* is the lbutt $\mathscr{P}$, where $\Sigma_{\mathscr{P}} := \Omega_{\mathscr{U}}$, $\Omega_{\mathscr{P}} := \Sigma_{\mathscr{U}}$, $Q := \{q_*, q_c, q_o\}$, $Q' := \{q_*\}$, and $\Delta_{\mathscr{P}}$ contains the rules below (the same notational conventions apply):

$$\begin{aligned}
\sigma_l &\to q_*(\sigma_l) & \mathrm{M}(q_{c\{c,*\}}(x), q_{o\{c,*\}}(y)) &\to q_*(M(x,y))\\
\sigma_I &\to q_*(\sigma_I) & \mathrm{M}(q_{\{*,o\}}(x), q_o(y)) &\to q_o(\mathrm{M}(x,y))\\
\sigma_C &\to q_c(\sigma_C) & \mathrm{O}(q_*(x)) &\to q_*(\mathrm{O}(x))\\
& & \mathrm{O/there}(q_*(x)) &\to q_*(\mathrm{M}(\mathrm{there}, x))\\
& & \mathrm{O/there}(q_{\{o,*\}}(x)) &\to q_o(\mathrm{O}(x))
\end{aligned}$$

The crucial step toward capturing MOM is the last rule of *Underspecify*, which tells the transducer that after it has rewritten one instance of O/there as O, it has to switch into state q_o, which tells it to always rewrite O/there as O. Only if it encounters a node of category C may the transducer switch back into its normal state q_* again.

We can alternatively restrict α to *Remove Features*, express the composition of *Underspecify* and *Path Condition* as an MSO constraint ψ on the surface language of *Remove Features* applied to the derivation language of $\mathscr{E}$ (since said language is regular), and let β be the inverse of α. All ψ has to do is pick out the path of nodes from a leaf of category I to the closest dominating node that is the mother of a node of category C and stipulate that no node in this path may be both the mother of an expletive and dominate an O-node belonging to the same path. This can easily be made precise.

$$\mathrm{Start}(X,x) \leftrightarrow X(x) \wedge \neg\exists y[X(y) \wedge \neg(x \approx y) \wedge y \lhd^* x]$$

$$\mathrm{End}(X,x) \leftrightarrow X(x) \wedge \neg\exists y[X(y) \wedge \neg(x \approx y) \wedge x \lhd^* y]$$

$$\begin{aligned}
\mathrm{IPdomain}(X) \leftrightarrow \exists!x[\mathrm{Start}(X,x)] \wedge \exists!x[\mathrm{End}(X,x)] \wedge \forall x[\mathrm{End}(X,x) \to I(x)] \wedge\\
\forall x\Big[\mathrm{Start}(X,x) \to \exists y\big[x \lhd y \wedge C(y) \wedge \neg\exists z[y \lhd z]\big]\Big] \wedge\\
\forall x,y,z[X(x) \wedge X(y) \to (x \lhd^* y \vee y \lhd^* x) \wedge (\neg X(z) \to \neg(x \lhd z \wedge z \lhd y)]
\end{aligned}$$

Here the predicates I and C pick out the same lexical items as σ_I and σ_C did before. Similarly, I use the predicate Exp in the statement of ψ to denote expletives. The path condition on derivation trees is then approximated by ψ as follows.

$$\psi := \forall X\Big[\mathrm{IPdomain}(X) \to \neg\exists x\big[X(x) \wedge \exists y[x \lhd y \wedge \mathrm{Exp}(y)] \wedge \exists z[X(z) \wedge O(z) \wedge x \lhd^* z]\big]\Big]$$

As in the case of Focus Economy, ψ by itself does not fully capture the constraint, but over the set of derivation trees of the MG $\mathscr{E}$ it does.

The original transducer architecture (depicted in Fig. 8) differs from the revised version (Fig. 9) in that while both yield the same output language, only the former properly captures the relation between trees established by MOM. In the alternative, no trees of the input are ever related to each other, there is no reference-set algorithm to speak of; instead, it simply enforces a well-formedness condition on derivation

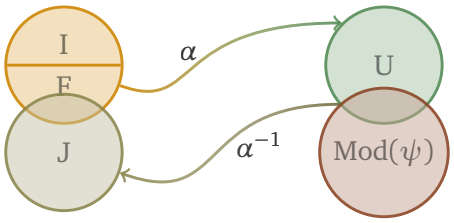


Figure 9: A different perspective on sMOM

trees. In fact, the removal and reinstantiation of features is redundant in this approach. If we proceed as I proposed originally, on the other hand, the result is a relation that will group trees into reference-sets and for each tree in reference-set R return the optimal trees in R as its outputs. One might say that both models capture the *weak relational capacity* of MOM insofar as they yield the correct output language, but only the more elaborate model of Fig. 8 faithfully represents MOM's *strong relational capacity*, the actual transduction.

6.4 *Empirical Evaluation*

As discussed above, the transducer model of MOM accounts for simple expletive/non-expletive alternations as in (15). Instead of going through another iteration of the same basic argument, let us look at a more complex example that we have encountered before, repeated here as (16).

(16) a. There was [a rumor [that a man was t_{DP} in the room]] in the air.
b. [A rumor [that there was a man in the room]] was t_{DP} in the air.

Recall that this was a problematic case for pre-Chomsky (2000) versions of MOM (i.e. csuMOM and isuMOM), because in the absence of stratified numerations the INC puts (16a) and (16b) in the same reference set, where they have to compete against each other. Under a sequential construal of MOM, then, (16a) will block (16b) as it opts for Merge rather than Move at the first opportunity.

Under the transducer conception of MOM (tMOM), on the other hand, (16) is a straightforward generalization of the pattern in (15). The underspecified derivation tree of both sentences is shown in Fig.10. When the underspecified derivation is expanded to full derivations again, all four logical possibilities are available: *there*-insertion in both CPs, Move in both CPs, *there*-insertion in the lower CP and Move in the higher one, and Move in the lower CP and *there*-insertion in the higher one. The last option is available because the transducer, which is in the "rewrite all instances of O/there as O"-state q_o after rewriting the label O/there as O, switches back into the neutral state q_* after encountering the CP headed by *that*. Thus when it encounters the second O/there node in the higher CP, it can once again choose freely how to rewrite it. Provided the four derivation trees obtained from the underspecified derivation aren't filtered out by the MG, they are in turn transformed into the following derived structures, all of which are grammatical:

(17) a. There was a rumor that there was a man in the room in the air.

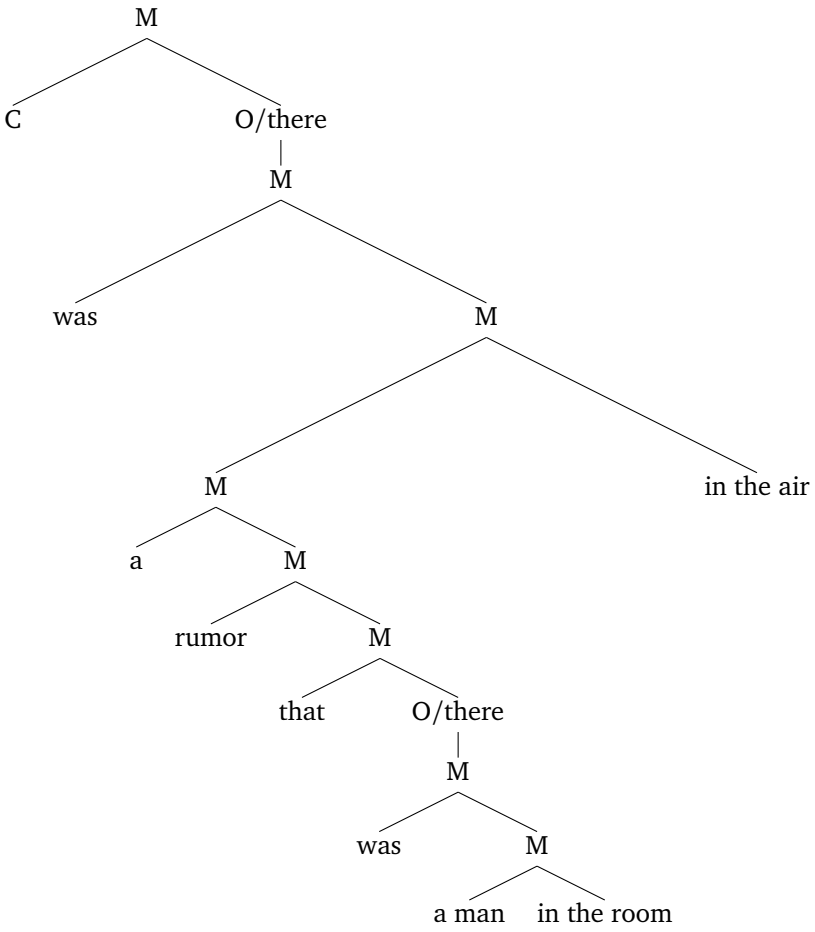


Figure 10: Underspecified Derivation Tree of (16a) and (16b)

b. There was a rumor that [a man]$_i$ was t_i in the room in the air.
c. [A rumor that there was a man in the room]$_i$ was t_i in the air.
d. [A rumor that [a man]$_i$ was t_i in the room]$_j$ was t_j in the air.

The identity of numerations condition of the original version of MOM entails that these four sentences belong to three distinct equivalence classes, one containing (17a), one containing (17b) and (17c), and one containing (17d). MOM enriched with stratified numerations, on the other hand, puts each sentence into its own equivalence class. Only tMOM lumps them all together into one equivalence class, which is the more plausible route to take, at least intuitively.

The very fact that Merge variants as well as Move variants can be obtained from the same underspecified derivation indicates that the transducer version is less of a relativized ban against Merge and more of a description of the set of possible continuations of a derivation once a choice pro-Merge or pro-Move has been made. This idea is actually what underlies the restatement of the transducer *Path Condition* in MSO terms by the constraint ψ. Empirically, this has the welcome effect that we do not run into the undergeneration problems that plague isMOM, and to a lesser degree csMOM. Consider the following utterances.

(18) a. It seems that John was in the room.
b. * John seems it was in the room.

The derivation for either sentence starts out by assembling the small clause [John [in the room]], which is subsequently merged with a T head (phonetically realized by *was*). Now isMOM would enforce base-merger of *it* into the specifier of the TP, rather than movement of *John* into said position. From there on, only ungrammatical structures can be generated. Either *John* remains in situ and the derivation crashes because of the unchecked case feature of *John*, or *John* moves over the expletive into SpecTP of the matrix clause, in violation of the Shortest Move Condition. The only grammatical alternative, (18a), cannot be generated because it is blocked by isMOM. With tMOM one does not run into this problem, as it will generate both sentences, but the second one will probably be filtered out by the MG itself because of the illicit movement step. The csMOM variant alternates between the two options: If (18b) is ungrammatical for independent reasons, (18a) does not have to compete against it and will emerge as the winner, just as with the transducer model. If (18b) is grammatical, it will block (18a), in line with isMOM.

This general theme is repeated in various configurations where other versions of MOM undergenerate. Shima (2000) lists a number of cases where Merge-over-Move makes false predictions and, in fact, something along the lines of a Move-over-Merge principle seems to be required.

(19) a. It is asked [how likely t_{John} to win]$_i$ John is t_i.
b. * John is asked [how likely t_{John} to win]$_i$ it is t_i.

The assembly of [is [how likely John to win]] proceeds as usual. At this point, a decision has to be made as to whether we want to move *John* into SpecTP or base-merge the expletive instead. The isMOM variant once again picks the base-merger route, so we end up with [it [is [how likely John to win]]. After this phrase is merged

with *asked* and *is*, *John* moves into the specifier of the matrix TP to get its case feature checked. Unless moving *John* is barred for independent reasons, (19b) will be grammatical, so that (19a) will be blocked under both indiscriminate and cautious construals of MOM. Thus we get the following contrast between different versions of MOM. The variant isMOM always blocks (19a), csMOM blocks it only if (19b) is grammatical, and tMOM never blocks it. So for both csMOM and tMOM we have to make sure that our MG $\mathscr{E}$ contains some locality condition that rules out (19b). A natural candidate would of course be the islandhood of [how likely John to win].

We also have to make further assumptions about $\mathscr{E}$ to rule out cases of superraising like (20a) and multiple occurrences of *there* as in (20b). On a conceptual level, this is a defensible move as the deviancy of those examples does not seem to be directly related to MOM, and they are hardly ever discussed with respect to MOM in the literature. However, if we really wanted to incorporate those restrictions into MOM, at least the ban against double *there* can easily be accommodated by changing from a "once you go O, you never go back" version of *Path Condition* to "once you choose, it's always O". This is easily accomplished by replacing the rule $\mathrm{O}/\mathrm{there}(q_*(x)) \rightarrow q_*(\mathrm{M}(\mathrm{there}, x))$ by the minimally different $\mathrm{O}/\mathrm{there}(q_*(x)) \rightarrow q_o(\mathrm{M}(\mathrm{there}, x))$.

(20) a. * A man seems there to be in the room.
b. * There seems there to be a man in the room.

Interestingly, at least German allows for multiple expletives to occur in a single clause, even within the *mittelfeld*, which is usually considered a part of the TP. Examples are given in (21) (my own judgments). As multiple expletives can be hosted by German TPs, the contrast between German and English can't be reduced to the fact that German mandatorily requires SpecCP to be filled and thus has two specifiers that may host expletives.

(21) a. Es/Da scheint da ein Mann im Garten zu sein.
it/there seems there a man in.the garden to be
b. Es/?Da scheint da ein Mann da im Garten zu sein.
it/there seems there a man there in.the garden to be
'There seems to be a man in the garden.'
c. Es/?Da scheint da ein Mann im Garten da zu sein.
it/there seems there a man in.the garden there to be
'There seems to be a man in the garden.'

If we assume that economy principles are universal, then any cross-linguistic variation has to arise from other grammar-internal factors. From a transducer perspective, though, there are no good reasons for such a stipulation. As long as language-specific variants of a constraint all belong to the same transducer class, they are all equally economic in a mathematical sense. In the case of *Path Condition*, the slight modification proposed above has absolutely no effect on the runtime-behavior of the transducer, nor is it in any tangible way less intuitive or less "Minimalist". Reference-set constraints must not be artificially kept away from matters of crosslinguistic variation, because this is an empirical domain where they are in principle superior to standard well-formedness conditions. This has not been noticed in the syntactic

literature yet — e.g. for Müller and Sternefeld (1996:491) "it is [...] not clear how a [reference-set; TG] constraint like Economy can be rendered subject to parametrization" — but in contrast to well-formedness conditions these constraints offer multiple loci of parametrization: the transductions α and β, and the definition of the filter as well as at which point of the transduction it is applied. Now that our formal understanding of reference-set constraints has finally reached a level where at least such basic questions can be given satisfactory answers, the initial empirical questions can be reapproached from a new angle that challenges the received wisdom on when, where and how reference-set constraints should be employed.

7 Shortest Derivation Principle

In the last section, I left open how to formalize oMOM, the variant of MOM which doesn't weigh violations depending on how early they happen in the derivation. In other words, oMOM simply counts the number of violations and picks the candidate(s) that incurred the least number of violations. This is very close in spirit to the *Shortest Derivation Principle* (SDP) of Chomsky (1991, 1995a), which I (and many authors before me) have also referred to as *Fewest Steps*.[5] The SDP states that if two convergent (i.e. grammatically well-formed) derivations are built from the same lexical items, pick the one with the fewest operations. Usually, the set of operations considered by the economy metric is assumed to comprise only Move, the reason being that Merge is indispensable if all the lexical items are to be combined into a single phrase marker. Naively, then, oMOM is but a variant of the SDP that doesn't penalize every instance of Move but only those where Merge would have been a feasible alternative. Even though I will refrain from discussing oMOM any further in this section and focus on the SDP instead, their close relation means that after reading this and the previous section, the reader will be in possession of all the tools required to formalize oMOM. In fact, I explicitly encourage the reader to draw at least a sketch of the implementation to test their own understanding of the material.

Returning to the SDP, I will explore two variants of this principle, one being the original proposal and the other one the result of extending the set of operations that enter the economy metric to Merger of phonologically unrealized material such as (certain) functional heads in the left periphery. The underlying intuition of this extension is that covert material should be merged only if is required for convergence. This would certainly be close in spirit to GB-analyses of English where main clauses are TPs unless a CP is required as a landing site for wh-movement or possibly QR. Curiously, this *prima facie* innocent modification has the potential to push the SDP out of the realm of linear transductions: the SDP restricted to Move can be defined by a linear transducer, whereas the SPD applied to Move and Merge of covert material is not, unless restrictions on the distribution of silent heads are put into place.

[5] Technically, Fewest Steps is the original formulation and the SDP its more "minimalist" reformulation that does away with representational machinery such as Form-Chain. This makes it a better fit for MGs in the sense of Stabler and Keenan (2003), and for this reason I prefer the name SDP over the better-known Fewest Steps.

7.1 The Shortest Derivation Principle Explained

To give the reader a better feeling for the constraint I do as in the previous sections and present a simple example first. It is a well-known fact that A-movement in English exhibits freezing effects. While arguments may be extracted from a DP in complement position, as soon as the DP A-moves to a higher position, usually SpecTP, extraction is illicit — the DP's arguments are frozen in place. This contrast is illustrated in (22).

(22) a. Who$_i$ did John take [$_{\text{DP}}$ a picture of t_i]?
b. * Who$_i$ was [$_{\text{DP}_j}$ a picture of t_i] taken t_j by John?

At first (22b) seems to be a mere instance of a CED-effect (Huang 1982) as in (23), so whatever rules out the latter should also take care of (22b).

(23) * Who$_i$ is [$_{\text{DP}}$ a picture of t_i] on sale?

The corresponding derivation for this analysis of the ungrammaticality of (22b) would be (24).

(24) a. [$_{\text{VP}}$ taken [$_{\text{DP}_j}$ a picture of who$_i$] by John]
b. [$_{\text{TP}}$ [$_{\text{DP}_j}$ a picture of who$_i$] T [$_{\text{VP}}$ taken t_j by John]]
c. [$_{\text{CP}}$ who$_i$ was [$_{\text{TP}}$ [$_{\text{DP}_j}$ a picture of t_i] T [$_{\text{VP}}$ taken t_j by John]]]

Notably, though, the DP in (22b) is not base-generated in subject position but in object position, so in theory it should be possible to extract the wh-word from the DP before it moves into subject position and thus becomes a barrier for movement in the sense of Chomsky (1986). There are two distinct derivations that make use of this loophole, the relevant stages of which are depicted in (25) and (26) below.

(25) a. [$_{\text{VP}}$ taken [$_{\text{DP}_j}$ a picture of who$_i$] by John]
b. [$_{\text{CP}}$ who$_i$ was [$_{\text{TP}}$ T [$_{\text{VP}}$ taken [$_{\text{DP}_j}$ a picture of t_i] by John]]]
c. [$_{\text{CP}}$ who$_i$ was [$_{\text{TP}}$ [$_{\text{DP}_j}$ a picture of t_i] T [$_{\text{VP}}$ taken t_j by John]]]

(26) a. [$_{\text{VP}}$ taken [$_{\text{DP}_j}$ a picture of who$_i$] by John]
b. [$_{\text{VP}}$ who$_i$ taken [$_{\text{DP}_j}$ a picture of t_i] by John]
c. [$_{\text{TP}}$ [$_{\text{DP}_j}$ a picture of t_i] T [$_{\text{VP}}$ who$_i$ taken t_j by John]]
d. [$_{\text{CP}}$ who$_i$ was [$_{\text{TP}}$ [$_{\text{DP}_j}$ a picture of t_i] T [$_{\text{VP}}$ taken t_j by John]]]

The first derivation can be ruled out on grounds of the extension condition, which bans countercyclic movement. The second, however, seems to be well-formed, provided that extraction of the wh-phrase is licensed by feature checking (in a phase-based approach, this could be handled by an EPP/OCC-feature, for instance). So we erroneously predict that (22b) should be grammatical.

Collins (1994) solves this puzzle by recourse to the SDP. Note that (24) and (25) involve one movement step less than (26). So if (26) has to compete against at least one of the two, it will be filtered out by the SDP. The filtering of (24) and (25), respectively, is then left to the subject island constraint and the ban against

countercyclic movement (whatever their technical implementation might be in our grammar).

The reader may be wondering why non-convergent derivations are suddenly parts of the reference-sets. From a formal perspective, there is little reason to argue against this shift in perspective; as we will see in a second, it changes nothing about our general procedure. However, from a linguistic point of view this point is definitely worth elaborating. The answer is that earlier in this section I allowed myself a small degree of sloppiness when explaining convergence, as the derivations above are in fact convergent. For even though all well-formed derivations are by definition convergent, the latter is not true in general. Convergence means that all items from the numeration were merged correctly and that no feature was left unchecked, while well-formedness also implies that no other constraints were violated.

7.2 A Model of the Shortest Derivation Principle

The SDP features interesting extensions of the constraints seen so far. As it punishes every single instance of Move, it is indeed a counting constraint, in contrast to Focus Economy and MOM, which relied on surprisingly simple well-formedness conditions on somewhat peculiar paths. Moreover, it involves two cycles of under-specification and filtration rather than one, and those two cycles furthermore happen to interlock in a non-trivial way. Nonetheless the formalized version of SDP actually uses the simplest transducers of all in this paper, so it should be easy to fathom for everyone who has already mastered Focus Economy and MOM.

As with MOM, we start out with the set of derivation trees of some MG $\mathscr{E}$. And as we did before, we immediately strip away all the features except the category feature, which is preserved so that distinct trees with identical string components won't be put in the same reference set later on.

Definition 39. *Remove Features* is the deterministic (one-state) relabeling that maps each $l := \langle \sigma :: f_1, \ldots, f_{base}, \ldots, f_n \rangle \in Lex_{\mathscr{E}}$ to $l' := \sigma_{f_{base}}$, where f_{base} is the base feature of l. The set of these simplified lexical items is denoted by Λ.

In the next step, we have to ensure that two derivations wind up in the same reference set if and only if they differ merely in their number of movement steps. To this end, we first define a transducer that deletes all unary branches (i.e. branches representing Move) from the derivation, and then another one which arbitrarily reinserts unary branches. This will generate derivations that were not present at the stage immediately before we removed all instances of Move, but as the reader might have guessed, this can easily be fixed by following our own example set in the previous section and use the input language as a filter — only this time the "input language" isn't the derivation language of $\mathscr{E}$ but the language serving as the input to the transducer that removed all movement nodes, i.e. the output language of *Remove Features*. The result of these three transductions and the filtration is a transduction that relates only those (feature-free) derivations that are built from the same lexical items.

Definition 40. *Remove O* is the deterministic ltdtt $\mathscr{R}$, where $\Sigma_{\mathscr{R}} := \Lambda \cup \{M, O\}, \Omega := \Sigma_{\mathscr{R}} \setminus \{O\}, Q = Q' := \{q\}$, and $\Delta_{\mathscr{R}}$ consists of the rules below:

$$\sigma \to q(\sigma) \qquad\qquad M(q(x), q(y)) \to q(M(x, y))$$
$$O(q(x)) \to q(x)$$

Definition 41. *Insert O* is the non-deterministic ltdtt $\mathscr{I}$, where $\Sigma_{\mathscr{I}} := \Omega_{\mathscr{R}}$, $\Omega_{\mathscr{I}} := \Sigma_{\mathscr{R}}$, $Q = Q' := \{q\}$, and $\Delta_{\mathscr{I}}$ contains the rules below, with $O^{\leq n}$ denoting n-many unary O-labeled branches or less for some fixed, non-negative n:

$$\sigma \to q(\sigma) \qquad\qquad M(q(x), q(y)) \to O^{\leq n}(M(x, y))$$

One strong restriction of *Insert O* is that at any node in the derivation tree it can only insert a finite number of movement steps. This is so because a transducer may only have finitely many rules and after every step in the transduction the transducer has to move one step up in the input tree, so it cannot remain stationery at one point and keep inserting one unary branch after another until it finally decides to move on. A transducer with such capabilities is said to have ϵ-moves, and such transducers do not share the neat properties of their standard brethren. However, the restriction to only finitely many unary branches per rewrite-step is immaterial for MGs. This follows from the simple observation that since the number of features per lexical item is finite, and so is the lexicon itself, there is a longest string of features for each grammar. The length of this string dictates how many movement steps may be licensed by a single lexical item, and thus there is an upper bound on the number of movement steps between any two instances of Merge. If we are dealing with more specialized types of Move such as Sidewards Movement (Nunes 2004; Drummond 2010), countercyclic movement or asymmetric checking, the argument may not go through without further assumptions, but for a canonical MG, the limits of the transducer are inconsequential because they are also limits of the grammar.[6]

The composition of *Remove Features*, *Remove O* and *Insert O* will be our reference-set algorithm. The next step, then, is the definition of the economy metric. But for this not much more work is needed, because the metric is already given by *Insert O*. The transducer all by itself already defines rel_1^{GEN} (see Def. 10 on page 11), the ranking of all output candidates relativized to those candidates that compete against each other, so all we have to do is follow Jäger's procedure as outlined in Sec. 2. Recall the basic intuition: the transducer defines a relation $<$ on the output candidates such that $o < o'$ iff o' is the result of applying the transducer to o. Given this relation, a few nifty regular operations are enough to filter out all elements that are not minimal with respect to $<$, i.e. the suboptimal candidates. The result will be a transduction mapping, as desired, inputs to the derivation tree(s) over the same lexical items that contain(s) the fewest instances of Move — it only remains for us to reinstantiate the features, which is taken care of by the inverse of *Remove Features*.

The overall architecture of the SDP model is depicted in Fig. 11 on the facing page. Switching back for a second into the OS perspective, it is also easy to see that

[6] That countercyclic movement poses a challenge is somewhat unsatisfying insofar as it surfaces in Collins's analysis of (25). Fortunately, though, his general argument goes through even if only (25) is taken into account.

both type-level optimality and output joint preservation are satisfied (if it weren't for the removal of features, the OS would even be endocentric), thereby jointly implying global optimality.

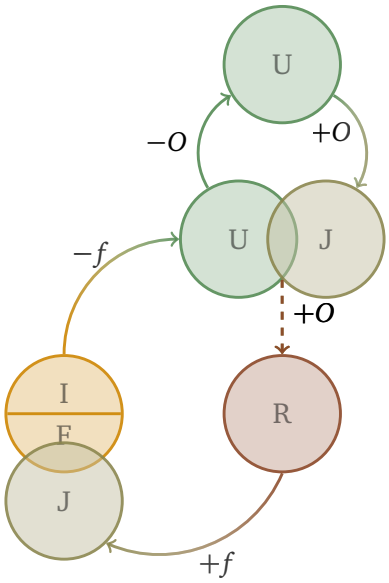


Figure 11: Architecture of the SDP

The astute reader may point out that my implementation of the SDP, while technically correct, leads to both underapplication and overapplication of the intended principle. Overapplication is caused by the indiscriminate removal of features, in particular movement-licensors as are involved in topicalization, wh-movement and (possibly) scrambling. As a consequence, these kinds of movement will appear redundant to the SDP and lose out to the derivation that involves only standard A-movement. This is easily fixed by "blacklisting" these features such that they have to be preserved by *Remove Features*.

Underapplication, on the other hand, is due to the lack of a transduction that would remove covert material whose only purpose is to host a movement-licensor feature. So if, say, a topicalization feature is always introduced by the category *Topic* à la Rizzi (1997, 2004), a derivation hosting this functional element will never compete against a derivation without it. For topicalization, this is actually a welcome result and presents an alternative for avoiding overapplication. In general, though, this must be regarded as a loophole in the SDP that needs to be fixed lest the principle can be deprived of any content by assigning every movement feature its own functional category. A solution is readily at hand: Extend *Remove O* and *Insert O* such that they may also remove or insert certain functional elements, just like MOM's *Underspecify* may remove instances of expletive *there* that can later be reinserted by *Path Condition*.

While the parametrization of *Remove Features* poses no further problems irrespective of the MG involved, extending *Remove O* and *Insert O* to functional categories will produce the correct results only if our initial grammar does not allow for re-

cursion in the set of categories that the transducer should remove. In other words, there has to be an upper limit on the number of removable categories that can be merged subsequently before non-removable material has to be merged again. This is because of the previously mentioned inability of linear transducers to insert material of unbounded size. On a linguistic level, the ban against recursion in the functional domain is fairly innocent as even highly articulated cartographic approaches give rise only to a finite hierarchy of projections.

Conclusion

I showed that despite claims to the contrary, reference-set constraints aren't computationally intractable — in fact, many of them do not even increase the expressivity of the underlying grammar. The route towards this result was rather indirect. I first introduced controlled OSs as a formal model for reference-set constraints, focusing on the subclass of output joint preserving OSs, which is general enough to accommodate most reference-set constraints. For this class I then gave a new characterization of global optimality and used it to argue that in general, reference-set constraints are globally optimal. The shift in perspective induced by controlled OSs also made it apparent that out of the other four conditions which together with global optimality jointly guarantee that an OS stays within the limits of linear tree transductions, two are almost trivially satisfied by reference-set constraints, with the only problematic areas being the power of GEN and the rankings induced by the constraints on the range of GEN. This highlights how surprisingly restricted reference-set constraints are in comparison to optimality systems, even though the latter seem to struggle with reference-set like conditions such as output-output correspondence (Benua 1997; Potts and Pullum 2002).

In order to demonstrate that GEN and the evaluation metric do not pose a problem either, I exhibited formally explicit implementations of three different reference-set constraints: Focus Economy, Merge-over-Move, and the Shortest Derivation Principle. The general approach followed a strategy of underspecification-and-filtration (Fig. 8) based on the following insights:

- A reference-set algorithm is likely to be computable by a linear transducer if there is a data-structure (e.g. derivation trees) such that all members of a reference-set can be uniquely described by this structure.
- Neither the mapping from inputs to underspecified structures nor the one from underspecified structures to output candidates may require insertion of material of unbounded size.
- The economy metric may be implemented as a well-formedness condition on underspecified structures, an instruction for how to turn those structures into output candidates, or a relation over underspecified structures computable by a linear transducer.

My estimate so far is that all reference-set constraints are compatible with the underspecification strategy, and that all syntactic reference-set constraints also adhere

to condition 2. Combined with the positive results obtained in this paper, this suggest that reference-set constraints are significantly better behaved than is usually believed. The overall picture that emerges is that of reference-set constraints as an unexpectedly undemanding kind of linguistic constraint.

One important issue had to be left open, though, namely the closure properties of Minimalist surface tree and derivation tree languages, in particular with respect to linear transductions and intersection with regular tree languages. Even though it has no immediate bearing on the formal models in this paper (the constraints can be emulated by conditions on the distribution of features in an MG), a lack of closure under these operations, in particular closure of Minimalist derivation tree languages under intersection with regular sets, would likely prove a severe impediment to the applicability of the underspecification-and-filtration strategy. But even then everything wouldn't be lost since closure under intersection with any regular language is arguably a more general property than what we actually need.

Acknowledgments

I am greatly indebted to Ed Stabler and Uwe Mönnich as well as the two anonymous FG2010-reviewers for their motivational comments and helpful criticism. The research reported herein was supported by a DOC-fellowship of the Austrian Academy of Sciences.

References

Aoun, Joseph, Lina Choueiri, and Norbert Hornstein. 2001. Resumption, movement and derivational economy. *Linguistic Inquiry* 32:371–403.

Benua, L. 1997. Transderivational identity: Phonological relations between words. Doctoral Dissertation, UMass.

Castillo, Juan Carlos, John E. Drury, and Kleanthes K. Grohmann. 2009. Merge over move and the extended projection principle: MOM and the EPP revisited. *Iberia* 1:53–114.

Chomsky, Noam. 1986. *Barriers*. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 1991. Some notes on economy of derivation and representation. In *Principles and parameters in comparative grammar*, ed. Robert Freidin, 417–454. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 1995a. Categories and transformations. In *The minimalist program*, chapter 4, 219–394. Cambridge, Mass.: MIT Press.

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

Chomsky, Noam. 2000. Minimalist inquiries: The framework. In *Step by step: Essays on minimalist syntax in honor of Howard Lasnik*, ed. Roger Martin, David Michaels, and Juan Uriagereka, 89–156. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2001. Derivation by phase. In *Ken Hale: A life in language*, ed. Michael J. Kenstowicz, 1–52. Cambridge, Mass.: MIT Press.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English*. New York: Evanston.

Collins, Chris. 1994. Economy of derivation and the generalized proper binding condition. *Linguistic Inquiry* 25:45–61.

Collins, Chris. 1996. *Local economy*. Cambridge, Mass.: MIT Press.

Drummond, Alex. 2010. Fragile syntax and sideward movement. Ms., University of Maryland.

Engelfriet, Joost. 1975. Bottom-up and top-down tree transformations — a comparison. *Mathematical Systems Theory* 9:198–231.

Fox, Danny. 1995. Economy and scope. *Natural Language Semantics* 3:283–341.

Fox, Danny. 2000. *Economy and semantic interpretation*. Cambridge, Mass.: MIT Press.

Frank, Robert, and Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24:307–315.

Gärtner, Hans-Martin. 2002. *Generalized transformations and beyond: Reflections on minimalist syntax*. Berlin: Akademie-Verlag.

Grodzinsky, Yosef, and Tanja Reinhart. 1993. The innateness of binding and coreference. *Linguistic Inquiry* 24:69–102.

Gécseg, Ferenc, and Magnus Steinby. 1984. *Tree automata*. Budapest: Academei Kaido.

Heim, Irene. 1998. Anaphora and semantic interpretation: A reinterpretation of Reinhart's approach. In *The interpretive tract*, ed. Uli Sauerland and O. Percus, volume 25 of *MIT Working Papers in Linguistics*, 205–246. Cambridge, Mass.: MIT Press.

Heim, Irene. 2009. Forks in the road to Rule I. In *Proceedings of NELS 38*, 339–358.

Hopcroft, John E., and Jeffrey D. Ullman. 1979. *Introduction to automata theory, languages, and computation*. Reading, Mass.: Addison Wesley.

Hornstein, Norbert. 2001. *Move! A minimalist theory of construal*. Oxford: Blackwell.

Huang, C.-T. James. 1982. Logical relations in Chinese and the theory of grammar. Doctoral Dissertation, MIT.

Johnson, David, and Shalom Lappin. 1999. *Local constraints vs. economy*. Stanford: CSLI.

Joshi, Aravind. 1985. Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In *Natural language parsing*, ed. David Dowty, Lauri Karttunen, and Arnold Zwicky, 206–250. Cambridge: Cambridge University Press.

Jäger, Gerhard. 2002. Gradient constraints in finite state OT: The unidirectional and the bidirectional case. In *More than words. A festschrift for Dieter Wunderlich*, ed. I. Kaufmann and B. Stiebels, 299–325. Berlin: Akademie Verlag.

Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. Manuscript, Xerox Research Center Europe.

Kepser, Stephan, and Uwe Mönnich. 2006. Closure properties of linear context-free tree languages with an application to optimality theory. *Theoretical Computer Science* 354:82–97.

Keshet, Ezra. 2010. Situation economy. *Natural Language Semantics* 18:pp–pp.

Kobele, Gregory M. 2006. Generating copies: An investigation into structural identity in language and grammar. Doctoral Dissertation, UCLA.

Kobele, Gregory M. 2010. Without remnant movement, MGs are context-free. In *MOL 10/11*, ed. Christian Ebert, Gerhard Jäger, and Jens Michaelis, volume 6149 of *Lecture Notes in Computer Science*, 160–173.

Kobele, Gregory M., Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In *Model Theoretic Syntax at 10*, ed. James Rogers and Stephan Kepser, 71–80. Workshop organized as part of the Europen Summer School on Logic, Language and Information, ESSLLI 2007, 6-17 August 2007, Dublin, Ireland.

Kolb, Hans-Peter, Jens Michaelis, Uwe Mönnich, and Frank Morawietz. 2003. An operational and denotational approach to non-context-freeness. *Theoretical Computer Science* 293:261–289.

Michaelis, Jens. 1998. Derivational minimalism is mildly context-sensitive. *Lecture Notes in Artificial Intelligence* 2014:179–198.

Michaelis, Jens. 2001. Transforming linear context-free rewriting systems into minimalist grammars. *Lecture Notes in Artificial Intelligence* 2099:228–244.

Müller, Gereon, and Wolfgang Sternefeld. 1996. A-bar chain formation and economy of derivation. *Linguistic Inquiry* 27:480–511.

Nakamura, Masanori. 1997. Object extraction in Bantu applicatives: Some implications for minimalism. *Linguistc Inquiry* 28:252–280.

Nunes, Jairo. 2004. *Linearization of chains and sideward movement*. Cambridge, Mass.: MIT Press.

Potts, Christopher. 2002. Comparative economy conditions in natural language syntax. Paper presented at the North American Summer School in Logic, Language, and Information 1, Workshop on Model-Theoretic Syntax, Stanford University (June 28), June 2002.

Potts, Christopher, and Geoffrey K. Pullum. 2002. Model theory and the content of OT constraints. *Phonology* 19:361–393.

Prince, Alan, and Paul Smolensky. 2004. *Optimality theory: Constraint interaction in generative grammar*. Oxford: Blackwell.

Reinhart, Tanya. 2006. *Interface strategies: Optimal and costly computations*. Cambridge, Mass.: MIT Press.

Rezac, Milan. 2007. Escaping the person case constraint: Reference-set computation in the ϕ-system. *Linguistic Variation Yearbook* 6:97–138.

Rizzi, Luigi. 1997. The fine-structure of the left periphery. In *Elements of grammar*, ed. Liliane Haegeman, 281–337. Dordrecht: Kluwer.

Rizzi, Luigi. 2004. Locality and left periphery. In *The cartography of syntactic structures*, ed. Adriana Belletti, volume 3, 223–251. New York: Oxford University Press.

Rogers, James. 1997. "Grammarless" phrase structure grammar. *Linguistics and Philosophy* 20:721–746.

Rogers, James. 1998. *A descriptive approach to language-theoretic complexity*. Stanford: CSLI.

Shieber, Stuart M. 2004. Synchronous grammars as tree transducers. In *TAG+7: Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms*, 88–95.

Shieber, Stuart M. 2006. Unifying synchronous tree adjoining grammars and tree transducers via bimorphisms. In *Proceedings of the 11^{th} Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 377–384.

Shieber, Stuart M., and Yves Schabes. 1990. Synchronous tree adjoining grammars. In *Proceedings of the 13^{th} International Conference on Computational Linguistics*, 253–258.

Shima, Etsuro. 2000. A preference for move over merge. *Linguistic Inquiry* 375–385.

Stabler, Edward P., and Edward Keenan. 2003. Structural similarity. *Theoretical Computer Science* 293:345–363.

Sternefeld, Wolfgang. 1996. Comparing reference-sets. In *The role of economy principles in linguistic theory*, ed. Chris Wilder, Hans-Martin Gärtner, and Manfred Bierwisch, 81–114. Berlin: Akademie Verlag.

Szendrői, Kriszta. 2001. Focus and the syntax-phonology interface. Doctoral Dissertation, University College London.

Thomas, Wolfgang. 1997. Languages, automata and logic. In *Handbook of formal languages*, ed. Gregorz Rozenberg and Arto Salomaa, volume 3, 389–455. New York: Springer.

Toivonen, Ida. 2001. On the phrase-structure of non-projecting words. Doctoral Dissertation, Stanford, CA.

Wartena, Christian. 2000. A note on the complexity of optimality systems. In *Studies in optimality theory*, ed. Reinhard Blutner and Gerhard Jäger, 64–72. Potsdam, Germany: University of Potsdam.

Wilder, Chris, and Hans-Martin Gärtner. 1997. Introduction. In *The role of economy principles in linguistic theory*, ed. Chris Wilder, Hans-Martin Gärtner, and Manfred Bierwisch, 1–35. Berlin: Akademie Verlag.

Affiliation

Thomas Graf
Department of Linguistics
University of California, Los Angeles
tgraf@ucla.edu

UCLA Working Papers in Linguistics, Papers in Mathematical Linguistics 1
Volume 15, Article 5: 1–30, 2010

Representational Maps from the Speech Signal to Phonological Categories: a Case Study with Lexical Tones

Kristine M. Yu

As the initial step in studying the acquisition of phonological categories from the speech signal, we describe representational issues for the target of learning, a probabilistic distribution of phonological categories over a phonetic parameter space. Our model system of study is cross-linguistic lexical tonal phonemes in tonal languages. We focus on two representational issues: temporal resolution of the extracted phonetic parameters and static and dynamic parameterizations of the speech signal. In a human perception study and exploratory computational modeling, we find that coarse sampling of absolute f0 and f0 velocity is sufficient for near-partitions of the phonetic parameter space for single-speaker tonal spaces in a range of tone languages.

Keywords phonetics, phonology, tone, learnability

Introduction

The phonetic realization of linguistic tone is widely believed to be simple and limited to a single dimension of fundamental frequency (the physical correlate of the auditory percept of pitch).

> ...tones typically involve a single primary acoustic dimension, namely, f0. This contrasts with the multiple acoustic dimensions such as formants or spectral peaks required for characterizing vowels and consonants. The variability problem with tones is therefore at least limited to a single dimension... (Gauthier, Shi, and Xu 2007:82)

> ...tone presents few, if any articulatory difficulties vs. consonants (which all languages have). Second, tone is acoustically (hence perceptually?) simple, F_0, vs. consonants and vowels. (Hyman 2010:1)

In this paper, we show that the phonological representation of tone in terms of phonetic parameters may indeed be simple, but not necessarily in the way that has been described above. Even an entirely f0-based parameterization of tone can be highly multidimensional, since we may choose multiple ways to parametrize f0, e.g. with f0 height values and with f0 velocity values, and we may choose to sample these values arbitrarily densely in time. Here we show that: (i) both f0 height and f0 velocity are relevant parameters for a range of tone languages, even for the

simplest level tone languages, and (ii) the relation between phonetic space and tonal categories may be simple, but in a way that may not be unique to tone: we show that coarse sampling of the relevant parameters suffices for good category separability—a near partition of the phonetic space—in a range of languages, and that humans can identify tones degraded to be coarsely sampled at a comparable level of accuracy to that for intact tones.

The work in this paper bears on defining the target of learning in the acquisition of lexical tonal categories from the speech signal, the initial step towards answering our larger research questions: (i) what the relation between phonetic spaces and tonal phonological categories is, i.e. how tones are phonetically realized, (ii) how that relation between the phonetic space and phonological categories could be learned, and (iii) how it is learned by L1 human learners. We frame our work broadly to scientifically explicate the universal structure in the phonetic parameter space across phonetically diverse tonal systems; we set up learning tonal categories as a model system for learning phonological categories to integrate the study of the acquisition of tone into the highly active research area of language acquisition in general. We take the broad perspective of Welmers (1973):

> In principle, the varieties and functions of tonal contrasts in language are of the same order as the varieties and functions of any other contrasts; the problems of tonal analysis are simply typical problems of linguistic analysis. (Welmers 1973:77)

Thus, we begin in §1 with preliminaries: we describe the learning problem in the context of phonological category acquisition, motivate the study of the target of learning (the map from the phonetic space to phonological categories), describe the larger research questions and methodological abstractions taken in the study and explicate our particular model system. In §2, we state the aspects of phonological representation, and more specifically, of tonal representation, that are the focus of this paper. These aspects are temporal resolution of the parameterized speech signal and static vs. dynamic properties of the speech signal. We end in §3 by briefly highlighting results from our own experiments and initial computational modeling work addressing these aspects.

1 Preliminaries

This paper investigates the learnability of lexical tonal phonological categories in tone languages. It is a preliminary step in the study of a much larger research question:

(Q0) *How do children acquire phonological categories from the speech signal?*

We address (Q0) using computational learning methods, like previous studies of phonological category learning, cf. de Boer and Kuhl (2003); Lin (2005); Toscano and McMurray (2010); Vallabha, McClelland, Pons, Werker, and Amano (2007), and moreover, we ground our modeling assumptions based on phonetic fieldwork and perception experiments we conducted.

While a complete answer to Q0 necessitates a battery of behavioral, physiological, production, and perceptual studies on infants from the womb to adulthood, particularly in the first years of life, our ability to probe infant knowledge of phonological categories and connect this knowledge to their language input is limited, cf. methodological approaches in Polka, Jusczyk, and Rvachew (1995); Werker, Shi, Desjardins, Pegg, and Polka (1998). Thus, we make the choice to generalize our study to *any learner* so that we can deploy mathematically-specified learners to learn from examples we have very fine control over. The advantage of focusing on computational approaches is that we can make a tight connection between the data that a learning machine gets (the domain of the learner, $\mathscr{D}$), how it went about the learning (the functional/algorithmic form of the learner, $\mathscr{A}$), and the target of learning (the codomain of the learner, $\mathscr{C}$). The challenge then is to also maintain a tight connection between the computational modeling and what we know about human learners.

Thus, we modify our original research question:

(Q0′) *How could a learner $\mathscr{A} : Data \to \mathscr{C}$ acquire lexical tonal categories from the speech signal in a way consistent with our knowledge about how humans do it?*

A key component in maintaining a tight connection between the computational modeling and human cognition is to have a clear picture of what the target of learning is (Dyson 2004; Minsky and Papert 1971). Thus, the goal of this paper is to define the codomain, $\mathscr{C}$, the target of learning in the acquisition of lexical tonal categories: we define what it means to have learned the lexical tonal categories of a tonal language; **we study the *learnability* of tonal spaces, conditioned on different representations of tonal examples, to understand how lexical tonal categories are defined**.

1.1 The Target of Learning: the Phonetics-Phonology Map

What does it mean to have learned the tones of a tone language, e.g. the tones of Mandarin: Tones 1-4, respectively, ˥ (high level), ˧˥ (rise), ˨˩˦ (fall-rise), ˥˩ (fall) (c.f. Fig. 1)? We assert that it means that the learner has learned a representational map:

(1) $$\mathscr{A} : Data \to \text{representational map}$$

and that this phonetics-phonology map is of the form:

(2) *Phonetics-Phonology:* {sequences of phonetic parameter vectors} → {sets of phonological categories}

where the phonological categories are lexical tonal categories.

We show a familiar example of a well-studied phonetics-phonology map in Fig. 2, a vowel formant plot (Peterson and Barney 1952). This is a two-dimensional map

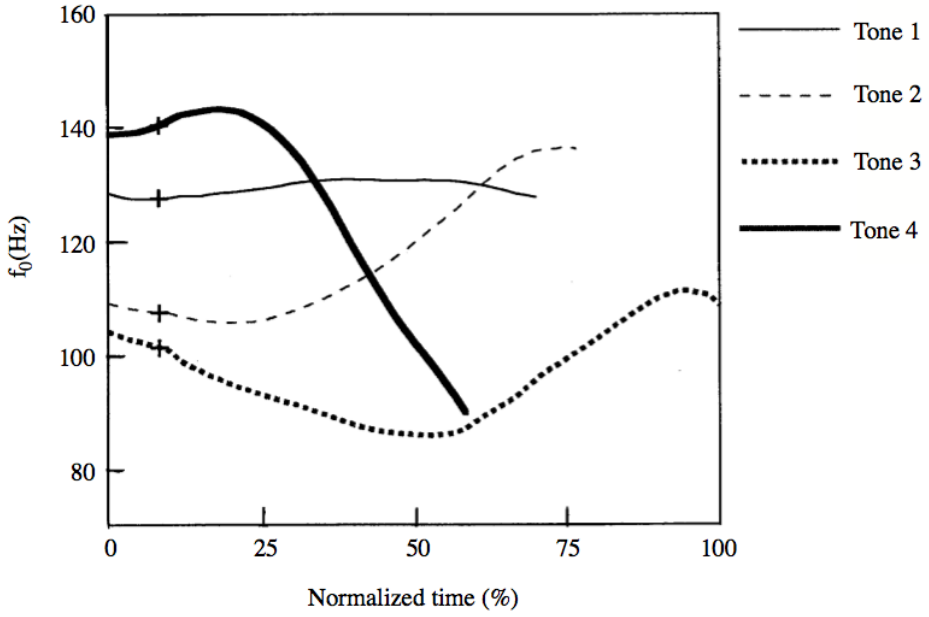


Figure 1: The tones of Mandarin (Xu 1997).

in $\langle F1_{SS}, F2_{SS}\rangle$ space (over the steady-state values of the first and second formants) which maps unit-length sequences of phonetic parameter vectors $\langle F1_{SS}, F2_{SS}\rangle$ to English vowel phonemes, cf. Table 1.

$\langle F1_{SS}, F2_{SS}\rangle$	English vowel phoneme	Note
$\langle 240, 2280\rangle$	{/i/}	Actual data point
$\langle 460, 1330\rangle$	{/ɝ/}	Actual data point
$\langle 475, 1220\rangle$	{/ʊ/}	Actual data point
$\langle 686, 1028\rangle$	{/ɑ, ɔ/}	Ambiguity
$\langle 400, 3500\rangle$	{/i/}	Not a data point
:	:	

Table 1: The representational map from steady state formant space to English vowel phonemic categories from Peterson and Barney (1952).

There are two things to note from Fig. 2 and Table 1 which are general properties of phonetics-phonology maps:

1. There are regions of $\langle F1_{SS}, F2_{SS}\rangle$ space where the same $\langle F1_{SS}, F2_{SS}\rangle$ point is mapped to multiple English vowel phonemes: regions where vowel ellipses overlap. This highlights that *ambiguity in phonetic-phonological maps implies a codomain of sets of phonological categories rather than of single phonological categories.*

2. The map is total within the vowel ellipses for $\langle F1_{SS}, F2_{SS}\rangle$ values, meaning that *all* $\langle F1_{SS}, F2_{SS}\rangle$ points included in the sets of $\langle F1_{SS}, F2_{SS}\rangle$ values bounded by the

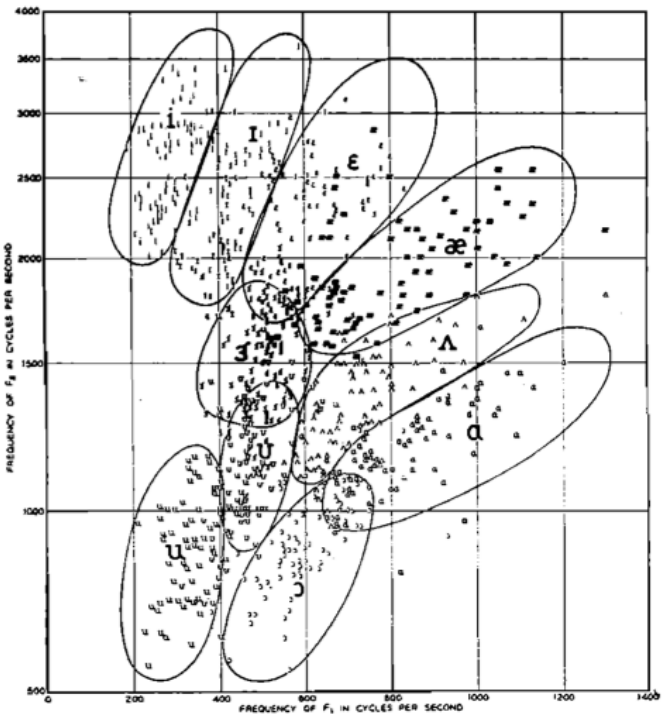


FIG. 8. Frequency of second formant *versus* frequency of first formant for ten vowels by 76 speakers.

Figure 2: A famous example of a well-studied phonetics-phonology map, the vowel formant plot (Peterson and Barney 1952).

> ellipses and not only the data points shown in Fig. 1 are mapped to a member (or multiple members) of the set of English vowel phonemes. Further, because the map is defined over a continuous space, it would never be possible to hear all the $\langle F1_{SS}, F2_{SS} \rangle$ points enclosed in the ellipses because there are infinitely many. Thus, *in learning a phonetic-phonological map defined over a* ***continuous*** *space, generalization occurs from a finite data sample to an infinite set*.

As Pierrehumbert (1990) discusses, phonetics-phonology representational maps have parallels to the "semantic" form-meaning representational maps in morphosyntax:

(3) Morphosyntax : {sequences of morphemes} → {sets of meanings}

There is ambiguity in form-meaning mappings in morphosyntax, too, especially when we abstract away from relevant context (e.g. pragmatic and prosodic context in morphosyntax; morphosyntactic context in phonetics-phonology); moreover, we add that generalization from a finite data sample to an infinite language occurs for both learning problems. The major structural difference between the phonetics-phonology and morphosyntax maps is that phonetics-phonology maps are defined in the real

rather than the discrete domain.[1] It is because of this structural difference that the mathematical machinery for studying the two different maps diverges.[2]

Our current understanding of the phonetics-phonology map, after Pierrehumbert (2003a), is in fact an elaboration of (2): we augment each phonological category in the codomain with the probability that the sequence of phonetic parameter vectors belongs to it; we elaborate the map from one mapping $\langle F1_{SS}, F2_{SS}\rangle$ to sets of phonological categories, e.g.

(4) $$\langle F1_{SS} = 686, F2_{SS} = 1028\rangle \mapsto \{/\text{ɑ}, \text{ɔ}/\}$$

to a probability distribution of the categories over $\langle F1_{SS}, F2_{SS}\rangle$ vectors, e.g.:

(5) $$\langle F1_{SS} = 686, F2_{SS} = 1028\rangle \mapsto \{p(/\text{ɑ}/) = 0.45,\ p(/\text{ɔ}/) = 0.55\}$$

With our full model of the phonetics-phonology map as a **probabilistic distribution of phonological categories over a phonetic parameter space**, the key questions we need to answer to characterize the map are:

(Q1a)	*What kinds of phonological categories are to be represented?*
(Q1b)	*What is the phonetic parameter space for the phonological categories defined in (Q1a)?*
(Q1c)	*What are properties of the distributions of the phonological categories of (Q1a) over the phonetic parameter space of (Q1b)?*

Phonological categories (Q1a) The choice of definition for the codomain of the phonetics-phonology map, the set of phonological categories, revolves around how contexualized the categories are. Peperkamp, Calvez, Nadal, and Dupoux (2006); Pierrehumbert (2003a,b) argue for the set to be a set of positional allophones, and for unification into phonemes using information from distributions of symbolic allophones or by using knowledge of the lexicon; Dillon, Dunbar, and Idsardi (Unpublished) argues for the set to be phonemes. Another option is to define the codomain over phonological features (Lin and Mielke 2008; Mielke 2008). Answering this question is not the focus of this paper, since we restrict attention here to tonal phonemes.

[1] There are also approaches to studying morphosyntax that model morphosyntax maps as being real-valued, cf. Widdows (2004): the co-occurrence of words in documents is used to determine similarity of word meanings, measured in real-valued vector spaces.

[2] It is possible to define a discrete phonetics-phonology map and thus study the phonological categorization problem using formal learning theory because we can represent the real-valued speech signal digitally to arbitrary precision in the limit, cf. Appendix A, (Jain, Osherson, Royer, and Sharma 1999). In fact, one may argue that the phonetics-phonology map is most correctly modeled over a discrete space because of precision limits in computing and biological systems (Blum 2004; Blum, Cucker, Shub, and Smale 1997). However, at the current stage of inquiry it is not clear how studying the phonetics-phonology learning problem using methods from formal learning theory gives us insight into how the learning occurs, and thus we do not pursue it here.

Phonetic parameter spaces (Q1b) It is answering (Q1b) that is the focus of this paper: to characterize the domain of the phonetics-phonology map by motivating which phonetic parameters are most significant for defining phonological categories; these are the dimensions that we want to define the distributions over. The set of phonetic parameters that may be extracted from the speech signal is obviously infinite in size and therefore must be constrained by some metric for computational tractability. For scientific purposes, too, we seek to limit the dimensionality of the phonetics-phonology map, i.e., the size of the parameter set, in order to have a succinct representation that is intelligible to the human scientist (Occam's razor). From the learner's perspective, a succinct learning target prevents overfitting to the input data and facilitates generalization to novel data (Duda, Hart, and Stork 2001:8-10), (MacKay 2003:343–349); from our scientific perspective, a succinct characterization of the representational map facilitates our ability to understand how the learning proceeds. In the best case, succinctness in the representational map results in no loss of information, i.e. without any smoothing out of the distributional modes corresponding to category structure in the phonetic space;[3] otherwise, the goal is succinctness with minimal loss of information.

We are in fact interested in characterizing three classes of phonetic parameter spaces to answer (Q0′), which is a question about lexical tone acquisition in general:

1. a universal parameter space $\mathscr{U}$ for all tone languages
2. the language-specific parameter space $\mathscr{L}$ for a given tone language
3. the speaker-specific parameter space $\mathscr{S}_{\mathscr{L}}$ for a given speaker of a given tone language.

By a parameter space, we mean the set of parameters over which the space is defined. By universal parameter space, we mean the smallest universal parameter space, the space which includes exactly and only the union of all language-specific parameter spaces.[4] To a first approximation, we assume:

(6) $$\forall \mathscr{L}, \forall \mathscr{S}_{\mathscr{L}},\ \mathscr{U} \supseteq \mathscr{L} \supseteq \mathscr{S}_{\mathscr{L}}.$$

This entails that the universal parameter space $\mathscr{U}$ can draw more distinctions than any tonal language-specific parameter space $\mathscr{L}$, which can, in turn, draw more distinctions than any speaker-specific parameter space for that language, $\mathscr{S}_{\mathscr{L}}$.

The assumption in (6) is motivated by the overarching idea based on empirical work on infant speech perception development over the past few decades that infants

[3] A simple example of succinctness without information loss is the expression of a finite language as a finite state automaton rather than as a list, since it takes fewer symbols to specify the finite state automaton than the list, and exactly and only the same sentences in the language are expressed (Meyer and Fischer 1971).

[4] The notion of a parameter space for all tone languages assumes that the class of tone languages is definable as a subset of all natural languages. Whether the restriction of languages to tone languages is available in acquisition is an open question, i.e. do children know they are learning a tone language, and if they do, under what conditions do they do this, and how do they do this? For the scope of this paper, we assume a restriction to a parameter space for tone languages for convenience.

begin as "citizens of the world" in having a universal ability to distinguish between sound categories and develop language-specific maps of the acoustic space through exposure to language input (Kuhl 2004). For instance, one of the first results of this kind was that English-learning infants showed behavioral responses consistent with the ability to discriminate between a velar stop (a sound in English) and a uvular stop (a sound not in English, but in Salish) at 6-8 months of age, but that by 10-12 months of age, they did not anymore (Werker and Tees 1984). Subsequent work confirmed and built on these results to flesh out a developmental timeline of perceptual reorganization of the acoustic space in which:

- Infants show a decline in their ability to discriminate nonnative vowel contrasts between 4-6 months (e.g. Polka and Werker 1994).
- Infants learning a non-tonal language show a decline in their ability to discriminate lexical tonal contrasts between 6 and 9 months (Mattock, Molnar, Polka, and Burnham 2008).
- Infants show a decline in their ability to discriminate nonnative consonantal contrasts between 6-8 and 10-12 months (e.g Werker and Tees 1984).
- Infants show improvement (facilitation) in their ability to discriminate native consonantal contrasts over the first years of life (Kuhl, Stevens, Hayashi, Deguchi, Kiritani, and Iverson 2006; Sundara, Polka, and Genesee 2006).
- Infants may be able to discriminate some native contrasts only after exposure to native language input[5] (Narayan, Werker, and Beddor 2010).
- A nonnative contrast that infants show a decline in discriminating can be learned by adult speakers of the same native language after significant exposure to the nonnative language (Tees and Werker 1984).

The cross-linguistic variability in the dimensions of acoustic spaces for phonological contrast and distributions of phonological categories over these spaces, as well as the change in the dimensions and distributions for infants due to language input show that *the phonetics-phonology map must be learned from language input*. It also motivates the need to study the phonetics-phonology map using cross-linguistic data to answer (Q0′).

The empirical evidence that: (i) language learners show decline rather than loss in sensitivity to particular phonetic dimensions, (ii) they can reactivate sensitivity

[5]Based on results like these, an alternative assumption to (6) is that

(7) $\forall \mathscr{L}, \mathscr{L} \supseteq \mathscr{U}.$

based on the idea that sensitivity to some phonetic parameters may become activated only after exposure to language input. We do not take this alternative assumption because there is, to date, little supportive evidence for it. More importantly, a negative result for infant sensitivity to a speech sound contrast is conditional on a given experiment using a given task. A positive result is conditioned in the same way, as well, but shows that, at least under some conditions, infants show sensitivity to the contrast, while a negative result does not imply that infants are not sensitive to the contrast under any conditions.

with later language exposure and training, and (iii) listeners show the ability to use a wide variety of cues in degraded speech[6] suggests that the model of the development of language- and speaker-specific spaces of each language involves *parameter tuning/re-weighting* rather than *parameter selection*. Even in cases where sensitivity to some phonetic parameter may be vanishingly small, the model should assign it a vanishingly small weight rather than remove the parameter from the space.

Note that even for the purposes of studying the phonetic parameter space, we must represent data with a set of initial parameters: this initial set should be exactly $\mathscr{U}$, which we assume to be a superset of the dimensions of $\mathscr{L}$ for any natural tone language $\mathscr{L}$, cf. (6), and which is a subset of the set of all acoustic parameters we could extract from the speech signal. But these are not well-defined lower and upper bounds on $\mathscr{U}$; we cannot know what $\mathscr{U}$ is before studying what it should be! Thus, we make a guess and initialize the parameter set of $\mathscr{U}$ based on cross-linguistic work on tonal production, perception, and automatic tonal recognition.

The distribution (Q1c) We assume that the distribution of phonological categories over the phonetic space is continuous. Since the details of the distribution depends strongly on how the phonological categories and the phonetic space is defined, we let our study of those determine characteristics of the distribution. These characteristics then inform how we constrain the type of distributions available in the hypothesis space for the learner in modeling the actual learning of the representational map.

1.2 Methodological Abstractions

In characterizing the phonetic parameter space (Q1b) for lexical tonal categories in this paper, we make three main methodological abstractions: (i) to sharpen the probabilistic distributions of phonological categories into partitions over the phonetic space, (ii) to use category separability as a metric for constraining the phonetic parameter space, (iii) to limit the context available to extract the phonetic parameters from, and (iv) to introduce linguistic structure into the unanalyzed speech signal. Characterizing the phonetic parameter space with these methodological abstractions in place still allows us to bear on questions (Q1a)–(Q1c).

Partitions over the phonetic space and category separability While the reality is that the phonetics-phonology map is a probabilistic distribution of phonological categories over the phonetic space, in characterizing the phonetic parameter space, we make the methodological abstraction that the map is a *partition* of phonological categories over the space: every point in the space maps to exactly and only one phonological category.

The reason for the abstraction is that most well-understood computational algorithms for classification give "hard" classifications, i.e. produce a partition of the space, rather than a probabilistic distribution over it (Wahba 2002). Moreover, while it is possible to elicit probabilistic confidence ratings in human perception experi-

[6] see Assmann and Summerfield (2004) for a general review of perception of degraded speech.

ments, e.g. using magnitude estimation (Bard, Robertson, and Sorace 1996), we use forced choice tasks in our perception experiments to match the hard classification of the computational algorithms.

Along with the methodological abstraction of modeling the phonetics-phonology map with partitions, we use the general metric of *category separability* to determine how relevant/informative phonetic parameters are for defining the tonal categories: more informative phonetic parameters define a space in which the tonal categories are better separated. As discussed by Nearey (1989), this category separability metric is *data analytic* because it is based on production data only, while ultimately, *perceptual* separability from listening experiments is what is directly relevant for the representational map. However, data analytic category separability certainly bears on perceptual separability.

Limiting context for phonetic parametrization We have already proposed a map (2) restricting the domain to phonetic parameters. We reiterate here that we are abstracting away from non-phonetic context, e.g. morphosyntactic information (the language model in automatic speech recognition), to constrain the research problem; Jansen (2008) calls this the "pure speech" setting. Moreover, we restrict the *temporal domain* for phonetic parameter extraction. The strongest such restriction is to restrict the extraction of phonetic parameters to only the unit to the classified, e.g. only from the syllable of the tone to be classified. In this paper, we start from this restriction, but we will ultimately allow parameter extraction from the preceding and following syllables as well. For fluent speech recognition, there is strong evidence that humans extract parameters from temporal domains wider than the unit to be classified, e.g. Ladefoged and Broadbent (1957); Wong and Diehl (2003).

Introducing linguistic structure in the speech signal While the original research question (Q0′) assumes extraction of parameters from the unanalyzed signal, for this paper, we extract parameters from speech segmented for syllabic structure for convenience. This is like having an oracle tell the classifier where syllable boundaries or onset/rime boundaries are. In future work, we can remove this extra information by implementing a sonority detector to find syllables, as in Jansen (2008).

1.3 The Model System for the Acquisition of Lexical Tones

With the larger research questions and the methodological abstractions set up, we turn to the model system under study.

The gross characterization of our model system is this:

- **Data**: monotones extracted from sentence-medial position in connected speech over cross-linguistic tonal language sample
- **Phonetic parameter space**: acoustic parameter space, extracted from the speech signal
- **Phonological categories**: lexical tonal phonemes (tonemes)

Like any other system studied in phonological category acquisition, the one we study here is a model system, and we study it with the same scientific motivation that a biologist studies a simple model organism like baker's yeast (the eukaryote with the smallest number of genes) to illuminate gene regulation in more complex systems such as humans (Fields and Johnston 2005). Clearly the model system can only capture certain aspects of the process of phonological category acquisition, highlighting some while muting others. In this section, we describe how we instantiate the model system for lexical tone acquisition to answer (Q0′).

Our research questions, as laid out in §1, dictate the following requirements for setting up a model system for studying lexical tone acquisition:

- A representative cross-linguistic sample to address the language-specific development of speech categorization
- A language sample relevant for modeling language input to infants
- Some controlled source(s) of variability to enable modeling the challenge of categorization in the face of variability

Cross-linguistic tone language sample We chose a sample of tonal languages to include: (i) register/level tone languages, with only level tones (Bole, Igbo), and (ii) contour tone languages with contour tones and level tones (Mandarin, Cantonese, Hmong)[7]. We summarize the diversity of the cross-linguistic tonal language sample below in Table 2, using International Phonetic Alphabet notation for the tonal inventory, and give recording details of the data currently available below in Table 3.

Language	Area	Tonal inventory	Phonation
Bole	Nigeria	˥, ˩ (H,L)	
Igbo	Nigeria	˥, ˧, ˩ (H, !H, L)	
Mandarin	Beijing, Taiwan	˥, ˧˥, ˨˩, ˥˩	creaky ˨˩, ˥˩
Cantonese	Hong Kong	˥, ˦, ˧, ˨˩, ˧˥, ˧	creaky ˨˩
Hmong	Laos/Thailand	˥, ˦, ˧, ˥˩, ˩˩, ˨˩, ˧˥	breathy ˩˩, creaky ˨˩

Table 2: Cross-linguistic sample of tonal languages recorded to provide language input

Language	Dialect	Recording location	Speakers
Bole	Fika	Potiskum, Nigeria	3M/2F
Igbo	Anambra	Los Angeles, CA	1M/2F
Mandarin	Beijing	Beijing, China	6M/6F
Mandarin	Taiwan	Los Angeles, CA	6M/6F
Cantonese	Hong Kong/Macau	Los Angeles, CA	6M/6F
Hmong	White	Fresno, CA	6M/5F

Table 3: Details for recordings of language sample

[7]The language sample was also chosen to exhibit a variety of tone-voice quality interactions. While beyond the scope of this paper, our cross-linguistic data and perception experiments suggest that the parameterization of the speech signal for tonal representation must include voice quality parameters, e.g. related to phonation, beyond simple f0-based parameters, cf. Lam and Yu (2010); Yu (2010).

Language input to infants and sources of variability Because infants exhibit perceptual knowledge before articulatory knowledge of speech sound categorization (Kuhl 2004), we restricted parameters to *acoustic parameters* and abstracted away from articulatory parameters.

Other work on learning tonal categories has emphasized that the majority of the input to the infant consists of multiple words so that contexual variation due to tonal coarticulation from neighboring tones is a regular part of the input the learner receives (Gauthier et al. 2007; Shi in press). Specifically, Gauthier et al. (2007); Shi (in press) claim that about 90% of parental speech to infants is multi-word utterances. Moreover, the majority of language data an infant hears is not speech directed to the infant, but, for instance, adult-to-adult speech. An estimate from van de Weijer (1998, 2002) is that only about 14% of the input is direct speech to the infant.

Because of the large amount of input that infants hear that is adult directed speech and multi-word utterances, Gauthier et al. (2007) modeled learning tone categories based on speech from adults rather than infant-directed speech, (and in general, research building tone recognizers is modeled on adult speech). This is of course a working hypothesis; surely the presence of infant directed speech and isolated words in the input could affect the character of the learning problem.[8] We follow this choice, taking our input to the learner to be adult connected speech. We capture the role of contextual tonal variation in creating variability in the input by collecting the full permutation set of bitones in connected speech for each language in the sample, and we capture interspeaker variation by recording multiple speakers of both genders.

This concludes our section on preliminaries, which we have deliberately kept broad in scope to illustrate our model system of lexical tone in context of the study of phonological (and language) acquisition in general. We now turn to describing our exploration of the two issues regarding f0 parametrization discussed in the introduction: coarse temporal resolution in parameterization and static and dynamic parametrizations of f0.

2 The Parametrization of f0 in Representational Maps for Lexical Tone

Gauthier et al. (2007), the only preceding computational modeling study of learning a tonal system (the four basic Mandarin tones), suggests that representing examples to the learner as densely sampled f0 velocity contours results in more robust tonal categories than representing examples as densely sampled f0 contours.

[8] For instance, note that the rationale for the ecological validity of adult connected speech given above assumes equal weighting in infant attention to all input regardless of whether it is directed to the infant. In fact, studies show biases for infant directed speech over adult speech and biases for the infant for their mother's voice and the importance of placing language input within social interaction (Kuhl, Tsao, and Liu 2003). Thus, it is not unreasonable to hypothesize that despite the relatively small amount of infant directed speech in the ambient input, it may be a rich source of information for infants about learning tone patterns. In fact, work has found correlation between the amount of exaggeration in infant directed speech in terms of the expansion of the vowel and tonal spaces in predicting an infant's ability to discriminate native consonant contrasts (Liu, Kuhl, and Tsao 2003; Xu and Burnham submitted).

Moreover, the study suggests that *parametrization of the speech signal as densely sampled f0 velocity contours (f0′) alone is sufficient for learning Mandarin tones*. The intuition for why f0 velocity might be more relevant than f0, and furthermore, sufficient alone for tonal classification, is that the derivative of a constant function is zero: f0 velocity provides a way of speaker normalization, of removing constant shifts due to different pitch ranges.

We generalize this hypothesis as an initialization for $\mathscr{U}$: $\mathscr{U}_G$ is a d-dimensional parameter space defined over d densely, uniformly spaced f0 velocity (f0′) samples from the syllable; each of the d samples contributes a dimension to the space, and the sampling rate is defined over time normalized by the syllable duration, t_{syll}, i.e., a sample taken at timepoint $t_{syll} = i$ is taken at $i/(d-1)$ of the way through the syllable:

$$\mathscr{U}_G = \{\text{f0}'(t_{syll} = i) \mid 0 \leq i \leq d-1, d \text{ “large”}\} \tag{8}$$

$\mathscr{U}_G$ assumes dense temporal sampling resolution and a parameterization including only a dynamic f0-based parameter.

We hypothesize, in contrast, that:

(H1)	*Coarse temporal sampling resolution of the parameterized speech signal is sufficient for good tonal category separability.*
(H2)	*The parametrization of the speech signal as f0 velocity contours is not sufficient for good tonal category separability cross-linguistically.*

2.1 Coarse Temporal Resolution (H1)

Increasing temporal resolution means increasing the dimensionality of the parameter space: each additional sample adds a dimension. Thus, coarse temporal resolution is necessary for a succinct tonal representation, which is desirable for generalization in learning a phonetics-phonology and for scientific understanding, cf. §1.1.

Linguistic models for the representation of tone implicitly advocate coarse temporal resolution. Chao (1930)’s tone letters used in the International Phonetic Alphabet for representing tones, e.g. ˩˧˥, suggest that three samples (and more specifically, three particular samples) over the tone are sufficient, as described in Chao (1968)’s model of Chinese tone systems in his grammar of Chinese:

> If we divide the range of a speaker’s voice into four equal intervals, marked by five points, 1 low, 2 half-low, 3 middle, 4 half-high, and 5 high, then practically any tone occurring in any of the Chinese dialects can be represented unambiguously by noting the beginning and ending points, and, in the case of a circumflex tone, also the turning point; in other words, the exact shape of the time-pitch curve,

> so far as I have observed, has never been a necessary distinctive feature, given the starting and ending points, or the turning point, if any, on the five-point scale. (Chao 1968:25)

The *modus operandi* in speech recognition, though, is to use a constant frame rate, sampling features every 10ms over 30ms windows (Young, Evermann, Gales, Hain, Kershaw, Liu, Moore, Odell, Ollason, Povey, Valtchev, and Woodland 2009), and Gauthier et al. (2007)'s sampling rate (30 samples/syllable) is close to this.

However, a survey of sampling characteristics in the automatic tonal recognition literature suggests that coarse sampling of f0 parameters can yield good performance, as summarized in Table 4 below. In the table, we also indicate the *clock* used for each study, by which we mean which temporal unit was used to define the (uniform) sampling rate. For the studies where we describe the sampling in terms of "slices", this means that features were extracted as averages over the slices, i.e. smoothed.

Study	Language	Clock	Sampling resolution
Zhang and Hirose (2004)	Mandarin	Absolute time	Fine, 10ms frame shift
Gauthier et al. (2007)	Mandarin	Normalized to syllable	Fine, 30 samples/syll
Ọdẹ́lọbí (2008)	Yoruba	Normalized to syllable	Medium, 9 slices/syll
Wang and Levow (2008)	Mandarin	Normalized to tone nucleus	Coarse, 5 samples/nucleus
Qian, Lee, and Soong (2007)	Cantonese	Normalized to rime	Coarse, 3 slices/final
Zhou, Zhang, Lee, and Xu (2008)	Mandarin	Normalized to nucleus	Coarse, 3 slices/nucleus

Table 4: Sampling characteristics of a selection of tone recognition studies

The predominance of coarse sample resolution and linguistically-tied clocks in recent tonal modelling is very striking, compared to the predominance of high frame rate and absolute time in sampling in general speech recognition. Note that no study sampled fewer than 3 times per tonal domain.[9] One automatic tonal recognition study of Mandarin even found that coarse sampling, with 4 samples/tone, outperformed dense sampling with 1 sample/10 ms (Tian, Zhou, Chu, and Chang 2004).

In summary, long-standing linguistic intuition and evidence from recent large-scale automatic tonal recognition studies converge to suggest that coarse sampling is sufficient in parametrization of tonal spaces. In our research, we confirm this with experimental and computational modeling work: (i) a human tonal perception experiment studying the effect of sampling resolution on Cantonese tonal perception and (ii) computational studies of the effect of sampling resolution on category separability over our cross-linguistic tonal sample.

[9]For Mandarin at least, the reason why is hinted at already in Chao (1968): "practically any tone occurring in any of the Chinese dialects can be represented unambiguously by noting the beginning and ending points, and, in the case of a circumflex tone, also the turning point." Two f0 feature samples is not sufficient to distinguish Tone 2 (rise) and Tone 3 (fall-rise) in isolation. Zhou et al. (2008) empirically studied this in their multilayer perceptron Mandarin tone recognizer: in varying the number of inputs to the neural network, they found that percent correct saturated after the number of inputs was increased from 2 to 3, and that the improvement was due to improvements in classification from Tone 3.

2.2 *Insufficiency of f0 Velocity Contours (H2)*

The second hypothesis is that f0 velocity contours, regardless of sampling resolution, are insufficient for good separability of tonal categories. The obvious counterexamples to an initialization $\mathcal{U}_G$ in (8) are a level/register tone language and a tone language with level and contour tones. The Mandarin tonal inventory that is the target of learning in Gauthier et al. (2007) is unusual in having no level tone contrasts.

We note that the level tone counterexample is not trivial, i.e. it is not enough to reject $\mathcal{U}_G$ with a thought experiment. Level tone sequences are not a series of step functions, but may in fact be realized as if they are contour tone sequences due to contextual tonal variation, cf. Figure 3 and Maddieson (1977:337).

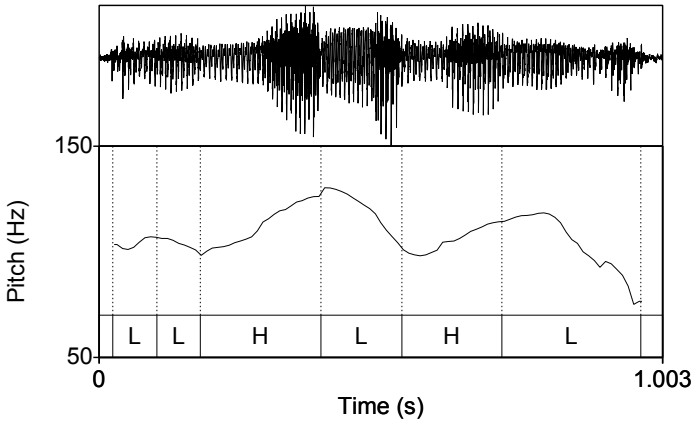


Figure 3: A sequence of tones in Bole, a tone language with H and L tones. Sequences of level tones in a level tone language are not necessarily sequences of step functions. Rather, they can show rises and falls due to tonal coarticulation. The sentence is *ànìn némà méngò*, 'The owners of prosperity came back.'

If $\mathcal{U}_G$ was the structure of the universal parameter space for tones, we might expect many tonal systems to consist of purely dynamic contrasts. In fact, a striking typological pattern in tonal inventories is that two-tone systems of this kind are not known to exist, as noted as early as the 1960s:

> The simplest language of [a pure contour tone system] would have two tonemes, one a glide upwards and one a glide downwards, with the level of the end points of complete irrelevance to the system. Here the contrast would be that of a rising contour opposed to a falling contour. No system this simple has come to my attention. (Pike 1964:9)

Instead, the dominant tonal system is an inventory of two level tones, H and L, like Bole in our language sample; in the statistical sample of tone languages in Maddieson (1978), about half of the languages are of this type.

To test the relevance of f0 velocity contours, we compare tonal category separability in our modeling using: (i) only f0 velocity (ii) only absolute f0, and (iii) both

f0 velocity and absolute f0. Given our goal of modeling human cognition, it would be useful to study how human tonal perception proceeds when only f0 velocity cues are present. However, factoring out all cues in the speech signal, including f0 height, except f0 velocity is not possible, although attempts of this kind have been made by psychophysicists (Dooley and Moore 1988; Divenyi 2004). Thus, we confine our studies bearing on (H2) to computational modeling studies.

3 Experimental and Computational Studies Bearing on the Parameterization of f0 in Tonal Spaces

In this section, we briefly summarize results from our own experimental and computational work bearing on the hypotheses H1 and H2. First, we discuss experimental results showing that human listeners can maintain tonal identification accuracy with stimuli degraded to be coarsely sampled (§3.1). Then we discuss exploratory computational studies of the parameterization of tonal spaces using coarse and dense sampling of absolute f0 and f0 velocity (§3.2).

3.1 Coarse Temporal Sampling and Human Tonal Perception

In a Cantonese tonal perception experiment in which we manipulated the sampling resolution in the stimuli presented to the listener, we showed that tonal identification accuracy under coarse temporal sampling down to 3 samples/syllable can be as high as accuracy with the intact signal.

Cantonese tritones $\langle wai\dashv, \{wai\urcorner, \nearrow, \dashv, \searrow, \nearrow, \dashv\}, mat\dashv \rangle$ extracted from connected speech by multiple speakers (3 M, 2 F) were presented to 39 native Cantonese listeners in sound-attenuated booths at City University of Hong Kong and UCLA.[10] The listeners were asked to identify the second tone in the tritone by a key press of the corresponding orthographic label. Sampling resolution varied from the intact signal, to 7, 5, 3, and 2 30.4-ms uniformly spaced samples (time-slices) per syllable. The stimuli were blocked by sampling resolution, and block order was pseudorandomized to be roughly uniformly distributed over sampling resolution.

The sampling resolution manipulation involved intermittently replacing the speech with noise 10dB higher than the signal amplitude, as in multiple phonemic restoration (Bashford, Riener, and Warren 1992; Miller and Licklider 1950), cf. Figure 4, using Matlab and Praat (Boersma and Weenink 2010).

A repeated measures ANOVA with SAMPLING RESOLUTION as a fixed effect and SUBJECT as a random effect showed a main effect for SAMPLING RESOLUTION: $F(4,152) = 28.6, p < 2.2 \times 10^{-16}$. Bonferroni corrected pairwise comparisons with the family-wise Type I error rate at 0.05 showed significant differences between the 2-sample condition and all other conditions, and between the 3-sample condition and the 7-sample and intact conditions. Thus, on average, listeners were able to maintain tonal identification accuracy down to 5 samples/syllable, and also, to some degree,

[10] Tritones rather than monotones or bitones were used to preclude a floor effect washing out any differences between sampling resolution conditions.

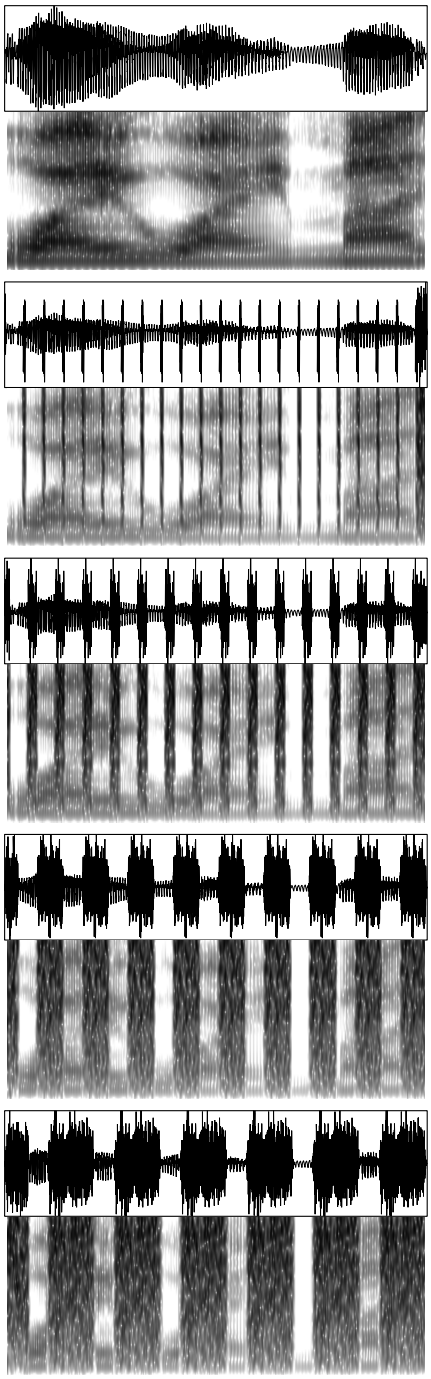


Figure 4: Waveforms and spectrograms of sample stimuli for sampling resolution from intact, to 7, 5, 3, and 2 samples/syllable over Cantonese tritones

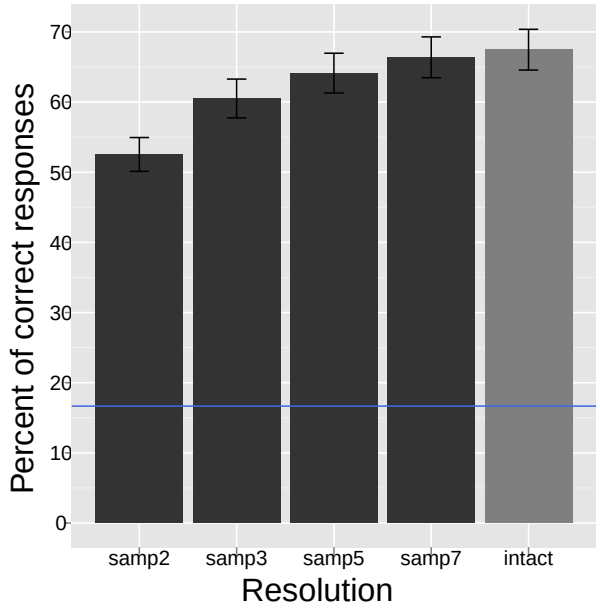


Figure 5: Comparison of tonal identification accuracy for different sampling resolutions. Tonal identification accuracy was maintained from the intact signal down to 3 samples/syllable. For all sampling resolutions, performance was also well-above chance (the blue line shows identification accuracy for at-chance performance (1/6)), and the error bars show ±1SE.

down to 3 samples/syllable, but not down to 2 samples/syllable, cf. Table 5 and Figure 5.

Resolution	Percent correct (SE)
samp2	52.54 (2.41)
samp3	60.51 (2.76)
samp5	64.13 (2.83)
samp7	66.38 (2.91)
intact	67.46 (2.9)

Table 5: Tonal identification accuracy for different sampling resolutions averaged over the listeners.

The Cantonese tonal perception results therefore support the hypothesis that coarse temporal resolution may be sufficient for good tonal category separability. However, the experimental results do not inform us as to what cues the listeners are using in those few samples to identify the tones with reasonably high accuracy.

3.2 Computational Modeling and the Parametrization of f0

In this section, we briefly summarize results from initial computational modeling bearing on hypotheses H1 and H2, regarding category separability under dense and

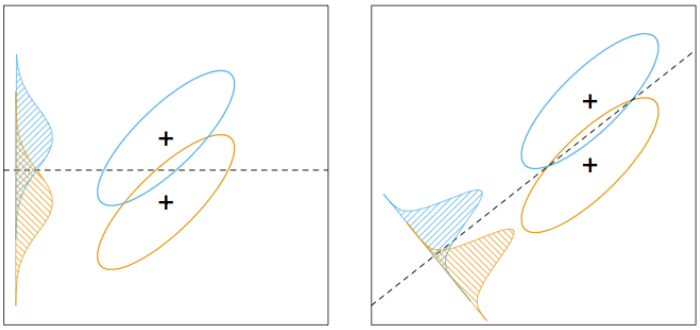


Figure 6: A geometric characterization of linear discriminant analysis (LDA) for a two-class problem, from Hastie et al. (2009:116). The objective is to maximize the ratio of the between-class variance to the within-class variance. Thus, though the projection in the left panel along the direction of the line connecting the centroids maximizes the between-class variance, the within-class variance is high and there is large overlap in the two classes. The projection on the right minimizes the ratio of between- to within-class variance and is the projection chosen in LDA.

coarse sampling, and the insufficiency of f0 velocity contours for good category separability. As an initial measure of category separability, we choose linear discriminant analysis to aid in exploratory visualization of the multidimensional parameter space.

3.2.1 *Linear Discriminant Analysis (LDA)*

Linear discriminant analysis is both a dimensionality reduction technique and a classification algorithm (Hastie, Tibshirani, and Friedman 2009:§4.3). As a dimensionality reduction technique, it chooses a projection of the data into a smaller-dimensional space such that the projection maximizes category (class) separability, where the class separability is measured as the ratio of the between-class variance (the variance of the projected class means) to the within-class variance in the projected data (the pooled variance about these means), cf. Figure 6.

As a classification algorithm, it defines a partition of the space by estimating linear decision boundaries and classifies an observation into the class with the nearest centroid, measured by Mahalanobis distance (a distance metric that is covariance-adjusted). Under strong (and typically false) assumptions about the distribution of the data, namely, that the distribution of data within each class is multivariate Gaussian with a common covariance matrix, linear discriminant analysis is equivalent to a Bayesian classifier (Hastie et al. 2009:439).

We are primarily interested in using linear discriminant analysis for the purposes of exploratory visualization of data in low dimensions because our data, unsurprisingly, fail to satisfy the assumption of multivariate normality with common covariance matrices, and because we are interested in trying methods that allow more complex decision boundaries than linear decision boundaries.

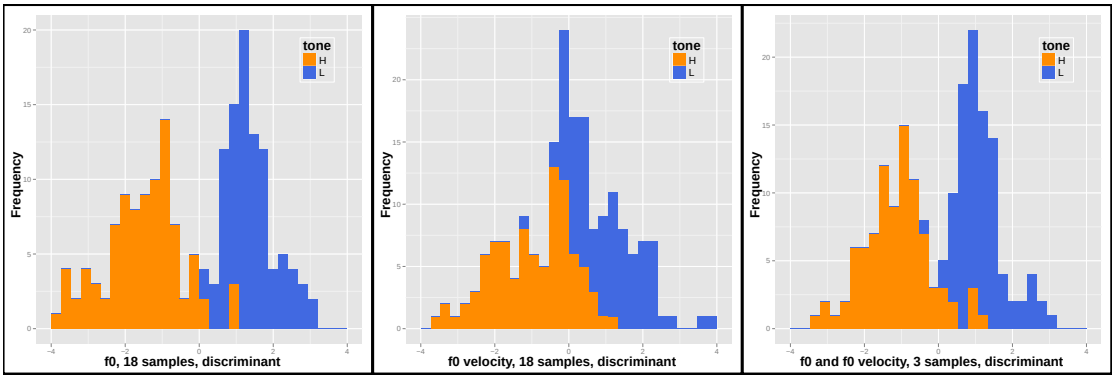
3.2.2 *Category Separability Under Coarse and Dense Sampling of f0-based Parameters*

Our initial modeling extracts parameters from only the tone to be classified, without any contextual information from neighboring tones, and we examine category separability for tonal spaces from single speakers.

By using LDA as implemented in R (R Development Core Team 2010) by Venables and Ripley (2002) to visualize our data, we compared category separability under coarse and dense sampling of: (i) absolute f0, (ii) f0 velocity, and (iii) both absolute f0 and f0 velocity. We calculated f0 averaged over coarse and finely divided uniform subsections (time slices) of the syllable using VoiceSauce (Shue, Keating, and Vicenik 2009) and calculated f0 velocity by taking differences between these averaged f0 values in R. To avoid linearly dependent parameters in parameter sets including both absolute f0 and f0 velocity, we calculated absolute f0 and f0 velocity separately with differing coarseness of subsection division, since the number of f0 velocity samples calculated from some number of absolute f0 samples is necessarily less than the number of absolute f0 samples.

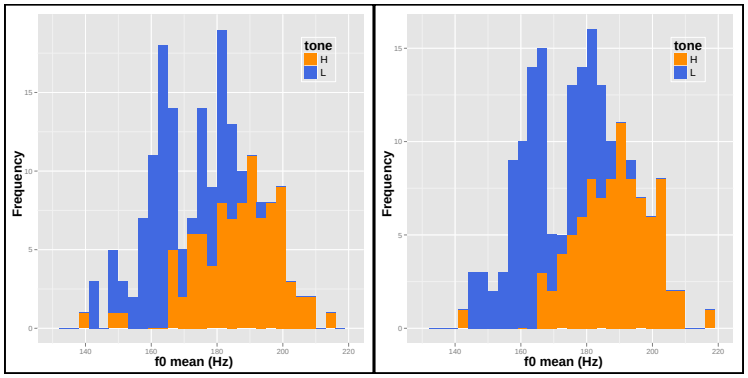
In general, our exploratory results suggest that for phonetic parameterization without any contextual parameters in single speaker spaces, coarse sampling (3 samples each) of absolute f0 and f0 velocity is sufficient for good category separability, and critically, category separability for coarse sampling of absolute f0 and f0 velocity is comparable or better than dense sampling (18 samples) of f0 velocity. Below, we exemplify our results with two examples.

In Figure 7, we show the phonetic parameter space for a Bole female speaker for densely sampled f0 velocity (18 samples), densely sampled absolute f0 (18 samples), and coarsely sampled absolute f0 and f0 velocity (3 samples each) after LDA dimensionality reduction. It is clear that category separability for densely sampled f0 velocity (Figure 7b) is poorest, while category separabilty for coarsely sampled absolute f0/f0 velocity (Figure 7c) is comparable to that for densely sampled absolute f0 (Figure 7a).



(a) Densely sampled absolute f0 (b) Densely sampled f0 velocity (c) Coarsely sampled absolute f0 and f0 velocity

Figure 7: The separability of Bole tones (H, L) for a single female speaker with coarse f0 and f0 velocity is similar to that with dense f0 sampling and better than with dense f0 velocity sampling.



(a) Mean absolute f0 (b) 1 absolute f0 sample

Figure 8: Category separability of Bole tones for a single female speaker with mean absolute f0 only (8a), or one sample of f0 at the midpoint from a 9-sample subdivison of the syllable (8b).

It is important to note that f0 velocity provides any evidence of category separability at all: this shows that it may be a relevant parameter for even the simplest of level tone systems, and more generally, that dynamic properties of f0 are relevant for level tones, contrary to characterizations of level tones that suggest one f0 height sample is enough to specify a level tone, while contour tones require multiple samples:

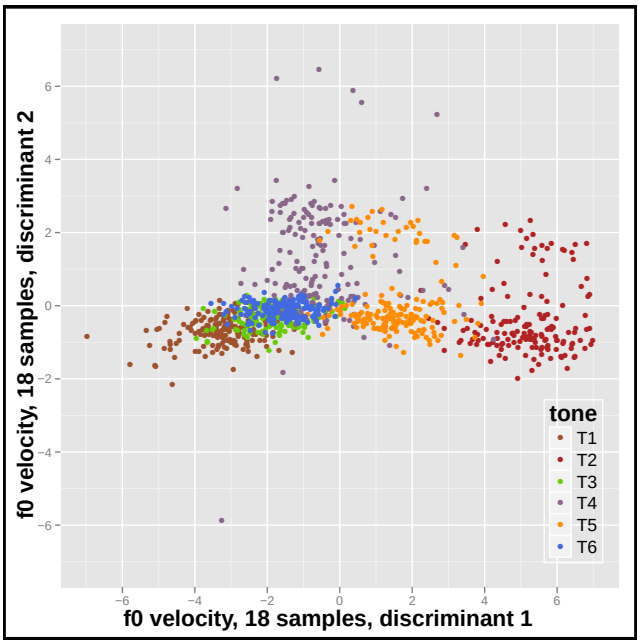
> If an adequate synthesis of a tone can be made by specifying a single level, it may be considered a level tone. But a tone represented by a pitch glide which cannot be generated by rule from the environment (i.e. not by a default) requires specification of several points. (Maddieson 1977:337)

In fact, category separation for the same speaker, using either only mean absolute f0 or one sample of absolute f0 from the midpoint is poor, cf. Figure 8, and the bimodal distribution of the L tone suggests latent category structure not captured by the 1-dimensional parametrization, at least without a relational parameterization of f0.

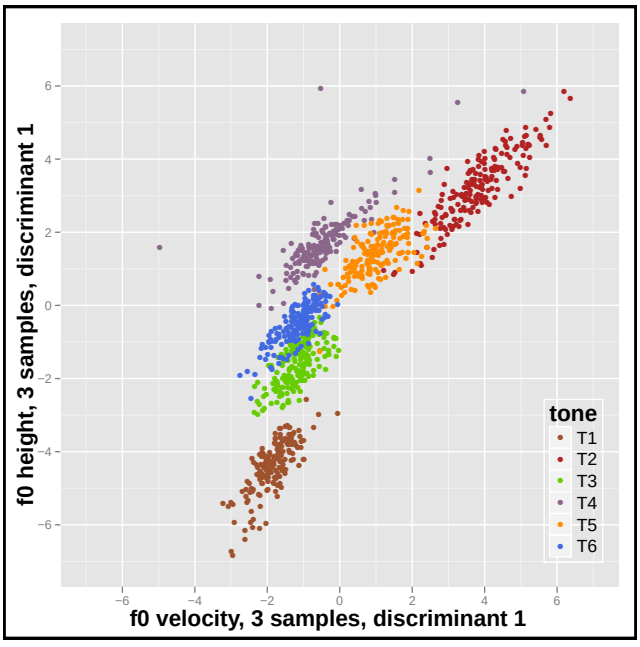
Results in a tonal system with many contours and levels, such as Cantonese, are even stronger: with densely sampled f0 velocity contours (18 samples), the three level tones show gross overlap, cf. Figure 9a. However, with coarse sampling of absolute f0 and f0 velocity (3 samples each), cf. Figure 9b, a near partition of the tonal categories, levels and contours, is obtained.

These results support Hypothesis H2: f0 velocity, regardless of density of sampling, is not sufficient for good category separability across all languages. However, coarse sampling of relevant parameters—both absolute f0 and f0 velocity—results in a near-partition of the phonetic parameter space.

While highly preliminary, our modeling results using linear discriminant analysis suggesting that coarse sampling of relevant features suffice for good category separability converge with results from automatic tonal recognition (Tian et al. 2004) and our own experimental work (§3.1), and extends those results to a larger range of languages.



(a) Densely sampled f0 velocity



(b) Coarsely sampled absolute f0 and f0 velocity

Figure 9: Category separability for the parameter space for a single male speaker of Cantonese. Cantonese level tones (Tones 1, 3, 6) cannot be separated with f0 velocity information alone, but coarsely sampled absolute f0 and f0 velocity parameterization results in a near-partition of the phonetic space.

We are currently implementing studies on the parameterization of the speech signal using multiclass support vector machines (Crammer and Singer 2001) and maximum entropy methods (Pietra, Pietra, and Lafferty 1997). Support vector machines are well-understood algorithms that, like LDA, calculate optimal separating hyperplanes (linear decision boundaries, i.e. lines in 2-D spaces) over a parameter space for separating classes. Moreover, they effectively allow the calculation of more complex, nonlinear decision boundaries in the parameter space by efficiently calculating optimal separating hyperplanes in higher-dimensional parameter spaces. Maximum entropy methods have the desirable property that the search space for optimizing the parameter set is convex. This means that we can avoid getting trapped in local minima in the search space, i.e. on a hypothesis for the characterization of the parameter set in which all small movements in the search space result in a less optimal hypothesis.

Conclusion

In conclusion, our work thus far suggests that coarse temporal sampling resolution of the parametrized speech signal is sufficient for good tonal category separability, given that relevant, informative parameters are sampled. This is supported by exploratory modeling using linear discriminant analysis across the level and contour languages of our sample and by the maintenance of tonal identification accuracy under stimuli degraded to be coarsely sampled in a Cantonese tonal perception experiment.

Our modeling work also suggests that the parametrization of the speech signal as f0 velocity contours alone is not sufficient for good tonal category separability cross-linguistically. The strongest evidence for this is the immense overlap of level tone categories in mixed level tone/contour tone systems in a phonetic parameter space defined only over f0 velocity, even for a single speaker. Category separability in supervised learning gives an upper bound for category separabilty in unsupervised learning; the failure of Cantonese level tones to be separated in a pure f0 velocity space with LDA therefore implies that linear clustering methods cannot succeed, either. Thus, Gauthier et al. (2007)'s suggestion that f0 velocity is sufficient for learning tones cannot be generalized cross-linguistically, although we point out that f0 velocity is a relevant parameter for level tone systems, a reflection of the fact that level tones can be realized as contours due to tonal coarticulation.

Our future work will continue to home in on a definition of the target of learning in the acquisition of tones from the speech signal. With a definition of tones that is well-motivated by what we know about human tonal production and perception, we can model how lexical tones could be learned from the speech signal in a way consistent with what we know about human cognition.

Acknowledgements

We would like to acknowledge Edward Stabler, and Meaghan Fowlie, Thomas Graf, Bruce Hayes, Patricia Keating, Russell Schuh, Megha Sundara, Colin Wilson, Kie Zuraw, and audiences at the UCLA Phonetics Seminar and NELS41 for illuminating discussions and feedback, Hiu Wai Lam for testing the perception experiment subjects, Eric Zee for letting us run experiments in his laboratory at City University of Hong Kong, all those who helped with linguistic consultation and recordings: Russell Schuh and Alhaji Maina Gimba (Bole), Echezona Okoli and Moses Chikwe (Igbo), Mandarin (Jianjing Kuang), Hiu Wai Lam, Shing Yin Li, Cindy Chan, Cedric Loke, and Vincie Ho (Cantonese), Chou Khang and Phong Yang (White Hmong), and all the dedicated undergraduates that helped with phonetic data processing: Samantha Chan, Chris Fung, Hiu Wai Lam, Wai-Ting Lam, Antonio Sou, Kristen Toda, Grace Tsai, and Joanna Wang. This work was supported by a NSF graduate fellowship, a Ladefoged Scholarship from the UCLA Department of Linguistics, and NSF grant IIS-1018863 to Abeer Alwan, Patricia Keating, and Jody Kreiman. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

A The phonetics-phonology map learning problem in formal learning theory

As stated in Footnote 2 on page 5, it is possible to define a discrete phonetics-phonology map and thus study the phonological categorization problem using formal learning theory because we can represent the real-valued speech signal digitally to arbitrary precision in the limit.

We assume the phonetics-phonology map is a *partition* of a finite set of phonological categories, *Cat*, over the phonetic parameter space:

Phonetics-Phonology:

(9) {sequences of phonetic parameter vectors} → {phonological categories}

Choose any phonetic parameter vector $\vec{v} \in \mathbb{R}^n$, for finite n, e.g. $\langle F1, F2 \rangle \in \mathbb{R}^2$. Digitize $\vec{v}$ with an n-bit quantization, where $n \in \mathbb{Z}$, and sample at some sampling rate s.[11,12] Then at each timepoint t of sampling, each entry of $digitized(\vec{v})(t)$ is in *PnF*, where *PnF* is the finite set of 2^n different symbols from the n-bit quantization.

Consider the language L which is a set of pairs: $\langle p, c \rangle$ where $p \in PnF$, $c \in Cat$, and assume L is in the class of r.e. languages. We would like to show that the class of such languages, $\mathscr{L}_{PnPh}$, is learnable by constructing a learner $\phi : (\mathbb{Z}^n \times Cat) \mapsto \mathscr{G}$ to map from the class of languages to a class of grammars.

By assumption that the phonetics-phonology map is a partition, (i) each p is paired with a unique c, i.e. L is a function (single-value language), and (ii) $\forall p$, p is mapped to some c, so this function is total. Thus, $L \in \mathscr{L}_{svt}$, the class of total

[11] Note that discrete phonetic parameterization (beyond digital speech processing) is not unusual, e.g. fundamental frequency is often parametrized as 5-valued (Chao 1930).

[12] This representation of the speech signal is based on Pierrehumbert (1990:379).

single-value languages and therefore the class of languages of phonetic parameter vector-phonological category maps is identifiable from positive data (Jain et al. 1999).

The learner for $\mathscr{L}_{svt}$ in Jain et al. (1999) is not computable as it relies on an enumeration of the grammars of all r.e. languages. Thus, even though recognizing that $\mathscr{L}_{PnPh} \subseteq \mathscr{L}_{svt}$ is sufficient for proving it is learnable (in the Gold sense), we may argue that this learnability result doesn't reveal the structure of the learning problem for phonetics-phonology maps.

However, we might also argue that the particular class of languages relevant for the phonetics-phonology map is a *proper subset* of the class of total single-value languages: $\mathscr{L}_{PnPh} \subset \mathscr{L}_{svt}$. In particular, we can assume a fixed finite phonetic parameter set for $\vec{v}$, a fixed n-bit depth for quantization, a fixed sampling rate s, and a finite, fixed set of phonological categories Cat. With these fixed bounds, $\mathscr{L}_{PnPh}$ is a finite subset of the finite languages. Since $\mathscr{L}_{PnPh}$ has finite cardinality, the VC dimension of this class is also finite and thus $\mathscr{L}_{PnPh}$ is PAC-learnable: the class of languages for phonetics-phonology maps is both computable and tractable.

Even with this result for computability and feasibility, we don't pursue a finite model for the phonetics-phonology learning problem. Although there may be grounds to model the speech signal with a discrete representation based on finiteness in the number of distinctions that human sensory systems can draw, that finiteness is vast: the cardinality of $\mathscr{L}_{PnPh}$, even if finite, is of a vastness of the order of magnitude so that idealization of phonetic parameterization as being real-valued and thus in an infinite space is appropriate.

Rather than assuming finite bounds and concluding that phonetics-phonology maps are learnable *unconditioned on the choice of phonetic parameterization, as long as the parametrization is finite*, we model the speech signal in $\mathbb{R}^n$ to impose structure on the vast hypothesis space for learning phonetics-phonology maps.

References

Assmann, Peter, and Quentin Summerfield. 2004. The perception of speech under adverse conditions. In *Springer handbook of auditory research*, volume 18, 231–308. New York: Springer-Verlag.

Bard, Ellen Gurman, Dan Robertson, and Antonella Sorace. 1996. Magnitude estimation of linguistic acceptability. *Language* 72:32–68.

Bashford, James A., Keri R. Riener, and Richard M. Warren. 1992. Increasing the intelligibility of speech through multiple phonemic restorations. *Perception and psychophysics* 51:211–217.

Blum, Lenore. 2004. Computing over the reals: where Turing meets Newton. *Notices of the American Mathematical Society* 51:1024–1034.

Blum, Lenore, Felipe Cucker, Michael Shub, and Steve Smale. 1997. *Complexity and real computation*. Springer.

de Boer, Bart, and Patricia K. Kuhl. 2003. Investigating the role of infant-directed speech with a computer model. *Acoustics Research Letters Online* 4:129–134.

Boersma, Paul, and David Weenink. 2010. Praat: doing phonetics by computer (version 5.1.32) [computer program]. `http://www.praat.org`.

Chao, Yuen-Ren. 1930. A system of tone-letters. *Le Maître Phonétique* 45:24–27.

Chao, Yuen Ren. 1968. *A grammar of spoken Chinese*. Berkeley, CA: University of California Press.

Crammer, Koby, and Yoram Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research* 2:265–292.

Dillon, Brian, Ewan Dunbar, and William Idsardi. Unpublished. A single stage approach to learning phonological categories: insights from Inuktitut .

Divenyi, Pierre L. 2004. Frequency change velocity and acceleration detector: a bird or a red herring? In *Auditory signal processing: physiology, psychoacoustics, and models*, ed. de Cheveigné A. McAdams S. Pressnitzer, D. and L. Collet, 176–184. Spring-Verlag.

Ọdẹ́lọbí, Ọdẹ́túnjí Àjàdí. 2008. Recognition of tones in Yorùbá speech: Experiments with artificial neural networks. In *Speech, audio, image and biomedical signal processing using neural networks*, 23–47. Berlin: Springer.

Dooley, Gary J., and Brian C. J. Moore. 1988. Duration discrimination of steady and gliding tones: A new method for estimating sensitivity to rate of change. *The Journal of the Acoustical Society of America* 84:1332–1337.

Duda, Richard O., Peter E. Hart, and David G. Stork. 2001. *Pattern classification*. John Wiley & Sons, Inc., 2nd edition.

Dyson, Freeman. 2004. A meeting with Enrico Fermi. *Nature* 427:297.

Fields, Stanley, and Mark Johnston. 2005. Whither model organism research? *Science* 307:1885–1886.

Gauthier, Bruno, Rushen Shi, and Yi Xu. 2007. Learning phonetic categories by tracking movements. *Cognition* 103:80–106.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The elements of statistical learning*. Springer, second edition.

Hyman, Larry M. 2010. Do tones have features? *UC Berkely Phonology Lab Annual Report* 1–20.

Jain, Sanjay, Daniel Osherson, James S. Royer, and Arun Sharma. 1999. *Systems that learn: An introduction to learning theory (second edition)*. Cambridge, Massachusetts: MIT Press.

Jansen, Aren. 2008. Geometric and landmark-based approaches to speech representation and recognition. Doctoral Dissertation, The University of Chicago.

Kuhl, Patricia K. 2004. Early language acquisition: cracking the speech code. *Nat Rev Neurosci* 5:831–843.

Kuhl, Patricia K, Erica Stevens, Akiko Hayashi, Toshisada Deguchi, Shigeru Kiritani, and Paul Iverson. 2006. Infants show a facilitation effect for native language phonetic perception between 6 and 12 months. *Developmental Science* 9:F13–F21.

Kuhl, Patricia K., Feng-Ming Tsao, and Huei-Mei Liu. 2003. Foreign-language experience in infancy: Effects of short-term exposure and social interaction on phonetic learning. *Proceedings of the National Academy of Sciences of the United States of America* 100:9096–9101.

Ladefoged, Peter, and D. E. Broadbent. 1957. Information conveyed by vowels. *The Journal of the Acoustical Society of America* 29:98–104.

Lam, Hiu Wai, and Kristine M. Yu. 2010. The role of creaky voice in Cantonese tonal perception. In *159th Meeting of Acoustical Society of America, April 2010*.

Lin, Ying. 2005. Learning features and segments from waveforms: a statistical model of early phonological acquisition. Doctoral Dissertation, University of California Los Angeles.

Lin, Ying, and Jeff Mielke. 2008. Discovering place and manner features: What can be learned from acoustic and articulatory data. *University of Pennsylvania Working Papers in Linguistics* 14.

Liu, Huei-Mei, Patricia K. Kuhl, and Feng-Ming Tsao. 2003. An association between mothers' speech clarity and infants' speech discrimination skills. *Developmental Science* 6:F1–F10.

MacKay, David. 2003. *Information theory, pattern recognition and neural networks*. Cambridge University Press.

Maddieson, Ian. 1977. Universals of tone. In *Universals of human language: Volume 2 phonology*, ed. Joseph H. Greenberg, Charles A. Ferguson, and Edith A. Moravcsik. Stanford University Press.

Maddieson, Ian. 1978. The frequency of tones. *UCLA Working Papers in Phonetics* 41:43–52.

Mattock, Karen, Monika Molnar, Linda Polka, and Denis Burnham. 2008. The developmental course of lexical tone perception in the first year of life. *Cognition* 106:1367–1381.

Meyer, A.R., and M.J. Fischer. 1971. Economy of description by automata, grammars, and formal systems. In *12th Annual IEEE Symposium on Switching and Automata THeory*, 188–191.

Mielke, Jeff. 2008. *The emergence of distinctive features*. Oxford University Press.

Miller, George A., and J. C. R. Licklider. 1950. The intelligibility of interrupted speech. *The Journal of the Acoustical Society of America* 22:167–173.

Minsky, Marvin, and Seymour Papert. 1971. Progress report on artificial intelligence. http://web.media.mit.edu/ minsky/papers/PR1971.html.

Narayan, Chandan R., Janet F. Werker, and Patrice Speeter Beddor. 2010. The interaction between acoustic salience and language experience in developmental speech perception: Evidence from nasal place discrimination. *Developmental Science* 13:407–420.

Nearey, Terrance M. 1989. Static, dynamic, and relational properties in vowel perception. *The Journal of the Acoustical Society of America* 85:2088–2113.

Peperkamp, Sharon, Rozenn Le Calvez, Jean-Pierre Nadal, and Emmanuel Dupoux. 2006. The acquisition of allophonic rules: Statistical learning with linguistic constraints. *Cognition* 101:B31–B41.

Peterson, Gordon E., and Harold L. Barney. 1952. Control methods used in a study of the vowels. *The Journal of the Acoustical Society of America* 24:175–184.

Pierrehumbert, Janet. 2003a. Probabilistic phonology: Discrimination and robustness. In *Probability theory in linguistics*, ed. Rens Bod, Jennifer Hay, and Stefanie Jannedy, 177–228. The MIT Press.

Pierrehumbert, Janet B. 1990. Phonological and phonetic representation. *Journal of Phonetics* 375–394.

Pierrehumbert, Janet B. 2003b. Phonetic diversity, statistical learning, and acquisition of phonology. *Language and Speech* 46:115–154.

Pietra, Stephen Della, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions Pattern Analysis and Machine Intelligence* 19:1–13.

Pike, Kenneth L. 1964. *Tone languages*. University of Michigan, Ann Arbor.

Polka, Linda, Peter W. Jusczyk, and Susan Rvachew. 1995. Methods for studying speech perception in infants and children. In *??????*, 49–89. ??/.

Polka, Linda, and Janet F. Werker. 1994. Developmental changes in perception of nonnative vowel contrasts. *Journal of Experimental Psychology: Human Perception and Performance* 20:421–435.

Qian, Yao, Tan Lee, and Frank K. Soong. 2007. Tone recognition in continuous cantonese speech using supratone models. *The Journal of the Acoustical Society of America* 121:2936–2945.

R Development Core Team. 2010. R: A language and environment for statistical computing. `http://www.R-project.org` ISBN 3-900051-07-0, Vienna, Austria.

Shi, Rushen. in press. Contextual variability and infants' perception of tonal categories. *Chinese Journal of Phonetics* .

Shue, Yen-Liang, Patricia Keating, and Chad Vicenik. 2009. VOICESAUCE: a program for voice analysis. *The Journal of the Acoustical Society of America* 126:2221.

Sundara, Megha, Linda Polka, and Fred Genesee. 2006. Language-experience facilitates discrimination of /d-/ in monolingual and bilingual acquisition of english. *Cognition* 100:369–388.

Tees, Richard C., and Janet F. Werker. 1984. Perceptual flexibility: maintenance or recovery of the ability to discriminate non-native speech sounds. *Canadian Journal of Psychology* 38:579–590.

Tian, Ye, Jian-Lai Zhou, Min Chu, and E. Chang. 2004. Tone recognition with fractionized models and outlined features. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 1, I–105–8 vol.1.

Toscano, Joseph C., and Bob McMurray. 2010. Cue integration with categories: Weighting acoustic cues in speech using unsupervised learning and distributional statistics. *Cognitive Science* 34:434–464.

Vallabha, Gautam K., James L. McClelland, Ferran Pons, Janet F. Werker, and Shigeaki Amano. 2007. Unsupervised learning of vowel categories from infant-directed speech. *Proceedings of the National Academy of Sciences* 104:13273–13278.

Venables, W. N., and B. D. Ripley. 2002. *Modern applied statistics with S*. Springer, fourth edition.

Wahba, Grace. 2002. Soft and hard classification by reproducing kernel hilbert space methods. *Proceedings of the National Academy of Sciences of the United States of America* 99:16524 –16530.

Wang, Siwei, and Gina-Anne Levow. 2008. Mandarin Chinese tone nucleus detection with landmarks. In *Proceedings of Interspeech 2008*, 1101–1104.

van de Weijer, Joost. 1998. Language input for word discovery. Doctoral Dissertation, Katholieke Universiteit Nijmegen, Nijmegen, The Netherlands.

van de Weijer, Joost. 2002. How much does an infant hear in a day? In *Proceedings of the GALA2001 Conference on Language Acquisition*, 279–282.

Welmers, Wm. E. 1973. *African language structures*. University of California Press.

Werker, Janet F., Rushen Shi, Renee Desjardins, Judith E. Pegg, and Linda Polka. 1998. Three methods for testing infant speech perception. In *Perceptual development: visual, auditory, and speech perception in infancy*, ed. A. M. Slater, 389–420. UCL Press.

Werker, Janet F., and Richard C. Tees. 1984. Cross-language speech perception: evidence for perceptual reorganization during the first year of life. *Infant Behavior and Development* 7:49–63.

Widdows, Dominic. 2004. *Geometry and meaning*. CSLI.

Wong, Patrick C. M., and Randy L. Diehl. 2003. Perceptual normalization for inter- and intratalker variation in cantonese level tones. *Journal of Speech, Language & Hearing Research* 46:413–421.

Xu, Nan, and Denis Burnham. submitted. Tone hyperarticulation in Cantonese infant-directed speech. *Developmental Science* .

Xu, Yi. 1997. Contextual tonal variations in Mandarin. *Journal of Phonetics* 25:61–83.

Young, Steve, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Garethm Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. 2009. *The HTK book*. Cambridge University Engineering Department, 3.4 edition.

Yu, Kristine M. 2010. Laryngealization and features for Chinese tonal recognition. In *INTERSPEECH-2010*.

Zhang, Jinsong, and Keikichi Hirose. 2004. Tone nucleus modeling for chinese lexical tone recognition. *Speech Communication* 42:447–466.

Zhou, Ning, Wenle Zhang, Chao-Yang Lee, and Li Xu. 2008. Lexical tone recognition with an artificial neural network. *Ear and hearing* 29:326–335. PMC2562432.

Affiliation

Kristine M. Yu
Department of Linguistics
University of California, Los Angeles
krisyu@ucla.edu