ELSEVIER

Theoretical Computer Science 293 (2003) 345–363

Theoretical Computer Science

www.elsevier.com/locate/tcs

Structural similarity within and among languages

Edward P. Stabler*, Edward L. Keenan

University of California, Los Angeles, CA, USA

Abstract

Linguists rely on intuitive conceptions of structure when comparing expressions and languages. In an algebraic presentation of a language, some natural notions of similarity can be rigorously defined (e.g. among elements of a language, equivalence w.r.t. isomorphisms of the language; and among languages, equivalence w.r.t. isomorphisms of symmetry groups), but it turns out that slightly more complex and nonstandard notions are needed to capture the kinds of comparisons linguists want to make. This paper identifies some of the important notions of structural similarity, with attention to similarity claims that are prominent in the current linguistic tradition of transformational grammar. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Invariant; Grammar; Linguistic universals

1. Introduction

Linguists have found it useful to employ various notions of structural similarity in the description of human languages:

(1) *Similar structures in a given language.* It is a natural idea that the following two sentences of English have the same structure:
(a) Bill laughs, (b) Sam cries.

(2) *Similar structures in different languages.* Suppose that your English differs from mine solely in that *Sam* is not a name in your language. In this case, it is still natural to assume that the previous two sentences, in your language and mine, respectively, have the same structure. Clearly language change involves, in part, many minor extensions of this kind.

(3) *Similar categories.* It is often proposed that all human languages have some "categories" in common, even when elements of these categories have different surface distributions. For example, it has been proposed that there is a category of adjectives that refer to nationality, and a different category whose elements refer to

* Corresponding author.
E-mail address: stabler@ucla.edu (E.P. Stabler).

PII: S0304-3975(01)00351-6

qualities, and that we find these categories in the following phrases from English and French, respectively,

(a) [$_{dp}$ an [$_{qual}$ expensive] [$_{nat}$ English] [$_{n}$ fabric]],

(b) [$_{dp}$ un [$_{n}$ tissu] [$_{nat}$ anglais] [$_{qual}$ cher]].

(4) *Similar grammatical operations.* Linguists are particularly interested in characterizing universal properties of human languages. In the tradition of transformational syntax, much attention has been given to the restrictions on range of possible discontinuous dependencies which, in an algebraic approach, can be created by structure building operations. Ross, Chomsky and many others observed that some sorts of dependencies never occur; the domains of the available operations may be restricted so that they can never be created.

We will consider how each of these ideas might be captured in an algebraic presentation of a language.

2. A grammar formalism

To set the stage for considering various important similarity claims, we introduce a formalism that captures some aspects of current transformational analyses of language [26, 6, 14, 16]. Instead of generating categorized strings, the grammars formalized here generate tuples of categorized strings, where the categorial classification of each string is given by a "*type*" and a sequence of features [20, 23, 27, 1]. We call each categorized string a "*chain*", and each expression is then a (nonempty, finite) sequence of chains.

Definition 1. A minimalist grammar $G = (\Sigma, F, Types, Lex, \mathscr{F})$, where

Alphabet $\Sigma \neq \emptyset$

Features $F = base$		(basic features, $\neq \emptyset$)
	$\cup\{=f \mid f \in base\}$	(selection features)
	$\cup\{+f \mid f \in base\}$	(licensor features)
	$\cup\{-f \mid f \in base\}$	(licensee features)
$Types = \{::, :\}$		(lexical, derived)

For convenience: *Chains* $C = \Sigma^* \times Types \times F^*$

Expressions $E = C^+$

Lexicon $Lex \subseteq C^+$ is a finite subset of $\Sigma^* \times \{::\} \times F^*$.

Generating functions $\mathscr{F} = \{merge, move\}$, partial functions from E^* to E, defined below.

Language $L(G) = \text{closure}(Lex, \mathscr{F})$.

For any $\mathtt{f} \in F$, the strings of category $\mathtt{f}$,

$$S_{\mathtt{f}}(G) = \{s |\ s \cdot \mathtt{f} \in L(G) \text{ for some } \cdot \in Types\}.$$

The generating functions *merge* and *move* are partial functions from tuples of expressions to expressions. We present the generating functions in an inference-rule format for convenience, "deducing" the value from the arguments. We write *st* for the concatenation of s and t, for any strings s, t, and let ε be the empty string.

merge: $(E \times E) \to E$ is the union of the following three functions, for $s, t \in \Sigma^*, \cdot \in \{:, ::\}$, $f \in base$, $\gamma \in F^*$, $\delta \in F^+$, and chains $\alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l$ $(0 \leqslant k, l)$

$$\frac{s ::= f\gamma \qquad t \cdot f, \alpha_1, \ldots, \alpha_k}{st : \gamma, \alpha_1, \ldots, \alpha_k} merge1$$

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

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

Notice that the domains of *merge*1, *merge*2, and *merge*3 are disjoint, so their union is a function.

move: $E \to E$ is the union of the following two functions, for $s, t \in \Sigma^*$, $f \in base$, $\gamma \in F^*$, $\delta \in F^+$, and *chains* $\alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l$ $(0 \leqslant k, l)$ satisfying:

(SMC) none of $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k$ has $-f$ as its first feature.

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

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

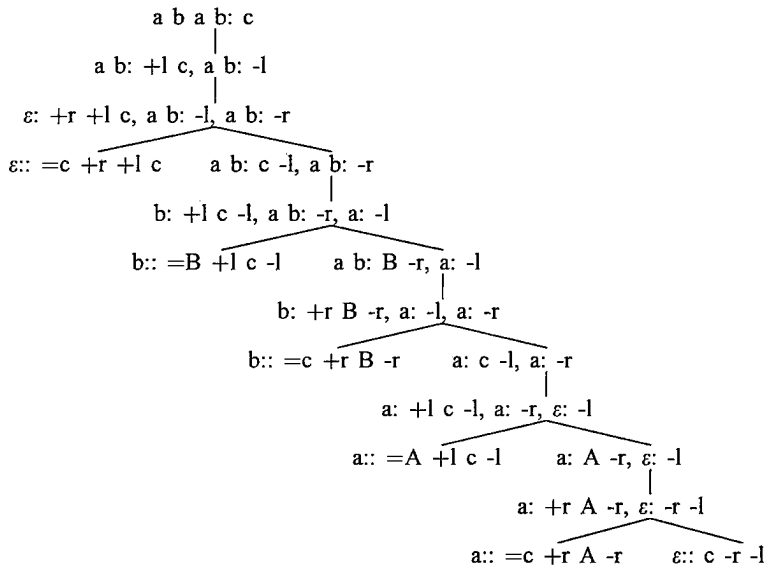
Notice that the domains of move1 and move2 are disjoint, so their union is a function. The (SMC) restriction on the domain of *move* is a simple version of the "shortest move condition" [2], briefly discussed in Section 6 below.

2.1. Grammar G1: $S_{\mathtt{c}}(G1) = \{xx |\ x \in \{a, b\}^*\}$

A simple grammar for the copy language is given by the following six lexical items:

ε:: c -r -l	ε:: =c +r +l c
a:: =c +r A -r	b:: =c +r B -r
a:: =A +l c -l	b:: =B +l c -l

With this grammar, we have derivations like the following:



S_c is not a context free language (CFL), but it is a tree adjoining language (TAL). Stabler [26] showed that there is a grammar in the formalism defined here which generates $a^n b^n c^n d^n e^n$. This latter language is neither a CFL nor a TAL, but is a multi-component tree adjoining language (MCTAL), as discussed in Section 3 below.

2.2. *Basic constituent orders: SOV, VSO, SVO*

Adapting a very simplified version of some suggestions from [13], it is easy to design naive grammars which derive the common orders of S(ubject), O(bject) and V(erb) in various languages.

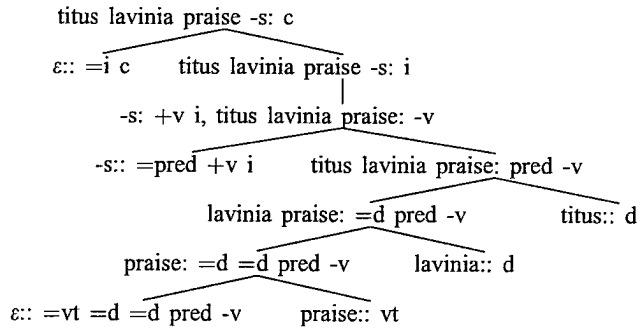
2.2.1. *Grammar NT: naive Tamil*

An SOVI language like Tamil can be obtained by letting the verb select its object and then its subject, and then moving the whole SOV complex to the "specifier" of i(nflection). The following 10 lexical items provide a naive grammar of this kind:

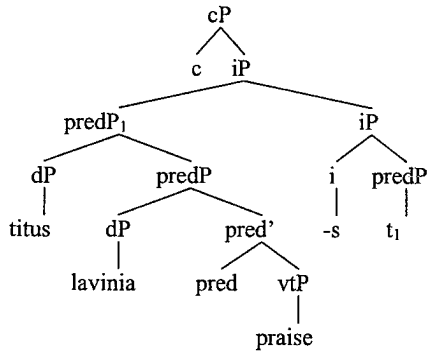
lavinia::d	titus::d
praise::vt	criticize::v
laugh::v	cry::v
ε::=i c	-s::=pred,+v,i
ε::=vt =d =d pred -v	ε::=v =d pred -v

Notice that the -s in the string component of an expression signals that this is an affix, while the -v in the feature sequence of an expression signals that this item must move to a +v licensing position.

With this lexicon, we have the following derivation of the string *titus lavinia praise -s* $\in S_c(NT)$:



The conventional depiction of the "derived structure" associated with this derivation would look something like this, with co-indexed "traces" to indicate movement relations:



These conventional structures show some aspects of the history of the derivations, something which can be useful for linguists even though it is not necessary for the calculation of derived expressions. A precise definition of the correspondence between our derivations and these structures is straightforward, but beyond the scope of this paper [26].

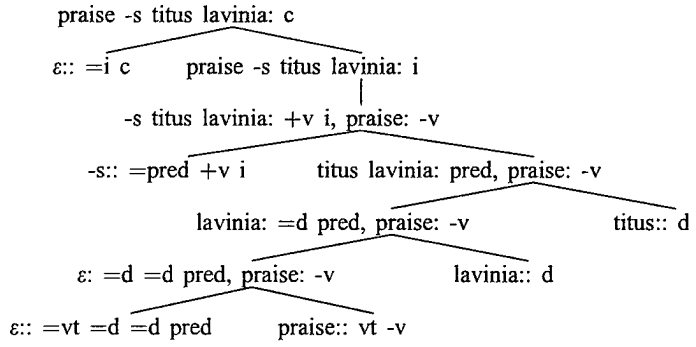
2.2.2. *Grammar NZ: naive Zapotec*

An VSO language like Zapotec can be obtained by letting the verb select its object and then its subject, and then moving just the lowest part of the SOV complex move to the "specifier" of i(nflection). The following 10 lexical items provide a naive grammar of this kind:

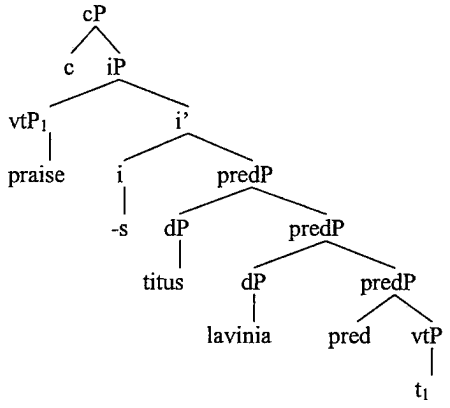
lavinia::d	titus::d
praise::vt -v	laugh::v -v

ε::=i c	-s::=pred,+v,i
ε::=vt =d =d pred	ε::=v =d pred

With this lexicon, the string *praise -s titus lavinia* $\in S_c(NT)$:



The conventional depiction of the "derived structure" associated with this derivation would look something like this:



2.2.3. Grammar NE: naive English

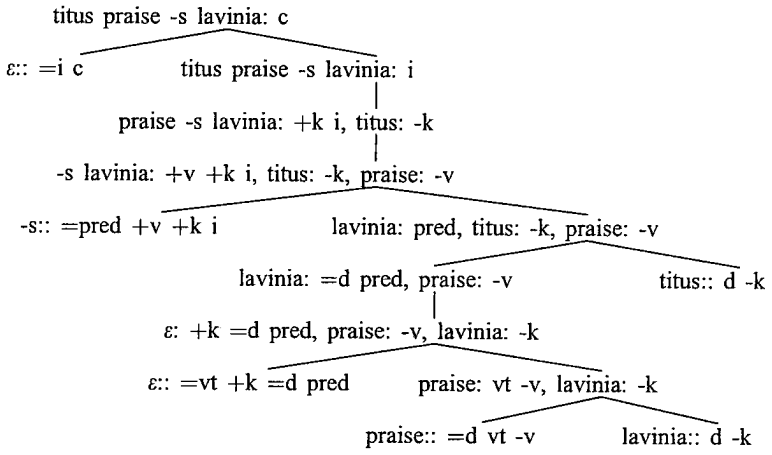
The following 16 lexical items provide a slightly more elaborate fragment of an English-like SVIO language:

lavinia:: d -k	titus:: d -k	who:: d -k -wh	
some:: =n d -k	every:: =n d -k	noble:: n	kinsman:: n
laugh:: =d v -v	cry:: =d v -v		

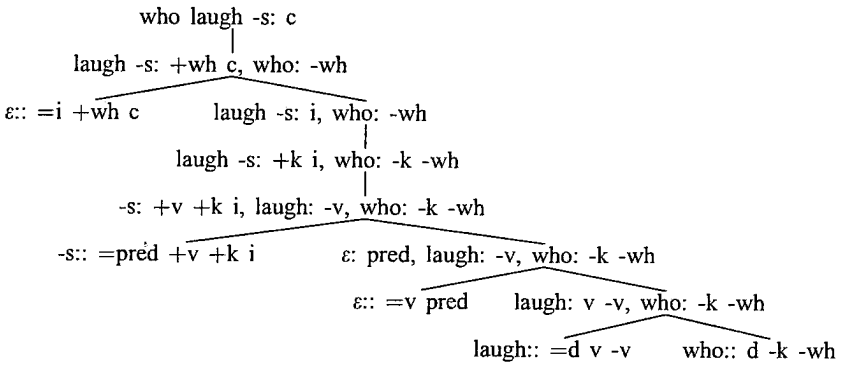
praise:: =d vt -v	criticize:: =d vt -v	
-s:: =pred +v +k i	ε:: =vt +k =d pred	ε:: =v pred
ε:: =i c	ε:: =i +wh c	

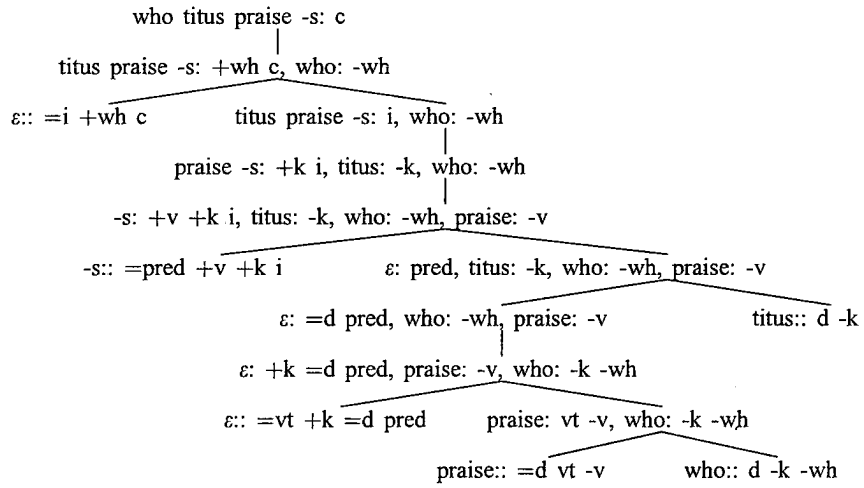
Notice that an SVIO language must break up the underlying SVO complex, so that the head of inflection can appear postverbally. This may make the SVIO order more complex to derive than the SOVI and VISO orders, as in our previous examples.

With this lexicon, we have the following derivation of the string *titus praise -s lavinia* $\in S_c(NE)$:

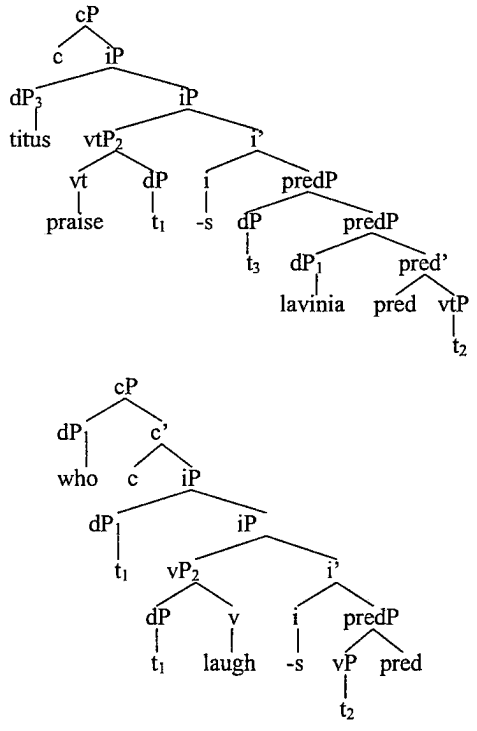


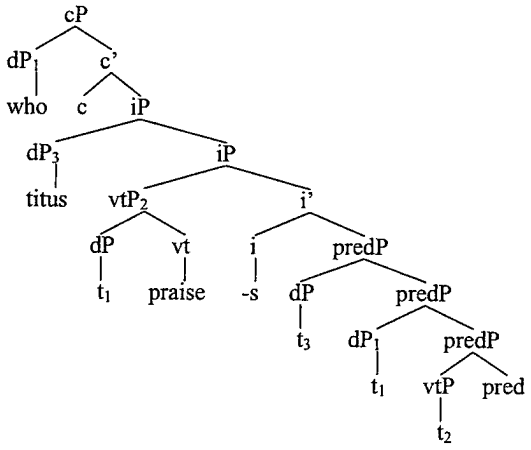
These lexical items allow wh-phrases to be fronted from their "underlying" positions, so we can derive *who laugh -s* and (since "do-support" is left out of the grammar for simplicity) *who titus praise -s*:





Derivations likes those shown above would traditionally be depicted with a final "derived structure" that would look roughly like the following, with co-indexed "traces" to indicate movement relations:





2.3. *Previous results*

Let the collection of "minimalist languages", MLs, be the collection of string languages S_c for some $G=(\Sigma,F,Types,Lex,\mathscr{F})$ and $c\in F$.

Expressive power: It is well known that CFLs $\subset$ MCTALs = MCFLs = LCFRLs [23, 27]. [14] showed that MLs $\subseteq$ MCFLs, and recently the converse is established in [15, 9], so we now have a rather remarkable convergence:

$$\text{MLs} = \text{MCTALs} = \text{MCFLs} = \text{LCFRLs}.$$

Recognition complexity: For any MG with category c, the language S_c can be recognized in $\mathcal{O}(n^{4m+4})$ where m is a constant depending on the lexicon [8].

Implementation: the inference schemes used to define the generating functions can be used with any implementation of chart-based deductive closure, such as the one in [24]. This kind of parsing strategy can also be implemented as a kind of constraint propagation [17].

3. Similar structures in a given language

As discussed in [11, 12] with respect to some simpler grammars, a natural notion of structure is induced by the definition of a language as a closure.

Notation 1. Given any grammar $G=(\Sigma,F,Types,Lex,\mathscr{F})$, it is convenient to restrict each generating function $F\in\mathscr{F}$ to $L(G)$.

For any $S\subseteq A$, and any total $h:A\to A$, let $h(S)=\{h(x)|\ x\in S\}$. For each $\langle a_1,\ldots,a_n\rangle \in A^n$, set $h(a_1,\ldots,a_n)=\langle h(a_1),\ldots,h(a_n)\rangle$.

So for any partial function $F: A \to A$, $h(F)$ is the function such that:

$$Dom(h(F)) = \{h(x) |\ x \in Dom(F)\}, \text{ and}$$
$$\forall x \in Dom(h(F)), \quad h(F)(x) = h(F(x)).$$

It immediately follows from this definition that for any total $h: A \to A$ and any partial $F: A \to A$, $h(F) = F$ iff $h(Dom(F)) = Dom(F)$ and for all $x \in Dom(F)$, $h(F(x)) = F(h(x))$:

$$\begin{array}{ccc} x & \xrightarrow{F} & F(x) \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle h} \\ h(x) & \xrightarrow[h(F)=F]{} & h(F(x)) = h(F)(h(x)) = F(h(x)) \end{array}$$

Definition 2. For any grammar G and any bijection $h: L(G) \to L(G)$, h is a syntactic automorphism (or symmetry) for $(L(G), \mathscr{F})$ iff for every $F \in \mathscr{F}$, $h(F) = F$.

The automorphisms $Sym(G) = \{h |\ h$ a syntactic automorphism for $(L(G), \mathscr{F})\}$.

Definition 3. Expressions $s, t \in L(G)$ have the same structure $s \simeq t$ iff for some $h \in Sym(G)$, $h(s) = t$. Clearly, $\simeq$ is an equivalence relation, so for any $s \in L(G)$ let $[s] = \{t |\ s \simeq t\}$.

The (structurally) invariant expressions $Inv(L(G)) = \{s \in L(G) |$ for all $h \in Sym(G)$, $h(s) = s\}$. So then, if $[s] = \{s\}$, $s \in Inv(L(G))$.

This same notion of invariant, that of being fixed by the syntactic automorphisms, applies to sets, functions and relations.

The (structurally) invariant sets $Inv(\mathscr{P}(L(G))) = \{S \subseteq L(G) |$ for all $h \in Sym(G)$, $h(S) = S\}$, where $\mathscr{P}(L(G)$ is the powerset of $L(G)$.

For each n, the (structurally) invariant n-ary relations,

$$Inv(\mathscr{P}(L(G)^n)) = \{R \subseteq L(G)^n |$ for all $h \in Sym(G)$, $h(R) = R\}.$$

We will call all these structural invariants grammatical constants.

Theorem 1. *In any grammar G, the generating functions, their domains, the sets of ith coordinates of their domains, and their ranges, are all grammatical constants.*

Notation 2. For any $i, j > 0$, and any $k \leqslant i$, consider any (partial) functions $f^i: L(G)^i \to L(G)$ and any $g^j: L(G)^j \to L(G)$, define the (partial) function $f^i \circ_k g^j: L(G)^{i+j-1} \to L(G)$ as follows:

$$\begin{aligned} Dom(f^i \circ_k g^j) = \{\langle a_1, \ldots, a_{k-1}, b_1, \ldots, b_j, a_{k+1}, \ldots, a_i\rangle | \\ \langle b_1, \ldots, b_j\rangle \in Dom(g^j), \\ \langle a_1, \ldots, a_{k-1}, g^j(b_1, \ldots, b_j), a_{k+1}, \ldots, a_i\rangle \in Dom(f^i)\}, \end{aligned}$$

$$f^i \circ_k g^j(a_1,\ldots,a_{k-1},b_1,\ldots,b_j,a_{k+1},\ldots,a_i))$$
$$= f^i(a_1,\ldots,a_{k-1},g^j(b_1,\ldots,b_j),a_{k+1},\ldots,a_i).$$

Theorem 2. *In any grammar G with generating functions $f^i, g^j \in \mathscr{F}$, if grammar G' is the result of adding $f^i \circ_k g^j$ to $\mathscr{F}$, these two grammars define exactly the same structure, in the sense that they define the same language, have the same symmetries, and have the same invariants.*

It follows that for any grammar that uses a finite lexicon and finite set of structure building functions to define an infinite language, there are infinitely many other grammars with exactly the same structure, since we can introduce arbitrarily many new, composed functions with no structural change. For any derived expression in any grammar, we can compose all the functions used in the derivation so that the expression is derived in one step from a tuple of lexical items.

The collection of grammatical subsets of the language has a structure:

Theorem 3. *For any grammar G, $Inv(\mathscr{P}(L(G)))$ contains $\emptyset$ and $L(G)$, which just says that the property of being in the language or not is a structural invariant.*

In fact, for any n, $(Inv(\mathscr{P}(L(G)^n)), \subseteq)$ is a complete atomic Boolean (i.e. complemented, distributive) lattice whose atoms are the $[s]$ for each $s \in L(G)^n$.

Thus the collections of invariants of any arity are closed under arbitrary intersections, unions, and complements. The collection $Sym(G)$ of symmetries also has structure:

Theorem 4. *For any grammar G,*
(a) *for any $f, g \in Sym(G)$, $f \circ g \in Sym(G)$;*
(b) *$id \in Sym(G)$, where id is the identity on $L(G)$, and $id \circ f = f$ for all $f \in Sym(G)$;*
(c) *for any $f \in Sym(G)$, $f^{-1} \in Sym(G)$;*
(d) *for any $f, g, h \in Sym(G)$, $(f \circ g) \circ h = f \circ (g \circ h)$;*
That is, $(Sym(G), \circ)$ is a group, the symmetry group of G.

The previous basic results, established in [12], do not depend on any specific properties of the grammar formalism, beyond the fact that the language is the closure of a set *Lex* under the (partial) functions in $\mathscr{F}$. The present paper is concerned with the possibility of comparing particular grammars and particular structures, so we present some first simple results to illustrate how the notions defined above do this.

Theorem 5. *In NE,* titus::d -k $\not\simeq$ praise::=d vt -v.

Proof. Since $\langle$praise::=d vt -v, titus::d -k$\rangle \in Dom(merge)$ *and there is no* $\sigma \in L(G)$ *such that* $\langle$titus::d -k, $\sigma\rangle \in Dom(merge)$, *it follows that no h that maps* titus::d -k *to* praise::=d vt -v *can be such that* $h^{-1}(merge) = merge$. □

Theorem 6. *In NE*, titus::d -k $\simeq$ lavinia::d -k.

Proof. Let $h_o: \Sigma^* \to \Sigma^*$ be the (total) function which simultaneously replaces all occurrences of *titus* with *lavinia* and all occurrences of *lavinia* with *titus*, leaving all other elements of the string unchanged. Now define (total) $h: L(G) \to L(G)$ which applies to any expression to yield the result of applying h_o to every string component. So, for example, when the expression is a single chain, $s \cdot \sigma$ (for any $s \in \Sigma^*, \cdot \in Types, \sigma \in$ Features*), then $h(s \cdot \sigma) = h_o(s) \cdot \sigma$.

Clearly, h is a bijection. To complete the proof, we show that $h \in Sym(NE)$ by showing that it fixes *merge* and *move*.

Consider *merge* first. It is clear that $h(Dom(merge)) = Dom(merge)$. (as can be shown by a simple induction on the lengths of derivations), so we just need to see that h commutes with *merge*. So consider arbitrary $\langle a, b\rangle \in Dom(merge)$. There are three cases:

(1) $\langle a, b\rangle \in Dom(merge1)$. Then a is $s ::= f\gamma$ and b is $t \cdot f, \alpha_1, \ldots, \alpha_k$ for some $s, t \in \Sigma^*$, $\cdot \in \{:, ::\}$, for $f \in base$, $\gamma \in F^*$, and for chains $\alpha_1, \ldots, \alpha_k$.
So we have, immediately from our definitions:

$$\begin{aligned}
h(merge(a,b)) &= h(st: \gamma, \alpha_1, \ldots, \alpha_k) && def.\ merge\\
&= h_o(st): \gamma, \alpha_1, \ldots, \alpha_k && def.\ h\\
&= merge(h_o(s) ::= f\gamma, h_o(t) \cdot f, \alpha_1, \ldots, \alpha_k)) && def.\ merge\\
&= merge(h(s ::= f\gamma, t \cdot f, \alpha_1, \ldots, \alpha_k)) && def.\ h\\
&= merge(h(a,b))
\end{aligned}$$

(2) $\langle a, b\rangle \in Dom(merge2)$. The argument is exactly analogous.

(3) $\langle a, b\rangle \in Dom(merge3)$. Again, the argument is exactly analogous.

For *move*, it is again clear that $h(Dom(move)) = Dom(move)$, and for arbitrary $a \in Dom(move)$, there are the two cases to consider:

(1) $a \in Dom(move1)$. Then by definition $a = s: +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t: -f, \alpha_{i+1}, \ldots, \alpha_k$ for some $s, t \in \Sigma^*$, $\cdot \in \{:, ::\}$, $f \in base$, $\gamma \in F^*$, and for chains $\alpha_1, \ldots, \alpha_k$. So we have, as in the previous cases:

$$\begin{aligned}
h(move(a)) &= h(ts: \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k) && def.\ move\\
&= h_o(ts): \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k && def.\ h\\
&= move(h_o(s): +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, h_o(t): -f, \alpha_{i+1}, \ldots, \alpha_k) && def.\ move\\
&= move(h(s: +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t: -f, \alpha_{i+1}, \ldots, \alpha_k)) && def.\ h\\
&= move(h(a))
\end{aligned}$$

(2) For $a \in Dom(move2)$, the argument is exactly analogous.

That completes the proof. □

So we see that the straightforward notion of "same structure" that is captured by the existence of an automorphism lets us capture the first kind of comparison mentioned in Section 1:

Theorem 7. *In NE,* titus laugh -s::$c \simeq$ lavinia cry -s::c.

Furthermore, our formal notion of grammatical constant corresponds quite closely to the linguists':

Theorem 8. *In NE, the following expressions are grammatical constants:*

who :: d -k -wh
the lexical affix-s
every empty expression

Proof. Consider the composition of functions that generates *who laugh -s* in one step, from the 5 leaves shown in the tree displayed in Section 2.2 above. Then we can proceed much as in 5. In the domain of this function, the arguments must be a sequence of expressions in which a category with feature $=$f is immediately followed by an element of category f, and the first can only be who:: d -k -wh, the only lexical item which could feed the movement. Looking at the sequences of this kind that the grammar can provide, the only possible variation is in the determiners (the names *titus*, *lavinia*) and in the verbs (*laugh*, *cry*). Using Theorems 1 and 2, the claims of this theorem follow. □

4. Similar structures in different languages

While the results of the previous section show that the notion of "same structure" captured in Definition 2 fit the linguists' fairly well for comparisons among structures in a given language, it is easy to see that those notions are too tight to allow useful comparisons across languages. If we add even just a single name to a language, the result is a different language that has no symmetries in common with the first. In fact, it is easy to see that

noble and kinsman::n $\not\simeq$ kinsman and kinsman::n.

So if we add any new noun to the language, there will be new equivalence classes of expressions, and the symmetry group of the extended language will not be isomorphic to the symmetry group of the original language because the number of automorphisms will change.

Returning to the linguist's perspective, we would like to say that adding a single name to a language does not (significantly) change the structure of the language. Other single lexical additions would change the language though. For example, adding a new

auxiliary verb like BE to the language would be significant, generating sequences like this one,

Subject BE -s V -ing Object.

How could we capture this kind of distinction between different kinds of extensions? A simple first idea is this.

Definition 4. When grammars G, G' are identical except that $Lex_G \subseteq Lex'_G$, we say that G' is a lexical extension of G. A lexical extension G' of G is free iff for every $s \in Lex_{G'} - Lex_G$, there is a $t \in Lex_G$ such that $s \simeq_{G'} t$.

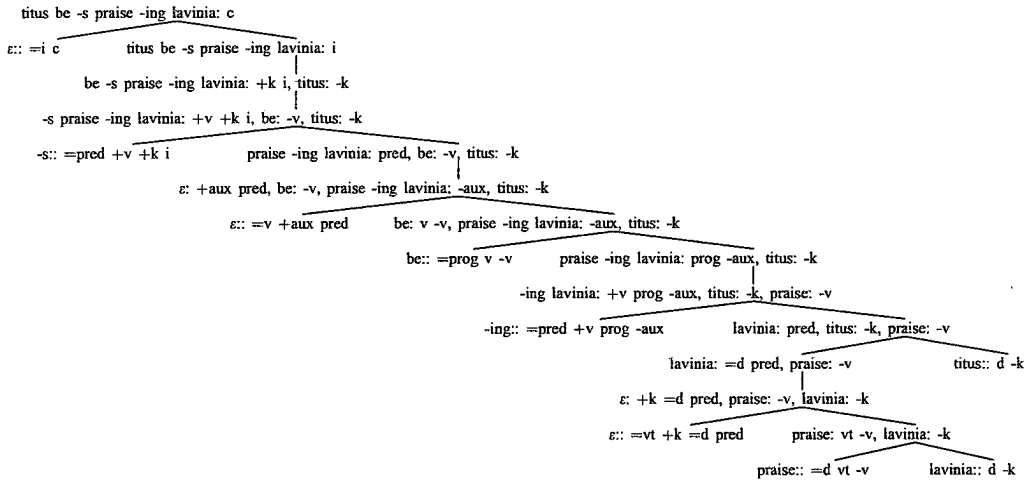
Theorem 9. *Let NE′ be the result of adding* sam::d -k *to NE. Then NE′ is a free lexical extension of NE.*

Theorem 10. *Let NE′ be the result of adding the following lexical items to NE.*

ε ::= v +aux pred be ::= prog v -v -ing ::= pred +v prog -aux

Then NE′ is a free lexical extension of NE. In fact, each of these new lexical items is a grammatical constant of NE′.

Notice first that, in *NE′*, we derive an English-like ordering of the auxiliary verb and inflectional affixes; so for example, *titus be -s praise -ing lavinia* $\in S_c(G)$:



This tree has 9 leaves, 9 lexical items. Notice that, reading the first 8 leaves of the tree from left to right, we have a category with feature =f followed by an element of category f.

Proof. Again, we procede as in the proof of Theorem 5. Consider the composition of functions that generates titus *be -s praise -ing lavinia* in one step, from the 9 leaves

shown in the tree displayed just above. In the domain of this function, the first 8 arguments must be a sequence of expressions in which a category with feature =f is immediately followed by an element of category f. Looking at the sequences of this kind that the grammar can provide, the only possible lexical items that can be mapped to something other than themselves by any automorphism are the determiners (the names *titus*, *lavinia*) and the verbs (*praise*, *criticize*). Using Theorems 1 and 2, the claims of this theorem follow. □

5. Similar categories

The previous section provides a way to characterize changes to a language that do not affect structure in a significant way, but linguists often want to compare languages that have no substantial overlap in the lexicon. We have no satisfactory way to make sense of these claims yet.

For example, in recent work, Cinque [3] has observed that in a striking range of languages, there seems to be a canonical preferred order for adverbs of various kinds: frankly<fortunately<probably<always<completely.

(Norwegian)
De forstå r enda ikke alltid helt hva jeg snakker om
They understand still not always completely what I talk about

(Serbo-Croatian)
Jam sam ga gotovo potpuno zaboravio
I have him almost completely forgot

(Albanian)
Ai nuk i kupton gjithnjë tërësisht vërejtjet
he not them understands always completely remarks

(Chinese)
wo ganggang wanquan wang-le ta-de dizhi
I just completely forgot his address

Some languages deviate from this order, but the deviations seem not to be haphazard. Sometimes the hierarchy flips into a near-mirror image [18, 21]. [1]

(Malagasy)
Manasa lamba tanteraka foana Rakoto
wash clothes completely always Rakoto

[1] Importantly, the mirroring we find in these sorts of cases is *never* perfect. If the mirroring were perfect, simpler treatments would be possible [10, 3, 18, 21, 25, 4].

A similar hierarchy of preferences seems to obtain among the adjectives in a wide range of languages [7, 25, 4, 5, 3] where we find the French neutral order mirrors, in part, the English order:

an expensive English fabric
un tissu anglais cher

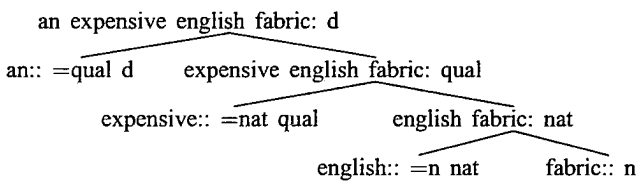
Dimitrova–Vulchanova proposes that the canonical hierarchy among adjectives is (in part) something like this: quality<size<shape<color<nationality<noun, but she points out deviations from the canonical order that are more complex than the mirror image:

(Albanian)
një fustan fantastik blu
a dress fantastic blue
fustan-i fantastik blu
dress-the fantastic blue

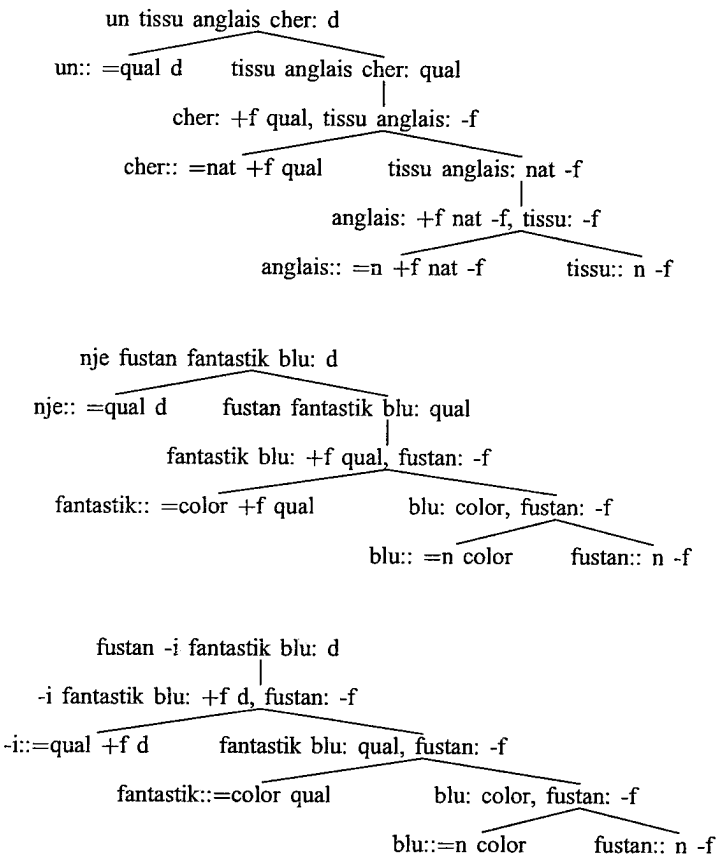
It is notable that the characterizations of the elements of the adjective and adverb hierarchies is semantic, while the "hierarchy" itself is syntactic. It is not obvious that there is a way to characterize the elements of the hierarchy in any other terms than these semantic ones, but to get the ordering facts, we could assign distinctive syntactic categories to these elements, and then design a syntax in which the deviations from the preferred order can be seen as exceptional.

We can easily design grammar fragments to illustrate this strategy. To get the canonical order, we can have categories qual, size, shape, color, nat, noun that select each other appropriately. Deviations from the canonical order can be obtained by moving sequences of selected elements.[2] To represent these possible selection possibilities elegantly, we use the notation ⩾qual to indicate a feature that is either qual or a feature that follows qual in this hierarchy. And similarly for the other features, so ⩾color indicates a feature in the set ⩾color={=color, =nat, =n}. Parentheses indicate optionality. Then the following very similar grammars for English, French and Albanian derive the rather different adjective sequences mentioned above:

English	French	Albanian
a(n)::⩾poss d -k	un::⩾poss d -k	-i::⩾poss +f d -k
expensive::⩾qual qual	cher::⩾qual +f qual (-f)	nje::⩾poss d -k
English::⩾nat nat	anglais::⩾nat +f nat (-f)	fantastik::⩾qual (+f) qual
fabric::n	tissu::n (-f)	blu::⩾color color
		fustan::n -f



[2] To obtain the desired orders by successive "head movement", the modifiers could be specifiers of corresponding "functional categories", but the present account which excludes head movement is simpler.



The fact that we have used the same features to state the grammars for the respective languages is clearly irrelevant when we come to the question of what syntactic sense there is, if any, to the claim that the English categories qual, size, shape, color, nat, noun are the same as the corresponding categories in French, Albanian, and other languages. Clearly, although these categories may be invariants in each language, we have no symmetry across languages that would justify calling them invariants across languages. Expressions using these categories in the different languages differ in their phonological forms, their distributions and in their complexities (the number of function applications required to construct them).

If each of these categories is an invariant in each human language though, and if the indicated hierarchy is realized by the "selects" relation, what we have is not a universal invariant, but a universal relation among language-specific invariants.[3] It is even possible that there are universal properties of these categories that uniquely identify them, providing an alternative to the semantic criteria.

[3] Other relations among invariants are pointed out in [11, 12].

6. Similar operations: constraints on movement

In transformational grammar, as also in categorial grammar and other traditions, an attempt has been made to identify the basic structure building operations in a language-universal way, though of course the operations of each grammar are distinct when taken extensionally to be a set of pairs of expressions. In transformational grammar in particular, considerable effort has been made to identify constraints on movement, which under the present construal turn out to be restrictions on the domain of the generating function *move*. In our definition of the structure building rules in Section 2, a condition of this kind is imposed on the domain of *move*, namely the "shortest move condition" (SMC). This condition does not allow movement to apply to any +f constituent in which there is more than one -f subconstituent available to move. Intuitively, the two -f subconstituents in such a construction are "competing" for the nearest available +f licensor, and since it is not possible for both to win, no movement is allowed.

This constraint has the important formal consequence that the number of extractions from each constituent can not exceed a constant bound k, which is the number of different licensing requirements that occur in the lexicon. However, the condition is perhaps too strong, as evidenced by multiple wh-movements, for example. It is interesting to consider how the SMC could be relaxed or modified to allow these.[4] For the moment, we simply observe that at least some of the proposals about how to exclude unwanted movements can be regarded as restricting the domains of the generating functions. Much work remains to be done in characterizing the constraints of movement, and more generally, relevant notions of similarity among operations.

7. Conclusions

The algebraic approach to grammar allows a rigorous exploration of claims about structural similarity. Claims to the effect that two expressions "have the same structure" can often be regarded as claims about the existence of a syntactic automorphism, a symmetry, that maps one to the other. This notion does not apply across different languages, but for closely related languages, we propose the notion of "free" lexical extensions. Claims about universal properties of human languages, though, typically involve languages that are very different from one another. It appears that claims about categorial identity across languages can only be regarded as claims about common properties of the categories of the respective languages. And at least some claims about "constraints on movement" can be regarded as universal properties of the generating functions.

[4] See for example the proposals of Richards [22] and Pesetsky [19].

References

[1] P. Boullier, Proposal for a natural language processing syntactic backbone, Tech. Rep. 3242, Projet Atoll, INRIA, Rocquencourt, 1998.

[2] N. Chomsky, The Minimalist Program, MIT Press, Cambridge, MA, 1995.

[3] G. Cinque, Adverbs and Functional Heads: a Cross-Linguistic Perspective, Oxford University Press, Oxford, 1999.

[4] M. Dimitrova-Vulchanova, G. Giusti, Fragments of Balkan nominal structure, in: A. Alexiadou, C. Wilder (Eds.), Possessors, Predicates and Movement in the Determiner Phrase, Amsterdam, Philadelphia, 1998.

[5] M. Dimitrova-Vulchanova, Modification in the nominal expression, in: Syntax and Semantics, Fall School, Norwegian University of Science and Technology, Trondheim, 1999.

[6] T.L. Cornell, A type logical perspective on minimalist derivations, in: Proc. Formal Grammar'97, Aix-en-Provence, 1997.

[7] J. Greenberg, Some universals of grammar with particular reference to the order of meaningful elements, in: J. Greenberg (Ed.), Universals of Human Language, Stanford University Press, Stanford, 1978.

[8] H. Harkema, A recognizer for minimalist grammars, in: Proc. Sixth Internat. Workshop on Parsing Technologies, IWPT'2000, 2000.

[9] H. Harkema, A characterization of minimalist languages, in: Proc. Logical Aspects of Computational Linguistics, LACL'01, Port-aux-Rocs, Le Croisic, France, 2001.

[10] R. Kayne, The Antisymmetry of Syntax, MIT Press, Cambridge, MA, 1994.

[11] E.L. Keenan, E.P. Stabler, Abstract syntax, in: A.-M. DiSciullo (Ed.), Configurations: Essays on Structure and Interpretation, Cascadilla Press, Somerville, MA, 1996, pp. 329–344.

[12] E.L. Keenan, E.P. Stabler, Syntactic invariants, in: Sixth Ann. Conf. on Language, Logic and Computation, Stanford, 1997.

[13] A. Mahajan, Eliminating head movement, in: The 23rd Generative Linguistics in the Old World Colloq., GLOW '2000, 2000.

[14] J. Michaelis, Derivational minimalism is mildly context-sensitive, in: Proc. Logical Aspects of Computational Linguistics, LACL'98, Grenoble, 1998.

[15] J. Michaelis, Transforming linear context free rewriting systems into minimalist grammars, in: Proc. Logical Aspects of Computational Linguistics, LACL'01, Port-aux-Rocs, Le Croisic, France, 2001.

[16] J. Michaelis, U. Mönnich, F. Morawietz, Algebraic description of derivational minimalism, in: Internat. Conf. on Algebraic Methods in Language Proceesing, AMiLP'2000/TWLT16, University of Iowa, 2000.

[17] F. Morawietz, Chart parsing and constraint programming, in: Proc. COLING-2000, 2000.

[18] M. Pearson, X(p)-movement and word order typology: 'direct' versus 'inverse' languages, in: Abstracts of GLOW '99, 1999.

[19] D. Pesetsky, Phrasal movement and its Kin, MIT Press, Cambridge, MA, 2000.

[20] C. Pollard, Generalized phrase structure grammars, head grammars and natural language, Ph.D. Thesis, Standford University, 1984.

[21] A. Rackowski, Malagasy adverbs, in: Ileana Paul (Ed.), The Structure of Malagasy, Vol. II. UCLA Occasional Papers in Linguistics 20, UCLA, 1998.

[22] N. Richards, The principle of minimal compliance, Linguist. Inquiry 29 (1998) 599–629.

[23] H. Seki, T. Matsumura, M. Fujii, T. Kasami, On multiple context-free grammars, Theoret. Comput. Sci. 88 (1991) 191–229.

[24] S.M. Shieber, Y. Schabes, F.C.N. Pereira, Principles and implementation of deductive parsing, Tech. Rep. CRCT TR-11-94, Computer Science Department, Harvard University, Cambridge, MA, 1993, available at `http://arXiv.org/`.

[25] D. Sportiche, Adjuncts and adjunctions, presentation at 24th LSRL, UCLA, 1994.

[26] E.P. Stabler, Remnant movement and complexity, in: G. Bouma, E. Hinrichs, G.-J. Kruijff, D. Oehrle (Eds.), Constraints and Resources in Natural Language Syntax and Semantics, CSLI, Stanford, CA, 1999, pp. 299–326.

[27] K. Vijay-Shanker, D. Weir, The equivalence of four extensions of context free grammar formalisms, Math. Systems Theory 27 (1994) 511–545.