Abstract Syntax

Edward L. Keenan & Edward P. Stabler
UCLA 1994

A naive view of language as a set of conventions or "games" might lead one to think that the semantic values of expressions are entirely arbitrary, the result of historical accident, unboundedly adjustable by innovation and fashion. A similar conclusion is sometimes reached on less naive views, ones which emphasize *l'arbitraire du signe* or *the autonomy of syntax.* Troubling for such views is the lack of semantic variability of grammatical constants (function words, closed class items) such as infinitival to and gerundive *-ing* in English, or boolean operators such as *and, or*, and *not*; or case markers such as nominative *-i/-ka* and accusative *-ul/-lul* in Korean.

This lack of semantic variability could be a biological accident, an arbitrary, universal requirement that certain categories in human languages contain grammatical constants with specially restricted syntactic and semantic properties.

Here we present a different view, taking some first steps toward a demonstration that these facts follow from a certain natural fit between syntactic and semantic properties of language. We present a theory independent characterization of the notion *grammatical structure* which covers that of *grammatical constant* as a special case. Then we show how the syntactic and semantic fixity of grammatical constants follow from some theory general axioms constraining the relation between form and meaning in natural language.

1 Defining the relation **has the same structure as**

We think of a *generalized grammar* G as consisting basically of a Lexicon (Lex), whose elements are categorized strings, and a set F of *structure building (generating) functions* which derive complex expressions from simpler ones beginning with Lex. L(G), the *language generated by* G, is the set of expressions (categorized strings) that can be derived from Lex by finitely many applications of the structure building functions.[1] It is easy to show that any set of expressions defined in any format whatever is generated by some generalized grammar. In this sense, the account of structure offered here is theory independent; it does not depend on any particular assumptions about what the generating functions are, how they are specified, or what sort of language is defined.

Given G and expressions σ, τ in L(G), we say that σ has the same structure as τ if and only if (iff) each can be derived from the other by an "appropriate" substitution of lexical items. For example given an English expression σ with derivation D(σ) we should be able to everywhere replace the NP (or DP) John in D(σ) with Bill and Bill with John yielding an expression D(τ) with the same structure as σ. By contrast we could not everywhere intersubstitute the adjectives eager and easy since To please John is easy is grammatical English and To please John is eager is not. So a rough guide as to whether a substitution is appropriate is whether the substitution everywhere preserves grammaticality.

In more detail, we represent the appropriate substitutions as functions h from L(G) to L(G) which satisfy the two conditions given below. We shall call such h's **structure maps** for L(G). The idea is that when σ is an expression, h(σ) is the expression which h substitutes for σ. And we choose the conditions h must satisfy so as to guarantee that for any σ, σ and h(σ) have the same structure.

A crucial intuition, well supported from the earliest work in generative grammar, is that whether two expressions have the same structure cannot be decided just by checking their internal structure (as represented by, say, their syntactic derivations). E.g. expressions like *John wanted to leave* and *John promised to leave* are structurally different though internally they appear similar. Comparable claims hold for *John is easy to please* and *John is eager to please*, and *John praised Bill* and *John praised himself.* Whether two expressions have the same structure depends in part on how they and their component expressions are structurally related to others.

[1] To be precise, we define a grammar as a 4-tuple, G=$\langle V_G, Cat_G, Lex_G, F_G\rangle$, where V_G is the vocabulary, Cat_G is the set of categories, Lex_G is a set of $\langle$string,category$\rangle$ pairs, and $F_G\rangle$is a set of partial functions that map n-tuples of $\langle$string,category$\rangle$ pairs to $\langle$string,category$\rangle$ pairs. The language L(G) is the closure of Lex_G under the functions in F_G. We leave off the subscripts when the context makes clear which grammar we are talking about.

Part of this intuition is captured by requiring that the collection of substitution values under h must be the entire language, not just a proper subset of it. That is, we want the set of h(σ)'s to have exactly the structure of L(G), and this might fail to happen if certain expressions, say reflexive pronouns or verbs like *promise*, were not present among the h(σ)'s. In such a case the set of h(σ)'s would lack certain structurally significant elements of L(G) as pretheoretically judged. Formally then we require that a structure map h for L(G) be onto (surjective). We shall also require that h map distinct expressions to distinct expressions, since identity of expressions is typically an important part of the structure of an expression. E.g. we do not want a structure map h to be able to map a natural sentence like *John neither laughed nor cried* to *John neither laughed nor laughed* since the latter is probably not even grammatical and in any event does not clearly, as pretheoretically judged, have the same structure as the former. So our first condition on structure maps is:

(1) Structure maps are bijective (one to one and onto).

Our second condition is the most fundamental one. It will guarantee for example that whenever a structure building function F applies to some sequence δ of expressions to yield some σ then it will also apply to h(δ) to yield h(σ). (If δ is a sequence, say $\langle \alpha, \beta\, gamma \rangle$, then by h($\delta$) is meant the sequence $\langle h(\alpha), h(\beta), h(\gamma) \rangle$.) The idea is that a generating function F treats any δ and h(δ) the same, the only differences in what F derives from δ and from h(δ) are due to h. Thus we want to say that for all δ and σ, F(δ) = σ iff F(h(δ)) = h(σ). A succinct way to say this is just to say that structure maps h fix the generating functions. That is,

(2) For all structure maps h and all generating functions f, h(f) = f.

We may now define an expression σ to *have the same structure as* an expression τ iff there is a structure map h such that h(σ) = τ. In such a case we write $\sigma \simeq \tau$ and say that σ is *(grammatically) isomorphic* to τ. One sees that $\simeq$ is an equivalence relation: each σ is isomorphic to itself; if $\sigma \simeq \tau$ then $\tau \simeq \sigma$, and if $\sigma \simeq \tau$ and $\tau \simeq \eta$ then $\sigma \simeq \eta$.

A linguistic property P is *structural* iff whenever P holds of an expression σ then P holds of all expressions isomorphic to σ. This just says that a structural property P is one that is fixed by all structure maps, that is, h(P) = {h(σ)|σ has P} = P, for all structure maps h. It follows that if a structural property fails of some σ then it fails of everything isomorphic to σ. Thus a structural property is one that can't tell the difference between isomorphic expressions σ,τ: either both have P or neither do, but it could not happen that one has P and the other doesn't. And more generally a linguistic relation R is structural iff whenever some σ stands in the relation to some τ then h(σ) stands in the relation to h(τ), all structure maps h. (The generalization to n-ary relations here is obvious).

A set of expressions (or sequences of expressions) is said to be *structural* (or *structurally definable*) iff the property of being in that set is structural. For example, given a grammar G, the set L(G) is structurally definable. This just says that whether a possible expression is grammatical is a structural property. Similarly standard structural relations like *is a constituent of, is a sister of, c- commands* are provably structural (Keenan & Stabler 1995). Finally, an expression σ is *structural* (a grammatical constant) iff the property of being σ is structural, that is, h(σ) = σ, all structure maps h. Provably,

(3) An expression σ is a grammatical constant iff σ is isomorphic only to itself.

Thus grammatical constants are those expressions which cannot be changed without changing structure. In derivational terms, they cannot be replaced by any other in all derivations, preserving grammaticality. Note that the criterion for an expression to be a structural one, a grammatical constant, is the same as for a property or relation to be structural. Namely it is mapped to itself by all structure maps.

Note that while relations like *is a constituent of* are structural in any grammar, a property like *being an expression of category X* may or may not be structural. For example, in the next section we define a simple language in which the property of being an NP is structural. But in the spirit of X-bar theory we leave open the possibility that, for example, NPs could be isomorphic to Ss, so we could find a structure map that mapped NPs to Ss and Ss to NPs. In this case, being an NP would not be a structural property. It does not seem then that we want to guarantee that the property of being an expression of a given category, even a "major" category, is always a structural property in human languages. But the following weaker condition is a plausible constraint on human languages:

(4) **A Language Universal ?**
$\forall$G, $\forall \sigma, \tau \in$ L(G), if Cat(σ) = Cat(τ) then if h is any structure map, Cat(h(σ)) = Cat(h(τ)).

That is, any given structure map will map expressions of the same category to expressions of the same category. The requirement here is that the category system reflect the real structure of the language, and so the claim may appear to be methodological. But it is also an empirical claim: the best theory of language will be one in which the category system reflects structure at least to this extent.

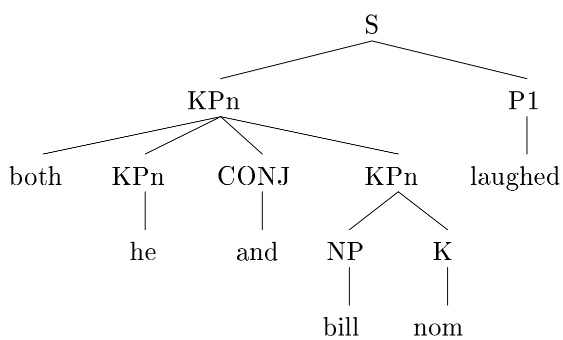
In a similar way the property of being a particular morpheme will typically not be a structural property, although it can happen. In Case Marked English below the property of being the expression ⟨*John*,NP⟩ is not structural while the properly of being the case marker ⟨*nom*,K⟩ is structural. This depends entirely on the particular grammar. It is possible to design artificial grammars with no constants, or grammars according to which every expression is a constant.

Before turning to axiomatic constraints between form and meaning we illustrate the ideas presented above with a simple little grammar of "Case Marked English" (CME).

2 Case Marked English: An Illustrative Example

We present a grammar of a familiar sort to illustrate the notions presented more abstractly above. Our purpose is to show that expressions in such familiar "languages" can be represented naturally in our function-argument formalism and "read" in a natural way. Before giving the grammar we illustrate one "typical" (short) expression and say informally how it is built. We design CME so that the NPs carry overt case markers, the category of the resulting expressions being "Kase Phrases" of two sorts: KPn's or "nominative Kase Phrases" and KPa's or "accusative Kase Phrases." The pronoun he is a lexical KPn, him and himself are lexical KPa's. We allow boolean combinations of expressions with *and, or,* and *nor* in most categories (but not K "Kase Marker" or NP).

(5) ⟨both he and bill-nom laughed, S⟩



We may read this tree as follows. Except for the particle *both* which gets inserted by the coordination rule, the leaves of the tree are lexical items: *he* is a lexical item of category KPn, that is, ⟨*he*,KPn⟩ is in the lexicon of CME. *and* is a lexical item of category CONJ, and so on. The second KPn in this sentence is formed by a structure building function (called Case Mark) which combines *bill* of category NP with *nom* of category K to form *bill-nom* of category KPn. The Coordination rule (called BOOL) combines the three expressions ⟨*he*,KPn⟩, ⟨*and*,CONJ⟩ and ⟨*bill-nom*,KPn⟩ to yield *both he and bill-nom* of category KPn. Note that *both* is introduced as part of the value of the function and does not itself have a category. Then the Predicate-Argument rule PA combines that KPn with the P1 *laughed* to form the sentence *both he and bill-nom laughed.*

More formally now, the **vocabulary V** for CME is:

john, bill, he, himself, him, laughed, cried, praised, criticized, both, and, either, or, neither, nor

The **categories Cat** for CME are: S, KPa, KPn, NP, K, P1, P2, CONJ.

The expressions in the **lexicon Lex** can be listed by category:

K:	nom,acc	P1:	cried, laughed
NP:	john, bill	P2:	praised, criticized
KPn:	he	CONJ:	and, or, nor
KPa:	himself, him		

The entries for K abbreviate ⟨*nom*,K⟩∈Lex and ⟨*acc*,K⟩∈Lex. The other lines in this listing are similarly interpreted. Elements of the lexicon are ordered pairs consisting of a string and a category, as is every other expression generated from the lexicon.

CME has three structure building functions: Casemark (CM), Predicate-Argument (PA), and Coordination (BOOL) given below. In defining each function we first give its domain and then its value at each element in its domain. Where $\sigma = \langle \text{s,C} \rangle$ is an expression we write string(σ) for its string coordinate, s, and Cat(σ) for its category coordinate, C. We use '+' for concatenation.

Casemark (CM):

$$\text{Dom(CM)} = \{\langle\langle \text{s,NP}\rangle, \langle \text{t,K}\rangle\rangle | \text{s,t} \in \text{V}^*\}$$

$$\text{CM}\langle \text{s,NP}\rangle,\langle \text{t,K}\rangle) = \begin{cases} \langle \text{s-t,KPn}\rangle & \text{if t = nom} \\ \langle \text{s-t,KPa}\rangle & \text{if t = acc} \end{cases}$$

Predicate-Argument (PA):

$$\mathrm{Dom(PA)} = \{\langle\sigma,\tau\rangle | (\mathrm{Cat}(\sigma) = \mathrm{KPa}\ \&\ \mathrm{Cat}(\tau) = \mathrm{P2})\ \text{or}\ (\mathrm{Cat}(\sigma) = \mathrm{KPn}\ \&\ \mathrm{Cat}(\tau) = \mathrm{P1})\}$$

$$\mathrm{PA}(\sigma,\tau) = \begin{cases} \langle \mathrm{string}(\sigma) + \mathrm{string}(\tau), \mathrm{S}\rangle & \text{if } \mathrm{Cat}(\sigma)\text{=KPn} \\ \langle \mathrm{string}(\tau) + \mathrm{string}(\sigma), \mathrm{P1}\rangle & \text{if } \mathrm{Cat}(\sigma)\text{=KPa} \end{cases}$$

Coordination (BOOL):

$$\mathrm{Dom(BOOL)} = \{\langle\rho,\tau,\eta\rangle | \mathrm{Cat}(\rho) = \mathrm{CONJ}\ \&\ \mathrm{Cat}(\tau) = \mathrm{Cat}(\eta) \in \{\mathrm{S,KPn,KPa,P1,P2}\}.$$

$$\mathrm{BOOL}(\rho,\tau,\eta) = \begin{cases} \langle \text{both+string}(\tau) + \text{and} + \mathrm{string}(\eta), \mathrm{Cat}(\tau)\rangle & \text{if } \rho\text{=}\langle\text{and,CONJ}\rangle \\ \langle \text{either+string}(\tau) + \text{or} + \mathrm{string}(\eta), \mathrm{Cat}(\tau)\rangle & \text{if } \rho\text{=}\langle\text{or,CONJ}\rangle \\ \langle \text{neither+string}(\tau) + \text{nor} + \mathrm{string}(\eta), \mathrm{Cat}(\tau)\rangle & \text{if } \rho\text{=}\langle\text{nor,CONJ}\rangle \end{cases}$$

Given this simple grammar, it is easy to prove basic facts like the following:

Facts about CME

a. ⟨*nom*,K⟩ and ⟨*acc*,K⟩ are both grammatical constants in CME. That is, for all structure maps h for L(CME), h(*nom*,K)=⟨*nom*,K⟩ and h(*acc*,K)=⟨*acc*,K⟩.

b. ⟨*and*,CONJ⟩ is not a grammatical constant in CME. There is a structure map which maps it to ⟨*or*,CONJ⟩.

c. For each category C the property of being an expression of category C is a structural property in CME. That is, for each such C and each expression $\sigma \in$L(CME), if Cat(σ)=C then for all expressions τ isomorphic to σ, Cat(τ)=C.

3 Axioms relating form and meaning

We now present four axioms concerning the relation between form and meaning in natural language which we hold to be universal.

Content Constraint (CC)	**Strong Compositionality**
Semantic Fixity (SF)	**Model Closure**

For reasons of space we just study the Content Constraint and the Semantic Fixity Constraint in this paper.

3.1 The Content Constraint

The Content Constraint, despite its name, is in fact purely syntactic, but it has a semantic motivation of the same sort we have for the "recoverability of deletions" condition.[2] It is stated formally as follows:

Content Constraint For F a generating function and $\delta, \delta' \in$ Dom(F), if $\delta \simeq \delta'$ & $\delta \neq \delta'$ then F(δ) $\neq$ F(δ').

CC says that the structure building functions of a grammar preserve the property of being distinct but isomorphic. The intuition is that since syntactically isomorphic expressions have meanings with similar compositional structure (see section 3.3 below for further discussion), the distinction between isomorphic expressions will typically signal some semantic distinction that should not be obliterated by any syntactic process. Distinct isomorphic expressions may (and usually do) have distinct denotations. For example in ordinary English, the proper nouns *Mary* and *Susan* are grammatically isomorphic and semantically comparable – both denote individuals in the universe of discourse, but they may denote different individuals. Similarly, the verbs *sing* and *dance* are distinct and arguably isomorphic, and they are semantically comparable in that both denote human activities. But in any given situation the individuals who are singing may not be those who are dancing. Now, taking *sing* and *dance* as distinct but isomorphic, the CC guarantees that their infinitival nominalizations, to sing and to dance are distinct (which they are), and their gerundive nominalizations, *singing* and *dancing*, are distinct (which they are). CC would be violated if English had a nominalizing operation ING* which derived a given form, say *blicking*, both from *sing* and from *dance*. Such a rule would lose the potential content distinction present in *sing* and *dance*.

[2] Chomsky (1965), Emonds (1985), etc.

From the assumption CC we can prove the following:[3]

(6) *Syntactic Fixity* Grammatical constants are either lexical items or they are derived from an appropriate number of grammatically constant lexical items

(6) allows for example that *neither... nor...* be a grammatical constant in ordinary English derived by negating the properly lexical constant *either... or...*. But (6) does severely constrain the acceptable derivations of grammatical constants. It fails for example if 'grammatical constant' is replaced by 'semantic constant'. For example, a derived expression such as *Either all swans are black or else some aren't* is semantically constant (always interpreted as True) even though it is derived from lexical items which include *swan* and *black* which are not semantic constants, just ordinary content items. CC rules out that we could build syntactic constants in such ways.

3.2 Model Theory

To explain the other general relations between form and meaning, it will be helpful to be more explicit about the nature of semantic interpretation. A minimal requirement on such a notion is that it be rich enough to characterize the entailment relation between expressions. At the level of sentence we say that a sentence S *entails* a sentence T iff T is interpreted as True in all the situations (models) in which S is interpreted as True.[4] A "situation" (or model) can be extensionally represented by a pair (E, μ) where E is a non-empty *universe* of (possibly abstract) objects about which we think of ourselves as speaking, and μ is a function which assigns to each lexical item a denotation defined in terms of E and the fixed set True, False of truth values. And for each model M = (E,μ) we define an *interpretation of* L(G) *relative to* M by saying how complex expressions are interpreted as a function of the interpretations of their parts, where the interpretation of the lexical parts is given by μ. Writing $[\![\cdot]\!]^M$ for the interpretation of L(G) relative to M we note that where σ is an expression of category Sentence $[\![\sigma]\!]^M$ is an element of True, False. And for σ and τ sentences we say that σ entails τ iff for all models M, if $[\![\sigma]\!]^M$ = True then $[\![\tau]\!]^M$ = True.

For example, consider simple models M = (E,μ) for CME. Let μ map lexical P1's to subsets of E and lexical P2's to binary relations over E (subsets of E×E). And μ maps the NPs to elements of E and expressions of category KPn to functions from the subsets of E into True, False. Finally, μ maps expressions of category KPa to functions from binary relations to sets (possible P1 denotations). In particular let it map ⟨*himself*,KPa⟩ to that function SELF from binary relations to sets given by:

$$\text{SELF(R)} = \{\text{a} \in \text{E} |\ \text{aRa}\}$$

Note that the sort of object an expression (in particular a lexical expression) can denote is determined by its grammatical category: P1s denote subsets of E, P2s subsets of E×E, Ss elements of {True, False}, NPs elements of E, KPn's functions from P(E), the set of subsets of E, into {True,False} and KPa's maps from P(E×E) into P(E). The denotations of ⟨*nom*,K⟩ and ⟨*acc*,K⟩ are given explicitly below, writing [A → B] for the set of functions from A into B:

(7) a. μ(*nom*,K) is that function NOM from E into [P(E) → {True,False}] given by:
NOM(b)(P) = True iff b ∈ P

b. μ(*acc*,K) is that function ACC from E into [P(E×E) → P(E)] given by:
ACC(b)(R) = {a ∈ E| aRb}

So, anticipating slightly, if in some model ⟨*John*,NP⟩ is interpreted as the object y and ⟨*criticize*,P2⟩ is interpreted as the binary relation CRITICIZE then ⟨*John-acc*,KPa⟩ will be interpreted as that function ACC(y) which sends CRITICIZE in particular to the set of objects x such that x stands in the CRITICIZE relation to y. So ⟨*criticized John-acc*,P1n⟩ will be the set of objects which stand in the CRITICIZE relation to John. ⟨*Bill-nom*,KPn⟩ will be interpreted as a function true of that set just in case the object that ⟨*Bill*,NP⟩ denotes is in that set.

[3]Syntactic fixity can be stated more precisely as follows:

If σ is a grammatical constant then either $\sigma \in$ Lex or for some F $\in F_G$ and some $\delta \in \text{Dom(F)} \cap \text{L(G)}^n$, $\sigma = \text{F}(\delta)$ and each $\delta_i \in \delta$ is a grammatical constant.

We prove that CC entails syntactic fixity as follows. Suppose that $\sigma \notin$ Lex. Then, from the definition of L(G), σ is derived; that is, $\sigma = \text{F}(\delta)$ for some generating function F and some sequence δ. We show that δ is grammatically constant. Suppose, leading to a contradiction, that δ is not constant. Then there is a structure map h such that $\text{h}(\delta) \neq \delta$. Now trivially $\delta \simeq \text{h}(\delta)$, since h is the desired structure map. But now the antecedent of CC is satisfied: $\delta \simeq \text{h}(\delta)$ and $\delta \neq \text{h}(\delta)$. So we infer that $\text{F}(\text{h}(\delta)) = \text{h}(\text{F}(\delta))$. Thus $\text{F}(\delta) \neq \text{h}(\text{F}(\delta))$. But $\text{F}(\delta)$ is σ, so this just says that $\sigma \neq \text{h}(\sigma)$, contradicting that σ is a grammatical constant. Thus δ must be constant after all.

[4]See, for example, Keenan & Faltz (1985) for a generalization of the entailment relation to the denotations of other sorts of expressions.

Finally we note that μ will interpret $\langle$*and*,CONJ$\rangle$, $\langle$*or*,CONJ$\rangle$ and $\langle$*nor*,CONJ$\rangle$ as the appropriate boolean operations. For example if σ and τ are expressions of category S then $\mu(and,\text{CONJ})(\sigma)(\tau)$ = True iff $\mu(\sigma) = \mu(\tau)$ = True. If σ and τ are of category Pn for n = 1 or 2, then $\mu(and,\text{CONJ})(\sigma)(\tau) = \mu(\sigma) \cap \mu(\tau)$, etc.

To define an interpretation of CME relative to a model M = (E,μ) we define the function $[\![\cdot]\!]^M$ from CME to CME as follows:

(8) For all σ in the language of CME,

a. if σ is in Lex then $[\![\sigma]\!]^M = \mu(\sigma)$;

b. if $\sigma = \text{CM}(\delta,\tau)$ then $[\![\sigma]\!]^M = [\![\tau]\!]^M([\![\sigma]\!]^M)$;

c. if $\sigma = \text{PA}(\sigma,\tau)$ then $[\![\sigma]\!]^M = [\![\tau]\!]^M([\![\sigma]\!]^M)$, and

d. if $\sigma = \text{BOOL}(\rho,\tau,\eta)$ then $[\![\sigma]\!]^M = [\![\rho]\!]^M([\![\tau]\!]^M,[\![\eta]\!]^M)$

Equipped with this basic notion of a model structure we can now meaningfully state the our additional axioms about form and meaning.

3.3 Compositionality and model closure

A standard notion of compositionality can be given as follows:

Ordinary Compositionality (OC) For all models (E,μ), all generating functions F, there is a function F' such that $\forall\delta \in \text{Dom(F)}, \ \mu(\text{F}(\delta)) = \text{F}'(\mu(\delta))$.

Intuitively, this says that for each way of building a syntactic structure, there is a corresponding way to build the semantic value of the complex from the semantic values of the parts. (Of course, if δ is an n-tuple of expressions $\langle\delta_1,\ldots,\delta_n\rangle$ then by $\mu(\delta)$ is meant $\langle\mu(\delta_1),\ldots,\mu(\delta_n)\rangle$).

We propose a stronger idea here, according to which the semantic function corresponding to any syntactic generating function does not vary from one model to another. So, for example, if a certain combination of phrases is interpreted as predication (perhaps formally realized as a certain pattern of function application, as in 8c, above), then it is interpreted that way in every model. More precisely:

Strong Compositionality (SC) For G a grammar, F a generating function, and (E,μ) and (E,μ') models of L(G), if δ and δ' are in the domain of F then if $\mu(\delta) = \mu'(\delta')$ then $\mu(\text{F}(\delta)) = \mu'(\text{F}(\delta'))$.

In these terms, we easily establish that SC implies (but is not implied by) OC.

Notice that neither OC nor SC says anything directly about the finiteness of the set of generating functions F nor about the finite representability of the corresponding semantic mechanisms. However, if there are finitely many syntactic generating functions F, SC but not OC entails that the there is just one finite set μ[F] of corresponding interpretive functions for all models. Still, it is not clear that SC inherits the computational motivation popularly associated with compositionality. That is, we finite speakers must clearly have a finite representation of the language: a finite representation of Lex and F is our only means of accounting for how finite creatures like us can understand novel utterances. But it is not clear that we 'evaluate' μ or anything like it.

A closely related idea is:

Model Closure (ISOM) The class of models for a language L(G) is closed under isomorphism. If π is a bijection with domain E, then if (E,μ) is a model so is $(\pi(E),\pi(\mu))$, where $\pi(\mu)$ is defined by: $\pi(\mu)(d) = \pi(\mu(d))$.

ISOM can be shown to block potential interpretations of natural language like that partially instantiated in (9) where P is a P1, 'c' is an individual constant (proper noun) and 'Pc' is an expression of category Sentence.

(9) $\mu(\text{Pc})$ = T iff either $\mu(\text{c})$ = j and $\mu(\text{c}) \in \mu(\text{P})$ or ($\mu(\text{c}) \neq$ j and $\mu(\text{c}) \in \mu(\text{P})$).

A function like μ above depends on certain objects being in the universe and having certain properties. If we allowed functions like μ the interpretation of derived expressions F(δ) would be dependent on things other than the interpretation of the expressions δ. Trading in one universe with j in it for another lacking j may allow bijections π as in ISOM such that the map $\pi(\mu)$ fails to be an interpretation.

A careful discussion of SC and ISOM is beyond the scope of this paper, and these have been considered before. The content constraint CC and semantic fixity, discussed in the next section, have not been proposed before.

3.4 Semantic Fixity

We turn now to a final general restriction on form and meaning, one whose consequences we explore in somewhat more detail. This restriction concerns semantic constants, expressions σ with the property that in each situation in which we use the language there is only one way of interpreting σ. Formally,

Definition Given a grammar G, an expression d $\in$ L(G) is a *semantic constant* iff for all models M=(E,μ) and M'=(E,μ'), $[\![d]\!]^M = [\![d]\!]^{M'}$.

In other words, given a universe E, all models with universe E interpret d the same way.

For example, a predicate like *equals*, =, in mathematical discourse, is fixed in denotation. Given a universe E of objects under discussion, a pair (x,y) of objects stands in the = relation iff x and y are the same object. In the same way, the mathematical quantifier *exists* is fixed in denotation given E since the objects that exist are exactly the elements of E. In general "logical" words, like *not, or,* and *every*, have their denotations fixed once the universe is given. This is not purely an arbitrary convention. These expressions are distinctive in that their denotations have distinctive properties. Their denotations are logical constants in the sense that they satisfy a very strong condition, known as permutation (or automorphism) invariance (PI). Roughly this says that their denotations remain unchanged if we trade in some individuals for others. That is, their denotations do not depend on which individuals have which properties or stand in which relations to others. For example, given a small universe E, let us list all the pairs $\langle$x,x$\rangle$that stand in the = relation. Now consider a "substitution" (= permutation) of E, that is, a bijection π from E to E If we go through our list and replace each pair $\langle$x,x$\rangle$ with the pair $\langle\pi(x),\pi(x)\rangle$ we find that the members of the new list are exactly the members of the old list (though their relative order in the written list may have changed). In this sense then interchanging pairs of objects systematically does not change the pairs that lie in the = relation. Nor will it change the objects with the existence property. In contrast, the denotation of a predicate like *sing* will vary according to who is singing. Then where b is singing and a is not, a permutation π which interchanged a and b (leaving everything else fixed) would change SING. Replacing the elements x in SING with π(x) the resulting list is different: $\pi(SING) \neq SING$.

These rather technical observations lead to a linguistic observation of modest interest. Namely, the range of semantically constant expressions in a given category is limited by the number of permutation invariant objects in the denotation set associated with that category. Whenever the universe E has at least two elements the number of PI elements of E is zero. Thus there can be no logical constants among the proper nouns and (recalling our introduction) any "universals" blocking this are not arbitrary at all). Moreover there are only two PI subsets of E, namely E and $\emptyset$, and so there are just two extensionally distinct one place predicates (P1s) which are semantically constant. We can represent them by *exist* and *not exist.* Similarly there are at most four distinct PI binary relations over E, of which = and $\neq$ receive natural expression, though the empty binary relation is expressible with phrases such as *is taller than but not as tall as* and the set of all pairs is denotable by expressions like *either is taller than or else isn't taller than*, etc.[5]

We propose that this kind of fit between syntax and semantics holds in human languages as well. In particular, we propose:

Semantic Fixity (SF) Grammatical constants are semantic constants.

Turning to CME, with models as defined in section 3.2, the coordinators $\rangle$*and*,CONJ$\rangle$, $\langle$*or*,CONJ$\rangle$, $\langle$*nor*,CONJ$\rangle$ are semantic constants. We observed in section 2 that these are not grammatical constants in CME, though it is plausible that in a more English-like grammar they would be. The lexical item $\langle$*himself*,KPa$\rangle$ is similar: it is a semantic constant, but not a grammatical constant in CME because no process in CME distinguishes himself from him. In the grammar of English, these two elements would be distinguished. By contrast, given an arbitrary universe of individuals, any of them could, in principle, be denoted by proper nouns like *Dana, Robin, or Pat.* So proper nouns are not semantically constant. Their denotations can vary relative to a fixed universe. Similarly one place predicates like *sleep* and *laugh* are not semantically constant. Given a universe including a,b, and c it might be that just b is laughing, or just both b and c, or none of them, etc. Finally, it is important to note that expressions like $\langle$*both John and neither John nor Bill*,KPn$\rangle$ are semantic constants. These are not grammatical constants, but SF does not suggest in any way that all semantic constants are grammatical constants. SF makes the converse claim, that grammatical constants are semantic constants.

Current work suggests that SF will be derived from a more general principle concerning the semantic fixity of the generating functions. But even taking SF as axiomatic, we still find two consequences of interest. First, SF severely constrains the form of a grammar for a given language. It might seem trivial for example to assume we could add

[5]See Keenan (1995) for an extensive discussion of permutation invariance and logical constants.

to a grammar of English a function ϕ that would do nothing more than derive the NP John from John. But if the domain of ϕ has just one element then provably that element is a grammatical constant (since structure maps must preserve the domains of the generating functions), whence by SF it must be a semantic constant. But in fact the NP John is not semantically constant. Hence ϕ, and functions like it, are not acceptable in a grammar of English, or of natural languages in general.

A second interesting consequence of SF comes from the fact that causative and applicative (Benefactive, Locative, Instrumental) affixes on verbs are (arguably) grammatical constants. By SF they must be semantic constants. But they are not logical constants, like *not, and, =,* etc. Whether John brought about a certain action, or whether an act was done for his benefit, are simply contingent facts about the world. They may be so in certain situations with a given universe and false in others with the same universe. We are thus led on linguistic grounds to posit a much richer model structure for natural languages than those used in standard logical study, one in which notions like AGENT, BENEFACTEE, INSTRUMENT,... are primitives and must be preserved by any structure preserving semantic operations.

We observed that logical constants, those elements with PI denotations, are not arbitrarily distributed among categories. But properly linguistic constants seem less constrained, though their distribution is less well understood. One case where a linguistically expressed contrast arises concerns valency affecting operations in natural language. Thus affixes which combine with two place predicates P2s to form P1s are often logical constants, or basically so allowing minor modification. Examples are Passive and Reflexive illustrated below.

(10) a. Mikango yanu i-na-thamangits-a mbuzi zathu — **active**
lions your they-pst-chase-asp goats our — (Chičewa; Baker)
'Your lions chased our goats'

b. Mbuzi zathu zi-na-thamangits-idw-a (ndi mikango yanu) — **passive**
goats our they-pst-chase-pass-asp by lions your
'Our goats were chased (by your lions)'

In (10b) the passive affix *-idw-* has combined with the TVP *thamangits* 'chase' to form a VP, *thamangits-idw-* 'be chased'. So *-idw-* has reduced by one the valency (= number of arguments required) of the original predicate. And for P a transitive verb the interpretation of P+*idw-* should be predictable from that of P, by Compositionality. To within a first approximation this dependency is given by:

(11) For P a TVP and M a model, $[\![\text{P}+idw\text{-}]\!]^M = \{\text{b} \mid \exists\text{a}\ \langle\text{a,b}\rangle \in [\![\text{P}]\!]^M\}$

Thus the predicate in (12b) from Kinyarwanda (Kimenyi, 1978) is intransitive, derived from the transitive predicate in (12a) by prefixing the verb root with ii.

(12) a. Umukoôbwa a-ra-reeb-a umuhuûngu
girl she-past-watch-asp boy
'The girl is watching the boy'

b. Umukoôbwa á-r-íi-reeb-a
girl she-past-self-watch-aspect
'The girl is watching herself'

And as with Passive, we can think of Reflexivization as a way of deriving a P1 from a P2 (and more generally a Pn from a Pn+1). A first pass at its semantic interpretation is:

(13) For P a TVP and M a model, $[\![\text{ii} + \text{P}]\!]^M = \{\text{a} \mid \langle\text{a,a}\rangle \in [\![\text{P}]\!]^M\}$.

The functions which interpret *-idw* and *ii-* are provably PI. By contrast functions which increase the arity of predicates are never basically PI. The most typical cases we are aware of, such as causatives and applicatives, associate a designated argument of the derived predicate with a specific semantic role – AGENT, BENEFACTEE, etc. We illustrate an instrumental from Kinyarwanda:

(14) a. Úmwáana a-rá-ry-a
child he-pres-eat-asp
'The child is eating'

b. Úmwáana a-rá-rí-iish-a ikánya
child he-pres-eat-inst-asp fork
'The child is eating-with the fork'

Here suffixation of *-iish* 'instrumental' to the intransitive verb in (14a) yields a transitive verb in (14b). But of course the binary relation denoted by the latter is not unrelated to the set denoted by the former. It satisfies (15), so if a child is eating with a fork, he is eating:

(15) For M a model and P a VP, if $\langle a,b \rangle \in [\![P+iish\text{—}]\!]^M$ then $a \in [\![P]\!]^M$.

More generally the interpretation of such valency increasing operators is given by (16), where p is an n-place predicate and δ is a "theta role," that is, a relation between individuals and predicate denotations:

(16) $F_{\delta,i}(R)(a1)\ldots(an+1) = T \Rightarrow a_i\ \delta\ R$

Roughly, we add argument position i with theta role δ, so that $a_1 \ldots a_{n+1}$ is mapped to True only if a_i does, in fact, bear δ to the relation R.

The sorts of relations expressed by causative and applicative operators are not PI in the classical sense. But linguistically the sense of AGENT, etc. is not felt to vary with contingent matters of fact in the same way that e.g. who is criticizing who varies from situation to situation. We may represent this by taking such relations as part of the model structure for natural language. Thus alongside truth values and a set of entities (and probably regions, temporal intervals, possibly events) we shall put relations like AGENT, etc. among the primitives of a model and just consider permutations of E which fix these relations. That is, which are such that $(b,R) \in$ AGENT iff $\langle \pi(b),\pi(R) \rangle \in$ AGENT. (Which is just to say π(AGENT) = AGENT). Expressions whose denotations are fixed (mapped to themselves) by all permutations of the universe that fix the primitive theta roles can be called semantic constants, in analogy with the logical constants whose denotations are fixed by all permutations. It is with this understanding of semantic constant that we interpret Semantic Fixity.

3.5 A remark on Emonds (1985) on grammatical constants

Emonds (1985, p.168n11) seems to identify something similar to our grammatical constants when he refers to "closed class items" and "designated elements." He suggests at one point that these elements are "lacking any purely semantic feature." Now we have a cogent way to formulate and understand this sort of claim. The point is not that these elements are lacking in semantic properties. Logical constants like *seventy-seven* and *eighty-seven* are not grammatically distinct, but they are clearly semantically distinct. But this is not a problem for SF, since the implication relation goes only in the other direction: grammatical constancy implies semantic constancy. In this sense, the grammatical constants have the special property in the language, not the semantic constants. Number names are simply among the infinitely many expressions that always denote PI elements of their denotation set but which are not grammatical constants. Thus we do not claim that semantic constants must be syntactic ones in natural languages, and we do not find this idea a reasonable constraint on natural languages. It would force all semantically unique behavior to be coded in the syntax.

More should ultimately be said about semantic fixity. What, exactly, is the motivation for this constraint? We must leave this question open here. If we had a convincing answer we would not need to take SF as an axiom, we would derive it from more primitive assumptions. And as we indicated earlier we think that in a correctly formulated system of form-meaning constraints SF will indeed be a theorem not an axiom. But still, why is it reasonable to expect it to hold (regardless of whether it is primitive or derived)? Our feeling here is that in practice the presence of a grammatical constant in an expression correlates with the application of some particular generating function. So the constant tells us how the derived expressions it occurs in were derived. We can in fact always modify the grammar so that the constant is not in the Lexicon but is introduced as part of the value of the function. And by Compositionality the interpretation of the derived expression should be determined once we know the function that applied and how its arguments were interpreted. But this would not be the case if the grammatical constant could have many different interpretations. It would be equivalent to saying that a given way of building a complex expression corresponded to many ways of interpreting the complex expression for fixed ways of interpreting the arguments. In effect what we are saying is that the role of grammatical constants is purely grammatical, their semantic contribution, which may be very non-trivial, is constant.

The view we are led to is opposed to both the extreme conventionalist view and the arbitrary universals view discussed in the introduction to this paper. But it is compatible with even quite strong "autonomy of syntax" views, including ones that require "semantics" to be "derivative" in the sense of being fundamentally interpretive, taking syntactic expressions as input. Indeed the semantic analysis we use here is precisely one of that sort.

References

Mark C. Baker (1988) *Incorporation.* Chicago: Chicago University Press.

Noam Chomsky (1965) *Aspects of the Theory of Syntax.* Cambridge, Massachusetts: MIT Press.

Noam Chomsky (1973) Conditions on transformations. In S.R. Anderson and P. Kiparsky, eds., *A Festschrift for Morris Halle.* New York: Holt, Rinehart and Winston.

Joseph Emonds (1985) *A Unified Treatment of Syntactic Categories.* Dordrecht: Foris.

Edward L. Keenan (1995) Logical objects. forthcoming.

Edward L. Keenan and Leonard M. Faltz (1985) *Boolean Semantics for Natural Language.* Dordrecht: Reidel.

Edward L. Keenan and Edward P. Stabler (1995) Abstract syntax. forthcoming.

Alexandre Kimenyi (1980) *A Relational Grammar of Kinyarwanda.* Berkeley: University of California Press.