Language Invariants
The Syntax and Semantics of Case Marking

Edward L. Keenan

University of California, Los Angeles

1. Introduction

In the spirit of the theme of this conference, Linguistics in the Twenty-First Century, I will consider what I consider to be two problems with current linguistic theories, specifically with Government Binding/Principles and Parameters theory and its extension in the Minimalist Program (Chomsky 1996, Hornstein 1995). We may reasonably expect solutions to specific instances of these problems in the next decade; perhaps even the general cases will be handled. Certainly much progress has been made in the last decade, since we may now at least formulate the problems with some clarity. In this paper we propose answers to one instance of each of these problems. We draw directly on *Bare Grammar* (Keenan & Stabler, 1997 and to appear), henceforth *BG*.

1.1. The integration problem refers to the absence of an integrated syntactic and semantic analysis of expressions. The theories cited above lack an explicit mode of semantic interpretation; they cannot even define fundamental semantic notions like *entailment*. Yet the theories make a variety of assumptions about semantic interpretation which motivate certain properties of syntactic analysis. Some instances drawn from the current literature are:

(1) a. Wh- elements must move to be interpreted
    b. If A antecedes an anaphor B then A C-commands B
    c. Case is uninterpretable.
    d. The LFs of different languages are formally similar

The claims in (1) relate the syntactic analysis of expressions to their semantic interpretations. One might expect them to follow from an integrated theory. But in the absence of a mode of semantic interpretation we might at best take them as axioms imposing boundary conditions on semantic interpretation.

The Linguistic Society of Korea (ed.), 1999,
Linguistics in the Morning Calm 4, 21-39.

But normally in theory construction the statements we take as axiomatic are felt to be basic truths, whereas the claims in (1) seem largely arbitrary. We can surely construct an integrated theory in which wh-elements are interpreted in non-argument positions, but why should we? Is movement to such positions essential? Is it intended that interpreting wh-elements in situ is logically or conceptually impossible (which is false)? Just why should (1a) be an axiom?

Even (1d), more global than (1a-c), is *prima facie* implausible. LFs for a given language L are constructed from the syntactically analyzed expressions of L and are intended to represent at least the "grammatically determined aspects of meaning" (Hornstein 1995: 7) associated with the expressions they are derived from. But expressions of different languages are different. What forces their LFs to collapse to the same or similar expressions? The study of semantically interpreted languages, e.g. sentential logic, predicate logic, shows that an arbitrary formula is interpreted semantically the same as infinitely many syntactically distinct ones. For example standard logical representations of the (a, b) pairs below are logically equivalent.

(2) a. If John's father is Greek, so is his mother
b. If John's mother isn't Greek, his father isn't either

(3) a. Not every student answered question 6.
b. Some student didn't answer question 6.

(4) a. Each student read at most two plays over the vacation
b. No student read more than two plays over the vacation

But there is no semantic reason to say that the LFs of (2a) and (2b) are identical; their semantic equivalence follows from the compositional interpretation of their parts. Analogous claims hold for (3a, b) and (4a, b). Thus expressions with different structures may have the same meaning, so we can in principle account for the semantic similarity among languages without having to claim that their LFs are identical. What motivates an "axiom" like (1d)?

Our specific concern here is (1c). BG exhibits a model of case marking which semantically interprets it in a way that is fundamental to understanding nuclear sentences. It is case marking, not linear order or hierarchical relations (C-command), that enables us to identify DP denotations with semantic arguments of predicates. It also yields interpretations in which anaphors in nuclear clauses may asymmetrically C-command their antecedents, contra (1b).

1.2. The structure vs. notation problem concerns with the absence in current theories of a notation free conception of language *structure*. For many theoreticians characterizing the structure of an expression without referring to a sketch of a labelled tree would be unthinkable, so tightly is our notion of structure tied to particular notational conventions. It is unsurprising then that many generalizations in current theories are primarily claims about the notation we may use to describe language structure rather than ones about language structure itself. I would include here

(5) a. Expressions are right branching
b. Expressions are at most binary branching
c. Specifiers precede heads, complements follow
d. Trace theory
e. A pronoun cannot be linked to a variable that doesn't C-command it.

The particular case of this problem we are concerned with here is the inability of current theories to represent the structural role of morphology independent of constituent structure. Even when nominative and accusative DPs in a language are regularly distinguished by morphology, current theories still distinguish them by their positions in trees. But BG provides a rigorous notion of *structure* in which identity of morphemes is structural in exactly the same sense in which notions like *is a constituent of* and *C-commands* are structural. Moreover, identity of morphemes and constituent structure may vary independently.

In our model the local Anaphor-Antecedent relation in Korean is structurally defined, based in part on morpheme identity, even though anaphors sometimes (but not always) asymmetrically C-command their antecedents. What counts is whether the case endings on the anaphor and its putative antecedent satisfy certain conditions, not whether the two DPs stand in a C-command relation or not.

1.3. Interlude: structure independent of notation This idea is not so familiar, so we shall give a variety of quickly sketched examples from different areas, not all of which will be familiar to all readers.

example There are a variety of notations and mildly different algorithms for representing division of numbers. Here are instances of a few I found:

HUNGARY
```
203 : 7 = 29
 63
```

BRAZIL
```
203 |_7_
 63  29
```

ITALY
```
203 | 7
 14   29
 63
 63
```

USA / CHINA / KOREA
```
   29
7 |203
   14
    63
    63
```

	~~6~~	
12th CENTURY ITALY/INDIA: 7	~~2~~~~0~~~~3~~	29
	~~1~~4~~3~~	
	~~6~~	

These examples have in common that they record our search for a number x with the property that if we multiply 7 by x we obtain 203. The generalization they all express is that 203 *divided by* 7 is 29. We could doubtless define some syntactic operations which would derive the Hungarian computation from the USA one or vice versa. But this notation juggling would in no wise account for what the two processes have in common. Rather, the explanatory account is that both notations express the division of one number by another. And this same explanatory account covers the case below as well, despite the fact that we have changed from decimal to fractional notation:

USA $3/4 \div 2/3 = 3/4 \cdot 3/2 = (3 \cdot 3)/(4 \cdot 2) = 9/8$

The algorithm used in this last case is significantly different from that used in the first USA case, but the generalization is the same: from given numbers m and n, both algorithms compute a number x such that $m \cdot x = n$.

example We commonly measure temperature in Celsius or in Fahrenheit. But the judgment that X is cooler than Y remains invariant under changes of scale and notation. Water has only one freezing point; it doesn't matter whether you call it 0 degrees Celsius or 32 degrees Fahrenheit.

example We use sentential logic (SL) to study boolean operations (expressed for example by *and*, *or*, and *not* in English). Using a common infix notation we prove in SL that the formula ((A or B) & $\neg$A) entails B. In prefix (Polish) notation we write that & or AB $\neg$A entails B. The judgment of entailment we express is the same in both cases, but the notation used to express the application of the boolean operations is (slightly) different. And we could, again without changing the logical reality we express, trade in the prefix notation for the corresponding suffix one (reverse Polish).

Examples of this sort can be multiplied many times over. And they support that distinct notations can express the same reality. In consequence we should be cautious about tying our generalizations too closely to a particular notation.

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

We turn now to the characterization of the notion *structure of an expression* as presented in BG and the particular structural and semantic role of nominal case marking it affords.

2. Bare Grammar

We present a formal conception of linguistic structure which provides a unified notion of *structure* in which morphological relations and classical constituency (hierarchical) ones are independently structural, with neither characterizable in terms of the other. We exhibit a model of a language, Little Korean (LK), in which Argument Structure (matching expressions to semantic arguments of predicates) and the Anaphor-Antecedent relation are determined by relations between case endings, not hierarchical structure or linear order. Our presentation will be informal; we refer the reader to *BG* for a fuller presentation.

2.1. Here first are some observable properties of Korean which motivate our grammar. In (6) we see that the relative preverbal order of arguments of a transitive verb is fairly free. These arguments carry postpositions glossed -NOM and -ACC. -NOM has the shape *-ka* on vowel final NPs and *–i* otherwise. Similarly -ACC = *-ul* consonant finally and *-lul* vowel finally. The Topic marker *-nun/-un* is similarly conditioned.

(6) a. John-i Mary-lul pinanhayssta
John-NOM Mary-ACC criticized
John criticized Mary

b. Mary-lul John-i pinanhayssta
Mary-ACC John-NOM criticized
John criticized Mary

The correlation of this word order freedom with overt case marking is built into LK in a way that is not consistent with standard assumptions in the GB/P&P tradition. Specifically in LK (6a, b) have the same degree of syntactic complexity: Their *tree* structures are isomorphic and they do not differ with regard to the movement history of their constituents.

Our major claim about this word order variation is that (6a, b) are free variants of each other, rather than one being syntactically and semantically understood in terms of the other. By contrast there are a variety of syntactic and semantic/pragmatic differences between active SVO sentences like (7a) in English and various grammatical ways of presenting the object before the subject, as Passive in (7b) and Topicalization in (7c).

(7) a. John criticized Mary
b. Mary was criticized by John
c. Mary(,) John criticized

The word order variation in Korean is not like that active-passive variation in (7a, b). For example, in (8a, b) the *-(l)ul* marked NP, the Patient, can be a (complex) reflexive, but in English (9a), the Patient NP cannot be reflexively bound to the Agent.

(8) a. John-i caki-casin-ul pinanhayssta
John-NOM self-emph-ACC criticized
John criticized himself
b. Caki-casin-ul John-i pinanhayssta
self-emph-ACC John-NOM criticized
John criticized himself

(9) a. *Himself$_i$ was criticized by John$_i$.
b. Himself$_i$ John$_i$ likes (no one else)

But (9) allows that the Korean variation might be like English Topicalization. The data in (10) - (17) however argue against this:

First, the antecedent of the reflexive may be quantified, (10, 11), or interrogative, (12), and the order of anaphor an antecedent is still free:

(10) a. Nwukwunka(-ka) caki-casin-ul pinanhayssta
someone-NOM self-emph-ACC criticized
Someone criticized himself
b. Caki-casin-ul nwukwunka(-ka) pinanhayssta
self-emph-ACC someone-NOM criticized
Someone criticized himself

(11) a. (Motun) haksayng-tul-i caki-casin-ul pinanhayssta
(all) student-pl-NOM self-emph-ACC criticized
(All) the students criticized themselves
b. Caki-casin-ul (motun) haksayng-tul-i pinanhayssta
self-emph-ACC (all) student-pl-NOM criticized
(All) the students criticized themselves

(12) a. Nwuka caki-casin-ul pinanhayss-ni
who self-emph-ACC criticized-?
Who criticized himself?
b. Caki-casin-ul nwuka pinanhayss-ni
self-emph-ACC who criticized-?
Who criticized himself?

But Topicalization of reflexives in these contexts in English varies from marginal to bad:

(13) a. *?Himself$_i$ someone$_i$ criticized b. *Himself$_i$ who$_i$ criticized?

Secondly, the reflexive-first order is natural in subordinate clauses in Korean, whereas Topicalization in English is largely a root clause phenomenon:

(14) a. Himself$_i$ John$_i$ criticized at the meeting
b. *the meeting at which himself John criticized

(15) a. Caki-casin-ul John-i hoyuy-eyse pinanhayssta
self-emph-ACC John-NOM meeting-LOC criticized
John criticized himself at the meeting

b. Caki-casin-ul John-i pinanha-n hoyuy-ka ecey iss-ess-ta
self-emph-ACC John-NOM criticized-adnom meeting-NOM
yesterday exist-past-decl
there was a meeting yesterday at which John criticized himself

Thirdly, the Topicalized (7c) differs strikingly from (7a). It logically entails it, but in distinction to (7a) it contrasts the Patient with other candidates in context. In Korean, merely placing an NP in clause initial position does not force contrast or emphasis. Rather in such cases we mark the focussed NP with the "topic" marker -(*n*)*un* and place it easily in the immediate preverbal position.

(16) John-i Mary-nun pinanhayssta
John-NOM Mary-TOP criticized
John criticized MARY (not someone else)

Lastly, in contrast to word order, relative case marking of anaphor and antecedent does not naturally vary (O'Grady 1985; Park 1986; Keenan 1988). A -NOM (*-i/-ka*) or a topic marked (*-un/-nun*) NP can antecede an -ACC (*-ul/-lul*) or -DAT (*-eke*) one, but we do not in general find -ACC/-DAT marked NPs interpreted as antecedents of -NOM marked reflexives (even if they precede them). Thus the expressions in (17) are generally bad, but reversing -NOM/-TOP and -ACC/-DAT marking produces fully grammatical ones.

(17) a. *Haksayng-tul-ul caki-casin-i pinanhayssta
student-pl-ACC self-emph-NOM criticized
The students criticized themselves
b. *Nwukwu-lul caki-casin-i pinanhayss-ni
who-ACC self-emph-NOM criticized-?
Who criticized himself?
c. *Caki-casin-i pinanha-n haksayng-ul manna-ss-ta
self-emph-NOM criticize-Adnom student-ACC meet-Past-Decl
I met the student who criticized himself

d. ??John-ul caki-casin-i pinanhayssta
John-ACC self-emph-NOM criticized
John criticized himself

e. *Caki-casin-un John-eke silmanghayssta
self-emph-TOP John-DAT disappointed
John was disappointed in himself

(18) **The Local Anaphor-Antecedent Relation in Korean**
In a transitive S, a constituent x is a possible antecedent of a constituent y iff x and y are co-arguments and y is suffixed -*(l)ul*

(18) of course does not pretend to be a complete characterization of the AA relation in Korean. We have not considered reflexives in ditransitive Ss, or in adjuncts, or as possessors, not to mention the long distance anaphoric possibilities of bare *caki*. But (18) suffices to challenge current theories, and more extensive treatments must be extensionally equivalent to it for the restricted data set (transitive Ss) it covers.

2.2. Little Korean

In general, we think of a *grammar* G as a way of defining and interpreting a set of expressions—that is, a *syntax* and a *semantics*. Our notion of syntax is *generative* (as per recent work in the Minimalist Program but in distinction to the constraint satisfaction systems presented by GB theory and HPSG). That is, a set of expressions is defined by giving an initial (finite) set of *lexical items* and a finite set of "rules", that is, functions, called *structure building* or *generating* functions. L_G, *the language generated by G*, is the set of expressions obtainable from the lexical items by applying the generating functions finitely many times. Lexical items, like expressions in general, are structured: they consist of a phonological (or gestural) sequence plus a category label. So we write <*John*, NP>, or $[John]_{NP}$, to represent the expression of category NP whose string part is *John* (often omitting the category label entirely when no confusion results).

To define the syntax of a grammar G it is necessary, and sufficient, to define four things: the set V_G of *vocabulary* items of G, the set Cat_G of *categories* of G, a subset Lex_G of $V^*\times Cat$ (we omit subscripts when no confusion results) and the set $Rule_G$ of *rules* of G. In principle these four components can vary independently.

BG imposes no structure on what we observe and imposes no limits on what we can describe. Any general constraints on the form or interpretation of natural languages must be stated explicitly, none follow from the form of the notation used:

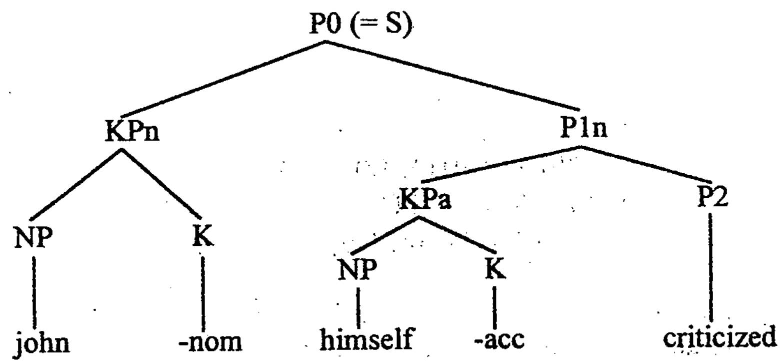
THE EFFABILITY THEOREM

I. For all sets V,CAT and all subsets L of $V^*\times CAT$ there is a G such that $L_G = L$

2. For any set V, any subset K of V* and any object C, there is a G such that the set of strings of category C is K.

We now present the grammar LK. We use English morphemes, such as *himself* for *caki-casin*, to emphasize that we are modeling a type of system of which Korean is merely a plausible instantiation. If further work on Korean motivates a different analysis we will still have shown one way in which syntactic relations can be formally handled morphologically without reducing morphology to syntax. We exhibit some crucial expressions generated by our grammar, and then we comment on their novel features including the semantic interpretation of the case markers. LK generates (8a) as represented in (19):

(19)

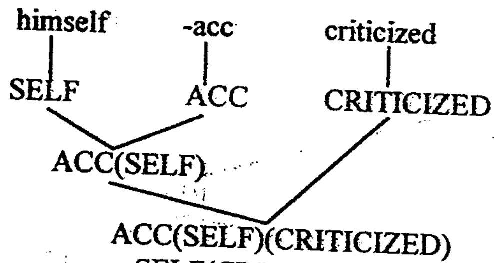


Syntactically, *criticized* (we ignore tense) is a lexical string of category P2, *two place predicate*. A generating function Case-Mark combines lexical NPs with the Kase Marker <*-acc,K*> to form KPa's such as <*john-acc*,KPa> and <*himself-acc*,KPa>. Case-Mark also derives KPn's such as <*john-nom*,KPn> from certain NPs and <*-nom*,K>. Of note is that the pair <*himself*,NP> and <*-nom*,K> is not in the domain of Case-Mark, so LK does not generate <*himself-nom*,KPn> and the result of interchanging *-nom* and *-acc* in (19) is not an expression of LK. A rule called PA (Predicate-Argument) combines KPa's, *accusative Kase Phrases*, with P2's to form strings like *himself-acc criticized* of category P1n, the category of *nominative one place predicates*. It also combines KPn's with P2's to form P1a's <*accusative P1s*>. So there is a kind of "case disagreement" built into the rules here. Also, for $\alpha \in \{n,a\}$, PA combines P1α's with KPα's to form Ss, enforcing a kind of "case agreement" here.

Semantically we interpret P2's as binary relations over the domain of objects under discussion. *criticize* is a lexical P2 (that is, <criticize,P2> is in the Lexicon of LK). For each domain E, <*criticize*,P2> denotes a set of pairs $<b,c>$ of elements of E, those for which b stands in the CRITICIZE relation to c. We write bCRITICIZEc for $<b,c> \in$ CRITICIZE. We interpret P1n's, such as *himself-acc criticized*, as subsets of the domain E (the set of those objects which have the property expressed by the P1). In (19) this is $\{b \in E | b\text{CRITICIZE}b\}$, the set of objects b which stand in the CRITICIZE relation to themselves. As the KPa *himself-acc* combines with the P2 *criticized* to form the P1n *himself-acc criticized* we shall interpret it as that function SELF which maps each binary relation R to $\{b \in E | bRb\}$, the set of those objects b in E which are related by R to themselves.

Now *himself-acc* is syntactically complex, consisting of the lexical NP *himself* and the K ("case marker") *-acc*. We obtain its denotation compositionally (in a trivial way in this case) by interpreting the NP *himself* as SELF and interpreting *-acc* as the identity function. Noting denotations in upper case we represent the compositional interpretation of the P1n in (19) by:

(20)



= SELF(CRITICIZED)
= $\{b \in E \mid b\text{CRITICIZED}b\}$

ACC is the identity map
def SELF

LK also provides us with some lexical P1n's, such as *laughed* and *cried*. And, using the third and last generating function BOOL, it forms boolean compounds of P1n's such as *both laughed and himself-acc criticized*, *neither laughed nor himself-acc criticized*, etc. The interpretation of such boolean compounds is done directly in the obvious set theoretic way: a conjunction of P1n's denotes the intersection of the sets denoted by each conjunct; a disjunction their union; a negation of a P1n, such as *not laughed*, its complement relative to the universe, that is, the set of objects in E which are not in the denotation of the P1.

How now do we interpret the KPn *john-nom*? It will be a function taking, among other things, P1n denotations (sets) as arguments and yielding as values S denotations, here truth values T ("true") and F ("false"). The functions we want are those called *individuals*:

(22) **Def** Given a domain E,

a. for each $b \in E$, I_b is the following function which maps unary relations (subsets of E) to zero-ary relations (truth values) and binary relations to unary relations (and more generally, but not our concern here, n+1-ary relations to n-ary relations):

i. for each subset P of E, $I_b(P) = T$ iff $b \in P$

ii. for each binary relation R on E, $I_b(R) = \{a \in E | aRb\}$

b. a function f is an *individual* (*over* E) iff for some $b \in E$, $f = I_b$

So in a situation in which *john* denotes I_j say, the sentence *john laughed* denotes T iff the object j is in the LAUGH set. Similarly *criticized john* denotes the set of objects which stand in the CRITICIZE relation to j.

Returning now to the interpretation of (19) we interpret the NP *john* as an individual and we interpret *-nom* as a function NOM mapping possible NP denotations (which we have not yet fully defined; we have only given some instances of them) G to functions which take sets among their arguments and map them to truth values, as per (23):

(23) NOM(G)(X) = G(X)

(24) gives a complete compositional interpretation of (19), where we assume that j is an element of the domain and that the NP *john* is interpreted as I_j:

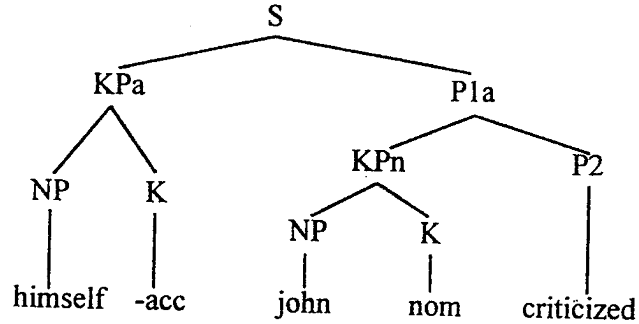
(24) NOM(I_j)(ACC(SELF)(CRITICIZE))

= NOM(I_j)($\{b \in E \mid b\text{CRITICIZE}b\}$)	def ACC, def SELF
= I_j($\{b \in E \mid b\text{CRITICIZE}b\}$)	def NOM
= True iff $j \in \{b \in E \mid b\text{CRITICIZE}b\}$	def individual
= True iff jCRITICIZEj	Set Theory; $j \in E$

Thus (8a) is true iff the JOHN object stands in the CRITICIZE relation to himself.

In (23) the interpretation of *-nom* is trivial, the identity function. But in the compositional interpretation of (25) below, which represents (8b) in which the anaphor asymmetrically C-commands its antecedent, *-nom* is not interpreted as the identity map.

(25)



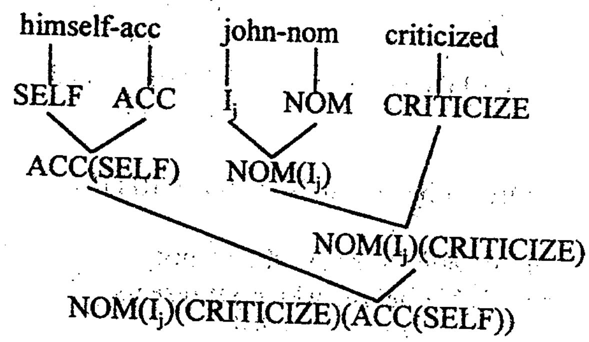
Here the KPn *john-nom* combines with the P2 *criticized* to form a P1a, the sort of P1 which seeks an accusative Kase Phrase (of which *himself-acc* is one), to form an S. We shall interpret P1a's as functions taking possible KPa denotations as arguments and yielding S denotations (T,F) as values. In (25) the P1a *john-nom criticized* is syntactically complex (There are no lexical P1a's in LK), so we expect its denotation to be a function of those of its parts: *john-nom* of category KPn, and *criticized* of category P2. P2's are already given as denoting binary relations. And KPn's have been interpreted as functions whose domain includes sets, which they map to truth values. But we shall now enlarge their domain to include binary relations, and in distinction to KPa's they do not map binary relations to sets, rather they map them to possible P1a denotations, as discussed above. And it is the interpretation of *-nom* which tells us just which P1a denotation we obtain:

(26) **Def** For G any map from P(E), the set of subsets of E, into {T,F}), and H any P1a denotation (any map from binary relations to sets), and R any binary relation,

NOM(G)(R) maps H to the truth value G(H(R))

The explicit compositional interpretation of (25) is given below:

(27)



= NOM(I_j)(CRITICIZE)(SELF)	Def ACC
= I_j(SELF(CRITICIZE))	Def NOM (26)
= $I_j(\{b \in E \mid b\text{CRITICIZE}b\})$	Def SELF
= T iff $j \in \{b \in E \mid b\text{CRITICIZE}b\}$	Def I_j
= T iff jCRITICIZEj	Set Theory; $j \in E$

So (25) and (19) are logically equivalent. But they do not differ in syntactic complexity and neither is derived from the other, nor is the interpretation of either stipulated as identical to that of the other. Rather in each case the expression is interpreted as a function of the interpretation of its constituents and it is simply a theorem that the truth conditions are the same, just as the logical equivalence of "not both A and B" and "either not A or not B" follows from the meanings of their parts and how they are composed.

Exercise Following the format above, show that the two Ss below are logically equivalent:

i . (john-nom bill-acc praised, S)   ii . (bill-acc john-nom praised, S)

To conclude our introduction to Little Korean we note informally the interpretation of boolean compounds. Ss and P1n's have basically been covered. Equally since P2s are interpreted as sets their conjunction, disjunction, and negation are interpreted by the intersection, union, and complement of the denotations of the expressions conjoined, disjoined, and negated respectively. Boolean compounds of KPn's (and KPa's) are interpreted pointwise: e.g. if X and Y are KPn's (KPa's) interpreted by F and G respectively then their conjunction is interpreted as that function $F \wedge G$ which maps each argument A to $F(A) \wedge G(A)$, etc. The deeper generalization is that conjunctions of expressions in a given category are interpreted as the *greatest lower bound* of the interpretations of the conjuncts, disjunctions as the *least upper bounds* of the disjunctions, and negations by complements, but these considerations go beyond what we need for Little Korean. Boolean compounds were primarily introduced here to provide for infinitely many expressions in many categories.

Observe that (25) and (19), our representations for (8b) and (8a) respectively, have the same degree of hierarchical, that is, constituency, structure. The *branching* structures for these two expressions are identical, but the two expressions themselves are not isomorphic in LK. That is because not all information relevant to ascertaining the *structure* of an expression is represented in a derivation tree.

For example, (25) tells us that some generating function has combined the NP *john* and the K *-nom* to yield the KPn *john-nom*. *john* here can be grammatically replaced by certain other NPs, like *bill*, but not by all, in particular not by *himself* preserving grammaticality. This derives from the definition of the domain of Case-Mark which does not include the pair <himself,NP> and <-nom,K>, information not represented in (25) which ultimately accounts for the fact that the P1a *john-nom criticized* in (25) and the P1n *himself-acc criticized* in (19) are not isomorphic expressions in LK. Support for this claim must await the formal, and grammar independent, definition of the relation *is syntactically isomorphic to*.

3. Syntactic Invariants of Little Korean

Here we provide a syntax independent definition of what is to count as a *structural property* of expressions, and more generally a *structural relation* between expressions. We express this in the more neutral terminology of *grammatical invariant*, and we will see that properties (like *is a S*), relations (like *C-commands*) and even specific expressions (like <-*acc*,K> in LK) are grammatical invariants in exactly the same sense.

Given an arbitrary grammar G, we get at the notion of the *structure* of an expression s by considering ways s can be changed without changing structure. We represent such *structure preserving* changes by functions mapping expressions to expressions. These functions are defined below and are called *syntactic automorphisms*. The idea is that if such a function h maps an expression s to an expression t, then t and s have the *same structure*, that is, they are *syntactically isomorphic*.

Then a property P of expressions will then be said to be a *structural* property, an *invariant*, iff whenever an expression s has P then every expression s is isomorphic to also has P. That is, s has P iff h(s) has P, all syntactic automorphisms h. Clearly if P were a property which held of some expression s but failed of some t isomorphic to s then whether an expression had P would not be predictable just from its structure, so P would fail to be a structural property in this case. Thus we think of structural properties as ones that cannot distinguish among expressions with the same structure. This comes down to saying that a property P is *invariant (structural)* iff for all syntactic automorphisms h, h(P) = P, where by h(P) is meant $\{h(x) \mid x \in P\}$.

For example, the property of being a masculine singular proper noun is likely to have a structural property of English. Plausibly (this is not an argument, just a thought experiment to help build our intuitive understanding of the notion "structural property") a function that mapped *John* to *Bill*, *Bill* to *Frank* and *Frank* to *John* and made no other changes except those induced by these (i.e. the function must map the complex expression *John laughed* to *Bill laughed*, etc.) is a syntactic automorphism of English. This would mean for example that no syntactic automorphism of English could map *John* to *Mary*, or to *sings*, or to *All cats are black*—which is reasonable. Rather h(*John*), the image of the proper noun *John* under an automorphism h, must always be a masculine singular proper noun.

Similarly a relation R between expressions is *invariant (structural)* iff whenever an expression s stands in the relation R to an expression t then for all syntactic isomorphisms h, h(s) stands in the relation R to h(t). And as with properties, this is equivalent to saying that R is *invariant (structural)* iff h(R) = R, all automorphisms h, where $h(R) =_{df} \{\langle h(s),h(t)\rangle \mid \langle s,t\rangle \in R\}$. And one proves that for any grammar G the relation *is a constituent of* (see below) is a structural relation on the expressions of L(G).

To make these intuitions concrete, we must define what is meant by a syntactic automorphism. The most crucial component of such a function is that, in an appropriate sense, it does not change the generating functions. They are what determines structure. That is, if a generating function F derives y from x then, for h any automorphism, F must derive h(y) from h(x), and conversely. A little more generally, if F is a two place function, like Case-Mark or PA in LK, and F derives y from $\langle x,x'\rangle$, that is, $F(x,x') = y$, then F derives h(y) from $\langle h(x),h(x')\rangle$ and conversely. That is, $F(x,x') = y$ iff $F(hx,hx') = h(y)$.

Stated generally these conditions say that $h(F) = F$. That is, thinking of F as a set of pairs $\langle x,y\rangle$, where x itself might be a sequence $\langle x_1,x_2,...\rangle$, by h(F) is meant the set of pairs $\langle h(x),h(y)\rangle$, where h(x) is $\langle h(x_1),h(x_2),..\rangle$ if x is the sequence $\langle x_1,x_2,...\rangle$. To say that $h(F) = F$ is to say that *h fixes F*, it does not change it; equivalently F is *invariant* under h. (Provably $h(F) = F$ iff $Dom(F) = Dom(h(F))$ and h commutes with F: that is, for all x in Dom(F), $F(h(x)) = h(F(x))$).

There is a final condition we require of automorphisms: they must be bijections. So they cannot map distinct expressions to the same one (= they are one to one, injective) and they cannot leave out any expressions, that is, they are *onto* (subjective): given an automorphism h any expression t has something mapped to it by h. (In fact onto-ness follows from fixing the generating functions in a wide variety of natural cases, namely ones in which there are no *isolated points*, that is, no expressions which fails to occur as a coordinate in the domain of some generating function).

The onto-ness condition is natural. Our intent is that given L(G) and an automorphism h, the set of images of expressions in L(G), namely $\{h(s)|s\in L(G)\}$, has all the structural properties of L(G). If we didn't require onto-ness of automorphisms of English it might happen that the set of images didn't include any reflexive pronouns, or infinitival *to*, or gerundive *-ing*, and thus, pretheoretically, would fail to have the same structure as the original language. Similarly requiring one-to-one-ness is reasonable. If we could map different expressions to the same one we would change structure in certain cases. For example if we map both *laughed* and *cried* to *laughed* then we would map *John both laughed and cried* to *John both laughed and laughed*; the former is grammatical, the latter not. Thus:

(28) **Def.** Given a grammar G,

a. a function h from L(G) to L(G) is a *syntactic automorphism* iff h is a bijection which fixes the generating functions (= $h(F) = F$, all generating functions F).

b. expressions s and t in L(G) are *isomorphic* iff there is an automorphism h such that $h(s) = t$ (in which case there is provably an isomorphism mapping t to s).

c. a linguistic object of any degree of set theoretic complexity is *invariant* (*structural*) if it is fixed, that is mapped to itself, by all the syntactic automorphisms.

The generating functions are by definition invariant. What is of interest is to see what other objects must be fixed if we fix the generating functions (= we do not change structure). For example in LK one proves that the property of being a KPn is invariant. That is, the set of expressions of category KPn is mapped to itself by all automorphisms of LK. Similarly in LK the ternary relation *s is a possible antecedent of t in u* (defined below) is invariant.

As a special case of (28c) we can ask of a given expression s whether all syntactic automorphisms map s to itself. If so then s itself is a syntactic invariant. One proves for example for LK that <-acc,K> is a syntactic invariant; so is <-nom,K>. That is, just as for properties and relations, a lexical item is structural iff it is mapped to itself by all the automorphisms. Using the term *morpheme* for an element of Lex, we have:

(29) For all grammars G, a morpheme s of L(G) is invariant (structural) iff h(s) = s, all syntactic automorphisms h of L(G).

(29) precisely presents the sense in which identity of morphemes can be structural. It also gives, to my knowledge, the only language independent characterization *grammatical morpheme* (*function word* in more traditional terminology). And we see that a statement like (18) describing the Anaphor-Antecedent relation in LK is at least a candidate for being a structural statement as it stands, even though it mentions particular morphemes. They are after all structural morphemes. And in fact (18), stated a little more formally, is a properly structural characterization of the local Anaphor-Antecedent relation in LK. To see this we first give some universal invariants and definitional procedures that lead from invariants to invariants.

3.1. Universal (=uniformly definable) Invariants

(30) Let G be an arbitrary grammar. Then,

1. ∅ and L(G) are invariant subsets of L(G). The latter just says that the property of being grammatical in G is an invariant property of expressions. (If this were not so we would just have the wrong definition of "structural invariant"). More generally, for each finite n, $L(G)^n$ is an invariant set of n-ary sequences of expressions of L(G).

2. Each generating function F is invariant (trivially); the domain and range of each generating such F is invariant, and more generally, the set of i^{th} coordinates of F is an invariant subset of L(G).

3. The collection of invariant subsets of $L(G)^n$ is closed under arbitrary intersections, unions, products and complements (relative to $L(G)^n$). This is

practical in that it guarantees if two properties P and Q are invariant so are those expressed by their conjunction, disjunction, negation, universal and existential quantification.

4. For each $s \in L(G)$, the set [s] of expressions isomorphic to s is an invariant subset of L(G). It is always non-empty (since s is in it) and no non-empty proper subset of [s] is invariant. That is, [s] is an atom in the complete atomic boolean lattice [L(G)] of invariant subsets of L(G).

5. The value of an invariant function at an invariant argument is itself invariant. (See Keenan (1996) for much more systematic discussion of invariants.

6. The relation *is a constituent of, CON*, is invariant, where CON $=_{df}$ the reflexive transitive closure of ICON, the *immediate constituent of* relation, defined by:

sICONt iff there is a rule F and expressions $d_1,\ldots,d_n$ such that $t = F(d_1,\ldots,d_n)$ and for one of the i's between 1 and n, $s = d_i$.

(The reflexive transitive closure of ICON is that relation R defined by: sRt iff either s = t or for some k>1 there is a sequence $\langle u_1,\ldots,u_k\rangle$ of expressions such that $s = u_1$, $t = u_k$ and for each i, $1 \leq i < k$, u_iICONu_{i+1}).

7. *C-commands* defined as follows is invariant: s *C-commands* t *in* u iff for some rule F and some tuple $\langle v_1,\ldots,v_n\rangle$ of expressions in Dom(F), $F(v_1,\ldots,v_n)$ is a constituent of u and for some i, $s = v_i$ and for some $j \neq i$, t is a constituent of v_j.

Some sets which one might have thought would be universally invariant are not. For example Lex_G fails to be invariant for certain G (but is invariant if no lexical item is the value of a generating function on expressions of L_G). Similarly the set of expressions of a given category C, $\{s \in L(G) \mid Cat(s) = C\}$, may fail to be invariant. There are Ls where all the phrases of one category can be mapped to ones of another category. E.g. BG presents a model of gender agreement in which the masculine Ns can be interchanged with the feminine Ns preserving structure (an unstable possibility that requires that many other conditions be met).

More locally, one might have thought that we could have isomorphically interchanged <*-nom*,K> and <*-acc*,K> in LK. They have the same category and seem to play comparable syntactic roles. But we cannot. One problem arises for example with Ss built from intransitive verbs, like *john-nom laughed*. If we traded in *-nom* for *-acc* and so *john-nom* for some KPa the result could not combine with a P1n like *laughed*. The only option would be to also trade in *laughed* for some P1a, but all of them are syntactically complex and it is easy to show that a simplex expression cannot be isomorphically mapped to a complex one in LK.

Equally in LK no automorphism maps one of (8a,b) to the other. The immediate reason is that we provably cannot isomorphically interchange P1n's with P1a's, and ultimately that derives from the fact that (*himself*,NP) has a restricted distribution. Returning now to the LK Anaphor-Antecedent relation in (18) we observe:

Fact The coargument relation in LK is invariant, where we define:

(31) s is a *coargument* of t in u iff for some v of category P2, either PA(s, PA(t,v)) or PA(t, PA(s,v)) are constituents of u.

Fact The property of being -(*l*)*ul* marked is structural, where we define:

(32) An expression s in LK is -(*l*)*ul marked* iff Cat(s) = KPa.

Note: -(*l*)*ul* marked expressions include ones like <*both himself-acc and john-acc*,KPa> whose string part is not a *constituent* suffixed with -(*l*)*ul*. From these two facts plus the closure of invariants under conjunction the invariance of the relation defined in (18) follows.

Thus we may give properly structural relations directly in terms of identity of morphemes (morphology) without coding these relation in terms of hierarchical structure. And we infer as a corollary that the (local) Anaphor-Antecedent relation in a language may be structurally characterizable even though there is no consistent C-command relation between an anaphor and its antecedent.

References

Chomsky, N. 1996. *The Minimalist Program*, second printing. The MIT Press.

Hornstein, N. 1995. *Logical Form: From GB to Minimalism*. Blackwell

Keenan, E.L. 1996. Logical Objects. ms. Dept. of Linguistics, UCLA

Keenan, E.L. 1988. On Semantics and the Binding Theory. In J. Hawkins *Explaining Language Universals*. Basil Blackwell. 105-144.

Keenan, E.L. and E.P. Stabler. 1997. Course Notes. European Summer School for Language, Logic and Information. Aix-en-Provence, August 1977.

Keenan, E.L. and E.P. Stabler. to appear. *Bare Grammar*. CSLI

O'Grady, W. 1985. An Argument for Discontinuous VPs in Korean. In *Language Research* 21.4 pp. 451-61.

Park, S.H. 1986. Parametrizing the Theory of Binding: the Implications of *caki* in Korean. In *Language Research* 22.2. 229-53.

Dept. of Linguistics, UCLA
405 Hilgard Ave
Los Angeles, CA 90025
USA
ekeenan@ucla.edu