Indices and the Theory of Grammar

Murat Kural
University of California, Irvine
mkural@uci.edu

George Tsoulas
University of York
gt3@york.ac.uk

November, 2004

1 Introduction

Indices have been used in the theory of grammar ever since Chomsky (1955) and throughout the history of generative grammar in order to mark and explain various types of grammatical dependencies. Perhaps the device of indexation is best known for being the centerpiece of the traditional binding theory, as well as control constructions involving PRO and null operators, e.g., in parasitic gaps, *tough*-construction, and so forth. Arguably, it is also a crucial component of combinatorial phenomena, such as thematic role distribution and, categorical and semantic selection. It has also been a suitable notation to mark scope domains and check the well-formedness of movement chains, as well as in the reconstruction of ellipsis.

However, in the last decade or so, the use of indices as a grammatical device have come to be regarded with great suspicion, and in fact, it has been to be among those elements that the theory of grammar should do without. This is most clearly expressed by Chomsky in a variety of places, so for example in (Chomsky, 1995, 381, footnote 7) he writes:

(1) . . . Thus, with sufficiently rich formal devices (say set-theory), counterparts to any object (nodes, bars, indices, etc.) can readily be constructed from features. There is no essential difference, then, between admitting new kinds of objects and allowing richer use of formal devices; we assume that these (basically equivalent) options are permitted only when forced by empirical properties of language.

Clearly, one cannot logically conclude that constructs are such as indices, bar levels and nodes are useless solely on the basis of the the conjecture that they are constructible out of features and/or set theory. There are two points to be considered here. The first one is the question of what constitutes the set of grammatical primitives, which is clearly an empirical matter. As such, one cannot make arbitrary decisions the basis of one's theoretical position. On the other hand, it is a commendable research strategy to the extent that it can be reduced to Occam's razor, though perhaps not always in an obvious way.

The second consideration, which is related to the first question, is the minimalist pursuit to see how far a theory can be pushed with a given set of minimal primitives. It centers around the equivalence between the richer use of existing formal devices and admitting new kinds of objects. However, equivalence may not be the entire story. It is perfectly possible, for instance, that the richer use of a given formal device introduces so much complication and inefficiency into the theory that it is not worth allowing, perhaps making it a more desirable course of action to simply *dis*allow the added functionality. By contrast, a new kind of object conceived specifically for a given purpose may turn out to be the more efficient option. Consider simply the fact that one may be faced with the option of using a rather ill-suited tool for a given purpose (because it is more primitive or it is useful in something unrelated) rather than a better suited – even one that is built for that purpose. Should the choice always be to *use the former* even if it seriously complicates the task and is inherently a lot less efficient? This strikes us as rather implausible. As for the idea that theory comparison should involve only the notion of simplicity measured in number of primitives without any consideration of coverage and efficiency, it strikes us as entirely unproductive.

Returning to our first remark now, it would seem that following through the logic of eliminating everything that may be constructed from something else, allegedly more primitive, would in the end lead to rather strange results. For example the concept of *Numeration* can be shown to be constructed out of primitives such as lexical items and features, yet it has not been eliminated from the theory since it is understood as a key tool in the reduction of the computational complexity of the theory. Therefore, there is justification to keep it as a distinct, though perhaps a derived concept, and the fact that its functions can be overtaken by making adjustments to how we access the lexicon during the derivation, among other things, does not appear to be an overriding reason to nullify its usefulness as a theoretical device.[1]

In later works, the stance expressed in (1) is hardened and indices are subjected to an outright ban, witness the following from (Chomsky, 2000, pp. 114)

(2) . . . It [The *Inclusiveness Condition*] also rules out introduction of traces, indices, λ-operators, and other new elements in the course of the derivation of C_{HL}. Recourse to such devices could be innocuous (e.g., if used for convenience to annotate properties that can be determined by inspection at LF) but questions arise if they enter into interpretation and function significantly within the computation — for example, percolation of indices, or operations that apply specifically to trace.

[1]The notion of computational efficiency cannot be dismissed as merely the province of a model of performance rather than competence since that would lead to concepts such as the Numeration, local economy, and so on also being rendered irrelevant. Needless to say, we assume that considerations of computational efficiency are integral to the theory of competence.

And in (Chomsky, 2001, p. 3)

(3) On such grounds we try to eliminate levels apart from the interface levels, and to maintain a bare phrase structure theory and the *inclusiveness condition*, which bars introduction of new elements (features) in the course of computation: indices traces, syntactic categories or bar levels, and so on.

(Epstein and Seely, 2002, p. 9) also brandish the inclusiveness condition to reject the existence of indices:

(4) Furthermore, under the inclusiveness condition, nonexplanatory representational encodings like subscripts, superscripts, bar levels, and star markings are no longer an option, but are instead explicitly prohibited.

And later in the same volume (McCloskey, 2002, 217) states:

(5) One can maintain that what the fronting rule targets is an indexed pronoun [. . .], but ceding to movement operations the power to detect and respond to such indices seems like a very dubious move (and one that is in violation of the principle of "inclusiveness" advocated in Chomsky 1995).

The reason why Epstein and Seely believe in the nonexplanatoriness of those *encodings* is, however, unclear. Aside from dogma no other reasons are given.

From the above, it seems to us that there are two issues at play. The first is whether the inclusiveness condition *does* prohibit the use of indices as Epstein and Seely (2002) seem to suggest.[2] The second issue is whether something akin to an index can be constructed out of features, the assumption here being that one would want to do something like this given that indices provide us with the means to capture certain very specific and independent relations. We will examine the question of inclusiveness here and leave the second one to the following sections. The inclusiveness condition (IC) is formulated by (Chomsky, 2000, 113) as follows:

(6) The *Inclusiveness Condition*: No new features are introduced by C_{HL}

The effects of this condition are further expounded in the remarks in (2), (3). Prohibiting C_{HL} from introducing new features is a necessary constraint in order to ensure that C_{HL} does not have access to the set of features $\mathcal{F}$ available to language $\mathcal{L}$ in the course of a derivation and alter the nature of lexical items (LIs). However, it is not clear that this limitation needs to be extended to syntactic elements other than features. In addition to indices, the other items that would seem to be included in the set of such prohibited items would be λ-

[2]Note here that the question raised by McCloskey (2002) is rather different. It is a different thing to wonder about whether a particular construction is more profitably analysed by making reference to indices or not. Here we are concerned with the question of principle.

operators and traces. One needs to consider the nature of these elements before deciding whether they do really fall within the purview of the IC.

The situation is resolved easily with respect to traces, by eliminating them altogether and adopting the copy theory version of movement instead.[3] λ-operators do not pose much problem either since they are not normally used in the syntax as such. They appear primarily in the interpretation of a variety of elements such as wh-pronouns and other relativisers in relative clauses. It would probably be rather unnatural to introduce a λ-operator directly in the derivation without it being provided by some lexical item, so it is very likely that they would not pose any significant problem.

Next, consider indices. If they fall within the same class as syntactic class as features, the IC would have the following correlate with respect to indices.

(7) No new indices are introduced by C_{HL}

In other words, there is nothing wrong really with indices as basic devices so long as they are not introduced in the course of the derivation — as opposed to, say, the operation that selects items from the lexicon and introduces them to the *Lexical Array* (LA) or *Numeration*, or the one that selects items from LA and introduces them to the derivation, where both operations would clearly have to be incorporated into an explicit system. A common line of thinking that counters this approach is one that is rather odd in nature: indices cannot be legitimate because C_{HL} can only manipulates features, and indices are not features. If, on the other hand, it is possible as Chomsky suggests in (1), to construct indices out of features or as features, then all will be well. The fundamental claim in (1) is that features and a sufficiently powerful formal apparatus should be enough to construct any device which may be of some use. The examples cited there are bars, nodes, and indices. Admittedly, the bare phrase structure model, (Chomsky, 1994) and subsequent work, has effectively eliminated labels, nodes, bars and so on[4] using features and set-theoretic notions and tools. Still though, indices remain, and as far as we know, there has been no serious effort to provide a reconstruction of the notion of an index in terms of features and set theory. It is precisely this *construction of indices out of features* that we show is neither feasible nor desirable.

We believe that it is possible to make two major arguments against indices, and that the one sketched out on the basis of the IC is just one of them. The second is the claim that indices are just a notational aid which should be replaced as soon as a more basic instrument is found. Clearly, there is enough justification to eliminate any given notation, operation, or taxonomy when there is sufficient overlap or redundancy in that aspect of the grammar. However, this would not be true if removing the component would inevitably lead to further complication elsewhere in the grammar. We will argue this to be the case with shifting the

[3] As Chomsky(2000, 2002) points out the copy theory of movement is not a departure from minimalist assumptions, rather it is the null hypothesis.

[4] For different views on how the elimination of these objects can be achieved, see Adger and Tsoulas (1999), Collins (2002).

work load from indices and indexation to features and feature checking. We hope to show that indices and features are ideally suited for the tasks they are designed for, and that because the fundamental differences in the way they operate, and the range of facts that hold in and across languages, they stand in a zero-sum relationship such that whatever is gained by eliminating either one would be lost in the complexity introduced into the other.

In the following sections we will first try to show that it is not possible to think of indices as features in the way we commonly understand the term. Then, we will turn to the question of whether indices should be allowed in the grammar in any way and what its consequences would be for the theory of grammar and particularly with respect to the IC. What lies at the heart of our claim is the notion that there is no homomorphism between indices and features. We will present three types of arguments to support this position, based on (a) their mathematical and logical properties in section 2, (b) the role they play in the grammar and the range of tasks they are intended for, in section 3, and (c) the types of constructions involving indexation that cannot be handled by features, without introducing complications.

2 Formal Differences

In order to identify why there is no homomorphism between features and indices, we need to look at their differences, and we start doing this by comparing their intrinsic properties. The purpose in this section is to show that that features and indices have very different characteristics as formal devices.

A closer examination of the range of features often used by formal linguists reveals that features themselves do not constitute a homogenous group. We can identify at least three types of formal features:

Lexico-semantic features (L/S features) such as [± animate] or [± abstract], which are primarily ontological statements.

Grammatical features (G features) like [± Wh] or [± Modal], which are grammar-internally taxonomic.

Morpho-syntactic features (M/S features) as in [± accusative] or [± past], which map a given item to a grammatical function or a structural position.

L/S and G features are in taxonomic and intersective by their nature, and M/S features are essentially additive. We consider each type feature individually before we turn to indices.

2.1 Lexico-semantic features

L/S features encode the taxonomic information regarding what type of an object a lexical item denotes, and how that object ontologically relates to other types of objects in the universe of discourse. They are grammatical representations of the speakers' knowledge about the real world. As such, shifting the L/S feature of the word *dolphin* from [+ animate, + fish] to [+ animate, – fish] would indicate only a change in the speaker's belief about dolphins and bears no consequences at all on any aspect of the grammar itself. Mismatching L/S features leads to a semantic anomaly at worst, where the resulting sentence requires a specialized context to be interpreted, but it is not syntactically ungrammatical:

(8) Bill ate sincerity.

(9) Mary drank the paper.

Violations of this type are usually of the "strange choice of words" variety, which is also the case when non-referential expressions *who* and *someone* are used in a context that is selected for a non-human entity.

(10) Who did you build?

(11) Bill built someone.

These sentences are perfectly interpretable under the right circumstances, e.g., if the "person" in question is in fact an android, a replica, or a clone.[5] In cases where *who*/*someone* and the verb *build* present a mismatch are in the same league as the use of the word *desk* to mean "chicken": a mistake in the choice of lexical items that leads to an unintelligible sentence, but not to structural incongruity.[6]

Because they are ultimately ontological statements, L/S features can be thought as instructions to find the location of a corresponding entity in the ontological tree. For instance, the object that corresponds to a [+ animate, + fish] noun, such as *goldfish*, must be located at a node that is dominated by the nodes for ANIMATE and FISH. Further, trees being what they are, L/S features can also be expressed in terms of set membership. The object corresponding to a [+ animate, + fish] word, such as *goldfish* must be in the intersection of the set of all animate things and the set of all fish things. Conversely, to verify that *dolphin* is [– fish], we need to look at the complement of the set of all fish, or make sure that the corresponding object is not dominated by FISH.

(12) X is [+ animate, + fish] iff
$(X \in \text{Animate}) \wedge (X \in \text{Fish}) \rightarrow X \in (\text{Animate} \cap \text{Fish})$

As noted, L/S features are intersective and taxonomic, and they establish an "is-a" relationship, i.e., a goldfish is an animate fish.

2.2 Grammatical features

G features are also taxonomic in their nature, which is to say, identifying *who* as [+ Wh, + N] is also to state that it lies in the intersection of all *wh*-expressions

[5]Witness, for instance, the subtitle of John L. Pollock's book *Cognitive Carpentry: A blueprint on how to build a person* Cambridge, Mass. MIT Press, 1995.

[6]This does not necessarily mean that syntax is completely blind to features of this type. they clearly play some role with respect constructions like dative shift, and so on. However, a mismatch can be repaired easily by making some minor adjustments in one's presuppositions.

that are nouns. Its L/S feature [+ human] indicates that *who* can be found by further intersecting the [+ Wh, + N] set with the set of words in the lexicon that refer to human entities.[7] In this sense, G features share with L/S features the property of indicating an "is-a" relationship.

What separates G features from L/S features is the fact that they constitute a relatively small set with closed membership. Since L/S features represent the ontology of the real world or some cognitive model thereof,[8] they are subject to all the same revisions, readjustments, and additions as our understanding of the universe is. G features, however, do not operate the same way. Once their grammar has matured, speakers no longer add new grammatical features, nor would they be able to globally reorganize G features the same way they would be able to do with L/S features, given the right circumstances. Grammatical categories are naturally resistant to historical change, and when a change occurs, it has an identifiable impact on the grammar. Changing all [– V] features to [+ V] or purging [+ count] entirely from the lexicon would have far-reaching consequences in the language. By contrast, if we somehow found out that what we thought were rabbits turn out to be reptiles, no grammar would miss a beat.

G features are often checked in the syntax, or they are regulated in other ways, such as subcategorization. Mismatch violations, where for instance, a noun is inserted into the structure instead of a verb, a wh-expression is left in situ in a wh-moving language, or one of the ϕ-features (person, number or gender) does not agree between a specifier and its head, the result would be syntactic ungrammaticality.

2.3 Morpho-Syntactic features

M/S features are like G features to the extent that deal with grammatical properties that play a role in the syntax. Unlike L/S and G features, M/S features are neither intersective nor taxonomic. They are additive. They combine with bare forms which lack the feature altogether to create more complex forms which are often, though not always, expressed by additional morphology. For example, [+ past] relates the root form *break* to the past form *break*-PAST. It does not, however, dictate the exact form the [+ past] of *break* would take. Whether it will be the regular form *breaked* or the irregular *broke*, and under what circumstances the choice will tilt towards either one is determined by factors outside the [± past] feature. What [+ past] does in this context is to simply map *break* to *break*-PAST, while the exact morphophonemics of *broke* versus *breaked* are decided according to the applicable spell-out rules of the grammar.[9] In this regard, it is important to separate the feature from its morphophonemics.

Note also that unlike L/S and G features, a [– M/S feature] is not charac-

[7]Although it is hard to talk about a purely ontological classification for a quantificational expression like *who*, when the sentence is being interpreted, its [+ human] feature forces the value assigned to *who* in the model to be chosen out of a set of candidates that are also in the intersection with human entities.

[8]The distinction is immaterial for the purposes of the present discussion.

[9]The same is true for [– past], which maps *break* to *break*-PRES.

terized by the absence of a [+ M/S feature]. As is the case with [– past] and [– plural], they actually indicate the presence of another [+ M/S feature], as in [+ pres] and [+ sing]. In fact, it may be more accurate to do away with the traditional [±] notation altogether, and use [past] and [pres] instead, a practice we will adopt in this paper.

The mapping that takes place with M/S features is between the base word W and the corresponding word W', which is a complex word that combines W with the appropriate morphology. This mapping is a bijection, i.e., it is a one-to-one and onto function.

a. It is a function: for every W there is one and only one W', i.e., for any verb like *keep* there can be only one *keep*-PAST, whatever actual form(s) are produced by the morphophonemic component.[10]

b. It is one-to-one: for any distinct pair of words W_1 and W_2, their corresponding morphologically complex forms W'_1 and W'_2 will also be distinct. For example, given verbs *jump* and *walk*, which are distinct, [past] would map them to *jump*-PAST and *walk*-PAST, which are also distinct.

c. It is onto: for any morphologically complex form W', there exists a simplex form W that maps onto it. In other words, for any word *keep*-PAST, there is necessarily a root, a *keep* that [past] maps onto *keep*-PAST.[11]

It is not clear how this property can be extended to L/S or G features since the only mapping done by those features is from a word to an entire set, e.g., the feature [+ fish] can be said to map the word *goldfish* to the set of all fish things, whereas [+ count] perhaps maps *pencil* to a set of all count nouns. So even under this interpretation M/S features would remain a different type of function.

In contrast with the "is-a" type relationship expressed by the L/S and G features, M/S features indicate a"has-a" type relationship, since its output form is a complex word that also contains the relevant morpheme.

2.4 Indices

Where L/S and G features are set membership statements regarding taxonomy and M/S features are additive operations that form a complex word, indices constitute a set of pointers that have no content or inherent value. They have no interpretation in isolation. All they really do is to group items together: if two items bear the same index, they belong in some ad hoc group, as members of the same arbitrarily selected set, and they share some key property. If an index is used to designate an identity in the interpretive model, all items that bear the same index will be understood to have the same identity designation,

[10] Although morphophonemics can produce competing forms based on the same input, e.g., *hung* and *hanged*, what matters in such cases is that the input form is unique, as in *hang*-PAST.

[11] With inherently plural words like "police" or "series", it can be argued that their roots are mapped onto an empty word, NULL, by [sing].

whether it comes about as a result of accidental coreference or binding. Likewise when the index is used to bind an argument to a slot in the θ-grid of a predicate, the argument is interpreted as having the same θ-role as indicated in the θ-slot.

An index can be loosely defined as a notational device that stands in for a set of arbitrarily collected elements, where the value of the index itself corresponds to the unique label assigned to that set. Members of the same set bear the same index, and conversely, coindexed items are members of the same set.[12] The act of indexing an item is one of associating the label of the set in question with the item itself.

(13) X_1 iff $\mathrm{X} \in \mathrm{Set}_1$ (First approximation)

The definition in (13) is workable to the extent that it is understood as a mathematical construct. It is, however, an odd statement to make from the point of view of human language as a computational system. It places no limitations on what kinds of objects a grammar can manipulate since sets can be constructed of any kinds of objects real or unreal, physical or abstract. One way to constrain the types of objects grammar can manipulate would be to restrict these sets only to one particular type of object, i.e., the reference of these entities. To this end we define a *pointer* as a grammatical construct that has no inherent value other than its ability to point at the denotation of the phrase or head it is assigned to.

(14) $Ptr(\mathrm{X})$ is a grammatical device that points at the denotation of X.

We now collect these pointers in sets and restate (13) in the following fashion.

(15) X_1 iff $Ptr(\mathrm{Y}) \in \mathrm{Set}_1$ (Final definition)

Coindexation can be expressed in these terms as a case of having two or more pointers collected in the same set.

(16) X_1 and Y_1 iff $Ptr(\mathrm{X}) \in \mathrm{Set}_1 \wedge Ptr(\mathrm{X}) \in \mathrm{Set}_1$

The revised definition of indexation in (15) preserves the ability of an index to refer to heterogeneous collection of objects while maintaining homogenous and well-defined sets. Any regulation of what kinds of objects can receive an index in the grammar can now be worked in to the definition of pointers in (14).

Given the definitions above, the interpretive component can now operate on the set as a whole and use its arbitrarily selected label as the value of the index being evaluated. An index would be associated with a particular identity by including in the set, the pointer that references the individual targeted in the model. On the other hand, if the index is being used to associate a θ-role with an argument, a pointer that references the relevant slot in the θ-grid would also

[12] We choose to label the sets using numerals, i.e., 1,2, 3, etc., rather then the conventional letter sequence, i.e., i, j, k, etc. or unconventional means, e.g., blue, red, yellow, etc., for expository reasons. As far as we know, there is no principled reason why labeling should follow a particular convention or other.

be included in the set.[13]

Even though they can also be construed as set membership statements, L/S and G features differ from indices in terms of how their membership is defined.

(17) a. $\forall$x, x $\in$ Animate iff animate(x) = true
b. $\forall$x, x $\in$ Noun iff noun(x) = true

To determine whether some new entity is to be classified as a member of a set designated by an L/S or G feature, all we need to do is to verify that it satisfies some preestablished criteria. With L/S features this is typically a matter of identifying what kind of object the denotation of the word is. A word is [+ animate] if its denotation is in the set Animate, and that is decided by looking at whether it is an animate entity, whichever way it is construed. With G features, it is typically a matter of observing the behavior of the word. A word is [+ N] if it belongs in the set Noun, which is decided by observing whether the word behaves like a noun.

There is no analogous diagnostic for indices. We cannot decide whether a nominal expression has the index 1 as opposed to 2 by observing whether its denotation has 1-like properties as opposed to 2-like properties. In fact, the pointers that are collected in a set may be pointing at objects that do not have any properties in common: a slot in the θ-grid of a lexical entry, an individual in a model, a noun phrase in a sentence, and so on.[14]

What separates indices from M/S features is the fact that whereas M/S features are functions, and more specifically, bijections, indices are not. An index does not map an item to another. In fact, given the view presented here, an index is simply an indication that a pointer pointing at it belongs in some set.

Coindexation connects two items, but it is not a function since an item can in principle be coindexed with any number of other items. There is no fixed number as to how many items can bear the same index, i.e., how any members a set Set_1 can have, nor a way to predict the number in any given situation, which is not what one expects from a function. It is, however, an equivalence relation since it is reflexive, symmetric, and transitive: X bears the same index as X (reflexive), if X bears the same index as Y, then Y bears the same index as X (symmetric), and if bears the same index as Y and Y, as Z, then X bears the same index as Z (transitive).

The operation of index assignment, on the other hand, is a function as it maps an item X to its index 1. It is an onto function: for every index n that is assigned there would be an item X mapped onto it. However, it is not a one-to-one function: for any two items X and Y, where X $\neq$ Y, it is not the case that the index they receive must be distinct. Otherwise there would have

[13]Note that the view presented here remains non-committal as to whether the set must be ordered or unordered, or how big the set can be. It is possible that some sets are ordered and/or have a limitation as to how many members it can have. Thematic role association may just be such a case.

[14]That is, nothing in common other than the fact that they are all pointed at by pointers in the same arbitrarily selected set

been no such thing as coindexation. Since it is not one-to-one, the operation of index assignment is not a bijection, where as M/S features are.

Since indices are intrinsically distinct from all three types of features discussed in this section, (L/S, G, and M/S), and since these differences are irreducible, we would have to conclude that there is no redundancy between these devices that can be exploited if indices were to be eliminated. As a result, recasting indices in terms of features would force us to create a whole new class of features with properties that are unlike any other.[15] Overall, nothing would have been gained with respect to complexity by eliminating indices if the same complexity must be reintroduced through some back channel.

In short, indexation is a specialized tool with its own unique properties that cannot be costlessly duplicated by features, which are designed to handle different types of tasks.

3 Mechanical differences

Next we shift our attention to another set of differences between indices and features, namely those that are based on the way in which both devices operate, and the types of tasks each one is designed to perform. The following will be discussed in this section:

- Indices do not have any intrinsic meaning, features do.
- Indices are not finite, features are.
- Indices function at a distance, features must be local.

Each claim above is discussed in some detail below.

3.1 Vacuousness of indices

It is in principle possible to assign any index to any constituent, since there is really no typology of indices that would make one index inherently different from another. For this reason, the indexation of a lexical item is ultimately an arbitrary process. By contrast, features are incorporated into the lexical representation of an item, and as such, their distribution is tightly regulated by taxonomic considerations and/or the internal composition of the lexical item. A word like *child* in *the child enjoyed his ice-cream* is [+ human] and cannot be switched to [– human]. It can however, receive the index 1 as opposed to 2, or vice versa, since unlike [± human], there is no specific meaning associated with the indices 1 or 2.

In other words, the grammar places no restrictions on what items can receive what index, whereas switching the value of a feature or removing it completely has an impact on the grammar or the grammaticality of a structure as the case

[15] Modifying an existing class of features and expand its duties to also cover for indices would merely be a sleight of hand since in reality, doing so would in fact be the same as positing a new class of features.

may be.[16] This is because features have an interpretive force associated with them in some component of grammar, from phonetics to semantics, whereas indices do not have any intrinsic interpretation. They are interpretable only relative to other indices that are assigned in a given structure.[17] In fact, the only thing we can discern from the index of an element X is whether it points at the same object as some other element Y, where X $\neq$ Y. For example, in *The child enjoyed his ice cream*, the interpretation of the index 1 assigned to *his* depends on what the index of *the child* is. If *the child* also has the index 1, that would produce the reading where the that the child is enjoying his own ice cream. Obviously, this reading remains the same if *his* is assigned the index 197 as long as *the child* is also assigned the index 197. The interpretation changes only when the index assigned to *his*, n, is not the same as the index assigned to *the child*, m, i.e., $n \neq m$

This idea is illustrated in the following examples, where regardless of what index is actually chosen for *his* in the following examples, the interpretation varies only on the basis of the index assigned to *the child*, whether they are contraindexed, as in (18), or coindexed, as in (19) below:

(18) a. The child_1 enjoyed his_2 ice cream.
b. The child_2 enjoyed his_1 ice cream.
c. The child_3 enjoyed his_1 ice cream.

(19) a. The child_1 enjoyed his_1 ice cream.
b. The child_2 enjoyed his_2 ice cream.
c. The child_3 enjoyed his_3 ice cream.

Features are very different in this regard. The interpretation of *his* as [+ animate] or [+ genitive] does not depend on whether *the child* is also [+ animate] and the fact that it is [+ nominative] has no bearing on what type of morphology *his* will display. That is, unlike indices, features do have intrinsic meaning.

3.2 Finiteness of features

Since they form the internal make-up of lexical items, features are necessarily finite. This point can be made trivially with G and M/S features, which are, by definition, coextensive with the number of grammatically meaningful categories. Since grammar is a closed system,[18] the number of G and M/S features are necessarily finite.

This is perhaps less obvious with L/S features since they are connected in some way to ontology. It should be noted, however, that the purpose of the L/S

[16] Clearly, binding considerations may dictate whether a given index leads to ungrammaticality, but it is not the index itself that causes the problem in those cases, it is the structural relation between the item in question and its antecedent.

[17] This is contrary to the view presented in Fiengo and May (1994), where indices are argued to be directly correlating with the identity of element that has received the index. Our account differs from theirs in that we define an index as the label of an unordered set whose membership is defined by listing.

[18] Any grammar that does not have a finite number of categories would not be learnable.

features is not to define the known world, which is the job of ontology, but to define and organize the lexicon. As such, even though the lexicon itself is open ended and has the ability to add new words, it is not unbounded, since it is by definition a finite repository. Further, it is not the case that every addition to the lexicon would necessitate a new feature. This is because L/S features, like G and M/S features, indicate types rather than individuals. For example, each new proper name we add into our lexicon does not introduce a new L/S feature.

Indices, on the other hand, are not finite. The grammar can allocate a new index for any new referent in the model, and because adding another index does not alter the organization of the grammar in any way, there is no upper bound as to how many entities can be referred to. Since they are unbounded, indices are perfectly suited as the notational device that designates the individual instantiations of the types that are encoded by the features.

The net result of this distinction is that features and indices may need to be activated at different components of the grammar and through different types of mechanisms. Since features are finite, they can be handled in the lexicon by incorporating a predetermined pool of possible features to draw from. Whether or not there is parametrization of grammars in terms of what features are available in a given language, the fact that they are finite suggests that they can be assigned and/or manipulated in the lexicon.[19] By contrast, since indices are not finite, they cannot be stored in the lexicon. Their assignment must be handled through a generative mechanism post-lexically, which would ensure that they can in principle be produced without any limitations.

3.3 Locality of features

Features function from a very short range and they typically require a specific type of structural relation to be licensed. In the case of M/S features, this is often done by having the feature and its licensor enter into a specifier-head relationship.[20] and G features are often involved in selection or subcategorization, which means they arguably require the feature to be in a head-complement relation with its licensor, or perhaps be contained within a complement that can inherit the G feature and pass it up the tree. In either case though, the relationship is defined in fairly strict locality terms.

This is not the case with indices. In fact, indices can function from an arbitrary distance, without there being any strict structural relationship between coindexed elements.[21] Two elements can be coindexed across clausal boundaries and across islands which would otherwise be blocking movement or some types of referential dependency. In fact, coindexation is even possible between two elements that are located deep down in two separate branches in a tree:

[19] This does not preclude the possibility that they are checked in the syntax.

[20] Note that we are framing the discussion in terms of spec -head agreement, clause boundaries as locality domains and so on, rather than the probe-goal system, the operation *Agree* and the theory of phases. This is completely inconsequential for the points made here.

[21] It is important to bear in mind that this is true for coindexation which is not necessarily the same thing as binding, which traditionally requires an additional c-command type relation between the binder and the bindee.

(20) The man who talked to me_1 knew what I_1 was going to do.

Without getting into specifics on how the reference of indexical pronouns are to be understood, note that both *me* and *I* are indexed to the same individual. Crucially, coindexation succeeds in cases where the two elements in question do not stand in any discernible structural relation.

In fact, some coindexed elements do not even occur in the same structure, which is what happens at the interpretive stage, where the nominal expressions in the syntax are mapped on to particular individuals. This can be seen in (21) below, where *a student* is understood to be referring to an individual named John, leads to the heterogeneous set in (21-b), where "John" stands for the individual named John.

(21) a. $[\text{A student}]_1$ finished $[\text{his}_1 \text{ assignment}]_2$.
b. $\text{Set}_1 = \{Ptr(\text{a student}),\ Ptr(\text{his}),\ Ptr(\text{“John”})\}$.

For the index to be able to encode reference to an individual in the model, its corresponding set must also contain a pointer that points at an individual, which is clearly a non-syntactic entity — arguably, this is what enables syntactic expressions to refer. Given the view that an index corresponds to arbitrarily constructed sets of pointers, it is not surprising that the range of elements that can be indexed is quite diverse. As long as grammar has a way of interpreting it, any pointer can be included in the set.[22]

The conclusion reached in the preceding section that features and indices are distinct tools designed to handle different types of tasks is further supported by their mechanical differences. Due to the fundamental nature of their differences, if indices were to be eliminated as a possible grammatical device, to be replaced by some type of feature, we would be forced to alter the basic properties of what makes something a feature and how features operate in the grammar, neither of which is a a trivial matter.

4 Features and contraindexation

As observed above, one of the main differences between features and indices is that while features must be checked locally, indices can be verified from a distance. The exact range of the distance is determined by the type of expression that is being coindexed and/or bound: while anaphors demand a local binder, pronouns specifically exclude local binders, and referential expressions exclude binders altogether.

The locality restriction on anaphor binding makes it amenable to a movement-based analysis, which was pursued for reflexives by Lebeaux (1983) and for reciprocals by Heim *et al.* (1991), both of which turn on the notion that an anaphor must be moved to the close proximity of its binder. Such an approach

[22]Thematic role association presents another instance of indexation introducing heterogeneous sets, which will be discussed in section 5.

lends itself well to a feature-driven account of anaphor binding.[23] However, the situation becomes more complicated in the cases of referential expressions and especially pronouns. Whereas referential expressions require simple contraindexation with all c-commanding arguments, pronouns require contraindexation with c-commanding arguments only inside a local domain, but have no restrictions or requirements with arguments that are outside this local domain.

It is a relatively straightforward affair to express anaphor binding in terms of feature checking, especially because of its local nature. What is not clear, however, is to what extent a feature-based analysis can account for cases of quantifier binding and the anti-locality of pronouns. Quantifier binding occurs when a pronoun is bound by (coindexed with and c-commanded by) a quantifier, where the value assigned to the pronoun in the model is the same as the value assigned to the quantifier. Anti-locality refers to the fact that a pronoun cannot be in the same local domain as its antecedent or its binder. Crucially, the anti-locality requirement on pronouns still applies in cases of quantifier binding.

Suppose for the sake of discussion, that the anti-locality of pronoun binding is enforced through the same mechanism that the locality of an anaphor is: by moving the pronoun into the checking domain of its potential binder and look at whether it has moved across some locality boundary in doing so. In effect, this would allow cases of co- and contra-indexation to be handled through feature checking. There are two ways that coreference can be reduced to feature checking in the case of pronouns.

a. The first option is to assume that feature checking leads to coreference. In that case, it should not be possible to check the feature of a pronoun against its potential binder inside its local domain. It should, however, be possible to check it against the potential binders that lie outside its local domain in an unbounded way.

b. The second option is to assume that feature checking leads to noncoreference. This would allow the movement of the pronoun to be limited inside its local domain, which is a plus, but it also forces quantifier binding to be handled without any feature, which puts it at odds with the way anaphor binding is handled.

The problem posed by the behavior of pronouns is apparent. It forces us to either adopt a new set of rules and principles in regulating movement specifically in the case of pronouns, or new guidelines for interpreting the dynamics of feature checking. In this section we evaluate both options and demonstrate the types problems that would be created if one attempted to reduce all forms of binding to feature checking.

[23]It must be noted that whether or not anaphors are made legitimate through feature checking does not rule out the possibility that they are interpreted through their indices.

4.1 Coreference through feature checking

The first option considered above, where coreference is achieved through feature checking in the case of pronouns, puts pronouns on a par with anaphors,but it is clearly untenable since it creates a situation that is the opposite of anaphors. Pronouns would have two choices: they would either have to stay put and not move at all, or if they did, they would have to move minimally outside its local domain, and it would have to move unbounded distances. Consider the following example where indices are used for purposes of illustration.

(22) Everyone$_1$ thinks someone$_2$ knows him$_{1/*2/3}$.

In this view, the index "3" in (22) would presumably be obtained by the non-movement of the pronoun, since it is not coreferential with any other argument in the same sentence. The unacceptability of index "2" suggests a prohibition against checking the feature of the pronoun in its local domain. The index "1", on the other hand, indicates that the pronoun is capable of checking its feature against an argument that lies outside its local domain. In other words, as far as pronouns are concerned, feature checking would have to be optional (index "3"), and when it happens, it must take place outside of its local domain (indices "1" and "2").[24]

We dismiss this approach because it not only violates our basic understanding of how movement works, but also has the potential to create more problems than it solves.

4.2 Noncoreference through feature checking

The second option derives noncoreference from feature checking, rather than coreference. The reason this appears to be the more promising route is the fact that it does not force on us a redefinition of what constitutes legal movement. Instead, it shifts the revision to the area of how feature checking is to be interpreted with respect to pronouns. It seeks to dictate configurations of noncoreference, that is,the anti-locality of pronoun binding, and lets coreference remain an option in other configurations, where it would presumably be regulated by some other component of grammar. It will be argued below, however, that his approach also has its own problems.

There are number of ways in which noncoreference can be implemented in terms of feature checking. Features are either atomic entities, as in [accusative], or they are composed of two parts: a key, which is its label, e.g., *wh*, and its value, e.g., [±]. Feature checking involves a comparison of two features F_1 and F_2. There are, therefore, at most four possibilities when two features are being compared. Both the labels and the values may be the same, only the label may be the same, only the value may be the same, and neither the label nor the

[24]Note that pure coreference cases where the antecedent of the pronoun is referential would introduce another layer of problem. Since a pronoun can be coreferential with a non-c-commanding antecedent, this would force us to assume feature checking in configurations where movement has traditionally been ruled out.

value may be the same. Of these four possibilities only the first three seem to be worth pursuing.[25]

1. Both the pronoun and the antecedent are [+ feature], but checking the features successfully leads to degradation in the sentence.

2. The antecedent is [+ feature], the pronoun is [- feature], and a pronoun is required to have a [+] match against a [–] within its binding domain.

3. The antecedent has the feature [+ feature], and the pronoun has a [+ feature′], and noncoreference is enforced when the two incompatible features successfully check against one another.

We now consider each of these options separately.

4.2.1 Matching a [+ feature] with a [+ feature]

In this scheme, contraindexation effects would be obtained by prohibiting the [+] value of a (referential) feature of a pronoun matching another [+] value of the same feature within the binding domain.

There is a good reason to reject this approach: Clearly a pronoun needs to be noncoreferential not only with a single argument but with all c-commanding arguments in its local domain.

(23) John_1 introduced Bill_2 to $\text{him}_{*1/*2/3}$.

Assuming that *Bill* has a feature like [+ bill] and *John* has [+ john] to handle their referential content, the pronoun *him* would not only have to have [+ bill], but also [+ john] to make sure that it remains noncoreferential with both arguments. This means pronouns would have to have the set of features $\{[+\text{ feature}_1], [+\text{ feature}_2], \ldots, [+\text{ feature}_n]\}$, where n is the number of referents in the local domain of the pronoun, and the content of the features being determined by what other expressions are chosen in the numeration. Further, this approach also creates a fairly odd situation where a pronoun ends up having multiple referential features, suggesting that perhaps it is supposed to refer to multiple individuals simultaneously — which is probably not the ideal use of a referential feature, if we were to posit one.

4.2.2 Matching a [+ feature] against a [– feature]

In this scheme, contraindexation would be ensured by requiring that a pronoun match a [+ feature] with its [–] value within its binding domain. It is different from the previous strategy in that it does not create a situation where a pronoun must have multiple referents simultaneously. It also appears to be capturing the intuition behind co- and contra-indexation, where [+] and [+] a

[25] The fourth choice would be to make the antecedent [+ feature] and assign the pronoun the feature [– feature′], but this one seems to present nothing new that is not covered by the choices presented above.

referential feature would be used in cases of coreference, [+] and [–] would be for noncoreference.

The problem this approach creates is that it attempts to encode not only who or what the pronoun refers to, but also who or what it does not refer to. In other words, it ends up encoding the complement set of [+ feature] in the set of possible referents. Therefore if the pronoun is [+ feature_1], it would also have to have the features {[– feature_2], [– feature_3], ... [– feature_n]}, where n is the number of referents in the discourse domain. This is clearly a type of redundancy that cannot be independently justified.

This approach also suffers from a problem observed in the previous case: the feature content of the pronoun would have to be determined not only by looking at the individual that it refers to, but also by looking at the list of all other referents introduced by the expressions in its local domain, or perhaps the list of all referents in the discourse domain. Either way, this approach obviously introduces a level of complexity that is much harder to justify. It turns out that this particular problem is avoided by the third approach discussed below.

4.2.3 Mismatching a [+ feature] against a [+ feature′]

In this approach the pronoun and the noncoreferential argument in the same local domain that it is contraindexed with are assigned two distinct and incompatible features, as in a [+ feature] and a [+ feature′]. This is, in effect, the "[+ i] feature" approach that aims to recreate indices in terms of features. To the extent that it does not force multiple referential features being assigned to a given pronoun, this is clearly an improvement over the approaches discussed above. The drawback, however, is that in order for features to recreate indices, they also need to adopt their formal properties, which would establish a unique class of features whose characteristics are radically different from the inherent characteristics of features presented and discussed in previous sections.

The following are some of the properties that referential features would need to have to be formal equivalents of indices:

Missing upper bound Whereas the conventional features discussed above, i.e., L/S, G, and M/S, all constitute a finite set, referential features that we would posit must be of the type [+ feature_1], [+ feature_2], ... [+ feature_n], i.e., they must be open ended and have no upper bound if they are to have the capacity to refer to any and all possible referents in the universe. However, features with no upper bound cannot possibly exist in the lexicon. It would be neither learnable nor feasible for speakers to maintain. So the device that is designed to handle references must not be a static component of grammar. It must be dynamically generated, and this is certainly not a type of behavior one associates with features.

Real time generation Having instances of the referential device generated dynamically requires a dedicated function whether it is a feature or an index. It turns out that on this count, generating indices is a much simpler task since it involves no more than producing a unique label. Features, on the other hand, can be slightly more complex, especially if they come as a key/value pair, i.e., [± feature]. If they come without a [±] value, and just as [feature], and given the fact that they would have to be generated by separate a function, presumably postlexically, calling it a feature and not an index would be no more than an empty gesture without substance or a simple theoretic sleight of hand.

Type of function It should not come as a surprise that since they would have been designed to account for the same range of behaviors as indices, referential features would also be an onto but not a one-to-one function. That is, each expression would have some reference in the interpretation model, but not every expression would be mapped onto a unique referent, since more than one argument would have to have the same feature in cases of coreference. As noted earlier, this would be unlike any of the well-established conventional features L/S, G, or M/S.

Identity assignment Given that quantifiers do not have any inherent reference, the feature of a pronoun that is bound by a quantifier, or must be free cannot be determined before the value of the feature assigned to the quantifier is determined.

(24) a. Everyone$_1$ likes him$_{*1/2}$.
b. Everyone$_1$ thinks I$_2$ like him$_{1/3}$.

Since the value of the quantifier is assigned at the level of semantic interpretation, the feature to be assigned to the pronoun cannot be determined before the sentence is submitted to the interpretive component. That would make it exceedingly hard for the values of referential features to be determined in the syntax. Yet, syntactic configurations are known play key a role in determining the range of possible identities a nominal expression can be associated with.

One way to get around this problem is to continue to assign the referential feature in the syntax, but let its content remain undetermined until the sentence reaches the level of semantic interpretation. This would mean that referential features are inserted initially without content, They do not refer to "Bob" or "Sue" at this stage, but instead, behave like placeholders whose content is filled in later on at the interpretive component at the end of the derivation.

On the one hand, we observe that this is more or less how indices work, while features whether they are L/S, G, or M/S, always have some content that is licensed, selected, regulated, or manipulated. They are part of the lexicon, and they never wait until the derivation is over to receive their actual content. So referential indices we posited in this section as an alternative to indexation fall completely outside of the proper boundaries of features. On the other hand, a potentially more troublesome question would be how these features, which presumably lack any content in the syntax, can be checked against one another.

We conclude then that features of the type [+ feature$_1$], [+ feature$_2$], ... [+ feature$_n$] are simply notational equivalents of indices, and all they really do is to disguise the indices {1, 2, ..., n}, as {[+ 1], [+ 2], ... [+ n]}. As such, nothing is really gained by substituting indices with the kind of referential features sketched out above, i.e., index features, as the difference appears more cosmetic than substantive. By contrast, keeping indices as an independently valid notational device allows us to maintain a clear division of labor between the tasks performed by each device, and it also helps us avoid terminological and notational confusion in the future when time comes, as if periodically does, to reevaluate the generative toolbox.

5 Indices and interpretation

An advantage gained from the view that an index is a collection of pointers with no intrinsic lexical, syntactic, or semantic content is that it allows index assignment to serve a valuable function in being able to connect disparate elements that otherwise do not even exist in the same domain. This ability plays an especially vital role in the interpretive component where elements of form are connected to elements of meaning. By design, features do not have any such capability as they can only be defined for grammatical entities, and presumably those that are represented in the lexicon.

One obvious area where indices manage to bridge two very different areas in language is the mapping of an LF and/or semantic representation of a sentence R to its corresponding interpretation model M. This is a process in which anaphors, pronouns, names, and other referential expressions in R are mapped onto particular individuals in M to test for the truth conditions of the sentence relative to M. Since individuals in a model and phrases in a sentence are both entities that can be pointed at, they can be collected in a set, which in turn can be labeled with an index. As a result, the mechanism of indexation is capable of connecting such different entities from different domains in terms of identity. Clearly, this is not a type of task that can be handled through features.

Another area where indexation plays a crucial role in the interpretive component is the association of an argument with its thematic role in terms of establishing (a) which predicate a given phrase is an argument of, and (b) which thematic role the argument associates with.

Even though they interact with one another in interesting ways, case and θ-roles are substantially different types of grammatical entities. Case is an additive property of a phrase. It can be detected by simply evaluating the morphological content of an expression. The only condition placed on case is that it needs to be licensed at some point in the derivation regardless of whether or not the licensor is located in the same clause that the expression originates. The notion that an an argument may move outside its own clause and seek an appropriate case licensor up along the tree has been the cornerstone of the traditional analyses of he Raising-to-Object and Raising-to-Subject constructions. The only information the licensor needs to have access to in the licensing process is what case is being licensed.

In θ-role licensing, it is not enough for the argument to enter into the correct structural relationship with the predicate that provides the θ-role. It is also crucially important for the interpretive component which predicate has θ-licenses it. In other words, labeling an argument as an “Agent” is not sufficient. Since a θ-role is essentially a relation between an argument and its predicate, an Agent would be more properly defined as *Agent*(DP, V). As such, no argument can be interpreted solely as an Agent or a Patient, but as the Agent or Patient of a particular verb. Consider the following example.

(25) Sue played the piano and Sam peeled the potatoes.

If we constructed a set of all arguments that bear the Agent relation with some predicate, both *Sue* and *Sam* would be in that set, just as they would be in a set that contains all nominative expressions in (25). However, unless *Sue* is mapped onto the individual who bears the Agent relation to *played the piano* and *Sam* bears the Agent relation to *peeled the potatoes*, neither argument would be properly interpreted in the model. Indices are perfectly suited for the task of connecting the base position of an argument with its surface position, whereas features are not designed to encode information that needs to be accessed from long distance.[26]

The second way in which indices play into the thematic interpretation of arguments is in terms of the process of θ-role assignment. Given that features are not appropriate devices in the syntax to keep track of what predicate a given argument bears its θ-role in connection with, it also becomes questionable whether features ought to play any role in determining the θ-role of an argument when it combines with a predicate through the Merge operation. If features are not going to be a factor in identifying the predicate-argument structure in the syntax, then there is no real motivation to maintain that the operation of θ-role assignment is an instance of feature assignment. Instead, one can construct an indexation-based assignment process, where arguments associate with their θ-roles by sharing an index with the corresponding slot in the θ-grid of the predicate they receive their θ-role from.

(26) $$\text{Sue}_1 \begin{bmatrix} \text{play} \\ \dots \\ <\text{Agent}_1,\ \text{Patient}_2> \end{bmatrix} [\text{the piano}]_2$$

The Agent and Patient arguments are identified in (26) by coindexing them with the corresponding slots in the θ-grid of the verb, also see Stowell (1981).[27] These “thematic” indices are not any different or more complex than the other

[26] Positing features like [+ play-piano] may superficially solve the immediate problem, but it creates an even bigger problem. It forces us either to create enormous numbers of features to differentiate [+ play-piano] from [+ play-clarinet], or to compose these features in the syntax in a process that duplicates Merge.

[27] In this work, we are not concerned with the question of whether the θ-grid is represented in the matrix of the verb *play* as its imported into the numeration or the coindexation spans across different components of the grammar, i.e., between the syntax and the lexicon the same way that the interpretive indices span across the syntax and the interpretive model.

indices we had been discussing in the previous sections. They indicate that a set has been created, with pointers pointing at the relevant slots in the θ-grid of the predicate and the arguments constructed in the syntax.

(27) a. $\text{Set}_1 = \{Ptr(\text{λx play(x, Patient)}),\ Ptr(\text{Sue})\}$
b. $\text{Set}_2 = \{Ptr(\text{λy play(Agent, y)}),\ Ptr(\text{[the piano]})\}$

The set in (27-a) indicates that the argument *Sue* is coindexed with the θ-slot of *play* reserved for the <Agent> argument, whereas the set in (27-b) indicates that the argument *the piano* is coindexed with the θ-slot of *kick* that is reserved for the <Patient> argument.

6 Implications for the *Inclusiveness Condition*

In this work, we argued that recasting indices in terms of features is not merely a case of adopting a richer use of an already existing formal device, but rather, a case of readmitting a new kind of object albeit with an old label. As good as it may look on paper, in practice, indices simply cannot reduced to features. That is because the gulf between the properties of indices and features is so great that there is no reasonable way to have features mimic the behavior of indices and still retain their fundamental character as features.

There is, however, a more profound question that arises in the context of the view of indexation we have adopted here, and that is how it impacts the status of the Inclusiveness Condition. One may argue that the of indexation ultimately undermines the validity of the IC, since it introduces a new construct into the C_{HL} after the numeration.

We disagree with that position. We do not believe our arguments here are necessarily in opposition to the IC. What it leads us to reject is Chomsky's unqualified proposal that new objects and extended usage of existing formal devices are equivalent. We think this assertion can be retained if the qualification *all things being equal* is added.

Obviously, we would also like to reject Epstein and Seely's (2002) monolithic interpretation of the IC. Instead adopt a conception that is modeled after Partee's (1984) interpretation of Montague's (1970) stance on the principle of compositionality. She writes:

> ...Montague took compositionality as a guiding principle in determining what sorts of things meanings should be and not as an independently falsifiable claim.

Our understanding of the IC is very close. We consider it a guiding principle in determining the types of objects syntactic computation should be able to manipulate. Under a somewhat more narrow interpretation the IC prohibits the addition of any *new features* into the numeration, and as such, indices fall outside the purview of the IC. In fact, if taken by its face value, the IC does not make any claims about the *existence* of bar-levels, nodes, indices, and so forth, neither does it intend to establish what the primitives of the theory ought to be. As Janssen (1997) puts it in the context of compositionality: *[It] is not a formal restriction on what can be achieved but a methodology on how to proceed.* In this sense, the IC would function much better as a constraint on procedure.

References

Adger, D. and Tsoulas, G. (1999). The syntax and semantics of merge. In P. Sauzet, editor, *Proceedings of the Third Langues et Grammaire*, pages 5–20. Departement des Sciences du Language.

Chomsky, N. (1955). The logical structure of linguistic theory. Revised 1956 version published in part by Plenum Press, 1975; University of Chicago Press, 1985.

Chomsky, N. (1994). Bare phrase structure. In G. Webelhuth, editor, *Government and Binding Theory and the Minimalist Program*, pages 381–439. Blackwell, Oxford.

Chomsky, N. (1995). *The Minimalist Program*. Cambridge, Mass.: MIT Press.

Chomsky, N. (2000). Minimalist inquiries: The framework. In R. Martin, D. Michaels, and J. Uriagereka, editors, *Step by step: Essays on minimalist syntax in honor of Howard Lasnik*. MIT Press, Cambridge Mass.

Chomsky, N. (2001). Derivation by phase. In M. Kenstowitz, editor, *Ken Hale: A life in language*, pages 1–52. MIT Press, Cambridge MA.

Chomsky, N. (2002). Beyond explanatory adequacy. Ms. MIT.

Collins, C. (2002). Eliminating labels. In S. D. Epstein and T. D. Seely, editors, *Derivation and representation in the minimalist program*, pages 42–61. Blackwell Publishing, Oxford.

Epstein, S. D. and Seely, T. D. (2002). Introduction: On the quest for explanation. In S. D. Epstein and T. D. Seely, editors, *Derivation and representation in the minimalist program*, pages 1–18. Blackwell Publishing, Oxford.

Fiengo, R. and May, R. (1994). *Indices and Identity*. LI Monographs. MIT Press, Cambridge, Mass.

Heim, I., Lasnik, H., and May, R. (1991). Reciprocity and plurality. *Linguistic Inquiry*, **22**, 63–101.

Janssen, T. M. (1997). Compositionality. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 417–473. Elsevier, Amsterdam.

Lebeaux, D. (1983). A distibutional difference between reciprocals and reflexives. *Linguistic Inquiry*, **17**, 723–730.

McCloskey, J. (2002). Resumption, successive cyclicity, and the locality of operations. In S. D. Epstein and T. D. Seely, editors, *Derivation and representation in the minimalist program*, pages 184–218. Blackwell Publishing, Oxford.

Montague, R. (1970). Universal grammar. *Theoria*, **36**, 373–398. Reprinted in *Formal Philosophy*, by Richard Montague, Yale University Press, New Haven, CT, 1974, pp. 222–246.

Partee, B. (1984). Compositionality. In F. Landman and F. Veltman, editors, *Varieties of Formal Semantics*, pages 325–343. Foris, Dordrecht.

Stowell, T. (1981). *Origins of Phrase Structure*. Ph.D. thesis, MIT, Cambridge, MA.