Computational Analogues of Constraints on Grammars: A Model of Syntactic Acquisition

Robert Cregar Berwick
MIT Artificial Intelligence Laboratory, Cambridge, MA

1. Introduction: Constraints And Language Acquisition

A principal goal of modern linguistics is to account for the apparently rapid and uniform acquisition of syntactic knowledge, given the relatively impoverished input that evidently serves as the basis for the induction of that knowledge -- the so-called *projection problem*. At least since Chomsky, the usual response to the projection problem has been to characterize *knowledge of language* as a *grammar*, and then proceed by restricting so severely the class of grammars available for acquisition that the induction task is greatly simplified -- perhaps trivialized.

The work reported here describes an implemented LISP program that explicitly reproduces this methodological approach to acquisition -- but in a computational setting. It asks: what constraints on a computational system are required to ensure the acquisition of syntactic knowledge, given relatively plausible restrictions on input examples (only positive data of limited complexity). The linguistic approach requires as the output of acquisition a representation of adult knowledge in the form of a grammar. In this research, an existing parser for English, Marcus' PARSIFAL [1], acts as the grammar. PARSIFAL divides neatly into two parts: an interpreter and the grammar rules that the interpreter executes. The grammar rules unwind the mapping between a surface string and an annotated surface structure representation of that string. In part this unraveling is carried out under the control of a base phrase structure component; the base rules direct some grammar rules to build canonically-ordered structure, while other grammar rules are used to detect deviations from canonical order.

We mimic the acquisition process by fixing a stripped-down version of the PARSIFAL interpreter, thereby assuming an initial set of abilities (the basic PARSIFAL data structures, a lexicon, and a pair of context-free rule schemas). The simple pattern-action grammar rules and the details of the base phrase structure rules are acquired in a rule-by-rule fashion by attempting to parse grammatical sentences with a degree of embedding of two or less. The acquisition process itself is quite straightforward. Presented with a grammatical sentence, the program attempts to parse it. If all goes well, the rules exist to handle the sentence, and nothing happens besides a successful parse. However, suppose that the program reaches a point in its attempt where no currently known grammar rules apply. At this point, an acquisition procedure is invoked that tries to construct a single new rule that does apply. If the procedure is successful, the new rule is saved; otherwise, the parse is stopped and the next input sentence read in.

The decision to limit the program to restricted sorts of evidence for its acquisition of new rules -- that is, *positive* data of only limited complexity -- arises out of a commitment to develop the weakest possible acquisition procedure that can still successfully acquire syntactic rules. This commitment in turn follows from the position (cogently stated by Pinker) that "any plausible theory of language learning will have to meet an unusually rich set of empirical conditions. The theory ... will have to be consistent with our knowledge of what language is and of which stages the child passes through in learning it." [2, page 218] In particular, although the final psycholinguistic evidence is not yet in, children do not appear to receive negative evidence as a basis for the induction of *syntactic* rules. That is, they do not receive direct reinforcement for what is not a syntactically well-formed sentence (see Brown and Hanlon [3] and Newport, Gleitman, and Gleitman [4] for discussion).[1] If syntactic acquisition can proceed using just positive examples, then it would seem completely unnecessary to move to any enrichment of the input data that is as yet unsupported by psycholinguistic evidence.[2]

Finally, since the program is designed to glean most of its new rules from simple example sentences (of limited embedding), its developmental course is at least broadly comparable to what Pinker [2] calls a "developmental" criterion: simple abilities come first, and sophistication with syntax emerges only later. The first rules acquired handle simple, few-word sentences and expand the basic phrase structure for English. Later on, rules to deal with more sophisticated phrase structure, alterations of canonical word order, and embedded sentences can be acquired. If an input datum is too complex for the acquisition program to handle at its current stage of syntactic knowledge, it simply parses what it can, and ignores the rest.

2. Constraints Establish the Program's Success

2.1 Current Status of the Acquisition Program

To date, the accomplishments of the research are two-fold. First, from an engineering standpoint, the program succeeds admirably; starting with *no* grammar rules and just two base schema rules, the currently implemented version (dubbed LPARSIFAL) acquires from positive example sentences many of the grammar rules in a "core grammar" of English originally hand-written by Marcus. The currently acquired rules are sufficient to parse simple declaratives, much of the English auxiliary system including auxiliary verb inversion, simple passives, simple *wh*-questions (e.g., *Who did John kiss?*), imperatives, and negative adverbial preposing. Carrying acquisition one step further, by starting with a relatively restricted set of context-free base rule schemas -- the X-bar system of Jackendoff [7] -- the program can also easily induce the proper phrase structure rules for the language at hand. Acquired base rules include those for noun phrases, verb phrases, prepositional phrases, and a substantial part of the English auxiliary verb system.

[1] But children might (and seem to) receive negative evidence for what is a semantically well-formed sentence. See Brown and Hanlon [3].

[2] There is a another reason for rejecting negative examples as inductive evidence: from formal results first established by Gold [5], it is known that by pairing positive and negative example strings with the appropriate labels "grammatical" and "ungrammatical" one can learn "almost any" language. Thus, enriching the input to admit negative evidence broadens the class of "possibly learnable languages" enormously. (Explicit instruction and negative examples are often closely yoked. Compare the necessity for a benign teacher in Winston's blocks world learning program [6].)

Of course, many rules lie beyond the current program's reach. PARSIFAL employed dual mechanisms to distinguish Noun Phrase and *wh*-movements; at present, LPARSIFAL has only a single device to handle all constituent movements. Lacking a distinguished facility to keep track of *wh*-movements, LPARSIFAL cannot acquire the rules where these movements might interact with Noun Phrase movements. Current experiments with the system include adding the *wh* facility back into the domain of acquisition. Also, the present model cannot capture *all* "knowledge of language" in the sense intended by generative grammarians. For example, since the weakest form of the acquisition procedure does not employ backup, the program cannot re-analyze "garden path" sentences and so deduce that they are grammatically well-formed.[3] In part, this deficit arises because it is not perfectly clear to what extent knowledge of *parsing* encompasses all our knowledge about language.[4]

2.2 Constraints and the Acquisition Program

However, beyond the simple demonstration of what can and cannot be acquired, there is a second, more important accomplishment of the research. This is the demonstration that *constraint* is an essential element of the acquisition program's success. To ease the computational burden of acquiring grammar rules it was necessary to place certain constraints on the operation of the model, tightly restricting both the class of hypothesizable phrase structure rules and the class of possible grammar rules.

The constraints on grammar rules fall into two rough groups: constraints on rule application and constraints on rule form. The constraints on rule application can be formulated as specific *locality principles* that govern the operation of the parser and the acquisition procedure. Recall that in Marcus' PARSIFAL grammar rules consist of simple production rules of the form **If** <*pattern*> **then** <*action*>, where a *pattern* is a set of feature predicates that must be true of the current environment of the parse in order for an *action* to be taken. *Actions* are the basic tree-building operations that construct the desired output, a (modified) annotated surface structure tree (in the sense of Fiengo [8] or Chomsky [9]).

Adopting the operating principles of the original PARSIFAL, grammar rules can trigger only by successfully matching features of the (finite) local environment of the parse, an environment that includes a small, three-cell look-ahead buffer holding already-built constituents whose grammatical function is as yet undecided (e.g., a noun phrase that is not yet known to be the subject of a sentence) or single words. It is Marcus' claim that the addition of the look-ahead buffer enables PARSIFAL to always *correctly* decide what to do next – at least for English. The parser uses the buffer to make discriminations that would otherwise appear to require backtracking. Marcus dubbed this "no backtracking" stipulation the *Determinism Hypothesis*. The Determinism Hypothesis crucially entails that all structure the parser builds is correct – that already-executed grammar rules have performed correctly. This fact provides the key to easy acquisition: if parsing runs into trouble, the difficulty can be pinpointed as the current locus of parsing, and not with any already-built structure (previously executed grammar rules). In brief, any errors are assumed to be *locally* and *immediately* detectable. This constraint on error detectability appears to be a computational analogue of the restrictions on a transformational system advanced by Wexler and his colleagues. (see Culicover and Wexler [10]) In their independent but related formal mathematical modelling, they have proved that a finite error detectability restriction suffices to ensure the learnability of a transformational grammar, a fact that might be taken as independent support for the basic design of LPARSIFAL.

Turning now to constraints on rule *form*, it is easy to see that any such constraints will aid acquisition directly, by cutting down the space of rules that can be hypothesized. To introduce the constraints, we simply restrict the set of possible rule <*patterns*> and <*actions*>. The trigger *patterns* for PARSIFAL rules consist of just the items in the look-ahead buffer and a local (two node) portion of the parse tree under construction– five "cells" in all. Thus, *patterns* for acquired rules can be assumed to incorporate just five cells as well. As for *actions*, a major effort of this research was to demonstrate that just three or so basic operations are sufficient to construct the annotated surface structure parse tree, thus eliminating many of the grammar rule actions in the original PARSIFAL. Together, the restrictions on rule patterns and actions ensure that the set of rules available for hypothesis by the acquisition program is finite.

The restrictions just described constrain the space of available grammar rules. However, in the case of phrase structure rules additional strictures are necessary to reduce the acquisitional burden. LPARSIFAL depends heavily on the X-bar theory of phrase structure rules [7] to furnish the necessary constraints. In the X-bar theory, all phrase structure rules for human grammars are assumed to be expansions of just a few *schemas* of a rather specific form; for example, XP-->...X..... Here, the "X" stands for an obligatory phrase structure *category* (such as a Noun, Verb, or Preposition); the ellipses represent slots for possible, but optional "XP" elements or specified grammatical formatives. Actual phrase structure rules are fleshed out by setting the "X" to some known category and settling upon some way to fill out the ellipses. For example, by setting X=N(oun) and allowing some other "XP" to the left of the Noun (call it the category "Determiner") we would get one verson of a Noun Phrase rule, NP-->Determiner N . In this case, the problem for the learner must include figuring out what items are permitted to go in the slots on either side of the "N". Note that the XP schema tightly constrains the set of possible phrase structure rules; for instance, no rule of the form, XP-->X X would be admissible, immediately excluding such forms as, Noun Phrase-->Noun Noun. It is this rich source of constraint that makes the

[3] A related issue is that the current procedure does not acquire the PARSIFAL "diagnostic" grammar rules that exploit look-ahead. Typically, diagnostic rules use the specific features of lexical items far ahead in the look-ahead buffer to decide between alternative courses of action. However, by extending the acquisition procedure -- allowing it to re-analyze apparently "bad" sentences in a careful mode and adding the stipulation that more "specific" rules should take priority over more "general" rules (an often-made assumption for production systems) -- one can begin to accomodate the acquisition of diagnostic rules, and in fact provide a kind of developmental theory for such rules. Work testing this idea is underway.

[4] In most models, the string-to-structural description mapping implied by the directionality of parsing is not "neutral" with respect speakers and listeners.

induction of the proper phrase structure from positive examples feasible; section 4 below illustrates how this induction method works in practice.

Finally, it should be pointed out that the category *names* like "N" and "V" are just arbitrary labels for the "X" categories; the standard approach of X-bar theorists is to assume that the names stand for bundles of distinctive features that do the actual work of classifying tokens into one category bin or another. An important area for future research will be to formulate precise models of how the feature system evolves in interaction with lexical and syntactic acquisition.

This research completed so far assumes that the acquisition procedure is initially provided with just the X-bar schema described above along with an ability to categorize lexical items as *nouns*, *verbs*, or *other*. In addition, the program has an initial schema for a well-formed predicate argument structure, namely, a predicate (verb) along with its "object" arguments. Other phrase structure categories such as *Prepositional Phrase* are inferred by noticing lexical items of unknown categorization and then insisting upon the constraint that only "XP" items or specified formatives appear before and after the main "X" entry. To take an over-simplified example, given the Noun Phrase *the book behind the window*, the presence of the non-Noun, non-Verb *behind* and the Noun Phrase *the window* immediately after the noun *book* would force creation of a new "X" category, since possible alternatives such as, NP-->NP [*the book*] NP [*behind...*] are prohibited by the X-bar ban on directly adjacent, duplicate "X" items.

The X-bar acquisition component of the acquisition procedure is still experimental, and so open to change. However, even crude use of the X-bar restrictions has been fruitful. For one thing, it enables the acquisition procedure to start without any pre-conceptions about canonical word order for the language at hand. This would seem essential if one is interested in the acquisition of phrase structure rules for languages whose canonical Subject-Verb-Object ordering is different from that of English. In addition, since so much of the acquisition of the category names is tied up with the elaboration of a distinctive feature system for lexical items, adoption of the X-bar theory appears to provide a driving wedge into the difficult problems of lexical acquisition and lexical ambiguity. To take but one example, the X-bar theory provides a framework for studying how items of one phrase structure category, e.g., verbs, can be converted into items of another category, e.g., nouns. This line of research is also currently under investigation.

3. The Acquisition Algorithm is Simple

As mentioned, LPARSIFAL proceeds by trying its hand at parsing a series of positive example sentences. Parsing normally operates by executing a series of tree-building and token-shifting grammar rule actions. These actions are triggered by matches of rule patterns against features of tokens in a small three-cell constituent look-ahead buffer and the local part of the annotated surface structure tree currently under construction--the lowest, right-most edge of the parse tree.

Grammar rule execution is also controlled by reference to base phrase structure rules. To implement this control, each of the parser's grammar rules are linked to one or more of the components of the phrase structure rules. Then, grammar rules are defined to be eligible for triggering, or *active*, only if they are associated with that part of the phrase structure which is the current locus of the parser's attentions; otherwise, a grammar rule does not even have the opportunity to trigger against the buffer, and is *inactive*. This is best illustrated by an example. Suppose there were but a single phrase structure rule for English, Sentence-->NounPhrase VerbPhrase. Flow of control during a parse would travel left-to-right in accordance with the S--NP--VP order of this rule, and could activate and deactivate bundles of grammar rules along the way. For example, if the parser had evidence to enter the S-->NP VP phrase structure rule, pointers would first be set to its "S" and the "NP" portions. Then, all the grammar rules associated with "S" and "NP" would have a chance to run and possibly build a Noun Phrase constituent. The parser would eventually advance in order to construct a Verb Phrase, deactivating the Noun Phrase building grammar rules and activating any grammar rules associated with the Verb Phrase.[5] Together with (1) the items in the buffer and (2) the leading edge of the parse tree under construction, the currently pointed-at portion of the phrase structure forms a triple that is called the *current machine state* of the parser.

If in the midst of a parse no currently known grammar rules can trigger, acquisition is initiated: LPARSIFAL attempts to construct a single new executable grammar rule. New rule assembly is straightforward. LPARSIFAL simply selects a new *pattern* and *action*, utilizing the current *machine state* triple of the parser at the point of failure as the new *pattern* and one of four primitive (atomic) operations as the new *action*. The primitive operations are: *attach* the item in the left-most buffer cell to the node currently under construction; *switch* (exchange) the items in the first and second buffer cells; *insert* one of a finite number of lexical items into the first buffer cell; and *insert a trace* (an anaphoric-like NP) into the first buffer cell. The actions have turned out to be sufficient and mutually exclusive, so that there is little if any combinatorial problem of choosing among many alternative new grammar rule candidates. As a further constraint on the program's abilities, the acquisition procedure itself cannot be recursively invoked; that is, if in its attempt to build a single new executable grammar rule the program finds that it must acquire still other new rules, the current attempt at acquisition is immediately abandoned. This restriction has the apparently desirable effect of ensuring that the program use just local context to debug its new rules as well as ignore overly complicated example sentences that are beyond its reach.

[5] This scheme was first suggested by Marcus [1, page 60]. The actual procedure uses the X-bar schemas instead of explicitly labelled nodes like "VP" or "S".

In a pseudo-algorithmic form, the entire model looks like this:

```
Step 1. Read in new (grammatical) example sentence.
Step 2. Attempt to parse the sentence, using modified
PARSIFAL parser.
2.1 Any phrase structure schema rules apply?
  2.1.1 YES: Apply the rule; Go to Step 2.2
  2.1.2 NO: Go to Step 2.2
2.2 Any grammar rules apply?
 (<pattern> of rule matches current parser state)
  2.2.1 YES: apply rule <action>; (continue parse)
        Go to Step 2.1.
  2.2.2 NO: no known rules apply;
          Parse finished?
        YES: (Get another sentence) Go to Step 1.
        NO: parse is stuck
              Acquisition Procedure already invoked?
                YES: (failure of parse or
                acquisition) Go to Step 3.4. or 3.2.3-4
                NO: (Attempt acquisition)
                      Go to Step 3.
Step 3. Acquisition Procedure
3.1 Mark Acquisition Procedure as invoked.
3.2 Attempt to construct new grammar rule
  3.2.2 Try attach
        Success: (Save new rule) Go to Step 3.3
        Failure: (Try next action) Go to Step 3.2.3
  3.2.3 Try to switch first and second buffer cell items.
        Success: (Save new rule) Go to Step 3.3.
        Failure: (Restore buffer and try next action)
            Re-switch buffer cells; Go to Step 3.2.4
  3.2.4 Try insert trace
        Success: (Save new rule) Go to Step 3.3.
        Failure: (End of acquisition) Go to Step 3.4.
3.3 (Successful acquisition)
  Store new rule; Go to Step 2.1.
3.4 (Failure of acquisition)
  3.4.1 (Optional phrase structure rule)
      Continue parse; Advance past current
       phrase structure component; Go to Step 2.1.
  3.4.2 (Failure of parse) Stop parse; Go to Step 1.
```

4. Two Simple Scenarios

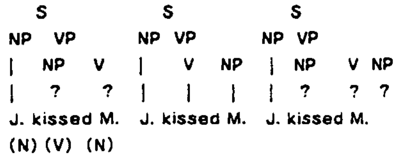
4.1 Phrase Structure for Verb Phrases

To see exactly how the X-bar constraints can simplify the phrase structure induction task, suppose that the learner has already acquired the phrase structure rule for sentences, i.e., something like, Sentence-->Noun Phrase Verb Phrase, and now requires information to determine the proper expansion of a Verb phrase, Verb Phrase-->???.

The X-bar theory cuts through the maze of possible expansions for the right-hand side of this rule. Assuming that Noun Phrases are the only other known category type, the X-bar theory then tells us is that these are the only possible configurations for a Verb Phrase rule:

Verb Phrase-->Noun Phrase Verb
Verb Phrase-->Verb Noun Phrase
Verb Phrase-->Noun Phrase Verb Noun Phrase

If the learner can classify basic word tokens as either nouns or verbs, then by simply matching an example sentence such as *John kissed Mary* against the possible phrase structure expansions, the correct Verb Phrase rule can be quickly deduced:



Only one possible Verb Phrase rule expansion can successfully be matched against the sample string, Verb Phrase-->Noun Phrase(NP) Verb(V) -- exactly the right result for English. Although this is but a simple example, it illustrates how the phrase structure rules can be acquired on the basis of a process akin to "parameter setting"; given a highly constrained initial state, the desired final state can be obtained upon exposure to very simple triggering data.

4.2 A Subject-Auxiliary Verb Inversion Rule

Suppose that at a certain point LPARSIFAL has all the grammar rules and phrase structure rules sufficient to build a parse tree for *John did kiss Mary*. The program now must parse, *Did John kiss Mary?*. No currently known rule can fire, for all the rules in the phrase structure component activated at the beginning of a sentence will have a triggering pattern roughly like *[=Noun Phrase?][=Verb?]*, but the input buffer will hold the pattern *[Did: auxverb, verb][John: Noun Phrase]*, and so thwart all attempts at triggering a grammar rule. A new rule must be written. Acting according to its acquisition procedure, the program first tries to *attach* the first item in the buffer, *did*, to the current active node, S(entence) as the Subject Noun Phrase. The *attach* fails because of category restrictions from the X-bar theory; as a known verb, *did* can't be attached as a Noun Phrase. But *switch* works, because when the first and second buffer positions are interchanged, the buffer now looks like *[John][did]*. Since the ability to parse declaratives such as *John did kiss...* was assumed, an NP-attaching rule will now match. Recording its success, the program saves the *switch* rule along with the current buffer pattern as a trigger for remembering the context of auxiliary inversion. The rest of the sentence can now be parsed as if it were a declarative (the fact that a *switch* was performed is also permanently recorded at the appropriate place in the parse tree, so that a distinction between declarative and inverted sentence forms can be maintained for later "semantic" use.)

5. Summary

A simple procedure for the acquisition of syntactic knowledge has been presented, making crucial use of linguistically- and computationally-motivated *constraints*. Computationally, the system exploits the local and incremental approach of the Marcus parser to ensure that the search space for hypothesizable new rules is finite and small. In addition, rule ordering information need not be explicitly acquired. That is, the system need not learn that, say, Rule A must obligatorily precede Rule B. Extrinsic ordering of this sort appears difficult (if not impossible) to attain under conditions of positive-only evidence. Third, the system acquires its complement of rules via the step-wise hypothesis of new rules. This ability to *incrementally refine* a set of grammar rules rests upon the incremental properties of the Marcus parser, which in turn might reflect the characteristics of the English language itself.

The constraints on the parser and acquisition procedure also parallel many recent proposals in the linguistic literature, lending considerable support to LPARSIFAL's design. Both the power and range of rule actions match those of constrained transformational systems; in this regard, one should compare the (independently) formalized transformational system of Lasnik and Kupin [11] that almost point-for-point agrees with the restrictions on LPARSIFAL. Turning to other proposals, two of LPARSIFAL's rule actions, *attach* and *switch*, correspond to Emonds' [12] categories of *structure-preserving* and *local (minor-movement)* rules. A third, *insert trace*, is analagous to the *move alpha* rule of Chomsky [13]. Rule application is correspondingly restricted. The Culicover and Wexler Binary Principle (an independently discovered constraint akin to Chomsky's Subjacency Condition; see [10]) can be identified with the restriction of rule pattern-matching to a local radius about the current point of parse tree construction (eliminating rules that *directly* require unbounded complexity for refinement). The remaining Culicover and Wexler sufficiency conditions for learnability, including their Freezing and Raising Principles, are subsumed by LPARSIFAL's assumption of strict local operation and no backtracking (eliminating rules that permit the unbounded *cascading* of errors, and hence unbounded complexity for refinement).

These striking parallels should not be taken -- at least not immediately -- as a functional, "processing" explanation for the constraints on grammars uncovered by modern linguistics. An explanation of this sort would take computational issues as the basis for an "evaluation metric" of grammars, and then proceed to tells us why constraints are the way they are and not some other way. But this explanatory result does not necessarily follow from the identity of description between traditional transformational and LPARSIFAL accounts. Rather, LPARSIFAL might simply be translating the transformational constraints into a different medium -- a computational one. Even more intriguing would be the finding that the constraints desirable from the standpoint of efficient parsing turn out to be exactly the constraints that ensure efficient acquisition. The current work with LPARSIFAL at least hints that this might be the case. However, at present the trade-off between the various kinds of "computational issues" as they enter into the evaluation metric is unknown ground; we simply do not yet know exactly what "counts" in the computational evaluation of grammars.

ACKNOWLEDGEMENTS

This article describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643.
The author is also deeply indebted to Mitch Marcus. Only by starting with a highly restricted parser could one even begin to consider the problem of acquiring the knowledge that such a parser embodies. The effort aimed at restricting the operation of PARSIFAL flows as much from his thoughts in this direction as from the research into acquisition alone.

REFERENCES

[1] Marcus, M., A **Theory of Syntactic Recognition for Natural Language**. Cambridge, MA: MIT Press, 1980.

[2] Pinker, S., "Formal Models of Language Acquisition," **Cognition**, 7, 1979, pp. 217-283.

[3] Brown, R., and Hanlon, C., "Derivational Complexity and Order of Acquisition in Child Speech," in J.R. Hayes, ed., **Cognition and the Development of Language**, New York: John Wiley and Sons, 1970.

[4] Newport, E., Gleitman, H., and Gleitman, L., "Mother, I'd Rather do it Myself: Some Effects and Non-effects of Maternal Speech Style," in C. Snow and C. Ferguson, **Talking to Children, Input and Acquisition**, New York: Cambridge University Press, 1977.

[5] Gold, E.M., "Language Identification in the Limit," **Information and Control**, 10, 1967, pp. 447-474.

[6] Winston, P., "Learning Structural Descriptions from Examples," in P. Winston, editor, **The Psychology of Computer Vision**, New York: McGraw-Hill, 1975.

[7] Jackendoff, R., **X-bar Syntax: A Study of Phrase Structure** Cambridge, MA: MIT Press, 1977.

[8] Fiengo, R., "On Trace Theory," **Linguistic Inquiry**, 8, no. 1, 1977, pp. 35-61.

[9] Chomsky, N., "Conditions on Transformations," in S.R. Anderson and P. Kiparsky, (eds.), **A Festschrift for Morris Halle**, New York: Holt, Rinehart, and Winston, 1973.

[10] Culicover, P. and Wexler, K., **Formal Models of Language Acquisition**, Cambridge, MA: MIT Press, 1980.

[11] Lasnik, H. and Kupin, J., "A Restrictive Theory of Transformational Grammar," **Theoretical Linguistics**, 4, no. 3, 1977, pp. 173-196.

[12] Emonds, J., **A Transformational Approach to English Syntax**, New York: Academic Press, 1976.

[13] Chomsky, N., "On Wh-movement," in P. Culicover, T. Wasow, and A. Akmajian, **Formal Syntax**, New York: Academic Press, 1977, pp. 71-132.