

## How Non-Context Free is Variable Binding?

William Marsh, Hampshire College &amp; University of California, San Diego

Barbara H. Partee, University of Massachusetts, Amherst

0. Introduction

Within and across theoretical frameworks, linguists have debated whether various phenomena related to variable binding should be handled within syntax, as part of a syntax-to-semantics mapping, or possibly by one or more separate levels of rules or principles specifically dedicated to such phenomena. Relevant phenomena include the interpretation of pronouns and reflexives as bound variables and the binding of gaps or traces by WH-phrases or other operators. Given current interest in the generative capacity of alternative syntactic theories, it is important to note that whatever level of grammatical description deals with variable binding must involve some non-context-freeness if certain natural restrictions on well-formedness (spelled out below) are taken to belong to grammar at all. In this paper we investigate the formal properties of variable binding via an investigation of two restricted versions of predicate logic. We first review the familiar fact that the set  $\mathcal{L}$  of formulas of first-order predicate logic (FOPL) (allowing free variables and vacuous quantifiers) is a context free (CF) language. We then consider two restricted sublanguages of  $\mathcal{L}$ , the set  $\mathcal{S}$  of Sentences of FOPL (formulas with no free variables), and the set  $\mathcal{H}$  of formulas of FOPL with no vacuous quantifiers. Both  $\mathcal{S}$  and  $\mathcal{H}$  can be shown by established methods to be non-CF. We then describe the indexed grammars of Aho (1968); the class of indexed languages falls strictly between CF and context-sensitive languages. We show that indexed grammars can capture the no-free-variables constraint by exhibiting an indexed grammar for  $\mathcal{S}$ . We then introduce and motivate our main conjecture, which is that indexed grammars cannot capture the no-vacuous-quantifiers constraint, i.e. that  $\mathcal{H}$  is not an indexed language. In the last part of the paper, we discuss the linguistic relevance of our formal investigations.

1. A CFG for a fragment  $\mathcal{L}$  of formulas of FOPL.

For the issues that will concern us in this paper, we can restrict our attention to a very simple fragment of FOPL, with just one quantifier, one binary connective, and one binary relation. To assure an infinite supply of variables from a finite terminal vocabulary, we must "encode" variables as strings. This could be done in many ways, e.g. using "prime" ( $'$ ) as a terminal symbol and generating  $x, x', x'', x''', \dots$ , or encoding subscripts as binary numbers and generating  $x_0, x_1, x_{10}, x_{11}, \dots$ . We choose to generate  $x, xx, xxx, \dots$  as our variables; we will standardly abbreviate them as  $x, x^2, x^3, \dots$ , or equivalently as  $x_1, x_2, x_3, \dots$  (The former abbreviation accords with the usual conventions of formal language theory, while the latter makes our formulas look more like standard FOPL.) The choice of encodings of variables can matter, since e.g. the binary number choice would require "string-matching" power to

check for identity or non-identity of variables in some cases where our notation or the prime notation would require only "counting" power.

Our fragment  $\mathcal{L}$  of formulas of FOPL is generated by the following CFG:

$$\begin{aligned} V_T &= \{\&, =, \forall, x, (\,)\} \\ V_N &= \{S, V\} \\ S &\rightarrow (S \& S) & V &\rightarrow Vx \\ S &\rightarrow (V = V) & V &\rightarrow x \\ S &\rightarrow \forall V S \end{aligned}$$

2. "No free variables" is not CF.

Let  $\mathcal{S}$  be the set of sentences of  $\mathcal{L}$ , i.e. the set of formulas of  $\mathcal{L}$  with no free variables. It is known<sup>1</sup> that  $\mathcal{S}$  is not CF.

Proof.

Let  $\mathcal{R}$  be the set of all formulas of  $\mathcal{L}$  of the form  $\forall x^i(x^j = x^k)$ .

$\mathcal{R}$  is a regular language.

$\mathcal{S} \cap \mathcal{R}$  is the set of all formulas of the form  $\forall x^i(x^i = x^i)$ .

$\mathcal{S} \cap \mathcal{R}$  is not CF. (Compare  $a^n b^n c^n$ .)

Since the intersection of a CF and a regular language is always CF,  $\mathcal{S}$  itself is therefore not CF.

Q.E.D.

3. "No vacuous quantifiers" is not CF.

Let  $\mathcal{H}$  be the set of formulas of  $\mathcal{L}$  with no vacuous quantifiers. That is, any occurrence of a quantifier  $\forall x_i$  in a formula of  $\mathcal{H}$  must be immediately followed by a formula in which  $x$  occurs free, so that the quantifier occurrence does actually bind at least one occurrence of a variable.  $\mathcal{H}$  is not CF.

Proof

Let  $\mathcal{R}'$  be the set of all formulas of  $\mathcal{L}$  of the form:

$$\forall x^i \forall x^j (x^k = x^m)$$

$\mathcal{R}'$  is a regular language.

$\mathcal{H} \cap \mathcal{R}'$  is the set of all formulas of the forms:

$$\begin{aligned} \forall x^i \forall x^j (x^i = x^j), \quad i \neq j, \quad \text{or} \\ \forall x^i \forall x^j (x^j = x^i), \quad i \neq j. \end{aligned}$$

(Note that if  $i = j$  in either subcase, the first quantifier is vacuous and the formula is not in  $\mathcal{H}$ ; hence the condition  $i \neq j$ .)

$\mathcal{H} \cap \mathcal{R}'$  is not CF. (Proof longer but straightforward.) Therefore  $\mathcal{H}$  is not CF.

Q.E.D.

4. Indexed grammars.

In his 1967 Princeton Ph.D. thesis, Alfred V. Aho defined indexed grammars, which characterize the class of indexed languages (the standard reference is Aho (1968), but a more readable introduction can be found in Hopcroft and Ullman (1979).) The indexed languages include all the CF languages and such non-CF ones as  $a^n b^n c^n$  and  $\alpha\alpha$  as well as  $\mathcal{S}$  as we shall see below. The indexed languages are very properly contained in the class of context-sensitive languages.

Informally: an indexed grammar is like a CFG but with "index strings" attached to its non-terminals. The index strings work

like push-down stores: only the most recently added index is "visible", and "reading" an index removes it.

More formally, one kind of normal form for indexed grammars can be defined as follows:

1. The vocabulary consists of three finite disjoint sets,  $V_T$  (terminals),  $V_N$  (non-terminals), and  $V_I$  (indices).

2. Rules are of three sorts:

- (i) Index-introducing rules are of the form  $A \rightarrow B\langle i \rangle$ ,  $A, B \in V_N$ ,  $i \in V_I$ .
- (ii) Index-removing rules are of the form  $A\langle i \rangle \rightarrow B$ ,  $A, B \in V_N$ ,  $i \in V_I$ .
- (iii) CF rules as in CFG's. In applying them, however, the index string (if any) on the mother node is copied onto all non-terminal daughters.

The next section includes an example of an indexed grammar and a derivation.

5. "No free variables" is an indexed language.

We show that indexed grammars can capture the "no free variables" restriction by exhibiting an indexed grammar for the language  $\mathcal{S}$  defined in section 2 and showing some relevant derivations. In the grammar below, an index (sub)string  $t^n$ s encodes the variable  $x^{n+1}$ ; the heart of the bookkeeping lies in the configuration in Figure 1, which shows how each  $V$  node ("variable") comes to have an index string that includes encodings of all the quantifiers that c-command it and hence could bind it. Subsequent rules which expand  $V$  constrain it to expand as one of the encoded variables.

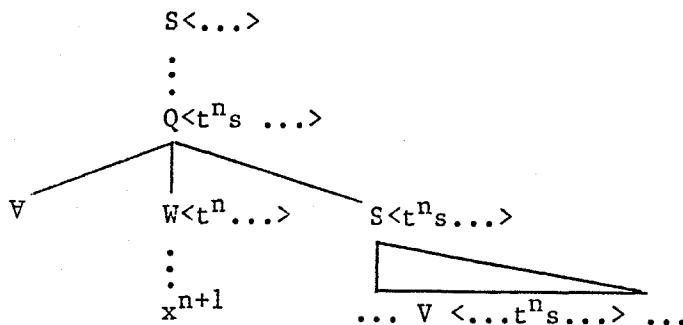


Figure 1

Here is an indexed grammar for  $\mathcal{S}$ :

	$V_T = \{ \&, =, V, x, (, ) \}$	
	$V_N = \{ S, V, Q, W, U, X, E \}$	
	$V_I = \{ s, t \}$ "indices"	
$S \rightarrow (S \& S)$	$S \rightarrow Q\langle s \rangle$	} index-introducing rules
$S \rightarrow (V = V)$	$Q \rightarrow Q\langle t \rangle$	
$Q \rightarrow \forall W S$		
$V \rightarrow W$	$W\langle t \rangle \rightarrow U$	} index-removing rules
$V \rightarrow E$	$W\langle s \rangle \rightarrow X$	
$U \rightarrow W x$	$E\langle t \rangle \rightarrow E$	
$X \rightarrow x$	$E\langle s \rangle \rightarrow V$	

Figure 2 shows the beginning of a derivation, which the reader can use to check the operation of the index-introducing, index-removing, and CF (index-propagating) rules. Filling in the missing steps in the derivations from the three W nodes will show that a W must expand as the first variable encoded in its index string. The V nodes in figure 2 remain to be expanded.

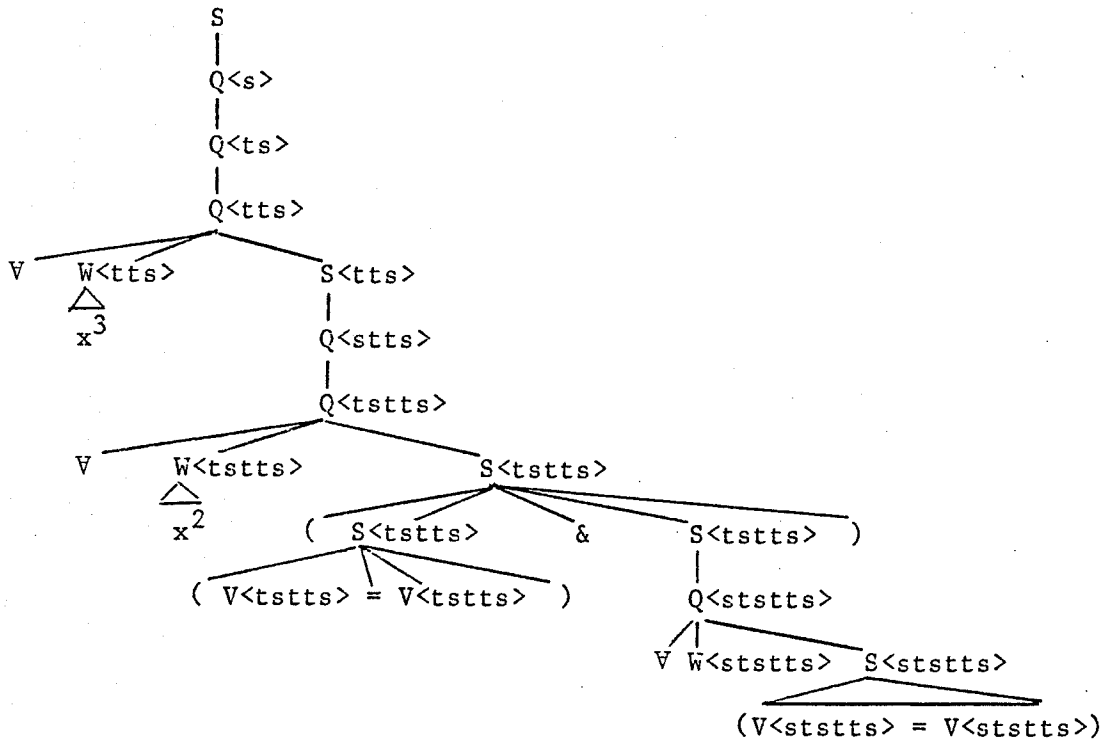


Figure 2

Now we want to show that each V node can be non-deterministically expanded into any of the variables encoded in its index string, but no others. We take  $V<tstts>$  as a representative and consider its possible expansion by cases in Figure 3.

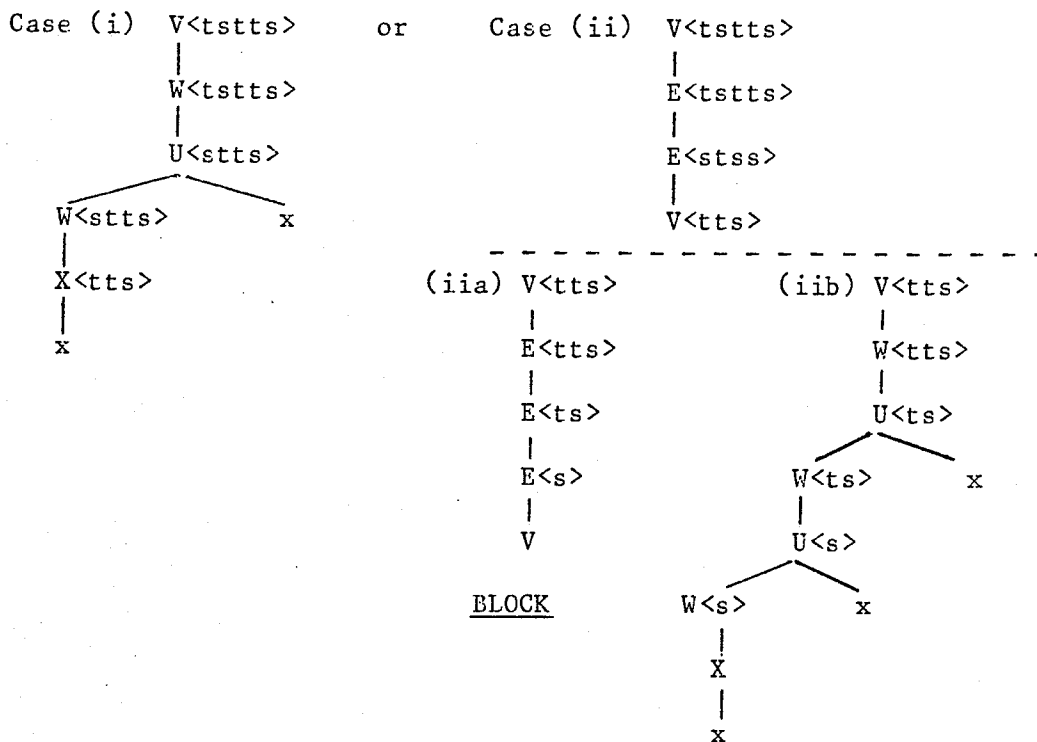


Figure 3

So  $V\langle tsts \rangle$  can expand as either  $x_2$  or  $x_3$ , and nothing else. Similarly  $V\langle ststs \rangle$  can expand as  $x_1$ ,  $x_2$ , or  $x_3$ . Sentences generated from the above tree are all sentences of the form

$$\forall x_3 \forall x_2 ((x_i = x_j) \ \& \ \forall x_1 (x_k = x_m)),$$

where  $i = 2$  or  $3$ ,  $j = 2$  or  $3$ ,  $k = 1$  or  $2$  or  $3$ ,  
 $m = 1$  or  $2$  or  $3$ .

Note that while  $\mathcal{S}$  disallows free variables, it does permit vacuous quantification, as in

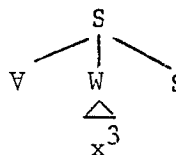
$$\forall x_3 \forall x_2 ((x_3 = x_3) \ \& \ \forall x_1 (x_3 = x_3)).$$

6. Our main conjecture: The "no vacuous quantifiers" language  $\mathcal{U}$  is not an indexed language.

The two constraints we have been considering seem intuitively quite parallel; we could paraphrase them as "every variable must be bound by a variable-binder" and "every variable-binder must bind a variable." Yet it was straightforward to write an indexed grammar for  $\mathcal{S}$ , while we have been unable to find one for  $\mathcal{U}$ , and we believe it to be impossible. There are not as many established techniques for proving languages to be non-indexed as there are for proving non-CF-ness, and finding a proof seems not to be easy.

We can offer some observations about why the method of encoding variables in indices that we used to block free variables doesn't help to block vacuous quantifiers:

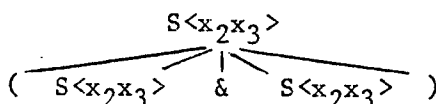
(i) Consider the configuration:



For  $\mathcal{S}$ , we encoded an  $x_3$  in the index of the lower  $S$  and propagated it downward to signal that an  $x_3$  may occur anywhere within that  $S$ . For  $\mathcal{H}$ , an  $x_3$  must occur somewhere within that  $S$ , and furthermore must occur free, i.e. with no more closely c-commanding occurrence of  $\forall x_3$  above it.

(ii) In the grammar for  $\mathcal{S}$ , each  $V$  node introduced by the rule  $S \rightarrow (V = V)$  carried an index string encoding the indices of all the quantifiers that bound it, and we could non-deterministically choose any of those indices as we expanded the  $V$ . But if we wish to prevent vacuous quantifiers, the expansions of the  $V$ 's can no longer be independent from one another: if there are two quantifiers  $\forall x_2 \forall x_3$  c-commanding a formula with three variables, and the first two variables are expanded as  $x_3$ , then the third must be expanded as  $x_2$ . But there is no bound on how far down the "licensing" variable may be.

(iii) Suppose we began the indexing as we did for  $\mathcal{S}$ ; then a node  $S\langle \text{tstts} \rangle$ , which we may abbreviate as  $S\langle x_2 x_3 \rangle$  would "mean" " $x_2$  and  $x_3$  must occur free below". Now consider the conjunction rule,  $S \rightarrow (S \ \& \ S)$ . We don't want to produce



as we did before, since it's not required that  $x_2$  and  $x_3$  both occur free in each conjunct. Rather, we would have to find a way to produce all of the following:

$S$	$\&$	$S\langle x_2 x_3 \rangle$
$S\langle x_2 \rangle$	$\&$	$S\langle x_2 x_3 \rangle$
$S\langle x_3 \rangle$	$\&$	$S\langle x_2 x_3 \rangle$
$S\langle x_2 x_3 \rangle$	$\&$	$S\langle x_2 x_3 \rangle$
$S\langle x_2 \rangle$	$\&$	$S\langle x_3 \rangle$
$S\langle x_2 x_3 \rangle$	$\&$	$S\langle x_3 \rangle$
$S\langle x_3 \rangle$	$\&$	$S\langle x_2 \rangle$
$S\langle x_2 x_3 \rangle$	$\&$	$S\langle x_2 \rangle$
$S\langle x_2 x_3 \rangle$	$\&$	$S$

We don't see how to do this with the permissible rules of index grammars.

Note: It's easy to imagine a natural bottom-up procedure which makes the free variables of the conjunction just the union of the free variables of the conjuncts; but we don't see any way to mimic such a procedure with an indexed grammar, which (a) must be defined top-down, and (b) can't 'read' an index symbol without erasing it.

(iv) Suppose we are expanding an  $S$  which has encoded on it that the variables  $x_1$ ,  $x_3$  and  $x_4$  must occur free within it. Then we must not allow the expansion  $S \rightarrow (V = V)$ , since that can have at most two distinct variables. But again the problem is that we can't read indices without erasing them; we can't check that there are more than two variables without losing our record of what the first two were.

### 7. A further conjecture.

The restriction against vacuous quantifiers has two aspects, of which one can be partly handled.

(i) The central problem: For any occurrence of a quantifier  $\forall x_i$ , there must be an occurrence of  $x_i$  in the  $S$  which it c-commands (but no limit on where within that  $S$  it occurs.)

(ii) The secondary problem: A quantifier  $\forall x_i$  may be vacuous even if it c-commands an  $x_i$ , if there is another "lower" occurrence of  $\forall x_i$  binding that  $x_i$ . If we could prevent a quantifier from c-commanding another quantifier which uses the same variable, we would have only the central problem to worry about.

Consider formulas in prenex form:

$$\forall x_1^{n_1} \forall x_2^{n_2} \dots \forall x_k^{n_k} (\dots), \text{ no quantifier in } (\dots)$$

If all the  $n_i$  could be required to be distinct, then preventing vacuous quantification for such formulas would involve only the central problem. But can even that be done by an indexed grammar?

Let  $\mathcal{A}$  be the set of all 'prenex prefixes' of the form

$$a b^{n_1} a b^{n_2} \dots a b^{n_k} \text{ such that if } i \neq j, \text{ then } n_i \neq n_j.$$

Conjecture 2:  $\mathcal{A}$  is not an indexed language.

Observation: If we require  $n_1 < n_2 < \dots < n_k$ , then  $\mathcal{A}$  as so modified is an indexed language. A natural way of generating it

changes the encoding of variables so that  $W t^{n_1} s t^{n_2} s \dots t^{n_k} s$  produces  $x_1^{n_1+n_2+\dots+n_k}$ . A similar move could block nested identical quantifiers in non-prenex formulas. And 'expressive power' is unaffected by this restriction.

But even if the quantifiers can be regimented in this way, the central problem remains, and our main conjecture stands, even for the set of formulas in prenex form with quantifiers regimented as above.

### 8. Linguistic issues.

Do these results or the outcome of our main conjecture affect the question of whether English or other natural languages are CF? Not necessarily. They don't even show that first-order predicate logic is non-CF, since there is no a priori reason to impose either constraint in the syntax of FOPL. It is not clear, moreover, that there is any reason to stipulate either restriction in the semantics of FOPL either. Vacuous quantifiers are benign but

otiose: a formula with vacuous quantifiers receives the same interpretation as the same formula with the vacuous quantifiers omitted. The set of sentences (or closed formulas) is often accorded special status: it is customary to first give a recursive definition of the set of formulas, then define what it is for an occurrence of a variable to be bound, then define free as not bound, then define a sentence as a formula with no free occurrences of variables. It is really the semantic interpretation rules that give the notions free and bound their significance, but it is necessary to have syntactic definitions of them as well in order to give a sound proof theory. It is less clear that there is any essential need for the notion of sentence; it plays no role in the recursive definition of the set of formulas, since it is crucial for the expressive power of the language to do the syntactic and semantic recursion on formulas rather than sentences. One reason for interest in sentences is that they have a truth value independent of assignment of values to variables, but that semantic property also holds of some open formulas such as  $x_3 = x_3$  and  $(Px_1 \& \sim Px_1)$ . Perhaps sentences are simply the largest natural syntactically definable subclass of formulas with a truth value independent of assignment. But even if the languages  $\mathcal{S}$  and  $\mathcal{U}$  are of no inherent syntactic importance, the notions of free and bound variables are, and we conjecture that close analogs of our formal results could be obtained for the problem of characterizing 'free variable occurrence' and 'bound variable occurrence' syntactically -- e.g. trying to generate a language in which an occurrence of  $x_i$  was preceded by a special symbol  $*$  if it was a free occurrence and by  $\$$  if it was a bound occurrence. We conjecture that there is an indexed grammar but no CF grammar which generates a  $*$  in front of every free occurrence (and only bound variables after  $\$$ ), but that no indexed grammar can generate a  $\$$  in front of every bound occurrence (and only free variables after  $*$ .) If these conjectures are correct, we could say that the distinction between bound and free variables is beyond the power of CF grammars, and one direction of the distinction is beyond the power of indexed grammars.

For natural languages, the difficulty of assessing the relevance of these formal results is compounded by the existence of controversial open questions about the best division of linguistic labor among different components of the grammar or different sets of rules and principles. Certainly it would not be unreasonable for a theorist to argue that if English was CF except for variable-binding phenomena, then that should count as some evidence for the existence of a CF syntactic component and a separate component for handling variable-binding. Another theorist might reply, "But at least you've conceded that there has to be some non-CF power somewhere".<sup>4</sup> It seems that in order to pursue such arguments fruitfully, we need formal measures of the power of systems of syntax-cum-semantics, including syntactic generative capacity, the "expressive power" of the semantics, and the power of the systems for mapping from syntax to semantics.



As a case in point, Higginbotham (forthcoming) argues that English is not CF on the basis of such that relative clauses. His argument depends on the claim that NP's like "every triangle such that two sides are equal", "a number system such that  $2 + 2 = 5$ ," and "costumes such that you can't tell the good guys from the bad guys" are ungrammatical. In a footnote he addresses the contention that such NP's are grammatical, argues that insofar as they are acceptable they are interpreted as elliptical, and that their necessarily elliptical mode of interpretation supports his position. But in the absence of any theory of what it would mean for a whole grammar, including interpretation, to be CF or not, it does make a difference whether the NP's in question are judged syntactically ill-formed or not; if not, Higginbotham's arguments have no effect on the question of whether English syntax is CF or not.

Suppose we did regard as syntactically ill-formed those NP's containing such that clauses in which there was no pronoun available to be construed as bound by the such that. And suppose our conjecture that  $\mathbb{H}$  is non-indexed is correct. It would then be interesting to ask whether Higginbotham's argument that English is not CF could be strengthened to an argument that English is not an indexed language. The constraint Higginbotham is focussing on is that such that cannot be a vacuous variable-binder, which is analogous to the constraint that makes  $\mathbb{H}$  non-indexed. But Higginbotham does not put overt indices (subscripts on pronouns and operators) into the syntax; he considers rather the problem of trying to generate just those strings to which a consistent indexing could be applied in such a way that no such that would come out vacuous. The formal problem appears at first to boil down to the question of whether the following language, which Higginbotham demonstrates to be non-CF, is also non-indexed:

$A = ab^nq(c_1 \cup c_2 \cup c_3 \cup c_4)^n d$ , with the further condition that in no initial segment does the number of occurrences of  $c_4$  ever exceed the number of occurrences of  $c_1$ .

This language is indexed; the following indexed grammar generates it.

$$\begin{array}{ll} V_T = \{a, b, c_1, c_2, c_3, c_4, d, q\} & V_N = \{S, T\} \quad V_I = \{i, k\} \\ S \rightarrow aT\langle k \rangle d & T\langle i \rangle \rightarrow bTc_1 \\ T \rightarrow bTc_2 & T \rightarrow T\langle i \rangle \\ T \rightarrow bTc_3 & T\langle k \rangle \rightarrow q \\ T \rightarrow bT\langle i \rangle c_4 & \end{array}$$

However, Higginbotham used only a subset of such that constructions, sufficient to demonstrate non-CF-ness. A larger subset, in which there was no bound on the number of NP positions within each relative clause, might well form the basis for a proof that English is non-indexed if such constraints are put into the syntax. (This leads to another formal conjecture, which we leave open: consider a language  $\mathcal{L}'$  like  $\mathcal{L}$  but with only a single

variable  $x$ , and consider the sublanguage  $\mathcal{L}'$  consisting of all those formulas of  $\mathcal{L}$  for which each occurrence of  $x$  could be assigned a subscript in such a way that the resulting formula contains no vacuous quantifiers. Is  $\mathcal{L}'$  also non-indexed?)

Another important caveat that must be borne in mind in trying to apply the formal results to natural languages is that neither the no-free-variable constraint nor the no-vacuous-quantifier constraint would necessarily force a language beyond CF if there were additional constraints on where variables and variable-binders could occur. As the GPSG literature amply demonstrates, the association of WH-phrases with gaps in English can be handled by a CFG; threats to CF-ness have arisen from languages like Swedish (Engdahl 1980) with fewer constraints on WH-extraction than English. Locality constraints of various kinds, as well as constraints on multiple extractions and crossed dependencies, can all help serve to limit the variable-binding possibilities to ones describable by CFG's even with constraints against free variables or vacuous quantifiers. Reflexives, for instance, seem generally to be interpreted as bound variables; but in most languages there are tight locality constraints which keep their distribution within the capacity of a CFG to describe.

The fact that words like every and some occur as determiners rather than sentence-prefixes in English and perhaps all languages means that the no-vacuous-quantifier problem does not arise for them; syntactically, each determiner occurrence is locally paired with a common noun occurrence, and in the semantics, the interpretation of the noun phrase introduces a variable for the quantifier to bind whether or not there are any subsequent pronouns interpretable as further variables bound by the same quantifier. Also, (thanks to Janet Fodor for raising this issue), one cannot conclude from the fact that a syntax like May's including a rule of Q-movement at LF may be able to guarantee no vacuous quantifiers that that syntax generates a non-CF language; it might, but one cannot be sure without further investigation, since there are additional constraints imposed by the grammar (e.g. each quantifier originates in a deep structure NP position, and Q-movement is subject to various constraints) which could in principle cut the language down to a CF subset of a non-indexed language.

English also contains "unselective quantifiers" (Lewis 1975) like always, in many cases, etc., but these typically do not require any overt syntactic element to bind; their semantics is notoriously flexible, but the question of what sort of power is required to describe their semantics is one which must await the development of formal measures of power in domains other than syntax.

One further point deserves mention. In order to generate the set of formulas of FOPL with CFG, we had to generate variables as phrases rather than as terminal symbols, so as to keep the terminal vocabulary finite. We don't know whether it's possible to interpret variables' compositionality if the interpretation of  $x^{n+1}$  must be built up from the interpretation of  $x^n$ . If an adequate

semantics for variable-binding phenomena requires an infinite stock of variables, one would want to know something about the formal properties of grammars with an infinite terminal vocabulary, which we do not. Such considerations might also weigh in favor of separating variable-binding from syntax proper in the division of labor among components.

To summarize this section, we have drawn no firm conclusions about the CF-ness of natural languages from our formal results, and have urged caution in any attempt to do so. If our conjecture is correct, some variable-binding languages are not only not CF, but non-indexed. This raises a host of interesting questions for further research, both on how natural languages manage (if they do) to keep these phenomena from forcing the syntax into non-CF-ness, and on how to characterize the formal properties of other components of a grammar so as to be able to pursue further the question of whether "there must be some non-CF-ness somewhere," and to define and explore potentially even more interesting questions about the formal properties of full grammars including syntax, semantics, and the mapping between them.

#### Acknowledgements

Out thanks to the gifted students in the 1982 and 1983 Hampshire College Summer Studies in Mathematics, and to the participants in the Fall 1983 seminar on Mathematical Linguistics taught by Marsh and Emmon Each at UMass., especially to Emmon Each for focussing attention on the need for formal studies of the power of combined systems of syntax and semantics and to Mats Rooth for helping to stimulate the genesis of our main conjecture. Marsh's work was partially supported by National Science Foundation grant #IST 8314396.

#### FOOTNOTES

1. This result and the corresponding result for the language defined in section 3 have been available in the "oral tradition" for some time, but we do not know of any standard written reference for them, perhaps because they are straightforward to prove when the question arises. The result for  $\mathcal{S}$  may have first come to attention in the study of the syntax of ALGOL, a programming language which requires all variables to be declared before they are used; if this requirement is put into the syntax, the language is not CF, but otherwise it is CF -- it is a matter of choice whether to call programs which fail to satisfy the requirement syntactically ill-formed or well-formed but uninterpretable (Pullum 1983). For standard methods for filling in the details of the proofs in sections 2 and 3, see Hopcroft and Ullman (1979).

2. While the set of  $\mathcal{S}$  of sentences of  $\mathcal{L}$  does have some independent interest in logic and some programming languages, the set  $\mathcal{U}$  does not, as far as we know.

3. We thank those colleagues who have tried and failed to prove or refute the conjecture in the six months since we came up with it. Should we thank them by name (they include some fine mathematicians who have worked with indexed grammars), or should we assume that they would prefer to stay off the record? Deep Throat thinks the conjecture is true.

4. The earliest reference we know of to the non-CF-ness of variable-binding phenomena and its relevance to the question of what component(s) should handle variable-binding phenomena in natural language is Fodor (1970), which is marred by a non-proof of the non-CF-ness of "no vacuous quantifiers" but includes a very careful discussion of the difficulty of drawing any immediate linguistic conclusions from the formal results. Partee (1979a) and (1979b) contain some speculations about the inter-connections among unbounded syntactic rules, variable-binding semantic rules, and the non-CF-ness of the set of sentences of FOPL; some of these are too vague to evaluate precisely, but at least one claim made in Partee (1979a) seems to be false, namely that 'it is clearly the "unboundedness" of variable binding that is responsible [for the non-CF-ness of  $\mathcal{S}$ ].

#### REFERENCES

- Aho, Alfred V. (1968) "Indexed Grammars - An Extension of Context-Free Grammars", JACM 15, 647-671.
- Engdahl, Elisabet (1980) Questions in Swedish, unpublished doctoral dissertation, University of Massachusetts, Amherst.
- Fodor, Janet, (1970) "Formal linguistics and formal logic" in John Lyons, ed., New Horizons in Linguistics, Penguin Books Ltd., Middlesex, pp. 198-214
- Higginbotham, James (forthcoming) "English is not a Context-Free Language", Linguistic Inquiry 15.2.
- Hopcroft, J.E. and J. Ullman (1979) Introduction to Automata Theory, Languages and Computation, Addison Wesley.
- Lewis, David (1975) "Adverbs of quantification", in E. L. Keenan, ed., Formal Semantics of Natural Language, Cambridge Univ. Press, Cambridge, (3-15).
- Partee, Barbara (1979a) "Montague Grammar and the Well-Formedness Constraint", in F. Heny and H. Schnelle, eds., Syntax and Semantics, vol. 10 (275-313).
- Partee, Barbara (1979b) "Constraining Montague grammar: a framework and a fragment", in S. Davis and M. Mithun (eds.) Linguistics, Philosophy and Montague Grammar, University of Texas Press, Austin (52-101).
- Pullum, Geoffrey K. (1983) "Context-freeness and the computer processing of human languages", Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics, pp. 1-6.