DEFINING NATURAL LANGUAGE GRAMMARS IN GPSG

Eric Sven Ristad

MIT Artificial Intelligence Lab
545 Technology Square
Cambridge, MA 02139

and

Thinking Machines Corporation
245 First Street
Cambridge, MA 02142

1 Overview

Three central goals of work in the generalized phrase structure grammar (GPSG) linguistic framework, as stated in the leading book "Generalized Phrase Structure Grammar" Gazdar et al (1985) (hereafter GKPS), are: (1) to characterize all and only the natural language grammars, (2) to algorithmically determine membership and generative power consequences of GPSGs, and (3) to embody the universalism of natural language entirely in the formal system, rather than by statements made in it.[1]

These pages formally consider whether GPSG's weak context-free generative power (wcfgp) will allow it to achieve the three goals. The centerpiece of this paper is a proof that it is undecidable whether an arbitrary GPSG generates the nonnatural language Σ^*. On the basis of this result, I argue that GPSG fails to define the natural language grammars, and that the generative power consequences of the GPSG framework cannot be algorithmically determined, contrary to goals one and two.[2] In the process, I examine the linguistic universalism of the GPSG formal system and argue that GPSGs can describe an infinite class of nonnatural context-free languages. The paper concludes with a brief diagnosis of the result and suggests that the problem might be met by abandoning the weak context-free generative power framework and assuming substantive constraints.

1.1 The Structure of GPSG Theory

A generalized phrase structure grammar contains five language-particular components (immediate dominance (ID) rules, metarules, linear precedence (LP) statements, feature co-occurrence restrictions (FCRs), and feature specification defaults (FSDs)) and four universal components: a theory of syntactic features, principles of universal feature instantiation, principles of semantic interpretation, and formal relationships among various components of the grammar.[3]

The set of ID rules obtained by taking the finite closure of the metarules on the ID rules is mapped into local phrase structure trees, subject to principles of universal feature instantiation, FSDs, FCRs, and LP statements. Finally, these local trees are assembled to form phrase structure trees, which are terminated by lexical elements.

The essence of GPSG is the constrained mapping of ID rules into local trees. The constraints of GPSG theory subdivide into absolute constraints on local trees (due to FCRs and LP-statements) and relative constraints on the rule to local tree mapping (stemming from FSDs and universal feature instantiation). The absolute constraints are all language-particular, and consequently not inherent in the formal GPSG framework. Similarly, the relative constraints, of which only universal instantiation is not explicitly language-particular, do not apply to fully specified ID rules and consequently are not strongly inherent in the GPSG framework either.[4] In summary, GPSG local trees are only as constrained as ID rules are: that is, not at all.

The only constraint strongly inherent in GPSG theory (when compared to context-free grammars (CFGs)) is finite feature closure, which limits the number of GPSG nonterminal symbols to be finite and bounded.[5]

1.2 A Nonnatural GPSG

Consider the exceedingly simple GPSG for the nonnatural language Σ^*, consisting solely of the two ID rules

[1] GKPS clearly outline their goals. One, "to arrive at a constrained metalanguage capable of defining the grammars of natural languages, but not the grammar of, say, the set of prime numbers."(p.4). Two, to construct an explicit linguistic theory whose formal consequences are clearly and easily determinable. These 'formal consequences' include both the generative power consequences demanded by the first goal and membership determination: GPSG regards languages "as collections whose membership is definitely and precisely specifiable."(p.1) Three, to define a linguistic theory where *"the universalism [of natural language] is, ultimately, intended to be entirely embodied in the formal system, not expressed by statements made in it."*(p.4, my emphasis)

[2] The proof technique make use of invalid computations, and the actual GPSG constructed is so simple, so similar to the GPSGs proposed for actual natural languages, and so flexible in its exact formulation that *the method of proof* suggests there may be no simple reformulations of GPSG that avoid this problem. The proof also suggests that it is impossible in principle to algorithmically determine whether linguistic theories based on a wcfgp framework (e.g. GPSG) actually define the natural language grammars.

[3] This work is based on current GPSG theory as presented in GKPS. The reader is urged to consult that work for a formal presentation and thorough exposition of current GPSG theory.

[4] I use "strongly inherent" to mean "unavoidable by virtue of the formal framework." Note that the use of problematic feature specifications in universal feature instantiation means that this constraint is dependent on other, parochial, components (e.g. FCRs). Appropriate choice of FCRs or ID rules will abrogate universal feature instantiation, thus rendering it implicitly language particular too.

[5] This formal constraint is extremely weak, however, since the theory of syntactic features licenses more than 10^{774} syntactic categories. See Ristad, E.S. (1986), "Computational Complexity of Current GPSG Theory" in these proceedings for a discussion.

$$S \rightarrow \{\}, H \mid \epsilon$$

This GPSG generates local trees with all possible subcategorization specifications — the SUBCAT feature may assume any value in the non-head daughter of the first ID rule, and S generates the nonnatural language Σ^*.

This exhibit is inconclusive, however. We have only shown that GKPS — and not GPSG — have failed to achieve the first goal of GPSG theory. The exhibition leaves open the possibility of trivially reformalizing GPSG or imposing ad-hoc constraints on the theory such that I will no longer be able to personally construct a GPSG for Σ^*.

2 Undecidability and Generative Power in GPSG

That "$= \Sigma^*$?" is undecidable for arbitrary context-free grammars is a well-known result in the formal language literature (see Hopcraft and Ullman(1979:201–203)). The standard proof is to construct a PDA that accepts all invalid computations of a TM M. From this PDA an equivalent CFG G is directly constructible. Thus, $L(G) = \Sigma^*$ if and only if *all computations of M are invalid*, i.e. $L(M) = \emptyset$. The latter problem is undecidable, so the former must be also.

No such reduction is possible for a proof that "$= \Sigma^*$?" is undecidable for arbitrary GPSGs. In the above reduction, the number of nonterminals in G is a function of the size of the simulated TM M. GPSGs, however, have a bounded number of nonterminal symbols, and as discussed above, that is the essential difference between CFGs and GPSGs.

Only weak generative power is of interest for the following proof, and the formal GPSG constraints on weak generative power are trivially abrogated. For example, exhaustive constant partial ordering (ECPO) — which is a constraint on strong generative capacity — can be done away with for all intents and purposes by nonterminal renaming, and constraints arising from principles of universal feature instantiation don't apply to fully instantiated ID rules.

First, a proof that "$= \Sigma^*$?" is undecidable for context-free grammars with a very small number of terminal and nonterminal symbols is sketched. Following the proof for CFGs, the equivalent proof for GPSGs is outlined.

2.1 Outline of a Proof for Small CFGs

Let $L_{(x,y)}$ be the class of context-free grammars with at least x nonterminal and y terminal symbols. I now sketch a proof that it is undecidable of an arbitrary CFG $G \in L_{(x,y)}$ whether $L(G) = \Sigma^*$ for some x, y greater than fixed lower bounds. The actual construction details are of no obvious mathematical or pedagogical interest, and will not be included. The idea is to directly construct a CFG to generate the invalid computations of the Universal Turing Machine (UTM). This grammar will be small if the UTM is small. The "smallest UTM" of Minsky(1967:276–281) has seven states and a four symbol tape alphabet, for a state-symbol product of 28 (!). Hence, it is not surprising that the "smallest G_{UTM}" that generates the invalid computations of the UTM has seventeen nonterminals and two terminals.

Observe that if a string w is an invalid computation of the universal Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ on input x, then one of the following conditions must hold.

1. w has a "syntactic error," that is, w is not of the form $x_1 \# x_2 \# \cdots \# x_m \#$, where each x_i is an instantaneous description (ID) of M. Therefore, some x_i is not an ID of M.
2. x_1 is not initial; that is, $x_1 \notin q_0\Sigma^*$
3. x_m is not final; that is $x_m \notin \Gamma^* f \Gamma^*$
4. $x_i \vdash_M (x_{i+1})^R$ is false for some odd i
5. $(x_i)^R \vdash_M x_{i+1}$ is false for some even i

Straightforward construction of G_{UTM} will result in a CFG containing on the order of twenty or thirty nonterminals and at least fifteen terminals (one for each UTM state and tape symbol, one for the blank-tape symbol, and one for the instantaneous description separator "#"). Then the subgrammars which ensure that $(x_i)^R \vdash_M x_{i+1}$ is false for some even i and that $x_i \vdash_M (x_{i+1})^R$ is false for some odd i may be cleverly combined so that nonterminals encode more information, and so on.

The final trick, due to Albert Meyer, reduces the terminals to 2 at the cost of a lone nonterminal by encoding the n terminals as $\log n = k$-bit words over the new terminal alphabet $\{0, 1\}$, and adding some rules to ensure that the final grammar could generate Σ^* and not $(\Sigma^4)^*$. The productions

$$N_4 \rightarrow 0L_4 1L_4 \mid 00L_4 \mid 01L_4 \mid 11L_4 \mid \ldots$$

are added to the converted CFG G'_{UTM}, which generates a language of the form

$$L_4 \rightarrow 0000 \mid 0001 \mid 0010 \mid \ldots \mid \epsilon \mid L_4 L_4$$

Where L_4 generates all symbols of length 4, and N_4 generates all strings not of length 0 mod k, where $k = 4$ (i.e. all strings of length 1,2,3 mod 4). Deeper consideration of the actual G_{UTM} reveals that the N_4 nonterminal is also eliminable.

Note that all the preceding efforts to reduce the number of nonterminals and terminals increase the number of context-free productions. This symbol-production tradeoff becomes clearer when one actually constructs G_{UTM}.

Suppose the distinguished start symbol for G_{UTM} is S_{UTM}. Then we form a new CFG consisting of all productions of the form

$$S \rightarrow \{Q - q_0\}\{\Sigma^p - \langle M \rangle\}\{N_4 \cup L_4\}$$

and the one production

$$S \rightarrow S_{UTM}$$

where $\langle M \rangle$ is the length p encoding of an arbitrary TM M, and L_4, N_4 are as defined above.

This ensures that strings whose prefix is "$q_0\langle M \rangle$" will be generated starting from S if and only if they are generated starting from S_{UTM}: that is, they are invalid computations of the UTM on M.

2.2 Some Details for $L_{(x,y)}$ and GPSG

Let the nonterminal symbols Γ, Q, and Σ in the following CFG portion generate the obvious terminal symbols corresponding to the equivalent UTM sets. B is the terminal blank symbol.

Then, the following sketched CF productions generate the IDs of M such that $x_i \mapsto_M (x_{i+1})^R$ is false for some odd i.

The S_4 and S_5 nonterminals are used to locate the even and odd i IDs x_i of w. S_{ok} generates the language $\{\Gamma \cup \#\}^*$.

$$S_4 \rightarrow \Gamma S_4 \mid \#S_5 \mid \#S_{odd}S_{ok}$$
$$S_5 \rightarrow \Gamma S_5 \mid \#S_4 \mid \#S_{even}S_{ok}$$

$$S_{odd} \rightarrow S_1\#$$
$$S_1 \rightarrow \Gamma S_1\Gamma \mid S_2 \mid S_6 \mid S_7$$
$$S_6 \rightarrow \Gamma S_6 \mid \Gamma S_3$$
$$S_7 \rightarrow S_7\Gamma \mid S_3\Gamma$$

$$S_2 \rightarrow \Sigma a \Sigma S_3 \Gamma b \Gamma$$
where $a \neq b$, both in Σ
$$S_2 \rightarrow aqbS_3\{\Gamma^3 - pca\} \quad \text{if } \delta(q,b) = (p,c,R)$$
$$aqbS_3\{\Gamma^3 - cap\} \quad \text{if } \delta(q,b) = (p,c,L)$$
$$S_2 \rightarrow aqB\#B\{\Gamma^3 - pca\} \quad \text{if } \delta(q,B) = (p,c,R)$$
$$aqB\#B\{\Gamma^3 - cap\} \quad \text{if } \delta(q,B) = (p,c,L)$$

$$S_3 \rightarrow \Gamma S_3\Gamma \mid QB\#B\Gamma\Gamma \mid \Sigma B\#B\Gamma$$

S_1 and S_2 must generate a false transition for odd i, while S_3 need not generate a false transition and is used to pad out the IDs of w. The nonterminals S_6, S_7 accept IDs with improperly different tape lengths. The first S_2 production accepts transitions where the tape contents differ in a bad place, the second S_2 production accepts invalid transitions other than at the end of the tape, and the third S_2 accepts invalid end of the tape transitions. Note that the last two S_2 productions are actually classes of productions, one for each string in $\Gamma^3 - pca$, $\Gamma^3 - cap, \ldots$.

The GPSG for "$= \Sigma^*?$" is constructed in a virtually identical fashion. Recall that the GPSG formal framework does not bar us from constructing a grammar equivalent to the CFG just presented. The ID rules used in the construction will be fully specified so as to defeat universal feature instantiation, and the construction will use nonterminal renaming to avoid ECPO.

Let the GPSG category C be fully specified for all features (the actual values don't matter) with the exception of, say, the binary features `GER`, `NEG`, `NULL` and `POSS`. Arrange those four features in some canonical order, and let binary strings of length four represent the values assigned to those features in a given category. For example, C[0100] represents the category C with the additional specifications (`[-GER]`, `[+NEG]`, `[-NULL]`, `[-POSS]`). We replace S_{odd} by C[0000], S_1 by C[0001], S_2 by C[0010], S_3 by C[0011], S_6 by C[0100], and S_7 by C[0101]. The nonterminal Γ is replaced by three symbols of the form C[11xx], one for each linear precedence Γ conforms too. Similarly, Σ is replaced by two symbols of the form C[100x]. The ID rules, in the same order as the CF productions above (with a portion of the necessary LP statements) are:

$$C[0000] \rightarrow C[0001]\#$$
$$C[0001] \rightarrow C[1100]C[0001]C[1101] \mid C[0010] \mid C[0100] \mid C[0101]$$
$$C[0100] \rightarrow C[1100]C[0100] \mid C[1100]C[0011]$$
$$C[0101] \rightarrow C[0101]C[1101] \mid C[0011]C[1101]$$

$$C[0010] \rightarrow C[1000]aC[1001]C[0011]C[1101]bC[1110]$$
where $a \neq b$, both in Σ
$$C[0010] \rightarrow aqbC[0011]\{\Gamma^3 - pca\} \quad \text{if } \delta(q,b) = (p,c,R)$$
$$aqbC[0011]\{\Gamma^3 - cap\} \quad \text{if } \delta(q,b) = (p,c,L)$$
$$C[0010] \rightarrow aqB\#B\{\Gamma^3 - pca\} \quad \text{if } \delta(q,B) = (p,c,R)$$
$$aqB\#B\{\Gamma^3 - cap\} \quad \text{if } \delta(q,B) = (p,c,L)$$

$$C[0011] \rightarrow C[1100]C[0011]C[1101] \mid$$
$$QB\#BC[1100]C[1101] \mid$$
$$C[1000]B\#BC[1100]$$

$$C[1100] < C[0001], C[0011], C[0100], C[0101] < C[1101]$$
$$C[1000] < a < C[1001] < C[0011] < C[1110]$$

While the sketched ID rules are not valid GPSG rules, just as the sketched context-free productions were not the valid components of a context-free grammar, a valid GPSG can be constructed in a straightforward and obvious manner from the sketched ID rules. There would be no metarules, FCRs or FSDs in the actual grammar.

The last comment to be made is that in the actual G_{UTM}, only the number of productions is a function of the size of the UTM. The UTM is used only as a convincing crutch — i.e. not at all. *Only a small, fixed number of nonterminals are needed to construct a CFG for the invalid computations of any arbitrary Turing Machine.*

3 Interpreting the Result

The preceding pages have shown that the extremely simple non-natural language Σ^* is generated by a GPSG, as is the more complex language L_{IC} consisting of the invalid computations of an arbitrary Turing machine on an arbitrary input. Because

L_{IC} is a GPSG language, "$= \Sigma^*$?" is undecidable for GPSGs: *there is no algorithmic way of knowing whether any given GPSG generates a natural language or an unnatural one.* So, for example, no algorithm can tell us whether the English GPSG of GKPS really generates English or Σ^*.

The result suggests that goals 1, 2, 3 and the context-free framework conflict with each other. Weak context-free generative power allows both Σ^* and L_{IC}, yet by goal 1 we must exclude nonnatural languages. Goal 2 demands it be possible to algorithmically determine whether a given GPSG generates a desired language or not, yet this cannot be done in the context-free framework. Lastly, goal 3 requires that all nonnatural languages be excluded on the basis of the formal system alone, but this looks to be impossible given the other two goals, the adopted framework, and the technical vagueness of "natural language grammar."

The problem can be met in part by abandoning the context-free framework. Other authors have argued that natural language is not context-free, and here we argue that the GPSG theory of GKPS can characterize context-free languages that are too simple or trivial to be natural, e.g. any finite or regular language.[6] The context-free framework is both too weak and too strong — it includes nonnatural languages and excludes natural ones. Moreover, CFL's have the wrong formal properties entirely: natural language is surely not closed under union, concatenation, Kleene closure, substitution, or intersection with regular sets![7] In short, the context-free framework is the wrong idea completely, and *this is to be expected*: why should the arbitrary generative power classifications of mathematics (formal language theory) be at all relevant to biology (human language)?

Goal 2, that the naturalness of grammars postulated by linguistic theory be decidable, and to a lesser extent goal 3, are of dubious merit. In my view, substantive constraints arising from psychology, biology or even physics may be freely invoked, with a corresponding change in the meaning of "natural language grammar" from "mentally-representable grammar" to something like "easily learnable and speakable mentally-representable grammar." There is no *a priori* reason or empirical evidence to suggest that the class of mentally representable grammars is not fantastically complex, maybe not even decidable.[8]

One promising restriction in this regard, which if properly formulated would alleviate GPSG's actual and formal inability to characterize only the natural language grammars, is strong nativism — the restrictive theory that the class of natural languages is finite. This restriction is well motivated both by the issues raised here and by other empirical considerations.[9] The restriction, which may be substantive or purely formal, is a formal attack on the heart of the result: the theory of undecidability is concerned with the existence or nonexistence of algorithms for solving problems with an infinity of instances. Furthermore, the restriction may be empirically plausible.[10,11]

The author does not have a clear idea how GPSG might be restricted in this manner, and merely suggests strong nativism as a well-motivated direction for future GPSG research.

Acknowledgments. The author is indebted to Ed Barton, Robert Berwick, Noam Chomsky, Jim Higginbotham, Richard Larson, Albert Meyer, and David Waltz for assistance in writing this paper, and to the MIT Artificial Intelligence Lab and Thinking Machines Corporation for supporting this research.

4 References

Chomsky, N. (1980) *Rules and Representations.* New York: Columbia University Press.

Gazdar, G., E. Klein, G. Pullum, and I. Sag (1985) *Generalized Phrase Structure Grammar.* Oxford, England: Basil Blackwell.

Higginbotham, J. (1984) "English is not a Context-Free Language," *Linguistic Inquiry* 15: 119–126.

[6] While 'natural language grammar' is not defined precisely, recent work has demonstrated empirically that natural language is not context-free, and therefore GPSG theory will not be able to characterize all the human language grammars. See, for example, Higginbotham(1984), Shieber(1985), and Culy(1985). For counterarguments, see Pullum(1985). Nash(1980), chapter 5, discusses the impossibility of accounting for free word order languages (e.g. Warlpiri) using ID/LP grammars. I focus on the goal of characterizing *only* the natural language grammars in this paper.

[7] The finite, bounded number of nonterminals allowed in GPSG theory plays a linguistic role in this regard, because the direct consequence of finite feature closure is that GPSG languages are not truly closed under union, concatenation, or substitution.

[8] See Chomsky(1980:120) for a discussion.

[9] Note that invoking finiteness here is technically different from hiding intractability with finiteness. Finiteness is the correct generalization here, because we are interested in whether GPSG generates nonnatural languages or not, and not in the computational cost of determining the generative capacity of an arbitrary GPSG. A finiteness restriction for the purposes of computational complexity is invalid because it prevents us from properly using the tools of complexity theory to study the computational complexity of a problem.

[10] See Osherson et. al. (1984) for an exposition of strong nativism and related issues. The theory of strong nativism can be derived in formal learning theory from three empirically motivated axioms: (1) the ability of language learners to learn in noisy environments, (2) language learner memory limitations (e.g. inability to remember long-past utterances), and (3) the likelihood that language learners choose simple grammars over more complex, equivalent ones. These formal results are weaker empirically than they might appear at first glance: the equivalence of "learned" grammars is measured using only weak generative capacity, ignoring uniformity considerations.

[11] An alternate substantive constraint, suggested by Higginbotham (personal communication) and not explored here, is to require natural language grammars to generate non-dense languages. Let the *density* of a class of languages be an upper bound (across all languages in the class) on the ratio of grammatical utterances to grammatical and ungrammatical utterances, in terms of utterance lengths. If the density of natural languages was small or even logarithmic in utterance length, as one might expect, and a decidable property of the reformulated GPSG's, then undecidability of "$= \Sigma^*$?" would no longer reflect on the decidability of whether the GPSG framework characterized all and only the natural language grammars. The exact specification of this density constraint is tricky because unit density decides "$= \Sigma^*$?" , and therefore density measurements cannot be too accurate. Furthermore, Σ^* and L_{IC} can be buried in other languages, i.e. concatenated onto the end of an arbitrary (finite or infinite) language, weakening the accuracy and relevance of density measurements.

Hopcroft, J.E., and J.D. Ullman (1979) *Introduction to Automata Theory, Languages, and Computation.* Reading, MA: Addison-Wesley.

Minsky, M. (1967) *Computation: Finite and Infinite Machines.* Englewood Cliffs, N.J: Prentice-Hall.

Nash, D. (1980) "Topics in Warlpiri Grammars," M.I.T. Department of Linguistics and Philosophy Ph.D dissertation, Cambridge.

Osherson, D., M. Stob, and S. Weinstein (1984) "Learning Theory and Natural Language," *Cognition* 17: 1–28.

Pullum, G.K. (1985) "On Two Recent Attempts to Show that English is Not a CFL," *Computational Linguistics* 10: 182–186.

Shieber, S.M. (1985) "Evidence Against the Context-Freeness of Natural Language," *Linguistics and Philosophy* 8: 333–344.