

To provide some details, Cleopatra's parsing is "lexically driven" in that "each word in the input sentence invokes procedures that direct the parsing process". Samad observes that "we can invoke any arbitrary function when evaluating a constraint". Similarly, the integration procedures that build meaning structures may also involve "arbitrary Lisp functions".

Cleopatra attacks ambiguity by use of "confidence levels" related to such phenomena as "the relative frequencies of occurrence of different senses of a word, the likelihood of particular structures, and the correspondence of conjuncts". Parsing is viewed as a parallel process, but the author clearly and honestly states that the existing program is breadth-first.

While the author claims to have achieved a "sharp contrast to the limited domains of previous natural language interfaces", I cannot concur. Although he does handle certain issues in greater detail than have most previous systems (for example, time references and some types of conjunction), there are also, as he so often and honestly admits, many typical distinctions that have been largely ignored (apparently, without suffering a great loss). For example, the system "is not very sophisticated about auxiliaries or adverbs" and does not yet "know about" person or number.

Samad suggests that the "final determinant" of a system with the practical aspirations of his is the evaluation it receives from its intended users. But his claim that "Cleopatra is more than a vehicle to demonstrate the feasibility of our ultimate goal. It is a useful CAD tool" is simply not substantiated. Although "we are confident that experimental studies will confirm the utility of Cleopatra", there's no indication that such investigations have been conducted.

In conclusion, the author is to be commended for his interest in building a complete system that can be used for some meaningful purpose. His frank and honest discussion of the details and limitations of his work are also to be praised, and the presence of some inherently interesting example sentences in the domain under study should be mentioned. However, I find little in the book that helps clarify any problems of language processing, nor do I suspect the techniques presented can provide any "value added" over what's available from existing literature. I am also disappointed by the fact that most of the book concerns implementational issues discussed at the level of data structures. Although I cannot recommend the book as a text, it could certainly be found useful as a case study.

REFERENCE

Brown, J. S. and Burton, R. R. 1975 Multiple Representations of Knowledge for Tutorial Reasoning. In Bobrow, Daniel G. and Collins, Allan (eds.). *Representation and Understanding*. Academic Press, New York, NY: 311-349.

Bruce Ballard is a Member of Technical Staff at AT&T Bell Laboratories in Murray Hill, NJ. He holds a Ph.D. from Duke University and has been working in the field of computational

linguistics since 1977. He has been on the faculty at Ohio State University and Duke University and currently teaches natural language processing at Rutgers University. Ballard's address is 3C-440A, AT&T Bell Labs, 600 Mountain Avenue, Murray Hill, N.J. 07974 USA. E-mail: allegra!bwb.att.com

THE FORMAL COMPLEXITY OF NATURAL LANGUAGE

Walter J. Savitch; Emmon Bach; William Marsh; and Gila Safran-Naveh (eds.)

(University of California, San Diego; University of Massachusetts, Amherst; Xerox Palo Alto Research Center; and University of Cincinnati)

(Studies in linguistics and philosophy 33)
Dordrecht: D. Reidel, 1987, xviii+451 pp.
ISBN 1-55608-046-8, \$69.00, Dfl 145.00, £ 44.95 (hb)

Reviewed by
Alexis Manaster-Ramer
IBM T.J. Watson Research Center

This book anthologizes a number of papers dealing with mathematical models of, and mathematical claims about, human languages. The collection begins with a stage-setting paper by Stanley Peters, "What is mathematical linguistics?" and gives the last word to Gerald Gazdar and Geoffrey K. Pullum in their "Computationally relevant properties of natural languages and their grammars". The papers in between are grouped into three sets:

- Early nontransformational grammar:*
Janet Dean Fodor, "Formal linguistics and formal logic".
Emmon Bach and William Marsh, "An elementary proof of the Peters-Ritchie theorem".
Thomas Wasow, "On constraining the class of transformational languages".
Gilbert H. Harman, "Generative grammars without transformation rules: A defense of phrase structure".
P. T. Geach, "A program for syntax".
Modern context-free-like models:
Geoffrey K. Pullum and Gerald Gazdar, "Natural languages and context-free languages".
Gerald Gazdar, "Unbounded dependency and coordinate structure".
Hans Uszkoreit and Stanley Peters, "On some formal properties of metarules".
Emmon Bach, "Some generalizations of categorial grammars".
More than context-free and less than transformational grammar:
Joan Bresnan et al., "Cross-serial dependencies in Dutch".
Stuart M. Shieber, "Evidence against the context-freeness of natural language".
James Higginbotham, "English is not a context-free language".
Christopher Culy, "The complexity of the vocabulary of Bambara".

Walter J. Savitch, "Context-sensitive grammars and natural language syntax".

William Marsh and Barbara H. Partee, "How non-context free is variable binding".

In addition the editors penned a general introduction, as well as brief prefatory notes to each of the three groups of papers. In this review, I will comment primarily on the editorial part of the work, and on the two articles (Peters's and Savitch's) that have been published here for the first time, with only scattered remarks about points raised by the other papers.

The title of this collection may mean different things to different people, but the editors make it clear that by **formal complexity** we should understand generative capacity, and for the most part weak generative capacity. Specifically, the book is organized around the still unsettled issue of the place where human languages as a class fall in the Chomsky hierarchy, with particular attention to the question of whether they are context-free or, if not, then by how much. The reader should beware of confusion with **computational complexity** (involving bounds on resources such as time and workspace) or **static complexity** (which deals with grammar or program size). In short, this volume is an invitation to the mathematical side of what is commonly, if somewhat inaccurately, known as **generative syntax**, the tradition of linguistic analysis presided over by Chomsky and occasionally haunted by the spirit of Montague.

The book begins with the editors' brief history of work on "a mathematically formal theory of syntax" (pp.vii-viii). This story begins with the rejection of finite-state models as too weak. "In response", transformations were introduced, but these were subsequently seen as being too powerful, in light of the Peters-Ritchie proof (for which see the Bach-Marsh paper in the volume). This result showed that *Aspects*-style transformational grammar generated all the recursively enumerable languages (despite, one might add, the constraints on "recoverability of deletion"). This in turn led to the introduction of context-free models, which are intermediate in formal complexity between finite state and transformational grammar. The focus here is on generalized phrase structure grammar, with distinctly less attention being devoted to Harman's "discontinuous-constituent phrase-structure grammar with subscripts and deletes" (derived in part from earlier work of Yngve's) and to categorial grammar. The last section of the book is supposed to "discuss alternatives to the context-free model" (p.viii), but that is not quite true. Actually, it focuses on a sample of the recent arguments, directed explicitly or implicitly at GPSG, that human languages are not context-free. While these articles imply the need for non-CF models, they do not necessarily present these. Savitch's excellent paper in this section shows why context-sensitive grammars are not a plausible basis for a linguistic theory, but it also offers no alternatives. To be sure,

Bresnan et al. give a brief introduction to, and an example of, lexical-functional grammar, and Bach's paper in the previous section discusses—but seems to shy away from—extensions of categorial grammar that allow (certain) non-CF languages (p.273).

But such real alternatives to GPSG as tree-adjointing and head grammars, both only slightly non-CF, and the more powerful but linguistically less attractive indexed grammars are only adumbrated in Gazdar and Pullum's epilogical paper. However, the principal current model of syntax, which also appears to be intermediate in power between CFG and TG, is the government-binding framework. This is referred to in passing by Gazdar and Pullum (p.401), but only to have scorn heaped on its lack of formal pretensions. The same criticism is presumably implicit in the editorial decision not to discuss GB, but while correct up to a point, it strikes me as ultimately sterile. Ask not whether "N. Chomsky and his students" have developed "a mathematical underpinning" for their theories "in terms of a class of admissible grammars" (*ibid*). ask rather how to do it yourself!

And, if you do, you will almost certainly find a formal complexity class that is incomparable to any of the Chomsky hierarchy classes, something that is also implied by Wasow's argument (p.65) that finite languages should be excluded from the class of possible human languages. There is, I believe, more mathematical interest to GB than meets the eye, even if this has not been brought out well in the literature. It is thus impossible to fault the editors for not reprinting any existing GB readings, but why not commission a special contribution on the formalization of GB, or at least raise the issue editorially?

The heavy emphasis on GPSG, its antecedents, and its consequences should not, however, discourage those readers whose allegiances lie elsewhere. While this volume has its limitations, they are more or less the natural frontiers of the discipline. First, they embody the consensus of many of those most active in formal syntax about what is important and what is not. Second, they reflect the neglect of formal methods and issues by the vast majority of syntacticians. Finally, they have to do with the need to tell a simple story to a reader who is likely to be a relative novice to the field. One has to begin somewhere, and Savitch et al. do begin at just about the right place for somebody interested in the current state of mathematical linguistics of the generative persuasion.

Where I do have a substantive quarrel with Savitch et al. is that their history of the field is so simple as to be misleading. Chomsky did not ever consider finite state models as a basis for linguistic theory, and transformations were not introduced in order to handle the (context-free) center-embedded constructions that seem to make English a non-regular (non-finite-state) language. Transformations were intended to do justice to those features of human language which immediate constitu-

ent and morpheme-to-utterance models, as Chomsky understood them, seemed to handle by contrived and epicyclical means, such as discontinuous constituents, unbounded branching, cross-classifying (complex) categories, zero constituents, separation of categorial and linear information, and a small amount of context sensitivity. The formal complexity argumentation was no more than a mathematical fig leaf on a soft linguistic underbelly. This is, of course, why Chomsky advanced TG, rather than context-free, context-sensitive, or even unrestricted (type-0) grammar, as the correct model of human language.

In this context, the argument that “logically”, if not chronologically, Harman’s proposal of an extended kind of CFG for English syntax was either a “complement” or a “follow-up” to the Peters and Ritchie result (pp.22–23) strikes me as completely at odds with the facts, for Harman’s motivation was avowedly to show that Chomsky had been wrong in denying the linguistic sufficiency of CFG, rather than to rein in the unbridled generative capacity of TG.

In the same way, the result that TGs are equivalent in formal complexity to unrestricted grammars and to Turing machines (i.e., generate all recursively enumerable languages) was not the reason that linguists started giving up on them. Rather, TG itself had been found unable to handle a variety of linguistic phenomena insightfully. Need I remind readers of global, deep structure, surface structure, or transderivational constraints? Who can forget the dark ages when English became a verb-first language? When verb-first languages could not have VPs? When free word order required scrambling? Without extra devices, transformations were seen to be inadequate. With them, they became unnecessary as well. The fact that TG was “too adequate” (p.vii) certainly bothered some people: this may have been a small part of the motivation for the early constraints on deletion developed by Chomsky and Matthews. And when these efforts failed to achieve recursiveness, as shown by Peters and Ritchie, it may be that this result had some psychological effect on the field, though hardly anyone understood it. But the literature attests that the renewed attempts in this direction undertaken by Peters (1973), Lapointe (1977), Wasow (in the volume), and others, were received with apathy. Constraining TG, never a burning issue, had become moot as post-*Aspects* models began taking over.

Of these, the approach which culminated in GB appears to have been motivated not at all by generative capacity considerations, and the various functional models little more. Only in the case of the self-consciously CF models of recent years has formal complexity been a significant issue, but even here the main motivation was linguistic (as noted by Gazdar and Pullum, (pp.405–407)), not formalistic, and usually had to do with the *inadequacy* of TG. For example, Bach (p.263) discusses a number of things that can be done by

his wrapping alias infixing operations within a weakly CF categorial model, but which *Aspects*-style TG is incapable of. Likewise, Gazdar argues for the GPSG treatment of coordination by focusing (in part) on the inability of TG to handle the conjunction of active and passive VPs (p.185). All this does not contradict the Peters–Ritchie theorem or Church’s Thesis (which holds that there is no “machine” more powerful than a Turing machine or, equivalently, no “grammar” more powerful than a TG). Rather it serves to highlight the fact that these results are strictly about *weak* generative capacity. So, if TG is still linguistically inadequate, that simply shows that weak generative capacity is not the measure of complexity that syntacticians have been applying to human languages all these years.

It should also be striking that, having rejected Chomsky’s arguments against phrase structure, the creators of GPSG chose to develop specifically a context-free alternative. If the picture painted by Savitch et al. were the correct one, then surely they would have reached to finite-state models instead, for it is obvious that Chomsky’s “devastating” (p.vii) arguments against the possibility that human languages are regular (finite state) suffered from much the same defects that Pullum and Gazdar’s paper discusses in connection with the arguments against context-freeness. This was shown by Daly (1974) and Levelt (1974), and even more telling is the fact that there have been several attempts to fix these arguments up, notably by Hugo Brandt Corstius (as reported in Levelt, 1974), Langendoen (1977), and—with particular elegance—by Gazdar and Pullum (p.394) themselves. The obvious conclusion is that CFG, with suitable extensions, seemed like a linguistically and computationally reasonable model, whereas regular grammars did not. The case against context-freeness *had* to be wrong, whereas the very similar case against finite-state models *had* to be correct.

Furthermore, Chomsky’s attack on phrase structure was explicitly directed at context-sensitive grammar, which he took to be a formalization of structuralist descriptive practices, yet no one has risen to defend CSG as a model for human language. Nor have Chomsky’s arguments been disputed by the defenders of phrase structure. Instead, from Harman on, the term **phrase structure** has been liberated from its formal definition and allowed to cover devices which get around Chomsky’s objections to CSG, and thereby implicitly attest to the validity of these objections. To be sure, Chomsky had done the opposite to the structuralists, by leaving out of his purported formalization of immediate constituent analysis, almost all the devices which made it workable. So perhaps it is not surprising that the editors of the book apparently tried to ignore the sordid details of the history of syntactic theory by sketching a separate history for the formal complexity field. There is nothing wrong in this per se, so long as readers are made aware that this is an extreme idealization of what actually happened.

This aside, the introductory passages are on the whole quite informative and accurate. For example, we are treated to a nice, straightforward account of the basic concepts and results of the theory of computation (pp.ix–xiii), followed by a good summary of the simplest techniques that can be used to show that a language is—or is not—context-free (p.xiii–xiv). (The reader should note, however, that there are other such techniques.) Another noteworthy feature is the real effort made to correct a number of misconceptions about formal models that are widespread in linguistics (e.g., pp.21, 136, 284). However, there are a few inaccuracies, which I would like to warn the reader about.

The statement that CFGs are a fully adequate model of programming language syntax (pp.vii–viii) contradicts the observation that most programming languages are not CF (p.284). The latter is correct, although it is also true that computer scientists have continued to use CFGs for (incomplete) specifications of programming language syntax and as the basis (but not the whole) of compiler design. In this context, I was also surprised to read that human languages “appear to be more context-free” than programming languages, and that the only proofs of non-CFness for human languages, especially English, have involved “marginal constructs”, such as the *respectively* and *such that* constructions. Actually, the notion of formal complexity that is at issue is essentially that of weak generative capacity, and in these terms it is impossible to distinguish construction(s).

More serious is the fact that this generalization does not do justice to the literature at the time the collection was being put together. The Pullum–Gazdar and Gazdar–Pullum papers cite—though sometimes with all too little faith—many other non-CF constructions in various languages. Notably, they refer to the fact that reduplication is widespread in the languages of the world (p.393) including English (p.399). Indeed, many linguists would probably agree that reduplication—and related phenomena, such as haplology—are universal in human language. However, they are not characteristic of programming languages. On the other hand, I believe that the features which make many programming languages non-CF have close parallels in human language. The declaration of identifiers in languages such as Pascal may be compared with various rules for topic, co-reference, and definiteness. The use of definite articles seems to include (though is not limited to) the ability to determine which “identifiers” (or, NPs) have been previously “declared” by the use of indefinite articles. Likewise, the uniqueness of statement labels in Pascal is related to various devices (never as successful as in programming languages but common nonetheless) for combatting ambiguity, such as the constraint against using, say, the same name to refer indiscriminately in the same discourse to two different individuals of that name.

Likewise, I question the statement that GPSG was

designed to be strongly equivalent to CFG “under many views of the meaning of strong equivalence” (p.136). I do not know of any well-defined sense of strong equivalence under which this is true, although, to be sure, the object grammar induced by a GPSG is, trivially, strongly equivalent to a CFG, on the assumption that we treat the complex nonterminal symbols of the former as equivalent to the unanalyzed symbols of the latter. Nor can I see why anybody would wish to come up with such a definition of strong equivalence, since it would only serve to deny the whole genius of GPSG. The point of creating GPSG, instead of just using CFG, was precisely to capture linguistic generalizations which simple CFG cannot express, in other words to increase the strong generative capacity of the latter without sacrificing its restricted weak generative capacity.

Turning now to the individual papers, I will, as I said, not attempt a detailed review, but only offer a few plaudits, comments, and emendations. One of the highly commendable things that Savitch et al. did in their prefatory remarks was to amplify or correct (e.g., pp.21, 135) incomplete or plain wrong statements in the individual papers. I begin with a few errata that they missed.

First, Pullum and Gazdar assume that it is possible to show that a language is not CF by the use of the pumping lemma (p.147). This is incorrect, inasmuch as there are non-CF languages which satisfy all the conditions of the pumping lemma, e.g., $\{a^n b^p c^q d^r\}$, where either $n = 0$ or $p = q = r$.

Likewise, Shieber makes a formal mistake in his argument that Swiss German is not CF. In the sentences he is considering the number of verbs that govern the dative must equal the number of actual NPs in the dative case in the sentence, and likewise for the accusative. This is crucial to the proof, which relies on the one-to-one correspondence of NPs and verbs in each case. However, he also notes that the NPs themselves are optional, and yet dismisses this as not “affect[ing] the proof” (p.332). This amounts to intersecting Swiss German with a made-up language in which there are always as many object NPs as transitive verbs. This latter language is not regular, and the intersection of a CFL with a non-regular language need not be CF, so the argument is vitiated. To be sure, it can be fixed, assuming Shieber’s data to be correct. The number of dative NPs must still be *no greater than* that of dative-governing verbs, and likewise for the accusative, and this is enough to imply non-context-freeness. The details are left to the reader.

Finally, the Culy paper discusses a reduplicative construction in Bambara of the form “Noun *o* Noun”, and insists that this is a part of the lexicon rather than the syntax of the language. The argument given for this is that this construction shows a tonal behavior (not described) that is not characteristic of comparable sequences of adjacent words. This does not follow, for at least two reasons. One, we have not been shown that

the behavior in question *is* characteristic of word-internal sandhi (in fact, the implication is that it is not). Two, unique (or, irregular) behavior, whether tonal or segmental, is quite common in external sandhi the world over, and hence cannot be a criterion for wordhood.

Finally, I turn to four papers (or at least to certain points raised in them) that I would like to call the reader's special attention to. The first is the introductory paper by Peters, and here I must first clear away a possible terminological problem. In talking about phrase structure grammar (pp.13, 15), Peters is referring neither to normal context-sensitive grammar (which was Chomsky's notion of phrase structure) nor to context-free grammar (which tends to be identified with phrase structure nowadays) nor yet to unrestricted (type-0) grammars (which would be the modern computer scientist's usage). Rather he is concerned with context-sensitive **node admissibility** systems (p.12). Peters refers to this as **parsing** by a phrase structure (i.e., context-sensitive) grammar, but this is a special technical sense of the term, as he notes, and quite distinct from parsing for normal CSGs. On this special interpretation, CSGs "parse" only the CFLs. However, when the reader is told (p.15) that TG can handle certain languages that PSG cannot, e.g., copying languages, he should not conclude that copying languages cannot be generated or parsed (in the usual sense of the term) by non-transformational grammars (e.g., CSGs, indexed grammars, TAGs, and head grammars).

Second, readers should be forewarned that this paper, although published here for the first time, was apparently written a number of years ago (in the early 1970s, I would say). As a result, it is quite dated in certain respects, which should not be taken as indications of a peculiar regression in the field of mathematical linguistics. Mathematical linguists do not spend their time these days on the learnability problem for TG or the "factual" question of whether the base component is innate (pp.15-16). However, in emphasizing the age of Peters's paper, I do not mean to discourage people from reading it. Quite the reverse, in fact, since I firmly believe in the value of continually re-examining "old" work, for this—more than anything—seems to lead to the kind of permanent revolution that characterizes science at its best.

A case in point: Peters's paper is the only place in the volume where Postal's much-maligned argument about Mohawk not being context-free is presented sympathetically. Presumably, this is because at the time the Postal work had not yet been maligned. Yet I think Postal was linguistically quite right, even if he mangled the mathematics of the argument a little more than is—or was—customary. For it does seem to be a fact that Mohawk reduplicates an intransitive subject or an object noun and incorporates one copy in the verb. Pullum and Gazdar in their celebrated attack on Postal (in their paper in the volume) point out that it is also possible to incorporate a noun stem without reduplicating, and to

leave a possessor noun outside the verb. Thus, in addition, to $N-V N$, where N is subject or object, we also get $N_1-V N_2$, where N_1 is subject or object and N_2 is the possessor of N_1 . Surely, while this may make Mohawk *weakly* CF, no one would ever write a grammar of this language that failed to distinguish these two constructions, and to do that we would need a grammar more powerful than CFG in just the way suggested by Postal (and Peters). Thus Postal deserves credit for first calling attention to the fact that human languages make use of reduplication in their syntax, something which has turned out to be true of perhaps all known languages.

Another "old" paper I would call attention to is Wasow's, not so much for its specific proposals about TG, but because it is perhaps the most lucid discussion of the whole question of formal complexity available. I wish the editors had put this paper up front, together with the Peters one. Savitch et al. note that "it can serve as another introductory article to the whole volume" (p.22), but the reader may miss this pointer, and judge the paper solely by its title, which would be a great pity. And while I am on the topic, I would like to call attention to a point made by Wasow, which has only very recently resurfaced in the literature. Many formal languages are obviously too *simple* to serve as possible human languages. Wasow cites finite languages and infinite mirror-image languages as examples, and there are many others, e.g., all languages over one letter. However, linguistic models have always allowed such degenerate cases. Chomsky was no sooner done arguing that human languages were neither finite nor regular than he introduced TGs, which clearly generate all such languages. Moreover, as noted, Wasow's position implies the all-important conclusion that the formal complexity of human languages cannot be measured in terms of the Chomsky hierarchy.

The third paper I would like to single out for praise is that of Pullum and Gazdar. Again, I would like to urge the importance of a point made here which is not directly reflected in the title, and which is too easily missed on a first reading. In their discussion of certain constructions in Dutch, Pullum and Gazdar allow sentences to be generated whose English equivalents would be things like "John will let Mary see Arabic write Peter" and comment that "[i]t is not for the syntax to rule out examples of this sort, for the above example is perfect on the assumption that there is a language or writing system called 'Peter' and a person named 'Arabic' has learned to write it" (p.158).

As far as I know this is the first clear statement in the linguistic literature of what some know as Ziff's Law, the proposition that essentially any string can be the name of someone or something. A corollary of this is, of course, that perhaps every string over the alphabet is a well-formed sentence of every language, since any part that offends thee can always be taken to be a sufficiently strange proper name.¹

And from this it follows that weak generative capacity, as usually conceived, cannot be the right notion of complexity to study in connection with human language (even if it were to be measured by a yardstick different from the Chomsky hierarchy).

Important as the three papers I have mentioned are, my favorite is the previously unpublished—and, this time, brand new—paper by Savitch on context-sensitive grammars. Most of the mathematics underlying the theory of computation and mathematical linguistics is familiar, but the implications of these results for empirical science are rarely discussed explicitly in the literature, and are not widely known. Savitch's contribution is one of the few places where a humdrum formal result, which has been known for a number of years, comes to life.

The point is essentially this: we know that the recursively enumerable (r.e.) languages are accepted by Turing machines, which may use an arbitrary amount of tape before accepting a given string, whereas the CSLs are accepted by (non-deterministic) TMs which have a certain (linear) bound on the amount of tape that may be used. Now, given any r.e. language, we can construct a rather similar CSL, in which the extra tape that was required for the computation is built into the input. To do this, a string of special marker symbols is appended to the end of the real input. A computation proceeds as it would in the original (unbounded) TM except that the marked cells are used instead of blank ones. If the TM runs out of the marked cells to hold intermediate results, then the computation is aborted without accepting. Intuitively, then, the CSL accepted is not very different from the underlying r.e. language, and we see that each r.e. language can be encoded as a CSL. Moreover, the process of converting this CSL to the underlying r.e. language is trivial: delete the marker symbols (though there is no algorithm for going from the r.e. language to the CSL).

Savitch argues, convincingly to my mind, that all this indicates that CSLs have all the structural complexity of r.e. languages, and hence are not suitable as a model of human language—or anything else. We have all been taught to think of the r.e. languages as including "everything", but in a real sense, so do the CSLs. For years, there was a feeling that recursiveness was something to strive for, but now we see that this was much too modest a goal. If human languages are characterized by certain structural universals (e.g., they all use reduplication but not prime length of a string as a grammatical device), then CSLs are already much too inclusive, for they contain all languages that can be characterized in such structural terms. Savitch's contribution should help open up the heavily fortified border between the mathematical theory of computation and empirical science. Perhaps it will be the beginning of a beautiful friendship.

In sum, this book is well worth careful study. It is by no means the last word on the subject, but for many it

may well be the first word, and Savitch et al. have done an excellent job, both in their selections and in their commentaries, of giving a solid introduction to a sparsely cultivated but already complex field. They have also done much to foster the dissemination and the comprehension of formal complexity results in linguistics and to encourage accuracy and lucidity in the formulation, presentation, and interpretation of such results.

REFERENCES

- Daly, R. T. 1974 *Applications of the Mathematical Theory of Linguistics*. Mouton, The Hague, Netherlands.
- Langendoen, D. T. 1977 On the inadequacy of type-3 and type-2 grammars for human languages, In *Studies in Descriptive and Historical Linguistics: Festschrift for Winfred Lehmann* Hopper, P. J. (Ed.). John Benjamins, Amsterdam, Netherlands, 159–172.
- Lapointe, S. 1977 Recursiveness and deletion, *Linguistic Analysis* 3:227–266.
- Levelt, W. J. M. 1974 *Formal Grammars in Linguistics and Psycholinguistics*. Mouton, The Hague, Netherlands.
- Peters, S. 1973 On restricting deletion transformations, In *The Formal Analysis of Natural Languages* Gross, Maurice; Halle, Morris; and Schützenberger, Marcel-Paul (eds.). Mouton, The Hague, Netherlands and Paris, France.
- Ziff, P. 1960 *Semantic Analysis*. Cornell University Press, Ithaca, NY.

NOTE

1. Ziff (1960, pp.85–86) drew a different conclusion, namely, that proper names are not words, but this would still spell the end of any real *raison d'être* for weak generative capacity studies in linguistics, since now when we look at sentences of a human language we must somehow distinguish those containing names from those that do not.

Alexis Manaster-Ramer received his Ph.D. in linguistics from the University of Chicago in 1981, and has taught linguistics at the University of Michigan, and computer science at Wayne State University. He is editor of the book *Mathematics of Language* (John Benjamins 1987), and is presently organizing the new Association for Mathematics of Language. Manaster-Ramer's address is IBM T.J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598.

BRIEFLY NOTED

THE CAMBRIDGE ENCYCLOPEDIA OF LANGUAGE

David Crystal
(University College of North Wales)

Cambridge University Press, Cambridge, England, 1987,
vii+472 pp.
ISBN 0-521-26438-3, \$39.50 (hb)

David Crystal is well known as a linguist who has worked in such assorted areas as stylistics and language pathology. In addition to his research writings, he is also a popularizer of linguistics, being the author of an introductory book and a radio broadcaster in the U.K.

The present volume is also aimed at the general public, and is a particularly ambitious attempt to present the fascination (and breadth) of the sciences of language. Topics covered