Linguistics and the future of computation

by DAVID G. HAYS

State University of New York
Buffalo, New York

My subject is the art of computation: computer architecture, computer programming, and computer application. Linguistics provides the ideas, but the use I make of them is not the linguist’s use, which would be an attempt at understanding the nature of man and of human communication, but the computer scientist’s use. In ancient India, the study of language held the place in science that mathematics has always held in the West. Knowledge was organized according to the best known linguistic principles. If we had taken that path, we would have arrived today at a different science. Our scholarship draws its principles from sources close to linguistics, to be sure, but our science has rather limited itself to a basis in Newtonian calculus. And so a chasm separates two cultures.

The scientific reliance on calculus has been productive. Often understood as a demand for precision and rigor, it has simultaneously made theoreticians answerable to experimental observation and facilitated the internal organization of knowledge on a scale not imagined elsewhere in human history. Very likely, a reliance on linguistic laws for control of science during the same long period would have been less successful, because the principles of linguistic structure are more difficult to discover and manipulate than the principles of mathematical structure; or so it seems after two thousand years of attention to one and neglect of the other. How it will seem to our descendants a thousand years hence is uncertain; they may deem the long era of Western study of mathematical science somewhat pathological, and wonder why the easy, natural organization of knowledge on linguistic lines was rejected for so many centuries.

However that may be, the prospect for the near term is that important opportunities will be missed if linguistic principles continue to be neglected. Linguistics is enjoying a period of rapid growth, so a plethora of ideas await new uses; the computer makes it possible to manipulate even difficult principles. Traditional mathematics seems not to say how computers much beyond the actual state of the art can be organized, nor how programs can be made much more suitable to their applications and human users, nor how many desirable fields of application can be conquered. I think that linguistics has something to say.

THREE LINGUISTIC PRINCIPLES

Since I cannot treat the entire field of linguistics. I have chosen to sketch three principles that seem most basic and far-reaching. Two of them are known to every linguist and applied automatically to every problem that arises. The third is slightly less familiar; I have begun a campaign to give it due recognition.

As everyone knows, the capacity for language is innate in every human specimen, but the details of a language are acquired by traditional transmission, from senior to younger. As everyone knows, language is a symbolic system, using arbitrary signs to refer to external things, properties, and events. And, as everyone certainly knows by now, language is productive or creative, capable of describing new events by composition of sentences never before uttered. Hockett* and Chomsky* explain these things. But of course these are not principles; they are problems for which explanatory principles are needed.

My first principle is stratification.¹² This principle is often called duality of patterning, although in recent years the number of levels of patterning has grown. The original observation is that language can be regarded as a system of sounds or a system of meaningful units; both points of view are essential. One complements the other without supplanting it. Phonology studies language as sound. It discovers that each of the world’s languages uses a small alphabet of sounds, from a dozen to four times that many, to construct all its utterances. The definition of these unit sounds is not physical but functional. In one language, two sounds with physically distinct manifestations are counted as functionally the same; speakers of this language do not acquire the ability to distinguish between the sounds, and can live out their lives without knowing that the sounds are unlike. English has no use for the difference between [p] and [p'], the latter having a little puff of air at the end, yet both occur: [p] in spin, [p'] in pin. Since other languages, notably Thai, use this difference to distinguish utterances, it is humanly possible not only to make the two forms of /p/ but also to hear it. Thus languages arbitrarily map out their alphabets of sounds.

Languages also differ in the sequences of sounds that they permit. In Russian, the word vzbalmoshnyj ‘extrava-
gant’ is reasonable, but the English speaker feels that the initial sequence /vzh/ is extravagant, because initial /v/ in English is not followed by another consonant, and furthermore initial /z/ is not. The Russian word violates English rules, which is perfectly satisfactory to Russian speakers, because they are unacquainted with English restrictions. As Robert Southey put it, speaking of a Russian,

And last of all an Admiral came,
A terrible man with a terrible name,
A name which you all know by sight very well
But which no one can speak, and no one can spell.

(Robert Southey, ‘The March to Moscow.’) Phonology, with its units and rules of combinations, is one level of patterning in language.

That languages are patterned on a second level is so well known as to require little discussion. English puts its subject first, verb second, object last—in simple sentences. Malagasy, a language of Madagascar, puts the verb first, then the object, last the subject.11 Other orders are found elsewhere. English has no agreement in gender between nouns and adjectives, but other languages such as French and Russian, Navaho and Swahili, do; nor is gender controlled by semantics, since many gender classes are without known semantic correlation. Gender is as arbitrary as the English rejection of initial /vzh/.

The units that enter into grammatical patterns are morphemes; each language has its own stock, a vocabulary that can be listed and found once more to be arbitrary. It seems true that some color names are universal—needed in all languages to symbolize genetic capacities—but other color names are also coined, such as the English scarlet and crimson, on arbitrary lines.1,4

The existence of a third level of symbolic patterning is best shown by psychological experiments. Memory for a story is good, but not verbatim. Only the shortest stretches of speech can be remembered word for word; but the ideas in quite a long stretch can be recited after only one hearing if the hearer is allowed to use his own words and grammatical structures.5 The comparison of pictures with sentences has been investigated by several investigators; they use models in which below is coded as not above, forget is coded as not remember, and so on, because they need such models to account for their subjects’ latencies (times to respond measured in milliseconds). Using such models, they can account for the differences between times of response to a single picture, described with different sentences, to an impressive degree of precision.12

Each level of symbolic patterning should have both units and rules of construction. On the phonological level the units are functional sounds; the rules are rules of sequence, for the most part. On the grammatical level the units are morphemes and the rules are the familiar rules of sequence, agreement, and so on. On the third level, which can be called semological or cognitive, the units are often called sememes; the morpheme ‘forget’ corresponds to the sememes ‘not’ and ‘remember’. The rules of organization of this level have not been investigated adequately. Many studies of paradigmatic organization have been reported, sometimes presenting hierarchical classifications of items (a canary is a bird, a dog is a quadruped, etc.), but this is only one of several kinds of organization that must exist. Classification patterns are not sentences, and there must be sentences of some kind on the semological level. Chomsky’s deep structures might be suitable, but Fillmore6 and McCawley13 have proposed different views. What is needed is rapidly becoming clearer, through both linguistic and psychological investigations. The relations that help explain grammar, such as subject and object, which control sequence and inflection, are not the relations that would help most in explaining the interpretation of pictures or memory for stories; for such purposes, notions of agent, instrument, and inert material are more suitable. But the organization of these and other relations into a workable grammar of cognition is unfinished.

Up to this point I have been arguing only that language is stratified, requiring not one but several correlated descriptions. Now I turn to my second principle, that language is internalized. Internalization is a mode of storage in the brain, intermediate between innateness and learning. Concerning the neurology of these distinctions I have nothing to say. Their functional significance is easy enough to identify, however.

What is innate is universal in mankind, little subject to cultural variation. Everyone sees colors in about the same way, unless pathologically color blind. Everyone on earth has three or more levels of linguistic patterning. The distinctions among things (nouns), properties (adjectives), and events (verbs) are so nearly universal as to suggest that this threefold organization of experience is innate. To have grammar is universal, however much the grammars of particular languages vary. The innate aspects of thought are swift, sure, and strong.

What is internalized is not the same in every culture or every person. But whatever a person internalizes is reasonably swift, sure, and strong; less than what is innate, more than what is learned. Besides the mechanisms of linguistic processing, various persons internalize the skills of their arts and crafts; some internalize the strategies of games; and all internalize the content of at least some social roles.

The contrast between learning and internalization is apparent in knowledge of a foreign language. A person who has learned something of a foreign language without internalization can formulate sentences and manage to express himself, and can understand what is said to him, although slowly and with difficulty. A person who has internalized a second language is able to speak and understand with ease and fluency.
Similarly in games, the difference between a master of chess, bridge, or go is apparent. But something more of the difference between learning and internalization is also to be seen here. The novice has a more detailed awareness of how he is playing; he examines the board or the cards step by step, applying the methods he has learned, and can report how he arrives at his decision. Awareness goes with learned skills, not with internalized abilities.

Internalized abilities are the basis of more highly organized behavior. The high-school learner of French is not able to think in French, nor is the novice in chess able to construct a workable strategy for a long sequence of moves. When a language has been internalized, it becomes a tool of thought; when a chess player has internalized enough configurations of pieces and small sequences of play, he can put them together into strategies. The musician internalizes chords, melodies, and ultimately passages and whole scores; he can then give his attention to overall strategies, making his performance lyrical, romantic, martial, or whatever.

What makes learning possible is the internalization of a system for the storage and manipulation of symbolic matter. If a person learns a story well enough to tell it, he uses the facilities of symbolic organization—his linguistic skills—to hold the substance of the story. Much of the content of social roles is first learned in this way; the conditions of behavior and the forms of behavior are learned symbolically, then come to be, as social psychologists put it, part of the self—that is, internalized. In fact, the conversion of symbolic learned material into internalized capacities is a widespread and unique fact of human life. It must be unique, since the symbol processing capacity required is limited to man. This ability gives man a great capacity for change, for adaptation to different cultures, for science: by internalizing the methods of science, he becomes a scientist.

The amount that a person internalizes in a lifetime is easy to underestimate. A language has thousands of morphemes and its users know them. Certainly their semantic and grammatical organization requires tens—more plausibly hundreds—of thousands of linkages. A high skill such as chess takes internalize knowledge of the same order of magnitude, according to Simon and Barenfeld.15

I think that internalized units are more accurately conceived as activities than as inert objects. All tissue is metabolically active, including the tissue that supports memory. Memory search implies an activity, searching, in an inactive medium, perhaps a network of nodes and arcs. More fruitfully we can imagine memory as a network of active nodes with arcs that convey their activity from one to another. A morpheme, then, is an activity seeking at all times the conditions of its application.

My third principle in linguistics is the principle of metalinguistic organization. Language is generally recognized as able to refer to itself; one can mention a word in order to define it, or quote a sentence in order to refute it. A very common occurrence in grammar is the embedding of one sentence within another. A sentence can modify a word in another sentence, as a relative clause:

The boy who stole the pig ran away.

A sentence can serve as the object of a verb of perception, thought, or communication:

I saw him leave. I know that he left.
You told me that she had left.

And two sentences can be embedded in temporal, spatial, or causal relation:

He ran away because he stole the pig.
He stole the pig and then ran away.
He is hiding far from the spot where he stole the pig.

An embedded sentence is sometimes taken in the same form as if it were independent, perhaps introduced by a word like the English that, and sometimes greatly altered in form as in his running away.

The definition of abstract terms can be understood by metalingual linkages in cognitive networks. The definition is a structure, similar to the representation of any sentence or story in a cognitive network. The structure is linked to the term it defines, and the use of the term governed by the content of the structure. Science and technology are replete with terms that cannot be defined with any ease in observation sentences; they are defined, I think, through metalingual linkages. What kind of program is a compiler? What kind of device can correctly be called heuristic? These questions can be answered, but useful answers are complicated stories about the art of programming, not simple statements of perceptual conditions, and certainly not classificatory statements using elementary features such as human, male, or concrete.

I can indicate how vast a difference there is between metalingual operations and others by proposing that all other operations in cognitive networks are performed by path tracing using finite-state automata, whereas metalingual operations are performed by pattern matching using pushdown automata. These two systems differ in power; a finite-state machine defines a regular language and a pushdown automaton defines a context-free language.

A path in a cognitive network is defined as a sequence of nodes and arcs; to specify a path requires only a list of node and arc types, perhaps with mention that some are optional, some can be repeated. A more complex form of path definition could be described, but I doubt that it would enhance the effectiveness of path tracing procedures. In Quillian’s work, for example, one needs only simple path specifications to find the relation between lawyer and client (a client employs a lawyer). To know that a canary has wings requires a simple form of path involving paradigmatic (a canary is a kind of bird) and syntagmatic relations (a bird has wings). The limiting factor is not the complexity of the path that can be
defined from one node to another, but the very notion that node-to-node paths are required. Pattern matching means fitting a template. The patterns I have in mind are abstract, as three examples will show. The first, familiar to linguists, is the determination of the applicability of a grammatical transformation. The method, due to Chomsky, is to write a template called a structure description. The grammatical structure of any sentence is described by some tree; if the template fits the tree of a certain sentence, then the transformation applies to it, yielding a different tree. These templates contain symbols that can apply to nodes in grammatical trees, and relations that can connect the nodes. Chomsky was, I think, the first to recognize that some rules of grammar can be applied only where structure is known; many phenomena in language are now seen to be of this kind. Thus linguists today ask for tree-processing languages and cannot do with string processors.

My second example is the testing of a proof for the applicability of a rule of inference. A proof has a tree structure like the structure of a sentence; whether a rule of inference can be applied has to be tested by reference to the structure. ‘If p and q then p’ is a valid inference, provided that in its application p is one of the arguments of the conjunction; one cannot assert that p is true just because p, q, and a conjunction symbol all occur in the same string.

Finally, I come to metalingual definition. The definition is itself a template. The term it defines is correctly used in contexts where the template fits. As in the first two examples, the template is abstract. A structure description defines a class of trees; the transformation it goes with applies to any tree in the class. A rule of inference defines a class of proofs; it applies to each of them. And a metalingual definition defines a class of contexts, in each of which the corresponding term is usable. Charity has many guises; the story-template that defines charity must specify all of the relevant features of charitable activity, leaving the rest to vary freely.

When the difference in power between finite-state and context-free systems was discovered, it seemed that this difference was a fundamental reason for preferring context-free grammars in the study of natural language. Later it became evident that the need to associate a structural description with each string was more important, since context-free grammars could do so in a natural way and finite-state automata could not. Today linguists and programmers generally prefer the form of context-free rules even for languages known to be finite state, just because their need for structure is so urgent. It may prove the same with pattern matching. In proofs, in transformations, and in definitions it is necessary to mark certain elements: the conclusions of inferences, the elements moved or deleted by transformation, and the key participating elements in definition. (The benefactor is charitable, not the recipient.) Until I see evidence to the contrary, however, I will hold the view that pattern matching is more powerful than path tracing.

Pattern matching is, surely, a reflective activity in comparison with path tracing. To trace a path through a maze, one can move between the hedges, possibly marking the paths already tried with Ariadne’s thread. To see the pattern requires rising above the hedges, looking down on the whole from a point of view not customarily adopted by the designers of cognitive networks. That is, they often take such a point of view themselves, but they do not include in their systems a component capable of taking such a view.

**COMPUTER ARCHITECTURE**

I turn now to the art of computation, and ask what kind of computer might be constructed which followed the principles of stratification, internalization, and metalingual operation. Such a computer will, I freely admit, appear to be a special-purpose device in comparison with the general-purpose machines we know today. The human brain, on close inspection, also begins to look like a special-purpose machine. Its creativity is of a limited kind, yet interesting nevertheless. The prejudice in favor of mathematics and against linguistics prefers the present structure of the computer; but a special-purpose machine built to linguistic principles might prove useful for many problems that have heretofore been recalcitrant.

Stratification is not unknown in computation, but it deserves further attention. The difficulties of code translation and data-structure conversion that apparently still exist in networks of different kinds of computers and in large software systems that should be written in a combination of programming languages are hard to take. The level of morphemics in language is relatively independent of both cognition and phonology. In computer architecture, it should be possible to work with notational schemes independent of both the problem and the input-output system. Whether this level of encoding both data and their organization should be the medium of transmission, or specific to the processor, I do not know. But it is clear that translators should be standard hardware items, their existence unknown in high-level languages. Sophistication in design may be needed, but the problems seem not insurmountable, at least for numerical, alphabetic, and pictorial data. The design of translators for data structures is trickier, and may even prove not to be possible on the highest level.

The lesson to be learned from the separation of grammar and cognition is more profound. Language provides a medium of exchange among persons with different interests and different backgrounds; how they will understand the same sentence depends on their purposes as well as their knowledge. Much difficulty in computer programming apparently can be traced to the impossibility of separating these two levels in programming languages. Programs do not mean different things in different contexts; they mean the same thing always. They are there-
fore called unambiguous, but a jaundiced eye might see a loss of flexibility along with the elimination of doubt. Many simple problems of this class have been solved; in high-level languages, addition is generally not conditioned by data types, even if the compiler has to bring the data into a common type before adding. More difficult problems remain. At Buffalo, Teiji Furugori is working on a system to expand driving instructions in the context of the road and traffic. He uses principles of safe driving to find tests and precautions that may be needed, arriving at a program for carrying out the instruction safely. Current computer architecture is resistant to this kind of work; it is not easy to think of a program on two levels, one of them providing a facility for expanding the other during execution. An interpreter can do something of the sort; but interpretive execution is a high price to pay. If computer hardware provided for two simultaneous monitors of the data stream, one executing a compiled program while the other watched for situations in which the compiled version would be inadequate, the separation of morphemics and cognition might better be realized.

In teaching internalization to students who are mainly interested in linguistics, I use microprogramming as an analogy. What can be seen in the opposite direction is the fantastic extent to which microprogramming might be carried with corresponding improvement in performance. If the meaning of every word in a language (or a large fraction of the words) is internalized by its users, then one may hope that microprogramming of a similar repertory of commands would carry possibilities for the computer somewhat resembling what the speaker gains, to wit, speed.

A computer could easily be built with a repertory of 10,000 commands. Its manual would be the size of a desk dictionary; the programmer would often find that his program consisted of one word, naming an operation, followed by the necessary description of a data structure. Execution would be faster because of the intrinsically higher speed of the circuitry used in microprogramming. Even if some microprograms were mainly executive, making numerous calls to other microprograms, overall speed should be increased. At one time it would have been argued that the art could not supply 10,000 widely used commands, but I think that time is past. If someone were inclined, I think he could study the literature in the field and arrive at a list of thousands of frequently used operations.

Parallel processing adds further hope. If a computer contains thousands of subcomputers, many of them should be operating at each moment. Even the little we know about the organization of linguistic and cognitive processing in the brain suggests how parallel processing might be used with profit in systems for new applications.

A morphemic unit is the brain seems to be an activity, which when successful links a phonological string with one or more points in a cognitive network. If these units had to be tested sequentially, or even by binary search, the time to process a sentence would be great. Instead all of them seem to be available at all times, watching the input and switching from latency to arousal when the appropriate phonological string appears. If each were a microprogram, each could have access to all input. Conflicts inevitably arise, with several units aroused at the same time. Grammar serves to limit these conflicts; a grammatical unit is one with a combination of inputs from morphemic units. When a morphemic unit is aroused, it signals its activation to one or several grammatical units. When a proper combination of morphemic units is aroused, the grammatical unit is in turn aroused and returns feedback to maintain the arousal of the morphemic unit which is thereupon enabled to transmit also to the cognitive level. Thus the condition for linkage between phonology and cognition is a combination of grammatical elements that amounts to the representation of a sentence structure. This is Lamb's model of stratal organization, and shows how grammar reduces lexical ambiguity. The problem it poses for computer architecture is that of interconnection; unless the morphemic units (like words) and the grammatical units (like phrase-structure rules) are interconnected according to the grammar of a language, nothing works. The computer designer would prefer to make his interconnections on the basis of more general principles; but English is used so widely that a special-purpose computer built on the lines of its grammar would be acceptable to a majority of the educated persons in the world—at least, if no other were on the market.

A similar architecture could be used for other purposes, following the linguistic principle but not the grammar of a natural language. Ware mentions picture processing and other multidimensional systems as most urgently needing increased computing speed. Models of physiology, of social and political systems, and of the atmosphere and hydrosphere are among these. Now, it is in the nature of the world as science knows it that local and remote interactions in these systems are on different time scales. A quantum of water near the surface of a sea is influenced by the temperature and motion of other quanta of water and air in its vicinity; ultimately, but in a series of steps, it can be influenced by changes at remote places. Each individual in a society is influenced by the persons and institutions close to him in the social structure. Each element of a picture represents a portion of a physical object, and must be of a kind to suit its neighbors.

To be sure, certain factors change simultaneously on a wide scale. If a person in a picture is wearing a striped or polka-dotted garment, the recognition of the pattern can be applied to the improvement of the elements throughout the area of the garment in the picture. A new law or change in the economy can influence every person simultaneously. Endocrine hormones sweep through tissue rapidly, influencing every point almost simultaneously. When a cloud evaporates, a vast area is suddenly exposed to a higher level of radiation from the sun.

These situations are of the kind to make stratification a helpful mode of architecture. Each point in the grid of
picture, physiological organism, society, or planet is connected with its neighbors on its own stratum and with units of wide influence on other strata; it need not be connected with remote points in the same stratum.

Depending on the system, different patterns of interaction have to be admitted. Clouds are formed, transported, and evaporated. Endocrine glands, although they vary in their activity, are permanent, as are governments. Both glands and governments do suffer revolutionary changes within the time spans of useful simulations. In a motion picture, objects enter and depart.

How the elements of the first stratum are to be connected with those of the next is a difficult problem. It is known that the cat's brain recognizes lines by parallel processing; each possible line is represented by a cell or cells with fixed connections to certain retinal cells. But this does not say how the cat recognizes an object composed of several lines that can be seen from varying orientations. Switching seems unavoidable in any presently conceivable system to connect the level of picture elements with the level of objects, to connect the level of persons with the level of institutions, to connect the elements of oceans with the level of clouds, or to connect the elements of the morphemic stratum with the level of cognition in linguistic processing.

In computation, it seems that path tracing should be implicit, pattern matching explicit. The transmission of activity from a unit to its neighbors, leading to feedback that maintains or terminates the activity of the original unit, can be understood as the formation of paths. Something else, I think, happens when patterns are matched.

A typical application of a linguistically powerful computer would be the discovery of patterns in the user's situation. The user might be a person in need of psychiatric or medical help; an experimenter needing theoretical help to analyze his results and formulate further experiments; a lawyer seeking precedents to aid his clients; or a policy officer trying to understand the activities of an adversary. In such cases the user submits a description of his situation and the computer applies a battery of patterns to it. The battery would surely have to be composed of thousands of possibilities to be of use; with a smaller battery, the user or a professional would be more helpful than the computer.

If the input is in natural language, I assume that it is converted into a morphemic notation, in which grammatical relations are made explicit, as a first step.

On the next level are thousands of patterns, each linked metalingually to a term; the computer has symbolic patterns definitive of charity, ego strength, heuristics, hostility, and so on. Each such pattern has manifold representations on the morphemic stratum; these representations may differ in their morphemes and in the grammatical linkages among them. Some of these patterns, in fact, cannot be connected to the morphemic stratum directly with any profit whatsoever, but must instead be linked to other metalingual patterns and hence ultimately to morphemic representations. In this way the cognitive patterns resemble objects in perception that must be recognized in different perspectives.

Grammatical theory suggests an architecture for the connection of the strata that may be applicable to other multistratal systems. The two strata are related through a bus; the object on the lower stratum is a tree which reads onto the bus in one of the natural linearizations. All of the elements of all of the patterns on the upper stratum are connected simultaneously to the bus and go from latent to arousal when an element of their class appears; these elements include both node and arc labels. When the last item has passed, each pattern checks itself for completeness; all patterns above a threshold transmit their arousal over their metalingual links to the terms they define. Second-order patterns may come to arousal in this way, and so on.

If this model has any validity for human processing, it brings us close to the stage at which awareness takes over. In awareness, conflicts are dealt with that cannot be reduced by internalized mechanisms. The chess player goes through a few sequences of moves to see what he can accomplish on each of them; the listener checks out those occasional ambiguities that he notices, and considers the speaker's purposes, the relevance of what he has heard to himself, and so on. The scientist compares his overall theoretical views with the interpretations of his data as they come to mind and tries a few analytic tricks. In short, this is the level at which even a powerful computer might open a dialogue with the user.

Would a sensible person build a computer with architecture oriented to a class of problems? I think so, in a few situations. Were listed some problems for which the payoff function varies over a multibillion-dollar range: foreign policy and arms control, weather and the environment, social policy, and medicine. With such payoffs, an investment of even a large amount in a more powerful computer might be shown to carry a sufficient likelihood of profit to warrant a gamble. In the case of language-oriented architecture, it is not hard to develop a composite market in which the users control billions of dollars and millions of lives with only their own brains as tools to link conceptualization with data. A president arrives at the moment of decision, after all the computer simulations and briefings, with a yellow pad and a pencil; to give him a computer which could help his brain through multistratal and metalingual linkages of data and theories would be worth a substantial investment.

Can these applications be achieved at optimal levels without specialized architecture? I doubt it. Parallel processing with general-purpose computers linked through generalized busses will surely bring an improvement over serial processing, but raises problems of delay while results are switched from one computer to another and does nothing to solve software problems. Specialized architecture is a lesson to be learned from linguistics with consequences for ease of programming, time spent in compilation or interpretation, and efficiency of parallel processing.
COMPUTATIONAL LINGUISTICS

I have delayed until the end a definition of my own field, which I have presented before. It should be more significant against the background of the foregoing discussion. The definition is built upon a twofold distinction. One is the distinction, familiar enough, between the infinitesimal calculus and linguistics. The calculus occupies a major place in science, giving a means of deduction in systems of continuous change. It has developed in two ways: Mathematical analysis, which gives a time-independent characterization of systems including those in which time itself is a variable—time does not appear in the metasystem of description. And numerical analysis, in which time is a variable of the metasystem; numerical analysis deals in algorithms.

Linguistics, also, has developed in two ways. The time-independent characterizations that Chomsky speaks of as statements of competence are the subject of what is called linguistics, with no modifier. This field corresponds to the calculus, or to its applications to physical systems. Time-dependent characterizations of linguistic processes are the subject matter of computational linguistics, which also has two parts. Its abstract branch is purely formal, dealing with linguistic systems whether realized, or realizable, in nature; its applied branch deals with algorithms for the processing of naturally occurring languages.

I have undertaken to show that the concepts of abstract computational linguistics provide a foundation for non-numerical computation comparable to that provided by the calculus for numerical computation. The work is still in progress, and many who are doing it would not be comfortable to think of themselves as computational linguists. I hope that the stature of the field is growing so that more pride can attach to the label now and hereafter than in earlier days.

REFERENCES