FINAL REPORT

Report No. 16

W. P. Lehmann, Director

E. D. Pendergraft, Associate Director

Prepared for

National Science Foundation
Grant NSF GN-54

Also for U.S. Army Signal Research and Development Labora[illegible]
Contract DA36-039 SC 78911 File Number 1E[illegible]-PM-59-[illegible]

LINGUISTICS RESEARCH CENTER
The University of Texas
Box 7247, University Station
Austin 12, Texas

LRC 63-P16

June 1963

TABLE OF CONTENTS

1 PURPOSE

This study is concerned with the development of means for automatic translation of foreign languages into English. Although German was chosen as the source language, the need for methods which would be generally applicable to other languages was recognized in the research from its inception in 1959. A long-range approach was taken to afford the opportunity for examination of fundamental principles and for consequent development of translation techniques of exceptional generality.

Generalized translation procedures will accordingly be the main subject matter of this report. One should not infer that we claim a practical solution to generalized translation, for this is not the case. However, a working hypothesis has derived from the four years of basic research which increasingly shows promise of accounting for some of the important empirical relations in human translation. This hypothesis is comprised of the specific techniques to be discussed.

We believe, therefore, that the techniques will provide a foundation for mechanically produced translations of much higher quality than those currently available.

Only fragmentary evidence will be offered to support this conviction, however, since our immediate purpose is to test the techniques rather than to defend them. We will exert our main effort in explaining the hypothesis, acknowledging that it is as yet essentially untested.

Reasonable precautions have been taken to insure its validity through manual simulation of the translation procedures. Nevertheless, because of the remarkable complexity of language, a really comprehensive evaluation will not be possible until the techniques can be carried out by computer programs. Even then verification will entail considerable ingenuity and labor.

These remarks would not be necessary if mechanical translation were universally accepted as a scientific problem. Unfortunately, a so-called practical orientation in translation research has often ignored the obvious principle that a hypothesis must first be formulated before it can be tested. Consequently, time has been lost in fruitless ad hoc experimentation.

This report is primarily intended to cover those phases of the research which led to formulation of our hypothesis and to principles of application. It includes the conclusions which resulted from a background study of earlier work and from a study concerned chiefly with linguistic analysis of German and English. The aim of the latter study was to evaluate a series of alternative structural theories which culminated in the hypothesis reported.

Numerous concepts which were discarded in reaching our present theoretical orientation will not be mentioned, since they would further complicate an already complicated explanation. Some of them undoubtedly would be better forgotten. We will also simplify some of our experiences in order to present a more coherent account. The necessity for these liberties in reporting exploratory research will no doubt be understood by our colleagues.

The need for methods to deal with semantic ambiguities in mechanical translation is widely recognized. To our knowledge, the structural hypothesis which we will outline is the first to suggest that generalized procedures analogous to those used in automatic syntactic analysis may be applicable in semantic analysis. Our next objective is to verify this principle empirically. A second attack on critical semantic problems will be made by means of generalized techniques which transfer unresolved ambiguities from one language to another through an interlingua.

The work reported was performed under sponsorship of U. S. Army Electronics Research and Development Laboratory, contract DA 36-039 SC 78911, during the period from 1 May 1959 to 30 April 1963. Substantial support from the National Science Foundation, through grants NSF G-19277 and GN-54, has materially increased our rate of progress

since 1 September 1961. Individual contributions of these two projects will not be distinguished because any attempt at separation would be artificial. The projects have cooperated closely in all lines of study.

Computer programming to support these experiments has been continued under U. S. Army Electronic Research and Development Laboratory contract DA 36-039 AMC-02162 (E). Descriptive linguistic research, by which German and English syntactic and semantic data will be collected for the experiments, is now being performed largely under National Science Foundation grant GN-54.

2 ABSTRACT

Four years of mechanical translation research are reported. A study was made of previous research efforts in the field. A working hypothesis was formulated for syntactic, semantic and pragmatic structural relations occurring in natural language. The general syntax of programming languages to be used in describing language data were derived from the formalized hypothesis, and programming criteria were extracted for generalized analysis and synthesis algorithms. A generalized algorithm for interlingual transfer was also derived from formalized languages used for interlingual description. All of the processes are stochastic. A computer system to implement the theory is being programmed. Syntactic analysis and all of its supporting programs have been completed, as well as programs which maintain semantic descriptions.

3 PUBLICATIONS, LECTURES, REPORTS AND CONFERENCES

On 8 February at Fort Huachuca the Linguistics Research Center formally demonstrated Syntactic Monolingual Recognition and supporting programs. Demonstration outputs were sent to the U. S. Army sponsor, who indicated that the outputs comprised satisfactory evidence of completion of this phase of the project. The outputs were also sent to Richard See of NSF.

Army personnel who observed the demonstration were: Jeff Abraham, Steve Ambagis, Capt. C. D. Broadbent, Bruno Schröder, and John M. Wright. Sam Holland of United Research Services, Inc., and James G. Renno, Jr. of Thompson Ramo Wooldridge, Inc. also observed the demonstration. The Linguistics Research Center was represented by Mrs. Ruth Ambagis.

Mr. Pendergraft addressed the Linguistics Club of The University of Texas on 11 February. His topic was "Structural Semantics".

Dr. John McCarthy, Computation Science Division, Stanford University, visited LRC on 21 February. He is primarily interested in research related to artificial intelligence.

On 25-27 February Dr. Lehmann attended the "International Conference on Vistas in Information Handling" in New York, where he spoke on "Structural Models for Linguistic Automation".

Günther Beyer from the Administrative Office with the German Military Representative to MC/NATO in Washington visited LRC briefly while in Austin (3-7 March) to interview students for teaching jobs in Germany. Mr. Beyer discussed with Dr. Tosh a glossary of German expressions compiled by Mr. A. Lissance, a translator for the State Department.

On 5 March Dr. Dasher, Department of Electrical Engineering, Georgia Institute of Technology, visited LRC and discussed with Dr. Tosh the research program of the Center.

Mr. Pendergraft spoke on "Linguistic Automation" to the San Antonio chapter of the American Institute of Aeronautics and Astronautics. He took the opportunity to talk with persons engaged in research for NASA, particularly Colonel Stapp, who served as chairman during the meeting.

On 19 March Dr. Lehmann gave an address to The University of Texas chapter of Scolia, in which he discussed the concepts and methods of research at LRC.

Mr. Estes and Miss Brady attended the NSF Seminar on Information Retrieval held in Washington on 19-20 March.

The seminar was devoted mainly to an exposition of the work done by Roger Needham and colleagues at the Cambridge Language Research Unit. Dr. Needham outlined his clump theory, and a general discussion ensued regarding the theory as a possible basis for an information retrieval system. His wife, Karen Sparck-Jones, discussed an experimental approach to constructing an automatic dictionary, including application of clump theory to semantic classification. Mr. Estes and Miss Brady spoke to the group on LRC research. They elaborated certain classification problems involved in writing second order rules.

Dr. Gerhard Reitz, Associate Project Manager of the machine translation project at Thompson Ramo Wooldridge, Inc., visited LRC on 20-22 March. He is primarily concerned with programming of linguistic processes. He was briefed on the underlying theoretical concept and the actual programming systems of LRC by Mr. Pendergraft, Mr. Jonas and Dr. Tosh.

On 28 March David V. Savidge, Manager, Systems Evaluation, Univac Division, Sperry Rand Corporation, visited LRC accompanied by Tom McFarlin, the local Resident Manager. Mr. Savidge is interested in the application of LRC systems to the documentation of programming languages.

Dr. E. S. Klima, Professor of Linguistics, Massachusetts Institute of Technology, visited LRC on 3-4 April.

He presented a paper entitled "A Transformational Approach to Syntax", and consulted briefly with Mr. Jonas and Dr. Tosh on LRC work on programming and descriptive linguistics.

Dr. Roger Needham and wife, Karen Sparck-Jones, visited LRC on 2-12 April. They presented five lectures, and discussed at length the work being done by the LRC staff. The work concerning semantic structure was of particular interest to them.

On 23-28 April Dr. Lehmann gave a series of lectures at The University of California at Los Angeles. The lectures were related to LRC research.

4 WORK ACCOMPLISHED

The four tasks required of this project were specified essentially as follows:

(a) A background study of previous mechanical translation research, with a view toward determining areas adequately covered by other efforts and work remaining to be done before an operational translation facility can be established.

(b) A theoretical study of the linguistic basis of mechanical translation and principles of implementation.

(c) A programming study to implement mechanical translation of at least one language into English.

(d) A documentation study to describe results of linguistic analysis, computer programs developed to implement translation, including all related instruction manuals, and estimates of readability of translation outputs.

The background and theoretical studies were completed under this contract. Computer programming was initiated and, with descriptive linguistic research required for the implementation, is still in progress. Results of linguistic analysis are being documented mechanically by means of display features of programs which maintain language data. Program writeups and instruction manuals are being prepared as individual programs become operational.

We will summarize the work accomplished toward each of these tasks.

4.1 Background Study

Approximately three months were spent exclusively in obtaining an orientation to previous work in the field. Our acquaintance with technical details of work done elsewhere was gained gradually over a much longer period, through attendance of conferences, wider coverage of literature, and personal conversations with other investigators.

The most recent survey of mechanical translation research, in 1959, was Delavenay's _La machine a traduire_ [1]. We consulted, as well, Bar-Hillel's _Report on the State of Machine Translation in the United States and Great Britain_ [2]; Booth, Brandwood and Cleave's _Mechanical Resolution of Linguistic Problems_ [3]; and descriptions of current work in the journal _Mechanical Translation_ [4], published at the Massachusetts Institute of Technology under the editorship of Yngve. With these and other sources, Delavenay's book provided an excellent introduction to the problems and achievements of research groups then in existence.

Following Bar-Hillel's advice, we concentrated our attention upon the work of the Georgetown group, since that group had been the most successful, in this country and abroad, in achieving translation by machine. The work of this group also exhibited the greatest diversity of approaches. We studied especially the Russian-to-English "SERNA System" described by Toma [5] and the "direct coding" technique explained by Brown in his "Manual for a Simulated Linguistic Computer" [6]. The latter had been applied in French-to-English translation, though its format was quite general. We attended demonstrations of these systems in Washington, and increased our understanding of them through numerous conversations with members of the Georgetown group, particularly its director, Leon Dostert.

The Georgetown experiments seemed to us to have been remarkably productive. The demonstrations, both of Russian translation and of French, were successful, at least to the extent that one would wish for a new science. But improvements were clearly necessary. What remained to be answered was whether the programs which had brought initial and limited success could be modified to secure translation of higher quality. This question troubled us for several reasons.

As needed improvements were made in the Georgetown algorithms, the computer routines had grown in size and complexity. Indeed, Brown's system had been designed to cope with mechanical problems of programs which would be expanded indefinitely by patching. We noted that:

(a) because there was an advantage in programming frequently occurring structures first, each modification became relatively more expensive in the total bulk of routines, and

(b) because the routines increased steadily in complexity, each modification became more difficult.

From the standpoint of linguistic description, both of these problems are unavoidable. Our concern was for the continued dependence which the Georgetown approach required between linguist and programmer. Since each linguistic modification called for resumption of program testing, the latitude of linguistic experimentation was continually limited by contingencies of program checkout. We felt that at some point the programmer would have to remove himself entirely from this cumbersome union, by providing the linguist with a suitable programming language or code.

Brown's "direct coding" technique had moved toward this goal, though it fell short of providing complete independence to the linguist. The Massachusetts Institute of Technology group, under the leadership of Yngve [7], had given more careful attention to this problem in developing the programming language COMIT. The two coding methods could not be easily compared with one another, however, because they were designed for different purposes.

In contrast to the Georgetown technique of gradually improving its translation algorithms through successive modifications, the M.I.T. group was committed to a detailed analysis of German and English as a prerequisite to any full-scale attempt to implement mechanical translation. Using Curme's Grammar of the German Language [8] as a basis, and extending its data through consultation with native informants, the group was trying to arrive at analyses which would satisfy all conditions and degrees of grammaticalness.

COMIT had accordingly been designed for scientific use in descriptive linguistic research, rather than for implementation of full-scale mechanical translation. By comparison, Brown's "direct coding" technique offered slight aid to the investigator. But, on the other hand, COMIT's usefulness in research had been achieved at the expense of running time on the computer. As a result, the programming language would be too costly in large applications.

We concluded that the Georgetown approach had overemphasized the pragmatic aspects of the problem at the expense of the scientific, while the M.I.T. group had done somewhat the opposite. Since it was apparent that a scientific foundation had not as yet been established for mechanical translation, we were sympathetic with the M.I.T. position. We did feel, however, that problems of linguistic theory and description could not be attacked adequately without a strong, and perhaps pragmatic, orientation in empirical research.

Our interest was drawn to the highly theoretical work of Bar-Hillel [9] at Hebrew University, Jerusalem, whose approach seemed even more remote from the empirical facts of language. Even though we were not then prepared to fully appreciate the value of Bar-Hillel's approach, it directed us toward a definite course. For it emphasized the fact that linguistic programming languages must serve a theoretical purpose, as well as an obvious empirical one, in the description of natural language.

If the linguist is to describe certain structural relations which he observes among units of some natural language, his programming language must contain either predicates or syntactic conventions which signify those relations. Juxtaposition of symbols in the programming language, for example, may signify the juxtaposition of units of the language being described, or a predicate denoting juxtaposition of the units may be used. In either case, the programming language has been so constructed that it can designate those relations of the natural language which the linguist desires to mention.

The *syntax* of a linguistic programming language is therefore determined by the structural theory to which the linguist subscribes.

We recalled that the close connection between philosophy and the syntax of scientific languages had been pointed out by Wittgenstein in his *Tractatus* [10], and that his thesis was later carried forward and elaborated by Carnap in *The Logical Syntax of Language* [11, p. 277]. In the latter, Carnap argues that the so-called philosophical problems concerning the foundations of individual sciences can be reduced to questions regarding the syntax of the languages to be used in scientific description. More succinctly, he says [11, p. 282]:

> ...the logic of science is nothing more than the language of science.
>
> ...problems of the current logic of science, as soon as they are exactly formulated, are seen to be syntactical problems.

Theoretical linguistics, as a consequence, has emerged out of concern with the syntax of scientific languages used for linguistic description, much as descriptive linguistics grew out of concern with natural languages. In seeking a theoretical foundation for mechanical translation, then, we are required to pay closer attention to the syntax of programming languages used by the linguist in recording language data.

We broadened our study, taking special notice of the formats employed by other groups in recording their data. The work of Rhodes [12] at the National Bureau of Standards drew our interest, as did theories developed along similar lines by Oettinger [13] and his group at Harvard. These investigators, together with Hays [14] and his colleagues Harper and Edmundson at the RAND Corporation, appeared to be making solid progress toward syntactic analysis techniques which were at least potentially general.

A question was immediately raised concerning the sense (or senses) in which a linguistic programming language may have a <u>general</u> syntax. It would be possible, for instance, for the linguist to choose as his programming language the assembly language of the particular computer upon which he happens to be working. Evidently the syntax of this language could accommodate the structural relations to which he subscribes, because it is capable of signifying, by suitable interpretation, any structural relations whatsoever.

Thus the syntax of a linguistic programming language may be general in the sense that it is capable of signifying the relations which occur in a wide range of structural theories.

The usefulness of COMIT in exploratory research derives primarily from this source. Brown's "direct coding" technique, by comparison, is somewhat restricted in its range of theoretical application. For what we desire in a programming language designed to implement translation is conditioned by the fact that we will implement only _one_ theory at a time. There is accordingly no need in the programming language for syntactic features which signify relations which occur in other theories. This surplus capability can only be wasteful in the resulting computer routines.

The desideratum of a programming language designed to implement mechanical translation is the capability to signify _only_ the structural relations which occur in the theory being implemented.

When this condition has been satisfied, the syntax of the programming language conveys maximal information about that particular theory. As we have noted elsewhere [15, pp. 87-88], our description of the theory is couched in whatever language we use to explain the syntax of the programming language, and not in the programming language itself. The former language is accordingly the vehicle of scientific explanation in theoretical linguistics, and the latter in descriptive linguistics.

A linguistic programming language may therefore have a general syntax in the sense that the latter is capable of signifying structural relations which occur in a wide range of natural languages.

Because important gains in efficiency may be obtained by restricting the range of theoretical application of a programming language, the fallacious inference is sometimes made that additional economies can be effected by restricting its range of descriptive application in mechanical translation to a certain language or pairs of languages. This opinion was widely accepted in 1959, though according to Delavenay [1, pp. 51-52] research groups in the Soviet Union were already considering the advantages of independent analysis and synthesis algorithms. The descriptive <u>ortho-language</u> used by Andreyev [16] and his colleagues, Ivanov and Melchuk, at Leningrad University greatly influenced our thinking on this point, as did their <u>intermediary language</u>, used for interlingual transfer. We had the opportunity to discuss the latter with Andreyev [17] during his visit to Cleveland in 1959; however, our knowledge of details at that time was at best fragmentary. In time, we nevertheless came to the following conclusion: <u>No</u> gain in operational efficiency results from restricting the range of description of a linguistic programming language to certain natural languages.

The basis of this important principle can be most readily comprehended through consideration of formalized descriptive languages. This will not hurt our argument, since we will also insist that the linguist make explicit his structural theory by formalizing his programming language. Secondly, we will assume that ambiguity should be eliminated from his programming language by techniques such as those described by Gorn [18] and Cantor [19]. For, although the natural languages which he will describe are notoriously ambiguous, there is no scientific advantage in describing ambiguity ambiguously. And, above all, we will distinguish the prescriptive and descriptive uses of formalization.

By _prescriptive formalization_ we mean those uses where a formalized language is actually being constructed for some specialized purpose by a procedure such as the following one described by Church in his _Introduction to Mathematical Logic_ [20, p. 48]:

> ...we begin by setting up, in abstraction from all considerations of meaning, the purely formal part of the language, so obtaining an uninterpreted calculus or _logistic system_. In detail this is done as follows.
>
> The vocabulary of the language is specified by listing the single symbols which are to be used. These are called the _primitive symbols_ of the language, and are to be regarded as indivisible in the double sense that (A) in setting up

> the language no use is made of any division of them into parts and (B) any finite linear sequence of primitive symbols can be regarded in only one way as such a sequence of primitive symbols. A finite linear sequence of primitive symbols is called a formula. And among the formulas, rules are given by which certain ones are designated as well-formed formulas (with the intention, roughly speaking, that only the well-formed formulas are to be regarded as being genuinely expressions of the language). Then certain among the well-formed formulas are laid down as axioms. And finally (primitive) rules of inference (or rules of procedure) are laid down, rules according to which, from appropriate well-formed formulas as premisses, a well-formed formula is immediately inferred as conclusion.

These features included by Church in his logistic system correspond to those which belong to the syntactical description of natural languages. The complete description of the formalized language is then obtained by adding semantical rules which provide the calculus with an interpretation through specification of the explicit objects or situations denoted by well-formed formulas. In practice this is usually accomplished by describing the denotata, or objects denoted by certain primitive symbols. The calculus may have many interpretations - indeed its usefulness for scientific explanation may depend upon this property - though there may be a principal or standard interpretation to guide our intuition. This convenience is not an essential part of formalization and it may in fact be detrimental. For example, it seems probable that the above terminology, which has its origin in the logical and mathematical interpretations of formalized languages, have been prejudicial against the use of formalization in linguistics.

It should be noted that the entire apparatus of proof is supplied by the syntactical description of the formalized language. As Bochenski points out in his recent book [21], Bolzano was a noteworthy precursor of this important discovery, but it was Frege who developed the notions of proof-theory with greater clarity than ever before. Hilbert led the way to applications of proof-theory in mathematics and Lukasiewicz in logic. The notion of a proof is explained by Church [20, p. 49] as follows:

> A finite sequence of one or more well-formed formulas is called a <u>proof</u> if each of the well-formed formulas in the sequence is an axiom or is immediately inferred from preceding well-formed formulas in the sequence by means of one of the rules of inference. A proof is called a proof of the last well-formed formula in the sequence, and <u>theorems</u> of the logistic system are those well-formed formulas of which proofs exist.

Evidently formalization involves certain operational commitments, called by Church [20, p. 50]:

> ...requirements of <u>effectiveness</u> as follows: (1) the specification of the primitive

> symbols shall be effective in the sense that there is a method by which, whenever a symbol is given, it can always be determined effectively whether or not it is one of the primitive symbols; (II) the definition of a well-formed formula shall be effective in the sense that there is a method by which, whenever a formula is given, it can always be determined effectively whether or not it is well-formed; (III) the specifications of the axioms shall be effective in the sense that there is a method by which, whenever a well-formed formula is given, it can always be determined effectively whether or not it is one of the axioms; (IV) the rules of inference, taken together, shall be effective in the strong sense that there is a method by which, whenever a proposed immediate inference is given of one well-formed formula as conclusion from others as premisses, it can always be determined effectively whether or not this proposed immediate inference is in accordance with the rules of inference.

The analogy of these requirements to similar concerns of descriptive linguistics should be obvious. Without elaboration we observe merely that they are requirements which must be satisfied by the language being used to describe the formalized language - not by the formalized language itself. The former is often called the <u>metalanguage</u>, the latter the <u>object language</u>. The meta-to-object language relationship is a relative one characterized by the fact that denotata of the metalanguage are features of the object language.

Prescriptive formalization accordingly focuses our attention upon the object language being designed and constructed as a specialized vehicle for scientific explanation.

The term descriptive formalization we have reserved for those contrasting uses where our purpose is to describe the structure of some one of the several thousand languages which already exist. Clearly in this case our interest is centered upon the metalanguage as the vehicle for scientific explanation.

The illusive distinction between prescription and description was resolved in logic and mathematics only after protracted debate. Because at base it merely reflects the specialized interests of theoretical and descriptive science which have already emerged in other fields, the consequences of its approaching resolution in linguistics can be predicted with reasonable confidence.

The linguist may accordingly resist this unfamiliar interpretation of his goal, preferring rather to call the framework of his syntactic description simply a grammar. But such disputes over terminology should not necessitate the rediscovery within linguistics of formal logic.

At least one methodological framework used widely in syntactics (i.e. context-free grammars) can be made explicit either as a mathematical model or as an uninterpreted calculus or logistic system. Although the former course was first taken in this study [22], the latter is more instructive. For it illustrates the various steps by which the linguist makes explicit his theoretical and descriptive assumptions. Through the procedure of <u>descriptive formalization</u> he constructs a formalized language whose axioms express the properties and relations which he wishes to describe in some historically given language. Thus, the methodological framework of the description is incorporated in the primitive symbols, formation rules, and rules of inference of the formalized language. The description itself is incorporated in its axioms and interpretation.

More important, when the syntactic description of a natural language has been couched in primitive theorems or axioms of a formalized language, operational problems of syntactic analysis and synthesis can be attacked in the frame of proof-theory. Syntactic synthesis may then be regarded as a deductive process by which empirically testable predictions of language behavior are generated as consequences of the axioms. Language behavior given empirically in some utterance of text may, similarly, be demonstrated to follow from the axioms through proofs constructed by the process of syntactic analysis.

Generalized methods of syntactic analysis and synthesis, as a consequence, are based on the well-known logical principle that inference processes proceed purely by the rules of inference, regardless of the particular axioms involved in the proof or of interpretations of the axioms. Programming criteria for syntactic analysis and synthesis may accordingly be derived from the rules of inference of the formalized language used for syntactic description, without reference to the particular axioms which describe individual languages, e.g., German or English.

Descriptive formalization, therefore, provides linguistics with an exact method for stating theoretical and descriptive hypotheses of language structure and with generalized operational techniques by which to evaluate the adequacy of such theories of inferring empirically testable consequences.

Further, as we set out to demonstrate, no gain in operational efficiency results when we restrict the range of descriptive application of the uninterpreted calculus, because programming criteria have been derived without reference to specific natural languages.

We will call each generalized algorithm which we derive from the uninterpreted calculus an <u>operational interpretation</u> of the calculus. The axioms and interpretation which describe some natural language will be referred to as a _linguistic_ _interpretation_. These terms will also be used to denote the research procedures by which such interpretations are obtained.

In summary, our postulates for general syntax in linguistic programming languages are as follows:

(a) The syntax incorporates the structural theory of the linguist in the sense that, when he has specified the syntax of his programming language, he has explained his theory.

(b) The syntax may be general in that it is capable of signifying relations which occur within a range of structural theories.

(c) General syntax having a wide range of theoretical application is useful in exploratory research but operationally expensive in the implementation of mechanical translation.

(d) Syntax designed to implement translation should be restricted to signify structural relations which occur in a particular theory, and no other.

(e) The syntax may be general in the second sense that it is applicable to the description of a wide range of languages.

(f) No gain in operational efficiency results from restricting the range of descriptive application of linguistic programming languages.

These principles led us to the conclusion that automatic programming techniques are feasible in handling language data. Although good initial progress had been made by other groups toward implementation of mechanical translation, more sophisticated compilers were needed to minimize the restrictive interdependence between linguist and programmer. Moreover, without the assistance of a general systems approach in translation research, it seemed doubtful that the complex patterns of natural language could be described within a satisfactory time-period, or that description of evolutionary changes in language occasioned by a rapidly advancing technology could be kept up to date.

4.2 Theoretical Study

Our basic methodology distinguishes eight phases in linguistic research:

(phase 1) formulation of descriptive theory, i.e., induction toward a new methodological framework for linguistic description.

(phase 2) descriptive formalization, i.e., explication of the methodological framework (through specification of primitive symbols, formation rules, and rules of inference of an uninterpreted calculus), (phase 3) operational interpretation, i.e., derivation of programming criteria and rationale for generalized analysis and synthesis algorithms (from the rules of inference of the calculus),

(phase 4) linguistic interpretation, i.e., description of one or more natural languages (through specification of appropriate axioms, and interpretation of the calculus),

(phase 5) linguistic verification, i.e., evaluation of the adequacy of each formalized language which has resulted from linguistic interpretation (through empirical experimentation with analysis and synthesis of language behavior),

(phase 6) operational verification, i.e., evaluation of the criteria and rationale underlying generalized analysis and synthesis algorithms (through operational testing with the various languages),

(phase 7) verification of descriptive formalization, i.e., evaluation of the adequacy of the calculus itself (through consideration of the collective results of empirical experimentation with the various languages),

(phase 8) verification of descriptive theory, i.e., evaluation of the methodological framework of linguistic description (through consideration of specific points of failure in all phases of the research).

The last phase, therefore, leads back to the first, just as all phases intricately interlace with the ones preceding.

In approaching the formulation of descriptive theory (phase 1) we came to the general conclusion that our methodological framework should include semantics. Because of the narrow frame of reference which has come into use in linguistics, especially since publication of Bloomfield's Language [23], one must look elsewhere for subject matter of descriptive semantics. A broader frame of reference has emerged from the tradition sustained through the Greek Sophists and the Hellenistic philosophies as a whole, through the scientia sermocinalis of medieval Europe, diverging among the formalists following Leibniz, British empiricists, and American pragmatists, and culminating finally in the writings of Peirce, Mead, and Morris within the field of semiotic.

In Morris' terminology [24], semiotic, the study of sign-processes, has three subdisciplines: syntactics, semantics, and pragmatics. The sign-process itself, semiosis, is described as [24, p. 81]:

> involving three (or four) factors: that which acts as a sign, that which the sign

refers to, and that effect on some interpreter in virtue of which the thing in question is a sign to that interpreter. These three components in semiosis may be called, respectively, the sign vehicle, the designatum, and the interpretant; the interpreter may be included as a fourth factor.

Syntactics is characterized by Morris as concerned with "the formal relation of one sign to another," semantics with "the relation of signs to the objects to which the signs are applicable," and pragmatics with "the relation of signs to interpreters." Linguistics may be subsumed under semiotics as the study of sign-processes in language.

Within this conceptual framework, it seems plausible that, in the behavioral process by which linguistic utterances are recognized, the interpreter must take account of sign vehicles before recognizing what the sign vehicles designate, and must take account of these designata before recognizing their interpretants. This observation is concerned with the order, not the status, of cognition. In other words, any attempt to simulate semiosis in language would require three separate subprocesses: the first to describe the sign vehicles of the utterance given for analysis, the second to describe the designata of the sign vehicles, and the last to describe the interpretants of the designata.

From these and other considerations, with Lamb [25], we concluded that semiotic description must be hierarchical. That is, descriptive formalization (phase 2) would require the construction of at least three formalized languages, and these would be arranged in a hierarchy, with symbols of the first formalized language denoting segmental units of the natural language, symbols of the second denoting units of the first, and so on.

The terms object language and metalanguage may be appropriately employed within such a hierarchy to distinguish, respectively, the language being described and the language in which the description is being given. But, as we have mentioned, these terms are relative, and would, for example, permit the second of the formalized languages to be both the metalanguage of the first and the object language of the third. Thus we will apply the term object language only to the natural language, and will refer to the three formalized languages as the syntactic, semantic, and pragmatic metalanguages, respectively.

4.2.1 Syntactics

The syntactic metalanguage upon which we based our studies may be sketched as a context-free [26] or Backus [27] system. It consists of:

(i) the primitive vocabulary of:

__constants__:

(1) $a_1, a_2, \ldots, a_p$

__predicates__:

(2) $P_1, P_2, \ldots, P_q$

classified as either __terminal__ or __non-terminal__ predicates, and connectives:

(3) $\varepsilon, \rightarrow$

(ii) the formation rules for:

__words__:

(4) a single constant is a word,

(5) if each of α and β is a word, then $\alpha\beta$ is a word,

__phrases__:

(6) a single predicate is a phrase,

(7) if each of Φ and Ψ is a phrase, then $\Phi\Psi$ is a phrase,

__sentences__:

(8) if α is a word and P_i a predicate, then $\alpha\varepsilon P_i$ is a sentence,

(9) if Φ is a phrase and P_j a non-terminal predicate, then $\Phi \rightarrow P_j$ is a sentence,

(iii) the primitive theorems or axioms, of two forms:

<u>rules</u>:

(10) $P_{i_1} P_{i_2} \ldots P_{i_n} \rightarrow P_j$

as sentences of the form (9), and

<u>codes</u>:

(11) $a_{j_1} a_{j_2} \ldots a_{j_m} \; \varepsilon \; P_i$

as sentences of the form (8), but with P_i a terminal predicate.

(iv) the rule of inference:

(12) if $\vdash \alpha_1 \varepsilon P_{i_1}, \alpha_2 \varepsilon P_{i_2}, \ldots, \alpha_n \varepsilon P_{i_n}$

and $\vdash P_{i_1} P_{i_2} \ldots P_{i_n} \rightarrow P_j$

then $\vdash \alpha_1 \alpha_2 \ldots \alpha_n \varepsilon P_j$

where $\vdash$ signifies that the sentence asserts a theorem,

and (v) the interpretation:

(13) each constant denotes an individual, primitive unit of the object language,

(14) if each of α and β is a word, then $\alpha\beta$ denotes the object language unit compounded of the unit denoted by α, immediately followed by the unit denoted by β.

(15) (The predicates denote syntactic properties of object language units, so that the sentence $\alpha \varepsilon P_j$ would be interpreted to state that the object

language unit α has the syntactic property P_j.)

The primitive units of the object language will be thought of as either phones in the spoken languages, or graphs in the written. Terminal predicates will denote elementary properties of syntactic distribution. The non-terminal predicates will denote defined properties of syntactic distribution. The codes will accordingly be referred to as syntactic codes, and the rules as syntactic rules.

All of the information necessary for operational interpretation (phase 3) has now been given. One may deduce programming criteria of generalized syntactic analysis and synthesis algorithms for the above system, even though no particular object language has as yet been described.

The syntax of any number of object languages may be investigated through linguistic interpretation (phase 4), i.e., research procedures oriented to the specific goal of discovering an appropriate set of syntactic codes, and syntactic rules for each language. We undertook such descriptive studies of English and German, using at first essentially the techniques of segmentation and classification of Harris [28], and later also those of Pike [29].

The procedures which we have called linguistic verification (phase 5) were carried out manually by constructing some of the proofs which would result from syntactic analysis of particular German and English texts [30], and, to a lesser extent, by random synthesis of German and English utterances. There were soon indications that these principles of verification alone would not be adequate for our purpose. For we had set out to design a syntactic metalanguage which would be capable of describing the _sign_ _vehicles_ of a given utterance, as well as the utterance itself. It was not clear how this could be accomplished if proofs were to be the only output of syntactic analysis.

One possible explanation was that the sign vehicles might occur as segmental units of the proofs, e.g., as individual rules or as _subproofs_ involving certain sequences of rules. It seemed reasonable that sign vehicles, the units of information which could be recognized in an utterance, should not differ substantially from the units of information needed to produce that utterance. The sign-system of the object language, therefore, might be investigated through structural analysis of _syntactic descriptions_ of utterances, much as the utterances themselves had been analyzed structurally in making the descriptions.

4.2.2 Semantics

An appealing aspect of this approach was that it might lead to a mechanism for selective production of utterances through controls exercised on syntactic rules. The other side of content analysis is of course content synthesis. As Hjelmslev [31, p. 55] has implied, the control which <u>content</u> exercises over <u>expression</u> in production of utterances cannot readily be separated from the control which <u>expression</u> exercises over <u>content</u> in recognition of utterances:

> Their functional definition provides no justification for calling one, and not the other, of these entities <u>expression</u>, or one, and not the other, <u>content</u>. They are defined only by their mutual solidarity, and neither of them can be identified otherwise.

To evaluate these conjectures, we studied methods for segmenting and classifying syntactic descriptions of utterances, taking <u>as</u> our corpus the proofs which had been produced in our previous experiment. In more familiar terms, we worked with the tree diagrams which are commonly used to represent syntactic descriptions of utterances. Roughly speaking, we parsed the trees into subtrees, and thus effectively the proofs into subproofs.

These experiments quickly demonstrated that the syntactic metalanguage lacked an essential feature: the capability to name points of juxtaposition of subtrees. Because segmental units of utterances have just one point of initiation and one of termination, this problem does not arise in syntactic description; the concatenation of object language units may be signified in the syntactic metalanguage simply by concatenation of constants. The segmental units of syntactic description likewise have just one point of initiation, but may have many terminal points, represented as branches of the subtrees. The terminal points, as a consequence, must be named in the syntactic description if they are to be mentioned in the semantic. We found that this problem could not be avoided by means of the purely positional (parenthesis-free) notation of simple connective languages, such as those described by Rosenbloom [32, p. 152f].

Through operational verification (phase 6), therefore, we had proceeded to verification of the descriptive formalization of syntax (phase 7) and found it defective.

A second difficulty arose in finding a suitable methodology for classification of subtrees resulting from the above segmental analysis. Here, we are indebted to Joos [33] for his work on semology and to Martin [34] for his observation that the semantic metalanguage need not describe directly the relation between expressions and the objects which they denote. As an alternative, the semantic description may concern itself entirely with relations among expressions. Hence the relations are, strictly speaking, syntactic; yet they are based on semantical relations between expressions and denoted objects. For example, taking <u>multiple denotation</u> as a semantical primitive, he defines the relation of comprehension between two expressions as follows [34, p. 104]:

> Let us say that an expression a of L <u>comprehends</u> an expression b of L if and only if a and b are both predicate constants and a denotes every object which b does.

The converse relation of comprehension he calls <u>subsumption.</u> His semantic metalanguage may then be based on either one of these relations. He says [34, p. 179f]:

> If we then introduce no further primitives, we gain a semantical metalanguage which speaks exclusively of the expressions of the object-language and not of its objects, because the relata of the comprehension and subsumption relations are exclusively expressions. In taking such a purely linguistic relation as a primitive we avoid the necessity of augmenting the meta-language by incorporating the object-language within it. The attendant gain in economy and simplicity is of course enormous.

Returning then, to our problem of classifying syntactic subtrees, we concluded that the criterion for considering one sign vehicle a substitute for another should be the existence of a common designatum, and that distribution classes of sign vehicles should correlate with certain sets of designata. Working on the assumption that segmental units resulting from our discovery procedure were descriptions of sign vehicles, we attempted to apply these principles in classifying subtrees.

We noted that our method for segmenting syntactic descriptions is formational in the sense that, like the syntactic description itself, it rests upon empirical evidence that certain utterances exist in the object language. In contrast, the method for classifying syntactic descriptions is transformational; it rests on evidence that native speakers of the object language can recognize or produce certain relations among utterances, e.g., through paraphrasing. We found, in keeping with Hjelmslev's prediction, that these two sources of information must be considered conjointly in the operations of semantic segmentation and classification.

Structural relations involved in the method of semantic analysis can now be made more precise. We will first make the various alterations which were found necessary in the formalization of syntax:

To the primitive vocabulary (i) add the infinite lists of unary connectives:

(16) 1, 2, 3, ...

called <u>superscripts</u>.

In the rules of formation (ii) replace (6) and (7) with these specifications for the production of phrases. Each phrase will now have a <u>degree</u>, signified by either zero or a positive integer:

(17) a single terminal predicate is a phrase of degree zero.

(18) if each of Φ and Ψ is a phrase of degree zero, then $\Phi\Psi$ is a phrase of degree zero,

(19) if P_j is a non-terminal predicate, then P_j^{1} is a phrase of degree 1,

(20) if P_j is a non-terminal predicate, ^{n} a superscript, and either Ψ or $\Phi_1\Phi_2$ a phrase of degree n-1, then any one of $P_j^{n}\Psi, \Phi_1 P_j^{n}\Phi_2$ or ΨP_j^{n} is a phrase of degree n.

An operation on phrases will also be needed. The operation has been described more carefully by Senechalle [35] than will be possible here. It may be imagined intuitively as the substitution of one entire phrase for a single, non-terminal predicate in another. Following the substitution, one must specify a way to compute the superscripts in the resulting phrase from those given in the operands. The binary connectives:

(21) $\overset{\frown}{1}, \overset{\frown}{2}, \overset{\frown}{3}, \ldots$

are used informally in the following explanation:

(22) If Φ and Ψ are phrases of degree m and n, respectively, and $P_i{}^k$ occurs in Φ, then $\Phi \overset{\frown}{k} \Psi$ is the phrase of degree m+n-1 obtained by substituting Ψ for $P_i{}^k$ in Φ and incrementing superscripts as follows: add k-1 to those in Ψ, and add n-1 to those remaining in Φ which are greater than k.

For example, if Φ and Ψ are the phrases $P_{i_1}{}^1 P_{i_2}{}^3 P_{i_3}{}^2 P_{i_4} P_{i_5}{}^4$ and $P_{j_1}{}^2 P_{j_2}{}^1 P_{j_3}$, respectively, then $\Phi \overset{\frown}{2} \Psi$ is the phrase $P_{i_1}{}^1 P_{i_2}{}^4 P_{j_1}{}^3 P_{j_2}{}^2 P_{j_3} P_{i_4} P_{i_5}{}^5$. The predicates without superscripts are of course terminal.

A second rule of inference will be added to (iv):

(23) if $\vdash \Phi \rightarrow P_j$, with $P_i{}^k$ occurring in Φ, and $\vdash \Psi \rightarrow P_i$, then $\vdash \Phi \overset{\frown}{k} \Psi \rightarrow P_j$,

and (12) will be modified so that all predicates occurring in the premises, except P_j, are terminal; it will read:

(24) if $\vdash \alpha_1 \varepsilon P_{i_1}, \alpha_2 \varepsilon P_{i_2}, \ldots, \alpha_n \varepsilon P_{i_n}$, with $P_{i_1}, P_{i_2}, \ldots, P_{i_n}$ terminal predicates, and $\vdash P_{i_1} P_{i_2} \ldots P_{i_n} \rightarrow P_j$, then $\vdash \alpha_1 \alpha_2 \ldots \alpha_n \varepsilon P_j$.

As a result of the last two alterations, the calculus will have derived theorems of the form (9) as well as of the form (8). A device will thus be provided for selective processes of syntactic synthesis, i.e., processes limited to syntactic theorems mentioned by the semantic metalanguage. More exactly, the collection of syntactic theorems of the form (9), which we will call a formation structure [22,35], will be (with the null string) the objects denoted by words of the semantic metalanguage.

The semantic metalanguage itself will be characterized as a second-stratum, context-free system. (In deference to Lamb, and because of unfortunate connotations in logic, we will abandon the term second-order system, used in earlier reports). In our outline, we will attempt to illustrate that there is a fundamental similarity between the two metalanguages, even though the semantic requires a more complex concept of juxtaposition among segmental units.

The semantic metalanguage consists of:

(i) the primitive vocabulary of:

constants:

(25) $b_1, b_2, \ldots, b_n$

each having a degree, signified by either zero or a positive integer,

predicates:

(26) $Q_1, Q_2, \ldots, Q_v$

each a one-place predicate, yet differentiated by a non-negative degree in the sense of being applicable to constants of a certain degree, and <u>connectives</u>:

(27) ε, $\rightarrow$, $\widehat{1}$, $\widehat{2}$, $\widehat{3}$, ...

(ii) the formation rules, for:

<u>words</u>:

(28) a single constant of degree n is a word of degree n,

(29) if α and β are words of degrees m and n, respectively, and $i \leq m$, then $\alpha \widehat{i} \beta$ is a word of degree m+n-1.

<u>phrases</u>:

(30) a single predicate of degree n is a phrase of degree n,

(31) if Φ and Ψ are phrases of degrees m and n, respectively, and $i \leq m$, then $\Phi \widehat{i} \Psi$ is a phrase of degree m+n-1.

<u>sentences</u>:

(32) if α is a word and Q_i a predicate, both having degree n, then $\alpha \varepsilon Q_i$ is a sentence of degree n,

(33) if Φ is a phrase and Q_j a non-terminal predicate, both having degree n, then $\Phi \rightarrow Q_j$ is a sentence of degree n,

(iii) the primitive theorems or axioms, either:

<u>rules</u>:

(34) $Q_{j_1} \widehat{i_1} Q_{j_2} \widehat{i_2} \ldots \widehat{i_{n-1}} Q_n \rightarrow Q_k$

each a sentence of the form (33), or

codes:

(35) $$b_{j_1} b_{j_2} \ldots b_{j_m} \varepsilon Q_i$$

each a sentence of the form (32), but with Q_1 a terminal predicate,

(iv) the rule of inference:

(36) if $\vdash \alpha_1 \varepsilon Q_{j_1}, \alpha_2 \varepsilon Q_{j_2}, \ldots, \alpha_n \varepsilon Q_{j_n}$

and $\vdash Q_{j_1} {}^{\widehat{i}_1} Q_{j_2} {}^{\widehat{i}_2} \ldots {}^{\widehat{i}_{n-1}} Q_{j_n} \rightarrow Q_k$,

then $\vdash \alpha_1 \alpha_2 \ldots \alpha_n \; \varepsilon \; Q_k$,

and (v) the interpretation:

(37) each constant of degree n denotes a syntactic rule of degree of n, where the degree of the syntactic rule is given by the number of occurrences of non-terminal predicates in its phrase,

(38) if α and β are words denoting the syntactic theorems $\Phi \rightarrow P_j$ and $\Psi \rightarrow P_i$, respectively, then $\alpha {}^{\widehat{k}} \beta$ denotes the theorem $\Phi {}^{\widehat{k}} \Psi \rightarrow P_j$, provided that the latter follows from the former as premises, or else $\alpha {}^{\widehat{k}} \beta$ denotes the null string,

(39) if α and β are words and either denotes the null string, then $\alpha {}^{\widehat{i}} \beta$ denotes the null string.

(40) (The predicates denote syntactic properties of syntactic theorems, and thus indirectly semantic properties of object language units.)

The primitive units denoted by semantic constants will be thought of as descriptions of <u>morphs</u>. Terminal predicates will again denote elementary properties of distribution, in this instance semantic properties. The non-terminal predicates will denote other distributional properties of descriptions of sign vehicles, and, in consequence of our hypothesis, properties of semantic distribution. Codes and rules will be called <u>semantic codes</u> and <u>semantic rules</u>, respectively.

The concept of morphological description which resulted from our studies is novel, though it requires very little reorientation from traditional views. We have therefore illustrated our full range of research procedures, reaching a new methodological framework through verification of descriptive theory (phase 8).

In our outline we have tried to exhibit only the gross structural relationship between elements of syntactical and semantical description. Several refinements will be necessary before the semantic metalanguage can become even an approximate instrument for descriptive research. Instead of adding these details to our already tedious explanation, we will discuss them informally.

The first problem which we encountered in using the metalanguage was the abundance of semantically equivalent expressions, not only in the object language, but in the metalanguage itself. One can show rather easily that particular pairs of the semantic words will denote the same theorems of the syntactic metalanguage. As far as we have been able to ascertain, these redundancies in the semantic description contribute only to complexity. We assume, therefore, that an efficient semantic metalanguage should be constructed so that the linguist would be able to _mention_ equivalent expressions but not _use_ them.

Redundancies of this kind may be avoided by restricting the semantic predicates in phrases to a standard order, the order which would result if the sequence of integers $i_1, i_2, \ldots, i_{n-1}$ in (34) was non-decreasing. Our ordering convention generalizes the concept of a push-down store, such as the one used by Yngve [36] in his model for syntactic synthesis. In Yngve's operational interpretation, substitution is always performed at the left-most, non-terminal predicate (of the derived premise); the names of substitution points are thus given positionally from left to right. The convention which we have chosen requires the substitution to take place at the non-terminal predicate with _1_ as its superscript, regardless of the position in the premise. (The same ordering convention is used throughout the hierarchy of formalized languages.)

Our second refinement was more fundamental, since its concern was the naming of substitution points in the semantic metalanguage itself. Superscripts may be used for this purpose, just as in the syntactic language. The necessary alterations are somewhat more complicated; still they parallel the ones above in every important detail. This extension of the language was of course a prerequisite to formalization of the pragmatic metalanguage.

4.2.3 Pragmatics

The pragmatic metalanguage which resulted from our hypothesis is isomorphic to the semantic in its primitive symbols and rules of formation and transformation, though it has of course a different linguistic interpretation. Needless to say, we understand very little about the application of those structural concepts in descriptive pragmatic research. The pragmatic hypothesis is considered an important long-range objective, nevertheless, and we have labored to make it explicit.

Speculative arguments, paralleling those already given for semantics, may be brought to bear on operations of pragmatic segmentation and classification. The segmental units in this would be semantic theorems, with substitution requiring the existence of a common interpretant. We anticipate that the units which Lamb calls <u>sememes</u> [25] will be described in our hierarchy by terminal pragmatic predicates. A model for symbolic uses of language, e.g., for metaphor and metonymy, may ultimately be obtained at this stratum.

A matter of more immediate concern in descriptive semantics is a formal way to recognize and produce repetitions of information content occurring within individual utterances of the object language. Sapir's [37, p. 88] analysis of concepts expressed by the simple sentence, <u>the farmer kills the duckling</u>, for example, isolates thirteen conceptual units:

I. Concrete Concepts:

1. First subject of discourse: <u>farmer</u>
2. Second subject of discourse: <u>duckling</u>
3. Activity: <u>kill</u>

_________ analyzable into:

A. Radical Concepts:

1. Verb: (<u>to</u>) <u>farm</u>
2. Noun: <u>duck</u>
3. Verb: <u>kill</u>

B. Derivational Concepts:

1. Agentive: expressed by suffix <u>-er</u>
2. Diminutive: expressed by suffix <u>-ling</u>

II. Relational Concepts:

Reference:

1. Definiteness of reference to first subject of discourse: expressed by first <u>the</u>, which has preposed position

2. Definiteness of reference to second subject of discourse: expressed by second _the_, which has preposed position

Modality:

3. Declarative: expressed by sequence of "subject" plus verb: and implied by suffixed _-s_

Personal relations:

4. Subjectivity of _farmer_: expressed by position of _farmer_ before kills; and by suffixed _-s_

5. Objectivity of _duckling_: expressed by position of _duckling_ after _kills_

Number:

6. Singularity of the first subject of discourse: expressed by lack of plural suffix in _farmer_: and by suffix _-s_ following verb

7. Singularity of second subject of discourse: expressed by lack of plural suffix in _duckling_

Time:

8. Present: expressed by lack of preterit suffix in verb; and by suffixed _-s_

Even this exhaustive analysis of potential designata fails to deal adequately with the predicate kills, for it neglects the information given by points of semantic substitution. As Quine [38, p. 140] has pointed out, we must distinguish between the predicate kills and the predicate schema ① kills ②. By so doing, we may recognize that ① kills ② and ① murders ② have one set of common designata, while ① kills ① and ① commits suicide have another.

As we have already mentioned, the information about substitution places is given in the semantic metalanguage by the superscripts of semantic rules. Clearly, then, our formalization must be extended to permit superscripts to occur more than once in the same rule. At the same time, we must restrict semantic substitutions which will be made at the various occurrences of a particular superscript so that expressions with common content will be produced at those positions. The schema ① kills ①, for instance, will permit the farmer kills himself, he kills himself, or even the farmer kills the farmer, but not the farmer kills the duckling.

An operational test for common content will therefore be needed in generalized semantic analysis and synthesis algorithms. The scheme which we are investigating is based on theoretical arguments that the linguistic interpretation can be made to satisfy these conditions: (a) two expressions will have equivalent information content if they have the same structure of non-terminal predicates in the semantic tree diagram, and (b) the information content of one expression will include that of another if the semantic diagram of the first is a subtree of the diagram of the second. If these conditions can be met in semantic description, then any collection of expressions whose semantic diagrams are subtrees of some common diagram will have common information content.

It should be noted that only the common diagram would then have to be stored in an information system, or translated in a translation system. But these principles have been studied only with simple examples; an enormous amount of work will be required to explore them adequately in various languages. Moreover, structural patterns of this type should be expected to occur in the pragmatic stratum, and perhaps in syntax.

4.2.4 Stochastic Description

The last metalanguage extension which we will discuss is probably the most important, because it reminds us that the subject matter of descriptive linguistics is, after all, human behavior. This fact alone should alert those who believe that a simple key will unlock the storehouse of language. If such a key exists, it is most likely to be found in the neglected distinction between alternative behavioral units and the choices which are made among them.

Thus far our metalanguages have been designed to describe behavioral units involved in recognition or production of language. The rules and codes which make up our descriptions are linked together by the rules of inference into a single labyrinth of alternatives. No attempt has been made to describe the hearer's or speaker's preference for one path or another.

Like other investigators who are attempting to deal with language in its full complexity, we have found empirically given utterances to be remarkably ambiguous, both syntactically and semantically. Pragmatic ambiguities will undoubtedly present themselves in good time. It should be obvious that <u>descriptive</u> linguistics will not be permitted to attack this problem by reducing the number of behavioral alternatives. Even if we were inclined to do so for so-called practical reasons, the result would be ultimately impractical. For we cannot <u>prescribe</u> the form of linguistic behavior which will be presented to automatic processes of analysis, or at least not with greater success than we can prescribe other forms of human behavior.

We have accordingly begun to treat linguistic analysis and synthesis as stochastic processes, so that we can preserve the full range of behavioral alternatives through improving our understanding of the choices among them. The stochastic model which we are studying was originally suggested by Solomonoff [39]. It describes the inferential steps in the construction of proofs as stochastically independent events, so that the conditional probability $P(B|A)$ that rule B will follow rule A in a proof is given by the product:

$$P(B|A) = P(A) \cdot P(B)$$

This seems a reasonable working hypothesis, since, unlike Shannon and Weaver [40, p. 7f], we are not interested in conveying information about linguistic structure in the probabilities. Structural information is described by the rules themselves, and in the codes.

The stochastic theory may be incorporated in the syntax of our syntactic, semantic and pragmatic metalanguages by replacing the two-valued concept of truth implicit in the rule of inference (23), and its correlates in the other languages, by a multi-valued concept. Probabilities $P(A)$ and $P(B)$ are then regarded as the truth values of the premises and $P(B|A)$ the value of the conclusion. Any assignment of probabilities may therefore be given to the axioms, provided that the values sum to 1 over all rules $\phi \rightarrow P_j$ or codes $a_i \varepsilon P_j$ having the same predicate P_j as the left-most symbol. The truth value of the conclusion of

(24), or its semantic or pragmatic correlate, is also taken to be the product of the values of the codes and single rule acting as premises.

The distinction which we have made between the description of structure and the description of choice has been discussed by Newell, Shaw and Simon [41, p. 15] in their study of problem solving processes in large mazes. As they indicate, the task of discovering solutions in a sufficiently small maze is trivial. The difficulties in complex problem solving arise from a combination of two factors: the size of the set of possible solutions which must be searched and the task of ascertaining whether a proposed solution actually satisfies the problem.

Researchers in mechanical translation have assumed almost universally that the size of the maze to be searched (i.e., the set of structural alternatives to be considered in analysis and synthesis) would be manageable. As this confidence was eroded by empirical data, they sought to deal with the problem by arbitrarily limiting the size of the maze. Restricting descriptions of structural alternatives is clearly not a solution to the problem, however, since it permanently cripples the process in many areas of language behavior.

We have therefore begun to distinguish between the processes of analysis and synthesis, by which the well-defined alternatives of the maze are discovered, and the heuristic choice processes, by which portions of the maze are explored for possible solutions. Following Newell, Simon and Shaw, we subclassify the choice processes as either solution-generating or solution-verifying processes. The former determine the order in which paths in the maze shall be explored; the latter whether a proposed solution is in fact a solution. Hence, in stochastic analysis or synthesis, the computation of probabilities for alternatives is the function of the solution generator.

The success of the solution verifier depends upon its ability to select a very small part of the maze, in finding a solution. Heuristic processes with this capability will be fundamental to efficient translation.

The manipulation of structural alternatives and choices among them has also been investigated by Ashby [42, p. 71], who makes a similar distinction between variables and parameters in cybernetic systems. We distinguish two basic types of self-organizing linguistic systems: (a) adaptive systems, in which the description of choice is manipulated dynamically, and (b) learning systems, in which manipulations of the structural description take place. Phenomena of style and dialect will certainly fall within the former category. Indeed, there appears to be no theoretical reason why all language should not be considered a single system, with language or dialect choices corresponding to separate adaptations; it is difficult to see how else we could account for the bilingual and multilingual speaker. The latter category, learning systems, will be essential in more sophisticated applications; for example, automatic processing of scientific material containing definitions or new concepts developed in the text. Knowledge of these phenomena is so meager that we can hardly appreciate our opportunities.

4.2.5 Interlingual Description

Programming languages for interlingual description may likewise be formalized, so that interlingual relations presumed to occur among two or more languages may be made explicit. The basis of formalization will not be given, because it should be obvious from the form of the transfer rules which we will discuss.

Various interlingual relations were considered during the study. The most obvious of these are, of course, relations among object-language units. Other possibilities are relations among units of the formalized languages used to describe syntactic, semantic or pragmatic properties of different object languages. Relations among single constants were considered, and those among predicates, words, or phrases occurring in proofs. A great deal of time was spent in studying relations among axioms of the different languages, and finally among theorems. The last was the unit chosen for implementation.

Three interlingual metalanguages are in fact being implemented. Since the function of transfer rules can be more easily comprehended in the context of transfer maintenance programming, they will be discussed in the next section.

4.3 Programming Study

The amount of labor involved in actually undertaking the description of a natural language must be experienced to be appreciated. Two poles of opinion have therefore developed about linguistic applications such as mechanical translation: (a) that the problem is trivial, and (b) that a solution is impossible. Both of these opinions are suspect since they rationalize the avoidance of the aforementioned labor, with obvious consequences. Fortunately, a more promising, though less comfortable, view has emerged in recent years: that the labor of descriptive research may be reduced through automated procedures.

We have been primarily concerned with following out the consequences of this last thesis. The procedures which we have called linguistic interpretation (phase 4) lend themselves to automatic handling, as do the more familiar procedures of linguistic verification (phase 5).

All of the procedures which we have investigated are generalized. Without the general programming techniques which result from descriptive formalization, relatively inconsequential aids could be offered the linguist. There are two major reasons for this, one operational and the other economic: the important operational advantage of generalized procedures is the reduction of restrictive interdependence between linguist and programmer. The economic advantage stems from the fact that developmental costs of computer programs may be distributed over many languages.

We distinguish two major categories of linguistic algorithms. Those which carry out procedures related to linguistic interpretation, we call language data processing algorithms. Procedures of this type may be characterized as supporting induction, since they help the linguist while he is formulating and manipulating theoretical constructs. Those of linguistic verification, on the other hand, support deduction, either by carrying out the predictive function of descriptive theories, or by producing demonstrations that such theories cover empirical data at hand. These we call linguistic information processing algorithms.

A block diagram of the computer system which has resulted from our hypothesis is provided on the last page of this report. The system consists of three major sections: the first comprised of language data processing algorithms, the second of information processing algorithms, and the third of executive algorithms which exercise general control over all processing sequences. This underlying Control section cannot be conveniently shown in the diagram, though it represents a large programming investment.

Elements of the Linguistic Information Processing section are located horizontally along the center of the diagram. Language data processing algorithms, and the data tapes which they provide for information processing algorithms, are grouped at the top and bottom of the diagram. The data tapes are represented by circles, the algorithms by boxes. Heavy lines indicate the algorithms which have been programmed and are in operational status.

Capabilities of the information processing algorithms may be grasped intuitively through knowledge of the data formats upon which they operate. Hence, we will sketch the essential functions of three major parts of the Language Data Processing section, and that of a fourth but minor part. The former are called <u>corpus</u>, <u>grammar</u>, and <u>transfer</u> maintenance, and the latter <u>request</u> maintenance.

4.3.1 Request Maintenance

All data formats used by the linguist in making descriptions are called <u>requests</u>. The Request Maintenance program is designed to check request formats before they are introduced into the system. In addition, it carries out the revision of errors on request tapes. Descriptive data can accordingly be accumulated on request tapes in batches large enough to take full advantage of the system's data processing capacity.

4.3.2 Corpus Maintenance

Corpus Maintenance is comprised of programs which carry out the linguist's requests for the revision or display of texts, or for conversion of selected samples of texts into the formats required as input to Monolingual Recognition. About 850,000 words of English, 750,000 of German, 3,000 of Chinese, and 30,000 of Russian are currently being maintained by these programs.

4.3.3 Grammar Maintenance

Grammar Maintenance programs are much more intricate, because they process the linguist's requests related to linguistic descriptions. At present, structural descriptions may be revised in either the syntactic or semantic metalanguages (Rule Revision). For his convenience, the linguist may use mnemonic symbols to represent syntactic or semantic predicates. Constants are represented numerically. The programs execute checking procedures which insure that the formation rules of the metalanguages have not been violated. Codes or rules which are written incorrectly are rejected by the maintenance programs, and messages explaining the reason for the rejection are displayed for the linguist. The interpretation of each metalanguage is also verified by thorough accounting procedures which insure that all constants have denotata. The probability associated with each rule is computed automatically from a numerical weight assigned by the linguist (Probability Revision). Next, the descriptions are selected for conversion into the data formats required by either monolingual recognition or production. The former conversion (Input Grammar Selection) combines the individual rules into a very compact, single branch-structure within which the nodes correspond to decision-points in the appropriate analysis process; duplicate rules are called to the attention of the linguist during the conversion procedure. The latter conversion (Output Grammar Selection) has not as yet been programmed. Finally, displays of rules or codes are provided in three different sort sequences which will expedite studies of analysis or synthesis, or of interpretation in the descriptive hierarchy (Grammar Display).

Approximately 5,400 syntactic rules of English are currently being maintained in the system, plus 2,800 of German, 2,400 of Chinese, and 2,200 of Russian. There are, in addition, about 3,600 rules for each of Chinese and Russian which record the information usually given by dictionary entries. Another 70,000 rules of this kind remain to be processed for English, and 45,000 for Russian. These totals reflect the status of descriptive research, not the complexity of the various languages.

More accurately, Grammar Maintenance programs process data formats of three types: (a) lexical data, consisting of the correspondences between lexical segments of the object language and syntactic constants, (b) syntactic data, consisting of the syntactic codes and correspondences between syntactic rules and semantic constants, and (c) semantic data, consisting of the semantic codes and correspondences between semantic rules and pragmatic constants. In each case, probabilities are associated with the rules.

Our failure to appreciate problems of representation at an early stage of the research led to a restriction on terminal predicates in the present programs. Each terminal predicate now applies to only one constant. This fault is being remedied.

4.3.4 Monolingual Recognition

Monolingual Recognition consists of three analysis algorithms which use these grammatical data, and three which display results of analysis if the linguist so desires.

Each of the analysis algorithms is designed to operate on the data from an entire text, or from any smaller sample which the linguist may request. Ambiguities resulting from the analysis are carried forward within a list-structure which contains no redundant information. Probabilities are computed for alternatives within the list-structure. The choice algorithms may be bypassed, or they may be used after any stage of analysis to select a specified number of highest-probability alternatives. (Lexical Choice is performed by counting.)

The Lexical and Syntactic Analysis programs are operational and are being used to process English, German, Russian and Chinese. Although careful time studies have not been made, the programs appear to be efficient, due in part to the large amount of data which can be processed on each pass. Specifications have been written for the Semantic Analysis algorithm, but it has not as yet been programmed.

The designs of these analysis programs are too complex to be described here; they will be described in detail in the documentation study, under contract SC 36-039 AMC-02162(E). It will be seen, however, that Lexical Analysis is a procedure which accepts the input corpus and produces a list-structure of syntactic constants. Syntactic Analysis accepts the list-structure of syntactic constants and produces one of semantic constants. Semantic Analysis will accept the list-structure of semantic constants and produce one of pragmatic constants. Transfer through the interlingua will be possible after any one of these steps.

4.3.5 Transfer Maintenance

Transfer data are maintained in the system as transfer rules, which may be formalized as sentences of one of the three interlingual metalanguages.

Lexical transfer rules have form:

(41) $a_{i_1} a_{i_2} \ldots a_{i_s} \varepsilon Sm$

where the a_i are constants denoting lexical segments of some object language, $a_{i_1} a_{i_2} \ldots a_{i_s}$ is a word denoting the concatenation of the segments, and Sm is a predicate denoting some property of interlingual substitution. Thus, as before, the sentence is interpreted as the statement that the property is true of the segment. The rule is recorded by the linguist, along with other rules, for example:

(42) $a_{j_1} a_{j_2} \ldots a_{j_t} \varepsilon Sm$

in which the word $a_{j_1} a_{j_2} \ldots a_{j_t}$ denotes a lexical segment of some other object language. The two segments, or any number of segments in any collection of languages, have thus been described as having the property of interlingual substitution denoted by Sm.

The procedure referred to as Interlingual Transfer Revision then finds the collection:

(43) $\alpha = \{S_{p_1}, S_{p_2}, \ldots, S_m, \ldots, S_{p_u}\}$

of all predicates recording properties of the segment denoted by $a_{i_1}a_{i_2}\ldots a_{i_s}$, and the collection:

(44) $\beta = \{S_{q_1}, S_{q_2}, \ldots, S_m, \ldots, S_{q_v}\}$

of those recording properties of the segment denoted by $a_{j_1}a_{j_2}\ldots a_{j_t}$, and so on. From the rules recording such properties of interlingual <u>substitution</u>, (41) and (42), etc., the process automatically constructs rules recording properties of interlingual <u>distribution</u>, having the form:

(45) $a_{i_1}a_{i_2}\ldots a_{i_s} \varepsilon D_\alpha$

$a_{j_1}a_{j_2}\ldots a_{j_t} \varepsilon D_\beta$

...

The predicate D_α denotes the class of interlingual substitution properties recorded by the predicates in α, D_β the class recorded by predicates in β, and so on.

A metric is also computed by the process as an estimate of the similarity of D_α and D_β, viz.:

(46) $$|D_\alpha - D_\beta| = \frac{N(\alpha \cap \beta)}{N(\alpha \cup \beta)}$$

The numerator indicates the number of predicates in the intersection of α and β, and the denominator the number in the union. These data are recorded on the Interlingua tape. The desired pairs or collections of input and output languages may be specified to the process.

Syntactic and semantic transfer rules have the more complex form:

$$(47) \quad b_{j_1}{}^{\hat{i}_1} b_{j_2}{}^{\hat{i}_2} \ldots {}^{\hat{i}_{p-1}} b_{j_p} \in S_n$$

The formation rules for words are analogous to (28) and (29). In syntactic transfer rules, the word denotes a theorem of the syntactic metalanguage of some object language. The word in semantic transfer rules denotes a semantic theorem of some semantic metalanguage. Properties of interlingual substitution and distribution are handled in the same manner that we have explained for lexical transfer rules. The metrics of interlingual similarity are computed for each stratum.

As a convenience in computation, the same collection of symbols is used in the system for syntactic constants (1) and interlingual constants (41) of lexical transfer rules. Semantic constants (25) and constants of syntactic transfer rules (47) are also grouped together, as are pragmatic constants and constants of semantic transfer rules. The output of Lexical, Syntactic or Semantic Analysis will, as a consequence, permit either further analysis or Interlingual Recognition as the next step in the overall process.

Interlingual Recognition, in alternative segmentations of the input, performs a one-one replacement of words by predicates of interlingual distribution. The predicates produced for each alternative segmentation constitute a phrase, either of the lexical form:

(48) $$D_{\alpha_1} D_{\alpha_2} \ldots D_{\alpha_n}$$

or of the syntactic or semantic form:

(49) $$D_{\alpha_1} {}^{\frown{i}_1} D_{\alpha_2} {}^{\frown{i}_2} \ldots {}^{\frown{i}_{n-1}} D_{\alpha_n}.$$

These are the input data required by Transfer and the Interlingua tape.

Transfer may be made into any other language which has been described in the system or, for simulation of paraphrasing, into the same language. Each predicate of (48) or (49) will be replaced by one which is true of units of the output language, and an estimate of interlingual similarity between the phrases will be computed according to the formula:

(50) $$|D_{\alpha_1} D_{\alpha_2} \ldots D_{\alpha_n} - D_{\beta_1} D_{\beta_2} \ldots D_{\beta_n}| = \frac{\prod_{i=1}^{n} N(\alpha_i \cap \beta_i)}{\prod_{i=1}^{n} [N(\alpha_i) + N(\beta_i) - N(\alpha_i \cap \beta_i)]}$$

This transfer procedure appears to maximize the probability that unresolved ambiguities will be translated correctly.

Monolingual Production will be procedurally the converse of Monolingual Recognition. List-structures similar to those used in analysis will be used by these programs. Note, however, that choice processes always follow analysis and precede synthesis. The computer system will be capable of carrying ambiguities through the entire translation algorithm, so that heuristic processes of choice may be investigated.

4.4 Documentation Study

Programs for automatic documentation of results of linguistic analysis were completed under this contract, and samples of syntactic output were forwarded to the sponsor. None of the results have been published, because programs to automate linguistic verification have just been made operational.

5 CONCLUSIONS

Mechanical translation has, without question, turned out to be a much more difficult problem than was anticipated in early investigations. By the same token, appreciation of the difficulties involved in mechanical translation research has increased our awareness of potential benefits which may result from its solution.

The profound effect which linguistic technology exercises upon society is grossly underestimated. Language is, after all, one of our key inventions. Its discovery led to our current concept of military, economic, political and scientific organizations and signaled the rapid growth in accomplishment which mankind is presently experiencing.

Yet, from the standpoint of linguistic technology, our social edifice has been based upon manual labor. The outcome of our rapid transition to linguistic automation is, therefore, difficult to predict. One thing is certain: our well-being as a society will depend upon our ability to understand the transition.

The most important conclusion of our research is the concept that all types of linguistic automation are based on a single theory. That is, the same structural relations in language which must be described to implement mechanical translation will also make possible the automation of other linguistic processes, such as abstracting, extracting and indexing of technical material.

We do not claim that the present form of our working hypothesis will provide this foundation for linguistic automation. Until system programming is completed, we will not be able to be conclusive in our evaluation of the structural theory reported. However, we do feel that it is an important step toward higher quality translation.

Our confidence is based on the methodology which we have formulated. It provides systematic procedures for perfecting and extending the scope of the working hypothesis on the basis of research experience. In our observation, early studies in mechanical translation were either excessively pragmatic or theoretically oriented without a strong enough foundation in empirical research. Successful investigation requires a balanced program of theoretical and empirical studies.

6 RECOMMENDATIONS

Our conclusion that all types of linguistic automation are based on a single theory suggests the following recommendation:

Basic linguistic research will be concerned not only with progress toward theoretical foundations but increasingly with descriptive studies of those languages, among the several thousand now in use, which are critical to our national interest. The language data which support linguistic automation will be a valuable national asset. Without a long-range plan for recording and disseminating these data as they are discovered,considerable research effort will be wasted.

REFERENCES

1. E. Delavenay, _La Machine a Traduire_. Paris, 1959.
2. Y. Bar-Hillel, _Report on the State of Machine Translation in the United States and Great Britain_. Office of Naval Research, NR 049-130.
3. A. D. Booth, L. Brandwood, and J. P. Cleave, _Mechanical Resolution of Linguistic Problems_. New York: Academic Press, 1958.
4. _Mechanical Translation_, V. H. Yngve, editor, M.I.T. Research Laboratory of Electronics.
5. P. Toma, _Serna System_, Georgetown Univ., 1959.
6. A. F. R. Brown, _Manual for a Simulated Linguistic Computer_. Georgetown Univ., 1959.
7. V. H. Yngve, "A Programming Language for Mechanical Translation," _Mechanical Translation_, vol. 5, July 1958.
8. G. O. Curme, _A Grammar of the German Language_, rev. ed. New York, 1922.
9. Y. Bar-Hillel, "The Present Status of Automatic Language Translation," _Advances in Computers_. New York: Academic Press, 1959.
10. L. Wittgenstein, _Tractatus Logico-Philosophicus_. New York: Humanities Press, 1961.
11. R. Carnap, _The Logical Syntax of Language_. London: Routledge and Kegan Paul, 1959.

12. "Machine Translation of Russian," _National Bureau of Standards Technical News Bulletin_, June 1959.

13. A. G. Oettinger, _Automatic Language Translation_. Cambridge: Harvard Univ. Press, 1960.

14. H. P. Edmundson, K. E. Harper, and D. G. Hays, _Studies in Mechanical Translation_ 1-9. Santa Monica: the RAND Corporation.

15. W. P. Lehmann and E. D. Pendergraft, "Structural Models for Linguistic Automation," _Vistas in Information Handling_ I. Washington: Spartan Books, 1963.

16. N. D. Andreyev, V. V. Ivanov and I. A. Melchuk, "Some Remarks and Suggestions Relative to Work on Machine Translation in the USSR," _Machine Translation and Applied Linguistics_, no. 4. Moscow, 1960 (JPRS 8026, from OTS).

17. N. D. Andreyev, N. D. Kremneva and N. N. Volkov, "The Influence of a Type of Intermediary Language on a Scheme of MT Algorithms." Moscow, 1961.

18. S. Gorn, "Detection of Generative Ambiguities in Context-Free Mechanical Languages," _Journal of the Assn. for Computing Machinery_, April 1963.

19. D. G. Cantor, "On the Ambiguity Problem of Backus Systems," _Journal of the Assn. for Computing Machinery_, October 1962.

20. A. Church, *Introduction to Mathematical Logic*. Princeton Univ. Press, 1956.

21. I. M. Bochenski, *A History of Formal Logic*. Notre Dame, 1961.

22. W. B. Estes, W. A. Holley and E. D. Pendergraft, *Formation and Transformation Structures*, Part I, LRC 63-WTM3. Austin: Linguistics Research Center, The University of Texas. May 1963.

23. L. Bloomfield, *Language*. New York: Henry Holt and Co., 1933.

24. C. W. Morris, "Foundations of the Theory of Signs," *International Encyclopedia of Unified Science*, vol. I, no. 2. Chicago: University of Chicago Press, 1955.

25. S. M. Lamb, *Outline of Stratificational Grammar*. Berkeley: University of California, 1962.

26. N. Chomsky, *Syntactic Structures*. The Hague: Mouton and Co., 1956.

27. J. Backus, ACM-GAMM Conference, ICIP, June, 1959. D. A. Cantor, "On the Ambiguity Problem of Backus Systems," *Journal of the Association for Computing Machinery*, vol. 9, no. 4, October 1962.

28. Z. S. Harris, *Structural Linguistics*. Chicago: University of Chicago Press, 1951.

29. K. L. Pike, "Dimensions of Grammatical Constructions," Language, Journal of the Linguistic Society of America, vol. 38, no. 3 (Part 4), July-September 1962.

30. E. Ruechardt, Sichtbares und Unsichtbares Licht. Berlin, 1952.

________________, Light, Visible and Invisible. Ann Arbor, 1958.

31. L. Hjelmslev, Omkring sprogteoriens grundlaeggelse. København, 1943. English edition: Prolegomena to a Theory of Language. Baltimore, 1953: translated by F. J. Whitfield.

32. P. Rosenbloom, The Elements of Mathematical Logic. New York: Dover, 1950.

33. M. Joos, "Semology: A Linguistic Theory of Meaning," Studies in Linguistics, 1958.

34. R. M. Martin, Truth and Denotation. Chicago: University of Chicago Press, 1958.

35. D. A. Senechalle, Q-Collections and Concatenation, LRC 63-WTM1. Austin: Linguistics Research Center, The University of Texas, January 1963.

________________, Introduction to Formation Structures, LRC 63-WTM2. Austin: Linguistics Research Center, The University of Texas, April 1963.

36. V. H. Yngve, "Left-to-Right Sentence Generation," (unpublished draft). Cambridge: MIT, August 1959.

__________________, "A Model and an Hypothesis for Language Structure," *Proceedings of the American Philosophical Society*, vol. 104, no. 5.

37. E. Sapir, *Language*. New York: Harcourt Brace and Co., 1921.

38. W. V. Quine, *Methods of Logic*. London: Routledge and Kegan Paul Ltd., 1952.

39. R. J. Solomonoff, *A Progress Report on Machines to Learn to Translate Languages and Retrieve Information*. Cambridge: Zator Co., 1959.

40. C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana: University of Illinois Press, 1949.

41. A. Newell, J. C. Shaw and H. A. Simon, *The Processes of Creative Thinking*. Santa Monica: The RAND Corporation, 1958.

42. W. R. Ashby, *Design for a Brain.* New York: John Wiley and Sons, 1960.

43. R. W. Jonas, *System Control* LRC 62-TP1. Austin: Linguistics Research Center, University of Texas, June 1962.

PERSONNEL

Administration:

Dr. W. P. Lehmann, Director

Mr. E. D. Pendergraft, Associate Director

Office:

Mrs. Z. Koprivnik, Secretary

Mrs. L. Childress

Mr. A. Dominguez

Miss H. A. Ireland

Descriptive Linguistics:

Dr. L. W. Tosh, Chief

Dr. S. N. Werbow, Consultant

Dr. W. Winter, Consultant

Chinese:

Miss A. O. Yue

English:

Dr. H. W. Law

Mr. R. B. Bojar

Mrs. M. S. Prince

Mr. K. W. Ryan

German:

Mr. T. Baker

Mrs. K. S. Heilbrunn

Miss E. Hoffmann

Mr. H. H. van Olphen

Mrs. N. R. Orme-Johnson

Russian:

Dr. H. K. Ulatowska-Cole

Systems:

Mr. R. W. Jonas, Chief

Operations:

Mr. O. H. Olson, Jr.

Mr. B. W. Allen

Mrs. R. B. Ambagis

Mrs. M. L. Burkland

Programming:

Mrs. B. C. Foster

Mrs. M. L. Hagemeyer

Mr. T. W. Hill

Mrs. O. D. Janca

Mr. S. A. McGall

Theoretical Linguistics:

Mr. W. B. Estes, Chief

Library:

Mr. J. Bunnag

Mathematics:

Miss J. M. Brady

Mr. D. A. Senechalle