Towards Tomorrow's Linguistics

Roger W. Shuy
Charles-James N. Bailey
Editors

Computer Simulation of Language Contact Models, pp.276-290
Sheldon Klein

1974

Georgetown University Press, Washington, D.C. 20007

COMPUTER SIMULATION OF LANGUAGE CONTACT MODELS[1]

SHELDON KLEIN

The University of Wisconsin

1. Background. For several years I and my students have been attempting to develop a computer based methodology for simulating individual and group language behavior. The quest has lead to the development of language learning programs, behavioral simulation systems, and special programming languages for formulating rules for social behavior and interaction integrated within a linguistic framework.

At this point, on the basis of the successes obtained to date, I wish to describe a methodology that appears to be emerging for the modelling and testing of language contact phenomena within the conceptual frameworks of a variety of linguistic theories and language contact models.

The methodology implies a generalized system consisting of the following components:

(a) an automated linguistic meta-model capable of representing a variety of formulations of linguistic theories within the general character of its notation.

(b) a generalized language learning program that is capable of learning grammars of natural languages, and conforming to any theoretical grammar model that can be represented in the automated linguistic meta-model.

(c) a simulation language in which can be formulated models of behavioral interaction in linguistic speech communities of varying size and on varying time scales.

It is the combination of these components that gives the system the power to model a variety of linguistic theories. The claim rests on the ability to incorporate a massive linguistic and contextual history of any discourse as part of the grammatical data, and the ability of the linguistic generative component to yield sentences that may themselves become behavioral rules governing the linguistic and extra linguistic behavior of simulated speakers. This last feature gives the system the power of the second order predicate calculus.

An early statement of the methodology is contained in Klein (1966). Therein is described a system for simulating a speech community model in which each speaker was represented by a generative grammar containing rules with dynamically modifiable frequencies of usage. Behavioral interaction rules determined who talked to whom and under what circumstances. An interaction consisted of the generation of a sentence by one speaker and the parsing of that sentence by another. Learning in the system consisted of simple adoption of an unfamiliar rule directly from the grammar of another speaker. The modification of the rule frequency parameters was a function of the changing role of that rule in a speaker's auditory experience. Twenty-five years in a hypothetical speech community were actually modelled on a computer. The grammars were trivial, and the social interaction model simple-minded, but the experiment served to establish the feasibility of the methodology. There was birth and death in the community, and new speakers entered the system with empty grammars, acquiring the rules of their community through interactions governed by a social model that paid attention to, and modified social status and age. The relative frequency of each rule in each speaker's grammar was tabulated continuously, as well as the relative frequency of the various rules in the community as a whole. The experiment was run three times on a computer. Each run used a different random number source for its decision-making purposes. The great similarity of the results from these runs indicated that the model was independent of random number source, and that it was a successful Monte-Carlo model demonstrating deterministic control over the phenomena.

The learning that took place in that early system consisted of outright theft of relevant rules from other grammars. Subsequently, several years were spent in developing language learning programs that actually synthesize grammars (Klein, Fabens, Herriot, Katke, Kuppin and Towster 1968 and Klein, Kuppin 1970). While the basic principles of automated grammar synthesis were formulated within particular grammatical models, they have a great deal of generality. The program included phrase structure, transformational and morphology learning capabilities. Their functioning included the continual positing of grammar rules sufficient to account for the input history and with some implied generality. At every state, the posited grammar was subjected to testing and revision through generation of test productions offered to a human informant for acceptance, rejection, or correction. Work on phonology learning programs is in progress.

2. Overview of the components of the simulation system

In the simulation system the role of the human informant in the language learning component is taken over by another part of the program associated with a separate generative grammar. The simulated person in a modelled learning situation would synthesize rules in response to productions from other generative grammars associated with other simulated people. Acceptance, rejection, or correction of test productions would be supplied by mechanisms utilizing the grammars of the simulated auditors to parse, interpret, and correct them. Learned rules in such a system are also associated with generative frequency parameters, and the learning mechanisms may link various rules to specific social contexts. It is also possible to associate more than one grammar with a speaker in the system, and to keep them applicable in totally independent contexts, or to permit them to influence each other as the particular simulation model may demand. A number of devices can achieve this influence: a single speaker might generate using one grammar in a conversation, and at the same time 'eavesdrop' with his second grammar, with learning taking place.

Alternatively, a monolingual speaker might acquire a second language within the structure of a single grammar, the new structures being integrated with his original grammar, but maintain special markings for usage in second language conversation situations. In all cases, the dynamics of the language learning mechanisms would permit controllable degrees of influence by the first language system.

The basis for both the special simulation language and notation for representing semantic model of the universe of a simulated speaker is described in Klein, Oakley, Suurballe and Ziesemer (1971). The two are significantly related. The simulation language defines rules for the dynamics of a changing universe, and those changes are registered as alterations in its semantic representation. The choice of notation for such a representation is crucial to the success of the system. The major handicap in global model construction is the lack of a common notation for relating diverse phenomena. The notation selected here is that of a network in the form of a directed graph with labelled edges. The use of such a device for semantic and logical manipulations and deductions has been prevalent since the thirteenth century (Gardner 1958). More recently, networks have been used in computer programs for proving theorems in the propositional calculus.

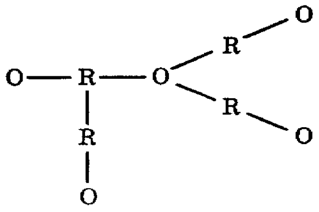
The network notation, in conjunction with certain dynamic features of the simulation language, is powerful enough to model most current linguistic theories. Fillmore (1968:87-88) acknowledged that his theory of case grammar could be formulated in such a network. Even lexically oriented models can be treated by using the network relations to compute the appropriate data for the lexicon.

The network is constructed from semantic triples that are generated by the simulation rules as changing events in the universe, and as learned information obtained from analysis of sentence productions by other simulated speakers. A semantic triple consists of α, γ, β, where α may be a semantic object or a semantic relationship, where γ may be just a semantic relationship, and β just a semantic object. A triple may also be formulated with either an empty α element or an empty β element (but not both). For example:

$O_i - R_j - O_k$ $\qquad$ $O_i - R_j$
$R_i - R_j - O_k$ $\qquad$ $R_i - O_j$
$\qquad\qquad\qquad\quad$ $R_i - R_j$

The connecting links have implied directional arrows, and a particular semantic relation may be viewed as the labelling on an edge of the graph.

The semantic representation of the universe as a whole is a connected graph, e.g.



and is modifiable by additions or deletions. The network can be used directly as the input to a generative system, or it can be used to generate data in other formats for the same purpose.

Because the system is a general meta-model for models, denotation of specific semantic objects and relations is not part of the basic system. Such objects are entered as data by the experimenter or produced during the course of a simulation. The generative mapping rules and relations are specified by the dictates of the particular linguistic theory being modelled.

Other features of the network representation that can be exploited in the simulation system rules and in models of various linguistic theories include the possibility of linking the semantic units to a list of lexical expressions that may contain such data as lexical roots that are 'synonyms' or expression variants for the semantic units. These variant units can be associated with probabilistic values for selection in productions. Such probability values are also amenable to modification dynamically during operation of the simulation system as a consequence of simulated event history. These items may also be marked for any kind of context.

The expression list may contain semantic triples as well as lexical items. These semantic triples may also be associated with frequency probability values and selectional markings. These triples too, may have their units linked to lexical expression lists that may also contain triples, thereby providing the generative system with the ability to use recursive type definitions, and in fact to yield an encyclopedic exposition as the generated expansion of a single semantic triple (if sufficient data is incorporated in the network).

Semantic units may also be associated with numerical values that can be modified by the simulation rules. An example of the exploitation of this feature might involve the choice of a semantic unit of neutral content such as 'affection'. An arbitrary numerical scale, say from +10 to -10, might be linked to emotionally loaded lexical variants: for example 'affection+9' might link to lexical items such as 'adore-', 'affection+1' to 'like-', 'affection-8' to 'hate-', 'affection-9' to 'loathe-'. The ability to modify such values dynamically provides a neat mechanism for controlling the expression of the growth or decline of feelings in relations between speakers in the system.

The system includes provision for linking each semantic relation to one of a limited number of syntactic dependency governance patterns. This information makes it possible to exploit well-developed computer programs for the generation of extended coherent discourse (Klein 1965a, 1965b; Klein, Oakley, Suurballe and Ziesemer 1971). It can also be used with intermediate steps in semantic parsing. The dependency information also makes it possible to compute co-occurrence selectional restrictions dynamically, or to convert semantic network information to data associated with lexical entries (Klein, Lieman, and Lindstrom 1968).

The dependency notation is a convenient syntactic counterpart of the semantic network, and helps to give the system the generality to model a wide range of theories.

Another key feature of the semantic triples is that each may be linked to a value representing the time it was set, or the time it was deleted from the network. 'Time' is, of course, in terms of the clock of the universe of the simulation system, and may be on any scale, from seconds to years. It is thus possible to compute precisely the relative order of all events in the simulated universe, and to use this data as context for linguistic and extra-linguistic rules.

Also, it is possible to have a single, global semantic network accessible to the generative components of all or just some individuals in the model. It is also possible to have private semantic networks for each individual instead of, or in addition to a global network.

The behavioral simulation language is a higher level programming language in which the rules for governing the behavior of individuals and events in the modelled universe of the simulation system may be written. Behavioral rules written in this simulation language react to and manipulate events through testing and modification of the semantic network. The rules that make changes are probabilistic, and the probabilities associated with the rules are derived from particular subconditions found to be present or absent in the semantic network at any given moment. The simulation language has the power of the first order predicate calculus. Deterministic rules may be formulated. The fact that the rules may also be probabilistic complicates the logical status, and logical theory has paid little attention to such complex systems.

The flow of the simulation can be controlled by a branching logic whereby different paths through the execution of whole blocks of rules may be controlled by various tests. Furthermore, each rule may itself be associated with an evaluation frequency parameter. Rules may be evaluated at any specified time interval. Within the same model one can have rules for short term, detailed behavior, rules for cyclic events that might occur weekly, monthly, or yearly, and rules for the occurrence of specific historic events.

The syntax of the behavioral simulation language permits the dynamic creation of classes of semantic objects, testing for the existence of specific relations in the network, and testing for the existence of complex network paths defined by Boolean conditions. For example, is there a path between semantic object A and semantic object B, through relation R_k or R_i and R_j but not through R_k?

Many subconditions may be formulated for a given rule. The satisfaction or nonsatisfaction of a subcondition may add or subtract fractional values in computing a final probability result. Consider a rule, 'John kisses Mary'.

This will become an event in the model at a specific moment in time only by a chance selection whose outcome can be affected by the existence of certain preconditions.

For example:

Rule: John kisses Mary

Conditions: Give up if John and Mary are not in the same physical location
If John loves Mary add .2
If Mary loves John add .2
If there are people present, subtract .3

If the outcome is for implementation of the event, a semantic triple expressing the pertinent data (including the time) is entered in the semantic network.

The rules and tests might be formulated more generally, i.e. 'Men kiss Women'. In this case the rule would automatically be computed for applicability to all semantic objects belonging to these classes at appropriate times.

Class may be modified during the flow of the simulation or even created and destroyed. One might for example create the set of all men who have kissed a girl between 10 and 11 a.m. in cities with population less than 10,000.

It is not necessary to add classes to the semantic network as semantic objects. The simulation language has devices for bookkeeping associated with class lists; each class concept contains a list of the basic semantic objects that are its elements. Of course one can also add the class as a semantic object, but this is a redundant action. If it is desired to provide a lexical representation of a newly created class concept, the term could be added to the lexical expression lists of the semantic objects that are its members. Thus, a special term representing the new concept 'small-town, mid-morning lover' could be added to the lexical lists associated with people fitting the description in the earlier example.

Similarly, a class that has lost functional significance could be deleted, and any associated lexical terms representing this class membership could be removed from lexical lists of the class member semantic objects, either immediately, or through a gradually decreasing selectional probability value.

The dynamic class creation property may also be used to retain memory of events in the universe that have been deleted from the semantic network. This device is an alternate to the previously mentioned possibility of associating a termination time for a semantic triple; for example, the set of all men who have kissed Mary but who no longer love her.

The features which give the system the power of the second order predicate calculus are dependent upon methodological techniques at the state of the art level in computer science systems programming technology. In simple descriptive terms, it is possible for an individual speaker in the system to generate a sentence that can itself be interpreted as a behavioral rule governing the flow of the model. Thus, a mother may say to a child, 'Don't talk to strangers unless someone you know is with you'. This sentence could be analyzed, reformulated, and submitted to a compiler that handles the syntax of the behavioral simulation rules. This command may then become one of the behavioral rules governing the action of the child. Of course it may be probabilistic. The learning of contradictory rules presents no problem because both may exist simultaneously, but with varying probabilities.

3. An overview of the integrated system

So far I have given a brief sketch of the features of the individual components of the simulation system. I would now like to describe some of the features that appear in the integrated whole.

Imagine a model of a speech community. Each individual in the model is associated with a semantic network (or portion thereof) representing his entire status and role in life and personal knowledge of his simulated universe. The range and level of detail is restricted to that which the experimenter has deemed appropriate for the simulation. For each individual, the semantic network is associated with generative rules cast in the notation of some particular theoretical model. All rules in the generative system are associated with selectional probabilities and markings for contextual usage.

The network of one individual may be linked to more than one set of generative rules, each representing another language or lect. Alternatively, a combined model in which a grammar has paths to alternative subgrammars containing blocks of rules, with entry to such blocks determined by context is also possible.

The most significant fact about this model is its automated four-dimensional nature. All modelled objects are continually subject to change with the passage of time. The flow of the simulation determines conversational interaction, and it is the conversational interaction among members of the community that determines linguistic variation and change.

The responsibility for effecting these changes rests with the language learning component: it controls learning and accommodation of old grammars to new productions according to the dictates of modelled learning theories.

A child's grammar might undergo major change in all of its data and rules at frequent intervals; the grammar changes of an adult might be limited to more superficial adjustments.

I can make the claim that the grammar system of any individual who has been 'born and raised' in the simulation model will be a function of his history of conversational interaction, mediated by the relative influence of these interactions as determined by the experimenter's socio-linguistic theory.

To explain why this can take place in the system for a broad range of theoretical models requires some explication of the general nature of language learning programs. One may make the following generalizations.

The basic learning mechanisms may be formulated as alternative tactics that may be used in varying combinations:

(a) they may include formulation of rules based on maximum generalization from the data;
(b) they may be based on minimum generalization from the data;
(c) existing rules may themselves be subjected to unification and generalization according to the principles of (a) and (b);
(d) for different portions of the grammatical system, the extremes of learning heuristics (a) and (b) may be applied in different degrees.

The learning rules can be formulated so as to permit learning heuristics that apply in a given conversational interaction to be determined by the behavioral rules of the model. Accordingly it becomes possible to control the kind of language learning that will take place on the basis of age and social context according to the dictates of the experimenter's model.

To understand the workings of the system it is imperative to understand another point. Automated language learning programs are inherently sensitive to training sequence. The same set of inputs presented to a language learning program will, in general, yield different grammars. All of the grammars will account for the given corpus, but each may imply varying productions outside that corpus. For obvious reasons, testing of the grammars' implications can never be complete, therefore the variation is inevitable. It can be argued that it is the same in real life.

At this point I will state a hypothesis that may be obvious, but which I believe has never been formulated in quite this way for natural languages:

> The formal properties of natural languages make language change during the transmission process a logically unavoidable consequence.

I base this statement on the formal proof by Shamir (1962) that implies its validity for context-free, phrase structure languages, and empirical results with context sensitive language learning programs (Klein, Fabens, Herriot, Katke, Kuppin and Towster 1968; Klein and Kuppin 1970; Feldman 1969; Feldman, Gips, Horning, and Reder 1969; Horning 1969).

It is obvious from this that, in the simulation system, the key factors that control grammar variation are the simulation rules that determine conversational interaction.

Control of the learning process sufficient to guarantee the maintenance of a speech community (avoidance of the Tower of Babel syndrome) is also insured through the conversational interaction process. A child producing a sentence an adult cannot parse and interpret semantically will continue a major process of modification of his grammar until adequate communication is established. The process may take simulated years.

Each individual tries to parse the productions of other speakers, and where the parsing is not wholly successful, he may either synthesize new rules, or rest content with a comprehension of the content deduced through ambiguity resolution techniques that make reference to the context of his semantic network.

In effect, the members of a simulated speech community can be made to act as a control on the deviancy of individual grammars.

Other powerful controls on the learning process exist. The simulation is run on a computer, and memory of past events is an easily provided benefit. Associated with each individual is a history of every speech production he has ever heard. He may be associated with a list of every one of his own productions that was ever rejected or corrected in his language learning history. This data is used to guarantee that the grammar can account for the valid sentences and avoid the faulty ones, or else differentiate their usage as a function of specific extra-linguistic context.

In consideration of both the tendency towards deviance and the pressure toward conformity the following should be true if the model is non-trival:

Construct a simulation model containing only adults, each with an identical semantic network, and each linked to identical grammatical rules. Permit the simulation to run for several generations, allowing birth and death, until the entire starting population is dead. At that point, every grammar of every living speaker in the system would be different. And yet a viable speech community can still exist. These varying grammars may still yield sentences that can be parsed by other members in the community, despite the apparent differences in the grammars. A functional approximate equivalence of the grammars can be maintained.

And, if the experimenter wishes, he can control the rules of interaction so that dialects appear over the generations with an increasing lack of mutual intelligibility. Again, this type of communal language change can be sensitively controlled by the social interaction model formulated in the simulation rules.

The problem of modelling a polylingual language contact situation does not require any special mechanisms. The computer programs are neutral to the content of the grammars associated with individuals. They may be different languages as well as different lects. In a similar manner, the program mechanism can be made to treat multilingual inputs as part of the learning experience of any individual. The formal program logic is identical. The effects of language contact on existing grammatical systems can, of course, be controlled by the experimenter according to the dictates of the model he is testing. He may attempt to simulate influence of one linguistic system upon another in any way he desires by varying the kinds of language learning heuristics that are permitted to operate in language contact interactions.

In this section I have made a distinction between semantic network, grammar rules, and behavioral rules. It should be apparent from earlier sections that the discussion applies as well to the learning of semantic relations and behavioral rules.[2] The system provides the linking mechanism that makes it possible to model semantic, social, and cultural change within the same conceptual framework.

4. Formulating the results

Vast amounts of tabulated information can be made available as the product of a simulation. The results can be formulated at the discretion of the experimenter. For example, it is possible to describe the language situation of the speech community in terms of a single, global grammar for the entire community, complete with frequency rules and relevant specifications of context through the simple device of introducing into the community a non-participant, universal 'listener'. This auditor would be like other simulated individuals except that he would process all conversations among all individuals at all times. If this auditor wished to query an informant about the acceptability of a production implied by a newly created rule, he might do so, but this interaction would be a meta-query, and have no effect on the system. The grammar formulated by the auditor could be made to follow many descriptive formats--the auditor could use more powerful learning mechanisms than those available to individual members of the community. If desired, the auditor could be primed with some variant of a universal grammar, and be required to learn rules within its framework. (All the individuals in a simulation could be made to function similarly.)

Statistical analysis of the results for correlations between facets of grammar and socio-demographic status is an easy matter. The computations can be carried on continuously during the flow of a simulation, providing analyses at every point in its history.

Some techniques for matching the results of simulations with predicted real world data have been described in Klein, Kuppin, and Meives (1969). The suggested methodology involved the use of census data for New Zealand coupled with grammatical data and functionalist ethnographic studies of the Maori, to formulate a model of Maori language variation over a 100-year period. Such a simulation would make predictions about the contemporary community that could be verified by live fieldwork. The linguistic link would be the set of all sentences ever generated in the history of the simulation. As a final condition of termination, each surviving individual could be required to attempt to parse every sentence ever produced by anyone in the community during the preceding 100 years. The results might be unique for each individual, and correlations with terminal socio-demographic factors could be tabulated. The same sentence list could then be offered to a real world sample of the Maori population in a questionnaire that would also be used to elicit socio-demographic data. A statistical analysis of these results could then be compared with the results for the simulation to determine its accuracy and, indirectly, verification of the assumptions and intermediate results.

A comment on time scale and sampling techniques is appropriate at this point. It is of course possible to construct a model that simulates interaction in great detail for a very brief period of time. In fact, one could devote an entire simulation to the modelling of a single hour's interaction of two or three speakers. It is also possible to model on a time scale covering months or years. For practical purposes, the longer the time span the less specific must be the detail. Several techniques are possible. One may decrease the overall detail for all individuals in the simulation, and attempt to work with what appear to be key items. Alternatively, one can work in great detail with a few individuals taken as a representative sample of a much larger population. Techniques for constructing such samples are described in Klein, Kuppin, and Meives (1969). There is yet a third technique: one may work with a detailed model of a large, full-scale population but sample the conversational interactions at infrequent intervals, say once a month or once a year. The learning rules in such a model would require that each sample interaction have a more significant impact on the structure of grammars than in other models.

5. Obtaining the data for testing

The range of simulation experiments that can be run is limited by the accessibility of the necessary linguistic and socio-demographic data. In many domains of interest one may wish to run a simulation ignoring all linguistic factors except phonology and lexicon. Certainly this restriction of subject matter has not inhibited scholars from publishing theories on the basis of such meager information. It may be the case that such data may be proven adequate to confirm limited theories.

A tempting domain for modelling is the field of Pidgin and Creole languages. In many cases a great deal of data exists or can be obtained through appropriate fieldwork. Modelling of Creole communities offers the possibility of simulating within a great range of time scales and levels of detail. Especially tempting is the creation of models to test hypotheses about the origin of Creoles. Consider the attempt to model the growth of Creoles from a unique origin and the attempt to synthesize several Creoles from independent sources. There are three possible outcomes: the first, annoying; the second, anticipated; and the third, amusing:

(1) Failure to produce anything resembling a Creole.
(2) Success with one model but failure with the other.
(3) Success with both models.

6. Implementation of the system

The system as described here is not in existence. Nevertheless, every component described in this paper has had a working, tested predecessor. The references indicate the documentation. The proposed systems described herein involve only extensions of existing programs, integrated in a single system running on a single, suitable computer. The simulation of Klein (1967) was run on a Philco 2000 computer, AUTOLING ran on a Burroughs 5500 and 6700, and the current version of the simulation language is operational on a Univac 1108.

Improvements and modification of the simulation language has been carried to a state in advance of that described in Klein, Oakley, Suurballe, and Ziesemer (1971). Classroom testing has yielded simulations and models of hockey games, a cocktail party, space wars, an orgy, a 1969 student street party that culminated in a riot, and a model of the acquisition of political power in a Polynesian society.

The system is entirely implementable within the contemporary state of the art of systems programming. The methodology draws upon the latest techniques associated with third and fourth generation computer system programming, and the state of the art in the construction of meta-compilers.

The construction of the system is inevitable.

NOTES

[1] The research described in this paper has been supported by The National Science Foundation, Grant No. GS-2595 and by The Wisconsin Alumni Research Foundation.

[2] The ability to model the learning of semantic relations makes it rather easy to handle presupposition theories. But the acquisition of semantic knowledge takes place at discrete, separate points in time. Because of the way this data is recorded and represented, the functions of a presuppositional model would be provided in almost any theoretical framework, without the need for an explicit recognition of the concept. Accordingly, I suspect the system could accommodate variants of presuppositional theory more powerful than any currently formulated.

REFERENCES

Feldman, J. A. 1969. Some decidability results on grammatical inference and complexity. Artificial Intelligence Memorandum, Computer Science Department, Stanford University.

_____, J. Gips, J. J. Horning, and S. Reder. 1969. Grammatical complexity and inference. Technical Report No. CS 125, Computer Science Department, Stanford University.

Fillmore, C. J. 1968. The case for case. In: Universals in linguistic theory. Edited by E. Bach and R. T. Harms. New York, Holt, Rinehart, and Winston.

Gardner, M. 1958. Logic machines and diagrams. New York, McGraw-Hill, 1-27.

Horning, J. J. 1969. A study of grammatical inference. Technical Report No. CS 139, Computer Science Department, Stanford University.

Klein, S. 1965a. Automatic paraphrasing in essay format. Mechanical Translation, Vol. 8, Nos. 3 and 4 combined.

_____. 1965b. Control of style with a generative grammar. Language. 41(4). Also in German translation, 1971 as Stilkontrolle mit einer generativen Grammatik. In: Literaturwissenschaft und Linguistik: Ergebnisse und Perspektiven, Band 1. Edited by J. Ihwe. Frankfurt, Athenäum.

Klein, S. 1966. Historical change in language using Monte Carlo techniques. Mechanical Translation, Vol. 9, Nos. 3 and 4 combined. Also in press in Languages in contact, edited by V. Rozencvejg. The Hague, Mouton.

_____ and T. A. Dennison. 1971. An interactive program for learning the morphology of natural languages. University of Wisconsin Computer Sciences Department Technical Report No. 144. Also in press, Proceedings of the 1971 International Conference on Computational Linguistics, Debrecen, Hungary.

_____, W. Fabens, R. Herriot, W. Katke, M. A. Kuppin, and A. Towster. 1968. The AUTOLING system. University of Wisconsin Computer Sciences Department Technical Report No. 43.

_____ and M. A. Kuppin. 1970. An interactive, heuristic program for learning transformational grammars. Computer Studies in the Humanities and Verbal Behavior. 3(3).

_____, M. A. Kuppin, and K. Meives. 1969. Monte Carlo simulation of language change in Tikopia and Maori. Preprints of papers presented at 1969 International Conference on Computational Linguistics, Sånga Säby, Sweden. Stockholm, KVAL.

_____, S. L. Lieman, and G. E. Lindstrom. 1968. DISEMINER: A distributional-semantics inference maker. Computer Studies in the Humanities and Verbal Behavior. 1(1).

_____, J. D. Oakley, D. A. Suurballe, and R. A. Ziesemer. 1971. A program for generating reports on the status and history of stochastically modifiable semantic models of arbitrary universes. University of Wisconsin Computer Sciences Department Technical Report No. 142. Also in press, Proceedings of the 1971 International Conference on Computational Linguistics, Debrecen, Hungary. Also in press, Statistical methods in linguistics, No. 8.

Shamir, E. 1962. A remark on discovery algorithms for grammars. Information and Control. 5(3).