ED 054 832 LI 003 108

TITLE Research 1970/1971: Annual Progress Report.

INSTITUTION Georgia Inst. of Tech., Atlanta. Science Information Research Center.

SPONS AGENCY National Science Foundation, Washington, D.C.

PUB DATE 71

NOTE 127p.;(130 References)

EDRS PRICE MF-$0.65 HC-$6.58

DESCRIPTORS Automation; Bibliographies; *Computer Science; *Information Processing; *Information Science; *Information Systems; Information Theory; Linguistics; *Research; Semiotics

IDENTIFIERS *Systems Design

ABSTRACT

The report presents a summary of science information research activities of the School of Information and Computer Science, Georgia Institute of Technology. Included are project reports on interrelated studies in science information, information processing and systems design, automata and systems theories, and semiotics and linguistics. Also presented in the report is a description of the programs of the School of Information and Computer Science, and a summary of research activities at the Information/Computer Science Laboratory. The report concludes with a bibliography of publications for the period 1970/71. (Author)

Work reported in this publication was performed at the School of Information and Computer Science, Georgia Institute of Technology. The primary support of this work came from the Georgia Institute of Technology and from agencies acknowledged in the report.

* * * *

The graduate School of Information and Computer Science of the Georgia Institute of Technology offers comprehensive programs of education, research and service in the information, computer and systems sciences. As part of its research activities the School operates, under a grant from the National Science Foundation, an interdisciplinary science information research center. Correspondence concerning the programs and activities of the School may be addressed to Director, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Georgia 30332. Telephone: (404) 873-4211.

GITIS-71-03

(Research Report)

RESEARCH 1970/1971:
ANNUAL PROGRESS REPORT

By the

Science Information Research Center

128 p.
130 Ref.

4 =>

~~School of Information and Computer Science~~, 1971
GEORGIA INST~~ITUTE~~ OF TECH~~NOLOGY~~ 3
Atlanta, Georgia

ACKNOWLEDGEMENT

The work reported in the paper has been sponsored in part by the National Science Foundation Grant GN-655. This assistance is gratefully acknowledged.

Wash. FGK57095

Vladimir Slamecka

VLADIMIR SLAMECKA
Project Director

ABSTRACT

The report presents a summary of science information research activities of the School of Information and Computer Science, Georgia Institute of Technology. Included are project reports on interrelated studies in science information, information processing and systems design, automata and systems theories, and semiotics and linguistics. Also presented in the report is a description of the programs of the School of Information and Computer Science, and a summary of research activities at the Information/Computer Science Laboratory. The report concludes with a bibliography of publications for the period 1970/71.

CONTENTS

PREFACE

The present report is the third annual account of major activities of the Science Information Research Center at the School of Information and Computer Science, Georgia Institute of Technology. The report covers the 12-month period extending from July 1970 to June 1971.*

The Georgia Tech Science Information Research Center was established in 1967, under partial sponsorship of the National Science Foundation. The principal objective of the Center is to contribute, through research, to the body of knowledge in information science, and to the utilization of scientific information. An important second function of the Center is research training, accomplished through the participation of graduate students in the research activities of the Center.

Since its establishment the Center has devoted a major part of its activities to two areas: studies in the scientific foundations of the discipline of information science, particularly in the theory of information and its processes; and the development of techniques for a more effective and efficient exploitation of scientific information, both by and outside the scientific community. During the past year the Center began to formulate a modest program of studies viewing science information as an instrument for science policy formulation. National concern with scientific knowledge has recently extended to issues of prudent policies for the management of knowledge as a national and international resource; central to these issues is not only the utilization of this resource but also its production, value, and effect on both science and other enterprises of society. In the years to come, the Georgia Tech Science Information Research Center proposes to emphasize research relevant to these problems.

The intent of this annual research report is to provide capsule summaries of research activities and results, since more formal publications in journals invariably incur delays. Included in this report are also a brief review of the present status of the academic programs of the School of Information and Computer Science, and a bibliography of issued or accepted publications of the School.

---

*For previous reports see (1) Slamecka, V. *The Georgia Tech Science Information Center: A Summary Report, 1967/1969.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1969. Research Report, GITIS-69-18. 30 p.; and (2) Georgia Institute of Technology, Science Information Research Center. *Research 1969/1970: Annual Progress Report,* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report GITIS-70-09. 84 p.

The annual report highlights the main research activities of the Science Information Research Center; it does not comprise the full record. Particularly lacking in it is an indication of the profound impact which the existence of the Center has had on the student body of the School of Information and Computer Science, on the Georgia Institute of Technology, and on the research and education communities of the State of Georgia. It is also in consideration of this effect that the faculty and students of the School of Information and Computer Science express a sincere acknowledgement to the sponsors of the Center: the Office of Science Information Service of the National Science Foundation, and the University System of Georgia, through the Georgia Institute of Technology.

VLADIMIR SLAMECKA
Director

Atlanta, Ga.
August, 1971

THE SCHOOL OF INFORMATION AND COMPUTER SCIENCE

The establishment of academic and research programs in the information, computer and systems sciences at the Georgia Institute of Technology has been a direct response to the concerns, voiced by both the Congress of the United States and the scientific community, with the management and utilization of man's knowledge. The response was the joint effort of the National Science Foundation, charged by Congress to attend to these concerns, and the Institute.

In 1961, under the sponsorship of the National Science Foundation, the Institute appointed a committee to study and develop long-range approaches to education in information science. The results of the committee's work were reported at two national conferences, conducted at the Atlanta campus of the Institute in October 1961 and April 1962, to a national audience of scientists, engineers, information specialists, librarians, educators and administrators.

Following the second conference, the Institute began to plan the establishment of degree programs in information science. Academic programs leading to the degree of Master of Science in Information Science were endorsed in the early Fall of 1962 by both the Graduate Council of the Institute and the Board of Regents of the University System of Georgia, and formally opened in September 1963 with a generous assistance by the Division of Education of the National Science Foundation. The doctoral program was inaugurated in 1968. In 1970, the name of the School and the designation of its degrees was changed to "Information and Computer Science."

The structure and content of the educational programs of the School were given by two types of need: the need of society for individuals educated or trained to perform specific functions; and the intrinsic need of a science to assure its own development and growth. The School perceived that its graduate programs must include a strong component concerned with the development of a theoretical base for the information-based professions; and it was equally obvious that they must not ignore the social mandate to supply professional personnel and effective methods capable of controlling and improving the functions of information management and transfer. Thus both theoretical and professional programs were indicated and implemented, forming a solid and logical foundation for continued development of the discipline, and justifying the societal role of the School.

At the present time, seven years since its establishment, the School of Information and Computer Science is the largest graduate department of the Georgia Institute of Technology, in terms of students enrolled and graduated.

The School currently offers extensive educational programs leading to the designated degrees of Doctor of Philosophy and Master of Science. The objective of the doctoral program is to prepare individuals for research or academic careers; the program is therefore theoretically oriented, and much emphasis is placed on the students' early involvement in research. The M.S. programs on the other hand are applied in character, seeking to educate competent professionals for two types of career: information systems engineering, and computer systems engineering.

Several other academic programs are also offered by the School in information/computer science and/or engineering, including: an off-campus M.S. degree program; an evening M.S. degree program; an undesignated B.S. degree program; an Institute-wide "minor" for the undergraduate division of Georgia Tech; and a curriculum in information/computer science for high school teachers. Among new programs scheduled for establishment in 1972 are a full graduate-level program in Biomedical Information and Computer Science, to be offered jointly with the School of Medicine, Emory University; and an extensive, formal baccalaureate degree program in information and computer science.

Table 1 summarizes the educational activities and programs of the School of Information and Computer Science since 1964. Tables 2 and 3 offer a statistical overview of the development of the School from the viewpoints of student enrollment and faculty staffing. Finally, a list of courses offered by the School is shown in Table 4.

References

1. Slamecka, V. "Georgia Institute of Technology, School of Information and Computer Science". In: Kent, A., and Lancour, H., (eds.) *Encyclopedia of Library and Information Science*. New York, Dekker, 1972. Volume IX (in printing.)

2. Georgia Institute of Technology. *Graduate Catalogue and Announcements, 1971/1972*. Pp. 118-125.

3. Slamecka, V. and Zunde, P. "A University-Wide Program in the Information Sciences." *Proceedings of the IFIP World Conference on Computer Education*, Amsterdam (Holland), 1970. Vol. II, pp. 229-234

Table 1. Educational Programs of the School of Information and Computer Science, 1964-1972

Program	Degree	Year of Implementation	Enrollment in Fall Quarter 1970
Information Systems Engineering	M.S.	1964	38
Computer Systems Engineering	M.S.	1966	43
Undergraduate Service Curriculum	Non-degree	1966	619
Information and Computer Science	Ph.D.	1968	13
Off-Campus Program at Lockheed-Georgia Co.	M.S.	1969	16
Undergraduate Minor in Information and Computer Science	Non-degree	1970	30
Curriculum in Information Sciences for High School Teachers	Non-degree (a)	1970	24
Information/Computer Systems Engineering (evening program)	M.S.	1971	30 (est. in 1971)
Graduate Program in Biomedical Communication	M.S. (b)	1967	30 (total graduates)
Graduate Program in Biomedical Information Science (c)	M.S. Ph.D.	1971/72	20 (est. in 1972)
Degree Program in Undergraduate Information/Computer Science	B.S.	1972	100 (est. in 1973)

(a) Applicable to the M.A. degree in Education
(b) Offered through a consortium of universities, administered by Tulane University; ceased in 1970
(c) Jointly with Emory University, School of Medicine

Table 2. Student Enrollment, 1964-1972

Fiscal Year	Total Enrollment	Total New	Degrees Awarded
1964/65	30	22	1
1965/66	38	15	23
1966/67	65	40	12
1967/68	108	46	34
1968/69	137	53	51
1969/70	155	84	40
1970/71	185	102	47
1971/72*	200	110	60

*Estimated

Table 3. Faculty/Staff Development, 1964-1972

Fiscal Year	Equivalent Full-Time*					
	Faculty		Pre-Doctoral & Lecturers	Graduate Assistants	Other Staff	Total
	Professorial	Instructors				
1964/65	3.44	–	–	1.55	2.26	7.25
1965/66	2.72	–	.68	.39	1.00	4.79
1966/67	4.60	.19	.77	2.84	2.36	10.76
1967/68	6.44	1.64	1.38	4.70	5.67	19.83
1968/69	7.27	1.23	3.36	3.29	5.66	20.81
1969/70	9.34	1.75	3.19	4.55	7.59	26.42
1970/71	10.80	1.39	3.72	3.90	7.72	27.53
1971/72 (Est.)	13.19	2.00	3.40	2.96	8.00	29.55

*1.00 equals full-time for one fiscal year

Table 4. Approved Courses Offered By School Of ICS

Course	Title
ICS 110	Information, Computers, Systems: An Orientation
ICS 151	Digital Computer Organization and Programming
ICS 215	Technical Information Resources
ICS 251	Automatic Data Processing
ICS 256	Computer and Programming Systems
ICS 310	Computer-Oriented Numerical Methods
ICS 325	Introduction to Cybernetics
ICS 336	Introduction to Information Engineering
ICS 342	Introduction to Semiotics
ICS 355	Information Structures and Processes
ICS 401/402	Languages for Science and Technology
ICS 404	Topics in Linguistics
ICS 406	Computing Languages
ICS 410	Problem Solving
ICS 415	The Literature of Science and Engineering
ICS 423	Mathematical Techniques for Information Science
ICS 424	Elements of Information Theory
ICS 436	Information Systems
ICS 445	Logistic Systems
ICS 452	Logic Design and Switching Theory
ICS 458	Computer Systems
ICS 607	Communication and Control of Information
ICS 608	Syntax of Natural Languages
ICS 609	Mathematical Linguistics
ICS 612	Graph Theory
ICS 616	Information Control Methods
ICS 621	Theory of Communication
ICS 625	Cybernetics
ICS 626	Information Processes I
ICS 627	Information Processes II
ICS 628	Theory of Models
ICS 629	Information Measures
ICS 632	Equipment of Information Systems
ICS 636/637	Information Systems Design I, II
ICS 638	Problems in Systems Design
ICS 642	Advanced Semiotics
ICS 645	Advanced Logic
ICS 646	Philosophy of Mind
ICS 647	Artificial Intelligence
ICS 652	Advanced Computer Organization
ICS 653	Computer Techniques for Information Storage and Retrieval
ICS 656	Computer Operating Systems
ICS 657	Design of Computer Operating Systems
ICS 658	Evaluation of Computer Systems
ICS 661	Computer Language Design
ICS 673	Organization and Management of Information Industry

Table 4. (Cont'd.)

Course	Title
ICS 682	System Theory I
ICS 683	System Theory II
ICS 700	Master's Thesis
ICS 701/702/703	Seminar
ICS 704/705/706	Special Problems in Information Science
ICS 704	Combinatory Logic and the Calculi of Lambda-Conversion (Special Problems Course)
ICS 706	Pattern Recognition (Special Problems Course)
ICS 706	Management Information Systems Design (Special Problems Course)
ICS 710	Philosophy of Language
ICS 726	Theory of Automata
ICS 736	Information Systems Optimization
ICS 738	Advanced Systems Design
ICS 761	Syntax-Directed Compilation
ICS 799	Ph.D. Dissertation Preparation
ICS 800	Doctor's Thesis

STUDIES IN SCIENCE INFORMATION

Structural Analysis of National Science Information Systems

V. Slamecka, P. Zunde, D. H. Kraus

Surveillance of the development of national science information systems in six countries with planned economies (Bulgaria, Czechoslovakia, Hungary, Poland, Romania, and Yugoslavia) was continued in 1970/1971 following the publication of a series of reports and papers [1-8] that concluded the first phase of the study. The initial study had produced detailed descriptive and, when possible, quantitative surveys of the development and apparent effectiveness of the national systems for scientific, technical, and economic information in the six countries.

In 1970, the M.I.T. Press offered to publish an abridged and updated version of the reports. It was felt desirable to visit the major information centers in each of the countries of the study to procure first-hand information as well as literature generally not available in the United States. Partial support for such travel was provided by NSF Travel Grant GN-885, and the visits were made during 1970. Interviews were granted and literature furnished or promised in four of the six countries (Bulgaria, Hungary, Poland, Romania) and in Paris by a UNESCO expert on the information systems of Eastern Europe. Literature for updating and revising the reports has arrived since that time. Detailed information on the revisions of the Romanian information system was received, enabling the researchers to issue a revised version of the full report on Romania [9]. In addition, a special study was made of the Bulgarian library and documentation system [10].

Work on the text of the book for the M.I.T. Press was begun early in 1971. Tentatively entitled *A Guide to Scientific, Technical, and Economic Information and Documentation in Eastern Europe,* the book will contain the following: a summary of the origins and evolution, current status, and developmental trends of information and documentation in the six countries of the study; a survey of international cooperation in information transfer and dissemination in the CMEA countries (the aforementioned six, the Soviet Union, Mongolia, and Cuba); a report on the International Scientific and Technical Information Center in Moscow; and brief accounts, by country, of the development, current status, and special features of scientific, technical, and economic documentation and information systems in each country, followed by a directory of its information and documentation centers, and a list of the publications of these centers. A bibliography will be supplied for each chapter and a subject index will be prepared for the volume.

References

1. Kraus, D. H. *Scientific and Technical Documentation and Information in Bulgaria*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1968. Research Report. 79 p. (PB 179 981.)

2. Kraus, D. H. *Scientific, Technical, and Economic Documentation and Information in Czechoslovakia*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1968. Research Report. 298 p. (PB 179 982.)

3. Kraus, D. H. *Scientific, Technical, and Economic Documentation in Hungary*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1968. Research Report. 144 p. (PB 179 983.)

4. Kraus, D. H. *Scientific and Technical Documentation and Information in Romania*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1968. Research Report. 91 p. (PB 179 985.)

5. Kraus, D. H. *Scientific and Technical Documentation and Information in Yugoslavia*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1968. Research Report. 120 p. (PB 179 986.)

6. Slamecka, V., Zunde, P., and Kraus, D. H. "On the Structure of Six National Science Information Systems," *International Forum of Informatics*, VINITI, Moscow, 1969, pp. 318-334.

7. Zunde, P. *Scientific and Technical Documentation and Information in Poland*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1968. Research Report. 345 p. (PB 179 984.)

8. Zunde, P. "Co-operation of the Information Agencies of the CMEA Countries," *International Library Review*, (1969), 1:513-535.

9. Kraus, D. H. *Scientific and Technical Documentation and Information in Romania, 1970*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report, GITIS-71-02. 75 p.

10. Kraus, D. H. "Bulgaria, Libraries in," *Encyclopedia of Library and Information Science*, New York, M. Dekker, 1970. vol. 3, pp. 471-484.

Predictive Models of Scientific Progress

P. Zunde, V. Slamecka

Progress in science is essentially determined by the stimulating effects of information accumulation and transfer. Scientific communication transmits and disseminates generated information among information generators, actual or potential, and stimulates them to produce new information. It is therefore reasonable to postulate that the rate of development in a particular field of science depends on intensity of information generation in that field and on the intensity of stimulation or incitement both from within that discipline and from without, by contributions from other disciplines. In other words, the more scientific activity taking place in a particular scientific discipline, such as physics, and the more these activities are stimulated by scientific work in various other disciplines, the greater we can expect the rate of the development of this discipline to be. We can assume that the intensity of information generation in various disciplines as well as the intensity of stimulation effect of generated information can be measured with some accuracy (and, indeed, have shown as part of this research project that such measurements are feasible at least under certain simplifying assumptions).

It is further hypothesized that although the stimulating effect of certain information generators (sources) might be of long range, the immediate predecessors in the information generation process are on the average the most significant stimulants or incitors and that they represent in a sense the accumulated effects of the whole past history. This assumption is strongly supported by the evidence gained from research on literature citations, which shows that literature cited in scientific publications is, for all practical purposes, limited to one or two decades preceding the publication date of the citing document [1].

Under these assumptions, the thrust of this research effort has been to model the process of science development as a Markov chain defined on the relative intensity of scientific productivity in a set of scientific disciplines as states, the state transition probabilities corresponding to the estimated degrees to which scientific production in one discipline stimulates scientific activity in another. Figuratively speaking, the process of science development is considered to be a result of a stimulating action of global stimulators or "incitors" in the sense that the rate of the development in a scientific discipline at time $t + \Delta t$ is a function of the relative intensity of the incitor action in that discipline at the time t and of the change of emphasis of the incitor action relative to other disciplines during the time Δt. Clearly, this cumulative action consists of all the contributions of individual incitors which thus provide the impact for further scientific activity.

Specifically, let us now consider the whole field of human knowledge - - or some portion of it - - divided into a number of disciplines, such as physics, chemistry, mathematics, medicine, social science, etc. Let $G_i = \{ g \}$ be a set of stimulators in the ith scientific discipline D_i and let $D = D(D_1, D_2, \ldots, D_n)$ be an ordered set of scientific disciplines. Associated with it is a set of nonnegative real numbers $p_i(t)$, $i = 1, 2, \ldots, n$, such that

$$0 \leq p_i(t) \leq 1, \quad \sum_{i=1}^{n} p_i(t) = 1.$$

These numbers are interpreted as the relative intensity of scientific "incitence" or stimulation in each of the scientific disciplines under consideration.

Further let $0 \leq p_{ij} \leq 1$ be another set of non-negative real numbers which represent the relative intensity of "incitence" which stimulators in the disciplines D_i exert on the scientific activity in the discipline D_j.

Under the assumption that scientific progress can be modeled as a first-order Markov process, it is then completely defined by the initial distribution, which we shall write in vector form as

$$p(0) = [p_1(0), p_2(0), \ldots, p_n(0)]$$

$$p_i(0) \geq 0, \quad \sum_{i=1}^{n} p_i(0) = 1$$

and by the matrix of transition probabilities

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix}$$

such that

$$\sum_{j=1}^{n} P_{ij} \geq 0$$

$$i, j = 1, 2, \ldots, n.$$

Application of this Markov chain model of science development has been demonstrated on a sample of citation data provided in an article by Earle and Vickery [2]. Our analysis was limited to the subset of data which gives the count of social science subjects only. The sample population consisted of 13,412 citations from 897 books and periodicals, of which 256 books and 75 periodical titles were classed as social science items. Analysis of the data in terms of the proposed model showed a clear tendency of shifting emphasis of scientific inquiry from science and technology to social science subjects [3].

However, although the conclusions obtained from the model seem reasonable and intuitively founded, the validity of the model can be best verified by future data.

References

1. Cole, P. F. "Journal Usage Versus Age of Journal." *J. Documentation,* 19:1 (1963).

2. Earle, P., and Vickery, B. "Social Science Literature Use in the U.K., as Indicated by Citations." *J. Documentation,* 25:123 (1969).

3. Zunde, P., and Slamecka, V. "Predictive Models of Scientific Progress." To appear in *Information Storage and Retrieval.*

Extended Effects of Information Processes and Processors

V. Slamecka, L. Chiaraviglio, W. T. Jones

The purpose of this project has been to obtain theories of information processes and phenomena that will furnish some clues as to the possible development of knowledge in an environment which contains an increasing number of artificial processors and other knowledge amplifying facilities. We are particularly interested in the pathologies that could be engendered by such amplification of the information processing capabilities of mankind.

The project is conceived as having two phases. The first phase, and the one in which we have been engaged up to this point, is concerned with studying the phenomena of information; the goal of this phase is to obtain phenomenal theories of the information processes. The second phase of the project has as a goal the development of theories of the causal connections that underlie the observed processes. At present there exist both causal and phenomenal theories which are able to account for some selected aspects of the information processes.

A variety of statistical models have been employed to achieve descriptions of some of the important aspects of the phenomena of information. Mathematical models originally developed for the purpose of accounting for epidemic, branching and neural net processes have been adapted to the description of some information processes [1-3]. These models are probably the most sophisticated theories that we have for dealing with some of the phenomena of information spread and growth.

It seems to us that many of the information processes of interest are analogous to the processes studied by geneticists and population biologists. The success obtained in adapting epidemic, branching and neural net models to information processes is evidence that the analogies may be fundamental. Indeed, these models can be viewed as special cases of some of the models employed by biologists.

In the epidemic and neural net models, the factor transmitted and the connection made are taken to be of a single type. The population of individuals that results at any stage of these processes is classified as a function of this single type of factor. Branching models are able to account for any finite set of types of factors. At any stage of the process we can obtain for each individual in the population the probabilities that the individual will have received any one of the factors in the set. The three types of models seem to be special cases of the models used in genetics. Epidemic and neural net models correspond to genetic models in which only one genetic factor is involved. Branching models correspond to genetic models which involve a set of unlinked factors. None of these models takes into account either linkages among the factors or selective pressures that act differentially on them.

We believe that it is worthwhile to try to adapt some of the concepts of genetics to information processes. Historical data is accumulating in the areas of the growth of science, the deployment of citations and other bibliometric processes. As a first step in the adaptation, it seems reasonable to develop for this type of data the information analogue of the concept of phenotype. For example, in the case of citation data, analogues of genetic maps would seem to be intuitively meaningful and not too hard to construct. With such "information maps" it may be possible to discriminate populations which with respect to the deployment of citations over time act in strict analogy to biological populations. This would enable us to give the concept of populations that share information (citations, in this example) a precise content.

One of the difficulties of the study of information processes is the construction of experiments that will illuminate the historical data we have. It may be noted that there was a time when biology was largely composed of natural history and there were relatively few experimental systems that could be used to illuminate the observed processes. Information science is presently in a similar situation. Genetic concepts allowed biology to go beyond natural history. A set of concepts must be found which will serve the same purpose in information science.

The development of a theory of information processes may parallel the development of biology. Our first need is a phenomenal theory of the processes that will focus our attention on the phenomenal invariants. Biology took great strides towards genuine causal theories with the classical work that associated physiological correlates to the phenomenal genetic markers. The recent advances in biochemical genetics have completed the main outlines of a causal theory of genetics. These advances were made possible by the highly developed phenomenal theories. Similarly, causal theories of information may have to wait for more highly developed phenomenal theories before significant bridges can be made to the "physiology" and "chemistry" of information processors.

There exists a body of knowledge concerned with the internal morphology and microprocesses that constitute many artificial and natural processors. In the case of artificial processors we have the obvious facility of accessing their "innards" to a degree that is still unachievable for most organisms. Yet there is not much that can be said about how the constituent microprocesses and morphology of processors give rise to the observed information phenomena. It is our opinion that the lack resides on the side of the analysis of information phenomena. Thus as a first step we are aiming to construct theories that will give us more powerful tools for the analysis of phenomena.

References

1. Goffman, W., and Newill, V. A. "Communication and Epidemic Processes." *Proceedings of the Royal Society*, 298:316-334 (April 1967).

2. Bellman, R. (ed.) *Multitype Branching Processes*. New York, American Elsevier, 1971.

3. Rapoport, A., and Rebhum, I. "On the Mathematical Theory of Rumor Spread." *Bull. Math. Biophysics*, 14:375-383 (1952).

Automatic Classification of Indexed Monographs

P. Zunde, P. J. Zando

The hypothesis which has been under investigation is that decisions involved in classifying monographs according to the Library of Congress or some other comparable classification scheme can be made on the basis of their internal (book) indexes and that a computer procedure can be designed to perform this task automatically.

In the first phase of the research, one class was selected from each of four diverse subject areas of the Library of Congress classification schedule, namely

(1)	MATHEMATICS
	Algebra
QA 266	Abstract Algebra
(2)	ECONOMIC THEORY
HB 501	Capital. Saving.
(3)	TRANSPORTATION AND COMMUNICATION
	Railways. United States
HE 2751	History. Statistics
(4)	PHILOLOGY. LINGUISTICS
	Language: General Works. Introduction. Philosophy. History. Comparative Philology.
	Science of Language. Origin, etc., of Language.
P 121	General Works. Methodology.

Random samples of three books were selected from each of the above categories and their indexes were compared for occurrence of identical terms. As a result of this comparison, similarity and analysis of variance values were calculated both within each of the four classes and between the samples of these classes.

In the second phase of research, additional classes were chosen such that for each one of the original classes, two more classes from the same subject areas were chosen as close as possible in subject matter coverage to the subject matter of the original classes within the LC classification system (except for TRANSPORTATION and COMMUNICATION, in which area only one additional class was chosen).

The additional classes were:

(1A) QA 266	MATHEMATICS Matrices
(1B) QA 267.5	MATHEMATICS Machine Theory. Abstract Machines Special types, A-Z.
(2A) HB 221	ECONOMIC THEORY Price. Theory of Price.
(2B) HB 601	ECONOMIC THEORY Profit. Income.
(3A) HE 2791	TRANSPORTATION AND COMMUNICATION Railways. United States By railroad (or company), A-Z.
(4A) P 105	PHILOSOPHY. LINGUISTICS Language: General works, etc. Science of Language, Philosophy, etc. 1860/80-
(4B) P 123	PHILOSOPHY. LINGUISTICS Language. General works, etc. Science of Language, Philosophy, etc. General Special (e.g. Analogy)

As in the first phase, random samples of three books were taken from each of these additional classes and their indexes compared for similarity both within and between the classes. Measures of similarity were then calculated for each pair of books.

It was concluded that enough variance was present between indexes of books of even closely related classes and that the similarity (or association) measures based on the co-occurrence of identical terms can be effectively used for discrimination purposes.

In the third and last phase of research, a profile for each of the eleven classes (1), (1A), (1B), (2), (2B), (3), (3A), (4), (4A), and (4B) was constructed by cumulating the indexes of three sample books in each class and weighing the individual terms by the frequency of their occurrence in the indexes. It was hypothesized that those profiles are valid intensive descriptions of the above-named classes of the Library of Congress classification system and that these descriptions are distinct and discriminative enough to be used as a decision criteria to assign books in terms of "best match" to LC classes by comparing their indexes with these class profiles. The hypothesis was tested by arbitrarily selecting one book (not used in previous samples) from each of the eleven classes listed above and by using discriminant analysis to obtain similarity measures and confidence limits for maximum likelihood decision to assign those books to appropriate classes. The procedure is easily adaptable to a full-scale computer processing and automation of classification.

The results of the test validated the hypothesis to a highly satisfactory degree.

References

1. U.S. Library of Congress. Subject Cataloging Division. *Classification.* Washington, D. C., GPO.

On Scientific-Technical Tape Information Services

V. Slamecka, J. Gehl

The purpose of this study was to trace the outline of some of the main features of scientific-technical tape services which have developed during recent years, and to exhibit commonalities and variations of characteristics of those services. The data base utilized for this analysis was provided by Kenneth D. Carroll's *Survey of Scientific-Technical Tape Services* [1].

The principal sponsors of the fifty-five active, commercially available services listed in the Carroll report are learned and professional societies, publishing firms, and commercial organizations. Virtually all of the organizations involved use their data bases to produce one or more publications; these are bibliographies, indexes, abstracts, thesauri, keyword supplements, patent review books, data books, or similar products under different names.

The subjects covered by the services span almost the entire range of scientific knowledge. (However, coverage is not equally balanced from subject to subject; chemistry and chemical engineering, for example, are specifically covered by eleven different tape services.) The rather crude measuring unit developed for determining the amount of information provided by the services is the number of items cited per tape. Of the total number of services for which information on this question is available, approximately one half cite more than 5,000 source items on each tape. Two of these in fact cite more than 20,000 such items. One is ICRS, the Index Chemicus Registry System tape, which cites 4,000 abstracts and 17,000 Wiswesser Line Notations on each monthly tape, for a total of 21,000 items; the other is Predicasts Corporation's F&S Index of Corporations and Industries, which includes approximately 25,000 source citations on each of its quarterly tapes.

Considering the wide variety of topics covered by these tape services, it is not surprising to find that the number of tapes issued each year is quite different from service to service. Virtually every conceivable time interval is represented -- weekly issues, three issues a month, biweekly issues, semimonthly, monthly, eleven issues a year, quarterly, every four months, semiannually, and annually.

Combining the information available on both the average number of source items cited on a tape, and the frequency of tape issues, it may be concluded that almost half of the services cite more than 25,000 source items annually. Of this group, seven cite more than 200,000 items annually, and of those seven, there are two which cite more than 300,000.

Of course, the cost of all this information is not always low. However, more than 75% of the services are offered at annual costs of $2,500 or less.

A large portion of current data bases are devoted to coverage of the journal literature. More than three out of four services are such that 50% or more of their data bases are devoted to journal coverage, and almost two out of three are such that journal coverage accounts for at least 80% of their total data base volume; a quite large percentage of this journal coverage is accounted for by English-language literature, and only one of the services surveyed indicated that its data base is predominantly (i.e., more than 50%) in a language other than English.

Approximately one out of three of the scientific-technical services for which information is available indicated that at least some part of their data base is devoted to coverage of the reports literature, but only three out of twenty devoted more than 10% of their data base to such coverage.

Also, nine services devote at least 25% of their data base to patent literature coverage; of those nine, there are four which are devoted exclusively to that purpose. Finally, three of the scientific-technical tape services are devoted exclusively to the coverage of papers presented at conferences, and the data bases maintained by three others of the services are devoted to statistical or historical data.

Techniques for searching the various data bases differ considerably from one tape service to the next. Beyond such standard items as author, title, and basic bibliographic information, most services allow searching of the data base in various other ways. Thus, searchable data elements for one or more services include: descriptors (with or without links and roles); keyword phrases; words in a document's abstract; the language in which a document is written; primary and secondary subjects of a document; indexing terms and title enrichment terms; and classification codes.

With reference to approximately one out of three presently available tape services, a tape subscriber will be required to develop his own software. In such a case, the subscribing institution will use the tapes strictly as additional input to its own system, and will use its own software and its own search strategy. However, in the remaining cases, the institution which produces the tape either already offers supporting software to its subscribers, or will develop whatever software is required for an interested customer. Also, some tape services have indicated that, although they do not themselves offer supporting software, various suitable search programs are available elsewhere on the commercial market. In addition, a number of the organizations producing scientific-technical tapes now offer, or are planning to offer, in-house search services -- retrospective, SDI, or both.

Of course, the premise which underlies the utility and validity of any comparative survey such as the one resulting from this study [2] is the necessity and sufficiency of the parameters in terms of which such comparisons are made. However, the premise may be unjustified; for we do not know whether the parameters of comparison are useful for either of the two major clients interested in surveys -- those attempting to select the best service for their needs, and those seeking to pool several tapes for a wider and more efficient service. Nor do we have any evidence that a much larger number of parameters (such as prepared by Schwartz [3]) can be employed to construct a decision-making algorithm for either category of potential users, even if one assumed the unlikely situation that such detailed descriptions of data bases can be obtained and made public.

Thus, while paying attention to monitoring the characteristics of tape services, perhaps even more attention should be given to the idea of surveying the customers themselves, actual and potential. A great deal more respect should be paid to the experience and recommendations of such customers, and to the fact that in proliferating diversity of technical design we are concerned with the management of information as an important national resource.

References

1. Carroll, K. D. *Survey of Scientific-Technical Tape Services.* American Institute of Physics, September, 1970.

2. Gehl, J., and Slamecka, V. "A Profile of Scientific-Technical Tape Information Services." To appear in *Proceedings of the NBS/COSATI Forum on Information Analysis Centers,* National Bureau of Standards, Gaithersburg, Md., May 1971.

3. Schwartz, J. "A Check List for the Examination of Data Base Systems." New York University, 1970. (Mimeographed.)

Automated Structuring of Natural Language Text

M. Valach

Text structuring is a process by which the words and concepts used in a given text T are crosslinked using relations given by both the grammar and the semantics of T. The following steps are essential to this process: recognition of the words and concepts of T; parsing and syntactical analysis of each sentence; and recognition of intersentence relations - - which is the basis both for concept extraction (identification) and for development of text structure linkages.

There are two points of view from which a text must be interpreted as a structure - - i.e., (1) the writer's viewpoint and (2) the reader's viewpoint.

1. It is the writer of the text who is transmitting his information via text. The structure of the text imposed by the writer has to satisfy various requirements which in turn depend on various circumstances. To name a few:

(a) To communicate the content of the text in a structure that comprises (directly or indirectly) all relations the author wants to give, where the elements of the structure are properly described, grouped, and mutually related.

(b) To relate his information to the knowledge of the assumed reader, so that the reader can more easily add the new information to whatever knowledge that reader already has.

(c) To provoke "side effects" in the reader, such as by arousing his curiosity, or providing satisfaction, or creating suspense, anxiety, or other emotional reactions.

Thus, the writer structures the text according to:

* What he wants to communicate;
* Who he assumes the reader to be; and
* How he wants to present his information.

2. A second interpretation of the text structure takes place from the reader's point of view. Indeed, each and every reader may interpret the text as being structured in a different way. A few of the factors influencing the interpretation of the text by the reader are:

* What he already knows about the subject;
* The type of person he is;
* How much confidence he has in the writer;
* His motivation for reading; and
* The mood he is in at the time of the reading.

The role of the reader is not only to recover the information content of the text as it is given by the meanings and relations among the words and concepts, but also to project or relate the content to his own knowledge, and this task comprises a substantial part of the process of interpreting the text. Thus, the writer's structure of the content is modified during this process by the reader's knowledge and the reader's habits of learning, and so the procedure for recovering the content of the text is strongly monitored by the reader.

Having this broad picture in mind, it can be seen that there is not just one simple structure into which a text can be converted by the text-structuring process; instead, text structuring is merely one step in the interpretation (understanding) process, a step resulting in a representation of the textual relations that can form a preprocessed data base for later interpretative processes. Yet the structuring itself has to be done with some particular purpose in mind indicating later use of the resulting structure.

It would therefore seem clear that the main goal should be to make the resulting structure of the text rich and complete enough for as broad a subsequent use as possible. The structure then becomes a new data base for the content-interpretation algorithms, which can be modified or monitored according to the goals of the interpretation rather than being restricted to more special-purpose structures at the beginning.

The text interpretation processes normally start with word classification and continue in the following steps:

a. Word recognition and lexical descriptions;
b. Sentence parsing;
c. Syntax of the sentence;
d. Context of the sentence;
e. Concepts in the text; and
f. Interpretation of the whole text in the context of the interpreter's knowledge.

In other words, the scheme looks as follows:

WORDS → PARSING → SYNTAX → CONTEXT → CONCEPTS → INTERPRETATION (1)

It is well known that in analyzing the relations and structures at a lower level - - higher levels being in the direction of the arrows in (1) - - ambiguities can be found where the clue to their solution lies in higher levels. It seems to us to be very important to realize that the analysis made at a lower level is not sufficient if it results in merely the "most probable" structure, if in fact more than one structure is possible. Our philosophy is that all possible alternatives unsolved at the lower level should be retained until such time as the clue justifying either their acceptance or their rejection is found. This philosophy of approach provides the motiviation for certain requirements we have established for our algorithms.

During the past year we developed the so-called Q-graph technique for parsing of sentences. The possible functional classes of the words in the sentence are matched with the Q-graph. The technique finds the parsing of the sentence - - including any existing ambiguities (alternative interpretations). The matching process, a quite simple and fast one, can also be used as a model for interpretation of incomplete sentences. The model clarifies the basis for expectation, suspense, and satisfaction in the listener and diagnoses how the classification of the analyzed part of the sentence needs to be rebuilt if a wrong assumption (wrong path in the Q-graph) was followed.

The program for the parsing of simple sentences was extended during the last year to the parsing of complex sentences which included relative sentences (clauses) embedded at different levels. The program recognizes the structure of the complex sentence, including the relative clauses, assigns a number to each sentence and outputs the corresponding tree of the numbered sentences.

Another extension of the program which was developed transforms a given sentence of the type described into an equivalent set of simple sentences. Thus, Figures 1, 2, and 3 show an example of a sentence comprising five relative clauses. In Fig. 1, the analyzed sentence is shown in the second column; the fourth column shows the vocabulary lookup allocation of possible functional classes. For example, the word "coast" in line 22 can function either as a noun singular (noun s), as an adjective (adj) or as a verb (infini). The word "living " in line 14 can function either as a noun (noun s), as an adjective (adj) or as a present participle (-ing).

Figure 2 shows the analyzed sentence after parsing into an equivalent set of simple sentences. The second column contains numbers showing in which group of the Q-graph the word was found; the third column shows the parsing; the fourth column shows the number of the simple sentence to which the corresponding word belongs. The remainder of the figure shows the analyzed sentence, where the words are shifted into different columns placing all words of the simple sentence into the same columns. For example,

1	THE	1	THE		
2	PICTURE	2	NOUN S	ADJ.	INFINI
3	WHICH	5	RELACC	RELNOM	
4	WAS	7	WAS		
5	GIVEN	8	PAST P	PAST	
6	TO	10	PREP.	TO	
7	MY	12	ADJ.	PRONOU	
8	BROTHER	14	NOUN S	ADJ.	INFINI
9	WHOM	17	RELACC		
10	MY	18	ADJ.	PRONOU	
11	FATHER	20	NOUN S	ADJ.	INFINI
12	WHO	23	RELNOM		
13	IS	24	IS		
14	LIVING	25	NOUN S	ADJ.	-ING
15	IN	28	PREP.		
16	CALIFORNIA	29	NOUN S	ADJ.	INFINI
17	WHICH	32	RELACC	RELNOM	
18	IS	34	IS		
19	ON	35	PREP.		
20	THE	36	THE		
21	WEST	37	NOUN S	ADJ.	INFINI
22	COAST	40	NOUN S	ADJ.	INFINI
23	SENT	43	PAST		
24	THE	44	THE		
25	LAST	45	NOUN S	ADJ.	INFINI
26	MESSAGE	48	NOUN S	ADJ.	INFINI
27	WAS	51	WAS		
28	PLACED	52	PAST P	PAST	
29	IN	54	PREP.		
30	THE	55	THE		
31	OFFICE	56	NOUN S	ADJ.	INFINI
32	WHICH	59	RELACC	RELNOM	
33	BELONGS	61	3PERS.		
34	TO	62	PREP.	TO	
35	CHARLIE	64	NOUN S	ADJ.	INFINI
36	.	67	.		

Fig. 1

1	1	THE	1	THE
2	1	NOUN S	1	PICTURE
3	5	RELNOM	2	WHICH
4	2	WAS	2	WAS
5	2	PAST P	2	GIVEN
6	3	PREP.	2	TO
7	3	ADJ.	2	MY
8	3	NOUN S	2	BROTHER
9	5	RELACC	3	WHOM
10	1	ADJ.	3	MY
11	1	NOUN S	3	FATHER
12	5	RELNOM	4	WHO
13	2	IS	4	IS
14	3	NOUN S	4	LIVING
15	3	PREP.	4	IN
16	3	NOUN S	4	CALIFORNIA
17	5	RELNOM	5	WHICH
18	2	IS	5	IS
19	3	PREP.	5	ON
20	3	THE	5	THE
21	3	ADJ.	5	WEST
22	3	NOUN S	5	COAST
23	2	PAST	3	SENT
24	3	THE	3	THE
25	3	ADJ.	3	LAST
26	3	NOUN S	3	MESSAGE
27	2	WAS	1	WAS
28	2	PAST P	1	PLACED
29	3	PREP.	1	IN
30	3	THE	1	THE
31	3	NOUN S	1	OFFICE
32	5	RELNOM	6	WHICH
33	2	3PERS.	6	BELONGS
34	3	PREP.	6	TO
35	3	NOUN S	6	CHARLIE
36	6	.	6	.

Fig. 2

1	1	THE	1	THE
2	1	NOUN S	1	PICTURE
3	5	RELNOM	2	WHICH
4	2	WAS	2	WAS
5	2	PAST P	2	GIVEN
6	3	PREP.	2	TO
7	3	ADJ.	2	MY
8	3	NOUN S	2	BROTHER
9	5	RELACC	3	WHOM
10	1	ADJ.	3	MY
11	1	NOUN S	3	FATHER
12	5	RELNOM	4	WHO
13	2	IS	4	IS
14	2	-ING	4	LIVING
15	3	PREP.	4	IN
16	3	NOUN S	4	CALIFORNIA
17	5	RELNOM	5	WHICH
18	2	IS	5	IS
19	3	PREP.	5	ON
20	3	THE	5	THE
21	3	ADJ.	5	WEST
22	3	NOUN S	5	COAST
23	2	PAST	3	SENT
24	3	THE	3	THE
25	3	ADJ.	3	LAST
26	3	NOUN S	3	MESSAGE
27	2	WAS	1	WAS
28	2	PAST P	1	PLACED
29	3	PREP.	1	IN
30	3	THE	1	THE
31	3	NOUN S	1	OFFICE
32	5	RELNOM	6	WHICH
33	2	3PERS.	6	BELONGS
34	3	PREP.	6	TO
35	3	NOUN S	6	CHARLIE
36	6	.	6	.

Fig. 3

"The picture was placed in the office" is sentence number 1, while "whom my father sent the last message" is relative clause number 3.

Figure 3 shows alternative parsings of the sentence. The word on line 14 has possible parsings as a noun, adjective, or present participle. Acceptance of anyone parsing has to be determined by some knowledge found only outside of the sentence itself. Thus, consider the sentence:

Happiness is living in California

From this example we see that, if the subject is a state of the mind, then "living" functions as a noun. In constrast, if the subject is the name of a person, then "living" functions as a present participle. In our original sentence it is a knowledge of the meaning of the word "father" which is the out-of-sentence clue to the proper interpretation.

Accomplishments

The name of the programming package which materializes our approach to text structuring is *English Text Processing (ETP)*. The ETP package is designed as a four-part package containing SEN, KNO, TEX and MOD, which will be described below.

Table 1 shows the full names of the packages and their interior groupings, including the names of corresponding subprograms. Asterisks mark those subroutines which have already been written, debugged and tested. Plus-signs indicate subroutines which have not yet been programmed but which have already received some consideration in our general concept of text processing.

Figure 4 shows how the different parts of ETP are mutually related. It is assumed that all of these use the same data base, with each contributing to its growth in its own way.

Table 1

SEN Package

SENTAL

ESR(*)	English Sentence Reader
ESVL(*)	English Sentence Vocabulary Lookup
GRAPAR(*)	Q-graph Parser of English Sentence

SENSTR

PST(*)	Parsing to Synthax Transformation

SENIN

SSS(+)	Simple Sentence Separation

TEX Package

TEXAL

ISEL	Intersentence Linkup (resulting from strictly grammatically worked relations)
AMEL	Amibiguity Elimination (resulting from textual relations)

TEXSTR

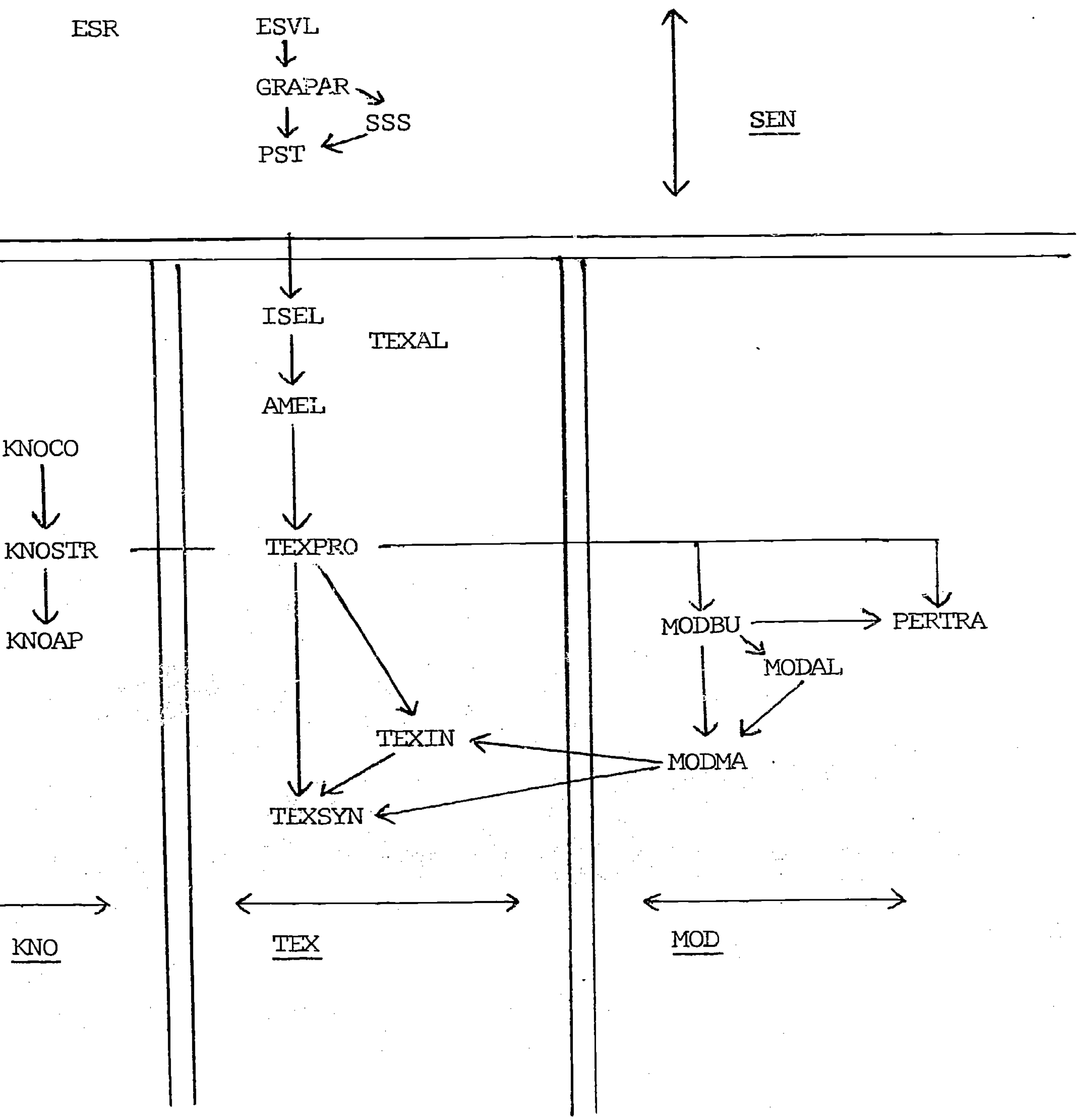
TEXPRO	Text to Knowledge Projection
TEXIN	Text Interpreter
TEXSYN	Text Synthax from Model (+) observation
TRAPER(+)	Transformation from One Person to Another

MOD Package

MODBU	Model Builder
MODAL	Model Analyzer
MOMA	Model Manipulation

KNO Package

KNOCO	Knowledge Collection
KNOUP	Knowledge Update
KNOSTR	Knowledge



ESR
ESVL
GRAPAR
SSS
PST
SEN
ISEL
TEXAL
AMEL
KNOCO
KNOSTR
TEXPRO
KNOAP
MODBU
PERTRA
MODAL
TEXIN
MODMA
TEXSYN
KNO
TEX
MOD

Fig. 4

The ESR subroutine (English Sentence Reader) was written, debugged and used in various other programs in which English Sentences are inputted for analysis. ESR separates words and markers for vocabulary lookup and further formating of the sentence items. It reads sentences from tape or cards.

The ESVL subroutine (English Sentence Vocabulary Lookup), incorporated into GRAPAR, performs indexical classification of sentence items (words and punctuation marks), identifying possible classes in which the particular word (in the actual form used in the sentence) may function. ESVL is a preprocessing subroutine for the Q-graph parsing method, used in GRAPAR.

GRAPAR (Q-Graph Parser for English Sentence) accomplishes the following:

1. Parses each sentence of the text from its beginning to the next period:
2. Allocates a number to each simple sentence (clause) that is part of a complex sentence or that is a relative embedded sentence;
3. Constructs the tree structures in which each node corresponds to one sentence number (allocated under 2) showing the structure of simple and relative embedded sentences of the analyzed sentence; and
4. In case there are more possible parsing alternatives (i.e., ambuiguity resulting from different possible parsings) the analysis processes each alternative equally (as described in points 1,2 and 3).

GRAPAR is the main subroutine of the group SEN which parses sentences by using the Q-graph technique [1]. A Q-graph is stored in the corresponding arrays in the memory and is considered to be part of input data for the GRAPAR subroutine.

PST (Parsing to Syntax Transformation), a subroutine that transforms a parsed sentence into syntactical structure (more precisely, into a diagram of the sentence SD) is under development. A special feature of PST that is being attempted is one which would derive the diagram of the sentence DS by transformation of the Q-graph into a corresponding sentence diagram graph called an SD-graph. If a successful, content-independent transformation can be found (as hoped), then very simple, straightforward, and fast rules will be the result. At this time, however, the Q to SD transformation is not yet completed and will require further research effort.

SSS (Simple Sentence Separation) subroutine processes the tree structure T of simple sentences (including relative sentences) outputted from GRAPAR. The end result is the set ST of simple sentences comprised in the original sentence OS.

The set ST of simple sentences together with the tree structure T contains all information necessary for the recovery of the originally analyzed sentence OS. Therefore ST and T are considered to be a result of equivalent transformation of the original sentence OS. The program demonstrates that both of the following transformations are possible - - either:

$$OS \rightarrow T, ST \qquad \text{(automated)}$$

or:

$$T, ST \rightarrow OS \qquad \text{(not yet automated)}$$

PERTRA (Narrating Person Transformation). A preliminary study has been conducted to collect the rules which govern the transformation of narrative text from one person to another. The situation is as follows: Having a text, narrated by one person P1, replace the person by some other person P2 from the text and have the text transformed as being narrated by P2. The interesting aspect of the situation is that the transformation requires what we have called a "model of the situations" brought out by the content of the text: The model helps to determine the proper relations between P2 and other described objects which are to be described or also transformed under the changed situation.

A table has been established for grammatically correct context-free transitions, and progress on the study continues.

In conclusion, we will recite some particularly interesting topics which are related to the described research and which offer excellent subjects for theses, seminars, or articles:

* Need of a "situation model" for text interpretation
* Two-step parsing using Q-graph
* Proposal for measuring the suspense created by the unfinished part of the sentence (using Q-graph approach)
* Development of patterns for learning the structure of the English sentence using Q-graphs
* Q-graph subgraphs as simplified English for the man-computer interface
* Change of the person narrating the same text
* Problem of the text syntax, created from the observation of the behavior of described subject
* Sentence as a vector of independent components, and its role in the text structuring process

References

1. Valach, M. "A Q-graph Approach to Parsing." To appear in: *Proceedings of the Conference on Linguistics*, University of Iowa, October, 1970.

Study Of Concept-Based Grammars

D. Rogers

The purpose of this study has been to provide insight towards synthesis (e.g., automatic abstracting) of natural-language information. That the internal structure of the sentence or smaller unit depends on the organization of the entire text has become increasingly evident to linguists (e.g., Zellig Harris, Kenneth Pike, and Robert Longacre). This study has partially analyzed the following interdependent features of a text: topics and comments, analyses of verbal activities, nominalizations of verbal activities, references to events and references to nominals, and the identification and range of various types of modalities.

In a text, a topic may be loosely defined as something being focused on (i.e., a subject) and a comment may be defined as a predication of that topic. For example, in an active-sentence construction such as the government recalls the ambassador the topic is the agent the government and the comment about the government is the predicate recalls the ambassador. On the other hand, in a passive-sentence construction such as the ambassador was recalled by the government, the topic the ambassador is the object of the verbal activity and the comment was recalled by the government contains within it the lexical expression of the agent of the verbal activity.

We should wish that a grammar adapted to synthesis of natural language information be at least able to recognize topic and comment in terms of the placement of agent and object markers. We can demonstrate the working of such a grammar in respect to the following, which represents a further analysis of the above two sentences.

	TOPIC	COMMENT		
(1)	the government	recall	s	the ambassador
		{agent } {object}	(present) [agent]	[object]

	TOPIC	COMMENT		
(2)	the ambassador	is recall	ed	by the government
		{ agent} { object}	(present) [object]	[agent]

In this notation, the verbal primitive recall (or, more precisely, the verbal root call with the particle re- "back") is marked in the lexicon[1] with the possibility of opening slots for the grammatical expression (marker) of the agent and of the object. This is signified in the notation by the braces around agent and around object beneath recall. Thus, this, the first stage in the explanation of either (1) or (2) is the following:

(3) VERBAL ROOT

recall
{ agent}
{ object}

A set of rules concerning verbal roots allows selection of a particular affix (form) to accompany the verbal root. However, the rule placing an affix also specifies a semantic interpretation on that affix. One of a set of modal affixes expressing semantic features such as various tenses, possiblity, necessity, question, condition, permission, command, etc., is obligatorily placed after the verbal root. In the analyses of both (1) and (2) a marker, let us say L1, is placed after the verbal root in (3) with the semantic interpretation of reference to the present. This reference is indicated in the notation by parentheses:

(4) VERBAL ROOT

recall	L1
{ agent}	(present)
{ object}	

This placement specifies (a) the modality present associated with the verbal activity involving the verbal root recall and (b) the potentiality for opening up a slot (or position) for the lexical representation of that feature (e.g., today, now, etc., as opposed to yesterday).

1. This may be considered analogous to the lexicon marking in Fillmore, Charles J., "The Case for Case," *Universals in Linguistic Theory*, ed. by Emmon Bach and Robert T. Harms (New York: Holt, Rinehart and Winston, 1968), pp. 1-88. However, an alternative analysis may be that the rules defining such entities as agent, object, recipient, etc. are pragmatic in nature.

To this point the analyses of both (1) and (2) are the same. However, a choice may be made to replace the marker L1 by s, a member of a set of finite-verb endings, to grammatically express (indicated in the notation by square brackets) the third-person singular agent as in (5) below or by ed, also a member of the set of finite-verb endings, to grammatically express a third-person singular object as in (6) below.[2] In both instances the reference to the present tense is retained.

```
(5)  VERBAL ROOT
     recall          s
     { agent}        (present)
     { object}       [agent]
                     [3rd person]
                     [singular]

(6)  VERBAL ROOT
     recall          ed
     { agent}        (present)
     { object}       [object]
                     [3rd person]
                     [singular]
```

In the string (6) above the combination of (present), [object], [3rd person] and [singular] obligatorily cause the placement of an is before the verbal root:

```
(7)  VERBAL ROOT
  is recall          ed
     { agent}        (present)
     { object}       [object]
                     [3rd person]
                     [singular]
```

Because the verbal root in the above is followed by a finite-verb ending, the string recalled is named a word. Likewise, the string in (5) is named a word because it ends in a finite-verb ending.

2. Person and number are specified by sets of rules operating together to select (or analyze) an affix.

Because the form s has been placed to grammatically express the agent in (5), the lexical representation of the agent must be placed in a slot before the verbal unit:

(8)	the government	recall { agent} { object}	s (present) [agent] [3rd person] [singular]

We note that in the above the verbal root is marked with the possibility of opening not only a slot for an agent but also a slot for an object. Because a form has not been placed to express the object in (8), the lexical representation of the object is placed in a position after the verbal unit as in (9) below. We note that now both the possibility for opening a slot for the grammatical expression of the agent (i.e., the slot s of recalls as opposed to the lexical representation the government) and the possibility for opening a slot for the grammatical expression of the object (i.e., the slot after the verbal unit) associated with the verbal root have been satisfied. This completes the explanation of (1).

(9)	the government	recalls {agent} {object} (present) [agent] [3rd person] [singular]	the ambassador [object]

Because the form ed has been placed to express the object in (7), the lexical representation of the object must be placed in a slot before the verbal root:

(10)	the ambassador is	recall {agent } {object }	ed (present) [object] [3rd person] [singular]

Because a form has not been placed to express the agent in (10), the lexical representation of the agent with a preceding by is placed in a position after the verbal unit. Thus, the following completes the explanation of (2):

(11)	the ambassador	is	recalled	by	the government
			{agent} {object} (present) [object] [3rd person] [singular]		[agent]

In both (1) and (2) the topic is defined as that entity, involved in the verbal activity, which has been expressed by the finite-verb form placed immediately after the verbal root and lexically placed in a position before the verbal unit. If, instead of selecting the finite-verb form <u>s</u> in (5) or <u>ed</u> in (6) to replace the <u>L1</u> of (4), a non-finite verb ending, e.g., <u>ing</u> with reference to an agent, is selected, we obtain (12) instead of (6). The rule retains the reference to the present specified by the placement of the affix <u>L1</u>.

(12) <u>VERBAL ROOT</u>

recall {agent} {object}	ing (present) (agent)

Because neither a reference to the object nor the grammatical expression of the object has been placed, the lexical representation of the object is placed in a position after the verbal unit as in (13) below.

(13) <u>VERBAL UNIT</u>

recall {agent} {object}	ing (present) (agent)	the ambassador [object]

Because the reference to the agent has been placed (i.e., in the replacement of <u>ing</u> for <u>L1</u> in (12)), the lexical representation of the agent must be placed in a slot before the verbal unit as in (14) below.

(14)	the government	recalling	the ambassador
		{agent} {object} (present) (agent)	[object]

We note that in the above structure the topic is formally defined as that entity, involved in the verbal activity, which has been expressed by the form placed immediately after the verbal root and lexically represented in a position before the verbal unit. The structure (14) may be placed in initial position where the string the government in (14) is the lexical representation of an entity involved in another verbal activity having a structure similar to (1) or (2). If the form placed immediately after the verbal root, e.g., declare, of this other activity expresses the function of the lexical representation the government in this other activity, the string (14) is placed in initial position with appropriate commas as in (15) and the string the government is the topic of both verbal activities.

		VERBAL ROOT	
(15)	the government, recalling the ambassador,	declare	s
		{agent} {object}	(present) [agent] [3rd person] [singular]
	war [object]		

However, if the form placed immediately after the verbal root declare grammatically expresses the object, then the structure (14) is placed with a preceding by after the verbal unit declared as in (16) below. This provides for two topics, the war and the government, with the additional information that the second topic is included within the comment about the first topic.

(16)	TOPIC	COMMENT			
					TOPIC
	war	is declare	d	by the	government,
		{agent} {object}	(present) [object] [3rd person] [singular]	[agent]	
	COMMENT				
	recall	ing	the ambassador		
	{agent} {object}	(present) (agent)	[object]		

Besides formally defining topics and comments, this procedure yields analyses of verbal activities expressed in surface strings. Such analyses, stated not in terms of surface strings, permits the recognition of the same verbal activity repeated as different surface forms in other portions of the text.

The grammar that is capable of handling natural language text in the way demonstrated above is constituted by a set of conditional rules and meta-rules concerning the order of the selection of the conditional rules. These conditional rules place the potentialities of the verbal root, such as the modal characteristics tense, possibility, necessity, question, condition, etc., and the grammatical expressions of the entities agent, object, recipient, etc., involved in the verbal activity. The meta-rules concerning the order of selection of conditional rules control the various manifestations of the verbal roots and nominal entities, thereby monitoring the deployment of meaning on the surface.

As can be seen by the above description, such a grammar would provide a facility which has the capability to synthesize information carried by natural language text. Such a facility is able to associate the relevant semantic interpretations with the surface strings of the text.

References

1. Harris, Z. S. "Discourse Analysis." *Language*, 28:1-30 (1952).

2. Harris, Z. S. "Linguistic Transformations for Information Retrieval." In: *Proceedings of the International Conference on Scientific Information (1958)*, Vol. 2. National Academy of Sciences - - National Research Council, Washington, D. C., 1959. pp. 937-950.

3. Pike, K. L. "Four Substitution Devices for Studying the Relation Between the Logical and the Grammatical Structures of Discourse." Paper presented at 1971 Summer Meeting of the Linguistic Society of America, State University of New York at Buffalo, July 29-31, 1971.

Toward a Theory of Mechanical Problem Analysis

M. Valach, H. J. Eiden

The object of this research has been to establish processes for carrying out the analysis of diverse classes of problems in terms of an arbitrarily selected set of problem solving procedures from various disciplines. The nature and purpose of problem analysis, in this case, is toward the possible production of a solution program based on some subset of the selected procedures.

A "Problem Analysis Machine" (PAM) is postulated which is meant to produce solution programs for problems given by a hypothetical user (U). The actual execution of resultant programs is left as the task of U. It is assumed that U is capable of executing a finite set of procedures, part of which are primitive (irreducible to simpler procedures), and the rest formed by various compositions of the primitives. The procedure set is called π. PAM is then said to have π-knowledge, i.e., for each member of π PAM has stored a description of the initial situation for which the procedure is normally elicited, and a description of the resulting situation following procedure application. In addition to the pair of descriptions called IS and GS, there is sufficient description of the procedure itself to distinguish it from other procedures having the same IS and GS, and a particular name by which the procedure is referred to. A particular member of π -knowledge is called a transition rule and is symbolized by a quadruple (IS, GS, M, t_i), the elements of which refer to the type of descriptive information mentioned above, respectively.

For any member of the transition rules $\{t_i\}$, the descriptive information of each IS, GS, and M is symbolized by (E,R, A_E, A_R) and is called λ -description. Such a quadruple consists of a naming of elements; relations that hold between these elements; attribute values assigned to the elements; and attribute values assigned to the relations. The transition rules are clustered in π -knowledge according to discipline; each such cluster is called a π -domain. Although overlapping, π -domains are distinquished by their λ -vocabularies; i.e., by the element, relation, and attribute names used in transition rules.

A problem statement is given to PAM as a triplet (IS, GS, C)p, where each member of the triplet is expressed in λ -format. C is an expression of constraint set on M and/or any intermediate situations arising out of the serial application of t_is. It is the task of PAM to attempt to construct a solution program for a given problem statement in terms of $\{t_i\}$. Such a program may include branching, looping, or recursion, depending upon the richness of the set $\{t_i\}$. A solution program defines a method of transformation of an of an ISp to a GSp under the constraints C.

To accomplish its task, PAM executes a problem analysis program which operates on the problem statement and π -knowledge, with the aid of what is called a reinterpretation dictionary, a picture maker, and a collection of rules of inference $\{r_i\}$. In its simplest mode of operation, the analysis program proceeds much like one would carry out a derivation in formal logic; i.e., t_is are selected and applied to ISp or inversely to GSp in a manner like the application of theorems or rules of inference. In more advanced modes, PAM performs either description-improvement or domain-shifting. Both processes are performed on any of the members of (IS, GS, C)p, any of IS_{t_i} , GS_{t_i}, and M_{t_i}, or on any intermediate situation description occurring in the midst of application of a series of t_is. Description improvement may be obtained with the use of any of the three aids listed above. In this case the reinterpretation dictionary provides a listing of equivalent terms in any particular λ -vocabulary. The picture maker accepts an old description, converts it into a whole graphic image, accepts the names of new elements and relations, and attempts to find these new features in the image; hence modifying the old description. The rules of inference are clustered in direct correspondence with π -knowledge, and represent the theorems and inference rules by which one situation may be implied from another in a given discipline. These rules are expressed as triplets (IS, GS, r_i), whose members are expressed the same as those in the t_is of the particular domain. Domain-shifting is performed for the express purpose of changing the description of a situation, constraint, or M to that of some corresponding entity in an alternate disciplinary domain. The suggested alternate domain is indicated by a correspondence of vocabulary in the reinterpretation dictionary, or by a requirement reflected in the user's problem statement; i.e., ISp, and GSp are classified in different domains. A domain-shift may be carried out by one of two methods. On the one hand domain-shifting rules may exist in $\{t_i\}$ which take one across domain boundaries; on the other, where such rules do not exist, the analysis program performs what is called "coding." Coding is simply the process of mapping the names (elements, relations, attributes) of all or part of one description onto those of another, where such a mapping may or may not be one-to-one. The overall purpose of domain-shifting is twofold. It is invoked in one case when a solution is allowed to be multidiscipline; in the second case, it is used to discover solution programs in an alternate domain that exhibit by analogy the solution program structure required in the domain of interest.

The goals of this research do not include the actual design of the system outlined here; rather the intention is to establish and substantiate necessary and sufficient processes. Complete documentation of this work will appear in a forthcoming Ph.D. dissertation [3].

References

1. Amarel, S. "On the Representation of Problems and Goal-Directed Procedures for Computers." *Communications of the American Society for Cybernetics*, Vol. I, No. 2, 1969, pp. 9-36.

2. Banerji, R. B. *Theory of Problem Solving: An Approach to Artificial Intelligence*. New York, American Elsevier, 1969.

3. Eiden, H. J. *Toward a Theory of Mechanical Problem Analysis*. Atlanta, Ga., Georgia Institute of Technology. Ph.D. Thesis. (Forthcoming.)

STUDIES IN INFORMATION PROCESSING AND SYSTEMS DESIGN

Interactive Preparation of State-of-the-Art Reviews

P. J. Siegmann, D. E. Rogers, J. Reutter

In response to the rapidly expanding volume of new technical literature produced in recent years, means have been sought for making literature reviewing resources more effective. One such means would be the development of automated tools to assist reviewers during the analysis and composition phases of reviewing, so that computerized aids could serve to eliminate the more clerical tasks of reviewing and thus free the reviewer for the more creative aspects of his role. In order to investigate the feasibility of implementing an automated approach to the writing of review articles, we have in this research effort specified preliminary designs for a prototype system.

The prototype system has the following features: on-line access to a central file of technical literature for purposes of review; interactive text-editing facilities for constructing reviews and abstracts via CRT terminals; a facility for temporary filing of partially constructed reviews; protection of source documents and working reviews from unauthorized modification; and recording of actions taken by the reviewer during the reviewing process.

The suggested system consists of a CRT terminal (for document display, text entry and text editing), a central processing unit, a secondary storage file system, and a set of programs. The system has four modes of operation: retrieval mode, print mode, marking mode and editing mode. Retrieval mode provides the terminal user the ability to retrieve any document to which he has authorized access; once retrieved, depending on the class of document, a document may be marked or edited and then refiled as modified. Print mode is used to print one or more copies of any formatted document at any on-line printer terminal. Marking mode is used to mark key words or phrases and to enter commentary among the text of an existing document. Editing mode is used for entering new documents, formatting or editing existing documents, or rearranging text in existing documents.

There are two types of storage in the system: file storage and working storage. File storage is used to permanently store documents. Working storage exists only while a reviewer is using the system; a block of such storage is dedicated to each terminal user while he is connected to the system. The working storage block is divided into one partition containing structured lists and another containing the actual document text.

Software for the prototype system is comprised of nine major components (see also Fig. 1):

* Communication Control (for terminal interaction, page buffering)
* Text Parser (for recognition of commands, syntax checking)
* Command Handler (for subroutine sequencing, posting to the history file)
* Marker Mode Handler (for control of commentary entry and keyword marking of source document)
* Document Editor (for performing editing functions not provided by terminal)
* Document Formatter (for performing formatting functions specified by terminal operator)
* Print Control (for providing off-line printing functions, such as reformatting, suppression of embedded control numbers and format statements, etc.)
* File Interface (for filing of new documents, protection of documents in secondary storage, and retrieval of documents)
* Document Transformer (for construction of working storage list structures during read-in of a document to be edited).

Formatted documents are structured into sections, paragraphs and sentences. Sections are uniquely numbered within each document, paragraphs within each section, and sentences within each paragraph. Working storage list structures consist of a section list with forward linked sections, forward linked paragraphs, and forward linked sentences as shown in Fig. 2. Each sentence entry contains a pointer to the core location of the first character of text for that sentence.

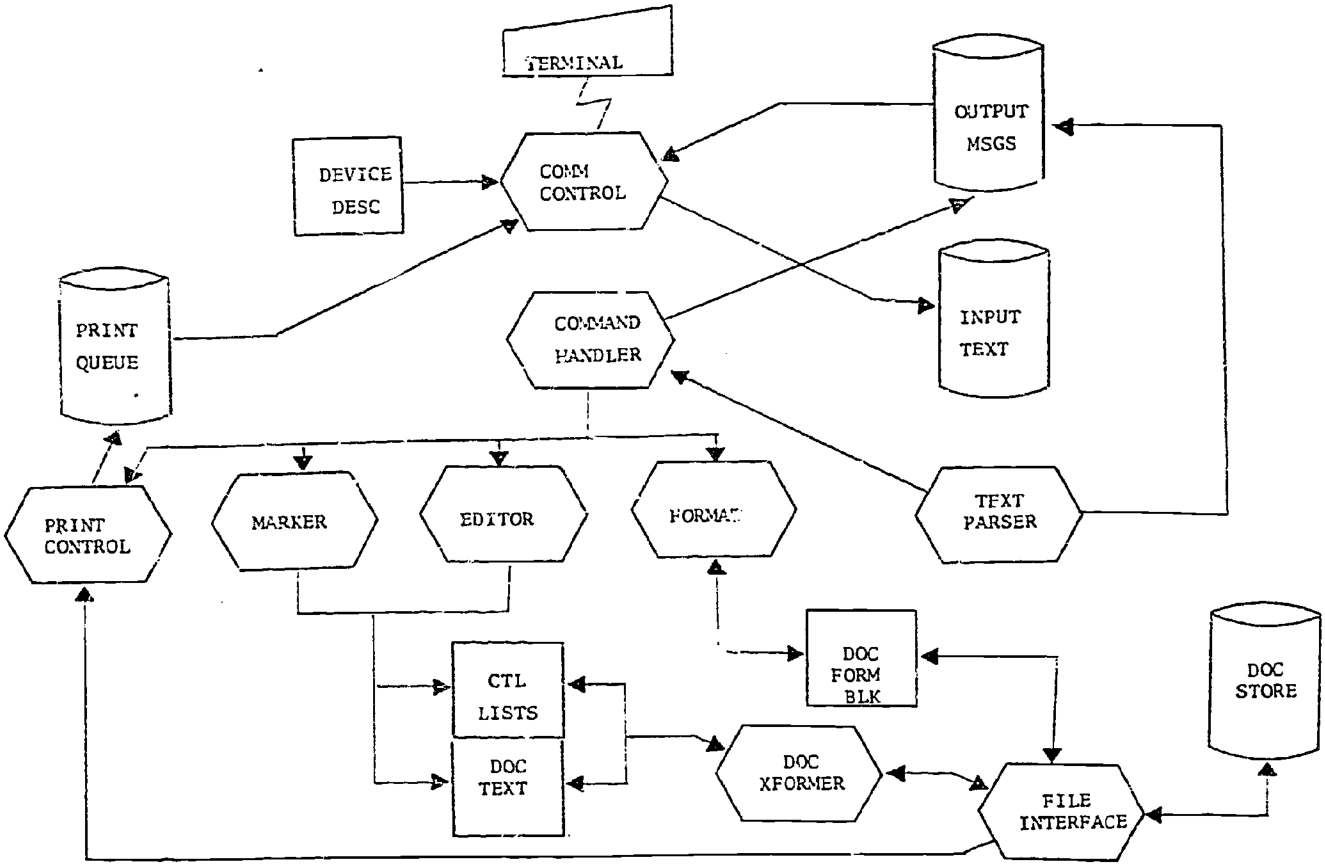


Fig. 1. Schematic of Prototype System Software

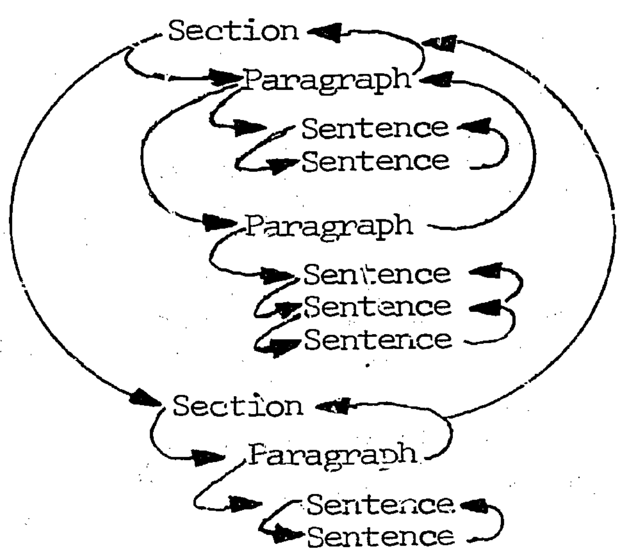


Fig. 2. List Structure for Working Storage

The user terminal proposed for the prototype system is a CRT device which handles most editing functions in a local mode. For laboratory purposes it would be desirable to have a programmable CRT memory, since this would allow experimentation with variations in the editing functions to be supported by the device. The display buffer is desired to be large enough to allow both rollup and rolldown, within reasonable limits. The screen should be capable of displaying a minimum of 1,800 characters of text in a single display.

Estimated hardware costs of the prototype system are $250/hour for CPU usage, and $5,000 for a CRT terminal. Estimated programming requirements for designing, coding and testing the nine software modules (containing an estimated 10,000 source statements if written in a high-level programming language with the facilities of PL/1) would be two man-years.

The study concluded that the prototype costs are prohibitive at this time.

References:

1. Reutter, J. "A Study of the Feasibility of Implementing a System for the Interactive Preparation of State-of-the-Art Reviews." Unpublished memorandum, Georgia Institute of Technology (School of Information and Computer Science) Atlanta, Ga., 1971.

Extending the Utility of Science Information to Education

V. Slamecka, A. P. Jensen

In attempts to extend the utility of science information data bases, the Georgia Tech Science Information Research Center pursues the potential of using such automated data bases in the process of self-instruction. The initial technical prerequisite has been the design and empirical evaluation of a prototype of a "knowledge utility" as a mechanism for the delivery of a nontrivial portion of the educational requirements. This prototype utility for self-instruction has been given the name "Audiographic Learning Facility (ALF)."

The concept of a self-instruction system, principally characterized by the absence of the live instructor as the primary and formal transmitter of knowledge, is shown schematically in Fig. 1.

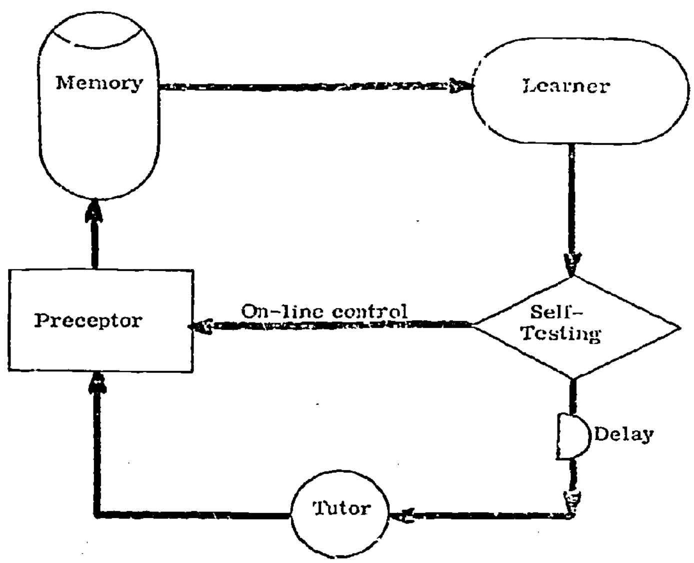


Fig. 1. The Self-Instruction System

The major components of this system are an inanimate, structured Memory for storing learning materials in a form suitable for transmission and for perception by remotely located learners, and a programmed Preceptor controlling the transmission.

The control over the process of self-instruction is partially vested in the programmed Preceptor, and in part it resides with the Learner. User-imposed control over the system is of two types. On-line control gives Learner the ability to start, stop and repeat a presentation, and to jump at any time to any other learning unit in the system. Using these commands, Learner can override the selection of learning units offered by Preceptor[1], and in such a manner participate, on-line, in the design of his learning strategy. The second control mechanism interposes between Learner and Preceptor the services of a human tutor; it is tantamount to an appointment or a conference with a teacher prior to overriding the programmed Preceptor. Incurred in this type of control will usually be a time delay.

The self-instruction system operates in two modes: scheduled and on-demand. Both modes of self-instruction can serve, optionally, either group audiences (e.g., a class) or individual learners.

The basic distinction of this pilot self-instruction system from other mechanized learning systems is in its storage of narrative-speech and line graphic "blackboard" lessons as the modular contents of Memory, and in its capability of actively involving Learners in the design of their learning strategies. The communication between Preceptor and Learner, and the transmission of audiographic learning materials, employ standard telephone lines. The implemented hardware system of the Audiographic Learning Facility has a capacity of approximately 120 hours of audiographic lectures, and it supports four remote, on-line learning sites. A limited version of the Preceptor software has been written.

Guidelines have been issued for the preparation of educational materials in audiographic storage form relative to (a) the identification of major concepts and learning goals, employing a directed graph approach; (b) the decomposition of major concepts into lessons and strategies, using precedence graphs to represent the latter; (c) lesson writing and recording; (d) preparation of introductory lessons explaining structures of learning concepts and strategies for particular learning goals; and (e) indexing learning materials for the purpose of updating the syndetic data-base aids.

Several introductory courses have been recorded by this method during the past year, including two courses on computer organization and programming, a course in discrete structures, and one in introductory cybernetics. Additional courses under recording at this time cover the subjects of environmental technology, world civilization, and a comparative study of programming languages.

---

[1] In its more advanced form, the Preceptor is itself a learning, self-organizing system striving to optimize its functions on the basis of certain categories of feedback/commands received from Learners. Among its other functions are monitoring Learner performance and collecting appropriate data useful for the management of the system.

During the past year the Audiographic Learning Facility has been tested empirically as the primary medium of delivery of learning materials in a six-week Summer Institute in Information/Computer Science for High School Teachers. During these six weeks, 20 high school teachers have used ALF in a self-instruction (group) mode for four consecutive hours per day, followed by four-hour periods of discussion and tutoring daily. Several methods of lecture recording have been tested: recording prior to presentation has averaged approximately 3 man-hours of faculty time per hour of recorded instruction; recording in live classroom consumes little more than the time of the live lecture.

This empirical, real-world use of ALF has been exceedingly successful. A remote application of ALF will take place in Fall 1972 between Georgia Tech and West Georgia College in Carrollton, Ga.

Among the key objectives of this project, as delineated in the Annual Progress Report for 1969/70, is the study of the extended use of science information banks for the semiautomatic creation and updating of "knowledge utilities" for education. The project will be approaching this study phase in the coming year, and will seek to devise methods of semiautomatic mapping of information from a science information bank into ALF.

References

1. School of Information and Computer Science. *Self-instruction Systems: An Alternate Socio-Technological Approach to National Education and Training.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1971. Research Report and Prospectus, 13 p.

2. Slamecka, V., Jensen, A. P., Valach, M., and Zunde, P. "A Computer-Aided Multisensory Instruction System for the Blind." Submitted for publication.

3. Slamecka, V., Jensen, A. P., and Zunde, P. "An Audiographic Repository of Knowledge for Conversational Learning." To appear in *Educational Technology.*

An Adaptive Spectrum Analysis Vocoder

J. C. Hammett

The objective of this research was to improve the performance of modern speech bandwidth compression systems; the properties of the vocoder (voice-coder) were examined, a potential improvement was proposed and incorporated into a vocoder design, and the performance of the resulting system was evaluated by computer simulation [1].

The phonemes of speech display a wide range of time-frequency properties, due to the extremes in the articulatory dynamics of speech production. The vocoder is based on the simplified model of speech production, an essentially "stationary" model. The relative validity of the model may be improved by matching the duration of the analysis window function to intervals of speech which are indeed "stationary." These observations motivated the design and experimental evaluation of a vocoder which adapts its time-frequency resolution properties to match the relative stationarity of different segments of input speech.

The homomorphic vocoder was selected as a test platform to evaluate the adaptive spectrum analysis strategy. The homomorphic vocoder was a natural choice for the simulation because its time and frequency properties may be readily manipulated - - time resolution by the duration of the analysis window function, and frequency resolution by the number of cepstrum coefficients transmitted.

An adaptive homomorphic vocoder was designed and a simulation system implemented on a large-scale digital computer. Experimental runs with the adaptive homomorphic vocoder were made with three test sentences, and the synthesized speech was judged in informal subjective listening tests. In one experiment (with a female talker) two adaptive modes were employed with window durations of 12.8 or 25.6 ms, frame intervals of 10 or 20 ms, and cepstrum truncation to 10 or 20 coefficients, respectively. The spectrum data rate was reduced to 3700 b/s and the synthesized speech judged to be of high "quality," retaining naturalness and recognition properties.

Two additional experiments (with male talkers) used window durations of 10 or 20 ms and a 3700 b/s data rate. The first of these resulted in synthesized speech judged to be of high "quality" but slightly less natural than the earlier result. The last experiment was conducted with a test sentence composed of voiced, non-nasal phonemes, which displayed no transitions in the spectrum rapid enough to warrant use of the 10 ms window mode, so the simulation operated as a conventional homomorphic vocoder with a 20 ms window. The result was judged to be reasonably good, but relatively not quite as good as the two previous results.

The tentative conclusion of the experimental phase of the investigation is that the adaptive strategy has potential for reducing vocoder data rates, while maintaining intelligibility, speaker recognition, and naturalness properties.

References

1. Hammett, J. C., Jr. *An Adaptive Spectrum Analysis Vocoder.* Atlanta Ga., Georgia Institute of Technology, 1971. 142 p. (Ph.D. Thesis.)

Computer Picture Processing

M. D. Kelly, D. R. Smith

The primary goal of this research has been the development and improvement of techniques for computer picture processing. (The term "picture processing" is used to denote the processing of pictures obtained from the outside world, and includes areas often called "pictorial pattern recognition" and "picture analysis and description.") Our most recent work has focused on edge detection and the application of Fourier optics to digital pictures.

Our work on edge detection has been an attempt to provide a unifying framework from which one can evaluate the widely varying edge detection algorithms which have been reported [3,4]. This has led to the development of criteria by which the effectiveness of these algorithms can be measured.

The measures developed for comparing edge detection operators were:

* Computation time
* Sensitivity to slope
* Sensitivity to width
* Ability to detect as distinct edges which are close together
* Sensitivity to noise
* Tendency to indicate unwanted edges in areas of gradually varying light intensity

Programs have been developed for evaluating edge detection operators according to the above criteria. Operators which have been tested include:

* Four point approximation to gradient
* Nine point approximation to gradient
* Five point approximation to Laplacian
* None point approximation to Laplacian
* The multiplicative operator of Rosenfeld [3].

The programs described above were written in standard algorithmic languages. In addition, the PAX language [2] for computer picture processing has been extensively tested to determine its merits for efficient development and evaluation of picture processing operators.

We have investigated applications of Fourier optics for computer picture processing. The spatial frequency domain provides a useful measurement space for the development and comparison of techniques for image enhancement and bandwidth compression.

In the area of image enhancement using the Fourier transform, programs have been implemented for edge detection and contrast improvement. These algorithms are high pass filters in the spatial frequency domain. The examination of the shape of the filters provides valuable insight into the behavior of the corresponding operator in the spatial domain.

Edge detection operators which have been tested in the Fourier domain include the gradient and Laplacian operators.

The methods of Fourier analysis have been used for bandwidth reduction of pictures for economical transmission and storage. Experimentation has ranged over quantization methods, sampling frequencies and patterns, and noise tolerances.

References

1. Abbamonte, M., et al. *Edge and Curve Enhancement in Digital Pictures*, 2. Report 70-103, University of Maryland Computer Science Center, 1970.

2. Butt, E. B., Snively, J. W. *The PAX II Pict..re Processing System*. Report 68-67, University of Maryland Computer Science Center, 1969.

3. Kelly, M. D. *Visual Identification of People by Computer*. Ph.D. Thesis, Stanford University, 1970.

4. Rosenfeld, A. *Picture Processing by Compu r*. New York: Academic Press, 1969.

Computer Structure for Description of Pictures

M. D. Kelly, M. L. Baird

The long-range objective of this research has been to develop a computer representation for pictorial concepts that can be used for machine recognition. The current work has been sharply limited in order to provide a base for evaluation of the potential of the method. The specific goals of the current work are: to develop a method for representing pictures in computers based on types of vertices and connectivity of vertices; and to describe pictorial concepts in terms of this representation method for purposes of recognition.

Objects considered are restricted to those with the following characteristics:

Two-dimensional

Monochromatic

Time-independent

No overlaying objects

Line drawings

Straight line segments only

Noise free

Input in symbolic format

The descriptive language for concepts is restricted to a language quite similar to the computer representation used.

The background of this research may be sketched as follows. In processing pictures with the aid of computers, the problem of recognition has been most extensively studied, and recognition has typically been posed as a problem of categorization. Most work has been based on the receptor/categorizer model (RCM) described by Marill and Green [2]. The task is to assign a token of one prototype to the correct prototype in a given finite set of picture prototypes. The methodology employed is that of attribute assignment and attribute value computation guided primarily by the desire to optimize the partitioning of the property space and to devise efficient inference techniques to make minimal error decisions.

The RCM fails to be useful for analyzing complex pictures where the structure and interrelationships among the picture components are important factors. This is illustrated by considering the one-dimensional pattern recognition task performed by a programming language translator. While one purpose of the syntax analysis phase of the compiler is to categorize an input program into one of two classes -- the class of syntactically correct programs or the class of syntactically incorrect programs -- the most important purpose is to obtain a description of the structure of the input program.

Many pictures are difficult or impossible to analyze within the RCM. Among these are mechanical drawings, flowcharts, complex biomedical pictures, circuits, and most three-dimensional objects represented in two dimensions. In these cases, a description of the picture in which meaningful relations among subparts of the picture are apparent is required. The RCM is then appropriately applied in the recognition of the basic components of the pictures.

Narasimhan is credited with first forcefully stating the case for a new approach to pattern recognition:

> Categorization, clearly, is only one aspect of the recognition problem; not the whole of it by any means. It is our contention that the aim of any pattern recognition procedure should not be merely to arrive at a 'Yes', 'No', 'Don't know' decision but to produce a structured description of the input picture. Perhaps a good part of this confusion about aims might have been avoided if, historically, the problem had been posed as not one of pattern <u>recognition</u> but of pattern <u>analysis</u> and <u>description</u>.[4]

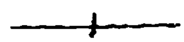
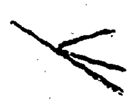
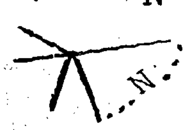
A new approach to the description, generation, and recognition of classes of pictures centers around linguistic methods. The basic idea is to extend the notions of syntax and semantics to n dimensions (n>1) and then apply an adaptation of the techniques of natural and artificial language processing. The linquistic approach has been broadly surveyed by Miller and Shaw [3]. The use of linguistic methods is basic to our work.

Thus, drawing from the ideas outlined above, a representation method for pictures has been developed. Pictures are represented by lists of vertices, attributes of vertices, and connectivity of vertices. This representation is similar in some ways to that used by Guzman [1]. Additional descriptive hierarchies have been introduced as well as the idea of partial orderings for vertex types and picture constituents.

The hierarchical representation includes picture elements, picture constituents, and primitive concepts.

Primitive picture elements are vertices. An element has a vertex type along with attributes specifying location and angles of incoming line segments.

VERTEX TYPES

Level 0	" "	null vertex	Type 0
Level 1			Type 1
Level 2	L90V		Type 2A (less than 90°)
	E90V		Type 2B (90° angle)
	G90V		Type 2C (> 90° & < 180°)
	180V		Type 2D (special case of type 1, 180° angle)
Level 3	FORK		Type 3A (3 lines forming angles < 180°)
	ARROW		Type 3B (3 lines forming one angle > 180°)
	T		Type 3C (3 lines, two colinear)
Level 4	K		Type 4A (2 colinear lines & 2 lines on same side)
	X		Type 4B (2 colinear lines & 2 lines on opposite sides)
	PEAK		Type 4C (4 lines forming single > 180°)
Level N	PEAK		Type NA (N lines forming angle > 180°)
	MULTI		Type NB (N lines, not included above)

Primitive picture constituents are closed areas, isolated lines, and reference directions. They are defined in terms of the primitive elements. Examples: isolated line; triangle; rectangle; square; N-sided polygon. To obtain constituents, several types of connectivity between vertices have been noted and formalized.

Primitive picture concepts include direction, orientation, and spatial relations between constituents.

We have introduced the notion of a pictorial concept and defined recognition as a hierarchial description of concepts, culminating in the most general class name of "object". The result is a hierarchy of descriptions of an object, each being more detailed than the former, until the most specific description is obtained. The most specific description is of course the raw data of the unprocessed picture.

The original statement (concept) is broken down into subconcepts, which in turn are reduced until primitive concepts are reached. Figure 1 illustrates the relationships between the description, generation, and recognition of complex objects in the system. Our current research encompasses that portion of the figure below the dotted line.

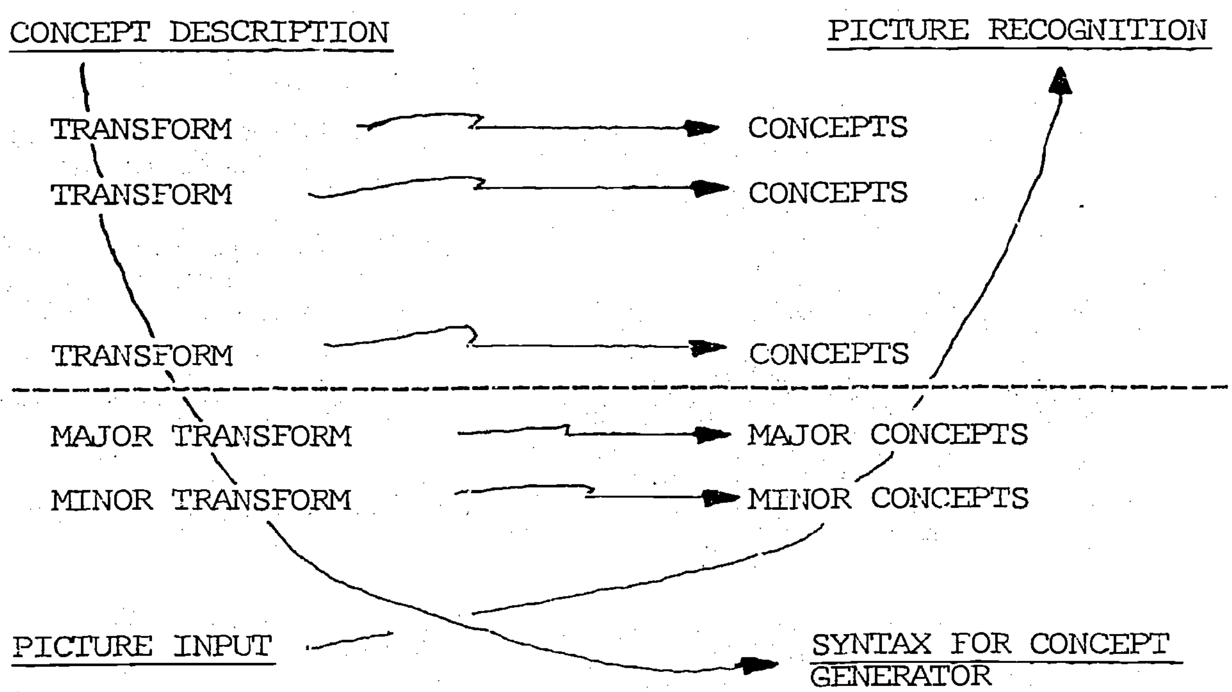


Fig. 1

References

1. Guzman, A. "Decomposition of Scenes Into 3-Dimensional Bodies." *Proc. AFIPS Fall Joint Comp. Conf.*, 1968, pp. 291-304.

2. Marill, T., and Green, D. M. "Statistical Recognition Functions and the Design of Pattern Recognizers." *ITE Trans. on Electronic Computers* EC-9. 4 (Dec 1960), pp. 472-477.

3. Miller, W. F., Shaw, A. "Linguistic Methods in Picture Processing: A Survey" *Proc. AFIPS Fall Joint Comp. Conf.*, 1968, pp. 279-290.

4. Narasimhan, R. *A Linguistic Approach to Pattern Recognition.* Report No. 21, Digital Computer Laboratory, University of Illinois, July, 1962.

Optimal Simultaneous Flow in Single Path Communication Networks

R. M. Siegmann

This research has resulted in the documentation of a method for finding an optimum solution to a communication network design problem in which only one path is selected for message transmission between each pair of stations in a directed network. The method finds a minimum cost network flow configuration which satisfies all the message flow constraints while allowing all messages to flow simultaneously, i.e., at the same time. The results are based on techniques drawn from the optimization literature and concepts taken from graph and network theory. The solution method can be applied in part or in whole to various types of scientific, technical, or business information networks.

In order to use the solution method, the exact configuration of nodes and arcs in a strongly connected communication network must be known. Also required are the unit cost and maximum number of messages allowed on each arc, the message flow requirements for each pair of network nodes, and the maximum length of any communication path. The solution method will find a least cost solution (if one exists) having a single path flow between each pair of stations and satisfying all the message flow constraints simultaneously. The optimum solution is expressed in terms of the sequence of arcs which define each path and the total cost of the network flow.

The solution method is based on three algorithms. The first is a versatile path finding technique called the path algorithm which finds the least cost restricted length path between all pairs of stations in the network. It produces the actual sequence of arcs along each of these paths and finds the maximum flow capacity for each path. The second and third algorithms integrate the path algorithm into a branch and bound technique to find a global solution to the network problem. All three algorithms are described, proved and illustrated [1].

Also produced by this research project were: (1) a computer simulation comparison of the relative speed of the path algorithm and one of the fastest known shortest path algorithms; (2) necessary conditions for a network to be solved by the solution method and bounds on the cost of a network solution; and (3) a complete example of the use of the solution method.

References

1. Siegmann, R. M. *Optimal Flow in Single Path Communication Networks.* Atlanta, Ga., Georgia Institute of Technology, 1971, 143 p. (Ph.D. Thesis).

Pre-scheduler and Management Model for Computer-User Systems

J. Gwynn, J. M. Hoffman

First, under limiting assumptions, a model is proposed for a single-processor, multiprogrammed computer serving a fixed user environment. Within the framework of this model, any workload can be assigned a cost. Second, an algorithm is devised to sequence piecemeal any workload such that the cost is reduced to a near-minimum value. Third, simulation is used to investigate the cost savings on numerous representative workloads for several operating system disciplines. Fourth, the effect of varying the various parameters of the model (subject to management decision) is investigated for some of the representative workloads.

Each user is assigned a constant loss rate prior to deadline and a possibly different constant loss rate after deadline; these loss rates reflect the user regret at not having the job completed. The computer is assigned a constant loss rate for failing to process at a rate faster than serial processing and a possibly different constant loss rate for processing faster than serially but slower than some nominally chosen multiprogrammed rate. The workload, when processed, incurs a loss which is a linear combination of the system regret and the sum of the individual user regrets. Management must assign the loss rates for the users, the deadline times, the system loss rates, the nominal multiprogramming rate, and the weight factors for the linear combination of user and system regrets.

Each user is assumed to submit only a known job to the system; the run time is assumed to be known at the time of submission. A job may be sub[illegible] one of two queues feeding the system, a preferenti[illegible] queue or an unknown arrival queue. Those users who [illegible] the preferential treatment queue are again determined [illegible]agement. Those jobs in the preferential treatment queue are known at the time of sequencing as to arrival time, processing time, and processing characteristics; they may be considered for sequencing although they may not arrive until later than the time of sequencing. Those jobs in the unknown arrival queue may be considered for sequencing only after they have arrived in the queue.

A procedure is devised to calculate the expected multiprogrammed run time for each job, once that job has entered the computer, from the running characteristics of that job and the other jobs in the mix. This is done for several different operating system disciplines, including roundrobin, first come/first served, and three queue reordering disciplines.

The multiprogrammed computer differs sufficiently from the industrial job shop that classical techniques of sequencing produce suboptimal results. However, a sequencing technique, resembling somewhat techniques which have been used in industrial job shops, is applied to the workload in an attempt to minimize the system-user regret. The workload, generally too large for sequencing all at once, is sequenced piecemeal. The results, while possibly suboptimal, still produce large percentage improvements over unsequenced workloads in the cases simulated.

The effect of management decisions is studied by varying several of the parameters reflecting management. Changes in user loss rates, user deadlines, system loss rates, nominal multiprogramming rate, system-user weight factors, and the assignment of jobs to the preferential and unknown queues caused, in simulation runs, changes in the workload sequence reflecting the corresponding parameter changes. The utility of the management model is thus demonstrated [1,2].

The contributions of this research can be outlined as: a workable management model of a computer-user environment; the consideration of the computer-user system as a job shop; scheduling when all jobs to be scheduled are not yet on hand; prescheduling a computer workload to reduce cost, within the framework of the model.

References

1. Hoffmann, J. M. *The Pragmatics of Computer Scheduling.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report, GITIS-70-10. 15 p. (PB 195 135.)

2. Hoffmann, J. M. *A Pre-scheduler and Management Model for a Class of Computer-User Systems.* Atlanta, Ga., Georgia Institute of Technology, 1971. 165 p. (Ph.D. Thesis.)

Multiprogramming Scheduling

J. Gwynn, E. M. Pass

Based on some measure of system effectiveness, an attempt is made to schedule for processor service those jobs submitted to the operating system of a multiprogrammed computer. Producing optimally or near-optimally, based on the chosen system effectiveness measure, is the goal.

Using throughput as the measure of effectiveness, three problems are considered. The first two are related and are solved theoretically; the third is solved, to some degree, by simulation.

First, consider a modified roundrobin operating system discipline in which the jobs are partitioned into categories. Within each category the jobs are assigned processor service in the usual roundrobin fashion. The categories are ordered as to priority, and roundrobin service begins in the highest priority category, with jobs in lower priority categories receiving no processor service until the jobs in all categories above are unable to be processed. Assuming the serial run times for all jobs are known, and the increase in run times because of jobs in higher priority categories receiving processor service is also known, the problem is to apportion the jobs on hand into the various categories in order to maximize throughput. This leads to a nonlinear system which can be solved using a response surface technique.

Second, consider the similar problem where, instead of a roundrobin assignment of processor service within each category, the exact sequence of processor service is any predetermined cycle sequence. With the same assumptions about the jobs submitted to the system, maximization of throughput leads to a linear system which can be solved by linear programming.

Third, consider a somewhat different problem collection of jobs with known run times, memory requirement, and arrival times is to be submitted. Assume further that the memory required is contiguous, non-overlayable, and is an integer multiple of some fixed page size. Finally assume that the run time of each job is independent of the jobs with which it is multiprogrammed, as is the case for severely I/O bound jobs.

To maximize throughput may be impossible by other than combinatorial means; however, a branch and bound technique combined with heuristics which avoid major memory checkerboarding leads to a near-optimal solution.

These results should provide a basis for the solution of less trivial, real-world problems leading to a doctoral dissertation.

The results of this research so far are presented in an unpublished internal memorandum [1].

References

1. Pass, E. M. "Two Approaches to Multiprogramming Scheduling." Unpublished internal memorandum, Georgia Institute of Technology (School of Information and Computer Science), 1971.

Problems in Operating Systems Design

C. R. Pearson

A study of the general design problem for operating systems has been made using the point of view of Cohen's transaction chart method [1]. Several unifying principles have been enunciated and a series of practical problems have been studied using these principles. The results of our studies are to be published in a forthcoming report [2].

The problems considered include the design of instrumentation packages for DOS/360 and for GEPAC 4000; the analysis of general design goals for operating systems; an analysis of the uses of SL/8 in implementing operating systems; and the preliminary design of an operating system for the ILLIAC IV.

The DOS/360 instrumentation package has been implemented and allows for the measurement of all parameters of interest for tuning or restructuring DOS, or for restructuring the installation (hardware & software). The package makes use of facilities already available in DOS/360: CE Area; Job Accounting; Supervisor hooks for customer supplied system functions; and transient functions.

The GEPAC 4000 instrumentation package takes a different approach to implementation by using a very few modifications to the vendor-supplied operating system to achieve the same result of being able to measure all parameters that are of interest both to the installation and to the systems designer.

A list of general design goals and objectives for operating systems was developed and a literature search made for other information on this subject. It was found that a significant treatment of operating system design goals does not exist in the published literature.

The design goals were classified and an analysis of the conflicts between the various goals was made.

The SL/8 language was proposed for designing and implementing operating systems. This analysis discovered that it could not serve that purpose for two reasons: first, it is hardware-limited to the PDP-8; and secondly, even for the PDP-8 it does not have all the capabilities required for implementing operating systems.

The transaction chart method was applied to the preliminary design of an operating system for the ILLIAC IV. This has been submitted to the University of Illinois for critique.

Participants in these studies have included: D. Abernathy, W. Ballard, J. Burke, L. Chappell, H. Cotton, J. Dyer, D. Hall, J. Ivey, J. Mancino, J. Norem, R. Thomas, R. White, J. Wilson.

References

1. Cohen, Leo J. *Operating Systems Analysis and Design*. London, MacMill and Co., 1970

2. Pearson, C. R. (ed.). *Problems in Operating Systems Design*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science). Research Report, GITIS-71-05. (Forthcoming.)

STUDIES IN AUTOMATA AND SYSTEM THEORIES

The Algebraic Theory of Abstract Computers

L. Chiaraviglio, J. H. Poore

An abstract computer is composed of a set of states S, a set of actions A, and a control unit C. The triplets (S,A,C) meet the following requirements: a) S and A are nonempty disjoint sets; b) A is a subset of the set of all functions from S into S; and c) C is a function from S onto A. The computer (S,A,C) operates by having C "read" a state s in S, obtaining the action C(s) in A, applying this action to the state s and thus obtaining the "next" state C(s)s in S. The state transition function T for the computer (S,A,C) is given by T(s) = C(s)s for every s in S.

The computer is said to be iterative, ordinarily recursive, or an omega computer if T^{α} is defined for every finite ordinal, $T^{\alpha+1}(s) = TT^{\alpha}(s)$ for every $s \in S$. The computer is said to be synchronous if time runs as the powers of T. That is to say that if at time t_o the computer is in state s_o, then at time t_{α} the computer is in the state $s_{\alpha} = T^{\alpha}(s_o)$.

The set P of processes of an iterative synchronous computer is the set of all sequences ρ in S^{ω} such that for every α in ω, $\rho(\alpha) = T^{\alpha}(\rho(o))$. The process ρ is terminating if there exists an α such that $\rho(\alpha) = \rho(\alpha+1)$. The computer is said to stop at the state $\rho(\alpha)$ when started at the state $\rho(o)$ if $\rho(\alpha) = \rho(\alpha+1)$. We may note that since T is a function, computers are capable of generating only two types of processes, namely: processes that are nonrepeating and processes that are periodic after some finite delay. Terminating processes are periodic processes of period one. Iterative or omega computers, if they stop, do so after a finite delay. Obvious generalization to limit ordinals beyond omega yields computers that may stop after nonfinite delays.

Computer processes are determined uniquely by the choice of first stage and by their control unit. The choice of starting state is the "external" control of the process. The "internal" control is given by the control function. Programming is the choice of initial state. Thus, computers that require choice of initial states are said to be programmable. Computers whose "internal" control is given by one control function are said to be centrally controlled. Computers whose states are disjoint from the action are said to be noninteractive.

If the set of states S of a computer (S,A,C) is a set of functions in Y^X, X and Y nonempty, then the computer is said to be a finitary action computer if and only if for every s in S, T(s) and s are functions in Y^X that differ only at a finite number of arguments. Abstract digital computers are a special case of iterative, synchronous, centrally controlled, noninteractive, programmable and finitary action computers.

Let B be any Boolean algebra, B^* the dual space of B, and $F(B^*)$ the field of sets over B^* that is isomorphic to B. Let H be the isomorphism in question. We may construct an algebra $\widehat{B}$ that is a subalgebra of the algebra of all functions from B^* into the simple Boolean algebra 0 and $\widehat{B}$ is also isomorphic to B. For every b in B, then the corresponding element $\widehat{b}$ in $\widehat{B}$ is determined by the condition that $\widehat{b}(x) = 1$ if and only if $x \varepsilon H(b) \subseteq B^*$. The correspondence $\widehat{}$ is an isomorphism. Thus we may without loss equate B and $\widehat{B}$. With this proviso, the definition of an abstract digital computer is as follows:

(B,A,C) is an abstract digital computer if and only if it is a synchronous, centrally controlled noninteractive, programmable, finatary action computer and B is a Boolean algebra. The finitary action of abstract digital computers is given by the requirement that for any a in A, b in B, there exists a finite subset of B^* such that the symmetric difference of b and ab is a function whose value is zero for every x in (B^*-K). The requirement makes sense in view of the identification of B and $\widehat{B}$.

There are essentially three ways of looking at computers. A computer may be viewed as a triplet of states, actions, and control function. It may be viewed as a pair composed of a set of states and a transition function. And it may be viewed as a set of processes, an appropriate subset of the set of all functions from some ordinal into the set of states. In all its characterization a computer is an algebra and for such algebras we may elaborate rather obvious representation theories.

Every computer is homomorphic to an abstract digital computer. If (S,T) is a computer, let H be a one-one mapping of S into 0^S that associates with each s in S the atom H(s) of 0^S such that $H(s)s = 1$ and let T´ be given by $T'H(s) = HT(s)$ for all atoms in 0^S and $T'(b) = b$ for b not an atom. $(0^S, T')$ is an abstract digital computer and H is a monomorphism. It may be noted that the action of $(0^S, T')$ are one-place set and reset functions.

Every abstract digital computer is isomorphic to a finite set-reset computer. A family of finite reset functions R is a function from the finite subsets of the dual space B^* of an algebra B into B^B such that for finite $K \subseteq B^*$ and b in B, $R(K)bx = 0$ if x is in K and $R(K)bx = bx$ otherwise. Similarly the finite set S is a function from finite $K \subseteq B^*$ into B^B such that for b in B, $S(K)bx = 1$ if x is in K and $S(K)bx = bx$ otherwise. A finite set-reset computer is an abstract digital computer whose actions are all of the form $S(K)R(J)$ for finite subsets K and J of B^*. If (B,T) is any abstract digital computer, then $Tb = S((b' \wedge T(b))^{-1}(1))R((b \wedge (Tb)')^{-1}(1))b$ since $(b' \wedge T(b))^{-1}(1)$ is the finite subset of B^* on which Tb differs from b and is valued 1 and $(b \wedge (Tb)')^{-1}(1)$ is the finite subset of B^* on which Tb differs from b and is valued 0. Thus the control function of the set-reset computer defines the same transition function and is indeed identical to (B,T). The two computers differ in their actions but not in their transition functions.

In the case of abstract digital computers a range of quantitative considerations may be handled through the theory of Boolean duality. If (A,T_A) and (B,T_B) are two abstract digital computers such that there exists an H in B^A for which $HT_A = T_BH$ and H is a Boolean homomorphism, then the dual of H, H^*, is a continuous mapping from the dual of B, B^*, into the dual of A, A^*. The study of such continuous functions on the dual spaces of the set of states of abstract digital computers allows us to countenance relations among computers that satisfy metric considerations.

References

1. Poore, J. H., Jr. *Towards an Algebra of Computation.* Atlanta, Ga., Georgia Institute of Technology, 1970. 75 p. (Ph.D. Thesis.)

2. Poore, J. H., Jr., and Chiaraviglio, L. "The Algebras of Abstract Digital Computers." (Forthcoming, 1971.)

Abstract Computers and Degrees of Unsolvability

L. Chiaraviglio, J. R. Horgan

Abstract digital computers have been shown to be more powerful than Turing machines. An abstract digital computer that is a halting decider for Turing machines has been constructed. Abstract digital computers are constructed which recapture the computations that ensue from any system of equations relative to which some functions are recursive in some given functions. Thus for each degree of unsolvability there is an abstract digital computer which "computes" the function of that degree.

References

1. Horgan, J. R., Roehrkasse, R. C., Chiaraviglio, L. "Abstract Digital Computers and Turing Machines." (Forthcoming, 1971.)

2. Horgan, J. R. *Abstract Computers and Degrees of Unsolvability.* (Doctoral dissertation, forthcoming, 1971.)

Abstract Computers and Automata

L. Chiaraviglio, R. C. Roehrkasse

For each class of automata an abstract digital computer is constructed that recaptures every computation of every automaton in the class including changes of state, changes in reading head position, and changes in tape. The abstract digitial computers recapture every detail of the computational procedure of each class of automata.

An abstract digital computer has been constructed for each of the classes of finite state automata, pushdown store automata, linear bounded automata, Turing machines and where appropriate their nondeterministic counterparts. Furthermore, the abstract digital computers that capture the computations of a given class of recognizers is monomorphic to the abstract computers that recapture the computations of more powerful recognizers. The relation between classes of recognizers and their grammars is obtained in the theory of abstract digital computers as morphisms between the relevant computers.

References

1. Horgan, J. R., Roehrkasse, R. C., Chiaraviglio, L. "Abstract Digital Computers and Turing Machines." (Forthcoming, 1971.)

2. Roehrkasse, R. C. *Abstract Digital Computers and Automata.* (Doctoral dissertation; forthcoming, 1971.)

Combinatonic Computers and Their Algebras

L. Chiaraviglio, G. Baralt-Torrijos

Combinatonic computers are interactive, noncentrally controlled computers. Intuitively speaking, a combinatonic computer is a set of processors that act upon other processors to produce a transition when the computer is unstable. Unstable configurations of processors may yield stable configurations hence terminating processes. Stable configurations are the terminations of processes. Of course, stable initial configurations do not yield processes and the computer is said to be inactive.

Each component processor in a configuration is called a combinaton. Each combinaton has a fixed mode of action and a fixed stability threshold. The threshold for each combinaton is given by at least two parameters: the minimal number of combinatons it needs to combine with in order to process, and the kinds of combinatons with which it is unstable. Thus if the set of combinatons in a configuration are said to be the environment of each component combination, then the processes that ensue from the configuration are locally controlled by the number and kinds of combinatons in the environment. The computer is locally controlled.

The computer is interactive since each configuration of combinatons is capable of interacting with other configurations to produce stable or unstable configurations. The computer is also noniterative since every process that yields a stable configuration occurs in one timespan jointly with all of its subprocesses.

Combinatons are elements of a monad M on which there is defined a left and right monotonic congruence $\equiv$. A combinatonic computer is a pair $(M, \leq)$ where $\leq$ is a reduction relation defined relative to the congruence $\equiv$. Irreducible elements, elements c in M for which there are no b in M such that $b \leq c$ and $b \neq c$, are said to be the stable configurations of the computer $(M, \leq)$. Not all computers need have irreducible elements nor need all the states of some computer be reducible.

The relation between abstract noniteractive, iterative centrally controlled computers and combinatonic computers is complex. For every combinatonic computer there exists an abstract computer that recaptures all of its computations but not conversely, But the notion of procedure in each case is not uniformly recaptured by these embeddings. This fact seems to indicate that combinatonic and abstract computers are genuine procedural alternatives in computation.

References

1. Baralt-Torrijos, G. *Combinatonic Computers.* (Doctoral dissertation; forthcoming, 1971.)

The Algebras of Programming Languages

L. Chiaraviglio, R. A. DeMillo

A program in some language L is a finite, ordered, and well-formed set of lines. An algebraic structure for L is obtained by considering a machine-free interpretation for all of its programs together with a congruence relation on such an interpretation. A machine-free interpretation of a programming language is obtained by considering a structure that is formed by the formal objects of the language.

Consider a programming language that has among its formal objects a set of labels L, a set of predicates P, a set of functors F, and well-formed lines of the form:

$$\ell_1\ P_1,\ \text{then}\ f_1\ \ell_2\ \text{else}\ f_2\ \ell_3\ ,$$

where ℓ_1, ℓ_2, ℓ_3 are labels, P_1 is a predicate, and f_1, f_2 are functors. To each such line there corresponds an element of L x P x F x L x F x L. We may identify in P x F x L x F x L a dummy quasi line e. Entering the line '$\ell_1 e$' in any program has no effect provided the resulting ordered set of lines is still a well-formed program.

Any function from L into P x F x L x F x L that has a value different from e for at most a finite subset of L is a program. A universal trace function for all programs in L relative to some nonempty set X is a function T_X that maps $(P \times F \times L \times F \times L)^L \times L \times P \times L \times F \times L$ into itself. This set together with T_X is an abstract, iterative, centrally controlled computer. The theory of computer morphisms may now be employed to obtain congruences among programs that are machine-free.

References

1. DeMillo, R. A., Chiaraviglio, L. *On the Applicative Nature of Assignment*. Atlanta, Georgia, Georgia Institute of Technology (School of Information and Computer Science), 1971. Research Report, GITIS-71-01. 25 p. (PB 195 875.)

On Chomsky's Context-Sensitive Language Which Is Not Context-Free

J. M. Gwynn, R. A. DeMillo, B. M. Saulnier

Chomsky [1] asserted that the context-sensitive language $L = \{a^n\, b^m\, a^n\, b^m \mid$ each of m, n is a positive integer$\}$ is not context-free. For this assertion he supplied a proof which, though appealingly simple, is in fact incorrect. However, to the authors' knowledge, nowhere in the existing literature has this error been pointed out.

That L is not context-free can possibly be derived from the body of theory which has built up since Chomsky's original assertion [2,3]; however, the literature lacks a short and simple proof. This note identifies the error in Chomsky's original argument and provides an equally succinct, but also correct, proof that L is not context-free.

Definitions. G is a context-free grammar if there exist a finite alphabet Σ, a finite terminal alphabet V_T, a finite nonterminal alphabet V_N, a start symbol $S \in V_N$ and a finite set $\mathcal{P}$ of production rules such that $\Sigma = V_T \cup V_N$, $V_N \cap V_T$ is empty, and each production rule in $\mathcal{P}$ is of the form $B \to \alpha$, where $B \in V_N$ and $\alpha \in \Sigma^*$. The string α is directly derivable from the string β, denoted $\beta \Rightarrow \alpha$, if there exist $B \in V_N$ and strings γ_1, γ_2, $\delta \in \Sigma^*$ such that $\gamma_1 B \gamma_2 = \beta$, $\gamma_1 \delta \gamma_2 = \alpha$, and the rule $B \to \delta$ is in $\mathcal{P}$. The string α is derivable from the string β, denoted $\beta \overset{*}{\Rightarrow} \alpha$, if there exist a positive integer n and strings $\{\gamma_i\}_{i=1}^{n}$ such that $\beta = \gamma_0$, $\alpha = \alpha_n$, and $\gamma_{i-1} \Rightarrow \gamma_i$ for $1 \le i \le n$. The relation $\overset{*}{\Rightarrow}$ is clearly transitive. The set of strings $\alpha \in V_T^*$ such that $S \overset{*}{\Rightarrow} \alpha$ is the language of G, denoted L(G). A set L of strings is a context-free language is there exists a context-free grammar G such that L = L(G).

The Authors' Proof. Assume L is a context-free language. Since L is infinite there is a context-free grammar G such that L = L(G) and for each $A \in V_N$ there exist infinitely many strings $\alpha \in V_T^*$ such that $A \overset{*}{\Rightarrow} \alpha$ (any other nonterminals having already been eliminated by direct substitution in the production rules). Denote V_N by $\{A_i\}_{i=1}^{k}$, and, for $1 \le i \le k$, let t_i be the largest integer q such that, for some string $\alpha \in V_T^*$, $A_i \Rightarrow \alpha$ and α is of length q. Let $r = \max_{1 \le i \le k} t_i$, and consider the string $a^r\, b^r\, a^r\, b^r \in L$. Since $S \overset{*}{\Rightarrow} a^r\, b^r\, a^r\, b^r$ and since each derivation has a final direct derivation, there exist strings $\alpha, \theta, \psi \in V_T^*$ and $A_j \in V_N$ such that

$A_j \Rightarrow \alpha$ and $S \overset{*}{\Rightarrow} \theta\, A_j\, \psi \Rightarrow \theta\, \alpha\, \psi = a^r\, b^r\, a^r\, b^r$. $\theta\, \alpha\, \psi$ must be in exactly one of the following forms, where α is enclosed in parentheses.

1. $a^m\ (a^s)\ a^{r-m-s}\ b^r\ a^r\ b^r$
2. $a^m\ (a^{r-m}\ b^s)\ b^{r-s}\ a^r\ b^r$
3. $a^r\ b^m\ (b^s)\ b^{r-m-s}\ a^r\ b^r$
4. $a^r\ b^m\ (b^{r-m}\ a^s)\ a^{r-s}\ b^r$
5. $a^r\ b^r\ a^m\ (a^s)\ a^{r-m-s}\ b^r$
6. $a^r\ b^r\ a^m\ (a^{r-m}\ b^s)\ b^{r-s}$
7. $a^r\ b^r\ a^r\ b^m\ (b^s)\ b^{r-m-s}$

For (1), (3), (5), and (7), $0 \leq m$, $0 \leq s \leq t_j \leq r$, and $m + s \leq r$; for (2), (4), and (6), $0 < s < r$, $0 < m < r$, and $r-m+s \leq t_j \leq r$. If $\beta \in V_T^*$ and $A_j \overset{*}{\Rightarrow} \beta$ then $S \overset{*}{\Rightarrow} \theta\ A_j\ \Psi \overset{*}{\Rightarrow} \theta\ \beta\ \Psi \in L$; moreover, there are infinitely many such strings β. However, in each of the seven possible forms of the strings α, θ, Ψ, it is clear that if $\beta \neq \alpha$ then $\theta\ \beta\ \Psi \notin L$. This contradiction completes the proof.

<u>Chomsky's Error</u>. Chomsky's argument is paraphrased as follows. Let G be a context-free grammar such that $L = L(G)$ and for each $A \in V_N$ there exist infinitely many strings $\alpha \in V_T^*$ such that $A \overset{*}{\Rightarrow} \alpha$. L infinite and V_N finite implies there exists $A \in V_N$ such that substitution for A is the final step in the derivation from S of infinitely many strings in L; let M be this infinite set of strings. Let r be the largest integer q such that, for some string $\alpha \in V_T^*$, $A \Rightarrow \alpha$ and α is of length q. M infinite implies the existence of positive integers s,t such that $s + t > r$ and $a^s\ b^t\ a^s\ b^t \in M$. Hence there exist strings $\theta, \Psi \in V_T^*$ such that $S \overset{*}{\Rightarrow} \theta A \Psi \Rightarrow a^s\ b^t\ a^s\ b^t$; moreover, there exist infinitely many strings $\alpha \in V_T^*$ such that $A \overset{*}{\Rightarrow} \alpha$ and $S \overset{*}{\Rightarrow} \theta\ A\ \Psi \overset{*}{\Rightarrow} \theta\ \alpha\ \Psi$. A contradiction supposedly arises because, regardless of what strings θ and Ψ are, infinitely many of the strings $\theta\ \alpha\ \Psi$ are not in L. The following example shows this reasoning to be fallacious.

Let the production rules of G contain $S \rightarrow a^2\ Aa^2b^3$, $A \rightarrow aAa$, $A \rightarrow ab^3a$, and no other rules in which $A \in V_N$ appears. Here $r = 5$, and the rule $A \rightarrow ab^3a$ is used for the final substitution in the derivation of $a^nb^3a^nb^3$ for each positive integer $n \geq 3$. Let $s = 3$, $t = 3$, $\theta = a^2$, and $\Psi = a^2b^3$; then $s \Rightarrow a^2\ Aa^2\ b^3 = \theta A \Psi \Rightarrow a^2(ab^3a)a^2b^3 = a^3\ b^3\ a^3\ b^3$.
For each positive integer n, $A \overset{*}{\Rightarrow} a^n\ b^3\ a^n$ and

$$s \Rightarrow \theta A \Psi = a^2\ Aa^2\ b^3 \overset{*}{\Rightarrow} a^2\ (a^n\ b^3\ a^n)\ a^2\ b^3 = a^{n+2}\ b^3\ a^{n+2}\ b^3 \in L.$$

Since no other strings are derivable from A, no contradiction arises.

References

1. Chomsky, N., "On Certain Foraml Properties of Grammars." *Information and Control* 2:137-167 (1959).

2. Chomsky, N. "Context-Free Grammars and Pushdown Storage," *Quart. Prog. Rept. No. 65*, MIT Res. Lab. Elect., pp. 187-194 (1962).

3. Parikh, R. J. "On Context-Free Languages." *JACM* 13:570-581 (1966).

Research on Interaction Within Systems

P. Zunde

Interaction is a problem of theoretical and practical interest in miltivariable systems in general and multivariable control systems in particular. Although the theory of multivariable control has made some advances in recent years, it is still lagging behind the development of the theory of univariable control - - i.e., of control processes in which there is only one input to and one output from the system. In the case of a multivariable system (that is, a system with several inputs and outputs), the classical design procedure is to select input-output pairs and to design conventional single-variable systems neglecting the interaction or couplings which might exist among them. However, it was realized during the last two decades that consideration of system interactions leads to the design of higher performance control systems; and other words, to the design of systems with shorter response times and higher flexibility. Major attention was therefore directed towards the design of so-called noninteracting controls, in which input-output pairs do not in any way affect other input-output pairs, so that each output depends only on one pre-selected input and is independent of all other inputs.

Mesarovic [1,2], Narendra and McBride [3], and others have shown that noninteracting controls are only one aspect of interaction problems, which is termed input or cross-transfer interaction, and is roughly defined as the extent to which all inputs affect all outputs. Thus, complete input noninteraction is the conversion of an n-input, n-output multivariable system into n single-variable inputs, in which each input affects one and only one output. Other types of interaction are "output interaction" (which is basically the reaction of other outputs to an external disturbance applied to some output variable) and "output dependence" (which refers to the possibility of obtaining any independent set of desired output functions by suitable manipulation of the input functions). However, the study of these more complicated aspects of interaction in systems is not too far advanced.

The purpose of our own research [4] has been to describe a technique of analysis and synthesis of interacting linear dynamical systems which is based on the concepts of selective controllability and selective invariance and which offers considerable freedom in implementing specified interaction or noninteraction requirements in systems design.

We therefore began our research by defining the concepts of, and deriving the necessary and sufficient conditions for, selective controllability and selective invariance of linear dynamical time-invariant systems. A set of variables (input, output, or state) of a dynamical system could then be defined as "noninteracting" if and only if each of the variables in the set is selectively invariant with respect to all other variables in the set. Otherwise it is interacting.

Based on a systematic application of the criteria of selective controllability and invariance, a methodology has been developed for (1) the analysis of interacting dynamical systems and (2) the design of dynamical systems with specified interaction or noninteraction characteristics. We have also been able to show the great flexibility and power of this analysis and synthesis method, and to demonstrate that it can also be extended to linear time-varying and non-linear systems.

References

1. Mesarovic, M. D. *The Control of Multivariable Systems.* New York, Wiley, 1960.

2. Mesarovic, M. D. *Measure of Interaction in a System and Its Application to Control.* Systems Research Report 4-A-62-A, Case Institute of Technology, Cleveland, Ohio, 1962.

3. Narendra, K. S. and McBride, L. E. *Comparison of Different Modes of Representation of Multivariable Systems.* Cruft Lab., Harvard University, Cambridge, Mass., Technical Report No. 356, 1962.

4. Zunde, P. "Synthesis of Interacting Dynamical Systems." In: *Proceedings of the IEEE International Conference on Systems, Networks and Computers,* Oaxtepec, Mexico, January 1971.

A Theory of Dynamic Group Behavior

J. Talavage

Due to the difficulties presented by experimental investigations of social behavior, it has seemed useful to establish models of such behavior, either via mathematics or on a computer. Computer simulations of social behavior have been developed by Gullahorn [2] who, however, states that verification of such a model is not feasible; this difficulty with verification is apparently due to the very large number of computer runs necessary to reliably generalize the results. On the other hand, mathematical models inherently deal in such generalities and so would apparently present less difficulty in this regard with respect to verification. Once verified, a mathematical theory of social behavior provides a sound and rigorous foundation on which to build computer simulations of more detailed situations. In addition, some of the results derived from such a theory may provide insight and point to new directions for experimentation.

The study of social behavior on a mathematical basis requires that rigor is pursued at several different conceptual levels. At the broadest level, the effects of interaction among individual persons with different psychological characteristics is considered (this is the level at which sociological results are obtained); however, prior to study at that "interaction level," one must first formalize (i.e., make rigorous) what is meant by "psychological characteristics." Such a formalization may be said to constitute the "psychological level" of study. If we wished, we could proceed on to a next level, perhaps called the "physiological level," and so forth. However, this study begins at the psychological level with an eye toward application of that formalism at the interaction level.

Our choice of the formalism at the psychological level, hereinafter referred to as the "person-model" or simply P, was influenced by the efforts of Homans [3,4], Miller, Pribram and Galanter [6], Kelly [5] and Fararo [1].

The notions of reward, expectation, etc., are directly pertinent to Homans-type social behavior, and so they are taken as a minimal set of notions to be explicitly represented in the person-model. The structure of the person-model P is such that each of the pertinent notions is related to a "subsystem" and these subsystems are interconnected in a logical manner.

The block diagram given as Figure 1 shows the basic structure of P.

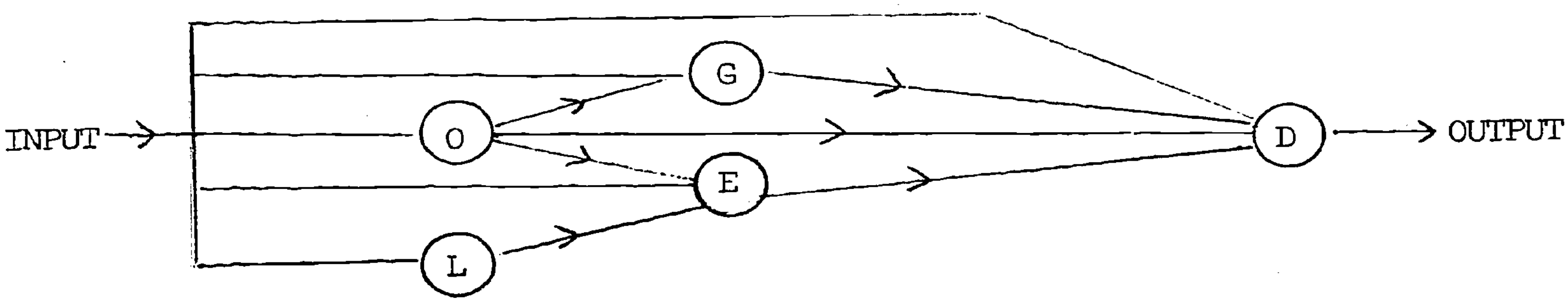


Fig. 1
P- Model Block Diagram

A person-model P is an ordered septuple (I, R, O, G, L, E, D) where:

(i) I and R represent sets of input events and output responses, respectively.

(ii) The system O is a static system characterized by the transfer function

$$o: I \rightarrow \Pi_R$$

Here, the function o represents the internal activity of generating candidate responses for a given input event.

(iii) The system G is a static system characterized by the transfer functions

$$g: I * \Pi_R \rightarrow \Pi_R * \Pi_I * I \qquad \text{and}$$

$$\bar{g}: I * \Pi_R \rightarrow \Pi_R * \Pi_I * I$$

The function g represents the internal activity of associating a reward-ordering of I with each candidate for response.

(iv) The system L is a core-transition system characterized by the core-transition function

$$\ell: I * Q \rightarrow Q$$

The function ℓ may be interpreted to represent the internal activity of changing the memory-state on account of the present memory-state and current input event.

(v) The system E is static with transfer function

$$e: \; I * Q * \Pi_R \to \Pi_R * \Pi_I * \bar{1}$$

where $\bar{1}$ represents the closed interval on the real line, [0,1].

The function e represents the internal activity of associating a probability of occurrence to each element of I, given a candidate for response from R and a current input event and a specified state of knowledge.

(vi) The system D is static and is characterized by the composition of transfer functions $d' \cdot m$ where

$$d': I * \Pi_R * \Pi_R * \Pi_I * I^{*\Pi_R * \Pi_I} * I^{*\Pi_R * \Pi_I * \bar{1}} \to$$

$$\Pi_{(I * R) * (I * R)}$$

and

$$m: \Pi_{(I * R) * (I * R)} \to R$$

The function $d' \cdot m$ represents the decision making activity of P insofar as P employs the outputs of the subsystems O, G and E to select a response.

It may be shown [9] that a person-model P is a sequential system with
(i) $\delta = \ell$
(ii) $\gamma = f$ where $f: I * Q \to R$ such that

$$f(i,q) = m\,(d'(i,o(i),\; g\,(i,o(i)),\; \bar{g}\,(i,o(i)), e(i,q,o(i))))\,.$$

Since we later consider multiple feedback interconnections of P-models, this investigation is formally similar to that of feedback-interacting sequential machines for which there is little evidence of study in the literature.

In order to model interaction of person-models so as to simulate group behavior, the notion of "interaction" itself needs to be formalized. This notion can be represented for two person-models by the diagram shown as Figure 2.

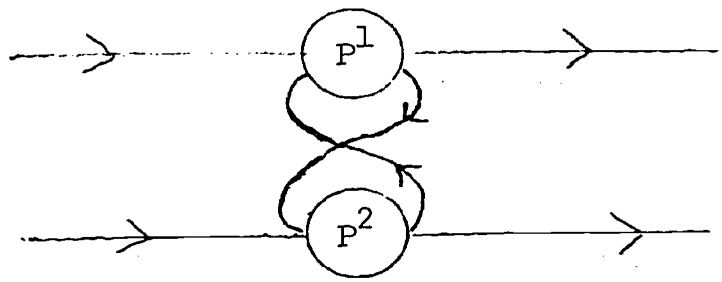


Fig. 2
2-Group Model Block Diagram

Denote the interaction of P^1 and P^2 by $P^1 \wedge P^2$. If we let Q^1, ℓ^1, and f^1 refer to the set of states, the next state function and the output function of P^1, and similarly for P^2, then it can be shown (9) that $P^1 \wedge P^2$ is a sequential system with

(i) $\delta = \ell$ where $\ell: (I^2 * I^1) * (Q^1 * Q^2) \to Q^1 * Q^2$ such that

$$\ell((i',i),(q,q')) = \ell^1(i,q),\ \ell^2(i',q'))$$

(ii) $\gamma = f$ where $f: (I^2 * I^1) * (Q^1 * Q^2) \to R^1 * R^2$ such that

$$f((i',i),(q,q')) = (f^1(i,q),\ f^2(i',q'))$$

The formalization of $P^1 \wedge P^2$ provides a framework for the formal study of both the static and dynamic characteristics of group behavior. That is, static behavior is that behavior which may be viewed as independent of time, and so is not associated with a learning capability for any of the individuals. This sort of behavior approximates the "steady state" or "equilibrium" situation for a group. (It should be noted that the investigation of static behavior is related to the "algebraic" models used by other authors.) The dynamic characteristics of group behavior are those which involve time as a dependent variable, and so can be associated with a learning capability for some or all of the individual members. This type of behavior approximates the situations in groups where at least one member is still assimilating information and possibly modifying his behavior accordingly.

Results from systems theory indicate that the most pervasive dynamic characteristic of systems is that of stability. That is, behavior is stable when it "settles down" to some steady state or equilibrium. In our own paper [10], necessary conditions on the internal characteristics of each member of a dyad are given for the dyad to settle down into some steady state situation.

Another important characteristic of dynamic systems is the notion of controllability. The behavior of a group may be said to be controllable if the "state" of the group may be modified (e.g., changed from "clique-ish" group to "cohesive" group) by some sequence of "external" or "environmental" inputs. The controllability of a person-model is formalized in our research and the controllability of n-groups is discussed.

References

1. Fararo, T. J. "Theory of Status." *General Systems*, Vol. 12, 1968.

2. Gullahorn, J. "Initial Efforts in Validating a Computer Model of Social Behavior." *Psychological Reports*, Vol. 19, p. 786, 1966.

3. Homans, G. C. *The Human Group*. New York, Harcourt, Brace and World, 1950.

4. Homans, G. C. *Social Behavior: Its Elementary Forms.* New York, Harcourt, Brace and World, 1961.

5. Kelly, G., and Bannister, D. *The Psychology of Personal Constructs.* London, Academic Press, 1968.

6. Miller, G., Galanter, E., and Pribram, K. *Plans and the Structure of Behavior*. New York, Holt, 1960.

7. Raiffa, H. *Decision Analysis*. Reading, Mass., Addison-Wesley, 1968.

8. Simon, H. *Models of Man - - Social and Rational*. New York, Wiley, 1957.

9. Talavage, J. J. *Social Aspects of Classroom Organization*. Final report for Office of Education Grant No. OEG-4-70-0010(057), Georgia Institute of Technology, October, 1970.

10. Talavage, J. J. "A Theory of Dynamic Group Behavior." (Forthcoming.)

STUDIES IN SEMIOTICS AND LINGUISTICS

Information Measurement and Value

P. Zunde

In order to measure information value it is neccessary to introduce some kind of utility function which would be defined on the Cartesian product set W x X x M, where: W is a set of objects elements of which are sign events or signs; X is a set of objects which are designata of the set of signs W; and M is a set representing the interpreters in terms of a complex structure of relationships including their internal states, their valuation system, and their goals. The utility function would then associate with the measured amount of information a specified quantity as its "value." In other words, this function should be so constructed that it would map the amount of information determined on syntactic and semantic levels into some set of pragmatic "values," reflecting the valuation judgments of the users in light of their goals, preferences, criteria, etc.

Formalization of the above-described process has been a principal goal of one area of our research. That the task we have chosen is a complex and difficult one may be suggested by a review of the work of such authors as Kharkevich [5], Hurley [4], Gavurin [2], Stratonovich [6], Bongard [1], and others, as summarized briefly below.

The Kharkevich approach to the problem culminated in the suggestion that information value be measured in terms of the increased probability of achieving certain goals. If, prior to the receipt of information, the probability of achieving a goal was p_o, and if after the receipt of information it became equal to p_1, then the value of information is assumed to be equal to:

$$V(i) = \log_2 p_1 - \log_2 p_o = \log_2 \frac{p_1}{p_o}$$

If we consider only situations with equiprobable outcomes, for which $N_o = \frac{1}{p_o}$ is the number of outcomes after receiving information, then

$$V(i) = \log_2 \frac{N_o}{N_1}$$

In this case the value of information is measured in units of the amount of information, i.e., in bits. Although superficially the above expression is similar to the measure of syntactic nature, so that the difference between the value of information and that of the amount of information might seem to disappear, the explanation is in the fact that the utility function appears here in the disguised form of an identity transformation, transforming bits of information amount into bits of information value.

A number of proposed methods in the theory of information value could be described as optimal strategy methods. Proponents of this approach to information value theory (such as W. V. Hurley, M. K. Gavurin, R. L. Stratonovich, and others) combine statistical information theory with decision theory, game theory, or optimal control theory. In all cases, either an appropriate penalty or a utility function is introduced, and the information is evaluated in the process of the minimization of losses or maximization of gains. The highest value is assigned to the information leading to a strategy which minimizes the losses or maximizes the gains.

The algorithmic approach to information value developed by M. M. Bongard and others is conceptually most remote from the statistical information theory. Whereas the previous two approaches assumed that the amount of information affects the value measure, Bongard postulated that the received amount of information might have no value at all, i.e., $V(i) = 0$, or might even have a negative value, i.e., $V(i) < 0$. The latter case is based on the assumption that the uncertainty can not only decrease but also increase - - if, for example, false information (i.e., misinformation) is received. Another important feature of Bongard's theory is the connection of the statistical characteristics of the amount of information with the algorithms of verification of problem solutions. The essence of this is as follows. Problem solving with respect to some element m_i consists of finding some subset A_j of a given finite set M such that $m_i \varepsilon A_j$. A certain probability distribution is defined on the set M which assigns measures to the elements m_j. A subset A_i is selected randomly and is substituted together with the object m_i into the solution algorithm. The solution algorithm determines whether m_i belongs to the subset A_j or not. Selections of subsets A_j are continued until a solution is found. Solution algorithms differ with respect to the initial probability distributions of selecting subsets A_j and the rules of their changes. The number of applications of the solution algorithm, leading to the problem solution, is a random variable $X(m_i)$, i.e.,

$$N(m_i) = \log E[X(m_i)]$$

The solution algorithm is then the vehicle which Bongard uses to introduce the concept of useful information. Namely, the degree of uncertainty which is removed as a result of obtaining the correct solution defines the value of useful information - - measured, again, as the amount of information, in bits. Information is useful if it reduces the uncertainty of the solution algorithm. The change of uncertainty is related to the redistribution of probabilities in the solution algorithm.

It is also being argued that information value can be measured by attaching appropriate utility functions to semantic information measures. If p is some proposition, the information value V(p) is defined as a function satisfying the following conditions

i) $0 \leq V(p) \leq 1$;
ii) $V(p) = 0$, if and only if p is true or false in the given language;
iii) $V(p) = 1$, if and only if all alternatives have equal probabilities. (Harrah, [3].)

As has been said, common to all proposed methods of measuring pragmatic information is the assumption of some utility function, given either explicitly (for example, in the form of a pay-off matrix in the game-theoretic approach), or implicitly (for example, in terms of goal achievement probabilities). Unfortunately, none of the suggested measures are versatile and comprehensive enough to claim universal applicability. Thus, the measure introduced by Kharkewich does not reflect the dependence of the information value on the significance of the goals. For any two goals, which have equal a priori and a posteriori probabilities, the information would have the same value, even though one of these goals might be trivial and the other most significant. The method proposed by Bongard does not take the internal states of the interpreter into consideration, nor does it reflect the dependence of value judments on various value realms. Similar limitations of one or another kind are inherent in the other information utility measures which were described in the literature, i.e., none of these measures satisfies the requirement of simultaneous functional dependence on the interpreter's internal states, his value judgments, and his sets of goals.

In conclusion, we may note that even though we are on the threshold of the new "information age," relatively little is known about the essence of information and various information processes, in which sign processes play a predominant role. Our own research effort [7,8,9] could be categorized as an attempt to analyze some of the aspects of these processes and to relate them to the problem of information measurement.

It is often claimed that science begins with measurement. If this is so, the stake which information science claims is not yet completely justified. To-date, the more successful attempts to quantify information have been based on approaches which essentially have abstracted from the pragmatic nature of information processes, i.e., from the user and the utility of information. Such, for instance, are the information theory of Shannon, with its emphasis on syntactic aspect, or the semantic information theory proposed by Carnap and Bar-Hillel. Relatively little has been done to formalize the theory on the pragmatic level.

In general, the pragmatic level of information processes can be characterized by the interaction between the information transmitted, the interpreters or users of information with their systems of value judgments, and the goals and objectives which information serves to promote. This in turn is directly related to the concepts of information value and utility. One of the main objectives of our research in this area has been to clarify these concepts.

It should be pointed out that the problem of information value is not only of theoretical but also of great practical importance, since a better understanding of the pragmatic aspects of information processes is vital to a more efficient utilization of the available information resources and processes.

References

1. Bongard, M. M. "On the Concept of Useful Information". *Problemy Kibernetiki*, No. 9, 1963, Moscow, pp. 71-102.

2. Gavurin, M. K. "On Information Value." *Vestnik Leningradskogo Universitete*, Sar. Mat., Nexh. i Astr., No. 19, 1963, pp. 27-34.

3. Harrah, D. *Communication. A Logical Model.* Cambridge, Mass., M.I.T. Press, 1967.

4. Hurley, W. V. "A Mathematical Theory of the Value of Information." *Int. Journal of Computer Mathem.*, Vol. 1, No. 2, 1965, pp. 97-146.

5. Kharkevich, A. A. "O cennosti informacii." *Problemy Kibernetiki*, No. 4, 1960, Moscow, pp. 53-58.

6. Stratonovich, R. L. "On Information Value." *Izv. Akad. Nauk. SSSR, Technicheskaie Kibernetika*, No. 5, 1965, pp, 3-12.

7. Zunde, Pranas. "Information Measurement and Value." A paper presented at the Third Annual Symposium of the American Society for Cybernetics in Gaithersburg, Md., October 14-16, 1969. To appear in *Communications of the American Society for Cybernetics*.

8. Zunde, Pranas. "Semantic and Pragmatic Use of Syntactic Information Measures." In: *Proceedings of the Meeting on Information Measures*, Kitchener-Waterloo, Ontario, Canada, 1970.

9. Zunde, Pranas. "On Signs, Information and Information Measures." *Instrumentation Society of America Transactions* 10:189-193, 1971.

Utilization of Semantic Information Measures (SIM)

P. J. Siegmann, M. Stapleton

The main emphasis of this project has been the development and utilization of semantic information measures to describe performance in man-machine interaction, with emphasis on human information processing. A conversational experimental methodology has been developed which permits the subject to control his rate of information acquisition by a query procedure. Project results demonstrate the potential value of SIM as a measure of man-system interaction. The following steps have been taken:

1. A literature search has been completed on the effects of negation on human information processing. A review of this literature provides the basis for design of a set of experiments to determine the effects of negation in information processing. The rationale and procedures required for the experiments are explicated in detail.

2. A flowchart analysis for a computer program to calculate semantic information has been completed. A computer program implementing the flowchart has been completed and is being tested at present.

3. A set of standard instructions for utilizing SIM with human subjects has been developed. These instructions are designed to insure that Ss have adequate understanding of the query procedure and have sufficient practice in controlling the rate of information acquisition in conversational experiments.

4. An experimental design has been explicated to determine the human use of disjunction in information processing. The experiment permits a comparison of inclusive and exclusive disjunction.

5. A measure has been developed for evaluating human performance in inducing generalizations from samples in an infinite language. This measure permits an evaluation of the efficiency of human performance in induction and provides a basis for experimental work in this area of human processing.

6. An initial review of the concept formation literature as it relates to information processing has been completed. This review is being expounded to include a critique from the view of SIM application.

7. A report has been completed on the implications of SIM for stimulus-response methodology.

References

1. Hovland, C. I. "A Communication Analysis of Concept Learning." *Psychological Reviews*, 59:461-472 (1952).

2. Hovland, C. I., and Weiss, W. "Transmission of Information Concerning Concepts Through Positive and Negative Instances." *Journal of Experimental Psychology*, 45:175-182 (1953).

3. Wales, R. J., and Grieve, R. "What Is So Difficult About Negation?" *Perception and Psychophysics*, 6:327-331 (1969).

4. Wason, P. C. "Psychological Aspects of Negation: An Experimental Enquiry and Some Practical Applications." *Communication Research Centre Papers*, PPer II, University College, London, 1962.

Taxonomy of Linguistic Structural Theories

R. Hawkey

Transformational grammar and tagmemics - - the two grammatical theories which have gained the greatest degree of acceptance among American linguists - - are primarily analytical in approach: for some language, a set of synchronic subject-data, finite or infinite, is separated into classes, and the classes interrelated according to one or another rationale provided by the theory; the classes to which subparts of a linguistic sequence are assigned and the interrelationships among these classes constitute the relevant (i.e., in terms of the theory) structure for the sequence. For most linguistic sequences, the structure specified by one of the above-mentioned grammatical theories will differ considerably from the structure specified by the other theory. The nature of the differences and similarities in the structural assignments of the grammatical theories is the subject of this research project, which has been undertaken to answer the following specific questions:

1. What are the formal, distributional, and functional properties of the members of linguistic classes specified by the respective theories? To what extent do these properties serve within the theory for the specification of classes, and to what extent are the properties accidental in reference to the theory?

2. In what sense may the linguistic classes of a specified structure be said to be interrelated? On what criteria specified by the theory are the linguistic classes related, and which properties characterize a structural relationship?

3. To what extent are the three theories comparable with one another, with traditional grammar, and with other generally accepted formal and structural theories (such as phonetics, phonemics, and morphemics), in terms of the universality of classes and relationships? To what extent are the classes and relationships specified for one language similar, on formal, distributional, and functional criteria, to the classes and relationships specified for other languages?

Our first report resulting from this study has been entitled "Critique of Certain Basic Notions in Chomsky's *Syntactic Structures*" [3]; it examines in detail Chomsky's [1,2] early theoretical position on grammaticality vs. acceptability; marginally grammatical sentences; explanatory power of grammar-assigned structures; justification for formulation of structural units; and motivation for permitting infinite recursion in generative grammars.

Our general conclusion from this research has been that, if it is granted that the grammaticality of a sentence is a function of the sentence's acceptability to a native speaker, then it must also be granted that (1) sentences are not to be considered grammatical simply because they are produced by a grammar; (2) a grammar itself cannot decide cases of marginal grammaticality; (3) a complex sentence containing a number of sentential recursions cannot be considered grammatical simply because each recursion has been specified by an elementary rule.

Chomsky's requirement that a grammar produce only and all of the grammatical sentences of a language, if understood literally, is seen to be unrealistic, since there appears to be no method by which to test whether or not the requirement has been met.

The structure which is assigned to sentences by a generative grammar is at least partially determined by the format (rewrite rules) in which statements of the grammar are expressed, and in many cases, categories are posited solely for purposes of convenience and simplicity, rather than on the basis of the characterizing properties of sentence elements. Supposed explanations of such phenomena as constructional homonymity, based on the assigning of more than one structural description to a sequence, are seen to be meaningless, and indicate only that certain terminal strings can be pointed out by the grammar in more than one way.

References:

1. Chomsky, Noam. *Syntactic Structures.* s'-Gravenhage, Mouton, 1962.

2. Chomsky, Noam. *Aspects of the Theory of Syntax.* Cambridge, MIT Press, 1965.

3. Hawkey, R. "Critique of Certain Basic Notions in Chomsky's *Syntactic Structures.*" To appear in *Folia Linguistica.*

Studies of Natural Language

J. Gough, C. Pearson, B. Saulnier

This section deals with the further development during the past year of our ostensive grammar (see references). Let us recall that we are envisaging a grammar in which the base component is designed to recapture natural language strings initially as logical proper noun structures. We begin with a new set of primitives, the universal selective indexes; indeed, these are indexical symbols in which the various dimensions of semiosis (the syntactic, the semantic, the pragmatic) appear to merge.

The indexical symbol THIS is regarded as representing the primitive level of linguistic symbolism, being the value of an idealized act of pointing or ostension and thus of unique referencing. Moreover, representing essentially the category of nominal expressions, the element THIS expresses unique quantification. Beyond the element THIS, we introduce and operate with the ostensive doublets HERE/NOW and HENCE/HITHER. We regard these pairs as representing complexified acts of ostension, when induced on the element THIS. We thereby attain to a linguistic level capable of expressing the semantics of space/time and origin/goal. The semantics of the THIS is increased in the process, yet not lost, for it is implied by these later ostensive pairs.

Part of the semiotic package of indexical symbols is their potential to combine with one another and to function as arguments. They thereby provide a context sensitive to the generation of relative terms, verbal and prepositional alike. We regard this as a crucial area of our research, since it provides a hierarchical generative scheme of relations that can be ordered by virtue of the indexical symbol schemata.

Our direction of research is thus motivated by a desire to identify in natural language the mechanism that creates the primitive level of symbolism, a symbolism which is both prior to and necessary for any ultimate linguistic expression of fact, knowledge, truth and falsity. The abstract grammar that we are envisaging continues to assume the appearance of a concept generator and processor that chronologically precedes the grammar of any specific language. This is leading us to examine the universal aspects of the proposed scheme.

(1) Internal Ostensive Semantic Configurations (IOSC) and the Generation of Prepositional Categories. The generation of ostensive schemata gives rise to ostensive semantic configurations in the base grammar. At present, these appear to be of two types: (1) internal OSC's, those possessing a design internal to a single ostensive categorical value (object, space/time, origin/goal) and (2) external OSC's, those possessing a design external to distinct ostensive categorical values when combined. One major payoff of IOSC's is the resultant

hierarchical generation of classes of prepositions. External OSC's lead to the generation of conceptual verbal categories. We will discuss only IOSC's in this section.

We begin with the simple ostension value THIS (IT). We regard the IT as a slot for variables. Assigning to it the variable j, we obtain our primitive IOSC:

IOSC-1

$$\text{THIS } (j)$$

the value of a single act of ostension. We can immediately increase its semantics by inducing on it a space/time ostension value by dint of a nonzero semantic transformation of the form:

$$\begin{gathered}\text{THIS } (j) \\ \Downarrow \\ \text{HERE/NOW } (\text{THIS } (j))\end{gathered}$$

and obtain the new internal OSC:

IOSC-2

$$\ll \text{HERE, AT THIS } (j) >,$$

$$< \text{NOW, AT THIS } (t_j) \gg$$

Here the THIS of the HERE element retains the original index j, with the THIS of the NOW element being indexed as t (time) of j.

The more complex space/time ostensive operator determines the ostensive value of the schema. In order for the element THIS (j), when distributed over the HERE and NOW, to have a semantic component equivalent to each, an appropriate locative and time categorical element must be combined with it. Thus, a prepositional category, which we designate AT, must be force-generated to realize the semantic equivalence. This category is identical for both space and time, thereby marking the significant semantic roles of the HERE and NOW elements. While positing such a prepositional category, we do not intend that it eventually surface as such. It may be realized by any number of overt linguistic devices. The significant feature at this point is that we obtain an internal OSC characterized by an idealized ostensive semantic value.

The semantics of the IOSC-2 can be increased in turn by inducing a HENCE/HITHER ostensive value on either the HERE or the NOW or on both components. Transforming only the space component, we have:

$$\text{HENCE/HITHER} \ll \text{HERE, AT THIS } (j) >,$$

$$< \text{NOW, AT THIS } (t_j) \gg$$

and obtain IOSC-3:

IOSC-3

$$\lll \text{HENCE, FROM HERE, FROM AT THIS } (o_j) >,$$

$$< \text{HITHER, TO HERE, TO AT THIS } (g_j) \gg,$$

$$< \text{NOW, AT THIS } (t_{o_j, g_j}) \gg$$

Here we have induced an origin-goal (HENCE/HITHER) ostensive semantics on a simple locative ostensive (HERE) semantics, while keeping the NOW component fixed. We index the origin componential THIS as o_j (origin of j) and the goal componential THIS as g_j (goal of j). As will become evident later, this scheme implies that both the origin and the goal are parts of an extended j. In this transformation, we force-generate the prepositional categorical doublet FROM...TO... and thereby obtain an equal ostensive value among the different linguistic expressions of the schema. Moreover, the functional semantics of both THIS and HERE is increased.

It is also possible to increase only the time component of IOSC-2, whereby we obtain:

IOSC-4

$$\ll \text{HERE, AT THIS } (j) >, \ll \text{HENCE, FROM NOW,}$$

$$\text{FROM AT THIS } (t_{o_j}) >, < \text{HITHER, TO NOW,}$$

$$\text{TO AT THIS } (t_{g_j}) \ggg$$

Here the origin/goal semantics is induced solely on the time component, with the HERE component left unchanged. Once again, as in the case of IOSC-3, the context sensitivity resulting from the HENCE/HITHER doublet force-generates the same prepositional categorical pair for the time element. The importance of the distinctive semantical roles of HERE and NOW at this level is again evident.

Finally, combining IOSC-3 and IOSC-4 gives:

IOSC-5

$$\begin{aligned}
&\langle\langle\langle \text{HENCE, FROM HERE, FROM AT THIS } (o_j) \rangle, \\
&\quad \text{HENCE, FROM NOW, FROM AT THIS } (t_{o_j}) \rangle\rangle, \\
&\langle\langle \text{HITHER, TO HERE, TO AT THIS } (g_j) \rangle, \\
&\quad \langle \text{HITHER, TO NOW, TO AT THIS } (t_{g_j}) \rangle\rangle\rangle
\end{aligned}$$

where an origin/goal semantics is induced on both the space and the time component.

In the derivation of the entire increasing-semantic scheme of ostensive complexification, we also obtain a vertical internal ostensive semantic configuration. The last or the lowest level of a given derivation determines the semantic value of the entire derivation. Nevertheless, we may wish to have the first level (or any other level) in the derivation be able to represent the semantics of the most complex level. This would correspond to postulating synonomy among levels. This becomes possible, if an appropriate noun and, if necessary, an appropriate prepositional subcategory is selected. For example, the string "this run from here at time t_i to here at time t_j" may be rendered equally well by "this run (over) this distance in this time." The factors determining this choice are now being explored. Yet we may say that a derivation will retain its entire scheme as, e.g., in

$$\begin{tikzcd}
& \text{IOSC-1} \arrow[d, Rightarrow] & \\
& \text{IOSC-2} \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] \arrow[dd, Rightarrow] & \\
\text{IOSC-3} \arrow[dr, Rightarrow] & & \text{IOSC-4} \arrow[dl, Rightarrow] \\
& \text{IOSC-5} &
\end{tikzcd}$$

where a zero-semantic transformation may combine the first level with the last to give us:

<< IOSC-5 > , < IOSC-1 >>

Here the IOSC-5 dominates the semantics of the pair and eventually dictates the choice of the simple nominal expression used to express this semantics. Indeed, this appears to be the second level of noun generation in the model of the ostensive grammar component. (For the first level of noun generation, see references 9 and 11.)

One additional remark regarding the IOSC's. We are still exploring the various combinations that are possible within the framework of our primitive elements and within the range of a single derivation. This will hold the key to the generation of more complex prepositions.

(2) <u>External OSC's and the Generation of Verbal Categories</u>. To appreciate and obtain external OSC's we start with the schema:

< THIS (IT), THIS (IT) >

an ordered pair of ostensive values with initially identical IOSC's. The semantic values typical of these identical IOSC's may be generated here externally as the verbal conceptual category of identity, giving us the protosentential schema equivalent to the English surface string "this is this," with the 'is' being tenseless at this level. We thus obtain our first EOSC:

EOSC-1

< IOSC-1, IOSC-I >

the semantic configuration necessary for the generation of the relative term of identity.

We can now increase the semantics of the second IOSC of the ordered pair and obtain:

EOSC-2

< IOSC-1, IOSC-2 >

⇓

EOSC-3

< IOSC-1, IOSC-3 >

or

EOSC-4

<IOSC-1, IOSC-4>

EOSC-5 ⇓

<IOSC-1, IOSC-5>

a list of external ostensive semantic configurations (EOSC's) paralleling that of the IOSC's.

We now exploit the semantics of the various configurations, in which the indexical symbols function as arguments, to generate a set of primitive verbal relative-term conceptual categories, which set we designate the BE class of predicates. The class is so designated, since the verbal BE, being the most neutral of verbals, is capable of expressing the entire range of semantics covered by the various subclasses of verbals.

We now generate the following set of primitive verbal categories by dint of zero-semantic transformations:

<EOSC-1>	⇒	BEI <EOSC-1>
<EOSC-2>	⇒	$\left\{\begin{matrix}\text{BESP}\\ \text{BETP}\end{matrix}\right\}$ <EOSC-2>
<EOSC-3>	⇒	BESE <EOSC-3>
<EOSC-4>	⇒	BETE <EOSC-4>
<EOSC-5>	⇒	BESTM <EOSC-5>

Here we have generated the verbal category BEI denoting 'the BE of identity'; BESP 'the BE of space (S) point (P) localization; BETP 'the BE of time (T) point (P) occurrence'; BESE 'the BE of space extension (E)'; BETE 'the BE of time extension (E)'; and BESTM 'the BE of space/time movement'. We must remark once more that the predicate subclasses represent conceptual categories with further syntactic potential.

These verbal categories may then be defined by phrase-structure rewrite rules of the form:

BEI → <THIS (IT), THIS (IT) BEING IDENTICAL>

BESP → <THIS (IT), THIS (IT) BEING LOCAL>

and the like. We take the relation joining the ordered pair to be that of identity. It is interesting to note that the right-hand side of these rules has a structure identical to that proposed by Gough and Chiaraviglio (1970) and Gough and DeMillo (1970) for the English noun phrase. We should also note that the verbal conceptual categories are generated ahead of specific noun subclasses.

The power inherent in this type of verbal generation permits us to proceed in a number of directions: (1) A direct act of ostension represented by a verbal; (2) Intralinguistic idealized ostension of a verbal category in terms of its EOSC; (3) Generation of a specific verbal category together with its EOSC as a potential sentence structure; and (4) Generation of specific subclasses of verbals. Directions (1) and (2) represent counterparts, with (1) relegated to extralinguistic acts of ostension and (2) to intralinguistic ostension or, better, cross-referencing. This division corresponds to that of ostensive and definitional knowledge.

Taking BESTM < EOSC-5 > as an example, direction (1) ultimately leads to strings such as "this is moving," whereas (2) gives "this (BESTM < EOSC-5 >) is (defined as) moving'; direction (3) will generate the structure inherent in strings of the form "this moves from here/now to here/now," and direction (4) generates subclasses of predicates such as "running is moving." Directions (3) and (4) may combine.

As in the case of prepositional subclasses, we are also exploring the generation of subclasses of verbals.

(3) Ostensive-Based Generation of Causation. At present we are suggesting only two types of causation: (1) internal causation and (2) external causation. We are well aware that the concept of causation represents a complex semantical problem, yet it is our opinion that the linguistic conceptualization of causation can at least be initiated within the framework of primitive schemata.

Internal causation is so named because it involves only the arguments that are already present in our original schemata. External causation, on the other hand, requires the introduction of a new THIS (IT) element that does not appear in any of the original schemata. That is, it has at least the form < THIS (IT) , PRED < EOSC >>, where the initial THIS (IT) is not an argument present in the given EOSC. This is not to say that we do not envisage combinations of the two types. To date we have only explored in some detail internal causation. We treat it here firstly as reflexive internal causation (RIC) and secondly as polar internal causation (PIC).

(a) Reflexive Internal Causation. Internal causation is theoretically possible within any of the EOSC schemata so far developed. However, we will exemplify RIC, using only schema EOSC-5, since it is the very schema underlying PIC as well. This will provide a comparison of the two types.

Starting with the EOSC-5 schema and its predicate BESTM:

BESTM < THIS (i) , <<< HENCE, FROM HERE, FROM AT THIS (o_j) >,

< HENCE, FROM NOW, FROM AT THIS (t_{o_j}) >> ,

<< HITHER, TO HERE, TO AT THIS (g_j) > ,

< HITHER, TO NOW, TO AT THIS (t_{g_j}) >>>>

where BESTM denotes the conceptual category of movement or motion, we transform it by copying the THIS (i) element and exporting it to the head position such that we have:

< THIS (i) , BESTM < EOSC-5 >>

Here we use the abbreviation EOSC-5 for the lengthy pairings. The copied, exported THIS (i) element is now set in a relationship to EOSC-5 and its predicate, which relationship we designate as LET. The latter predicate can be generated by a transformation rule of the form:

< THIS (i) , BESTM < EOSC-5 >> ⇒

LET < THIS (i) , BESTM < EOSC-5 >>

Here we bank on the newly established context sensitivity to generate the LET predicate. We can then interpret this transform roughly as "this (i) let this (i) move from here/now to here/now."

We may note again that the concept of LET requires further study relative to its various modalities. Its form above encompasses a number of varieties.

(b) Polar Internal Causation (PIC). We wish to demonstrate that the semantics of the schema BESTM <EOSC-5> underlies such polar predicate types as GIVE and RECEIVE. That is, they both share a common semantic structure prior to their differentiation. They become differentiated as soon as we decide whether the THIS (o_j) of the HENCE component or the THIS (g_j) of the HITHER component is to be copied and exported to the new head position and then set in a new relationship to the original schema.

Beginning once more with the schema:

BESTM < EOSC-5 >

we transform it to:

< THIS (o_j) , BESTM < EOSC-5 >>

where the THIS (o_j) comes from the HENCE component. By transformation rule, we obtain:

LET < THIS (o_j), BESTM < EOSC-5 >>

which we read roughly as "this (o_j) let this (i) move from this (o_j)/now to this (g_j)/now." Here the LET conceptual category of causation added to the origin semantics inherent in the HENCE component forms the basic semantics for the concept of GIVING.

To obtain the predicate category RECEIVE, we begin with the same basic schema:

BESTM < EOSC-5 >

emphasizing again the common kernel semantics of movement for both GIVE and RECEIVE. This time, however, we copy and export to the head position the THIS (g_j) of the HITHER component and obtain by transformation the schema:

LET < THIS (g_j), BESTM EOSC-5 >

which may be read roughly as "this (g_j) let this (i) move from this (o_j)/now to this (g_j)/now." Here the goal semantics is dominant. Moreover, the LET-causation is more that of PERMIT. Together these semantic notions form the minimum semantics for RECEIVE. We should also point out that the above derivations represent the chronolcgy of the polar derivation of GIVE/RECEIVE.

The above schemata should give us the semantics basic to the various subclasses with members such as "give, receive, buy, sell," and the like. This will probably require the introduction of a THUS modal generator, which we are examining at present. Of course, a modal operator of this type would operate on other schemata besides the ones just given. This aspect is also under examination.

(4) Concluding Remarks. We have recently begun to examine the relations underlying case generation. We feel that the establishment of the various schemata with their respective relationships will lead among other things to the generation of a case system, where necessary, as well as to other relational schemes such as found in Finnish and Hungarian. It is our opinion that an inventory of cases in the deep structure is unrealistic, since it can be demonstrated that they are not all initially equally primitive.

Allied with the study of case is that dealing with the generation of subclasses of prepositions. Here we have begun to examine further classes of prepositions relative to our initial two classes AT and FROM...TO.... We are concentrating on such prepositional relationship as expressed by IN, ON, INTO, and the like.

We have initiated formalization of the grammar, though this will proceed rather slowly until more structure is identified and generative rules are provided.

References

1. Chiaraviglio, L., and Gough, J., Jr. "Towards a Syntax-Monitored Semantic Pattern Recognition." In: *Progress of Cybernetics: Proceedings of the International Congress of Cybernetics, 1969* (ed., J. Rose), vol. III, London, Gordon & Breach, 1970, pp. 1267-1279.

2. Georgia Institute of Technology. Science Information Research Center, *Research 1969/1970: Annual Progress Report.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report, GITIS-70-09. 84 p.

3. Gough, J., Jr. *A Study of the Intraclass Structural Potential of the German Locative Adverb.* Unpublished doctoral dissertation, Harvard University, 1965.

4. Gough, J., Jr. "On the German Locative: A Study in Symbols." *Mechanical Translation,* 10:68-84, (1967).

5. Gough, J., Jr. "The Syntax-Based Semantics of the English Determiners 'Ø', 'A', 'THE'." *Papers in Linguistics* 1(1):41-48 (July 1969).

6. Gough, J., Jr. "The Referential-based Generation of the English Locative." Paper presented at the Third Southeastern Conference on Linguistics, University of North Carolina, Raleigh, N. C. (April 1970).

7. Gough, J., Jr. *A Semiotic Base for Grammar*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report, GITIS-70-05. 11 p. (PB 190 817).

8. Gough, J., Jr. "The Referential-Based Generation of Three-Place Predicates of the Type GIVE, RECEIVE and the Like." Paper presented at the Fifth Southeastern Conference on Linguistics, University of Maryland, College Park, Maryland (May 1971).

9. Gough, J., Jr., and Chiaraviglio, L. "On the Base Referential Structure of the English Noun Phrase." *Foundations of Language* 6(4):447-462 (November 1970).

10. Gough, J., Jr., and DeMillo, R. "Towards an Ostensive Grammar." Paper presented at the Eighth Annual Meeting of the Association for Computational Linguistics, Ohio State University, Columbus, Ohio (July 1970).

11. Pearson, C. R. "Ostensive Information: A Measurement of Information associated with Computer Addressing, Indexing, and Data Structuring." Paper presented at the Tenth Annual Conference of the Southeastern Region of the ACM, Mobile, Alabama (June 1971).

Development of an Ostensive Calculus

C. Pearson, J. Gough

It has been our intention to formulate a calculus for analyzing, interpreting, and developing the semantics of ostensive grammar. The work has been divided into three phases corresponding to development of the tools to handle the three primitive operators $\perp$ (This); $\vdash\!N$ (Here/Now); and $\vdash\!\!\dashv$ (Hence/Hither). This paper reports on the results of the first phase.

One of the main goals of ostensive grammar is to represent the processes of natural language in a way that explicitly displays the three dimensions of semiosis: semantics, syntactics, and pragmatics. The semantic representation is based on a concept of ostension (pointing). Thus, one of the primitive operators of our calculus must be a pointing operator. In our calculus, this is represented by "$\perp$" which corresponds linguistically to Peirce's universal and particular selective indexes. We feel that we now have adequate machinery to analyze the semantics of that part of ostensive grammar that employs only "$\perp$" of the three primitive operators.

Primitive Symbols

The THIS operator symbol "$\perp$" has already been mentioned as one of the primitive symbols of our calculus. Accordingly we do not try to define "$\perp$" from any symbols more primitive.

However, it may be worthwhile to discuss ostension further in order to permit an intuitive feel for its role in natural language. One feels that at the simplest level of semantics, namely the singular term, or proper noun, the meaning of the term is wrapped up in the object of the term itself; that is, by the extension of the singular terms. This can perhaps best be illustrated by a small scenario. If I am in a small room with several persons, one of whom has just volunteered to do some task, and another person asks me who has volunteered, I may answer by saying. "John Brown." If you then say, What do you mean?", I may answer most directly by pointing to John Brown and saying, "This person".

Now this illustrates the basic ostension process - - namely, pointing to one of Quine's middle-sized, real, physical objects [3] and at the same time (if one happens to speak one of the English languages, such as British or American) uttering the word, "This." One of the assumptions of ostensive grammar is that the semantics of all natural language can be explicated via a generalization of this process. Our symbol, "$\perp$", is meant to be the formalization (for the purposes of the calculus) of our intuitive notions of the basic ostension process.

In our intuitive discussion of ostension, we saw that pointing serves to select, from among all of the real, middle-sized, physical objects that exist (ostensible things), one particular one on which to focus our attention. We shall accordingly let "D" be a primitive symbol of the calculus and its intuitive meaning shall be that of the domain of ostensible things.

We shall assume that there is available an unlimited supply of variables which we shall symbolize by x,y,z, etc. The range of these variables will become clear as we proceed to develop the calculus.

Our basic development will be at the level of set theory. In particular we will assume the notions of "set" and "function" as understood. The symbol "F" will be primitive to our theory. We will think of F intuitively as a set of set functions. Set functions are functions from sets into sets. Finally we shall assume available a supply of symbols, "f", "g", "h", etc. We shall regard these as names of the individual set functions which belong to F.

To summarize, we now list our set of primitive symbols.

I. Primitive Symbols

1. ℩ (This)
2. D (Domain of Ostensible Things)
3. x,y,z, etc. (Variable Symbols)
4. F (A Set of Set Functions)
5. f,g,h, etc. (Functions From Sets Into Sets)

Primitive Predicates

Since D is to represent the domain of only those things which are real, middle-sized physical objects, we shall want some way of representing the more abstract entities. We shall accordingly assume the ability to form sets of things in D, ordered pairs of things in D, sets of ordered pairs of things in D, sets of sets of things in D and in general any finitely constructable set based ultimately on things in D. We can monitor this process via the set of function constants f,g,h, etc. For instance, let $\mathfrak{F}$ and $\mathfrak{G}$ be families of sets and let $f: \mathfrak{F} \rightarrow \mathfrak{G} \ni: \forall (S \in \mathfrak{F}): f(S) = \mathcal{P}(S)$, that is, for any given set S, f(S) corresponds to forming all of the possible subsets of S and treating them as a single collection. Hence f(D) is the collection of all possible sets of things in D. The only argument we shall use for the functions f,g,h, etc., is D and we shall never need to catalog the actual extent of D, hence we shall always treat it in a formal or symbolic manner, we shall treat D as primitive. The entities f(D), g(D), h(D), etc. will correspond to our intuitive notion of the various semantic categories. We shall see later that if f is given as in the above example, then f(D) will correspond to the category of concrete, absolute, general terms.

Now in explicating the semantics of a term of the language, we must realize that it may belong to any semantic category. Accordingly we will need to develop a means for selecting from among all the semantic categories of the language one particular one on which to focus our attention. This appears to be very close to our notion of ostension described intuitively above. We accordingly generalize the notion of ostension to include pointing to semantic categories. We formalize this notion in the linguistic string

$$"\perp(f \in F):\ P(f)"$$

This is a formal representation of such signs as <This (IT) BEING>; <This (IT) RELATES>; <This (IT) ABSTRACTS>; etc., commonly used in ostensive grammar.

If $f = 1$, i.e., $f(D) = D$, then the symbol "$\perp(x \in f(D)): P(x)$"

represents our earlier notion of pointing to one of the real, middle-sized physical objects that exist, or selecting it for our concentrated attention. Now we may generalize this notion of ostension in another way. If f is any function constant ostended by the semantic category selector "$\perp(f \in F): P(f)$," then we let

$$"\perp(x \in f\ (D)):\ P(x)"$$

ostend an individual member of that category. Note that an individual in f(D) need not correspond to an individual in D. For instance

$$"\perp(x \in \wp(D)):\ P(x)"$$

selects out a set of individuals in D.

We may thus list our two primitive predicates as:

II. Primitive Predicates

1.	$\perp(f \in F): P(f)$	<This(IT)BEING> <This(IT)RELATES> <This(IT)ABSTRACTS> etc.
2.	$\perp(x \in f(D)): P(x)$	<This(IT)BE> etc.

where

P(x) is the partial predicate "I am pointing to x."

Term Production Rules

In the phase 1 grammar there is only one basic production. This corresponds to a complete semantic ostension and is formed by a primitive predicate of type one juxtaposed to a primitive predicate of type two in which the category in which the second ostension takes place agrees with the category ostended by the first ostension. We can thus write this as

$$\text{"}\perp(f \varepsilon F).\ P(f):\ \perp(x \varepsilon f(D)).\ \ P(x)\text{"}$$

Hence, we see that a complete ostension is composed of two separate and distinct ostension processes. The first ostension selects the semantic category and the second ostension selects the unique individual in that category.

Let us look at several examples from ostensive grammar. First the ostensive structure <This(IT)BE> is actually a more complex structure than originally thought. We see it now as the double ostension structure:

$$\langle \text{This}(IT_1)\text{BE, This}(IT_2) \rangle$$

↑ Selects Category of Individuals — ↑ Selects Individual In D

Double Ostension:

1. <u>Selects Semantic Category</u> — 2. <u>Selects Single Unique Individual</u>

We next look at the ostensive structure for general terms:

$$\langle \text{This (IT)BEING} \rangle.$$

This also turns out to be a double ostensive structure similar to the previous one except that we direct the first ostension to pick out the category of "general term" and we then direct the second ostension to pick an individual out of this category.

$$\langle \text{This }(IT_1)\text{BEING, This }(IT_2) \rangle$$

↑ Selects Category of General Terms $f = \wp$ — ↑ Selects Single Subset of D

Double Ostension:

1. <u>Selects Semantic Category $f \varepsilon F$</u> — 2. <u>Selects Unique Individual in $F(D)$</u>

To summarize then, in the phase one grammar there is only one term production rule:

III. Term Production Rules

1. $\perp(f \varepsilon F).\ P(f): \perp(x \varepsilon f(D)).\ P(x)$

Predicate Transformation Rules

The principle derivation of ostensive grammar studies so far involves an ostensive copy transform and a predication of identity. In order to analyze this derivation we need to build up two more concepts: that of transformation of predicates and that of building sentences from predicates and terms. In this section we motivate the two predicate transformations required so far.

We look first at the semantics of the two American terms "Moon" and "natural satellite of the Earth." Now, "Moon" happens to be a singular term, i.e., in the American grammar, a proper noun. In this case the term designates that object of the sky that "Moon" names uniquely. On the other hand, "natural satellite of the Earth" is a general term, capable of denoting multiply. Hence the term designates that set of natural astronomical objects that are satellites of the Earth. This set i single element of the power set of D, i.e., it is a single, unique, subset of D. Now the astronomical facts just happen to be that although "natural satellite of the Earth" happens to be a general term, capable of denoting multiply, it in fact denotes only one individual. That is, it designates a singleton subset of D. There is a natural relationship between the ostensive semantics of "Moon" and of "natural satellite of the Earth" that we would desire to explicate. Since the designation of "Moon" is the Moon, and the designation of "natural satellite of the Earth" is the singleton set whose only member is the Moon, we see that this relationship is one between singleton set and member of that singleton set. This leads us to establish the following pair of transformation rules:

$$\perp(f \varepsilon F).P(f): \perp(x \varepsilon f(D).P(x) \Leftrightarrow \perp(2^f \varepsilon F).P(2^f): \perp(\{x\} \varepsilon 2^f(D)).P(\{x\}).$$

that is the abstracting rule:

$$\perp(f \varepsilon F).P(f): \perp(x \varepsilon f(D)).P(x) \Rightarrow \perp(2^f \varepsilon F).P(2^f).\ \perp(\{x\} \varepsilon 2^f(D)).P(\{x\})$$

and the concretizing rule:

$$\perp(2^f \varepsilon F).P(2^f): \perp(\{x\} \varepsilon 2^f(D)).P(\{x\}) \Rightarrow \perp(f \varepsilon F).P(f): \perp(x \varepsilon f(d)).P(x)$$

The abstracting rule has been applied implicitly in ostensive grammar to produce a proto-general term from a singular term; for example, "dog" from "this dog". But now we see that this rule may also be applied at any level - - for instance, to produce a proto-abstract term from a general term. Example: "doghood" from "dog."

In addition we now see that the concretizing rule may also be applied to reduce a proto-general term to a singular term or to reduce a proto-abstract term to a general term to give only two examples.

In summary then we have one pair of predicate transformations given by the symmetric form 1:

IV. Predicate Transformation Rules

1. $\perp(f \in F).P(f){:}\ \perp(x \in f(D)).P(x) \Leftrightarrow$

$\perp(2^f \in F).P(2^f){:}\ \perp(\{x\} \in 2^f(D)).P(\{x\})$

Sentence Production Rules

In this section we explicate the concept of building sentences in ostensive grammar from the terms and predicates of ostensive grammar.

In light of our goal of explaining, or deriving the ostensive structure that predicates identity between the results of two ostension processes:

< This(IT), This (IT) BEING >,

we will establish only two sentence production rules, producing sentences predicating ostensive identity between their two terms.

Gough [1] states that in the ostensive structure

<This(IT), This (IT) BEING>

the comma may be interpreted as the relation of equality of referential scope, i.e., "This = this being", or "This is this being." In other words, for this specific example, the singleton set of the individual which is the referential scope of the singular term represented ostensively by < This(IT)> is identical to the individual member of the power set of D, or the subset of D, which is the referential scope of the general term represented by the ostensive structure <This (IT) BEING> and whose referential scope includes just the one individual ostended by <This (IT)>. If therefore we let the operator "A" represent the abstracting transformation and if two ostension processes are such that $g = 2^f$ and $y = \{x\}$ we then specifically allow the following sentence to be constructed:

$A[\perp(f \in F).P(f){:}\ \perp(x \in f(D)).P(x)] =$

$\perp(g \in F).P(g){:}\ \perp(y \in g(D)).P(y).$

Ostensive grammar has not treated the inverse of this situation, i.e., the concretizing process whereby an abstract term of singular referential scope is ostended and the referential scope of that term is equated to the result of ostending the term of less abstractness whose designation is the referential scope of the abstract term. But the treatment is so similar to that already developed that we proceed to develop the sentence structure required to do this. It is conceivable that this could be a useful analytical tool for ostensive grammar.

Accordingly we specify if $f = 2^g$ and $x = \{ y \}$ and C represents the concretizing transformation, or the inverse of the abstracting transformation, then $C [\perp (f \varepsilon F).P(f): \perp(x \varepsilon f(D)).P(x)] = \perp(g \varepsilon F).P(g): \perp(y \varepsilon g(D)).P(y)$.
is a well-formed sentence in ostensive calculus.

The two well-formed sentences established so far are summarized here:

V. Sentence Production Rules (T)

1. If $g = 2^f$ and $y = \{ x \}$, then $A[\perp(f \varepsilon F).P(f): \perp(x \varepsilon f(D)).P(x)] = \perp(g \varepsilon F).P(g): \perp(y \varepsilon g(D)).P(y)$.
2. If $f = 2^g$ and $x = \{ y \}$, then $C[\perp(f \varepsilon F).P(f): \perp(x \varepsilon f(D)).P(x)] = \perp(g \varepsilon F).P(g): \perp(y \varepsilon g(D)).P(y)$.

Derivation of Ostensive Structures

This section was originally motivated by the following type of argument. We repeat it here for its pedagogical usefulness.

We can ostend the object of a general term with singular referential scope in two ways. We can enter the grammar directly at the general term level and ostend the individual singleton set designated by the general term; or we can enter the grammar at the singular term level, ostend the individual designated by the singular term, and abstract it to general term status by considering it as a singleton set. These two ostension processes should yield equivalent ostensive results. This latter process may be viewed as generating proto-general terms from individual terms.

Let $\perp(f \varepsilon F).P(f): \perp(x \varepsilon f(D)).P(x)$

be a term. We define an ostensive copy transform which operates on this term as follows: The ostensive copy transform produces an entirely new ostension process at the level of sets of entities ostended by the original term, thus yielding two distinct terms. Thus

$$\perp(f \varepsilon F).P(f): \perp(x \varepsilon f(D)).P(x) \xrightarrow{OCT}$$
$$\perp(f \varepsilon F).P(f): \perp(x \varepsilon f(D)).P(x),$$
$$\perp(2^f \varepsilon F).P(2^f): \perp(y \varepsilon 2^f(D)).P(y).$$

We next define an identity predication transform which operates on pairs of terms by abstracting the first term and then identifying the variable y of the second term with the variable $\{x\}$ of the abstract. Thus,

$$\iota(f \in F).P(f):\iota(x \in f(D)).P(x),\ \iota(2^f \in F).P(2^f):\iota(y \in 2^f(D)).P(y).\ \xRightarrow{\text{IPT}}$$
$$A[\iota(f \in F).P(f):\iota(x \in f(D)).P(x)],\ \iota(2^f \in F).P(2^f):\iota(\{x\} \in 2^f(D)).P(\{x\}).$$

We now see that by the rules of sentence production we are allowed to form one sentence from these two structures as follows:

$$A[\iota(f \in F).P(f):\iota(x \in f(D)).P(x)] =$$
$$\iota(2^f \in F).P(2^f):\ \iota(\{x\} \in 2^f(D)).P(\{x\}).$$

Let us see how this would work in a concrete example from ostensive grammar.

Let the term $\iota(f \in F).P(1):\ \iota(x \in 1(D)).P(x)$ be represented in ostensive grammar by the ostensive structure

< This BE This (IT)>

and the term $\iota(f \in F).P(\mathcal{P}):\ \iota(y \in \mathcal{P}(D)).P(y)$ be represented by < This BEING This (IT) >. We represent the ostensive copy transform by "$\xRightarrow{\text{OCT}}$" and the identity predication transform by "$\xRightarrow{\text{IPT}}$" and let the sentence developed above be represented by

<(<This BE This (IT)>),< This BEING This (IT)>> .

We can then reason as follows:

{ < This BE This (IT) > } $\xRightarrow{\text{OCT}}$ {<This BE This (IT) >, <This BEING This(IT)> } $\xRightarrow{\text{IPT}}$
{<(<This BE This (IT)>), <This BEING This (IT)>> }.

This represents one of the principal results of the first order ostensive grammar. Thus the phase one calculus has achieved at least one significant result.

Replacing each of the above by an older but less accurate although simpler notation we can state the above result as a theorem of ostensive calculus. Theorem. The following deduction of ostensive grammar is valid (in the sense of ostensive calculus).

1. < This (IT) BE > $\xRightarrow{\text{OCT}}$
2. < This (IT), This (IT) BEING > $\xRightarrow{\text{IPT}}$
3. < This (IT) BE This (IT) BEING >

Proof: by the previous demonstration.

Ostensive Information

One additional result of ostensive calculus is the development of a measure of ostensive information that was reported on last year. This has been applied to the problems of computer addressing, indexing, and data structuring. A paper reporting this application was presented to the 10th Annual Conference of the ACM Southeastern Region [2].

References

1. Gough, J., Jr., and DeMillo, R. "Towards an Ostensive Grammar." Paper presented at the Eighth Annual Meeting of the Association for Computational Linguistics, Ohio State University, Columbus, Ohio, July 22, 1970.

2. Pearson, Charls R. "Ostensive Information: A Measure of Information Associated With Computer Addressing, Indexing, and Data Structuring." Paper presented at the Southeastern Regional Meeting of the ACM, Mobile, Alabama, June 17-19, 1971.

3. Quine, W.V.O. *Word and Object.* Cambridge, M.I.T. Press, 1960.

THE INFORMATION AND COMPUTER SCIENCE LABORATORY

The past year's activity at the Information and Computer Science Laboratory has included experimental work in human speech synthesis and computer picture processing; the design of a compiler, an operating system, and an operating system simulator; the implementation of a system that allows the utilization of the Laboratory's PDP-8 as a terminal computer in conjunction with operations to a remote U-1108; and the production of various utility programs intended for general use at the laboratory. These projects are described in the capsule summaries which follow.

(1) Human Speech Synthesis. Digitized speech samples were compressed through a variety of Vocoder techniques via digital filtering on a large-scale computer, and the resulting compressed speech data was reprocessed to produce digitized speech at the input sampling rate. The new digitized speech was punched on paper tape and loaded onto the PDP-8 disk using a utility program; then, the samples were read into memory at a high rate (using double buffering techniques), the data was used to drive the D-A converter (on-line to the PDP-8), the output was put through a standard amplifier, and various selections of recognizable speech were produced. Thus, various digitizing techniques for speech compression and production could be readily evaluated by a "live" test.

(2) Compiler and Operating System. An operational ALGOL compiler was developed, and an interrupt-driven operating system was designed to execute compiled programs in a multiprogrammed environment. The compiler interprets ALGOL statements and generates absolute binary code sequences directly. The Operating System was modeled after the Burroughs B5500 MCP, utilizing code-disk swapping, dynamic memory allocation, and other high-level techniques usually found on operating systems of larger-scale computers. Such techniques permit a working system given the quite small (8K) core memory but rather adequate (512K) disk space available on the PDP-8.

(3) PDP-8 Used For Terminal Computing. A program was written to permit use of the PDP-8 as a terminal computer in conjunction with remote operations to the U-1108 under the EXEC8 Operating System (Demand mode). This program added considerable power when communicating with the larger computer as opposed to normal teletype remote capability; for instance, single-key commands could be initiated to send a string of control characters automatically, such as passwords, frequently used command strings, etc. Another feature permitted text to be composed in a disk file on the PDP-8 off-line from the larger computer, then transmitted automatically to U-1108 FASTRON Drum Storage automatically during an on-line session. A concurrently operating program on the U-1108 was required for this capability. Additionally, characters were echoed on the KV8I display scope and recorded on the Inktronic printer for faster, quieter operation. The program incorporated features to utilize the 300 baud capacity to the U-1108, permitting much faster operating characteristics. Work is in progress to utilize the 2400 baud synchronous line to the U-1108 at a future date.

(4) Operating System Simulator. An operating system simulator was written in 8K FORTRAN on the PDP-8. The program was written to accept parameter cards giving certain operating system features (time slice, core limitation, maximum number of jobs in mix, roundrobin philosophy vs. first-come-first serve philosophy, etc.), followed by up to fifty cards indicating the job stream information. The program would then operate upon the input data and produce an output listing indicating how well the particular operating system combination had carried out the job stream, and what the "throughput" values were. This general-purpose simulator enabled useful research into the theory of operating systems design by manipulation of key parameters.

The main-line portion of the program consisted of reading in the parameter values and establishing linkages to the portion of the program which implemented a particular algorithm corresponding to the six possible operating system philosophies available. Each of these program sections was stored as a subroutine which could be called into core memory from its disk-resident location by the relocatable loader.[1] The output listing to the line printer consisted of, in each case, beginning and terminating times and values for each of the jobs in the job stream, and--at the completion of all jobs--quantitative summaries of I/O and memory cycle percentages, total run times, multiprogramming factors, and "system efficiency." Optimally, a trace of each job's time allocations by the system was available.

(5) Computer Picture Processing. This project used the PDP-8 to experiment with computer picture processing capabilities. The scope display was utilized for visual inspection of the resulting constructed images. Input would be accepted in the form of a matrix of light intensity point values; a programmed edge-detector or evaluation algorithm would then be used in an attempt to reconstruct, as accurately as possible, the original picture. Other picture processing areas were investigated, such as a "growing" picture with life-like combinatorial properties.

(6) Utility Programs. A number of utility programs for general use have been produced, including: a test-grading program utilizing mark-sense answer cards and correct-answer and weighing cards as input, and producing analyses of the grades as well as a sorted listing of the grades; a FOCAL program used to plot any given function on the incremental plotter; a high-speed card-to-disk program; a scope-oriented directory listing program; and a Graf-Pen data entry program.

---

[1] The program performed scheduling and a simulated execution of each job based upon its processor/input-output and core size as well as the parameters supplied by the user (O/S philosophy, available memory, etc.). The goal was to maximize utilization of the system through a multiprogramming mixture of the jobs.

BIBLIOGRAPHY OF PUBLICATIONS 1970/1971*

Bauman, R. B. *A Personnel Management Information System*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report, GITIS-70-06. 36 p. (PB 195 136.)

Chiaraviglio, L. "Towards a Computer Model of the Evolution of DNA Sequences." In: *Progress of Cybernetics: Proceedings of the International Congress of Cybernetics, 1969* (ed., J. Rose), Vol. III, London, Gordon & Breach, 1970, pp. 1019-1034.

Chiaraviglio, L., and Gough, J., Jr. "Towards a Syntax-Monitored Semantic Pattern Recognition." In: *Progress of Cybernetics: Proceedings of the International Congress of Cybernetics, 1969* (ed., J. Rose), vol. III, London, Gordon & Breach, 1970, pp. 1267-1279.

DeMillo, R., and Chiaraviglio, L. *On the Applicative Nature of Assignment*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1971. Research Report, GITIS-71-01. 25 p. (PB 197 875.)

Dunham, K. B. "An Algorithm to Produce Complex Primes." *Communications of the ACM*, Algorithm #372, Vol. 13, No. 1, January 1970, p. 52ff.

Georgia Institute of Technology, Science Information Research Center. *Research 1969/1970: Annual Progress Report*. Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report, GITIS-70-09. 84 p.

Gough, J., Jr., and Chiaraviglio, L. "On the Base Referential Structure of the English Noun Phrase." *Foundations of Language* 6:447-462 (November 1970).

Hammett, J. C., Jr. *An Adaptive Spectrum Analysis Vocoder*. Atlanta, Ga., Georgia Institute of Technology, 1971. 142 p. (Ph.D. Thesis.)

Hoffman, J. M. *A Pre-scheduler and Management Model for a Class of Computer-User Systems*. Atlanta, Ga., Georgia Institute of Technology, 1971. 165 p. (Ph.D. Thesis.)

*Includes papers, theses, and reports published (or accepted for publication) since July 1970.

Kelly, M. D. "Edge Detection in Pictures by Computer Using Planning." In: *Machine Intelligence 6* (ed., Bernard Meltzer), Edinburgh University Press, 1971, pp. 397-409.

Kraus, D. H. "Bulgaria, Libraries in." In: *Encyclopedia of Library and Information Science* (ed., Allen Kent and Harold Lancour), New York, M. Dekker, 1970, pp. 471-484.

Kraus, D. H. *Scientific and Technical Documentation and Information in Romania, 1970.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1971. Research Report, GITIS-71-02. 75 p.

Poore, J. H., Jr. *Toward An Algebra of Computation.* Atlanta, Ga., Georgia Institute of Technology, 1970. 75 p. (Ph.D. Thesis.)

Roehrkasse, R. C., and Smith, W. D. *Simulation of Operating Systems.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report, GITIS-70-11. 24 p. (PB 195 138.)

School of Information and Computer Science. *Self-Instruction Systems: An Alternate Socio-Technological Approach to National Education and Training.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1971. Research Report and Prospectus, 13 p.

Siegmann, R. M. *Optimal Flow in Single Path Communication Networks* Atlanta, Ga., Georgia Institute of Technology, 1971. 143 p. (Ph.D. Thesis.)

Slamecka, V., and Jensen, A. P. *The Audiographic Learning Facility: Objectives and Design.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Research Report, GITIS-70-08. 44 p. (PB 195 137.)

Slamecka, V., and Zunde, P. "A University-Wide Program in the Information Sciences." *Proceedings of the 1970 IFIP World Conference on Computer Education, Vol. II.,* Amsterdam, Holland, August 1970, pp. 229-232.

Slamecka, V., Zunde, P., and Dexter, M. E. "Data Collection, Organization and Use." *Proceedings of the 1970 International Systems Meeting,* Las Vegas, Nevada, 1970, pp. 136-146.

Wright, D. *Computer Software Characteristics of the Audiographic Learning Facility.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1970. Internal Research Memorandum, GITIS-70-07. 20 p. (PB 195 157.)

Zunde, P. "Structural Models of Complex Information Sources." *Information Storage and Retrieval,* 7:1-18 (June 1971).

Zunde, P., and Dexter, M. E. "Indexing Consistency and Quality." In: *Introduction to Information Science,* (ed., Tefko Saracevic). New York, R.R. Bowker Co., 1970, pp. 563-572. (Reprinted.)

Zunde, P., and Slamecka, V. "Distribution of Indexing Terms for Maximum Efficiency." In: *Introduction to Information Science,* (ed., Tefko Saracevic). New York, R.R. Bowker Co., 1970, pp. 279-284. (Reprinted.)

Accepted for Publication

Gehl, J., and Slamecka, V. "A Profile of Scientific-Technical Tape Information Services." To appear in *Proceedings of the NBS/COSATI Forum on Information Analysis Centers,* National Bureau of Standards, Gaithersburg, Md., May 1971.

Hawkey, K. "A Critique of Certain Basic Notions in Chomsky's *Syntactic Structures.*" To appear in *Folia Linguistica.*

Pearson, C. R. (ed.). *Problems in Operating Systems Design.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1971. Research Report, GITIS-71-05. (To appear.)

Poore, J. H., Jr., Barralt-Torrijos, J., and Chiaraviglio, L. *On the Combinatory Definability of Hardware and Software.* Atlanta, Ga., Georgia Institute of Technology (School of Information and Computer Science), 1971. Research Report, GITIS-71-04. (To appear.)

Slamecka, V., Jensen, A. P., and Zunde, P. "An Audiographic Repository of Knowledge for Conversational Learning." To appear in *Educational Technology.*

Slamecka, V., and Zunde, P. "A Provisional Bibliography on Library Operations Research." To appear in *Library Quarterly.*

Valach, M. "A Q-graph Approach to Parsing." To appear in *Proceedings of the Conference on Linguistics,* University of Iowa, October 1970.

Zunde, P. "Synthesis of Interacting Dynamical Systems." To appear in *Proceedings of the IEEE International Conference on Systems, Networks and Computers,* Oaxtepec, Mexico, January 1971.

Zunde, P. "Semantic and Pragmatic Use of Syntactic Information Measures." To appear in: *Proceedings of the Meeting on Information Measures,* Kitchener-Waterloo, Ontario, Canada, 1970.

Zunde, P. "On Signs, Information and Information Measures." To appear in *Transactions of the Instrumentation Society of America*.

Zunde, P., and Slamecka, V. "Information Utilities." To appear in *Journal of the American Society for Information Science*.

Zunde, P., and Slamecka, V. "Predictive Models of Scientific Progress." To appear in *Information Storage and Retrieval*.