American Journal of Computational Linguistics Microfiche 55

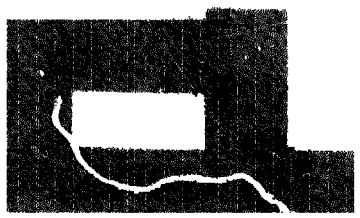
THE FINITE STRING

NEWSLETTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

VOLUME 13 - NUMBER 7 DECEMBER 1976

ACL New officers for 1977 2
Call for papers 3
Minutes, 1976 business meeting 4
Secretary-Treasurer's report 7
Financial report 9
Humanities - 3rd International Conference 10
Linguistic and Literary Analysis - 5th International . . . 11
Graphics and Interactive Techniques - 4th Annual . 12
Undergraduate Curricula and Computing Conference . . . 13
Representation and Understanding, edited by Daniel G.. Bobrow and Allan Collins. Reviewed by John Mylopoulos 14
The Role of Speech in Language, edited by James F Kavanagh and James E. Cutting. Reviewed by Sieb Nooteboom 26
Algebraic Parsing of Context-Free Languages Stephen F. Weiss and Donald F. Stanat 38
A Comparison of Term Value Measurements for Automatic Indexing - Gerard Salton 61
SNOPAR A Grammar Testing System - T P. Kehler 84

AMERICAN JOURNAL OF COMPUTATIONAL LINGUISTICS is published by the Association for Computational Linguistics. EDITOR: David G. Hays, 5048 Lake Shore Road, Hamburg, New York, 14075. EDITORIAL ASSISTANT: William Benson. SECRETARY-TREASURER: Donald E. Walker, Stanford Research Institute, Menlo Park, California 94025.

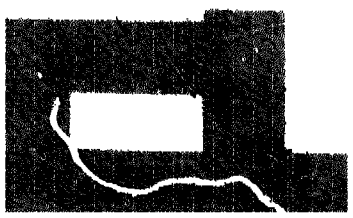


NEW OFFICERS FOR 1977

ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

President	PAUL CHAPIN National Science Foundation
Vice President	JONATHAN ALLEN Massachusetts Institute of Technology
Secretary-Treasurer	DONALD E. WALKER Stanford Research Institute
Executive Council	JERRY HOBBS City University of New York

Continuing members of the Executive Committee are Bonnie Nash-Webber (through 1977) and Timothy C Diller (through 1978). Continuing members of the Nominating Committee are William A. Woods, Jr. (1977) and Aravind K Joshi (1978) The Editor is a member of the Executive Committee ex officio



15TH ANNUAL MEETING

ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

PALMS LOUNGE, WALSH BUILDING

GEORGETOWN UNIVERSITY

WASHINGTON, D. C.

MARCH 16 - 17, 1977

CALL FOR PAPERS

1-page abstract, with title but no name
Letter with author's name and paper title

DEADLINE January 1, 1977
Members of ACL should have received prior notice of this deadline by letter.

ADDRESS Jonathan Allen
Room 36-575
Massachusetts Institute of Technology
Cambridge 02139

The Georgetown University Round Table (on Linguistics and Anthropology) will be held immediately following the ACL meeting.

ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

MINUTES: 11th Annual Business Meeting
8 October 1976
Cliff Hotel, San Francisco, California
President Joan Petrick pres d@ng

ANNOUNCEMEN

Petrick announced that Hood Roberts, Secretary-Treasurer of the ACL for the past five years has resigned from that post, coincident with his departure from the Center for Applied Linguistics to establish his own company. Don Walker has been appointed to the position on an interim basis for the remainder of the year. Petrick expressed the gratitude and appreciation of the Association for the dedication and service Roberts has provided during his tenure a sentiment strongly supported by the members present.

REPORTS FR@M THE SECRETARY-TREASURER

Walker read the Secretary-Treasurer's Report and the Financial Report, both of which had been prepared by Roberts. Copies are attached to these Minutes. Membership renewals, billing practices, and the financial status of the Association were discussed. Petrick announced that John Moyne, as Chairman of the Membership Committee, was preparing a campaign to recruit new members.

AMERICAN JOURNAL OF COMPUTATIONAL LINGUISTICS

Petrick reviewed the status of the AJCL in the absence of Dave Hays, its Editor. In discussing Hays' recent survey of the membership about the Journal, Petrick remarked on its quality and thoroughness, both in preparation and in the analysis of the results. Over 200 members responded, an unusually high percentage; they strongly supported continuing publication in microfiche form. There was also considerable interest expressed in having The Finite String available in hard copy and in making it possible to acquire full size copies of certain articles. Petrick announced an Executive Committee decision, contingent on adequate financial support, that at least part of the contents of the Finite String would be issued in hard copy form, particularly those items of key importance and timely interest to the membership. The cost of making hard copies of articles available would be determined, and members would be notified accordingly.

Petrick announced that the Executive Committee had voted to increase the Editorial Board of the AJCL from 14 to 15 members and to establish three year terms of office with five new members to be appointed each year on a regular basis.

DUES

Petrick announced that the increases in expenses associated with the AJCL and with the preparation and distribution of a hardcopy newsletter required raising the dues $5 to a new total of $15 for individual memberships. A suggestion was made from the floor that a class of family membership be established that would allow reduced dues for one of two spouses so that both could be members but only one copy of publications would be received.

THE MEL CUK CORRESPONDENCE

Petrick announced that the Executive Committee had decided to publish all the information that could be gathered about recent events associated with Igor Mel'cuk. Mel cuk was fired from his long term position as Senior Research Fellow of the Institute of Linguistics in the Academy of Sciences of the USSR, ostensibly on the basis of a letter he had submitted to the New York Times. The letter, which was published in January, expressed disagreement with the criticisms of Andrei Sakharov made by the Soviet press. In March, Mel cuk was fired; subsequently he prepared a letter describing the circumstances and asked that it be brought to the attention of American scientists. Questions had been raised about the appropriateness of publishing such correspondence on the grounds that it might hurt either Mel cuk or the ACL or both. An extensive discussion from the floor indicated that a variety of positions were taken on the issue. Petrick assured the members that the Soviet position would be represented to the extent that information about it was available.

NEXT ANNUAL MEETING

The 15th Annual Meeting of the ACL is being planned for Washington, D.C., in conjunction with the Georgetown University Round Table on Languages and Linguistics. Tentative dates are 15-16 March 1977.

REPORTS

Martin Kay reported briefly on COLING 76, the 6th International Conference on Computational Linguistics, which was held at the University of Ottawa, Ottawa, Canada, from 28 June to 2 July 1976. The next conference is scheduled for Varna, Bulgaria, in 1978.

Walker announced that the next International Joint Conference on Artificial Intelligence will be held 22-26 August 1977 at the Massachusetts Institute of Technology in Cambridge, Massachusetts.

Jane Robinson reported on Local Arrangements, with particular emphasis on the banquet scheduled for the evening, shortly after the conclusion of the Business Meeting.

Paul Chapin reported for the Program Committee. Of the 21 abstracts submitted, 14 were accepted; he expressed his appreciation to the Committee members for their assistance. His experience with publicity about the Call for Papers suggested that a check list be established to provide more effective notification.

Bob Barnes reported for the Nomination Committee that the following slate of officers had been proposed:

President: Paul Chapin, NSF
Vice President: Jonathan Allen, MIT
Secretary Treasurer: Don Walker, SRI
Executive Committee: Jerry Hobbs, CUNY
Nominating Committee: Stan Petrick, IBM

A motion that the slate be accepted unanimously was carried.

Bonnie Nash-Webber expressed the appropriate sentiments in the form of a Resolutions Committee Report.

NEW BUSINESS

The microfiche question was raised again, and Petrick reviewed the results of the questionnaire, the decision to provide newsletter information in hard copy form, and the provision of hard copies of selected articles at cost.

The meeting adjourned.

Donald E. Walker
Secretary-Treasurer, Pro-Tem

ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

Secretary-Treasurer's Report

Dear Colleagues:

I am always sorry on those rare occasions when I cannot attend the annual meeting of ACL; my Angst is even greater now that this meeting is my last one as your secretary-treasurer. My annual report to you typically consists of statements about membership and finances. This will be a typical report.

Membership:

When the new journal was first issued in 1974 there was a dramatic increase in the number of ACL members--from just under 400 in 1973, to over 800 by early 1975. Since then, these impressive gains have been so seriously eroded that our current membership stands at 580 (445 individuals and 135 institutions). A total of 212 individuals and 46 institutions who had paid for 1975 did not renew for 1976, although each week several renewals continue to dribble in. Several reasons might be thought of for the decline:

1. The heavy promotional activities at the beginning brought in some members who really weren't as interested in computational linguistics as they may have thought they were.

2. Some members do not like microfiches.

3. The recently established method of billing members for their annual dues (including the dues notice on one of the opaque cards in the journal) which was conceived as an economy measure clearly failed to produce results, and I would urge--pragmatically--that this method never be tried again.

It is hoped that the newly reactivated membership committee, under the chairmanship of John Moyne, will be able to devise creative answers to this chronic problem.

Finances:

In an organization such as ours, where the association is almost entirely dependent on the payment of annual dues, even a slight drop in membership causes serious problems. This years financial situation was further exacerbated by three additional things:

1. The continued, unrealistically low dues rate of $10, for which members are receiving nearly 2,000 microfiche pages yearly. (This problem was not unexpected and was discussed at the last Annual Meeting in Boston, and there were good reasons for leaving the dues at $10 until such time as the ACL decided what to do about the journal.)

2. Inflation

3. Unexpected charges -- primarily the $1,130.00 for refreshments (coffee and pastries) which were generously provided by the Sheraton Hotel at the last annual meeting. (I now believe that the Sheraton chain, indeed, is owned by ITT.)

The customary categorized financial statement is given below. Although the statement reflects ACL's income and expenses, some adjustments within these figures will be made later, pending a detailed allocation of the costs incurred in and income derived from the TINLAP volumes.

Respectfully submitted,

A. Hood Roberts

6 October 1976

ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

Financial Report for 1976

Balance as of October 30, 1975	$ 2,106.19
Receipts:	
Membership dues 1975-1976	14,355.27
ACL 76 meeting receipts to date	357.15
	$16,818.61
Disbursements:	
Administrative costs, office supplies, mailing, and AJCL costs not covered by CAL Account 317	$ 4,777.64
Membership ACAL	50.00
AFIPS dues 1976	500.00
Annual meeting costs 1975-1976	1,130.58
Paid out of ACL membership receipts into CAL Account 317 for AJCL, as required by NSF	9,039.60
	$15,497.82
Balance as of October 1, 1976	$ 1,320.79

THIRD INTERNATIONAL CONFERENCE ON

COMPUTING IN THE HUMANITIES

2 - 5 AUGUST 1977

WATERLOO, ONTARIO

SPONSORED BY THE UNIVERSITIES OF MONTREAL AND WATERLOO

THEMES Frontiers between language and literature, Fine arts; Graphics, Historical studies, Information retrieval; Input techniques, Lexicography, Literary stylistics; Medieval studies; Music; Photocomposition, Public service systems, Semantics

INTERNATIONAL COMMITTEE F. V. Spechtler, Austria; J. R. Allen, Canada, A. Jones, England, I T. Piirainen, Finland, L. Fossier, France; W. Lenders, Germany; M. L. Alinei, Holland, S. C. Loh, Hong Kong, F. Papp, Hungary, B. Jónsson, Iceland; S. K. Havanur, India; U. Oman, Israel, L. F. Lara, Mexico, K. Hyldgaard-Jensen, Sweden, J. Joyce, USA, J. Raben, USA.

REGISTRATION Professor J. S. North
Chairman, ICCH3
Department of English
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

American Journal of Computational Linguistics *Microfiche 55 : 11*

FIFTH INTERNATIONAL SYMPOSIUM ON THE USE OF COMPUTERS IN LINGUISTIC AND LITERARY ANALYSIS

UNIVERSITY OF ASTON IN BIRMINGHAM

3 - 7 APRIL 1978

Authorship studies
Concordances
Classical studies
Input-output
Oriental studies
Software
Stylistic analysis
Syntactic analysis
Text editing
Language-oriented groups
Education
Lexicography
Literary statistics

ADDRESS FOR CORRESPONDENCE

Professor D. E. Ager
CLLR
Department of Modern Languages
University of Aston in Birmingham
Gosta Green
Birmingham B4 7ET
England

FOURTH ANNUAL CONFERENCE

COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES

HYATT HOUSE, SAN JOSE, CALIFORNIA

JULY 20 - 22, 1977

CALL FOR PAPERS

TOPICS

Graphical theory and techniques such as languages, hardware, software, tools, portability, standards, device independence, line graphics, raster graphics, data structures, satellite systems, human factors, applications in the area of environmental, urban, transportation, cartography, biomedicine, animation, computer aided design, art, music, business, statistics, recreational graphics, decision making, and computer graphics education.

Papers may report original work, unusual or unique applications or techniques of computer graphics, or they may evaluate graphical specifics

DEADLINE

A short abstract is requested by December 1, 1976, and the final paper must be submitted by May 2, 1977

PROGRAM CHAIRMAN

James E George 415-447-1100 Ext 3360
Los Alamos Scientific Laboratory
P. O. Box 1663, MS 272
Los Alamos, New Mexico 87545

CALL FOR PAPERS: 1977 CONFERENCE ON COMPUTERS IN THE UNDERGRADUATE CURRICULA

MICHIGAN STATE UNIVERSITY, EAST LANSING

JUNE 19-22, 1977

SUBSTANCE

Reports of actual experience with computer use in a specific course or sequence of courses, in any field except computer science. No proposals; no repetition of previous reports without substantial new results. Survey papers only with synthesis or thorough evaluation.

FORMAT

Original manuscript suitable for reproduction in the proceedings. Typed, double spaced, up to 15 pages. 8"x10" pictorial matter, glossy B&W photographs or photographable drawings.

Title page. Authors' names, complete mailing address, telephone numbers, if multiple, indicate which handles correspondence and will deliver the talk. Each page should have the principal author's name on it.

DEADLINE - JANUARY 15, 1977

ADDRESS

Gerald L. Engel, Virginia Institute of Marine Science, Gloucester Point, Virginia 23062.

TRAVEL GRANTS

A limited number of partial travel and subsistence grants may be available to speakers and others from minority institutions and small golleges. Information and applications from CCUC/8 Travel Grant Committee, Eppley Center, MSU, East Lansing 48824

REPRESENTATION AND UNDERSTANDING

STUDIES IN COGNITIVE SCIENCE

EDITED BY DANIEL G. BOBROW AND ALLAN COLLINS
Xerox Palo Alto Research Center and Bolt Beranek and Newman

Academic Press, Inc.
New York

LC 75-21630 $15.00 ISBN 0-12-108550-3

REVIEWED BY JOHN MYLOPOULOS

Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 1A7

A major goal of Artificial Intelligence research today is to design systems that "understand" a body of knowledge, i.e. use it whenever appropriate. The representation of the knowledge available to such an "understander" system is an important issue for the system's design and is intimately related to the proposed uses of that knowledge. This book includes a collection of thirteen papers.written by some of the best known researchers who are currently working on understander systems. The papers were selected among those presented at a conference held in memory of Jaime Carbonell.

The contents of the book are as follows

I. Theory of Representation

1. Dimensions of Representation

Daniel G. Bobrow

2. What's in a Link: Foundations for Semantic Networks

William A. Woods

3. Reflections on the Formal Description of Behavior

Joseph D. Becker

4. Systematic Understanding:

Synthesis, Analysis, and Contingent Knowledge

in Specialized Understanding Systems

Robert J. Bobrow & John Seely Brown

II. New Memory Models

5. Some Principles of Memory Schemata

Daniel G. Bobrow & Donald A. Norman

6. A Frame for Frames:

Representing Knowledge for Recognition

Benjamin J. Kuipers

7. Frame Representations and

The Declarative-Procedural Controversy

Terry Winograd

III. Higher Level Structures

8. Notes on a Schema for Stories

David E. Rumelhart

9. The Structure of Episodes in Memory
Roger C. Schank

10. Concepts for Representing Mundane Reality in Plans
Robert P. Abelson

IV. Semantic Knowledge in Understander Systems

11. Multiple Representations of Knowledge
for Tutorial Reasoning
John Seely Brown & Richard R. Burton

12. The Role of Semantics
in Automatic Speech Understanding
Bonnie Nash-Webber

13. Reasoning From Incomplete Knowledge
Allan Collins, Eleanor H. Warnock,
Nelleke Aiello, & Mark L. Miller

As stated in the book's introduction, the section on "Theory of Representation" deals with general issues regarding the representation of knowledge, while that on "New Memory Models" discusses the implications of the assumption that input information is always interpreted in terms of large structural units derived from experience. The section titled "Higher Level Structures" focuses on the representation of plans, episodes and stories within memory. Finally, the section on "Semantic Knowledge in Understander Systems" describes on-going work of the SOPHIE, SPEECHLIS and SCHOLAR projects at BBN.

In attempting to review the papers that appear in this book collectively rather than individually, we arrived at a slightly

different taxonomy than that used by the book's editors. For the discussion that follows, numbers from 1 to 13 refer to the papers in the book. Other references, numbered from 14 on, are given at the end of the review.

1 . Comparison of Representations and General Criteria for their Adequacy.

Many different paradigms have been proposed for representations, including ones based on Predicate Calculus, Production Systems, Semantic Networks, frames", and ones that are PLANNER- or ACTOR-like. Winograd [14] provides an excellent comparison of those paradigms. An important and controversial aspect of current work on representation is the debate on the distinction between and desirability of declarative vs. procedural representations, and episodic vs. semantic memory organizations

A number of papers either provide criteria for comparison among different representations or general frameworks within which these representations can be described and discussed. Others present adequacy criteria for representations or discuss one or more of the controversies mentioned in the previous paragraph.

[1] proposes several "dimensions" along which representations can be compared and illustrates the use of those dimensions for the comparison of three very simple representations for digitized binary pictures. This paper also serves as an extended introduction to the rest of the book.

Becker describes in [3] how computer science concepts such as scheduling, backtracking, interrupts etc. can be used to model aspects of (human or machine) behaviour such as goals, conflicts, spheres of influence and decision making. Although the paper does make several interesting points, the lack of rigor hurts the discussion. For example, the last section of the paper presents an argument in favour of the view that behavioural descriptions are relative in the sense that behaviour admits many different, and possibly ambiguous, descriptions, unlike, say, a capacitor charging or discharging. But surely one could argue that the capacitor's behaviour could also admit different and ambiguous descriptions, such as "the capacitor is delaying a signal", "the capacitor is filtering out certain undesirable frequencies" etc. If one accepts this view, then there is no straightforward, absolute, canonical or true description for anything, nor just for behavioural systems. Perhaps the author is trying to establish a different point. If so, we missed it.

[4] presents the SCA model, which is intended to provide a framework for designing and comparing understander systems. The discussion gives accounts of two modules that are part of the model, the first to integrate incoming information to the system's knowledge base, the second to use the knowledge base in order to answer questions. Three existing systems are described within the framework of the SCA model as evidence of the model's adequacy. As admitted by the authors, however, the model is a very partial answer to an overall organization for

a system involving many processes. It should be added that with the discussion being so general and devoid of detail, it is hard to see whether a genuine contribution is being made or whether the model's apparent ability to fit different existing systems is precisely due to the lack of detail.

The paper by D Bobrow and Norman [5], proposes (memory) schemata as the constructs in terms of which the organization and operation of a memory can be described. The properties schemata should have are then discussed and many requirements are set forth for the adequacy of a representation Some of these are the use of context-dependent descriptions to access schemata, the accountability of all inputs, i.e. the ability of a memory system to account for all inputs, no matter how trivial, at some level, and the distinction between data-limited and resource limited processes. The overall framework that emerges is quite interesting because it takes into account issues regarding the design of large resource-limited systems that had only been studied in the past in Operating Systems literature.

The first part of Winograd's paper [7] deals with the declarative vs. procedural representation controversy and the trade-offs involved. The controversy is an old one within computer science and includes, among other things, the merits and demerits of a (declarative) representation that allows programs to be represented as data. The discussion in the paper is quite well-written and argues convincingly that the basic trade-off between the two different types of representations is one of modularity,

for declarative, vs. flexible interaction among different facts, for procedural.

Schank's paper [10] includes a discussion on whether the organization of human memory is episodic or semantic. An episodic memory organization implies that knowledge is stored as temporally dated episodes and events, with temporal spatial relations linking these events. A semantic memory organization, on the other hand, involves time-invariant knowledge a person possesses, e.g., "all elephants are animals". A corollary of these definitions is that an episodic memory organization favours temporal and causal connectives (e.g., THEN, REASON, ENABLE etc.), whereas a semantic memory organization uses extensively the "ISA hierarchy" (e.g., "an elephant is-a animal"). The discussion presented in the paper on this issue is somewhat confusing since at one point (pp. 255-256) the two types of organization are contrasted as if they were mutually exclusive, while later on (p. 263) the paper argues for a combination of the notions of semantic and episodic memory. In either case, Schank's work certainly makes a convincing argument in favor of an episodic memory organization by showing how it can be used to represent the meaning of a paragraph.

II . Critique and Extensions of Representation of Knowledge Paradigms

Several papers, including some that were mentioned in the previous section, criticize, refine, or extend one of the existing paradigms for the representation of knowledge.

The most notable example among those in this category is Woods' paper [2] which criticizes many (mis)uses of semantic networks by pointing out situations where their semantics are poorly defined or even inconsistent. Particular attention is paid to the representation of quantification and that of relative clauses.

As many of the readers undoubtedly know, Minsky's influential paper introducing "frames" [15] provides more of an ideology than a theory for representing knowledge. Kuipers in [6] argues in favor of a number of properties frames should have, such as the ability to describe an object or situation to varying degrees of detail, the ability to be instantiated and the ability to handle small perturbations of expected input data without major failures. He illustrates the desirability of these features with a simple example of object recognition.

The second half of Winograd's paper makes an attempt to synthesize declarative and procedural aspects of a representation. His proposal is based on frames and uses a generalization (ISA) hierarchy having a number of features, including the ability to associate procedures to objects on the hierarchy which specify how to perform different operations on those objects. Many of the ideas in [5] and [7] have been incorporated in KRL [16], as developed by D. Bobrow and Winograd.

III . Representing Different Kinds of Knowledge

Information entering an understander system may have many different "forms", i.e. it may be coded as photographs or line

drawings, simple sentences or paragraphs or even complete stories. Moreover, it may have different "content" i.e. involve a fairy tale world of kings and dragons, a blocks world of cubes and pyramids, a social, mental or physical world. One important aspect of the representation problem is the definition of a collection of knowledge, defined by a restriction on its form and/or content, and the investigation of the adequacy of a particular representation.

As mentioned earlier, Woods' paper does discuss the representation of quantification in terms of semantic networks, where the form of the knowledge involved is presumably (first order) Predicate Calculus and the content is unconstrained. It also discusses the representation of relative clauses and complex sentences where the form is natural language and the content is, again, unconstrained.

Rumelhart's paper [8] is primarily concerned with the discovery of structure underlying simple stories. The structure is defined in terms of a phrase structure grammar with semantic rules associated to each production. The paper certainly follows the general trend towards studying linguistic units larger than sentences, such as paragraphs, dialogues or stories. Whether the methodology used (in particular, phrase structure grammars) will be found adequate for the description of structure in stories remains to be seen.

Schank [9] deals mainly with the problem of constructing a structure of causally-linked actions and changes of states

(episodes) from a paragraph. When episodes are used to make sense of new inputs in often-experienced situations, they are called "scripts". The paper ends with a brief introduction of scripts. More details about them can be found in more recent publications by Schank and his students, e.g. [17,18].

Rumelhart's and Schank's work are related in that they both attempt to define the structure of a collection of knowledge limited with respect to form (stories for Rumelhart, paragraphs for Schank) and unconstrained with respect to content. Moreover, both papers agree that the underlying representation used must involve causally-linked events, and the causal connectives they employ are similar.

Abelson's paper is concerned with the representation of "mundane reality" involving social actions. The approach he follows is to postulate a number of primitive states and actions for achieving these states, in terms of which hopefully all simple social behaviour can be described. The discussion of the primitives is quite thorough, but the examples given do not provide sufficient evidence that the primitives proposed are in fact descriptively adequate. Abelson's work is complementary to Schank's in several respects and there is more recent joint work on the subject [19].

IV'. On-going Projects involving Understander Systems

The last three papers of the book discuss particular projects involving the design and implementation of understander systems.

[11] describes the scope, basic methodology, and achievements of SOPHIE, a knowledge-based computer aided instruction (CAI) system which attempts to teach a student about electronic circuits by asking questions, answering questions and letting him try out his ideas. Of particular interest to computational linguists should be the section describing the "semantic grammar" developed by Burton to handle the types of sentences expected during a dialogue on electronic circuits.

Nash-Webber [12] provides an overview of the BBN SPEECHLIS project in the context of a discussion on the use of semantic knowledge for speech understanding. Finally, [13] discusses some of the inference rules implemented or being considered for implementation by the SCHOLAR project whose aim is to develop a knowledge-based CAI system that teaches geography. The reader may find many of the rules stated in the paper completely reasonable and yet quite shaky from a logical point of view. For example, one rule (the uniqueness assumption) states that if only one thing is found, it can be assumed that it constitutes a complete set. Thus if someone knows of only one city called "Springfield" and located in Massachussetts, he can use the uniqueness assumption to reply "no" to "Is Springfield in Kentucky?" even though there may well be such a city.

The papers in this section constitute an important complement to the rest of the book which often involves discussions that are too far removed from the reality of an implemented (or implementable) system.

Overall, this book provides an excellent review of the state of the art, circa 1975, on the problem of representing knowledge.

•It should be apparent from the previous discussion that the book assumes a familiarity with basic issues of representation and understander system design. For more introductory discussions, the reader is referred to [14] or Schank and Colby [20].

References

14. Winograd, T. "Five Lectures on Artificial Intelligence" Stanford AI-Memo 246, September 1974.

15. Minsky, M. "A Framework for Representing Knowledge" in Winston P. (Ed.) The Psychology of Computer Vision, McGraw Hill, 1975.

16. Bobrow, D. and Winograd, T. "A KRL User's Manual" (unpublished).

17. Schank, R. "Using Knowledge to Understand" TINLAP Proceedings pp. 117-121, June 1975.

18. Schank, R. and the Yale AI Group "SAM ---a Story Understander" Yale University, Dept. of Computer Science, August 1975.

19. Schank, R. and Abelson, R. "Scripts, Plans and Knowledge" Proceedings IJCAI, pp. 151-157, September 1975.

20. Schank, R. and Colby, K. (Eds.) Computer Models of Thought and Language, Freeman, 1973.

American Journal of Computational Linguistics *Microfiche 55 : 26*

THE ROLE OF SPEECH IN LANGUAGE

EDITED BY JAMES F. KAVANAGH (GROWTH AND DEVELOPMENT BRANCH, NATIONAL INSTITUTE OF CHILD HEALTH AND HUMAN DEVELOPMENT) AND JAMES E. CUTTING (DEPARTMENT OF PSYCHOLOGY, WESLEYAN UNIVERSITY)

The MIT Press

Cambridge, Massachusetts 02139

1975

xiv + 335 pages $15.00 ISBN 0-262-11059-8

REVIEWED BY SIEB NOOTEBOOM

Instituut voor Perceptie Onderzoek

Postbus 513 den Dolech 2 Eindhoven 4502

The book under review contains the proceedings of a small conference (22 participants) with the same title, held in October 1973 at the Urban Life Center, Columbia, Maryland. The conference was one in a series called "Communicating by Language", sponsored by the National Institute of Child Health and Human Development (NICHD). There are 19 papers, divided into 3 major sections, viz.

I The development of speech in man and child

II Language without speech (dealing with sign language)

III Phonology and language

Some papers are followed by comments of one of the participants each paper or coherent group of papers is followed by a summary of the open discussion. A separate IVth section of the book contains reflections on the conference by Ira J. Hirsh. Refe-

rences are presented at the end of each paper. The editors have provided a name index and a subject index at the end of the book.

Many linguists and psycholinguists take it for granted that language can be studied without studying speech. Likewise many speech researchers seem to work from the view that the production and perception of speech can be studied without studying language. This situation leads Alvin Liberman to state in his "Introduction to the conference" that "our topic --the role of speech in language--is not an established one; no one has made it the direct and primary object of his research." Although this statement is perhaps too categorical, it certainly is valid for most of the field. (An obvious exception, to my mind, is among others Professor Lindblom of the University of Stockholm, who systematically explorés the explanatory value of quantitative models of speech production and perception in phonology, e.g. Lindblom 1972, 1975). The organizers of the conference, Kavanagh and Liberman, have taken care to select well-known researchers with different backgrounds and different interests to discuss the various problems which may be derived from the central question: "do we increase our understanding of language when we take into account that it is spoken?"

The resulting texts make interesting reading, although one will look in vain for a convincing answer to the initial question. Different investigators have different opinions and the present state of knowledge does not seem to make it

possible to settle the matter. In most papers specialist knowledge is freely intermixed with speculation, and it is not always easy to tell the one from the other. The discussions generally serve more to continue speculation than to criticize in detail each other's thinking. These remarks are not meant as a criticism of the conference and its proceedings. They intend to give an indication, however, of the style of this book, and a warning that one will not find here a thorough discussion of empirical data or explicit, testable theories, that could be of use in more practically oriented work. Instead one finds a number of inspiring expositions of such diverse topics as similarities and dissimilarities between human and animal communication systems, the evolutionary connections between language, speech, and tool-making, the primacy of production or perception in the phylogenesis and the ontogenesis of speech, the primacy of signs or speech in the evolution of language, the articulate structure of signs in those who have sign language as their first language, the origins of phonological change, and the parallels in phonological and other linguistic organization of language.

Below I will make a few remarks on a few selected topics:

a) The evolution of speech and language
b) Spoken language and sign language
c) Innate feature detectors
d) The absence of prosody

I will not attempt to cover in this review all papers in the book.

A. THE EVOLUTION OF SPEECH AND LANGUAGE

In a number of places in this volume attempts are made to relate results of recent empirical studies of several kinds to theoretical ideas on the evolution of speech and language in early man. So Peter Marler gives an interesting description of communication systems in nonhuman primates and birds. His data on monkeys show a difference between discrete signal systems, consisting of a limited number of acoustically well-distinguished sound signals, used by monkeys living in dense forests and having little visual contact, and graded signal systems displaying continuous variation of sound signals, used by terrestrial monkeys. The bird data on the white-crowned sparrow lead him to the concept of an innate auditory template for bird song, modifiable by a suitable external model and serving for the development of vocal behavior. In his speculations on the origin of speech Marler emphasizes the importnace of the evolution of innate but modifiable auditory templates for speech sounds, serving to distinguish between acceptable and nonacceptable models for vocal development, for classifying acceptable sounds into subcategories and for developing speech. He also assumes that, while categorical processing was developed as an aid in identifying sounds from memory, continuous sensory processing of sounds was retained, thus leading to an intermingling of categorical and noncategorical (discrete and graded) processing. He finally suggests that "The substitution of categorical for continuous processing

of speech sounds may have directly facilitated the introduction of syntax as a radical innovation in primate communication".

There appear to be two basic assumptions underlying Marler's reasoning. One is that comparative studies of sensory and vocal behavior in animals and man may lead to interesting theories about specific properties of the human brain underlying man's capacity for speech and language. The other is that such studies may clarify the order in which postulated changes in vocal perception and development might have occurred in the evolution of early man. There is an important difference between these two assumptions. Whereas the former may lead to theories or hypotheses which in principle might become testable, the latter does not, at least not within the limits of this reviewer's imagination. Obviously this lack of testability is common to many speculations about the evolution of human behavior. This has in the past not kept scientists from making reasonable guesses particularly about the evolution of language and speech, and probably will not do so in the future In this volume both Hewes in his comments on Mattingly's paper and Liberman in his own contribution relate the genesis of language to toolmaking. Hewes observes similarities between syntactic structures and the prescribed order of the various steps necessary for the manufacture of flakes from a prepared Levallois core. Liberman, taking the same line of thought, states that the Levallois toolmaking technique cannot reasonably be described by means of a phrase-structure grammar. A

transformational grammar which formally incorporates a memory is necessary. As far as I understand his reasoning this is so because in making a particular chip one has to keep two things in mind, both the last chip that has been made and the final form of the tool. It seems to me, however, that in order to give his argument its force it still has to be shown that there is a fundamental difference in the necessary complexity of underlying mental structures between Levallois toolmaking and many forms of goal-oriented behavior we find in higher animals.

Liberman also suggests that the final crucial stage in the evolution of human language would appear to be the development of the bent two-tube supralaryngeal vocal tract of modern man, which allows its possessors to generate acoustic signals that (1) have very distinct acoustic properties and (2) are easy to produce, being acoustically stable. Reconstructions from fossils tell him that the Neanderthal hominids had to do without this asset, and therefore probably retained a communication system with a mixed phonetic level that relied on both gestural and vocal components. At this point the reader particularly feels the need for an expert criticism of the validity of such reconstructions.

B. SPOKEN LANGUAGE AND SIGN LANGUAGE

The question whether speech or gestural communication has been more important in the evolution of human language came up several times during the conference. In reaction to Mattingly's

idea that "speech exemplifies a thoroughly and peculiarly human kind of knowing" Hewes commented that the depigmentation of the volar skin would indicate the antiquity of nonvocal communication. Indirect support for this supposed antiquity of gestural communication comes from some fascinating studies of American Sign Language (ASL), according to Bellugi and Klima a full-fledged language of its own, and not a derivative or degenerate form of written or spoken English. Stokoe argues for the antiquity of sign language from a possible parallel between ontogeny and phylogeny. It appears to be the case that the infant with deaf parents, learning ASL as its first language, begins putting wordlike signs into sentencelike structures at an earlier age than the child making two-word or three-word sentences in speech.

Bellugi and Klima have studied sign language from historical changes in the form of signs, in short term memory experiments, by analyzing a collection of "slips of the hand", and by comparing American Sign Language with Chinese Signs, in all cases with profoundly deaf people who use sign language as their primary form of communication. They show that signs in ASL are not simply signals which differ uniquely and holistically from one another but are, rather, highly coded units. They also provide evidence that grammatical processes bear the marks of the particular transmission system in which the language developed. This seems to be confirmed in Huttenlocher's

contribution, comparing the encoding of spatial relations in ASL and natural language (= spoken American English)

It is too early to draw any definite conclusions from these studies of sign language on the interdependence of natural language and speech, as the structure of sign language is only beginning to be understood. But it is certainly of much interest to students of language behavior that the human perceptual and cognitive systems appear to be so flexible that profoundly deaf people may develop visual communication systems among themselves which, if not equal in expressive power and speed of communication to natural spoken languages, at least come close to them. Further comparisons between the syntax of natural spoken languages and sign languages may lead to more caution in interpreting current ideas about what is and what is not innate in our linguistic abilities. Similarly comparisons between the efficiency of speech perception and the efficiency of visual sign perception might well make us wonder whether speech perception is as special as some theorists like to make us believe.

C. INNATE FEATURE DETECTORS

The idea that speech perception is mediated by, possibly innate, speech specific feature detectors was given considerable attention in the conference. This idea supported Marler's extrapolation from innate auditory templates in birds to innate auditory templates in humans. Studdert-Kennedy provides a

careful survey of the current empirical evidence concerning the perceptual processing of consonants and vowels, from which he concludes that the "human cortex is supplied with sets of acoustic detectors tuned to speech, each inhibited from output to the phonetic system in the absence of collateral response in other detectors".

Cutting and Eimas present evidence that such feature detectors are innate. Eimas has shown that very young infants, one month and four months of age, can discriminate much better between different speech sounds that belong to different phonemic categories than between different speech sounds belonging to the same phonemic category in adult speech. One may concur, however, with the doubt expressed by Hirsh in his reflections on the conference whether Eimas's data are about speech or about general auditory perception. One may feel similar doubts about the interpretation Eimas and Cutting give to the data stemming from the selective adaptation paradigm, introduced in speech perception studies by Eimas and Corbit in 1973 and since then used by an increasing number of investigators. In selective adaptation studies it is shown that repeated stimulation with a particular acoustic configuration, for instance a syllable ba, may change the response distribution in a phoneme identification task, for instance the binary forced choice between ba and pa measured with stimuli taken from the acoustic continuum between ba and pa. In this case the number of pa-responses would increase at the cost of the ba-responses. The

interpretation is that there are feature detectors which can be fatigued by repeated stimulation. By carefully studying which acoustic configurations lead to shifts in particular response distributions, it would be possible to find out what information is extracted by particular feature detectors. Cutting and Eimas argue for the existence of phonetic, speech specific, feature detectors. More recent studies show that categorical perception and selective adaptation are not unique to speech perception (Cutting, Rosner and Foard 1976). Furthermore, to my knowledge, nobody has yet seriously discussed the difficulties for a theory of "wired-in" feature detectors stemming from perceptual normalization experiments in which it is shown that response distributions in phoneme identification tasks may shift systematically due to the immediate environment of the test segment (e.g. Fourcin 1972).

D. THE ABSENCE OF PROSODY

The volume under review is not only remarkable for the many interesting and stimulating papers it contains but also for what it does not contain. In a collection of papers with the title "The role of speech in language" one would have expected to find at least one contribution seriously discussing the relation between speech prosody and linguistic structure. It is ironical that the only paper in which intonational contrast is given moré attention than obligatory lip service is Stokoe's contribution "The shape of soundless language", dealing with

sign language Stokoe's treatment of intonation and its kinesic correlate in sign language seems to make explicit why so many speech researchers do not pay attention to speech prosody. He suggests that intonational contrasts "are not necessarily linguistic and have more affinity with other systems that signal affect than with phonemic contrasts. There remain then only phonemic contrasts between consonant and consonant, vowel and vowel, and tone and tone (when so used) as the indisputably linguistic, basic features of language". One may fear that this undue overemphasis on phonemic contrast in speech perception research will persist until speech scientists turn away from the study of isolated CV-syllables and start wondering about the perception of normal spontaneous connected speech.

REFERENCES

Cutting, J. E., Rosner, B. S., Foard, C. F. (1976) Perceptual categories for musiclike sounds: implications for theories of speech perception. *Quarterly Journal of Experimental Psychology*, 28:361-378.

Fourcin; A. J. (1972) Perceptual mechanisms at the first level of speech processing. In: A. Rigault and R. Charbonneau, eds. *Proceedings of the VIIth International Congress of Phonetic Sciences, Montreal 1971*. Mouton, The Hague.

Lindblom, B. E. F. (1972) Phonetics and the description of language. In: A. Rigault and R. Charbonneau, eds. *Proceedings of the VIIth International Congress of Phonetic Sciences, Montreal, 1971*.

Mouton, The Hague.

Lindblom, B. E. F. (1975) Experiments in sound structure. Plenary paper, presented at the VIIIth International Congress of Phonetic Sciences, Leeds 1975.

American Journal of Computational Linguistics *Microfiche 55 : 38*

Algebraic Parsing of Context-Free Languages

Stephen F. Weiss and Donald F. Stanat

Department of Computer Science
University of North Carolina
New West Hall 035A
Chapel Hill 27514

Abstract

A class of algebraic parsing techniques for context-free languages is presented. A grammar is used to characterize a parsing homomorphism which maps terminal strings to a polynomial semiring. The image of a string under an appropriate homomorphism contains terms which specify all derivations of the string. The work describes a spectrum of parsing techniques for each context-free grammar, ranging from a form of bottom-up to top-down procedures.

ALGEBRAIC PARSING OF CONTEXT-FREE LANGUAGES

I. Introduction

For many years syntactic analysis and the theory of formal languages have developed in a parallel, but not closely related, fashion. The work described here is an effort to relate these areas by applying the tools of formal power series to the problem of parsing.

This paper presents an algebraic technique for parsing a broad class of context-free grammars. By parsing we mean the process of determining whether a string of terminal symbols, χ, is a member of the language generated by grammar G (i.e., is $\chi \in L(G)$?) and, if it is, finding all derivations of χ from the starting symbol of G. We hope that posing the parsing problem in purely algebraic terms will provide a basis for examination and comparison of parsing algorithms and grammar classes.

Section II presents an overview of the algebraic parsing process. It provides a general notion of how the method works without going into detail. Section III contains the algebraic preliminaries and notational conventions needed in order to describe the parsing method precisely. The formal presentation of the parsing method and the proof of correctness form Section IV. Section V contains some interesting special cases of the theorem and presents some examples of parses.

II. Overview of the algebraic parsing process

The algebraic parsing formalism described here is applicable to all context-free grammars $G = \langle V_N, V_T, P, S \rangle$ except those that contain productions of the form $A \rightarrow B$ where A and B are both nonterminals, or erasing rules such as $A \rightarrow \epsilon$. The parsing process consists first of constructing (on the basis of the grammar G) a polynomial and a function defined on polynomials. A parse of χ is obtained by repeated applications of the function to a polynomial $P(\chi)$. The process has two features worthy of note. First, it produces all parses of χ in parallel. Second, the process of converting a grammar into the required algebraic form is straightforward and does not alter the structure of the grammar. This property, the preservation of grammatical structure, is particularly important in areas such as natural language analysis where the structure that a grammar provides is as important as the language it generates.

The polynomials we will use have terms of the form (Z,Δ), where Z is a string over an extended alphabet and Δ represents a sequence of productions of G. The process begins with a polynomial of ordered pairs representing χ, the string to be parsed. A function is repeatedly applied to the polynomial; the number of applications necessary is bounded by the input length. If the resulting polynomial contains a term (S,Δ) where S is the starting symbol in G, then Δ represents the production sequence used in generating χ from S. If no such pair occurs, then χ is not in L(G), and if multiple pairs

occur (S,Δ_1), (S,Δ_2), ... then χ is ambiguous and the Δ's specify the several parses. A precise formulation of the polynomial and the operations on it is given below.

III. Algebraic preliminaries and notation

A semigroup is formally defined as an ordered pair $\langle S,\cdot\rangle$ where S is a set (the carrier) and $\cdot$ is an associative binary operation. Similarly, a monoid is a triple consisting of a set, an operation and a two-sided identity (e.g., $\langle S,\cdot,1\rangle$). We will feel free to denote a monoid or semigroup by its carrier.

For any set V, V^* denotes the free monoid generated by V; $V^* = \langle V^*,\text{concatenation},\Lambda\rangle$. Similarly, V^+ denotes the free semigroup generated by V; $V^+ = \langle V^+,\text{concatenation}\rangle$. We denote the length of a string χ in V^* or V^+, by $|\chi|$.

For an arbitrary alphabet V, we define $\bar{V} = \{\bar{v}\,|\,v\in V\}$. The free half-group generated by V, H(V), is defined to be the monoid generated by $V \cup \bar{V}$ together with the relation $aa = 1$, where 1 is the monoid identity and a is any element of V. Note that in H(V) the elements of $\bar{V}$ are left inverses but not right inverses of the corresponding elements of V. We denote the extended alphabet $V \cup \bar{V}$ by Σ.

If $T = \langle T,\cdot,1\rangle$ and $Q = \langle Q,+,0\rangle$ are monoids, we denote by $T \times Q$ the product monoid $\langle T \times Q,\otimes,(1,0)\rangle$. The carrier of $T \times Q$ is the cartesian product $T \times Q$ and the operation $\otimes$ is defined to be the component-wise operation of T and Q:

$$(a,b) \otimes (c,d) = (a\cdot c,b+d).$$

A <u>semiring</u> is an algebraic system $\langle S,+,\cdot,0\rangle$ such that

$\langle S,+,0\rangle$ is a commutative monoid,

$\langle S,\cdot\rangle$ is a semigroup,

and the operation $\cdot$ distributes over +:

$$a\cdot(b+c) = a\cdot b + a\cdot c,$$

$$(a+b)\cdot c = a\cdot c + b\cdot c.$$

A <u>semiring</u> is commutative if the operation is commutative. A <u>semiring with identity</u> is a system $\langle S,+,\cdot,0,1\rangle$ where $\langle S,+,\cdot,0\rangle$ is a monoid. The semirings used in this paper are commutative and have identities. Furthermore, in each case the additive identity is a multiplicative zero:

$$0\cdot x = x\cdot 0 = 0.$$

The <u>boolean semiring B</u> consists of the carrier $\{0,1\}$ under the commutative operations + and $\cdot$, where $1\cdot 1 = 1+x = 1$ and $0+0 = 0\cdot x = 0$ for all $x \in \{0,1\}$.

For an arbitrary monoid M we denote by R(M) the semiring of polynomials described as follows:

1) Each <u>term</u> is of the form $c\alpha$ where $c \in B$ (the boolean semiring of coefficients) and $\alpha \in M$.

2) Each <u>polynomial</u> is a formula sum (under +) of a finite number of terms.

3) Addition and multiplication of terms is defined as follows:

   a) $b\alpha + c\alpha = (b + c)\,\alpha$

   b) $(b\alpha)(c\beta) = (bc)(\alpha\beta)$.

4) Addition and multiplication of polynomials is performed in the usual manner consistent with 3).

Note that all coefficients of R(M) are either 1 or 0. We will adopt the usual convention of not explicitly writing 1 for the terms with that coefficient and omitting terms with a coefficient of 0.

A context-free grammar is a system $G = \langle V_N, V_T, P, S \rangle$ where V_N and V_T are finite, disjoint, non empty sets denoted non-terminal and terminal symbols respectively. We denote by V the set $V_N \cup V_T$. The symbol S is the distinguished nonterminal from which all derivations begin, and P is the set of productions of G. A context-free grammer is proper if it does not contain productions of the form $A \rightarrow \epsilon$ (erasures) or $A \rightarrow B$ where A and B are both nonterminals.

It can easily be shown that the set of languages generated by proper context-free grammars is exactly the set of context-free languages. In addition, an arbitrary context-free grammar can be made proper by a straightforward method which alters the structure of the grammar very little. In this study we will deal with only proper context-free grammars. This guarantees that all terminal strings have a finite number of derivations in G, and thus makes possible our goal of finding all derivations of an input.

Productions of G will be indexed by integers. Thus $A \xrightarrow{i} M$ denotes that $A \rightarrow M$ is the i^{th} production in P. We will deal only with left-most derivations. A leftmost derivation is completely specified by the initial sentential form and the sequence of production indices. If $\Delta \in I^*$ is the sequence of production indices in the leftmost derivation of $N \in V^+$ from $M \in V^+$, we write $M \overset{\Delta}{\Longrightarrow} N$. The length of a derivation Δ is denoted by $|\Delta|$, and is equal to the number of production indices in Δ.

We will use, but not formally define, the notion of height of a

derivation', meaning the height of the corresponding derivation tree or the length of the longest path from the root to the frontier of the tree. The height of a derivation Δ will be denoted by $h(\Delta)$.

Since 'derivation' will always mean 'leftmost derivation' in the sequel, the following assertions hold:

Assertion 1: A derivation is of height 0 if and only if it is of length 0. A derivation is of height 1 if and only if it is of length 1.

Assertion 2: Let G be a proper context-free grammar, and

$$A \overset{\Delta}{\Longrightarrow} M$$

where $|\Delta|>0$. Then Δ is of height less than or equal to $|M|$.

Assertion 3: Let $G = \langle V_N, V_T, P, S\rangle$ be a context-free grammar, I an index set for P, and let the j^{th} production of G be

$$A \overset{j}{\rightarrow} a_1 a_2 \cdots a_m \qquad a_i \in V.$$

Let $j\Gamma$ be a derivation

$$A \overset{j\Gamma}{\Longrightarrow} M \qquad M \in V^+$$

of height $n + 1$. Then

$$\Gamma = \Delta_1 \Delta_2 \cdots \Delta_m \qquad \Delta_i \in I^*$$

and

$$M = M_1 M_2 \cdots M_m \qquad M_i \in V^+$$

and for all i, $1 \leq i \leq m$,

$$a_i \overset{\Delta_i}{\Longrightarrow} M_i$$

is a derivation of height n or less.

The algebraic structure used in this work is the semiring of polynomials $R(H \cdot I^*)$ where $H = H(V)$, the free half-group generated by V, and I is the index set of the set of productions P. We will use an initial segment of the natural numbers, {1,2,3,...}, as the index set I. Each term of a polynomial from $R(H \cdot I^*)$ consists of an element from $H \cdot I^*$ together with a coefficient from the boolean semiring B. The elements of $H \cdot I^*$ will be the basis for calculating the parses of a string λ. The elements of H will interact to determine if a product of terms characterizes a derivation. If so, the associated element of I^* is the sequence of production indices of the derivation.

The following notational conventions will be observed.

$G = \langle V_N, V_T, P, S \rangle$

$V = V_N \cup V_T$

$\Sigma = V \quad \bar{V}$

$Z \in \Sigma^+$

$z \in \Sigma$

$i, j, k \quad m, n \in \underline{N}$ (set of natural numbers).

$I \subseteq \underline{N}$

$\Delta, \Gamma, \theta \in I^*$

$\chi \in V_T^+$

$a, b, c \in V$

$A, B, C \quad V_N$

$M, N, P, Q \in V^*$

δ, g, Ψ, ν will denote functions. For the function g,

$$g^1(x) = g(x) \text{ and } g^k(x) = g(g^{k-1}(x)).$$

IV. An algebraic parsing theorem

Theorem (version 1): Let $G = \langle V_N, V_T, S, P\rangle$ be a proper context-free grammar. Then there exist homomorphisms ν, g, and δ,

$$\nu: V^* \to R(V \times I^*)^*$$

$$g: R(\Sigma \times I^*)^* \to R(\Sigma \times I^*)^*$$

$$\delta: R(\Sigma \times I^*)^* \to R(H \times I^*)$$

and a special polynomial $p \in R(\Sigma \times I^*)^*$ such that for every $\chi \in V_T^+$, $\chi = \chi_1 \cdots \chi_n$, $\chi_i \in V_T$,

$$\delta g^n\left[\prod_{i=1}^{n} p^n\ \nu(\chi_i)\right]$$

contains a term (S,Δ) if and only if Δ is a leftmost derivation of χ from S.

Construction for the proof:

Let $V = V_1 \cup V_2$ be an arbitrary exhaustive division of V:

$$V_1 \cup V_2 = V.$$

The construction is most economical when V_1 and V_2 are disjoint, but this is not required.

a.. $\nu: V^* \to R(V \times I^*)^*$

The function ν is the homomorphism induced by the following:

$\nu(a) = (a,\Lambda)$, $a \in V$ and Λ is the identity in I^*.

Since ν is a homomorphism, $\nu(\Lambda) = \Lambda$.

b. $g: R(\Sigma \times I^*)^* \rightarrow R(\Sigma \times I^*)^*$

The function g is the homomorphism induced by defining g on the generators of the domain as follows:

1. $g(\bar{a},\Delta) = (\bar{a},\Delta)$; $\bar{a} \in \bar{V}$, $\Delta \in I^*$

2i. $g(a,\Delta)$ contains the term (a,Δ); $a \in V$

2ii. If $A \rightarrow ab_1 \ldots b_n$ is the i^{th} production of P and $a \in V_1$ then $g(a,\Delta)$ contains $(A,i\Delta)(\bar{b}_n,\Lambda) \ldots (\bar{b}_1,\Lambda)$.

2iii. There are no other terms in $g(a,\Delta)$.

Note that because g is a homomorphism, $g(\Lambda) = \Lambda$, where Λ is the identity of the monoid $(\Sigma \times I^*)^*$

c. $\delta: R(\Sigma \times I^*)^* \rightarrow R(H \times I^*)$

The function δ is the canonical homomorphism which coalesces a product in $(\Sigma \times I^*)^*$ into a single ordered pair by component-wise multiplication of the first entries (thus allowing cancellation in H) and catenation of the second entries. For example,

$$\delta[(a,\Delta_1)(\bar{b},\Delta_2)(b,\Delta_3)(c,\Delta_4)] = (ac,\Delta_1\Delta_2\Delta_3\Delta_4).$$

d. The polynomial p is an element of $R(\Sigma \times I^*)^*$ defined as follows:

1. p contains the summand Λ;

2. If $a \in V_2$ and $A \rightarrow ab_1 \ldots b_n$ is the j^{th} production of P then p contains the summand $(A,j)(\bar{b}_n,\Lambda) \ldots (\bar{b}_1,\Lambda)(\bar{a},\Lambda)$.

3. p contains no other summands.

We adopt the convention that $p^k = \Lambda$ for $k \leq 0$.

Note that since p contains Λ, p^k contains Λ as well as all summands of p^j for $j \leq k$.

For notational convenience we adopt the following conventions. First, where no ambiguity can result, products in $R(\Sigma \times I^*)^*$ of the form

$$(z_1,\Delta_1)(z_2,\Delta_2) \ldots (z_n\Delta_n) \qquad z_i \in \Sigma,\ \Delta_i \in I^*$$

will be abbreviated as:

$$(z_1z_2 \ldots z_n,\ \Delta_1\Delta_2 \ldots \Delta_n).$$

No cancellation is implied by this notation since cancellation cannot occur in $R(\Sigma \times I^*)^*$. Second, we define the function Ψ_k as follows:

$$\Psi_k\colon V^* \to R(\Sigma \times I^*)^*$$

$$\Psi_k(a_1a_2 \ldots a_n) = \prod_{i=1}^{n} p^k\, \nu(a_i)$$

where $a_i \in V$ and p is the polynomial defined above. Note that, if $k \leq 0$, then $\Psi_k(a_1a_2 \ldots a_n) = \nu(a_1a_2 \ldots a_n)$, and $\Psi_k(\Lambda) = \Lambda$. Using this notation, we can re-state the theorem as follows:

Theorem (version 2): Let $G = \langle V_N, V_T, P, S\rangle$ be a proper context-free grammar. Then there exist maps Ψ, g and δ such that

$$\Psi\colon V^* \to R(\Sigma \times I^*)^*$$

$$g\colon R(\Sigma \times \perp^*)^* \to R(\Sigma \times I^*)^*$$

$$\delta\colon R(\Sigma \times I^*)^* \to R(H \times I^*)$$

such that for every $\chi \in V_T^+$, $\chi = \chi_1\chi_2 \cdots \chi_n$, $\chi_i \in V_T$, $\delta g^n \Psi_n(\chi)$ contains a term (S,Λ) if and only if $S \overset{\Delta}{\Longrightarrow} \chi$.

The proof of the theorem rests on three lemmas. Lemma I implies the "if" part of the theorem; Lemma III implies the "only if" part. Lemma II is used in the proof of Lemma III.

Lemma I: Let $M \in V^+$, $A \in V$, and $A \overset{\Delta}{\Longrightarrow} M$. Then for all $k > h(\Delta)$, $\delta g^k \Psi_k(m)$ contains (A,Δ).

Proof (by induction on $h(\Lambda)$, the height of the derivation Δ):

Basis: If $h(\Delta) = 0$, then $\Delta = \Lambda$ and $M = A$. Then $\Psi_k(A) = p^k(A,\Lambda)$. Since Λ is a summand of p, it follows that (A,Λ) is a summand of $p^k(A,\Lambda)$, and therefore (A,Λ) is a summand of $\delta g^k \Psi_k(A,\Lambda)$. Thus the derivation $A \overset{\Lambda}{\Longrightarrow} A$ is represented in $\delta g^k \Psi_k(A)$ by (A,Λ), which establishes the basis.

Induction: Let Δ be a derivation of height $n + 1$, $A \overset{\Delta}{\Longrightarrow} M$. By assertion 3,

$$\Delta = j\Gamma_1\Gamma_2 \cdots \Gamma_r$$

$$M = M_1M_2 \cdots M_r$$

where

$$A \overset{j}{\rightarrow} a_1a_2 \cdots a_r$$

and

$$a_i \overset{\Gamma_i}{\Longrightarrow} M_i$$

where $h(\Gamma_i)$ n.

Since $\delta g^{k}\psi_{k}(M) = \delta g^{k}\psi_{k}(M_1)\delta g^{k}\psi_{k}(M_2) \ldots \delta g^{k}\psi_{k}(M_r)$, if $k \geq n$ then by the induction hypothesis, $\delta g^{k}\psi_{k}(M_j)$ contains the summand (a_j,Γ_j). Consider the term of $g^{k}\psi_{k}(M_1)$ which cancels to (a_1,Γ_1) in $R(H \times I^*)$. This term must be of the form $(a_1,\Gamma_1')T$, where Γ_1' is a prefix of Γ_1. Either $a_1 \in V_1$ or $a_1 \in V_2$. The sum $\delta g^{k+1}\psi_{k+1}(M_1)$ contains $\delta g g^{k}\psi_{k}(M_1)$, which contains $\delta g(a_1,\Gamma_1')T$. If $a_1 \in V_1$, then $g(a_1,\Gamma_1')$ contains $(\overline{Aa_2a_3\ldots a_r}, j\Gamma_1')$, and $\delta g(a_1,\Gamma_1')T$ contains $(\overline{Aa_2a_3\ldots a_r}, j\Gamma_1)$. On the other hand, the sum $\delta g^{k+1}\psi_{k+1}(M_1)$ also contains $\delta p g^{k}\psi_{k}(M_1)$. If $a_i \in V_2$, then $(\overline{Aa_1a_2\ldots a_r}, j)$ is a summand of p, and therefore $\delta p(a_1\Gamma_1')T$ contains $(\overline{Aa_2a_3\ldots a_r}, j\Gamma_1)$. Thus in either case, $\delta g^{k+1}\psi_{k+1}(M_1)$ contains the summand $(\overline{Aa_2a_3\ldots a_r}, j\Gamma_1)$ and since every summand of $\delta g^{k}\psi_{k}(M_j)$ is a summand of $\delta g^{k+1}\psi_{k+1}(M_i)$, it follows that $\delta g^{k+1}\psi_{k+1}(M)$ contains

$$(\overline{Aa_2a_3\ldots a_r}, j\Gamma_1)(a_2,\Gamma_2)(a_3,\Gamma_3)\ldots(a_r,\Gamma_r)$$

$$= (A, j\Gamma_1\Gamma_2\ldots\Gamma_r) = (A,\Delta).$$

This completes the proof.

<u>Lemma II</u>: Let $a \in V$, $\Gamma \in I^*$. For $k \geq 0$, all terms of $g^{k}(a,\Gamma)$ are of the form $(b,\Delta\Gamma)(\bar{c}_m,\Lambda)\ldots(\bar{c}_1,\Lambda)$ where $b \in V$, $\bar{c}_i \in \bar{V}$, $m \geq 0$, $\Delta \in I^*$ and $b \overset{\Delta}{\Longrightarrow} ac_1\ldots c_m$.

For notational convenience we abbreviate $c_1\ldots c_m$ by N. Hence we denote $(b,\Delta\Gamma)(\bar{c}_m,\Lambda)\ldots(\bar{c}_1,\Lambda)$ by $(bN,\Delta\Gamma)$.

Proof by induction on k, the number of applications of g. By definition, $g^{0}(a,\Gamma) = (a,\Gamma)$ which establishes the assertion for the

value k = 0.

Assume the assertion holds for k < n and consider $g^{n+1}(a,\Gamma) = gg^{n}(a,\Gamma)$. By the induction hypothesis, all terms of $g^{n}(a,\Gamma)$ are of the form $(b\bar{N},\theta\Gamma)$ where $b \overset{\theta}{\Longrightarrow} aN$. Hence terms of $g^{n+1}(a,\Gamma)$ are of the form $g(b\bar{N},\theta\Gamma)$. Since g limited to $\bar{V}$ is the identity, $g(b\bar{N},\theta\Gamma) = [g(b,\theta\Gamma)](\bar{N},\Lambda)$. By definition of g, $g(b,\theta\Gamma)$ contains only terms of the form $(C\bar{M},j\theta\Gamma)$ where $C \overset{j}{\rightarrow} bM$ is a production. Therefore terms of $g^{n+1}(a,\Gamma)$ are of the form

$$(C\bar{M},j\theta\Gamma)(\bar{N},\Lambda) = (C\overline{\bar{M}\bar{N}},j\theta\Gamma)$$

and since $C \overset{j}{\rightarrow} bM$ and $b \overset{\theta}{\Longrightarrow} aN$ it follows that $C \overset{j\theta}{\Longrightarrow} aNM$.

Corollary: All terms of $g^{k}(a\bar{M},\Gamma)$ are of the form $(b\overline{NM},\Delta\Gamma)$.

Lemma III: If $\delta g^{k}\Psi_{k}(M)$ contains $(A\bar{N},\Delta)$, then $A \overset{\Delta}{\Longrightarrow} MN$.

Proof by induction on the length of M:

Basis: Let $a \in V$ and assume

$$\delta g^{k}\Psi_{k}(a) \text{ contains } (A\bar{N},\Delta).$$

If p_i represents an arbitrary summand of p other than Λ, then every term of $g^{k}\Psi_{k}(a)$ can be represented in the form

$$\prod_{i=1}^{n} g^{k}(p_i)\ g^{k}(a,\Lambda)$$

where $0 \leq n \leq k$ and n denotes the number of nontrivial summands of p which are factors of the term.

By construction, every summand of p is either $\Lambda\cdot$ or of the form

$$(B_i\bar{P}_i, j_i) \text{ where } B_i \in V_N,\ P_i \in V^+,\ j_i \in I$$

and $B_i \overset{j_i}{\rightarrow} P_i$ is a production in G.

By Lemma. II. every term of $g^k(B_i\bar{P}_i, j_i)$ is of the form:

$$(C_i\bar{M}_i\bar{P}_i, \Gamma_i j_i) \text{ where } C_i \in V_N,\ M_i,\ P_i \in V^*,\ \Gamma_i \in I^*$$

and $C_i \overset{\Gamma_i}{\Longrightarrow} B_iM_i$;

By the same lemma, it follows that every term of $g^k(a,\Lambda)$ is of the form

$$(C_{n+1}\bar{M}_{n+1}, \Gamma_{n+1}) \text{ where } C_{n+1} \in V,\ M_{n+1} \in V^*,\ \Gamma_{n+1} \in I^*$$

and $C_{n+1} \overset{\Gamma_{n+1}}{\Longrightarrow} M_{n+1}$.

Hence every term of $g^k\Psi_k(a)$ is of the form

$$\left[\prod_{i=1}^{n} (C_i\bar{M}_i\bar{P}_i, \Gamma_i j_i) \;\middle|\; (C_{n+1}M_{n+1}, \Gamma_{n+1}) \quad 0 < n \leq k \right]$$

where $C_i \overset{\Gamma_i j_i}{\Longrightarrow} P_iM_i$ for $1 \leq i \leq n$ and $C_{n+1} \overset{I_{n+1}}{\Longrightarrow} M_{n+1}$ (1)

By assumption there is a term t of $g^k\Psi_k(a)$ such that $\delta[t] = (\Lambda\bar{N}, \Lambda)$; t must be in the form indicated above. In order for t to cancel under δ, the following must be true:

$C_1 = A$ since C_1 cannot cancel from t,

$\bar{P}_i = \bar{Q}_i\bar{C}_{i+1}$ for $1 < i \leq n$ since $C_2 \ldots C_{n+1}$ must all cancel from t.

Therefore

$$t = \left[\prod_{i=1}^{n} (C_i \bar{M}_i \bar{Q}_i \bar{C}_{i+1}, \Gamma_i j_i) \right] (C_{n+1} \bar{M}_{n+1}, \Gamma_{n+1}).$$

This cancels to $(A\bar{N}, \Delta)$ as required with

$$A = C_1$$

$$N = M_{n+1} Q_n M_n Q_{n-1} M_{n-1} \cdots Q_1 M_1$$

$$\Delta = \Gamma_1 j_1 \Gamma_2 j_2 \cdots \Gamma_n j_n \Gamma_{n+1}.$$

Then by (1),

$$C_i \overset{\Gamma_i j_i}{\Longrightarrow} C_{i+1} Q_i M_i, \quad 1 \leq i \leq n, \text{ and}$$

$$C_{n+1} \overset{\Gamma_{n+1}}{\Longrightarrow} M_{n+1}$$

Hence, since $C_1 = A$,

$$A \overset{\Gamma_1 j_1 \Gamma_2 j_2 \cdots \Gamma_n j_n \Gamma_{n+1}}{\Longrightarrow} M_{n+1} Q_n M_n Q_{n-1} M_{n-1} \cdots Q_1 M_1$$

and thus

$$A \overset{\Delta}{\Longrightarrow} N.$$

This establishes the basis.

Induction: Assume that for all $M \in V^*$ such that $|M| \leq n$, if $\delta g^k \Psi_k(M)$ contains $(A\bar{N}, \Delta)$ then $A \overset{\Delta}{\Longrightarrow} MN$. Let $\hat{M} = Ma$ be a string such that $|Ma| = n+1$ and $\delta g^k \Psi_k(Ma)$ contains $(A\bar{N}, \Delta)$. Because δ g and Ψ are homomorphisms,

$$\delta g^k \Psi_k(Ma) = [\delta g^k \Psi_k(M)][\delta g^k \Psi_k(a)].$$

Then $\delta g^k \psi_k(M)$ must contain a term (T_1, Δ_1) and $\delta g^k \psi_k(a)$ must contain a term (T_2, Δ_2) such that $T_1 T_2 = A\bar{N}$ and $\Delta = \Delta_1 \Delta_2$.

In order for this to occur, T_2 must be of the form $(B\bar{N}_2)$ where $B \in V$, $N_2 \in V^*$, and T_1 just be of the form $(A\bar{N}_1\bar{B})$ where $A \in V$, $N_1 \in V^*$, and $\bar{N} = \bar{N}_1\bar{N}_2$. (If T_1 and T_2 were not of this form, cancellation to $A\bar{N}$ would be impossible.) Thus $\delta g^k \psi_k(M)$ contains $(A\overline{N_1 B}, \Delta_1)$, and by the induction hypothesis

$$A \overset{\Delta_1}{\Longrightarrow} MBN_1.$$

Also $\delta g^k \psi_k(a)$ contains $(B\bar{N}_2, \Delta_2)$ and by the basis

$$B \overset{\Delta_2}{\Longrightarrow} aN_2.$$

It follows that

$$A \overset{\Delta_1\Delta_2}{\Longrightarrow} MaN_2N_1$$

and since $\hat{M} = Ma$ and $N = N_2N_1$,

$$A \overset{\Delta}{\Longrightarrow} \hat{M}N$$

which completes the proof.

The theorem now follows from Lemmas I and III and Assertion 2. The 'if' part follows from Lemma I and Assertion 2, and the 'only if' part follows immediately from Lemma III for the special case of $N = \Lambda$.

As we have stated the theorem, the length of χ is used to determine a sufficient number of applications of g and ψ. Alternatively, the theorem could be formulated in terms of the heights of derivations

of χ; if Δ is a derivation of χ of height k, then for every $n \geq k$, the term (S,Δ) will be in the polynomial $\delta g^n \Psi_n(\chi)$. Furthermore, it follows from Lemma III that no harm is done by choosing the value of n too large, i.e., no 'false' derivation terms will occur.

In the first statement of the theorem, the derivation terms are obtained from the polynomial $\delta g^n \prod_{i=1}^{n} p^n \nu(\chi_i)$ which can be re-written in the form

$$\delta \prod_{i=1}^{n} g^n [p^n \nu(\chi_i)]$$

Although we have used a constant value of n (equal to the length of χ) for both the powers of the map g and the polynomial p, some economy can be gained in this respect. In fact, the powers of g and p can decrease from left to right so long as they remain large enough to perform the appropriate computations on the suffix strings of χ. Thus, the theorem is true (but considerably more difficult to prove) if one instead uses a parsing polynomial of the form

$$\delta \prod_{i=1}^{n} g^{n-i+1} [p^{n-i+1} \nu(\chi_i)].$$

V. Special cases of the theorem

A number of interesting special cases occur based on the choice of V_1 and V_2.

Case 1. $V_1 = V_T$.

$V_2 = V_N$.

The function g handles all productions of the form

$$A \rightarrow \alpha M \qquad \alpha \in V_T, \; M \in V^*,$$

while p handles productions of the form

$$A \to BM \qquad B \in V_N,\ M \in V^*.$$

Notice that since g is nontrivial on only V_T, g need be used only once; i.e.,

$$g^k(\alpha,\Gamma) = g(\alpha,\Gamma) \qquad k > 1.$$

The parsing polynomial is then

$$\delta\{g[\Psi_k(\chi)]\}.$$

The special case of $V_1 = V_T$ and $V_2 = V_N$ results in a particularly simple form if the grammar is in Greibach no mal form. The polynomial $p = (\Lambda,\Lambda)$ and therefore has no effect. Since g need only be applied once, all derivations are found in one step.

Example 1:

$$G = \langle\{S,A,B\},\ \{a,b\},\ S,\ P\rangle$$

$P =$ 1. $S \to a\Lambda$

2. $A \to AB$

3. $A \to A$ $\qquad V_1 = \{a,b\}$

4. $B \to b$ $\qquad V_2 = \{S,A,B\}$

$$p = (\Lambda,\Lambda) + (A,2)(\bar{B},\Lambda)(\bar{A},\Lambda)$$

$$g(a,\Lambda) = (a,\Lambda) + (S,1)(\bar{A},\Lambda) + (A,3)$$

$$g(b,\Lambda) = (b,\Lambda) + (B,4).$$

For the string χ = aabb, the parsing polynomial $g[\Psi_k(\chi)]$ then contains (among other things) for all $k \geq 2$,

$$g(a,\Lambda)\ p^2\ g(a,\Lambda)\ g(b,\Lambda)\ g(b,\Lambda).$$

This contains:

$$[(S,1)(\bar{A},\Lambda)][(A,2)(\bar{B},\Lambda)(\bar{A},\Lambda)(A,2)(\bar{B},\Lambda)(\bar{A},\Lambda)][(A,3)][(B,4)][(B,4)].$$

Applying δ we get

$$(S,122344).$$

Case 2. $V_1 = V.$

$V_2 = \phi.$

The entire job of parsing is now done by g, since the polynomial p is equal to (Λ,Λ). Hence the parsing polynomial is

$$\delta[g^k(\chi,\Lambda)].$$

Example 2: We use the same grammar and input string as above.

$V_1 = \{S, A, B, a, b\}.$

$V_2 = \phi.$

$g(S,\Lambda) = (S,\Lambda)$

$g(A,\Lambda) = (A,\Lambda) + (A,2)(\bar{B},\Lambda)$

$g(B,\Lambda) = (B,\Lambda)$

$g(a,\Lambda) = (a,\Lambda) + (S,1)(\bar{A},\Lambda) + (A,3)$

$g(b,\Lambda) = (A,\Lambda) + (B,4).$

The parsing polynomial for aabb is

$$g^k(a,\Lambda)\ g^k(a,\Lambda)\ g^k(b,\Lambda)\ g^k(b,\Lambda).$$

For $k \geq 3$, this contains

$$[g^1(a,\Lambda)][g^3(a,\Lambda)][g^1(b,\Lambda)][g^1(b,\Lambda)]$$

which in turn contains

$[(S,1)(\bar{A},\Lambda)][g^2(A,3)][(B,4)][(B,4)]$ after one application of g,

$[(S,1)(\bar{A},\Lambda)][g^1(A,23)(\bar{B},\Lambda)][(B,4)][(B,4)]$ after two; and

$[(S,1)(\bar{A},\Lambda)][(A,223)(\bar{B},\Lambda)(\bar{B},\Lambda)][(B,4)][(B,4)]$ after three.

Applying δ results in (S,122344) as before.

Case 3. $V_1 = \phi$.

$V_2 = V$.

Now the entire parse is handled by p. The parsing polynomial becomes

$$\delta[\Psi_k(\chi)].$$

VI. Observations

The major theorem presented here shows how context-free parsing may be carried out by purely algebraic means. All parses of an input string are developed in parallel and the process is guaranteed to terminate. As we have described the process, the number of terms of a parsing polynomial for a string $\chi \in V_T^+$ is unreasonably large. However, most of the terms in such a polynomial are not associated with a derivation in the grammar, and methods exist for reducing the computation by disregarding dead-end terms before they are completely evaluated. By applying such techniques in a straightforward fashion, and choosing V_1 and V_2 in various ways,

the algebraic method can be associated in natural ways with classical parsing techniques. For example, the algebraic process in case 1 above is a goal directed top-down approach similar to the predictive analyzer. Case 2 is the algebraic version of generalized bottom-up.

Parsing algorithms are typically so different one from another that they are incomparable. But using techniques described above, many parsing algorithms may be posed in a single algebraic framework. This may facilitate the comparison and evaluation of parsers and of various classes of grammars.

REFERENCES

Chomsky, N. and M. Schutzenberger (1963), The Algebraic Theory of Context-Free Languages, in "Computer Programming and Formal Systems". (P. Braffort and D. Hirschbert, Eds.), North Holland, Amsterdam.

Ginsburg, S. and H. G. Rice (1963), Two Families of Languages Related to ALGOL, JACM 9, pp. 350-371.

Shamir, Eliahu (1967), A Representation Theorem for Algebraic and Context-Free Power Series in Non-Commuting Variables, Information and Control 11, pp. 239-254

Stanat, D. F. (1972), Approximation of Weighted Type 0 Languages by Formal Power Series, Information and Control 21, pp 344-381.

Stanat, D. F. (1972), A Homomorphism Theorem for Weighted Context-Free Grammars, J. Comput. System Sci. 6, pp. 217-232

Weiss, S. F., D. F. Stanat and G. A. Mago (1973), Algebraic Parsing Techniques for Context-Free Grammars, in "Automata, Languages and Programming" (M. Nivot, Ed.), pp. 493-498, North Holland/American Elsevier.

American Journal of Computational Linguistics *Microfiche 55 : 61*

A Comparison of Term Value Measurements for Automatic Indexing

Gerard Salton

Department of Computer Science
Cornell University
Ithaca, New York 14853

This work was supported in part by the National Science Foundation under grant GJ 43505.

Abstract

A number of statistical theories have been proposed capable of identifying individual text words that are most useful for the content representation of written texts and documents. Among these are parameters based on the variance of the word-frequency distribution (NOCC/EK), and on information theoretical (signal-noise S/N) premises. These formal parameters are related to practical automatic indexing techniques--most notably to the discrimination value (DV) method, capable of generating content identifiers (individual words, phrases, and word classes) that distinguish the various texts and documents from each other. It is shown that terms with favorable formal parameters also exhibit desirable semantic characteristics in that such terms are concentrated in documents judged relevant by the respective user populations, and vice-versa for terms with unfavorable formal properties.

1. Theories of Term Importance

Automatic indexing may be considered to be a two-step process: first the automatic identification of linguistic entities useful for the representation of document content, and then the assignment to the prospective content identifiers of weights reflecting their importance for content description. Since these tasks must ultimately depend on a study of the texts or documents under consideration, a great deal can be learned by examining

the occurrence patterns of words and other linguistic entities in the documents of a collection. Indeed, among the theories of term importance which have been studied in recent years, the best known ones are based on the respective frequency distributions across a variety of written texts.

A) Variance-Based Measures

The most widely used of the statistical theories distinguishes so-called "specialty" words from "nonspecialty" words by assuming that a deviation from randomness in the occurrence pattern of certain text words is indicative of specialization and hence of good content identifiers. Thus the best content descriptors are terms whose occurrence patterns deviate most strongly from randomness. Since a random sprinkling of the occurrences of a given text word across the documents of a collection leads to word frequency distributions which follow the Poisson model, a comparison of the actual frequency characteristics of a given term with the Poisson distribution leads to the appropriate distinction between good content words and poor ones.

More specifically, since the variance V^k of the frequency distribution of term k is proportional to the total frequency of occurrence F^k for terms whose distribution obeys the Poisson model, a measure of term importance is obtainable by using a formula based on the ratio of V^k to F^k. Some typical formulas used for this purpose are V^k/F^k and $n^2 \cdot V^k/F^k$ where n is the collection size. [1,2,3] The basic mathematical formulations are collected in Table 1.

Formulas	Explanation
n	number of documents in collection
f_i^k	frequency of term k in document i
b_i^k ($b_i^k = 1$ when $f_i^k \geq 1$; $b_i^k = 0$ when $f_i^k = 0$)	binary frequency of term k in document i
$F^k = \sum_{i=1}^{n} f_i^k$	total frequency of term k in collection
$B^k = \sum_{i=1}^{n} b_i^k$	document frequency of term k in collection (number of documents in which the term occurs)
$\bar{f}^k = \frac{F^k}{n}$	average frequency of term k in collection
$v^k = \frac{1}{n} \sum_{i=1}^{n} (f_i^k - \bar{f}^k)^2$ $= \frac{1}{n} \sum_{i=1}^{n} (f_i^k)^2 - (\frac{F^k}{n})^2$	variance of frequency distribution

Basic Frequency Formulas

Table 1

One such variance-based measure used by Dennis under the name of NOCC/EK [3] may be computed as

$$\text{NOCC/EK} = \frac{n}{F^k} \sum_{i=1}^{n} (f_i^k)^2 - F^k. \qquad (1)$$

It is obvious from this formulation that the most effective terms are those whose occurrence frequencies f_i^k in the individual documents deviate strongly from the average frequency F^k/n.

B) Signal-Noise Measure

Another measure based on the characteristics of the frequency distribution of individual text units across the documents of a collection is the signal-noise ratio which varies with the skewness of the frequency distribution. This measure has the form of entropy and assigns the highest value to those terms whose occurrence characteristics exhibit the greatest variation from one document to another; contrariwise low values are assigned to terms with relatively similar frequency patterns in each of the documents of a collection. [3,4] The idea is that terms with even frequency distributions which may occur an identical number of times in each document of the collection cannot be used to distinguish the documents from each other; hence, their assignment for purposes of content representation is counter-productive. The reverse obtains for terms with skewed frequency distributions.

The signal noise value $(S/N)^k$ for term k is defined as

$$(S/N)^k = \log F^k - \sum_{i=1}^{n} \frac{f_i^k}{F^k} \log \frac{F^k}{f_i^k} \qquad (2)$$

The negative term in expression (2) is known as the noise N^k; it is maximized for even distributions where $f_i^k = F^k/n$ for all f_i^k. The properties of the signal-noise measure are thus very similar to those described earlier for the variance-based formulas.

C) Information Theoretic Considerations

The foregoing development leads to a distinction among the terms in accordance with the relative sizes of the individual term frequencies f_i^k in the documents and the total collection frequency F^k. A question arises about the preferred size of the collection frequency F^k (or of the document frequency B^k) for terms that are useful as content identifiers. This problem may be tackled by having recourse to certain information-theoretic concepts. Consider the task of supplementing a set of existing index terms identifying a collection of documents by addition of a certain number of new terms. Each new term is then most effective when

a) it provides maximum additional reduction in uncertainty among the documents of the collection (that is, its assignment breaks up existing subsets of documents that cannot be distinguished by the existing term assignments into substantially smaller subsets);

b) it exhibits little redundancy with the previously available terms so that its assignment does indeed optimally divide the various document sets.

The first property is obviously not fulfilled for terms with low document frequency B^k, that is, those assigned to very few documents in the collection, because their assignment provides little additional discrimination among the documents; the second property, on the other hand, does not obtain for terms of high document frequency that may be assigned to a very large number of documents, because such terms will obviously exhibit a good deal of redundancy with the already existing terms.

The conclusion is that the best terms are those whose document frequency B^k, or total frequency F^k, is neither too large nor too small, and whose frequency distribution is skewed in that for some documents, f_i^k is much larger than $\frac{F^k}{n}$, and for some others f_i^k is much smaller than $\frac{F^k}{n}$.

D) The Discrimination Value Model

The discrimination value model uses as a point of departure the retrieval capability of the various index terms; specifically, a good content-indicative term is designed to help in the retrieval of material that is wanted (thus enhancing the recall), and in the rejection of material that is extraneous (thus enhancing the precision)*. To produce high recall, that is to retrieve most everything that is relevant, the terms used to identify documents and user queries must be fairly general in nature; high precision, on the other hand, that is the rejection of the nonrelevant material, depends on the use of reasonably specific content identifiers. The indexing problem then reduces to the choice of terms that are specific enough to produce high precision while also being general enough to produce high recall.

In the discrimination value model, the assumption is made that the best terms in this respect are those which cause the maximum possible separation among the documents in the "document space". Consider, in particular, a collection of documents each identified by a set of content identifiers, or index terms. The index term sets for two given documents can be compared to produce a similarity coefficient measuring the closeness between the respective documents.

---

* Recall is the proportion of relevant material retrieved while precision is the proportion of retrieved material that is relevant. An effective retrieval system is one which produces the highest possible precision for a given level of recall.

The existence of the term sets representing the various documents, and the possibility of computing similarity measures between documents can be used to define a document space for the collection. In such a space two documents appear in close proximity when their similarity coefficient is large; contrariwise, documents exhibiting little similarity are widely separated in the document space. One may then conjecture that a document space which is "bunched up", in the sense that all documents exhibit somewhat similar term sets is not useful for retrieval, since one document cannot then be distinguished from another. On the contrary, a space which is spread out in such a way that the documents are widely separated from each other may provide an ideal retrieval situation since some documents may then be retrieved — hopefully the relevant ones — while others can be rejected.

This suggests that the value of an index term can be ascertained by measuring the amount of spreading in the document space which occurs when that term is assigned to the documents of the collection. Specifically, if Q is the density of the document space without term k present among the content indicators, and Q_k is the density after term k is assigned, then for a good term $Q - Q_k > 0$, since the space will have spread after term k is assigned. Conversely for poor terms $Q - Q_k < 0$.* [5,6] An appropriate

---

* The density of the space might be computed, for example, as the sum of all pairwise similarities between distinct document pairs, that is

$$Q = \sum_{i \neq j} S(D_i, D_j). \qquad \begin{matrix} 1 \leq i \leq n \\ 2 \leq j \leq n \end{matrix}$$

where $S(D_i, D_j)$, $0 \leq S \leq 1$, is the similarity between documents D_i and D_j.

measure of term importance is then the <u>term discrimination</u> value, DV_k, defined as

$$DV_k = Q - Q_k. \qquad (3)$$

It may be of interest to inquire into the relationship between the discrimination value of a term and the statistical (frequency) parameters introduced earlier. The following conclusions are reached from a study of the indexing vocabularies in several different subject areas, relating the document frequency of a term to its discrimination value: [5]

a) terms with very low document frequency that may be assigned to very few documents in a collection are generally poor discriminators; when the terms are arranged in decreasing order of their discrimination values (where rank 1 is assigned to the best discriminator, rank 2 to the next best, and so on) such terms exhibit ranks in excess of $t/2$ for a total of t existing terms;

b) terms with high document frequencies, comprising those that are assigned to more than 10 percent of the documents of a collection are the worst discriminators, with average discrimination ranks (ranks in decreasing discrimination value order) near t;

c) the best discriminators are those whose document frequency is neither too high nor too low — with document frequencies between $n/100$ and $n/10$ for n documents; their average discrimination ranks are generally below $t/5$ for t terms.

The vector space analysis then appears to confirm the conclusions derived earlier from the statistical models, that terms which appear in a collection with great rarity or excessive frequency are not optimal for content description purposes.

2. Comparison and Evaluation

The discrimination value analysis can be used to derive an effective indexing policy: since the best terms appear to be those with medium document frequencies, such terms can be directly assigned as content identifiers without further refining transformations. On the other hand, terms with excessively high document frequencies must be made more specific thereby decreasing the frequency of their assignment to the queries and documents of the collection; contrariwise, terms with low document frequencies must be made more general by increasing their assignment frequencies. [5] This can be achieved by joining two or more high frequency terms into term phrases, while assembling a number of low frequency terms into term classes. Obviously, a term phrase exhibits a lower assignment frequency than any phrase component, and vice-versa for a term class which replaces a number of individual class elements.

It was shown earlier that the use of phrases and term classes (thesaurus) constructed in accordance with the frequency requirements imposed by the discrimination value theory produces substantial improvements in retrieval effectiveness (recall and precision). In the present work, additional relationships are examined between the statistical and the vector space models. However, instead of actually using the various term sets in a retrieval environment, an attempt is made to relate the formal frequency and vector space properties of the terms to the semantic characteristics of these terms.

Specifically, consider a collection of documents in a given subject area and an appropriate set of user queries pertaining to that area. For each user query, the set of documents can be partitioned into two subsets consisting of the

relevant set R and the nonrelevant set I, respectively. Relevance is assumed to be user-specified in such a way that a relevant item is assumed to be one which is related in some sense to the information need expressed by the various user queries. The linguistic, or semantic, character of a given term can now be introduced by assuming that the most valuable content identifiers assigned to a collection of texts are those which are concentrated in the documents specified as relevant to the respective queries, as opposed to the nonrelevant ones, contrariwise, the less valuable terms will be concentrated in the nonrelevant items.

The discussion may be formalized by using the concept of <u>term relevance</u> TR. [7] Consider a term k contained in query Q; the term relevance TR(k) may be defined as

$$TR(k) = \frac{r_k}{|R|-r_k} \Big/ \frac{h_k}{|I|-h_k}, \qquad (4)$$

where r_k and h_k are the number of documents containing term k that are relevant and nonrelevant respectively to query Q, and $|R|$ and $|I|$ are the total number of relevant and nonrelevant documents for that query.* When a term k occurs in more than one query, its term relevance may be taken as the average of the relevance values obtained for the various queries.

---

* The mathematically undesirable situation when $|R| = r_k$ or when $h_k = 0$ is not likely to occur in a practical environment.

It is clear from the function (4) that high values are assigned to those query terms which are prevalent in the relevant items and rare in the nonrelevant, and vice-versa for those prevalent mainly in the nonrelevant. Furthermore, the terms falling into the former class are likely to be more useful for content representation than those in the latter.

To verify the relationships between the statistical models of word importance and the vector space model, document collections are used in three different subject areas, including aerodynamics (CRAN), medicine (MED) and world affairs (TIME). The vocabularies and user populations are disjoint for these three areas. Results which carry through for all three cases should be extendable to other subject fields as well. The basic collection statistics are contained in Table 2.

It may be seen from the Table that the term relevance is defined for only a relatively small number of terms for each collection, namely 458, 172 and 375 for CRAN, MED, and TIME, respectively. The reason is that a term relevance value is computable only for terms which occur jointly in certain query-document pairs. For small experimental collections operating with a restricted number of queries the size of the corresponding term sets is obviously limited.

Consider now the comparison of the standard statistical term value measures with the term discrimination values obtained by the vector space transformations. Table 3 shows the values of the NOCC/EK and S/N measures (expressions (1) and (2)) obtained for the 50 terms with highest discrimination values and the 50 terms with lowest discrimination values for each of the three test collections. The range of the respective values is given in each case, as well as the average values for each set of 50 terms in percent (that is, on

Characteristics:	CRAN 424	MED 450	TIME 425
Subject area	aerodynamics	medicine	world affairs
Number of documents	424	450	425
Number of user queries	155	24	83
Number of terms assigned to collection	2651	4726	7569
Number of terms occurring jointly in queries and document sets	458	172	375

Basic Collection Statistics

Table 2

a scale of 0 to 100). T test values are also shown representing the probability that the two sets of 50 values (for the high DV and low DV terms) could have been derived from a common probability distribution by chance. In statistical significance testing, a t-test value smaller than 0.05 is normally taken to imply a significant difference; that is, the hypothesis that the two sets of values do in fact originate from a common distribution is rejected in such a case. [8]

It may be seen that the ranges of values for the statistical parameters NOCC/EK and S/N exhibit substantial differences for all three collections. The same is true for the corresponding average values. Moreover the differences are in all cases statistically significant. It is then clear that a high discrimination value reflected in the ability of a term to expand the document space upon assignment to the collection also implies favorable statistical parameters in terms of variance and skewed frequency distributions; the converse is true for the low discrimination values.

At the bottom of Table 3, range and average values are given for those terms among the sets of 50 terms for which the term relevance is defined (that is, those which co-occur jointly in some query-document pair). Again the term relevance values are substantially different for the two classes of DV terms, and these differences are statistically significant.

Also included in Table 3 are the multiplicative factors which relate the average values for the 50 high discriminators and the 50 low discriminators for each of the three measures (that is, the factor by

which the low average value must be multiplied to obtain the high). It may be seen that this factor is much higher for the term relevance than for either of NOCC/EK or S/N. The actual factors for the term relevance are 6.66, 80.0 and 36.33 for the CRAN, MED, and TIME collections, respectively. This indicates that the high discriminators have very much higher average term relevance than the low discriminators; alternatively expressed, there is substantial agreement between the semantic term relevance concept and the automatically derived term discrimination values.

The data already included in Table 3 are shown in term relevance order in Table 4. The output of Table 4 contains range and average values for NOCC/EK, S/N, and DV for the 50 terms with highest term precision and the 50 terms with lowest precision for the CRAN and TIME collections, respectively. Averages are produced for only 30 high and 30 low precision terms for the MED collection because in the medical environment the small number of available queries (24) made it possible to compute term precision values for only 172 terms in all.

It is clear from the output of Table 4 that the differences in the respective values are substantial in all cases, and the t-test values indicate that they are fully significant. For the three collections under study, the evidence indicates that terms with favorable formal parameters tend to be concentrated in documents identified as relevant by the user population, and vice-versa for terms with unfavorable formal parameters. Also shown in Table 4 are average document frequency ($\overline{B}^k$) and average total frequency ($\overline{F}^k$) values for the high and low relevance terms respectively. It may be seen that the

high relevance terms exhibit a much lower frequency spectrum (as expected for good discriminators) than the low relevance terms. Once again, it appears that the term relevance reflecting the semantic properties of the terms in their particular collection environment effects a division among the terms very similar to that obtained by the discrimination value computations.

In earlier work it was shown that the discrimination value theory which leads to the assignment to queries and documents of medium frequency terms (including also phrases constructed from high frequency terms, and term classes made up of low frequency terms) exhibits effective retrieval characteristics. [4,5,6] Typical average retrieval precision values for three different recall levels (recall of 0.1, 0.5, and 0.9) are shown for the three collections in Table 5. The output shows that the use of medium-frequency phrases and term classes improves performance by about 20 percent compared with the assignment of single terms alone. The comparison of Tables 3 and 4 between discrimination values on the one hand, and statistical and semantic parameters on the other, indicates that the same theory which produces such effective retrieval characteristics also conforms to the known statistical and linguistic theories of term behavior.

		50 Terms with High Discrimination Values		50 Terms with Low Discrimination Values
CRAN 424				
NOCC/EK	range	4455 to 925		1599 to 450
	average (in percent)	33.96%		10.96%
	t-test		0.00002	
	average high/average low		3.09	
S/N	range	1.954 to 0.699		1.222 to 0.000
	average (in percent)	60.18%		59.95%
	t-test		0.00002	
	average high/average low		1.00	
Term Relevance TR	range	392.66 to 0.00		74.35 to 0.00
	average (in percent)	14.06% (21 terms only)		2.11% (24 terms only)
	t-test		0.02208	
	average high/average low		6.66	

a) CRAN 424 Collection

Comparison of Statistical Models in

Term Discrimination Values

Table 3

		50 Terms with High Discrimination Values		50 Terms with Low Discrimination Values
MED 450				
NOCC/EK	range	9215 to 1359		7614 to 531
	average (in percent)	29.51%		15.61%
	t-test		0.00002	
	average high/average low		1.89	
S/N	range	2.792 to 0.693		1.738 to 0.126
	average (in percent)	48.46%		23.93%
	t-test		0.00002	
	average high/average low		2.03	
Term Relevance TR	range	874.00 to 0.00		9.43 to 0.00
	average (in percent)	16.0% (12 terms only)		0.20% (24 terms only)
	t-test		0.04274	
	average high/average low		80.0	

b) MED 450 Collection

Comparison of Statistical Models with

Term Discrimination Values (cont.)

Table 3

		50 Terms with High Discrimination Values		50 Terms with Low Discrimination Values
TIME 425				
NOCC/EK	range	13010 to 2330		4712 to 451
	average (in percent)	37.5%		10.81%
	t-test		0.00002	
	average high/average low		3.46	
S/N	range	2.966 to 1.424		1.876 to 0.231
	average	68.85%		26.44%
	t-test		0.00002	
	average high/average low		2.60	
Term Relevance TR	range	2454,00 to 62.62		27.73 to 0.44
	average (in percent)	15.26%		0.42%
		(12 terms only)		(23 terms only)
	t-test		0.03921	
	average high/average low		36.33	

c) TIME 425 Collection

Comparison of Statistical Models with

Term Discrimination Values (cont.)

Table 3

	50 High Relevance Terms $\overline{B}^k$=10.3 $\overline{F}^k$=24.6	50 Low Relevance Terms $\overline{B}^k$=58.9 $\overline{F}^k$=84.0
NOCC/EK	3657 to 420 average 38.95%	1584 to 432 average 20.66%
	t-test 0.00002 average high/average low 1.89	
S/N	1.953 to 0.000 average 42.81%	0.998 to 0.045 average 20.63%
	t-test 0.00002 average high/average low 2.08	
DV	1.223 to 0.002 average 65.52%	0.075 to -1.283 average 25.06%
	t-test 0.00140 average high/average low 2.61	

a) CRAN 424 Collection

Comparison of Term Relevance with

Term Discrimination Values

Table 4

	30 High Relevance Terms $\overleftarrow{R}^k$=9.5 $\overrightarrow{F}^k$=24.0	30 Low Relevance Terms $\overleftarrow{B}^k$=22.5 $\overrightarrow{F}^k$=41.9
NOCC/EK	2648 to 521 average 48.01%	2248 to 440 average 36.33%
	t-test 0.02378 average high/average low 1.32	
S/N	1.664 to 0.126 average 61.0%	1.259 to 0.000 average 46.33%
	t-test 0.00272 average high/average low 1.32.	
DV	0.135 to 0.006 average 62.11%	0.688 to -1.030 average 56.11%
	t-test 0.00671 average high/averag low 1.11	

b) MED 450 Collection

Comparison of Term Relevance with

Term Discrimination Values (cont.)

Table 4

	50 High Relevance Terms $\overline{B}^k$=12.5 $\overline{F}^k$=45.5	50 Low Relevance $\overline{B}^k$=94.5 $\overline{F}^k$=164.8
NOCC/EK	13010 to 417 average 16.1%	2266 to 431 average 3.4%
	t-test 0.00002 average high/average low 4.74	
S/N	2.966 to 0.000 average 42.31%	1.371 to 0.126 average 19.25%
	t-test 0.00002 average high/average low 2.20	
DV	0.156 to 0.000 average 94.05%	0.004 to -1.862 average 83.0%
	t-test 0.00148 average high/average low 1.13	

c) TIME 425 Collection

Comparison of Term Relevance with

Term Discrimination Values (cont.)

Table 4

Average Retrieval Precision For Various Recall Levels	CRAN 424	MED 540	TIME 425
A) Low Recall (0.1)			
i) single terms	.6844	.7891	.7496
ii) single terms, phrases and term classes	.8299 (+18%)	.9002 (+12%)	.8398 (+11%)
B) Medium Recall (0.5)			
i) single terms	.3131	.4384	.6351
ii) single terms, phrases and term classes	.4455 (+30%)	.5644 (+28%)	.7006 (+ 9%)
C) High Recall (0.9)			
i) single term	.1265	.1768	.3865
ii) single terms, phrases and terms classes	.1458 (+13%)	.3594 (+51%)	.4821 (+20%)

Recall-Precision Performance for

Medium Frequency Terms

(Discrimination Value Theory)

Table 5

References

[1] A. Bookstein and D.R. Swanson, Probabilistic Models for Automatic Indexing, Journal of the ASIS, Vol. 25, No. 5, September-October 1974, p. 312-318.

[2] D.C. Stone and M. Rubinoff, Statistical Generation of a Technical Vocabulary, American Documentation, Vol. 19, No. 4, October 1968, p. 411-412.

[3] S.F. Dennis, The Design and Testing of a Fully Automatic Indexing-Searching System for Documents Consisting of Expository Text, in Information Retrieval: A Critical Review, G. Schecter, editor, Thompson Book Co., Washington, 1967, p. 67-94.

[4] G. Salton, A Theory of Indexing, Regional Conference Series in Applied Mathematics No. 18, Society for Industrial and Applied Mathematics, Philadelphia, 1975.

[5] G. Salton, C.S. Yang and C.T. Yu, A Theory of Term Importance in Automatic Indexing, Journal of the ASIS, Vol. 26, No. 1, January-February 1975, p. 33-44.

[6] G. Salton, A. Wong, and C.S. Yang, A Vector Space Model for Automatic Indexing, Communications of the ACM, Vol. 18, No. 11, November 1975, p. 613-620.

[7] C.T. Yu and G. Salton, Precision Weighting — An Effective Automatic Indexing Method, to be published in Journal of the ACM, 1976.

[8] D. Williamson, R. Williamson, and M. Lesk, The Cornell Implementation of the SMART System, in The SMART Retrieval System, G. Salton, editor Prentice-Hall, Englewood Cliffs, NJ, 1971, Chapter 2.

American Journal of Computational Linguistics *Microfiche 55 : 84*

S N O P A R : A Grammar Testing System

T. P. Kehler

Department of Mathematics
Texas Woman's University
Denton, Texas 76204

A grammar testing program has been developed which permits modeling augmented transition network grammars as a series of SNOBOL4 functions. SNOPAR is designed for linguistics teaching and research. Emphasis is placed on the development of small to medium grammars in a variety of languages. The system has been used so far to develop a grammar of English for use in a transformational grammar course and to develop small grammars of a Nigerian and an American Indian language. Intended applications of SNOPAR are in field linguistics and grammar model testing.

The main part of the program is the routine PARSER. When PARSER is called with a lexicon and grammar, input strings are parsed according to the model grammar. The PARSER functions available for grammar development are CAT, PARSE, SETR, GETR, RESET, TESTR, GETF, GETCL, TO, BACK, FINDWRD, and BUILDS. The function operations and descriptions of their arguments are given in Table 1. After a parsing, PARSER returns control to the user permitting examination of stacks and registers at all

TABLE 1

PARSER FUNCTIONS

CAT	looks up the word class of the current first word in the input string. If the word is not in the lexicon an add routine is called which permits additions. If CAT succeeds by matching the current word class with its argument, the word is removed from the input string and pushed onto a stack (SAVEW). If it fails an alternate class is tested, provided that the alternate flag is on. Fail return leaves the surface string unaltered.
PARSE	calls the function given by its argument and if successful pushes the structure returned by the function onto a stack (SAVEQ) and assigns the structure to the Q register.
SETR	sets the values of registers. It has three arguments level, register name, and value. Each call of SETR causes the register name specified to be placed on a list for the specified level. SETR entries are treated as stacks, providing automatic saves for recursive calls.
GETR	returns the contents of the register name specified by its argument, and pops it off the stack saving the last value.
TESTR	looks at the value of the register name specified by its argument without popping it off the stack.
RESET	changes the value of a register without changing stack levels.
GETF	looks up the feature value for a feature specified by its argument of the current value of the word register. Any word can be specified by giving a second argument. If GETF fails for the word it looks at the root form of the word for certain features
GETCL	looks up the word class of the word specified by its argument.
TO	has as its argument, the new state label. It pushes the label onto a stack (PATH); outputs the state, outputs the contents of the Qregister, and transfers control to the new state.
BACK	backs to the state specified by its argument.
FINDWRD	tests for the word specified by its argument.
BUILDS	builds a structure from the register name list.

levels. In the examination stage, traces may be turned on lexical entries may be examined or minor changes to the grammar may be made. Functions available for the examination of stacks, registers and lexicon are POP, OUT, GETR, LOOKLEX, and TRACE. A function GETENG is also available for dictionary lookup in other languages. PARSER requires approximately 150 lines of SNOBOL code and is currently operating on a DEC 10. A batch version has been tested on an IBM 360

In order to use PARSER, a grammar and lexicon must be developed as disc files. Since the grammar is developed as a separate file different components of the grammar can be tested and put together in a variety of configurations. If a lexicon is not developed as a disc file prior to a parse, it may be entered from the terminal A simple grammar which produces surface structure trees is shown in Example 1 along with a sample parsing. A portion of the lexicon is shown at the bottom of the page. Example 2 shows the use of the GETF function to handle agreement between plural adjectives and a plural marker in Angas, a Nigerian language. Example 3 shows a grammar which handles sentence embedding in English. Some sample parsings are shown. The model used for the Example 3 grammar is basically the one developed in English Transformational Grammar by Jacobs and Rosenbaum. A basic case grammar for English as well as a semantic oriented grammar for Choctaw (an American Indian language) are in development.

The complete SNOPAR system has in addition to PARSER a routine for generating grammars from a state transition graph and a register action table. This routine called NEW guides the user through a state transition graph and register actions to produce a grammar compatible with PARSER. The SNOPAR NEW routine is still in development. The current routine allows development of small grammars. The new developments will provide diagnostics of grammar errors. SNOPAR also has a line editor (FIXUP) and disc I/O commands. The complete system allows repetitive testing of model grammars, permits editing; and has trace capabilities for grammar debugging.

Example 1

```
S       PARSE(NP())          :S(TO(.SNP))
        CAT('AUX')           :S(TO(.QUES))F(FRETURN)
SNP     SETR(.S,'TYPE','DCL')
        SETR(.S,'SUBJ',Q)
TRYVP   PARSE(VP())          :S(TO(.POPS))F(FRETURN)
QUES    SETR(.S,'TYPE','QUESTION')
        SETR(.S,'AUX',Q)
        SETR(.S,'TENSE',GETF('TNS'))
        PARSE(NP())          :S(TO(.QNP))F(FRETURN)
QNP     SETR(.S,'SUBJ',Q)            :(TO(.TRYVP))
POPS    SETR(.S,'PRED',Q)
        $ = BUILDS(S)        :(RETURN)
NP      CAT('DET')           :S(TO(.DET))
        CAT('PRO')           :S(TO(.PRO))
        CAT('NPR')           :S(TO(.NPR))F(FRETURN)
NPR     SETR(.NP,'PROP',Q)           :(TO(.POPNP))
PRO     SETR(.NP,'PRO',Q)            :(TO(.POPNP))
DET     SETR(.NP,'DET',Q)
ADJ     CAT('ADJ')           :F(TO(.TRYN))
        SETR(.NP,'ADJ',Q)            :(TO(.ADJ))
TRYN    CAT('N')             :F(FRETURN)
        SETR(.NP,'N',Q)
TRYPP   PARSE(PP())          :F(TO(.POPNP))
        SETR(.NP,'PP',Q)             :(TO(.TRYPP))
POPNP   NP = BUILDS(NP)      :(RETURN)
PP      CAT('PREP')          :F(FRETURN)
        SETR(.PP,'PREP',Q)
        PARSE(NP())          :F(FRETURN)
        SETR(.PP,'PREPNP',Q)
        PP = BUILDS(PP)      :(RETURN)
VP      CAT('V')             :F(FRETURN)
        SETR(.VP,'V',Q)
        PARSE(NP())          :S(TO(.VNP))
TRYVPP  PARSE(PP())          :F(TO(.POPVP))
        SETR(.VP,'PP',Q)             :(TO(.TRYVPP))
VNP     SETR(.VP,'NP',Q)             :(TO(.POPVP))
POPVP   VP = BUILDS(VP)      (RETURN)

 TY LEXENG.1
        DID= (AUX)(TNS PAST).
        CAN= (AUX)(TNS PRES).
        COULD= (FORM 'CAN).
        WILL= (AUX)(TNS FUT).
        THE= (DET).
        A= (DET).
        AN= (DET).
        THAT= (CLIND).
        BOY= (N)(NBR SING).
        BOYS= (N)(NBR PL).
        GIRL= (N)(NBR SING).
        GIRLS= (FORM GIRL)(NBR PL).
        MAN= (N)(NBR SING).
        MEN= (N)(NBR PL).
        WOMAN= (N)(NBR SING).
        WOMEN= (N)(NBR PL).
        TABLE= (N)(NBR SING).
```

```
DID YOU WALK TO THE VILLAGE
DID YOU WALK TO THE VILLAGE
STATE QUES
COMPLEMENT STRING: YOU WALK TO THE VILLAGE
BUILD STRUCTURE DID
STATE PRO
COMPLEMENT STRING: WALK TO THE VILLAGE
BUILD STRUCTURE YOU
STATE POPNP
COMPLEMENT STRING: WALK TO THE VILLAGE
BUILD STRUCTURE YOU
STATE QNP
COMPLEMENT STRING: WALK TO THE VILLAGE
BUILD STRUCTURE (NP(PRO YOU))
STATE TRYVP
COMPLEMENT STRING: WALK TO THE VILLAGE
BUILD STRUCTURE (NP(PRO YOU))
STATE DET
COMPLEMENT STRING: VILLAGE
BUILD STRUCTURE THE
STATE TRYN
COMPLEMENT STRING: VILLAGE
BUILD STRUCTURE
STATE POPNP
COMPLEMENT STRING:
BUILD STRUCTURE VILLAGE
STATE TRYVPP
COMPLEMENT STRING:
BUILD STRUCTURE (PP(PREP TO)(PREPNP (NP(DET THE)(N VILLAGE))))
STATE POPVP
COMPLEMENT STRING:
BUILD STRUCTURE (PP(PREP TO)(PREPNP (NP(DET THE)(N VILLAGE))))
STATE POPS
COMPLEMENT STRING:
BUILD STRUCTURE:
(VP(V WALK)(PP (PP(PREP TO)(PREPNP (NP(DET THE)(N VILLAGE))))))

STATE S
COMPLEMENT STRING:
BUILD STRUCTURE:
(S(TYPE QUESTION)(AUX DID)(TENSE PAST)(SUBJ (NP(PRO YOU)))
(PRED (VP(V WALK)(PP (PP(PREP TO)(PREPNP (NP(DET THE)(N VILLAGE)))))))
)


DO YOU WANT TO EXAMINE THE REGISTERS ?
YES
*
OT      OUTPUT = POP(PATH)      ;S(OT)F(EXAS\S\MIN)
EP\P\OF
POPS
POPVP
TRYVPP
POPNP
TRYN
DET
TRYVP
QNP
POPNP
PRO
QUES
DO YOU WANT TO EXAMINE THE REGISTERS ?
^C
```

ANGAS NOUN PHRASE

```
NP      CAT('NOUN')          :S(TO(.POS))F(FRETURN)
POS     SETR(.NP,'NOUN',Q)
        CAT('POSPRO')        :F(TO(.KI))
        SETR(.NP,'POSPRO',Q)      :(TO(.ADJ))
KI      CAT('KI')            :F(TO(.ADJ))
        SETR(.NP,'POSS',Q)
        CAT('NPR')           :F(FRETURN)
        SETR(.NP,'NPR',Q)
ADJ     CAT('ADJ')           :F(TO(.KOM))
        SETR(.NP,'ADJ',Q)          :(TO(.DET))
KOM     FINDWRD('KOMEYE')          :F(TO(.DET))
        SETR(.NP,'REL',Q)           :(TO(.ADJ))
DET     CAT('DET')           :F(TO(.PL))
        SETR(.NP,'DET',Q)
PL      CAT('PL')            :S(TO(.PLT))
        IDENT(GETF('PL',GETR('ADJ')),'PL')        :S(FRETURN)F(TO(.NUM))
PLT     SETR(.NP,'PL',Q)
        IDENT(GETF('PL',GETR('ADJ')),'-PL')       :S(FRETURN)
NUM     CAT('NUM')           :F(TO(.POPNP))
        SETR(.NP,'NUM',Q)
        IDENT(WORD,'BAP')          :S(TO(.TMWA))F(TO(.POPNP))
TMWA    IDENT(GETR('PL'),'MWA') :F(FRETURN)
POPNP   NP = BUILDS(NP) :(RETURN)
EOG
END
```

ANGAS LEXICON

```
L       L<'AS'> = '(NOUN)(ENG DOG)'
        L<'MAT'> = '(NOUN)(ENG WOMAN)'
        L<'FANA'> = '(POSPRO)(ENG,MY)'
        L<'RIIT'> = '(ADJ)(PL -PL)(ENG GOOD)'
        L<'RIIT-RIIT'> = '(ADJ)(PL PL)(ENG GOOD)'
        L<'BIJIM'> = (ADJ)(PL -PL)(ENG BIG)'
        L<'NAN-NAN'> = '(ADJ)(PL PL)(ENG BIG)'
        L<'GAK'> = '(NUM)(ENG ONE)'
        L<'BAP'> = '(NUM)(ENG TWO)'
        L<'NYII'> = '(DET)(ENG THIS)'
        L<'DA'> = '(DET)(ENG THE)'
        L<'CE'> = '(DET)(ENG A)'
        L<'MWA'> = '(PL)(ENG PLUR)'
        L<'BULUS'> = '(NPR)(ENG NAME)
        L<'KI'> = '(KI)(ENG POSSESSIVE)
*EX$$
```

```
STATE POPNP
COMPLEMENT STRING:
BUILD STRUCTURE MWA
STATE NP
COMPLEMENT STRING:
BUILD STRUCTURE (NP(NOUN AS)(POSPRO FANA)(ADJ NAN-NAN)(DET CE)(PL MWA)
)
ENGLISH: DOG MY BIG A PLUR
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
AS FANA BIJIM CE MWA
AS FANA BIJIM CE MWA
STATE POS
COMPLEMENT STRING: FANA BIJIM CE MWA
BUILD STRUCTURE AS
STATE ADJ
COMPLEMENT STRING: BIJIM CE MWA
BUILD STRUCTURE FANA
STATE DET
COMPLEMENT STRING: CE MWA
BUILD STRUCTURE BIJIM
STATE PLT
COMPLEMENT STRING:
BUILD STRUCTURE MWA
STATE NP
COMPLEMENT STRING: DID NOT PARSE
BUILD STRUCTURE MWA
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
AS MWA
AS MWA
STATE POS
COMPLEMENT STRING: MWA
BUILD STRUCTURE AS
STATE KI
COMPLEMENT STRING: MWA
BUILD STRUCTURE
STATE ADJ
COMPLEMENT STRING: MWA
BUILD STRUCTURE
STATE KOM
COMPLEMENT STRING: MWA
BUILD STRUCTURE
STATE DET
COMPLEMENT STRING: MWA
BUILD STRUCTURE
STATE PL
COMPLEMENT STRING: MWA
BUILD STRUCTURE
STATE PLT
COMPLEMENT STRING:
BUILD STRUCTURE MWA
STATE POPNP
COMPLEMENT STRING:
BUILD STRUCTURE MWA
STATE NP
COMPLEMENT STRING:
BUILD STRUCTURE (NP(NOUN AS)(PL MWA))
ENGLISH: DOG PLUR
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
```

```
        FUNCTION DEFINITIONS
GRAM     DEFINE('S()N')
         DEFINE('ES()')
         DEFINE('NP()M,N')
         DEFINE('PP()')
         DEFINE('VP()N')
         DEFINE('IO()')
*        S PARSER
         PARSE(S())
         OUT('S',STR,Q)   :(NXT,COM)
S        PARSE(NP())              :S(TO(,SNP))
         CAT(,AUX)              :S(TO(,Q))
         PARSE(VP())      :S(TO(,IMP))F(FRETURN)
SNP      SETR(,S,'SUBJ',Q)
         SETR(,S,'TYPE','DCL')
         PARSE(VP())      :S(TO(,POPS))
         CAT(,AUX)        :S(TO(,AX))F(FRETURN)
IMP      SETR(,S,'TYPE','IMP')
         SETR(,S,'SUBJ','(PRO YOU)')      :(TO(,POPS))
Q        SETR(,S,'AUX',Q)
         SETR(,S,'TNS',GETF('TNS'))
         SETR(,S,'TYPE','Q')
         PARSE(NP())                          :S(TO(,QNP))F(FRETURN)
AX       SETR(,S,'AUX',Q)
         SETR(,S,'TNS',GETF('TNS'))
         FINDWRD('HAVE') SETR(,S,'HA','HAVE')
         PARSE(VP())                           :S(TO(,POPS))F(FRETURN)
QNP      SETR(,S,'SUBJ',Q)
         FINDWRD('HAVE') SETR(,S,'HA','HAVE')
         PARSE(VP())                      :S(TO(,POPS))F(FRETURN)
POPS     SETR(,S,'PRED',Q)
         # = BUILDS('/S/TYPE/SUBJ/PRED/')                    :(RETURN)
* NP PARSER
NP       CAT('DET')                            :S(TO(,DET))
         CAT('PRO')                    :S(TO(,PRO))
         CAT('NPR')                           :S(TO(,NPR))
         PARSE(ES())      :S(TO(,NPES))
                                                  :(TO(,PLNP))
DET      SETR(,NP,'DET',Q)
ADJ      CAT('ADJ')                           :F(TO(,N))
         SETR(,NP,'ADJ' M,Q)
         BUMP('M')                             :(TO(,ADJ))
N        CAT('N')                      :F(FRETURN)
         SETR(,NP,'N',Q)                       :(TO(,NPP))
POSPRO   SETR(,NP,'PRO',Q)             :(TO(,ADJ))
NPR      SETR(,NP,'NPR',Q)
         IS(GETF('CASE'),'POS') CHGNAM('NP','NPR','POSNPR')  :F(TO(,NPP))
         PARSE(ES())      :F(TO(,ADJ))
         SETR(,NP,'POSS',Q)
         NP = BUILDS('/NP/NPR/POSS/')      :(RETURN)
POPNP    PARSE(ES())      :S(TO(,NPES))
         NP = BUILDS(NP)  :(RETURN)
NPP      PARSE(PP())                   :F(TO(,POPNP))
         SETR(,NP,'NPP' N,Q)
         BUMP('N')                        :(TO(,NPP))
PRO      GETF('CASE') 'POS'            :S(TO(,POSPRO))
         SETR(,NP,'PRO',Q)             :(TO(,POPNP))
PLNP     CAT('ADJ')          :F(TO(,NPL))
         SETR(,NP,'ADJ' ,Q)
         :(TO(,PLNP))
```

```
NPL     CAT('N')        IS(GETF('NBR'),'PL')          :F(FRETURN)
        SETR(,NP,'N',Q)                        :(TO(,POPNP))
NPES    SETR(,NP,'COMP',Q)
        NP = BUILDS(NP) :(RETURN)
*               PP PARSER
PP      CAT('PREP')     :S(TO(,PREP))F(FRETURN)
PREP    P<'PREP'> = Q
        PP = '(PREP ' P<'PREP'>   NP()
        ')'   :S(RETURN)F(FRETURN)
*       VP PARSER
VP      CAT('V')              :F(TO(,AUXBE))
        SETR(,VP,'TNS',GETF('TNS'))
        HASNAM('S','AUX') GETR('TNS')
        IS(GETF('VTYP'),'TRANS')
        :S(TO(,TRANS))F(TO(,ITRAN))
TRANS   SETR(,VP,'VT',Q)                       :(TO(,VNP))
ITRAN   SETR(,VP,'V',Q)              :(TO(,NTPP))
TADJ    CAT('ADJ')                   :S(TO(,VADJ))
        PARSE(NP())                  :S(TO(,NTNP))
NTPP    PARSE(PP())                  :F(TO(,POPVP))
        SETR(,VP,'VPP' N,Q)
        BUMP('N')           :(TO(,NTPP))
VADJ    SETR(,VP,'ADJ',Q)
        PARSE(ES())         :S(TO(,VADJES))
        VP = BUILDS(VP)
VDJPP   PARSE(PP())         :F(RETURN)
        VP = VP Q           :(TO(,VDJPP))
VADJES  SETR(,VP,'ADJES',Q)        :(TO(,POPVP))
NTNP    SETR(,VP,'NTNP',Q)                 :(TO(,POPVP))
VNP     PARSE(IO())         :S(TO(,IOI))
        PARSE(NP())         :F(FRETURN)
        SETR(,VP,'OBJ',Q)
        PARSE(IO())                  :S(TO(,IOL))F(TO(,POPVP))
IOI             SETR(,VP,'IO',Q)
        PARSE(NP())                  :S(TO(,VIONP))F(FRETURN)
VIONP   SETR(,VP,'OBJ',Q)                      :(TO(,POPVP))
IOL     SETR(,VP,'IO',Q)                       :(TO(,POPVP))
AUXBE   CAT('V','ALT')  :S(TO(,BE))
        IS(TESTR('TYPE'),'Q') IS(TESTR('AUX'),'BE')      :F(TO(,TRYES))
        SETR(,VP,'V',GETR('AUX'))              :(TO(,PAS))
TRYES   IS(TESTR('IF')) PARSE(ES())            :F(FRETURN)
        NP = Q  :(RETURN)
BE      SETR(,VP,'V',Q)
        SETR(,VP,'TNS',GETF('TNS'))
PAS     CAT('V')                     :F(TO(,TADJ))
        WORD 'ING'                   :S(TO(,ING))
        IS(GETF('VTYP'),'TRANS')               :F(FRETURN)
        GETF('TNS') 'PPRT'           :S(TRPAS)F(FRETURN)
ING     VP =
        SETR(,VP,'AUX','BE')
        SETR(,VP,'TNS','PPRG')
        SETR(,VP,'V',Q)
        PARSE(NP())         :S(TO(,PRNP))
        VP = BUILDS(VP)
PRPP    PARSE(PP())         :F(RETURN)
        VP = VP Q           :(TO(,PRPP))
PRNP    SETR(,VP,'PRNP',Q)             :(TO(,POPVP))
TRPAS   VP =
        SETR(,VP,'AUX','BE')
        DIFFER(TESTR('TYPE'),'Q') SETR(,VP,'TNS','PPRT')
```

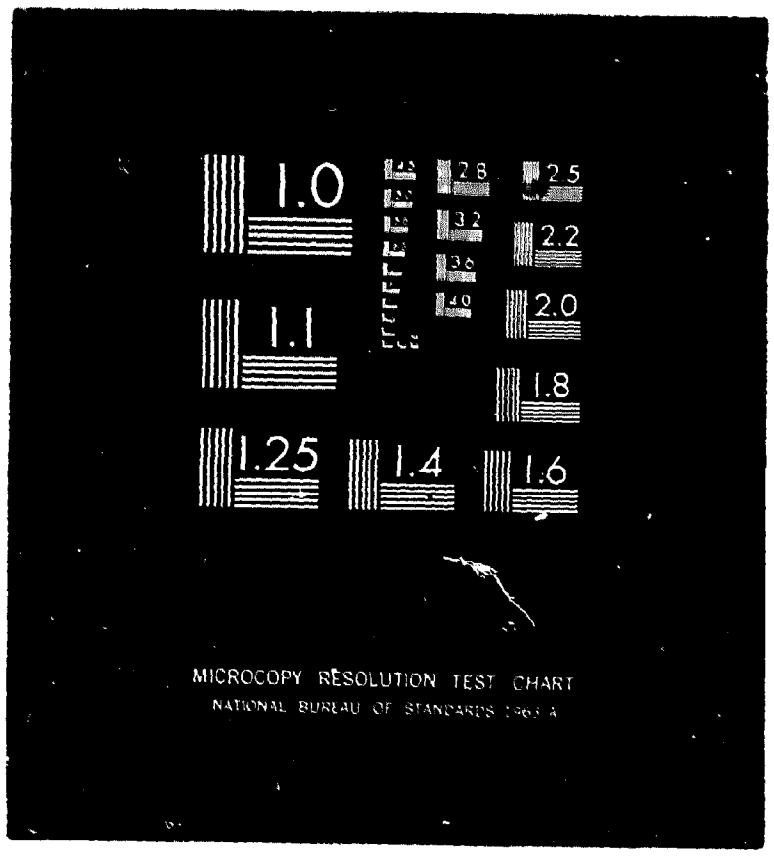
```
        GETR('V')
        SETR(,VP,'V',Q)
        PARSE(IO())..     :S(TO(,PIO))
        FINDWRD('BY')              :S(TO(,PNPTST))
        FINDWRD('FROM')            :S(TO(,PNPTST))
        PARSE(ES())       :S(TO(,VPES))F(TO(,CHGSBJ))
CHGSBJ  SETR(,VP,'OBJ',R<'SUBJ'>)
        IS(TESTR('TYPE'),'DCL') RESET('TYPE','TRPAS')
        IS(TESTR('TYPE'),'Q') RESET('TYPE','QPAS')
        RESET('SUBJ','SOMEONE')           :(TO(,POPVP))
VPES    SETR(,VP,'OBJES',Q)
        VP = BUILDS(VP)
TRPP    PARSE(PP())       :F(RETURN)
        VP = VP Q         :(TO(,TRPP))
PIO     SETR(,VP,'IO',Q)
        FINDWRD('BY')     :S(TO(,PNPTST))
        FINDWRD('FROM')            :S(TO(,PNPTST))F(FRETURN)
PNPTST  PARSE(NP())       :S(TO(,PNP))F(FRETURN)
PNP     SETR(,VP,'OBJ',GETR('SUBJ'))
        RESET('SUBJ',Q)
        IS(TESTR('TYPE'),'DCL') RESET('TYPE','TRPAS')
        IS(TESTR('TYPE'),'Q') RESET('TYPE','QPAS')
        IS(R<'IO'>)                :F(TO(,POPVP))
        PARSE(IO())                :S(TO(,PIOL))F(TO(,POPVP))
PIOL    SETR(,VP,'IO',Q)           :S(TO(,POPVP))
POPVP   VP = BUILDS(VP)            :(RETURN)
*       INDIRECT OBJECT
IO      FINDWRD('TO')              :S(TO(,IOTO))
        FINDWRD('FOR')             :S(TO(,IOFOR))F(FRETURN)
IOTO    SETR(,IO,'PREP','TO')
        PARSE(NP())                :S(TO(,IONP))
ADD,TO  STR = 'IO ' STR  :(FRETURN)
IOFOR   SETR(,IO,'PREP',Q)
        PARSE(NP())       :S(TO(,IONP))
        STR = 'FOR ' STR           :(FRETURN)
IONP    SETR(,IO,'IONP',Q)
        IO = BUILDS(IO)  :(RETURN)
ES      CAT('CLIND')      :S(TO(,TES))
        FINDWRD('TO')     :S(TO(,THV))
        FINDWRD('HAVING')          :S(TO(,ESVP))
        IS(GETF('TNS'),'PPRG')     :F(FRETURN)
        PARSE(VP())       :F(FRETURN)
        ES = Q   :(RETURN
TES     SETR(,ES,'CL,IND ',Q)
        PARSE(S())        :S(TO(,POPES))F(FRETURN)
THV     SETR(,ES,'INF', TO')
        FINDWRD('HAVE')   :S(TO(,ESVP))
        PARSE('    '))    :F(ADD,TO)
        SETR(,ES,'ESVP',Q)
        ES = BUILDS('/S/TYPE/SUBJ/ESVP/')          :(RETURN)
ESVP    SETR(,ES,'AUX','HAVE')
        PARSE(VP())       :F(FRETURN)
        SETR(,ES,'ESVP',Q)
        ES = BUILDS('/S/TYPE/SUBJ/AUX/ESVP/')      :(RETURN)
POPES   ES = Q   :(RETURN)
END
```

```
JOHN MAY BELIEVED TO HAVE BEEN DELAYED
JOHN MAY BELIEVED TO HAVE BEEN DELAYED
STATE 2
COMPLEMENT STRING: DELAYED
BUILD STRUCTURE:
(1 (TYPE TRANS) (SUBJ SOMEONE) (PRED (VP (AUX BE) (TNS PERF PPRT)
(V BELIEVE) (OBJ1 (1 (TYPE TRANS) (SUBJ SOMEONE) (AUX HAVE)
(E1VP (VP (AUX BE) (TNS PPRT) (V DELAY)) (OBJ (NP (NPP JOHN)))))))))

DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
I WANT TO GO
I WANT TO GO
STATE 1
COMPLEMENT STRING: GO
BUILD STRUCTURE:
(1 (TYPE DCL) (SUBJ (NP (PRO I))) (PRED (VP (TNS PRES) (VT WANT)
(OBJ (NP (COMP (1 (TYPE DCL) (SUBJ (NP (PRO I))) (E1VP (VP (V GO)
(TNS PRES)))))))))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
I THINK THAT I SAW YOU WITH HER
I THINK THAT I SAW YOU WITH HER
SAW NOT IN LEXICON
LEXICON ADD. TO ABORT- PARSE TYPE STOP. ELSE TYPE YES
YES
WORD?
SAW
FEATURE STRING
(VA (VTYP TRANS) (TNS PAST))
WORD?

STATE 3
COMPLEMENT STRING: HER
BUILD STRUCTURE:
(1 (TYPE DCL) (SUBJ (NP (PRO I))) (PRED (VP (TNS PRES) (VT THINK)
(OBJ (NP (COMP (1 (TYPE DCL) (SUBJ (NP (PRO I))) (PRED (VP (TNS PAST)
(VT SAW) (OBJ (NP (PRO YOU)) (PREP WITH (NP (PRO HER)))))))))))))
```

```
INPUT STRUCTURE TO BE PARSED
JOHN'S BELIEVING THAT MARY IS GOING TO THE VILLAGE IS MYSTERIOUS
JOHN'S BELIEVING THAT MARY IS GOING TO THE VILLAGE IS MYSTERIOUS
STATE S
COMPLEMENT STRING: MYSTERIOUS
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(NPR JOHN'S)(POSS (VP(TNS PRES)(VT BELIEVE)
(OBJ (NP(COMP (S(TYPE DCL)(SUBJ (NP(NPR MARY)))(PRED (VP(AUX BE)
(TNS PRNG)(V GO))(PREP TO(NP(DET THE)(N VILLAGE)))))))))))
(PRED (VP(V BE)(TNS PRES)(ADJ MYSTERIOUS))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
THAT HE,BROKE HER DISH IS SERIOUS
THAT HE BROKE HER DISH IS SERIOUS
STATE S
COMPLEMENT STRING: SERIOUS
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(COMP (S(TYPE DCL)(SUBJ (NP(PRO HE)))
(PRED (VP(TNS PAST)(VT BREAK)(OBJ (NP(PRO HER)(N DISH))))))))
(PRED (VP(V BE)(TNS PRES)(ADJ SERIOUS))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
THE BOY BREAKING THE GLASS IS MULLIGAN
THE BOY BREAKING THE GLASS IS MULLIGAN
STATE S
COMPLEMENT STRING: MULLIGAN
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(DET THE)(N BOY)(EMB (VP(TNS PPRG)(VT BREAK)
(OBJ (NP(DET THE)(N GLASS))))))(PRED (VP(V BE)(TNS PRES)
(NTNP (NP(NPR MULLIGAN)))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
JOHN'S BEING THIN IS NICE
JOHN'S BEING THIN IS NICE
STATE S
COMPLEMENT STRING: NICE
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(NPR JOHN'S)(POSS (VP(V BE)(TNS PPRG)
(ADJ THIN))))(PRED (VP(V BE)(TNS PRES)(ADJ NICE))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
TO BE A MAN WAS HIS DREAM
TO BE A MAN WAS HIS DREAM
STATE S
COMPLEMENT STRING: DREAM
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(COMP (S(TYPE )(SUBJ )(ESVP (VP(V BE)
(TNS PRES)(NTNP (NP(DET A)(N MAN))))))))(PRED (VP(V BE)
(TNS PAST))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
BREAKING DISHES IS RECKLESS
BREAKING DISHES IS RECKLESS
STATE S
COMPLEMENT STRING: RECKLESS
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(COMP (VP(TNS PPRG)(VT BREAK)(OBJ (NP(N DISHES))
```

```
THE BOY RUNNIJ\J\NG TO THE HOUSE IS JOHN
THE BOY RUNNING TO THE HOUSE IS JOHN
STATE S
COMPLEMENT STRING: JOHN
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(DET THE)(N BOY)(EMB (VP(V RUN)(TNS PPRG)
(VPP (PREP TO(NP(DET THE)(N HOUSE))))))))(PRED (VP(V BE)
(TNS PRES)(NTNP (NP(NPR JOHN))))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
BREAKING DISHES IS RECKLESS
BREAKING DISHES IS RECKLESS
STATE S
COMPLEMENT STRING: RECKLESS
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(COMP (VP(TNS PPRG)(VT BREAK)(OBJ (NP(N DISHES)))
))))
(PRED (VP(V BE)(TNS PRES)(ADJ RECKLESS))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
I WAS THINKING THAT YOU B\B\WERE CONSERVATIVE
I WAS THINKING THAT YOU WERE CONSERVATIVE
STATE S
COMPLEMENT STRING: CONSERVATIVE
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(PRO I)))(PRED (VP(AUX BE)(TNS PPRG)
(V THINK)(PRNP (NP(COMP (S(TYPE DCL)(SUBJ (NP(PRO YOU)))
(PRED (VP(V BE)(TNS PAST)(ADJ CONSERVATIVE)))))))))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
JOHN WAS BELIEVED TO BE DELAYED
JOHN WAS BELIEVED TO BE DELAYED
STATE S
COMPLEMENT STRING: DELAYED
BUILD STRUCTURE:
(S(TYPE TRPAS)(SUBJ SOMEONE)(PRED (VP(AUX BE)(TNS PRES)(V BELIEVE)
(OBJES (S(TYPE TRPAS)(SUBJ SOMEONE)(ESVP (VP(AUX BE)(TNS PPRT)
(V DELAY)(OBJ (NP(NPR JOHN)))))))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
NO
INPUT STRUCTURE TO BE PARSED
THAT HE BROKE HER DISH WAS SERIOUS
THAT HE BROKE HER DISH WAS SERIOUS
STATE S
COMPLEMENT STRING: SERIOUS
BUILD STRUCTURE:
(S(TYPE DCL)(SUBJ (NP(COMP (S(TYPE DCL)(SUBJ (NP(PRO HE)))
(PRED (VP(TNS PAST)(VT BREAK)(OBJ (NP(PRO HER)(N DISH)))))))))
(PRED (VP(V BE)(TNS PAST)(ADJ SERIOUS))))
DO YOU WANT TO EXAMINE THE REGISTERS ?
YES
```

END


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A