American Journal of Computational Linguistics *Microfiche 29*

THE FINITE STRING

NEWSLETTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

VOLUME 12 - NUMBER 5 NOVEMBER 1975

With this mailing, AJCL completes the substantive part of its second year of publication. Index guides will be mailed to all 1975 subscribers early in 1976; the topical index will again be printed on the tabbed cards. This mailing contains a questionnaire about the format and scope of AJCL; responses will give the Association and Editorial Board guidance in policy development.

AMERICAN JOURNAL OF COMPUTATIONAL LINGUISTICS is published by the Center for Applied Linguistics for the Association for Computational Linguistics

EDITOR: *David G. Hays, Professor of Linguistics and of Computer Science, State University of New York, Buffalo*

EDITORIAL ASSISTANT: *William Benzon*

EDITORIAL ADDRESS· *Twin Willows, Wanakah, New York 14075*

MANAGING EDITOR: *A. Hood Roberts, Deputy Director, Center for Applied Linguistics*

ASSISTANT MANAGING EDITOR: *Penny Pickett*

PRODUCTION AND SUBSCRIPTION ADDRESS *1611 North Kent Street, Arlington, Virginia 22209*

TABLE OF CONTENTS

ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

Officers 1976 3

Executive Committee Meeting 4

Financial Report 6

PERSONAL NOTES 7

NSF: Reorganization 8

MT: Latsec Shows in Zurich 9

MEETINGS

XIIth International Congress of Linguists 10

ACM Conference - Employment Register 12

Compcon 76 13

Cybernetics and Systems Research 14

Literary and Linguistic Computing 16

Charles S Peirce Society 17

Computer Applications to Learning 18

Historical Linguistics 19

Short announcements and reminders 20

GRANTS

ACLS: Soviet Studies, East European Languages 22

Special Libraries Association Scholarships 23

ACKNOWLEDGMENT 24

COMPUTER GENERATION OF SENTENCES BY SYSTEMIC GRAMMAR

John Self 25

INTERPRETATION & INTEGRATION OF SENTENCES INTO A C-NET

Th R. Hofmann 46

OFFICERS 1976

President	STANLEY R. PETRICK IBM Research
Vice-President	JOSEPH E. GRIMES Cornell University
Secretary-Treasurer	A. HOOD ROBERTS Center for Applied Linguistics
Executive Committee	TIMOTHY C. DILLER Sperry-Univac
Nominating Committee	ARAVIND K. JOSHI University of Pennsylvania

Continuing members of the Executive Committee are Charles J Rieger III (through 1976) and Bonnie Nash-Webber (through 1977 Continuing members of the Nominating Committee are Robert Barnes (1976) and William A Woods, Jr. (1977) The Editor is a member of the Executive Committee ex-officio.

EXECUTIVE COMMITTEE MEETING

Present at the meeting in Boston, October 30, 1975, were President Joshi, Vice-President Petrick, Executive Committee Member Nash-Webber, Editor Hays, Secretary-Treasurer Roberts, Past Presidents Barnes and Woods, and Martin Kay, guest.

Current membership is

Exchange, gift, life	45
Individual, paid through 1974	187
Individual, paid through 1975	517
Institution, paid through 1974	44
Institution, paid through 1975	111

Nash-Webber reported that about 600 copies of the Proceedings of the June Interdisciplinary Workshop have been sold; the text is used in courses at Berkeley (O'Malley) and Wisconsin (Klein) Edited discussions are in preparation.

The Committee decided to hold the 1976 meeting of ACL at San Francisco in October, jointly with the ASIS meeting. Discussion with Kay did not lead to generally satisfactory proposals for a combination of the ACL meeting with the International Conference at Ottawa.

MORE

The Committee decided to establish permanently the reduction of meeting registration fees for students, but not to introduce a reduced membership fee

The format of the Journal was discussed; a questionnaire is to be distributed to members, calling for their opinions about several combinations of printed and microfiched publication.

The Committee decided to offer a limited subsidy to the newsletter proposed by Hans Karlgren of Stockholm; the proposal is for 10 issues per year, free distribution, and a publication limited to announcements of significant events and other matters of general interest, with occasional short technical notes.

FINANCIAL REPORT

July 26, 1974 - October 30, 1975

Balance as of July 26, 1974		$2,264.20
RECEIPTS		
Membership dues '74, '75	$15,108.76	
Redistribution of AFIPS surplus	1,645.00	
Sales of back issues, TFS	83.00	
ACL '75 meeting receipts to date	587.00	
		$17,423.76
		$19,687.96
DISBURSEMENTS		
Administrative costs, office supplies, PO mailing, AJCL costs not covered by Acc't #317	$ 3,250.78	
TFS (microfiched, old issues bound)	115.22	
Membership, ACAL	50.00	
AFIPS dues 1974	500.00	
AFIPS dues 1975	500.00	
Annual meeting costs, 1974 and 1975 to date	196.89	
Paid out of ACL membership receipts into CAL account #317 for AJCL: from		
1974 receipts	7,968.88	
1975 receipts	5,000.00	
		$17,581.77
Balance as of October 30, 1975		$ 2,106.19
SAVINGS		
Certificates and accounts now worth		$ 6,607.55

A. Hood Roberts, Secretary-Treasurer

PERSONAL NOTES

DILLER, TIMOTHY C. To Speech Communications Group, Sperry-Univac, Univac Park, P. O. Box 3525, St. Paul, Minnesota 55101. From SDC.

ROOCH, MELVIN R., MAJOR. USDAO, Box 2, American Embassy, APO San Francisco 96262. Chinese and Japanese characters Artificial intelligence.

SAGER, NAOMI. NYU Linguistic String Project, Warren Weaver Hall, 251 Mercer Street, New York 10012. (The Project has moved to quarters in the building tne the Courant Institute of Mathematical Sciences.)

SALTON, GERARD. ASIS Award for best information science book of 1975: Dynamic Information and Library Processing.

SOMERVILLE, CRAIG A. To 4 Sherry Road, Troy, New York 12180. From Birmingham, MI.

WILKS, YORICK. To Department of Artificial Intelligence, University of Edinburgh, Scotland. From Fondazione Dalle Molle, Castagnola, Switzerland.

NATIONAL SCIENCE FOUNDATION REORGANIZATION

NSF is reorganizing itself into six directorates; the new arrangements are intended to be complete by October 31, 1975:

Division of Mathematical, Physical, and Engineering Sciences

Dr. Edward Creutz, Assistant Director

Division of Astronomical, Earth, and Ocean Sciences

Dr. Robert E. Hughes, Assistant Director

Division of Biological and Social Sciences

Dr. Richard C. Atkinson, Acting Assistant Director

Division of Science Education

Division of Research Applications

Dr. Alfred J. Eggers, Jr., Assistant Director

Division of Scientific, Technological and International Affairs

Dr. Robert E. Hughes, Assistant Director (acting)

Division of Administration

Mr. Eldon T. Taylor, Assistant Director

LATSEC SHOWS MT IN ZURICH

A forthcoming report in the ALLC Bulletin written by Herbert Bruderer describes a demonstration conducted by Peter P. Toma of LATSEC, Inc., in the Institut für Informatik of the University of Zurich. The report describes the system as having been proved practical by the American Air Force since 1970, and says that the Systran system was used for the Apollo-Soyuz flight.

Mr. Bruderer writes that the text was furnished by the Grenoble MT group; he describes the program as requiring a large computer. According to his report, a Russian text of 30,000 words was translated into English in 6 minutes on an IBM 370/155.

XIITH INTERNATIONAL CONGRESS OF LINGUISTS

VIENNA

AUGUST 29 - SEPTEMBER 2, 1977

President of the Organizing Committee is Professor Wolfgang U. Dressler; Secretary General, Dr. Oskar E. Pfeiffer; Secretary, Eva M. Schaup.

Plenary sessions: Basic problems of semantics
Language and society
Logically-based syntax versus autonomous generative syntax
Word formation
Diachrony: reconstruction
History of linguistics: aims and methods

Round-tables: Linguistics as an empirical science
Language and music

Sections: All plenary topics; and also
Indo-European studies: wave theory
Textlinguistics: dialogue
Linguistics and the computer
Contributions of dialectology to linguistic theory

MORE

Language and literature
Contrastive linguistics
Grammar and pragmatics of deixis
Semiotics of human and animal language
The interplay between diachronic and synchronic phonology

Working groups: Speech acts
Speech errors slips of the tongue')
Lexicography
Other topics to be arranged

Registration: Send name, address, institution, name of any accompanying person, etc , with the fee of AS 750 to the Secretariat, Congress of Linguists, Postfach 35, A-1095 Wien, Austria. (The congress account is No. 65-28715 Creditanstalt-Bankverein Vienna, branch Schubertring.) After May 1976, the fee is AS 900; at the Congress, AS 1000. Students half price.

Contributions: Send title and short summary of a 15-minute lecture (for a section) or short statement (for a working group

The second circular will be sent in May 1976 to registered participants.

EMPLOYMENT REGISTER

ACM Computer Science Conference
Disneyland Hotel, Anaheim, California
February 10-12, 1976

Applicants and positions available will be listed in books open for consultation by any conference participant. Staff will operate a message desk and maintain employer sign-up sheets at the conference to facilitate making contacts.

Applicants and employers must register in advance. The charge is $5 for applicants (free to students; $5 additional for an anonymous listing) and $20 for employers. Deadline for submission is January 20, 1976.

Forms can be obtained from

Orrin E. Taulbee
Computer Science Employment Register
Department of Computer Science
University of Pittsburgh
Pittsburgh, Pennsylvania 15260

COMPCON76

TWELFTH IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE

JACK TAR HOTEL
SAN FRANCISCO, CALIFORNIA
FEBRUARY 24-26, 1976

MICROPROCESSOR TUTORIAL: FEBRUARY 23

KEYNOTE SPEAKER: EDWARD E. DAVID, JR.
HISTORICAL SPEAKER: J. PRESPER ECKERT

SESSION TITLES

The personal computer: dream or reality? - What will be the social impact of computers? - Is 'distributed computer systems just a buzzword? - When will design automation come of age? - What will microprocessor hardware evolve into? - What is the futu of language directed machines? - Where is the money coming from? - What will happen with computer networks? - What can we expect in data communications techniques? - What will win the solid state memory race? - What's going on in the rest of the world? Computer science: is it related to computing? - What can we expect in microprocessor software? - Will software engineering get us good software? - What can we expect in hardware design techniques?

SHORT CONTRIBUTIONS: DEADLINE JANUARY 1, 1976

5-minute presentations of recent work or experience can still be offered. A 200 to 300 word summary, with name and affiliation, goes to Martin Graham, Department of Electrical Engineering and Computer Science, University of California, Berkeley 94720.

FOR A PROGRAM, WRITE TO: SIDNEY FERNBACH, COMPUTER DEPARTMENT L-61, LAWRENCE LIVERMORE LABORATORY, PO BOX 808, LIVERMORE 94550.

THIRD EUROPEAN MEETING ON

CYBERNETICS AND SYSTEMS RESEARCH

UNIVERSITY OF VIENNA

APRIL 20-23, 1976

LIST OF SYMPOSIA ON NEXT FRAME

CONTRIBUTIONS

Deadline: January 1, 1976

Form: Abstract - one A4 page, including full title of the paper, author's name and affiliation

Full paper - 3 to 4,000 words on A4 paper

Three copies of abstract and paper required

FORMAT NOTES

Full text double-spaced; abstract may be single-spaced. Typed symbols only; if handwritten symbols must be used, author is to specify precise shape and meaning on a separate sheet Photographs to be on hard glossy paper; at least one copy of any drawing, map, or diagram must be in India ink on hard paper. (The announcement says that abstracts of papers should be received by January 1, but speaks of using full text for refereeing; AJCL cannot resolve the conflict.) Advance notice of films or slides, noting size and quantity, is needed.

ADDRESS

Osterreichische Studiengesellschaft für Kybernetik

Schottengasse 3 A-1010 Wien 1. Austria

SYMPOSIA

General systems methodology

G. Klir, USA

Biocybernetics and theoretical neurobiology

L. Ricciardi, Italy

Cybernetics of cognition and learning

G. Pask, UK

Structure and dynamics of socio-economic systems

K. A. Hammeed, UK

Health-care systems

J. Milsum, Canada

Cybernetics in organization and management

F. deP. Hanika, UK

Engineering systems methodology

F. Pichler, Austria

Computer simulation methods and languages

G. Chroust, Austria, and J. P. C. Kleijnen, Netherlands

Computer linguistics

W. Dressler, Austria

Computer performance control and evaluation

N. Rozsenich and L. Heinrich, Austria

Fuzzy mathematics and fuzzy systems

H.-Z. Zimmermann, BRD

FEE

AS 1500, participants; AS 750, contributors

Association for Literary and Linguistic Computing

International Meeting : Annual General Meeting

December 13, 1975 - University of Amsterdam

PROGRAM

COMPUTER CONTROLLED SAMPLING FOR BILANGUAGE DICTIONARY COMPILATION

R. D. Bathurst (United Kingdom)

LES ACTIVITES DU LEXIQUE INTELLECTUEL EUROPEEN--TRAITEMENT ELECTRONIQUE DES TEXTES D AUTEURS

T Gregory (Italy)

ON LEXICOGRAPHICAL COMPUTING--SOME ASPECTS OF THE WORK FOR A MEXICAN SPANISH DICTIONARY

M. Alinei (Netherlands)

THE RECOGNITION OF FINITE VERBS IN FRENCH TEXTS

J. S. Petöfi (West Germany)

ADDRESS

A. van Wijngaarden (Netherlands)

Inquiries: Mrs. J. M. Smith, 6 Sevenoaks Avenue, Heaton Moor, Stockport, Cheshire SK4 4AW, England.

INTERNATIONAL CONGRESS OF THE

CHARLES S. PEIRCE SOCIETY

STUTTGART, WEST GERMANY

JUNE 16-19, 1976

PRAGMATISM AND SEMIOTIC

About eight invited papers in plenary sessions and volunteered papers in concurrent sessions.

Submissions and requests for information can be addressed to

Professor Carolyn Eisele
215 East 68th Street, Apt. 27E
New York, New York 10021

NATIONAL ASSOCIATION OF USERS OF

COMPUTER APPLICATIONS TO LEARNING

1975 CONFERENCE

HOLIDAY INN, OTTAWA CENTRE

OTTAWA, ONTARIO, CANADA

NOVEMBER 20-22, 1975

PROGRAM TOPICS

- CURRENT EVENTS IN CAI
- TEACHING A CAI AUTHOR LANGUAGE
- CAL DELIVERY SYSTEM
- DISTRIBUTED APPROACH TO CAL
- GUIDANCE INFORMATION RETRIEVAL
- TUTORIAL TECHNIQUES
- USING PLATO
- IMPACT OF EDUCATIONAL COMPUTING
- SYSTEMS APPROACH TO INSTRUCTION
- COMPUTER TECHNOLOGY FOR ELEMENTARY SCHOOLS
- COMPUTER BASED TESTING

ADDRESS

NAUCAL Registration Chairman
Algonquin College
1385 Woodroffe Avenue
Ottawa, Ontario K2G 1V8
Canada

SECOND INTERNATIONAL CONFERENCE ON

HISTORICAL LINGUISTICS

UNIVERSITY OF ARIZONA
TUCSON
JANUARY 12-16, 1976

Registration deadline:	December 26, 1975
Fees:	$112 private room
	$ 77 twin bed
	$ 73 each of two in double bed

Fee include room, ground transport, and conference activities.

Address:	William M. Christie
	Department of English
	University of Arizona
	Tucson 85721

MEETING BRIEFS

SHORT ANNOUNCEMENTS AND REMINDERS

COLING 76

The 1976 International Conference on Computational Linguistics will be held at Ottawa, June 28 - July 2, 1976. Details on Microfiche 17; registration form on Card 63. Deadline for abstracts of contributions: December 1.

1976 N C C

National Computer Conference, New York, June 7-10, 1976. Call for papers on Card 62; deadline January 5. Registration $60; write to AFIPS, 210 Summit Avenue, Montvale, New Jersey 07645.

LINGUISTIC SOCIETY OF AMERICA

Hyatt Regency, San Francisco, December 27-30, 1975. A session on computational linguistics is in preparation.

INFORMATION THEORY

1976 International Symposium, Ronneby, Sweden, June 21-24, 1976 Information from Jack Salz, Bell Laboratories, Room 1G-509, Holmdel, New Jersey 07733.

DIGITAL COMMUNICATION

1976 Zurich Seminar, March 9-11. Information from A. Kundig, Technisches Zentrum PTT/V907, CH-3000 Bern 29, Switzerland.

M O R E

APL 76

Fundamental issues and techniques in practical use of the APL programming language. Ottawa, September 22-24, 1976. Information from C. A. Wogrin, University Computing Center, University of Massachusetts, Amherst 01002.

PROGRAMMING SMALL PROCESSORS

SIGMINI and SIGPLAN. Delta Towers Hotel, New Orleans, March 4-6, 1976. Information from Lawrence J. Schutte, Room 6B-302, Bell Telephone Laboratories, Naperville, Illinois 60540.

PATTERN RECOGNITION

Third International Joint Conference, Coronado, California, November 8-11, 1976. Information from Allen Klinger, UCLA, 3531-C Boelter Hall, Los Angeles 90024. Deadline for submission: March 1, 1976.

NATIONAL FEDERATION OF ABSTRACTING AND INDEXING SERVICES

Information--Dilemmas, Decisions, Directions. Theme sessions on such topics as changing patterns of primary sources, document delivery, user education, indexing systems and current research related to abstracting and indexing. Program Chairman is John E. Creps, Jr., Engineering Index, Inc. Christopher Inn, Columbus Ohio, March 9-10, 1976.

COMPUTERS IN THE UNDERGRADUATE CURRICULA

Binghamton, New York, June 14-16, 1976. Papers invited on actual experience: concrete results, specific materials, problems, programs, and measures of success or accomplishment. Deadline for full papers: January 15, 1976. Information from CCUC/7, Computer Center, SUNY, Binghamton 13901.

American Council of Learned Societies - Grants

SOVIET STUDIES

Grants will be offered subject to refunding, for research in the social sciences and humanities relating to Revolutionary Russia and the USSR; sponsored jointly by the ACLS and the Social Science Research Council. Emphasis is placed on interdisciplinary studies and on applications which bring to Soviet studies insights of sociology, social psychology, cultural anthropology, economics, law, and geography. Only in exceptional cases will grants be made in support of travel for brief visits abroad or to relieve scholars of the necessity of teaching beyond the conventional academic year. Grants will rarely exceed $8,500. The Ph.D. or its equivalent is required. Deadline: December 31, 1975.

EAST EUROPEAN LANGUAGES

Grants will be offered subject to refunding, for scholars and graduate students (who have completed at least one year of graduate study at the program deadline) for intensive, intermediate level study abroad of the languages of Albania, Bulgaria, Czechoslovakia, Greece, Hungary, Poland, Romania, and Yugoslavia; sponsored jointly by ACLS and SSRC. Grants are also offered for enrollment in language courses on all levels in the US if such courses are not available in the regular program of the home institution. $300 to $1000. Deadline: February 1, 1976.

SPECIAL LIBRARIES ASSOCIATION SCHOLARSHIP PROGRAM 1976/77

Three $2,500 scholarships will be awarded by Special Libraries Association for the academic year 1976/77 The awards, to be granted in May 1976, are for graduate study leading to a master's degree at a recognized school of library or information science in the United States or Canada Preference will be given to those applicants interested in pursuing a career in special librarianship. Awards are made without regard to race, sex, age, religion, or ethnic background.

Special Libraries provide research and information services to business, industry and government. *Special Librarians* are men and women trained in the theory and practice of library or information science as well as in the fundamentals of a particular subject field. Specialists are needed in many organizations, among which are research institutes, newspapers, insurance companies, banks, law firms, hospitals, and governmental agencies. Subject specializations may include the social sciences, economics, the fine arts, engineering, and the physical and biological sciences.

ELIGIBILITY: College graduates or college seniors with an interest in special librarianship. Work experience in a special library is helpful

Citizens of the United States or Canada

QUALIFICATIONS: Definite interest and aptitude for special library work

Good academic record

Financial need.

APPLICATIONS: May be requested by writing to

Special Libraries Association
Scholarship Committee
235 Park Avenue South
New York, N.Y. 10003

Applications must be completed and returned by January 15, 1976

ACKNOWLEDGMENT

The following text was inadvertently omitted from the revised manuscript submitted by Allen Klinger and published in AJCL, Microfiche 21, 2-25:

This research was sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR-72-2384. The United States Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation hereon.

The author would like to express his appreciation for this support.

In addition, the author is also grateful for the partial support of Defense Advanced Research Projects Agency. Work presented here began under Contract No. DAHC15-73-C-0181.

American Journal of Computational Linguistics *Microfiche 29 : 25*

COMPUTER GENERATION OF SENTENCES BY SYSTEMIC GRAMMAR

JOHN SELF

Department of Information Science
University of Melbourne
Parkville, Victoria, Australia 3052

ABSTRACT

The paper describes a computer model of systemic grammar, a generative grammar for natural language. A program is explained which given the features of an item, determines the structure of that item according to a systemic grammar specified as data. The program thus demonstrates the principles of systemic grammar, a brief summary of the mechanics of which is also included. Some implications of the program for systemic grammar itself are discussed. In particular, it is shown that previous definitions of the operation of structure-building rules require modification.

1. Introduction

This paper describes a computer model of systemic grammar, a grammar for natural languages developed by Halliday and colleagues at University College, London (Halliday, 1961, 1970). Systemic grammar has recently been of interest to computational grammarians, primarily as a result of the impressive work of Winograd (1972), who developed a natural language understanding system one component of which was strongly influenced by the principles of systemic grammar. More recently, Power (1974) has also investigated how systemic grammar can be used to analyse natural language. There have, however, been no attempts to use a computer to investigate systemic grammar itself. As Friedman (1971) says, in introducing her computer model of transformational grammar, adequate natural language grammars are bound to be so complex that some mechanical aid in investigating their properties will be mandatory.

The aims, then, of developing a computer model of systemic grammar are threefold. First, the model enables the grammar to be tested, i.e. it enables contradictions, ambiguities and incompletenesses in the grammar to be found. Secondly, the model enables systemic grammar itself to be improved, since the consequences of adjusting parameters and rules can be more easily followed. And, thirdly, the model serves as a demonstration of how systemic grammar 'works'

Earlier descriptions of systemic grammar were somewhat incomplete, but that of Hudson (1971) seems sufficiently precise to encourage the feeling that a computer program could be based upon it. The program described below generates (in the linguistic sense) natural language sentences, i.e. "assigns structural descriptions to sentences" (Chomsky, 1965). It is not concerned directly with understanding or producing sentences.

2. The Mechanics of Systemic Grammar

This section briefly describes the generative apparatus of systemic grammar - for a fuller discussion, and for linguistic justifications of the processes, the reader is referred to Hudson (1971), from which the example grammar and generations given later are taken.

In systemic grammar, "structures are entirely predictable from features: given all of an item's features, we can predict exactly what its structure will be" (Hudson, pg 87). In general terms, an item's features or classes are those categories to which it belongs irrespective of the particular sentence to which the item belongs; an item's functions are those categories to which it belongs as a result of its role in a sentence. For example,

"must" has the features MODAL-VERB, FINITE-VERB (among others) and in the sentence

"Must it grow darker?" has the functions !PRE-SUBJECT, !MOOD-FOCUS (among others)
(A preceding ! will be used to distinguish functions from features.) An item's <u>structure</u> is defined by its immediate constituents' functions and the sequence in which they occur.

Given all the features that an item has, the item's structure may be determined, according to systemic grammar, by the sequential application of rules of four kinds:

(1) <u>feature-realisation rules</u>

In the simplest case, these rules are of the form "if item has feature x then its structure will contain function y" - y is said to be the <u>realisation</u> of x. Some rules are conditional in that the realisation only holds if certain other features are or are not present. Also, some rules specify that two functions must be <u>conflated</u>, i.e. both functions apply to the same immediate constituent. (Further details of these and the following rules are given later when the program is discussed.) The application of the feature-realisation rules provides an unordered set of functions, some of which may be conflated.

(2) <u>structure-building rules</u>

These rules expand and order this set of functions to provide the structure of the item. Structure-building rules are themselves of four kinds, which in the simplest case are of the following forms

(a) addition rules: "if function y (or some combination of functions) is present, then so must be function z (possibly conflated with other functions)".

(b) conflation rules: "if some condition expressed in terms of functions is satisfied then some function must be conflated with certain other functions".

(c) sequence rules: "if two functions y and z are present then y must be conflated with, precede or not follow z".

(d) compatibility rules: "functions y and z must not be conflated".

Addition and conflation rules are only applicable if the resultant structure does not conflict with a sequence or compatibility rule. Structure-building rules are not extrinsically ordered in any way. After applying these rules, we have a complete specification of the item's structure, in that we have specified function-"bundles", each of which consists of the functions of one of the immediate constituents of the item.

(3) function-realisation rules

These rules specify which features are implied by an item's functions. They are of the form "if a structure contains function y the corresponding item must have feature x". Applied to the function-bundles obtained from (2), these rules help to determine the features possessed by the immediate constituents.

(4) systems

System networks specify which features are implied by other features. These networks are equivalent to rules of the form "if feature x is present then so is one (or all) of a set of features, and conversely". These rules expand a set of features (possibly the result of applying (3) , not necessarily into a complete set, since some features may be freely selected. The feature-realisation rules may then be recursively applied to this set of features, if required.

3. The Program

The program reads in a definition of a systemic grammar (provided as data so that it may be changed without necessitating major modifications to the program) and generates a structure from a specified list of features. The interested reader should have no difficulty in relating the rules given below to the grammar given by Hudson (pg. 53-101). The rules are shown in the form in which they are presented to the program, and are numbered to ease explanation and understanding of the program's execution. In order to enable the reader to follow the computer generations given later, an English interpretation of selected rules follows:

(1) feature-realisation rules

Rule 13 (below) means "if an item has the feature INTERROGATIVE then, provided it also is DEPENDENT, its structure contains the function !QUESTION (which is thereby introduced if not already present) conflated with !BINDER". Similarly, rule 32

NO	FEATURE	REALISATION	CONDITION
1	(CLAUSE	!PROCESS)	
2	(PHRASE	!HEAD)	
3	(WORD	!STEM)	
4	(INDEPENDENT	-)	
5	(DEPENDENT	!BINDER)	
6	(DEPENDENT	!SUBJECT)	
7	(DEPENDENT	!FINITE)	
8	(IMPERATIVE	-)	
9	(INDICATIVE	!SUBJECT)	
10	(INDICATIVE	!FINITE)	
11	(DECLARATIVE	-)	
12	(INTERROGATIVE	!MOOD-FOCUS	INDEPENDENT)
13	(INTERROGATIVE	(+ !QUESTION = !BINDER)	DEPENDENT)
14	(POLAR	-)	
15	(NON-POLAR	-)	
16	(WH	(+ !QUESTION = !MOOD-FOCUS)	INDEPENDENT)
17	(ALTERNATIVE	!ALTERNATIVE)	
18	(SUBJECT-FOCUS	(!OUESTION = !SUBJECT)	WH)
19	(SUBJECT-FOCUS	(!ALTERNATIVE = !SUBJECT)	ALTERNATIVE)
20	(MODAL	!MODAL)	
21	(NON-SUBJECT-FOCUS	-)	
22	(NON-MODAL	-)	
23	(INTRANSITIVE	-)	
24	(TRANSITIVE	!GOAL)	
25	(TRANSITIVE	!ACTOR	(NOT ACTOR-UNSPECIFIED))
26	(TRANSITIVE	!TRANSITIVE)	
27	(ATTRIBUTIVE	(+ !ATTRIBUANT = !SUBJECT))	
28	(ATTRIBUTIVE	!ATTRIBUTE)	
29	(ATTRIBUTIVE	!COPULAR)	
30	(NON-ATTRIBUTIVE	(+ !ACTOR = !SUBJECT))	
31	(NON-ATTRIBUTIVE	!INTRANS)	
32	(ACTIVE	(!ACTOR = !SUBJECT))	
33	(PASSIVE	(!GOAL = !SUBJECT))	
34	(PASSIVE	!PASSIVE)	
35	(ACTOR-SPECIFIED	(+ !AGENT = !ACTOR))	
36	(ACTOR-UNSPECIFIED	-)	

may conflate !ACTOR and !SUBJECT, but only if both are already present. (There are complications, explained later, when the function to be conflated with, e.g. !BINDER and !SUBJECT above, is absent.)

(2) structure-building rules

(a) Addition rule 2 means "!MODAL and !PASSIVE, if present, must be conflated with !POST-SUBJECT, added if necessary". Rule 1 means "if !MOOD-FOCUS is present but not conflated with !SUBJECT then !PRE-SUBJECT must be added if not already present"

NO	ADDITION RULE	CONDITION
1	((+ !PRE-SUBJECT)	(+ !MOOD-FOCUS # !SUBJECT))
2	((+ !POST-SUBJECT = !MODAL !PASSIVE))	
3	((+ !POST-VERB = !ATTRIBUTE !ACTOR !GOAL))	
4	((+ !EN = !PROCESS)	(+ !PASSIVE))

(b) Conflation rule 1 means "if !MOOD-FOCUS is not conflated with !QUESTION then !PRE-SUBJECT and !MOOD-FOCUS must be conflated, if present".

NO	CONFLATION RULE	CONDITION
1	((!PRE-SUBJECT = !MOOD-FOCUS)	(!MOOD-FOCUS # !QUESTION))
2	((!PROCESS = !COPULAR !TRANSITIVE !INTRANS))	

(c) Sequence rule 1 means "whichever of !MOOD-FOCUS or !BINDER is present, if either, will precede or be conflated with the first of !PRE-SUBJECT and !SUBJECT, if present, which, if both are present, will be in the specified order, and !POST-SUBJECT, if present, will follow the last of these functions, if any, and !PROCESS, if present, will follow or be conflated with the last of these functions if any, and !POST-VERB, if present, will follow the last of these functions, if any".

```
NO  SEQUENCE RULE

1   ((!MOOD-FOCUS OR !BINDER)
      => (!PRE-SUBJECT -> !SUBJECT)
      -> !POST-SUBJECT
      => !PROCESS
      -> !POST-VERB)
2   (!FINITE
      = (!PRE-SUBJECT
         = (!MODAL -> !PASSIVE)
         -> !PROCESS))
```

(d) Compatibility rule 1 means "!POST-SUBJECT must not be conflated with !PRE-SUBJECT"

```
NO  COMPATIBILITY RULE

1   ((!POST-SUBJECT # !PRE-SUBJECT))
2   ((!POST-VERB # !QUESTION))
3   ((!POST-VERB # !BINDER))
4   ((!POST-VERB # !SUBJECT))
```

(3) function-realisation rules

Rule 12 means "if an item has none of the functions !SUBJECT, !GOAL, !ATTRIBUTE or !AGENT then if it has !BINDER, it has the feature CONJUNCTION"

NO	FUNCTION	REALISATION	CONDITION
1	(!COPULAR	COPULAR-VERB)	
2	(!EN	EN-FORM)	
3	(!FINITE	FINITE-VERB)	
4	(!INTRANS	INTRANSITIVE-VERB)	
5	(!MODAL	MODAL-VERB)	
6	(!PASSIVE	BE)	
7	(!PROCESS	LEXICAL-VERB)	
8	(!TRANSITIVE	TRANSITIVE-VERB)	
9	(!AGENT	PREPOSITIONAL)	
10	(!ALTERNATIVE	DISJUNCTIVE)	
11	(!ATTRIBUTIVE	(OR ADJECTIVAL NOMINAL PREPOSITIONAL))	
12	(!BINDER	CONJUNCTION	(NOT (OR !SUBJECT !GOAL !ATTRIBUTE !AGENT)))
13	(!GOAL	(OR NOMINAL DEPENDENT))	
14	(!QUESTION	QUESTIONING)	
15	(!SUBJECT	(OR NOMINAL DEPENDENT))	

(4) <u>systems</u>

Rule 10* means "an item with feature INTRANSITIVE also has one of the features ATTRIBUTIVE and NON-ATTRIBUTIVE, and also has the features naming its supersystems, i.e. 9, 23 and 1, i.e. CLAUSE and ITEM". The *OR in the subsystems column indicates

NO	NAME (IF ANY)	SUPERSYSTEM	SUBSYSTEMS
1	(ITEM	-	(*OR 23 24 15))
23	(CLAUSE	1	(AND 2 9))
2	(-	23	(*OR 3 25))
3	(INDEPENDENT	2	(OR IMPERATIVE 26))
25	(DEPENDENT	2	27)
26	(INDICATIVE	3	27)
27	(-	(OR 25 26)	(AND 4 8))
4	(-	27	(OR DECLARATIVE 5))
5	(INTERROGATIVE	4	(OR POLAR 28))
28	(NON-POLAR	5	(AND 6 7))
6	(-	28	(OR WH ALTERNATIVE))
7	(-	28	(OR SUBJECT-FOCUS NON-SUBJECT-FOCUS))
8	(-	27	(OR MODAL NON-MODAL))
9	(-	23	(OR 10 11))
10	(INTRANSITIVE	9	(OR ATTRIBUTIVE NON-ATTRIBUTIVE))
11	(TRANSITIVE	9	(OR ACTIVE 12))
12	(PASSIVE	11	(OR ACTOR-SPECIFIED ACTOR-UNSPECIFIED))
24	(PHRASE	1	(AND 13 14))
13	(-	24	(OR NOMINAL ADJECTIVAL ADVERBIAL PREPOSITIONAL))
14	(-	24	(*OR NON-QUESTIONING QUESTIONING))
15	(WORD	1	(OR 29 CONJUNCTION))
29	(VERB	15	(AND 16 19))
16	(-	29	(*OR 17 18))
17	(NON-FINITE-VERB	16	(*OR FORM-0 EN-FORM ING-FORM))
18	(FINITE-VERB	16	(OR PAST-VERB PRESENT-VERB))
19	(-	29	(*OR 20 22))
20	(GRAMMATICAL-VERB	19	(*OR 21 MODAL-VERB))
21	(NON-MODAL-VERB	20	(*OR DO BE HAVE))
22	(LEXICAL-VERB	19	(OR COPULAR-VERB INTRANSITIVE-VERB TRANSITIVE-VERB))

* the systems are so numbered to correspond with Hudson's labellings (pg. 71).

that the first named feature or rule is the "default" option, taken unless there are environmental reasons for selecting another.

The rules of the grammar are in fact input and stored in the form of McCarthy lists (McCarthy, 1965), and the program is written in a list-processing extension of BCPL (Self, 1975).

It is important to realise that the rules are not extrinsically ordered in any way, and that the program may (conceptually) execute the rules in any order, with the objective of finding a structure consistent with all rules. Hence, rules are executed recursively, with backtracking when inconsistenices become apparent.

The generation of the structure of a sentence with the features CLAUSE, INDEPENDENT, INDICATIVE, INTERROGATIVE, NON-POLAR, WH, SUBJECT-FOCUS, NON-MODAL, TRANSITIVE, PASSIVE and ACTOR-UNSPECIFIED, e.g. "Which of the tents were errected?" (Hudson, pg. 100), is shown below. Each piece of output is preceded by an indication of the rule that has been executed, e.g. FR 1 indicates the first feature-realisation rule. In the printout of structures,

A
B

indicates that A, B, .. (which may be functions or structures) are conflated. Similarly,

```
->          =>          ?
   A           A           A
   B           B           B
```

indicate, respectively, that A precedes B, that A precedes or is conflated with B, and that the order of A and B is undetermined. When A, B, .. are all functions, then these appear as, e.g. (= A B ..).

```
(GENERATE (CLAUSE INDEPENDENT INDICATIVE INTERROGATIVE
           NON-POLAR WH SUBJECT-FOCUS
           NON-MODAL TRANSITIVE PASSIVE ACTOR-UNSPECIFIED))

FR  1  (!PROCESS)
FR  9  (!SUBJECT !PROCESS)
FR 10  (!FINITE !SUBJECT !PROCESS)
FR 12  (!MOOD-FOCUS !FINITE !SUBJECT !PROCESS)
FR 16  (!FINITE !SUBJECT !PROCESS (= !QUESTION !MOOD-FOCUS))
FR 18  (!FINITE !PROCESS (= !QUESTION !MOOD-FOCUS !SUBJECT))
FR 24  (!GOAL !FINITE !PROCESS (= !QUESTION !MOOD-FOCUS
           !SUBJECT))
FR 26  (!TRANSITIVE !GOAL !FINITE !PROCESS (= !QUESTION
           !MOOD-FOCUS !SUBJECT))
FR 33  (!TRANSITIVE !FINITE !PROCESS (= !GOAL !QUESTION
           !MOOD-FOCUS !SUBJECT))
FR 34  (!PASSIVE !TRANSITIVE !FINITE !PROCESS (= !GOAL
           !QUESTION !MOOD-FOCUS !SUBJECT))

SQ  1
      ?
         !PASSIVE
         !TRANSITIVE
         !FINITE
         =>
            (= !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            !PROCESS
SQ  2
      ?
         !TRANSITIVE
         ->
            ?
               (= !FINITE !PASSIVE)
               (= !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            !PROCESS
```

```
ADDN  2
      ?
         !TRANSITIVE
         ->
            ?
               (= !POST-SUBJECT !FINITE !PASSIVE)
               (= !GOAL !OUESTION !MOOD-FOCUS !SUBJECT)
            !PROCESS
SQ  1
      ?
         !TRANSITIVE
         ->
            (= !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            (= !POST-SUBJECT !FINITE !PASSIVE)
            !PROCESS
ADDN  3
      ?
         !TRANSITIVE
         ->
            (= !POST-VERB !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            (= !POST-SUBJECT !FINITE !PASSIVE)
            !PROCESS
COMP  2
      ?
         !TRANSITIVE
         ->
            (= !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            (= !POST-SUBJECT !FINITE !PASSIVE)
            !PROCESS
ADDN  4
      ?
         !TRANSITIVE
         ->
            (= !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            (= !POST-SUBJECT !FINITE !PASSIVE)
            (= !EN !PROCESS)
CONF  2
      ?
         ->
            (= !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            (= !POST-SUBJECT !FINITE !PASSIVE)
            (= !TRANSITIVE !EN !PROCESS)
ADDN  3
      ?
         ->
            (= !POST-VERB !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            (= !POST-SUBJECT !FINITE !PASSIVE)
            (= !TRANSITIVE !EN !PROCESS)
COMP  2
      ?
         ->
            (= !GOAL !QUESTION !MOOD-FOCUS !SUBJECT)
            (= !POST-SUBJECT !FINITE !PASSIVE)
            (= !TRANSITIVE !EN !PROCESS)
```

```
(!GOAL !QUESTION !MOOD-FOCUS !SUBJECT)

FNR 13  ((OR NOMINAL DEPENDENT))
FNR 14  (QUESTIONING (OR NOMINAL DEPENDENT))
SY 14  (PHRASE QUESTIONING (OR NOMINAL DEPENDENT))
SY  0  (PHRASE QUESTIONING NOMINAL)

(!POST-SUBJECT !FINITE !PASSIVE)

FNR  3  (FINITE-VERB)
FNR  6  (BE FINITE-VERB)
SY 21  (WORD VERB GRAMMATICAL-VERB NON-MODAL-VERB BE
          FINITE-VERB)

(!TRANSITIVE !EN !PROCESS)

FNR  8  (TRANSITIVE-VERB)
FNR  2  (EN-FORM TRANSITIVE-VERB)
FNR  7  (LEXICAL-VERB EN-FORM TRANSITIVE-VERB)
SY 22  (WORD VERB LEXICAL-VERB EN-FORM TRANSITIVE-VERB)
SY 17  (NON-FINITE-VERB WORD VERB LEXICAL-VERB EN-FORM
          TRANSITIVE-VERB)
```

Thus, the structure generated is

```
            "Which of the tents were erected?"
                          |
        ┌─────────────────┼─────────────────┐
   !GOAL            !POST-SUBJECT      !TRANSITIVE
   !QUESTION        !FINITE            !EN
   !MOOD-FOCUS      !PASSIVE           !PROCESS
   !SUBJECT

"Which of the tents   were       :    erected?"
```

The program may then generate the features of the immediate constituents, using the function-realisation rules and systems, and then repeat the above process to determine the structure of the immediate constituents. The first stage of this is indicated above.

Of course, this is a particularly simple sentence and structure designed to make it easy to see what the program does, and it should be clear that considerably more complicated grammars can also be handled. The generative process itself will not usually involve such straightforward intermediate structures or proceed so immediately to the final structure. For example, the generation of the structure of a sentence such as "Must it grow darker?", requiring five 'loops' of the structure-building rules before a structure compatible with all rules is obtained, is as follows:

```
(GENERATE (CLAUSE INDEPENDENT INDICATIVE INTERROGATIVE POLAR
           MODAL INTRANSITIVE ATTRIBUTIVE))

FR  1  (!PROCESS)
FR  9  (!SUBJECT !PROCESS)
FR 10  (!FINITE !SUBJECT !PROCESS)
FR 12  (!MOOD-FOCUS !FINITE !SUBJECT !PROCESS)
FR 20  (!MODAL !MOOD-FOCUS !FINITE !SUBJECT !PROCESS)
FR 27  (!MODAL !MOOD-FOCUS !FINITE !PROCESS (= !ATTRIBUANT
           !SUBJECT))
FR 28  (!ATTRIBUTE !MODAL !MOOD-FOCUS !FINITE !PROCESS
           (= !ATTRIBUANT !SUBJECT))
FR 29  (!COPULAR !ATTRIBUTE !MODAL !MOOD-FOCUS !FINITE
           !PROCESS (= !ATTRIBUANT !SUBJECT))

SQ  1
       ?
          !COPULAR
          !ATTRIBUTE
          !MODAL
          !FINITE
          =>
             !MOOD-FOCUS
             ->
                (= !ATTRIBUANT !SUBJECT)
                !PROCESS
SQ  2
       ?
          !COPULAR
          !ATTRIBUTE
          ->
             ?
                (= !FINITE !MODAL)
                *>
                   !MOOD-FOCUS
                   (= !ATTRIBUANT !SUBJECT)
             !PROCESS
```

```
ADDN  1
      ?
         !PRE-SUBJECT
         !COPULAR
         !ATTRIBUTE
         ->
            ?
               (= !FINITE !MODAL)
               =>
                  !MOOD-FOCUS
                  (= !ATTRIBUANT !SUBJECT)
            !PROCESS
SQ  1
      ?
         !COPULAR
         !ATTRIBUTE
         ->
            ?
               (= !FINITE !MODAL)
               ->
                  (=> !MOOD-FOCUS !PRE-SUBJECT)
                  (= !ATTRIBUANT !SUBJECT)
            !PROCESS
SQ  2
      ?
         !COPULAR
         !ATTRIBUTE
         ->
            =>
               !MOOD-FOCUS
               (= !FINITE !MODAL !PRE-SUBJECT)
            (= !ATTRIBUANT !SUBJECT)
            !PROCESS
ADDN  2
      ?
         !COPULAR
         !ATTRIBUTE
         ->
            =>
               !MOOD-FOCUS
               (= !POST-SUBJECT !FINITE !MODAL !PRE-SUBJECT)
            (= !ATTRIBUANT !SUBJECT)
            !PROCESS
COMP  1
      ?
         !COPULAR
         !ATTRIBUTE
         ->
            =>
               !MOOD-FOCUS
               (= !FINITE !MODAL !PRE-SUBJECT)
            (= !ATTRIBUANT !SUBJECT)
            !PROCESS
```

```
ADDN  3
      ?
        !COPULAR
        ->
           =>
              !MOOD-FOCUS
              (= !FINITE !MODAL !PRE-SUBJECT)
           (= !ATTRIBUANT !SUBJECT)
           !PROCESS
        (= !POST-VERB !ATTRIBUTE)
SQ  1
      ?
        !COPULAR
        ->
           =>
              !MOOD-FOCUS
              (= !FINITE !MODAL !PRE-SUBJECT)
           (= !ATTRIBUANT !SUBJECT)
           !PROCESS
           (= !POST-VERB !ATTRIBUTE)
CONF  1
      ?
        !COPULAR
        ->
           (= !MOOD-FOCUS !FINITE !MODAL !PRE-SUBJECT)
           (= !ATTRIBUANT !SUBJECT)
           !PROCESS
           (= !POST-VERB !ATTRIBUTE)
CONF  2
      ?

           (= !MOOD-FOCUS !FINITE !MODAL !PRE-SUBJECT)
           (= !ATTRIBUANT !SUBJECT)
           (= !COPULAR !PROCESS)
           (= !POST-VERB !ATTRIBUTE)
ADDN  2
      ?
        ->
           (= !POST-SUBJECT !MOOD-FOCUS !FINITE !MODAL
                 !PRE-SUBJECT)
           (= !ATTRIBUANT !SUBJECT)
           (= !COPULAR !PROCESS)
           (= !POST-VERB !ATTRIBUTE)
COMP  1
      ?
        ->
           (= !MOOD-FOCUS !FINITE !MODAL !PRE-SUBJECT)
           (= !ATTRIBUANT !SUBJECT)
           (= !COPULAR !PROCESS)
           (= !POST-VERB !ATTRIBUTE)
```

4. Conclusion

The obvious conclusion - that the mechanics of systemic grammar (as described by Hudson) are sufficiently well-defined to form the basis of a computer model - is, for linguistic descriptions, a significant one. However, the program also demonstrates that some rule-descriptions require clarification. For example, feature-realisation and function-realisation rules are implicitly unordered (since features and functions are unordered), or, more precisely, the rules are to be considered to apply simultaneously. This causes problems with those rules which prevent features being introduced (e.g. rules such as feature-realisation rule 32, which means "if !SUBJECT is present the realisation is as stated, otherwise the first function, i.e. !ACTOR, must not be introduced by any other feature-realisation rule"). The solution seems to be to reapply the rules, recursively, until a structure is produced which is compatible with all the rules.

More seriously, the expectation that the structure obtained is independent of the order of application of structure-building rules is not realised, at least for the grammar specified. For example, considering the second of the above generations and applying rules SQ 1, SQ 2, ADDN 2, SO 1, ADDN 1 (in that order) to the set of functions obtained by the feature-realisation rules, we generate

```
?
   !PRE-SUBJECT
   !COPULAR
   !ATTRIBUTE
   ->
      =>
         !MOOD-FOCUS
         (= !SUBJECT !ATTRIBUANT)
      (= !FINITE !MODAL !POST-SUBJECT)
      !PROCESS
```

This structure cannot satisfy both sequence rules, i.e. the generative process is blocked. Clearly, either the grammar requires modification or it does matter which order the structure-building rules are applied. Hudson (personal communication) has concluded that there are linguistic grounds for ordering structure-building rules, so that 'abnormal' cases precede 'normal' ones, with the latter only applying if the former had not already been applied. Whether it is possible to do sc consistently requires further experimentation.

Further possible extensions to the work could involve trying to specify a lexicon so that the generative process ends up with a structure with words as leaves, and one could also attempt to apply the rules in reverse, i.e. to start with a string of words and produce a structural description. Both problems are, of course, very difficult ones.

Acknowledgement

The author is very grateful to Dr. R.A. Hudson University College, London) for his comments on this work.

References

Chomsky, N. (1965). Aspects of the theory of syntax. Cambridge, Mass.: MIT Press.

Friedman, J. (1971). A computer model of transformational grammar. New York: American Elsevier.

Halliday, M.A.K. (1961). Categories of the theory of gramma . Word 17, 241-292.

Halliday, M.A.K. (1970). Language structure and language function. In J. Lyons (ed.), New horizons in linguistics, London: Pelican.

Hudson, R.A. (1971). English complex sentences. London: North-Holland.

McCarthy, J. (1965). Lisp 1.5 programmer's manual. Cambridge, Mass.: MIT Press.

Power, R. (1974). A computer model of conversation. Ph.D. thesis, University of Edinburgh.

Self, J.A. (1975). CBLP - a computer based learning programming system. J. of. Inst. of Computer Sciences, in press.

Winograd, T. (1972). Understanding natural language. New York: Academic Press.

American Journal of Computational Linguistics *Microfiche 29 : 46*

INTERPRETATION & INTEGRATION OF SENTENCES INTO A C-NET[01]

TH. R. HOFMANN

Groupe d'Etudes pour la Traduction Automatique
Université Scientifique et Médicale de Grenoble

ABSTRACT

A detailed proposal for the representation & manipulation of C-nets (suitable for computer programming) in interpreting pronominal reference. It is shown how this theory accounts for the disambiguation of pronominal reference, & the determination of focus & comment, more completely than any existing semantic or syntactic theory.

The theory of C-nets appears to be the most adequate linguistic theory for semantic analysis of content [my TREES]. We explore here the possibility of automating this analysis to aid in automatic translation. Translation involves analysis of content, without which it can only be a matching of lexical & syntactic structures between languages. Such matching has been shown inadequate by many researchers. Besides being necessary for automatic translation, an automated analysis of content is necessary for other tasks such as constructing general question-answering systems, voice-writers, automatic indexing & abstracting, propaganda measurement & explication, fallacy finding, & others.

In addition to being useful, & in the final analysis, necessary for practical problems in computer understanding or decoding of human language, this theory [see my DESCRIPTIONS] is shown to allow disambiguation of pronouns which are left as ambiguous by contemporary theories of semantics or syntax, but which in fact are not ambiguous. As will be noted but not explored, this theory also allows the consistent determination of the focus of a sentence from its context, i.e. not using position or prosodic features.

The theory of C-nets is in rapid evolution[02]. The version described here is chosen partly because it matches the deep structures obtained by the present English grammar of the TAUM project at the Université de Montréal [TAUM]. In that system, a sentence is reduced to its deep structure by a Q-system[03] grammar. This deep structure would then be converted into a D-net, a temporary C-net, which is then interpreted lexically (i.e. lexical items are replaced by their concepts) & integrated into an overall C-net. This C-net represents the integrated meaning of that & previous sentences. In this theory, successful integration models the comprehension of the sentence, as described in my [C&C]. The interpretation & integration of D-nets are described here, with a demonstration of how later sentences are disambiguated in terms of the comprehension of the earlier ones.

In continuations (a) & (b) below, the syntactic structures are not significantly different. Yet both pronouns <u>she</u> refer unambiguously to different people; to Mary in (a) & to Susie in (b).

Mary told John that Susie was coming,
(a) but she said it softly.
(b) but she didn't arrive for an hour.

Any non-integrative semantic theory is forced to claim that these <u>she</u>'s both refer to the same person, or that they are both ambiguous. Either alternative is wrong; these pronouns refer unambiguously to different people. A sentence often picks up meaning from its linguistic context. The same sentence may express different things in different contexts. The mechanism by which this can happen is explained below.

The only way to determine the correct referents of these pronouns is to take account of the relationships expressed by the verbs <u>told</u>, <u>said</u>, <u>coming</u> & <u>arrive</u>. An adequate way of doing this is to build a C-net for the 1st sentence & then integrate the continuations into it, as outlined in my [INTEGRATIVE].

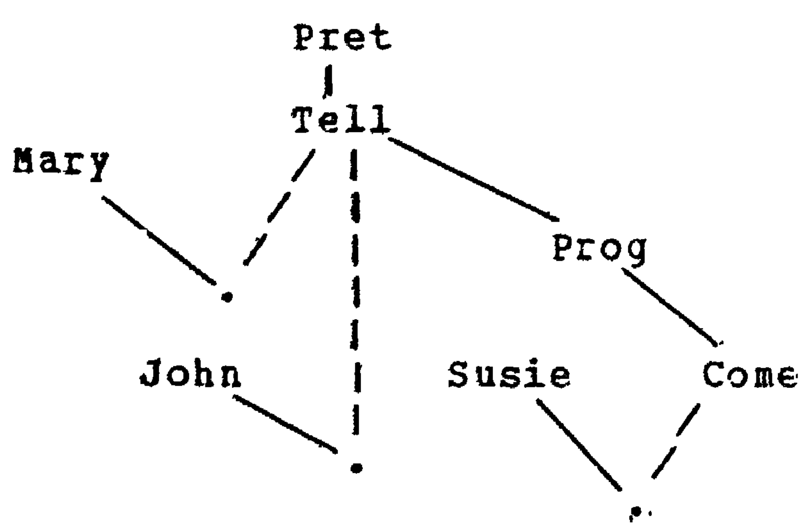
A C-net is a directed, labelled graph with ordered arcs. We can represent a C-net by a list of items, 1 for each node. Each item then consists of: an index number for the node, a label which indicates its meaning, & an ordered list of the indices of the nodes it dominates. The particular index numbers assigned to the nodes are irrelevant, except that "1" & "2" are reserved to designate the speaker & addressee, respectively. Indeed, the index numbers are not part of the C-net. They are needed only to make a linear representation of a C-net, such as is needed for computer manipulation. Also it is customary to capitalize labels of nodes (i.e. semantic atoms), & for simplicity, we take Mary(x) as the meaning of <u>Mary</u>. Thus, a sentence;

Mary told John that Susie was coming.

has a C-net:

3	Mary(4)	8	Come(9)
4	P.	9	P.
5	Tell(4,6,7)	10	Susie(9)
6	P.	11	John(6)
7	Prog(8)	12	Pret(5)[05]

This list of nodes with connections to dominated nodes represents the graph below. In this & following graphs, domination is represented downward as in a dependency tree.



The "referential point" ("P." in the list notation &

"." in the graph notation) is roughly equivalent to the "individual variable" in the predicate calculus & the "referential index" in recent transformational studies. In denotational interpretation, points stand in correspondence with (i.e. they refer to) portions of the universe of interpretation. In the synatx of C-nets, they do not dominate any other predicate (node label).

The above representation[06] is subjected to lexical interpretation rules. These rules explicate concepts (the C-nets of lexical items, see my [C&C]) in terms of their semantic components. They replace single lexical items in a C-net by a small network of unanalysable predicates (semantic atoms). These rules operate successively on a C-net. Each lexical interpretation rule replaces the node(s) at the left by the set of nodes at the right. i, j, k & m are variables ranging over node numbers. A value is assigned to each variable during the operation of a rule, & n-1 is defined as the highest node number in the C-net when the rule applies. All new nodes (n, n+1, &c) which are not dominated by that which dominates i (the node being replaced) are presuppositions[07].

Lexical Interpretation Rules

```
                            ┌ i    ┌Susie┐(j)
i ┌Susie┐(j)          --->  |      └Mary ┘
  └Mary ┘                   | n    Hum(j)
                            └n+1   ♀(j)

                            ┌ i    ┌Harry┐(j)
  ┌Harry┐(j)          --->  |      └John ┘
  └John ┘                   | n    Hum(j)
                            └n+1   ♂(j)

                            ┌ i    ∂(k,n) [08]
                            | n    Say(j,m)
i Tell(j,k,m)        --->   |n+1   Anim(j)
                            └n+2   Hum(j)

                            ┌ i    β(n)
                            | n    ∂(n+1,j)
i Come(j)            --->   |n+1   P.
                            |n+2   Place(n+1)
                            └n+3   ∂(n+1, 1)
```

The effect of the 1st 2 rules is to add the features Hum(x) & ♀(x) or ♂(x) onto the points dominated by the proper names, Susie, Mary, Harry, John. They express the linguistic fact that these names are human names (unlike e.g. Fido), & are used for females & males respectively. In the 3rd rule, the atoms Hum(x) & Anim(x) are included in the meaning of Tell(x,y,z) to restrict collocational possibilities. They thus describe selectional restrictions, & will sometimes disambiguate an otherwise ambiguous sentence.[09] Because the predicates on any point must be non-contradictory, the use of a non-human subject for tell Tell(j,k,l) causes a contradiction around node j. The predicate Place(x) occurs in the 4th rule for the same reason.

There is also a general set of rules by which duplicate nodes are removed;

<u>Duplicate-Reduction Rule Schema</u>

```
i P□()      ┐  --->    i P□()
j P□()      ┘         & j := i

i P□(k)     ┐   --->    i P□(k)
j P□(k)     ┘          & j := i

i P□(k,l) ┐   --->    i P□(k,l)
j P□(k,l) ┘          & j := i

                 &c
```

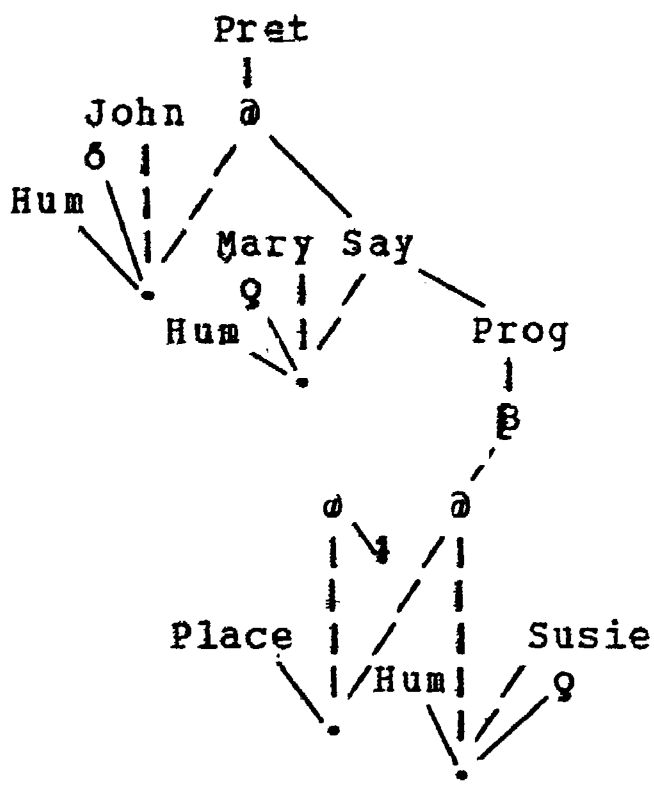
where j := i effects the replacement of all occurrences of j in the C-net by i, & P□ is a variable ranging over node labels.

Most generally, the lexical interpretation rules may be applied in any order & as many times as needed; then this duplicate-reduction rule removes any duplicate nodes introduced by the lexical rules. To our present knowledge, However, it appears that the lexical rules can be constrained to operate only once on a C-net (at all applicable places simulataneously) & this constraint will likely require some ordering between the rules[10]. My working assumptions are that lexical rules can be ordered so that each can apply only once (everywhere) & that the duplicate-reduction rule operates last.

By this process, the 1st graph above for <u>Mary told John that Susie was coming</u> is converted into its final form below. Since there are no previous sentences, there is no integration to be done. This D-net is therefore also the C-net for the discourse up to this poi

3 Mary (4)
4 P.
5 ∂ (6, 15)
6 P.
7 Prog (8)
8 β (16)
9 P.
10 Susie (9)
11 John (6)
12 Pret (5)
13 Hum (4)
14 ♀ (4)
15 Say (4, 7)
16 ∂ (17, 9)
17 P.
18 Place (17)
19 ∂ (17, 1)[11]
20 Hum (9)
21 ♀ (9)
22 Hum (6)
23 δ (6)

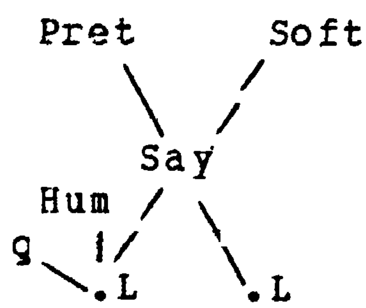
or the graph;



The 1st continuation, (a): but she said it softly yields the D-net below after the lexical interpretation rules & the duplication reduction rule have applied. A D-net is analogous to a deep structure. It represents the meaning of a single sentence prior to integration, i.e, in isolation from the text it is a part of. It is integrated into the existing C-net to form a new C-net.

24 Soft(25)
25 Say(27,26)
26 P. L
27 P. L
28 Hum(27)
29 Q(27)
30 Pret(25)

or;

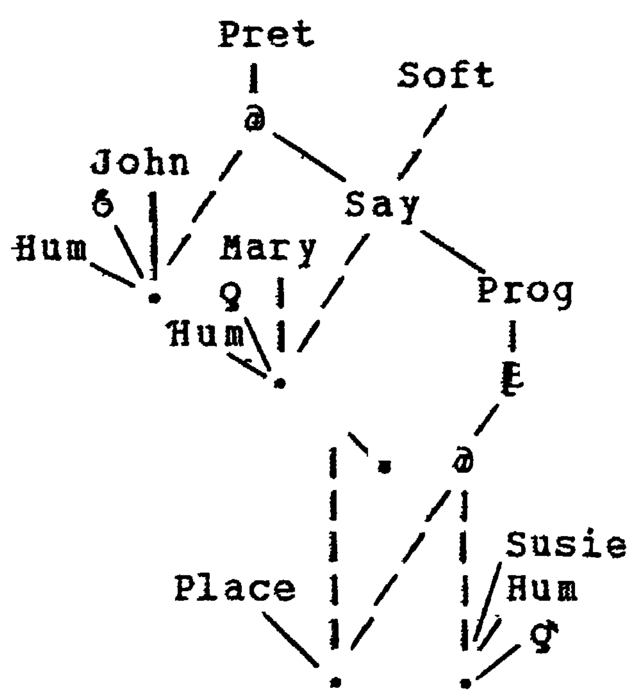


The points in a D-net may be annotated by an "L" for definite reference. This "L" is peculiar to D-nets, & indicates that the addressee should be able to find something it refers to, either in the C-net already existing, or in the context. When integrated, the "L" disappears because the L-point is either identified with a point in the C-net, or it refers to some extra-cognitive structure. This "L" derives from definite expressions such as *the*, *this*, *he*, &c.

This D-net is integrated in the only possible way, namely with

30 := 12
27 := 4 (i.e. she = Mary)
26 := 7 (i.e. it = that Susie is coming)
28 := 13
29 := 14
25 := 15 (i.e. say ≡ tell)
31 := 16

With the duplicates removed, the only addition is 24 Soft(25) [12]. These remaining additions thus obtained necessarily include the focus (comment) & new presuppositions. The "meaning" of this continuation in a loose sense (what information does it transmit which is not already known) is simply the predicate Soft(15). All the rest of the sentence is redundant in the sense that it does not contain anything new. It was necessary, however, to allow the listener to determine what is soft.



The referents of the pronouns are determined as a by-product of the integration. Because this continuation cannot be integrated in any other way, she must refer to Mary, a feminine antecedent which is not the closest one.

There are several possible strategies for integrating a D-net, D, which results from a continuation sentence into the C-net, C, resulting from accretions from all the previous sentences. Basically, some of the nodes in D are found in C, & the D-net is traced out in C. In general, the D-net will contain some nodes not in C, & certain types[13] of the nodes in D may be missing in the C-net. The 1st D nodes to be sought in C should have a high information content to make this procedure more effective & for this reason we have chosen to start with the highest nodes in D: those nodes in D which are not dominated by any other nodes. In the example above, nodes 29, 28, 30, 24, & 31 are all without domination, but 29, 28, & 31 are all directly predicating on points, which leaves 24 & 30 as highest. Nodes with these labels, Soft(▼) & Pret(▼), are sought in C. An equivalent for 24 is not found, but 30 matches 21. Then, following domination lines downward from 30 & 21, everything matches except for 5 @(6,15). @(x,▼) is one of the nodes which can be skipped in integration[14]. If the highest nodes do not yield an integration, the next highest are used until integration is possible. Integration is accomplished by a set of node equivalences such as explained above. Once these are effected, the duplicate reduction rule will remove all the nodes deriving from D which were already in C. The only nodes of D remaining are those which were not already in C.

The pronoun she in this continuation is referentially ambiguous by any method of analysis which handles sentences in isolation. It is rendered unambiguous, but wrongly so, if it is assumed to refer to the nearest preceding feminine antecedent. No English speaker can mistake that its referent is Mary; in reality this <u>she</u> is not ambiguous. But the only way of obtaining the correct referent for it is to utilize the information contained in the verbs <u>say</u> & <u>tell</u>. This cannot be done simply by the similarity in the meanings of these verbs. In a different continuation,

> she wouldn't be able to tell him anything until arriving,

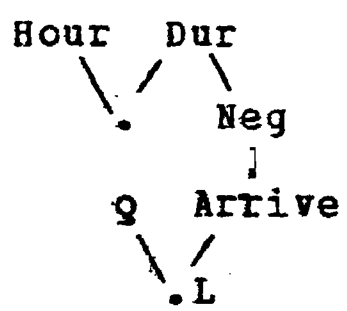
the <u>she</u> no longer refers to Mary but to Susie, even though the main verb of the continuation is identical to the verb of the initial sentence.

In that continuation, or in the less complex one (b):

> she didn't arrive for an hour,

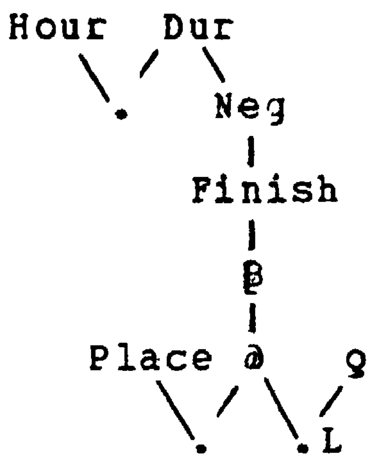
the pronoun refers to Susie, & not to Mary, primarily because of the verb <u>arrive</u> This continuation results in a C-net;

24 Hour(25)
25 P.
26 Durimg(25,27)
27 Neg(28)
28 Arrive(29)
29 P. L
30 Q(29)



This is converted into:

24 Hour(25)
25 P.
26 Dur(25,27)
27 Neg(28)
28 Finish(31)
29 P. L
30 Q(29)
31 β(32)
32 ∂(33,29)
33 P.
34 Place(33)

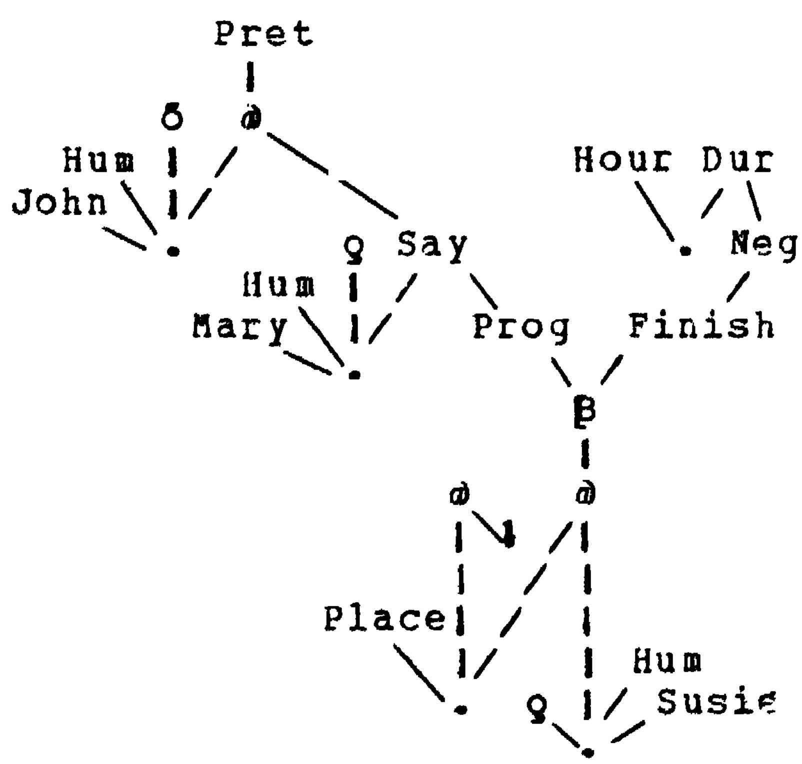


by the lexical conversion rules

```
                          ┌ i    Finish(n)
                          │ n    β(n+1)
i Arrive(j)      --->     │n+1   ∂(n+2,j)
                          │n+2   P.
                          └n+3   Place(n+2)


i ∂(k,m)        ┐  --->   [same as above]
m Arrive(j)     ┘         & n+2 := k
```

It is integrated unambiguously into the C-net as:



with the equations,

29 := 9 (i.e. she = Susie)
33 := 17
30 := 21
31 := 8 (i.e. arfive ≡ come)
32 := 16
34 := 18

When the duplicates are removed, the addition is that Susie's coming was *not finished for an hour*, the italicized portion being the comment.

In the more complex continuation mentioned above, *she wouldn't be able to tell him anything until arriving*, the *she* refers to Susie, even though the subject & the 1st verb significant for disambiguation point to Mary as the referent. This desirable result follows from the fact that the subject of *arrive* is the same point as the subject of *tell*. With the integration strategy used here, both Tell(x,y,Ψ) & Arrive(x) are on the same level in the D-net, but the sub-network dominated by Arrive(x) is a perfect match, while the sub-network around Tell(x,y,Ψ) has no match at all (because of Tell's different 3rd actant). Hence the *she* refers unambiguously to Susie. Interesting enough, if we insert the word *more*, *she wouldn't be able to tell him anything more until she came*, the 1st *she* refers to Mary, & the 2nd to Susie. The cause of this is that the interpretation of *more* inserts an extra Tell(x,y,Ψ) into the D-net, which will match the Tell(x,y,Ψ) in the C-net from the 1st sentence, but the only arrival remains Susie's.

We have given a detailed description of the process by which different continuations of the same sentence have their subject pronouns interpreted as a function of their main verbs. The integration process, which places interpretations on pronouns, also interprets definite occurrences of unmodified & non-specific nouns as repetitions of previously mentioned more specific (or more modified) nouns, fills in deleted nominals on the basis of prior context & interprets generic or unmodified verbs as repetitions of more specific verbs previously mentioned. It must also handle verb phrase deletion, occurrences of do so & gapping. The theory of which this is based is described in my [C&C] & [DESCRIPTIONS & its relation to other semantic theories is discussed in my [TREES], [REPRESENTATION], & in Paillet [PROBLEMES].

The essential contribution of this paper has been a formal description of the process of integration. That process is central to any integrative semantic theory, of which the C-net theory is only 1 (see. my [APPROACH]). The formalization of integration presented here is undoubtedly wrong in some aspects, & requires further research for improvement & verification. As stated here, it is apparently adequate for most cases of pronouns (but see note [11], & excluding deictic uses of pronouns & the pariphrastic it).

The integration of the D-net derived from a sentence is a model of the user's comprehension of the sentence, without, of course, modelling his evaluation of the truth of the sentence or the motive behind its use. These are usually dependent on the universe of interpretation & observation of the speaker. Integration provides a possible next area for research toward fully automated translation because it promises to provide a comprehensive description of the content of the paragraph & the contribution each sentenc makes to that content.

Presently conceived as an adjunct to an automated translation system, (it provides full information as to deletions, anaphora of definite articles & pronouns, & interpretations of words), the C-net could provide the total input into a target language rhetoric. This system will make "versions", translations wherein the content, but not necessarily the words, syntactic constructions or even the order of exposition are preserved.

As opposed to a translation, a "version" cannot be made with present theory, because that reqires an adequate theory of rhetoric; how the material for a sentence is selected out of a complex C-net, what constraints there are for selection of topics, comments, focuses, &c. Much of this is unknown at present, though the next several years may show a great expansion of our knowledge. (See I. Bellert for a direct attack on this problem.)

Ordinary translations is, however, a matching of syntactic & lexical structures <u>as far as possible</u>, without modifying the meaning. C-nets provide a means to do this. With an adequate representation of meaning, syntactic & lexical matching can be done. The result can then be tested for changes in meaning by building a C-net from the target language expressión for comparison with the original C-net. Modification of the target expression can then be made to make the input & output C-nets match to any desired degree of accuracy. A simpler means for good translation is found in my [TRANSLATION].

- - - N O T E S - - -

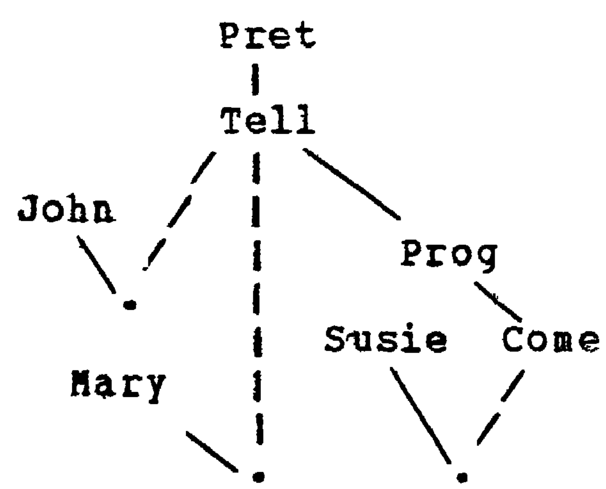
01 This is an extended version of an article "Interpretation & Integration of Sentences into a C-network" which was written at Groupe de recherche sur la traduction automatique, Université de Montréal in the summer of 1971 & appeared in Kittredge [ETUDES]. The terminology & notation has also been revised to be consistent with more recent work on C-nets.

02 Compare, e.g. my [TREES] & [JUDGING].

03 The Q-system is a high level programming language for string manipulation. See A. Colmerauer, 'Les systèmes-Q ou 1 formalisme pour analyser & synthétiser des phrases sur l'ordinateur', in TAUM '71 (1971) Groupe de recherche sur la traduction auotmatique, Université de Montréal.

05 Prog(ψ) & Pret(ψ) are abbreviations for the meanings of the English formatives for the progressive aspect, & the preterite or past tense.

06 In contrast to this, John told Mary that Susie was coming has a C-net,



The difference in the list notation is that node 5 is Tell(6,4,7) instead of Tell(4,6,7).

07 See discussion in my [JUDGING].

08 $\partial(x,y)$ stands for an abstract locative atom of meaning which is not realized exactly by any word in English. French à is closer to $\partial(x,y)$. Recently discovered evidence leads to the belief that Tell(i,j,k) has been incorrectly analysed here, & that this $\partial(x,\psi)$ does not occur at all. A better analysis is "to cause him to come to know it by saying it".

```
i Tell(j,k,m)         --->  ┌ i    ==>(n,n+1)
                            | n    Say(j,m)
                            |n+1   β(n+2)
                            └n+2   Know(k,n)
```

The point of this paper, & the demonstrations remain unchanged under this modification, although most of the C-nets are revised.

09 The TAUM syntax presently attempts this sort of disambiguation by checking the appropriate NP for having the necessary features, as proposed in Chomsky [ASPECTS]. McCawley [ROLE] has argued (p. 132f) that this cannot be done generally by syntactic means, because the critical words may be indefinitely distant, c.f.:

 I will eat whatever Maria bakes.
*I will eat whatever Maria believes.
 I will eat whatever Maria believes Tom to have baked.
*I will eat whatever Maria believes Tom to have dreamed.

10 That is, that the rules can be ordered extrinsically & applied in sequence; after 1 rule has been applied throughout the network, it cannot be reapplied. This will likely require the linguistician to arrange rules in an extrensic order as some rules need the output of others in an extrensic order to work.

11 The point '1' introduced from the analysis of come could now be re-interpreted as the subject (4) or indirect object (6) of Say(4,7). It is not done here for clarity of exposition. With the revision mentioned in note 08, there is a simple generalization to identify the points which '1' may be re-interpreted as. There is a list of predicates (including say, know &c) whose 1st actants may replace a '1' dominated by their 2nd actants.

12 Note that the information by which but contrasts with and is lost in integration, consistent with the principle of integrative semantics that only the information transmitted, not its order nor the speaker's reaction to it, is represented. Conjunctions or successive sentences are simply integrated together: but indicates some surprise or perhaps inappropriateness. If it turns out desirable or even necessary to include such emotive information, semantic atoms for such may be defined & will be predicated on the element which is focussed, by a slightly more sophisticated system. Here, it would result in an additional node Surprise(24). This would allow the reconstruction of the 2-clause syntactic form & the use of but from the integrated network.

[13] It is not yet known which elements may be missing from a D-net without blocking an integration. I assume that there is a finite list of such elements, including the performative atoms marking the type of illocutionary act (statement, question, &c).

[14] i @(j,k) can be substituted for by i=j during integration. See note [13].

--- REFERENCES ---

Kittredge, R. (Ed)
[ETUDES] Etudes en Linguistique Appliquée (1971) Groupe de recherche sur la traduction automatique, Université de Montréal.

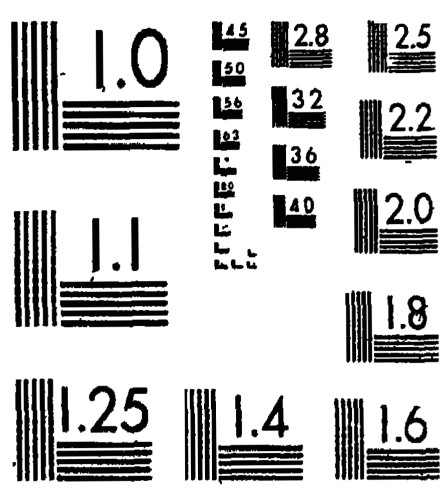
Chomsky, N.
[ASPECTS] Aspects of the theory of syntax (1965) Cambridge (M.I.T. Press).

Hofmann, Th. R.
[APPROACH] 'An integrative semantic approach to intersentential phenomena' (1975) Proceedings of the 1st National Congress on the Application of Computers & Mathematical Models to Linguistics, Sofia.
[C&C] 'Comprehension & concepts in semantics' (1972) unpublished.
[INTEGRATIVE] 'Integrative semantics' (1973) Cahiers Linguistiques 2: 19-38.
[JUDGING] 'Verbs of judging: an exercise in further semantic description' (1973) Cahiers linguistiques d'Ottawa 3: 59-67.
[REPRESENTATION] 'Semantic representations & the theory of language' (1973) Language Sciences 28: 22-24.
[TRANSLATION] 'C-nets in translation of natural language' in Kittredge [ETUDES].
[TREES] 'Meaning doesn't grow on trees' (1973) Language Sciences 26: 1-4.

McCawley, J.
[ROLE] 'The role of semantics in grammar' (1968) in Bach & Harms (Ed) Universals in linguistic theory, New York (Holt).

Paillet, J-P.
[PREREQUIS] 'Prérequis pour l'analyse sémantique' (1973) Cahiers Linguistiques 2: 1-18.
[PROBLEMES] 'Problèmes de notation pour la semantique' (1974) Langages 32: 27-69.

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A