DOCUMENT RESUME

ED 404 795 EC 305 313

AUTHOR	Loritz, Donald; Zambrano, Robert
TITLE	Using Artificial Intelligence To Teach English to Deaf People. Final Report.
INSTITUTION	Gallaudet Univ., Washington, DC.; Georgetown Univ., Washington, D.C. School of Languages and Linguistics.
SPONS AGENCY	Office of Special Education and Rehabilitative Services (ED), Washington, DC.
PUB DATE	31 Jul 90
CONTRACT	H180P80020-89
NOTE	75p.
PUB TYPE	Reports - Descriptive (141)
EDRS PRICE	MF01/PC03 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Software; Deafness; English (Second Language); *Grammar; Grammatical Acceptability; *Hearing Impairments; Higher Education; High Schools; Second Language Learning; Syntax; *Word Processing; *Writing (Composition); Writing Skills
IDENTIFIERS	*Gallaudet College DC

ABSTRACT

This report describes a project to develop an English grammar-checking word processor intended for use by college students with hearing impairments. The project succeeded in its first objective, achievement of 92 percent parsing accuracy across the freely written compositions of college-bound deaf students. The second objective, ability to use the application on affordable microcomputers, was not quite met because adequate system performance required slightly more expensive computers than originally intended. The third objective, to demonstrate the system in the Gallaudet College (Washington, DC) community, was achieved by installation of the program in the college's writing laboratory and the remedial English program. The main body of the report consists of three separate papers. The first is "Computerized Checking of Deaf Students' English Syntax" by Donald Loritz and Robert Zambrano. This paper describes the system requirements and the software, "Ms. Pluralbelle," which, at a student's command, fully parses individual sentences or entire essays. Software evaluation data are included. The second paper, "Computerized Diagnosis of Deaf Students' Syntax" (Donald Loritz and others), describes "ENGPARS," the Pluralbelle parser, designed for checking the English syntax of learners of English as a Second Language. It details and provides diagrams of the program's output of grammar "maps," which diagnose the differential English syntactic competence of learners. The third paper, "Generalized Transition Network Parsing for Language Study," by Donald Loritz describes a generalized transition network system, GPARS, particularly as it has been developed for the instructional parsing of English by students with deafness. (Individual papers contain references.) (DB)

USING ARTIFICIAL INTELLIGENCE TO TEACH ENGLISH TO DEAF PEOPLE

Final Report

LANGUAGE RESEARCH LABORATORIES
School of Languages & Linguistics
Georgetown University
Washington, DC 20057

USING ARTIFICIAL INTELLIGENCE TO TEACH ENGLISH TO DEAF PEOPLE

Final Report

on Grant #H180P80020-89

from

The United States Department of Education

Office of Special Education and Rehabilitative Services

Technology, Educational Media, & Materials
for the Handicapped Program

to

GEORGETOWN UNIVERSITY

SCHOOL OF LANGUAGES AND LINGUISTICS

Donald Loritz, Ed.D.
Principal Investigator

in consortium with

GALLAUDET UNIVERSITY

Robert Zambrano, D.A.
Co-principal Investigator

31 July 1990

CONTENTS

USING ARTIFICIAL INTELLIGENCE
TO
TEACH ENGLISH TO DEAF PEOPLE

Executive Summary

of a proposal completed under grant #H180P80020-89 from the United States Department of Education, Office of Special Education and Rehabilitative Services, Technology, Educational Media, and Materials for the Handicapped Program, to Georgetown University, Donald Loritz, Ed.D., principal investigator, in consortium with Gallaudet University, Robert Zambrano, D.A., co-principal investigator.

Some 1.2 million American children are hearing-impaired. When impairment occurs early in life, the child faces great problems learning the grammar of English. This is a costly national problem in terms of both the waste of human talent and the price of solutions.

In an 18 month project, we developed an English grammar-checking word processor, "Ms Pluralbelle", to alleviate this problem among hearing-impaired students who are beginning their postsecondary education at Gallaudet University, and we demonstrated its use withing the Gallaudet community.

Our first objective was to achieve 92% parsing accuracy across the freely-written compositions of college-bound deaf students. The evaluation presented in Section 1.0 of the Final Report shows that this objective was met.

Our second objective was to achieve this performance on affordable microcomputers, specifically $600 IBM PC clones. This objective was not quite met. At project end, adequate system performance requires $950 IBM AT clones. We believe this still qualifies our system as "affordable", but inasmuch as schools will depend on hand-me-down equipment, it will unfortunately somewhat delay the spread and adoption of the system.

Our third objective was to demonstrate the system in the Gallaudet community. Evaluation of the program in the broader context was disrupted by a campus-wide computer virus in the last semester of the project. Still, this objective has been met In two respects:

(1) the system is available in a user-friendly and disseminable form, as Ms Pluralbelle, Version 2.0. A copy of the system disks is supplied with this report.

(2) the system is now installed and in use at Gallaudet's Northwest (college preparatory) campus, and on the Main Campus in the Gallaudet Writing Laboratory, as well as in the "remedial" English Language Program where system development was conducted.

While the virus disrupted demonstration and evaluation, it allowed laboratory work to proceed more rapidly, producing prototypes of diagnostic analyses of students' writing which are available under Ms. Pluralbelle. This work is reported in Section 2.0 of the Final Report.

Our final objective was to make the system readily disseminable to the deaf community, as well as other language-disadvantaged communities. We had expected to enlist the support of The Lisp Company in this effort. Unfortunately, the president of The Lisp Company and creator of TLC-Lisp suffered a stroke during the project grant period. We have consequently negotiated an agreement with The Lisp Company which makes H.C. Enterprises its agent, and distribution of the system as shareware has begun.

Unless users indicate a willingness to to pay more in exchange for more intensive product support, our philosophy is to keep Ms. Pluralbelle as affordable as possible. Shareware distribution means that the system can be copied at no charge and evaluated by anyone who thinks it might be helpful. If found helpful, the user is encouraged to register his or her copy for $15. Registration entitles the user to the most recent version of the system and basic product support.

Computerized Checking of Deaf Students' English Syntax

Donald Loritz
Department of Linguistics
Georgetown University
Washington, DC 20057, USA
loritz@guvax.bitnet

Robert Zambrano
English Department
Gallaudet University
Washington, DC

Abstract: Ms Pluralbelle is a system for checking the English syntax of ESL learners. The system was particularly developed for and qualified on the English of deaf learners. It requires an MS-DOS microcomputer equivalent to an IBM AT with a hard disk. It presents itself to the student as a word-processor. At a student's command it fully parses individual sentences or entire essays.

Keywords: parsing, syntax, computer-augmented instruction, ATN, augmented transition networks, CALL, computer-assisted language learning, CAI, computer-assisted instruction, IBM PC, MS-DOS.

Donald Loritz [Ed.D., Applied Psycholinguistics, Boston University] has been Assistant Professor of Computational Linguistics at Georgetown University. He has published articles on instructional parser design and instructional digitized speech processing.

Robert Zambrano, [D.A., Catholic Unversity], is Associate Professor of English at Gallaudet University.

Introduction

Nature unquestionably intended for a first language to be learned at a mother's knee, and it would be nice if student-teacher ratios could be lowered so that second languages could be learned in the approximately the same way. But society is seldom as patient with its students as a mother is with her children. Learning a natural language is a slow, difficult, often tedious and, therefore, expensive process.

Second language teaching "methodologies" have been developed to make the second language learning process more cost-effective. Traditional, grammar-based second language instruction has sought to lower the cost of language instruction by enabling the student to self-correct through the application of grammar rules. The central problem with this method has been that it interposes the requirement to learn an intermediary third language, grammar, between the learner's first and second languages. If a computer could be programmed to correct learners' essays, at least at the level of mechanical syntactic correctness, students would not have to learn grammar rules, and much of teachers' valuable time would be saved. Ms. Pluralbelle is an integrated English parser and word processor developed for this purpose.

Language learning is an especially expensive process when deaf children must learn to write a language they have never heard in order to communicate with a hearing world. Ms. Pluralbelle has been particularly designed to meet the needs of deaf students, and, more specifically, those deaf students seeking admission to post-secondary education.

System Specifications and Background.

The Ms. Pluralbelle system runs on IBM AT microcomputers or any compatible machine with a hard disk and 585K of free RAM. Miss Pluralbelle originated as the Apple II parser, Miss Fidditch (Loritz, 1984). When Apple effectively abandoned development of the 85816 RISC microprocessor, Miss Fidditch was ported to TLC-Lisp on the ubiquitous IBM PC, and renamed Mrs. Grundy (Loritz, 1988). Since Mrs. Grundy is a registered trademark of Archie Comics, the final system was ineluctably named Ms. Pluralbelle.

The parser. The Pluralbelle parser is referred to as ENGPARS. ENGPARS is a Generalized Transition Network parser for ENGLISH. ENGPARS is a special case of GPARS. GPARS is a Generalized Transition Network parsing system (GTN). GTNs are derived from the well-known Augmented Transition Network parsing algorithm (Bobrow & Fraser, 1969; Woods, 1972; Bates, 1978; Winograd, 1983), but extended to accommodate a variety of natural languages. Other GPARS systems, analogous to ENGPARS, exist for Russian, Chinese, Japanese, Uzbek, and other languages. GPARS is implemented in GLISP, a dialect of TLC-Lisp86 (John Allen, 1985) and TLC-Lisp386 (Wagner 1989). Educators will recognize Lisp as the parent language of LOGO.

Although one sometimes reads of "ATN grammars", grammatical theory is sometimes also considered to be independent of the ATN/GTN formalism. When useful, we distinguish between computational formalisms and "lambda grammar", the grammatical theory underlying GPARS systems. Lambda grammar is the product of two scientific traditions. The first is Grossberg's Adaptive Resonance models of human cognition (Grossberg 1980, 1986). The second is the past three decades of research in computational grammars (Chomsky, 1957; Fillmore, 1968; Bobrow & Fraser, 1969; Kaplan & Bresnan, 1982; Winograd, 1983). Lambda grammar borrows eclectically from this latter work, but distinguishes itself by rejecting strong claims that serial, computational architectures model human cognitive processes. In particular, lambda grammar asserts that language is learned, rather than acquired, principally through the agency of Peircean abduction implemented at a neuronal level of detail.

The word processor. The Pluralbelle word processor is built on top of the GLISP text editor. The text editor's underlying command set is the WordStar command set, but, because the editor and word processor are also written in Lisp, the command set is highly customizable. Auxiliary functions tend to be mapped to the WordPerfect function key command set. The current standard command set also recognizes the standard IBM cursor-control keys, and these latter are virtually the only keys the student user needs to learn to operate the Pluralbelle word processor.

Student files are maintained as MS-DOS text files. A multi-user version of Ms. Pluralbelle manages student files in discreet subdirectories to provide elementary security where multiple students use a single machine.

A hypertext help system provides context-sensitive help, but will be replaced by a user-directed, browseable help function in subsequent versions.

In this paper we will not discuss the philosophical or grammatical bases of the ENGPARS system further. Rather, we will focus on the integrated Ms. Pluralbelle system, and its past and prospective functions.

In educational settings the Ms. Pluralbelle system can perform at least three functions. First, it can serve as a simple word processor. Second, it can provide diagnostic analyses of student writing. Third, it is a grammar-checker for ESL students. The first of these is by now well-known and researched. The second is promising, but requires technical discussion and further research. It is the last function which we will discuss here.

System Description.

To be useful, a grammar-checking system must be accurate within the linguistic domain for which it is designed. For instructional purposes, the system must also be "student-courteous". In discussing the former criterion, we will refer specifically to "ENGPARS", the Generalized Transition Network parsing component of Ms. Pluralbelle. We will refer to "Ms. Pluralbelle" where primary interest resides in the integrated system and its user interface.

Accuracy within Domain.

To illustrate the accuracy of ENGPARS we created a stratified random sample of 42 student essays of deaf college applicants. The essays had been graded as high-passing, passing, failing, or low-failing by college entrance examiners. The 42 selected essays contained 474 sentences (N_s) which were then parsed by ENGPARS.

These 474 sentences were then parsed by ENGPARS. For comparison, they were also parsed by <u>Gramamtik IV</u>, a well-known style-checking program. After parsing, grammatical sentences passed as grammatical by the parsers, and non-grammatical sentences rejected by the parsers were scored as "hits". Non-grammatical sentences accepted by the parsers were scored as "misses". Grammatical sentences rejected by the parser were scored as "false alarms". By these measures, ENGPARS achieved an accuracy of approximately 90%. However, judgements of grammaticality are sometimes a matter of degree: consider i - vi:

i. ?Colorless green ideas sleep furiously.
ii. ?The King of France is bald.
iii. ?Eins within a space ere wohned a Mookse.
iv. ?John is seeing me next month.
v. ?John was seeing me next month.
vi. *John seed me last month.

Because grammaticality is not categorical, further description of the input sentences is necessary. Column 1 of Listing 1 gives summary ENGPARS output for every 20th sentence in the sample. For comparison, column 2 gives Grammatik IV's analysis of the same sentences.

	ENGPARS	**GRAMMATIK**
a.	He then leaves. OK {H}	He then leaves. OK {H}
b.	It is positively wonderful to see us growing up together. OK {H}	It is positively wonderful to see us growing up together. OK {H}
c.	First, you will get fine bills. OK {H}	First, you will get fine bills. OK {H}
d.	The first reason is that child who uses the drugs. OK {H}	The first reason is that child who uses the drugs. OK {H}
e.	Without a high school diploma and having a job is low chance to get. OK {M}	Without a high school diploma and having a job is low chance to get. OK {M}
f.	What is good to quit if you won't study or learn a thing. OK {M}	What is good to quit if you won't study or learn a thing. OK {M}
g.	*Some teachers # is not intere sting in discuss with students . --> Number conflict: "teachers -- is". {H}	*Some teachers [#Be sure you are using 'is' with a singular subject. ('It is.').] is not interesting in discuss with students. {H}

h.	Sometimes they like to take a nap for a while. OK {H}	*Sometimes they like to take a nap [#Specify how long.] for a while. {H}
i.	*Some don't # . -->Main verb missing. {H}	Some don't. OK {H}
j.	*Jerry is seeing ¦ me next month. -->? {?}	Jerry is seeing me next month. OK {?}
k.	*If not, you could get low # attendance grade and it can pu ll your grades down. -->Try ... "a low" ... {H}	*If not, you could get low attendance grade and it can pull your [#Avoid ending a sentence with a preposition.] grades down. {F}
l.	*Also, I have a close friend who often # invite me to her home for one or two days. -->Number conflict: friend -- invite {H}	Also, I have a close friend who often invite me to her home for one or two days. OK {M}
m.	The two things are to play with my brother and to share our feelings. OK {H}	The two things are to play with my brother and to share our feelings. OK {H}
n.	Many jobs are related to work with people or hands. OK {H}	*Many jobs [#Passive voice: 'are related'. Consider revising using active voice.] are related to work with people or hands. {F}
o.	* # Good group of students teaches together in class. -->Try " A good...". {H}	Good group of students teaches together in class. OK {M}
p.	*They'd want to know what you did and how good ¦ were you in your work experience. -->? {H}	They'd want to know what you did and how good were you in your work experience. OK {M}
q.	*I've been ripped ¦ offf a lot even since I got my license to drive.	*I've [#Passive voice: 'been ripped'. Consider revising using active voice.]

-->Unknown words: offf ripped {H}

been ripped offf a lot even since I got my license to drive. {H}

r.

* And it teachs you how to write term paper, also.
-->"teachs" must end with "-es." {H}

And it teachs you how to write term paper, also.
OK {M}

s.

I have several positive things to say about why it is good ab out ¦ having a sister.
-->Sentence too long. {H}

*I have several positive things to say about why it
[#Be sure you are using 'is' with a singular subject. ('It is.').]
is good about having a sister.
{?}

t.

*I notice the Freshmen and Sophomore students are uncontrolled of how to study and ¦ balance their time.
--> Sentence too long. {H}

*I notice the Freshmen and Sophomore students
[#Passive voice: 'are uncontrolled'. Consider revising using active voice.]
are uncontrolled of how to study and balance their time.
{F}

u.

*The third reason is that pare nts need their child to work to earn money to # support family's need.
--> Sentence too long. {H}

The third reason is that parents need their child to work to earn money to support family's need.
OK {M}

v.

They are tired of doing a lot of homework from the different teachers.
OK {H}

*They
[#Passive voice: 'are tired'. Consider revising using active voice.]
are tired of doing
[#Simplify.]
a lot of homework from the different teachers. {F}

w.

*The bad things about quitting school is very difficult for you ¦ and your parents.
-->Sentence too long.
-->Number conflict: "things -- is". {H}

The bad things about quitting school is very difficult for you and your parents.
OK {M}}

Listing 1. Sample analyses by ENGPARS and Grammatik IV.

Simple, correct sentences (a,b) are easy to analyze correctly, but semantic, diction, and rhetoric errors (c) are extremely difficult--so difficult that we ignore them in scoring parser performance. Suh errors must be left to teachers. Similarly, we find students sometimes produce grammatical sentences by accident (d). It is possible, but unlikely, that (d) occurred in a context where a human editor would have left (d) unchanged. We cannot reasonably expect serial computers to resolve such inter-sentential, semantic errors, and we ignored them in scoring.

Even within reasonable expectations, however, misses do occur. Bizarre sentences (e,f) sometimes find obscure ways of slipping through a program's filters.

It seems natural to refer to programs like Ms. Pluralbelle or _Grammatik_ as "grammar-checkers", but one must be wary of the natural implication that they are also "grammar-correctors". Copmuters are not natural, and these programs are, at best, "error-_detectors_". Both programs detect the error in (g), but problems begin when the programs try to offer corrections. In (g) the _Grammatik_ error message is likely to be unintelligible to many learners. The Pluralbelle strategy illustrated in (w) gives multiple messages, but this can be confusing. Usually, Pluralbelle seeks to give only one error message per sentence as in (g). We will discuss error messages further below.

Although ENGPARS and _Grammatik_ so far look similar, there are deep and fundamental differences between the programs, their philosophies, and the domains to which they are best-suited. Thus ENGPARS simply accepts (h), but _Grammatik_ objects to "for a while" on stylistic grounds. On the other hand, ENGPARS rejects sentences which _Grammatik_ accepts (i,j). For scoring, we gave each parser the benefit of the doubt (i), or half-points (j) in cases where the appropriateness of the system's analysis is questionable. But ENGPARS rejects (i) and (j) because it actually parses its input. That is, it tries to assign a "deep structure" to every sentence. _Grammatik_ basically only scans a sentence for local patterns. In cases like (k-l) these differences become apparent. In (k) we scored the _Grammatik_ analysis as a false alarm because "down" does not function as a preposition in this case, but it could also have been scored as a miss on the article error. Similarly, we think _Grammatik_ is wrong to analyze "are related" as a passive in (n).

Sentences like (o) may appear simple, but they in fact require deep analysis. Thus ENGPARS is able to detect the number conflict within the relative clause in (l), but _Grammatik_ is not.

Neither program professes to be a spelling checker, but ENGPARS does flag unknown words (q). ENGPARS also performs morphological parsing so it is able to detect the error in (r). (In q, _Grammatik_ is given the benefit of the doubt. "Been ripped" _is_ a passive, even though we find its use here quite acceptable.)

A current limitation of ENGPARS is that some sentences are too long and complex for analysis. Ms. Pluralbelle elaborates the terse "Sentence too long" message with the suggestion that the student split the sentence into two or three smaller sentences, and where this might be good advice, we award ENGPARS a hit (s-u). (We do not understand the Grammatik error message in (s). In (t) we think it misleads the student to call "are uncontrolled" passive.)

Another limitation of ENGPARS is that the system only analyzes a sentence up to the first error (although the interactive Pluralbelle interface makes it easy for the student to fix the first error and then reparse the sentence to discover subsequent errors). Only occasionally does ENGPARS find an alternate analysis which allows parsing to continue past the first error (w). In contrast, the Grammatik approach allows multiple errors to be identified within a sentence (although in (v) we again disagree with its passive analysis, and we do not understand the second error message).

Descriptive statistics. The raw scores of hits, misses, and false alarms for ENGPARS are given in Table 1. Table 2 converts the raw scores to rate scores (percentages).

Group	N_e	N_s		Hits		FalseAlarms		Misses
HiPass	9	89	(65)	72.0	(61.5)	15.0	(1.5)	2.0
Pass	12	134		122.5		5.5		6.0
Fail	11	129		118.0		6.5		4.5
LoFail	10	122		112.0		5.5		4.5
Totals	42	474		424.5		32.5		17.0
(Adj)			(450)		(414.0)		(19.0)	

Table 1. Frequency of hits, false alarms, and misses for four groups of students.

Group	N_e	N_s		Hits		FalseAlarms		Misses
HiPass	9	89	(65)	.809	(.947)	.169	(.231)	.022
Pass	12	134		.914		.041		.045
Fail	11	129		.915		.050		.035
LoFail	10	122		.918		.045		.037
Totals	42	474		.896		.069		.036
(Adj)			(450)		(.920)		(.042)	(.038)

Table 2. Percentage of hits, false alarms, and misses for four groups of English learners.

Tables 1 shows that there were approximately 50 ENGPARS parser errors in the corpus. As described in the discussion of Listing 1, a few sentences were scored as "half-hits" or "half-false-alarms", accounting for the half-points in Table 1.

Adjusted scores. As also noted above, the GPARS86 parsing system is limited by the architecture of the Intel 8086 microprocessor. In 8086 machines a "segment" of memory can only be 64K bytes long. This imposes a limit on the length of sentences which can be parsed under the GPARS86 system. The maximum parsable sentence length depends upon a variety of factors, but, in general, sentences over 20 words in length cannot be parsed. In this case, the system simply returns a "sentence-too-long error". (GPARS86 is now being ported to 80386-specific code. When completed, GPARS386 systems will parse sentences of virtually unlimited length).

For the lower three groups, most sentences which were rejected because they were too long were also grammatically incorrect. But the abnormally high false alarm rate among the HiPass students was directly attributable to long-but-correct sentences. Such sentences characterize a level of writing skill at which ENGPARS was expected to lose effectiveness. When the 80386 version of ENGPARS is implemented it is also reasonable to expect these longer sentences to parse nearly as accurately as shorter sentences. When these sentences are removed from the sample, the parenthetical, adjusted values of Tables 1 and 2 are obtained.

Discussion of accuracy results. We measured overall accuracy of the ENGPARS system at 90%. When run against the same corpus, Grammatik IV achieved an accuracy score which was approximately 30% lower, but we do not impute any inferential significance to these figures. The two programs were written for quite different domains, and these must be taken into account. In particular, it should be noted that Grammatik is approximately 60 times faster than Ms. Pluralbelle. It makes little difference if a student must wait 10 seconds or .016 second for a sentence to parse, but few journalists who use Grammatik would commit the errors characteristic of ESL students or have the patience to wait for Ms. Pluralbelle to parse a 5,000-word story.

Courtesy.

As John Higgins has been careful to point out, calling computers "user-friendly" rather debases the meaning of friendship. Instead, computers and computer systems should be respectful of and deferential to students and other users. Our term for this is "user-courteous": Systems should be easy to learn and easy to use. They should neither confuse nor insult the intelligence of the user. In the specific case of Ms. Pluralbelle, we address this general issue under specific issues pertaining to the system interface, fluency, error messages, user help, and user training.

Interface. Ms. Pluralbelle presents itself to the student as a simple word processor, as in Figure 1.

```
Parse: S)entence, C)omposition, P)rint

      *There is big city.  I went to shopping and
  surprised that in the shopping mall.  *It was a
  very small.  *Also there weren't that good to
  sell clothing, shoes.  I also went to resturants.
  *I only have to go a long far away where there
  is a big city.
      I also want to go again.  *We ussually wants
  to find a happy life in a new place.

      Try "a big"...

      Press any key to continue.
```

Figure 1. The Ms. Pluralbelle student interface.

Standard IBM PC cursor-control keys manage cursor movement. There is underlying support of the WordStar command set, but students do not need to use or be aware of these more powerful features. The <F1> key is always used for help. The <Esc> key is always used to exit a subprocess. In Figure 1, it would exit editing of the document and prompt the user to save the file. One backup copy is maintained automatically. In a multi-user version of the system, each student is assigned his or her own directory to avoid accidental overwriting or erasure of other students' files.

Control-p key combinations initiate Parsing and Printing. As illustrated in Figure 1, a student parses a sentence by moving the cursor to the first letter of the sentence and pressing Ctrl-p (^p) followed by <u>s</u>. If an error is detected, a message window pops up on the screen.

^Pc parses the entire composition. In this case, sentences which did not parse are marked with asterisks, and their corresponding error messages are stored on disk. A student may return to the essay at any time and retrieve error messages for a specific sentence by placing the cursor on the sentence's asterisk and pressing <F1> (help).

Fluency and error messages. One presumably does not want beginning students to be corrected for "advanced" errors (e.g., "*If I was ..."). Each student can therefore be assigned a fluency level between 1.0 and 5.9. In theory this will limit error messages to only those errors whose detection would be appropriate to the student's fluency level. In the absence of norms for various fluency levels, we have only assigned students the "intermediate" level 3.5. (But laboratory testing has been conducted at level 5.9.)

Even if only appropriate errors were flagged, there would remain the problem of explaining the errors in a manner the student can understand. For deaf students (as one might also expect for polyglot ESL classes), English error messages and help screens are not particularly informative. In our tests, students frequently spent more time puzzling over error messages than simply hypothesizing and testing alternative English structures. Finally, we simply turned error messages off. By configuring Ms. Pluralbelle with the flag ERRMESSF set to NIL, the system only notes that and where an error occurred in the sentence.

On the other hand, Other groups of students may expect or be better able to benefit from specific error messages, but insofar as turning specific error messages off promotes student hypothesizing and hypothesis testing, there are good psychological grounds for this approach.

User help. If the cursor is not on an asterisk when <F1> is pressed, context-sensitive user help is invoked. The current Ms. Pluralbelle help system is a hypertext system, but in the next version, we will abandon hypertext for a system which is more easily modified by teachers. The new system will allow teachers to change any and all help files, conceivably even completely translating them to the learners' L1.

User training. Without a feasible L1 interface, training our deaf students to use the Pluralbelle system proved to be particularly difficult. In the early stages of development, parser accuracy was only on the order of 80%, frustrating some students. Without a backlog of student essays, we attempted to train students on their own essays, so this frustration was compounded by self-consciousness arising from the necessity of training deaf students in the presence of a (hearing) programmer and an interpreter.

Training is easier now that accuracy has increased, but the use of a set of training essays is still highly recommended because it enables students to achieve autonomy within the Pluralbelle system before their own essays and egos become involved.

Conclusions. Computers can be powerful tools for language learning, teaching, and analysis, but they will never be as good at teaching language as a good human teacher, and heretofore, computers were so expensive that those who could afford them could also afford human teachers. Several recent programs like <u>Grammatik IV</u> have demonstrated how the microcomputer can make computer-assisted language analysis cost-effective. With the declining costs of microcomputers and augmented with artificial-intelligence techniques, systems like Ms. Pluralbelle can be expected to find increasing utility in language learning environments.

REFERENCES

Allen, John. TLC-LISP/86. Redwood Estates, CA: The LISP Company, 1985.

Bates, Madeleine. "The Theory and Practice of Augmented Transition Networks." In Natural Language Communication with Computers, Ed. L. Bolc. New York: Springer Verlag, 1978.

Bobrow, D.G. and B. Fraser. "An Augmented State Transition Network Analysis Procedure. IJCAI, 1 (1969), 557-67.

Chomsky, N. 1957. Syntactic Structures. The Hague: Mouton.

Grossberg, S. 1980. "How Does a Brain Build a Cognitive Code?." Psychological Review 87, 1-51.

Grossberg, S. 1986. "The adaptive self-organization of serial order in behavior: speech, language, and motor control." In Pattern Recognition by Humans and Machines, Ed. E. C. Schwab and Howard C. Nusbaum. Orlando: Academic Press, 1986, 187-295.

Kaplan, R. and J. Bresnan. "Lexical Functional Grammar." In The Mental Representation of Grammatical Relations, Ed. J. Bresnan. Cambridge, MA: The MIT Press, 1982.

Loritz, D. "The Parsing of Learner Sentences." Demonstration presented at the 18th Annual TESOL Convention. Houston, TX, March 1984.

Loritz, D. 1988. The Ms. Grundy parsing system". CALICO 1988 Symposium, Salt Lake City, Utah. February 27.

Wagner, R. TLC-LISP/386. Minneapolis, MN: Seeker Technologies, 1989.

Winograd, T. Language as a Cognitive Process. Vol. 1: Syntax. Reading, MA: Addison-Wesley, 1983.

Woods, W.A. "Transition Network Grammars for Natural Language Analysis." Communications of the ACM, 13 (1970), 591-606.

Computerized Diagnosis of Deaf Students' Syntax

Donald Loritz
Afsar Parhizgar

Department of Linguistics
Georgetown University
Washington, DC 20057, USA
loritz@guvax.bitnet

Robert Zambrano
English Department
Gallaudet University
Washington, DC

Abstract: ENGPARS is a parsing system designed for checking the English syntax of ESL learners. Its output can be displayed in state diagrams or "maps" which diagnose the differential English syntactic competence of learners. 843 sentences from passing and failing writing examinations of deaf college applicants were parsed by ENGPARS. Differential competency maps comparing the passing and failing groups are presented and discussed as a methodology for diagnosis and language teaching.

Keywords: parsing, syntax, computer-augmented instruction, ATN, augmented transition networks, CALL, computer-assisted language learning, CAI, computer-assisted instruction, IBM PC, MS-DOS.

1.0 Introduction

Ms. Pluralbelle (Loritz, 1990a, 1990b) is a English grammar-checking system. Although created for deaf learners of English, it may be useful with other groups of English learners. It has been designed to run on affordable, IBM AT-compatible microcomputers. Ms. Pluralbelle presents itself to the learner as a word processor, and performs exhaustive linguistic parsing in checking students' syntax.

Although Ms. Pluralbelle is designed to be used interactively by students, teachers and researchers can also use its underlying parsing system, ENGPARS, to produce detailed analyses of learners' syntax. One such analysis, a batch mode process which we call "differential syntactic competency mapping", is presented here.

ENGPARS is based upon GPARS, a Generalized Transition Network parsing system. Ms. Pluralbelle, ENGPARS, and GPARS are all implemented in GLISP, a dialect of TLC-Lisp (Allen, 1985; Wagner, 1990).

Section 2.0 of this paper describes the competence mapping method. Section 3.0 presents the resulting maps. Section 4.0

discusses the results as they suggest limitations of and prospects for use of the competency mapping methodology.

2.0 Method: Competence Mapping.

Grammar maps are sets of paths an ATN parser takes through a network. A "complete" map describes a complete grammar. Learners only know part of the complete grammar of their target language, so their maps are "incomplete" or partial. "Completeness" is relative, so diagnostic inferences must be based upon relative, "differential" maps.

2.1 Parse paths.

When ENGPARS parses the sentence "the man runs", one output is the "parse path" of the sentence (Listing 1).

```
        (s/ A)
        (s/ 4)
        (s2/ 8)
        (s/preadv 4
               (np/ B
                  (npk/ D)
                  the__detnil
                  (npk/det D)
                  (npk/quant C)
                  (npk/adjp B)
                  man__malehuman
                  (npk/n1 D)
                  (npk-comp F)
                  (npk/head H)
                  (npk/npk A) )
               (np/2b A)
               (np/nphead G)
               (np/pp 4)
               (np/np 4) )
        (s/preadv4b A)
        (s/topic C)
        (s/gsub E)
        (s/prev C)
        run_s_basicprocv
        (s/v1 D)
        (s/v1/advp A)
        (s/mv L)
        (s-conj B)
        (s/s E)
        fs__puncnil
        (s/s J) )
))))))
```

Listing 1. Parse path of "The man runs".

ATN grammars use state diagrams or "maps". The reader who is not familiar with ATN diagrams is invited to trace the highlighted section of Listing 1 on the first two maps of Section 6.0. Good, standard introductions to ATN grammar are Bates 1978, Winograd 1983, and Allen, 1987.

In parsing the NP "the man", the parser "seeks" an NP by entering the state NP/ (boxed) of the NP network (Map NP, in Section 6.0). Four "arcs", emanate from the "state" NP/ (A-D). Arc A is a "Fail" arc. If the current word could not begin a Noun Phrase (e.g., if it were a finite verb), arc A would cause the NP seek to fail. Arc B is labeled "npk/". Such labels, ending with a slash, conventionally label subnetworks. In this case, arc B instructs the parser to seek a noun phrase kernel: Control is transferred to the NPK/ subnetwork. The NPK/ subnetwork is diagrammed in Map NPK.

In state NPK/, arc A is again a Fail arc. "The" is not a pronoun, so arc B is not taken. Similarly, "the" is not the word "all", so arc C cannot be taken. But "the" is a determiner, so arc D is taken, and control passes to the state NPK/DET. "The" is "consumed", and the current word is advanced to "man".

In state NPK/DET arcs A-C are not satisfied. Arc D is a "jmp" (jump) arc. JMP arcs have few, if any conditions. Here, control passes to the state NPK/QUANT.

Listing 1 tells us that arc C is taken from state NPK/QUANT to state NPK/ADJP. There, "man" is recognized as the head noun on arc B. "Man" is consumed, and control passes to NPK/N1.

The remainder of the NPK/ network is traversed in similar fashion until state NPK/NPK is reached. We have successfully sought a "Noun Phrase Kernel". Arc A is a "send" arc which returns us to the calling network, NP/.

Having successfully sought and found a Noun Phrase Kernel on arc B of state NP/, we are lead to state NP/NPHEAD. The process continues in this manner. The reader may wish to trace the entire path of Listing 1 against the NP and S maps given in Section 6.0.

2.2 Complete partial, and differential grammar maps.

The path traced in parsing Listing 1 describes the syntactic structure of "The man runs". If we parsed thousands of grammatical English sentences and recorded all of their paths, the union set of paths would describe a grammar of English. That is essentially what has been done to produce the ENGPARS grammar. Section 6.0 gives the maps of the resulting "complete" ENGPARS grammar. While it is doubtful that any grammar of English will ever be absolutely "complete", the grammar represented in Section 6.0 has achieved 90% accuracy within the present corpus and is considered relatively complete with respect to the "partial" grammars of English learners.

Learners of a language know only a part of the grammar of a language. Thus, the partial map, Map NP_{part} in Section 7.0 shows the part of the NP grammar used in essays which failed to pass a college entrance writing examination. Comparison with the complete Map NP shows that these students only used a limited smallish subset of the complete NP grammar of English.

Such partial maps give a "perspicuous" view of what aspects of grammar fall within the competence of a given learner or learners, but they fail to tell us if the missing arcs reflect critical patterns of English grammar, or patterns which are simply infrequent. Differential maps, which compare the grammars of two groups of writers, are more diagnostically useful.

2.3 Parse trees.

We note in passing, that parse trees are also available as output from the ENGPARS system. A sample parse tree for the toplevel sentence node of the preceding sentence is given in Listing 2.

```
'((const s/) (xsent t)
      (wf the__detnil)
      (illf t)
      (constwn 1)
      (accscope nil)
      (num init)
      (wf nil)
      (topic np/1)
      (gsub np/1)
      (stype d)
      (tv run_s_basicprocv)
      (actor np/1)
      (mv run_s_basicprocv)
      (mvr run_s_basicprocv)
      (surfargs ((sv sv)))
      (vparticles nil)
      (accscope t)
      (endpunc fs__puncnil) )
)
```

Listing 2. Toplevel parse tree for "The man runs".

Parse trees and parse paths can be engineered to contain equivalent information. In practice, however, parse paths tend to aggregate data while parse trees tend to segregate data. All Noun Phrases will contribute to path representations like Map NP, but tree representations tend to subclassify constituent phrases. For example, the noun phrase "np/1" in Listing 2 has been subclassified as a topic, a grammatical subject, and an actor.

The analyses and maps presented here have not used ENGPARS' parse tree output, but it is important to recognize that such information is readily available to future research.

2.4 Data.

The differential maps in Section 7.0 were automatically produced from essays written by deaf college applicants. The essays were graded as part of a regular college admission process. Fifty-seven essays which met the criteria were parsed by the ENGPARS system. Sixteen essays which failed to meet entrance criteria were also parsed. Parsing the essays yielded parse paths for various well-formed phrases according to Table 1.

Phrase Type	Passing	Failing	Pass/Fail
S/	678	165	4.11
NP/	985	233	4.22
SUB/	40	4	10.00
SCONJ/	47	6	7.83
ADJP/	139	22	6.32
TOCOMP/	154	34	4.52
ADVP/	142	36	3.90
RELC/	39	14	2.29

Table 1. Occurrences of major phrase types in grammatical sentences of passing and failing essays of deaf college applicants.

In Table 1 each embedded and conjoined phrase (or clause) is counted once. Thus the NP "Bill and Sue" counts as two NPs. The rows of Table 1 are also non-exclusive. For example, at least 94 (47 x 2) of the sentence phrases (S/) are found in the 47 conjoined sentences (SCONJ/) of the passing group.

Given that the ratio of passing to failing essays in the sample was 57/16 = 3.56, Table 1 shows marginal grammatical superiority for the passing essays on well-formed sentences, noun phrases, to-complements, and adverb phrases. More marked superiority is seen in the production of grammatical subordinate clauses (SUB/), conjoined sentences (SCONJ/), and adjective phrases (ADJP).

The anomalous result for the relative clauses (RELC/) is an artifact of the low $\underline{n}$ of RELCs and "echoic" constructions: On several assigned topics like "Why it is good to have a brother" the failing essays particularly included echoic sentences like "There are three reasons why it is good to have a brother". In such cases ENGPARS treats "it is good to have a brother" as a relative clause attached to the head "why".

The idea behind differential competency maps is similar to that of Table 1, except that it is more detailed: we compute a diffence measure for every arc of the grammar.

2.5 The ϕ^* difference measure.

Computing a difference measure for each arc of the grammar is complicated by several factors. Rather than ratios or other measures of the magnitude of difference, we would like a statistic that reflects the probability that an observed difference in grammar is significantly different from chance variation. A straightforward statistic would be the calculation of chi-square on the frequency with which any given arc is taken relative to all other arcs leaving the state, as in Table 2.

	Passing	Failing	Marginal Σ
Arc of interest:	p_i	f_i	I
All others:	p_o	f_o	O
Marginal sums:	P	F	

Table 2. 2 x 2 contingency table for calculating X^2 for an arc.

Unfortunately, improvements in grammar are likely to appear in relatively infrequent states, and values of chi-square are highly dependent upon N = P + F. To circumvent this, we calculate ϕ:

$$i. \quad \phi = \left(\frac{\chi^2}{N} \right)^{\frac{1}{2}}$$

ϕ corrects chi-square for N. It assumes the value of 1.0 when all cases fall on a diagonal of the 2 x 2 chi-square table and 0 when the distribution does not depart from the expected proportional distribution. Since we wish to indicate whether a change in grammar is toward or away from greater competence, we also wish ϕ to be signed. "Signed ϕ" is easily computed by ii:

$$ii. \quad \phi' = sign\left(\frac{p_i}{p_o} - \frac{f_i}{f_o} \right) \phi$$

Unfortunately, the chi-square calculated on the model of Table 2 is usually too local to be important. For example, once we get to the state NP/NPCONJ, whether the NP conjunct ends with a comma (arc B) or not (arc A) is of less interest than the fact that the parse got to NP/NPCONJ at all. To obtain a more global measure, we calculate ϕ^* by substituting X^{2*} into i, where X^{2*} is

iii. $\chi^{2*} = \frac{n(p_i f_o - p_o f_i)^2}{F \cdot P \cdot I \cdot O}$, and

$$f_o = 165, \quad p_o = 678$$

that is, the sentence proportion from Table 1 is used as an estimate of the proportion of "other" arcs in Table 2.

In our data (ϕ^*) ranged between +0.88 and -1.04. The differential grammar maps of Section 7.0 were then plotted in double lines for all arcs with $\phi^* > .30$ and in dashed lines for all arcs with $\phi^* < -.30$. Thus, double lines indicate arcs and grammatical features which were markedly more frequent in passing essays, while dashed lines indicate arcs and grammatical features which were markedly more common in failing essays.

3.0 Results.

In this section we comment on the differential maps of Section 7.0. The distinctions shown by double and dashed lines emerge clearly from the maps, but the labels on the maps are necessarily terse and require comment. In Section 7.0, the maps are arranged with the simpler NP/ϕ and NPK/ϕ networks presented first. The large S/ϕ network follows, extending over 5 pages. Then the smaller, minor-phrase subnetworks follow without comment.

The NPs of passing students (Map NPϕ) are "heavier". They show more NP conjuncts (e.g., on arcs NP/NP_B, $NP\text{-}CONJ_E$, and $NP/NPCONJ_A$). They also show more prepositions ($NP/NPHEAD_A$) and prepositional phrases (NP/PP_A). The RELC/ topic effect discussed in connection with Table 1 is reflected in arcs $NP/NPHEAD_E$, NP/PP_E, and $NP\text{-}RELC_B$.

The same tendency toward "heavier" NPs is shown in the Noun Phrase Kernel network (Map NPKϕ) with more adjective phrases registered on arc $NPK/QUANT_A$. From NPK/ADJP, more capital letters are also registered on $NPK/ADJP_A$ and arc NPK/CAP_C indicates that these mostly belonged to "unknown" words -- probably proper nouns which were not found in the ENGPARS lexicon.

Simultaneously with these "heavier" features, there was also a tendency toward greater usage of pronouns (arcs $NPK/_B$, $NPKPRO_A$, and $NPK/HEAD_E$. This could be a reflection of better "cohesion" in the passing essays. Markedly more of these pronouns were also possessives (NPK/PRO_A). In a significant number of cases, however, single quotes apparently did not mark possessives, but rather a contracted verb (e.,g., "He's" or "She'll", arc NPK/QS_B).

On the other hand, the failing essays showed more simple-noun kernels ($NPK/_F$) and deictic elements (NPK/PRO_E). More unknown words ($NPK/_E$, $NPK\text{-}COMP_D$), wh-pronouns ($NPK\text{-}PRO_C$), and gerund heads ($NPK/ADJP_C$) are topic effects.

In Map $S\phi(a)$, we notice that the failing essays used more sentence-initial conjunctions (e.g., "And that's the truth", arc $S/_B$), and more yes-no questions beginning with modals on arc $S/_C$ (e.g. "Can you believe it?"). Failing essays also used proportionately more initial adverb phrases ("Usually I went to school", arc $S2/_C$). The passing essays again showed more contracted verb forms ($S/GSUB_A$), and they showed more sentence-initial subordinated sentences ($S2/_D$).

In Map $S\phi(b)$, the failing essays showed more sentences using simple verb constructions (arcs $S/V1/ADVP_A$ and $S/V1/ADVP_G$). The failing essays use more of the verbs HAVE and BE along several arcs leading to S-ASPV. However these common verbs appear to have been used largely in simple constructions: For example, the failing essays show proportionately more progressive constructions on arc $S/BE/ADVP_C$ and equational copular sentences on arc $S/BE/ADVP_F$. The absence of arc $S/HAVE_B$ (cf. Map S/ in Section 6.0) indicates no usage of simple present or past perfect aspect even in the passing essays, but arc $S/BE/ADVP_A$ indicates a higher incidence of passive constructions in the passing essays. The pattern of simple constructions in the failing essays is repeated in map $S\phi(c)$ on arc S/MV_G where modal verbs (can, will, etc.) function as main verbs. On the other hand, passing essays show more non-progressive BE sentences on arc S/MV_E.

Map $S\phi(d)$ shows the passing essays to have more sentence complements on arcs S/VP_F, $S\text{-}COMP_C$, and $S\text{-}COMP_E$. Finally, Map $S\phi(e)$ shows passing essays to have more conjoined sentences ($S\text{-}CONJ_D$) and more sentence-final subordinate clauses (S/S_C).

4.0 Discussion and prospects for further research.

The kinds of analysis outlined above suggest many opportunities for future research and instructional application, but they also have several limitations.

4.1 Limitations.

Computational error analysis is only marginally feasible. Once a parser encounters an error, the rest of the sentence can no longer be parsed with confidence. As a result, we have undertaken computational <u>competency</u> analysis -- not <u>error</u> analysis. A small measure of confidence can be gained by reverting to bottom-up parsing (Mellish, 1989), but it is unclear if this small measure of confidence is worth the corresponding increase in computational cost and complexity. This limitation upon computational error analysis is finally a fundamental limitation upon Turing machines, and it is inescapable.

All grammars leak. ENGPARS has been developed and qualified against the corpus from which the preceding sample was drawn over the course of nearly two years. In our most recent tests it has achieved accuracy scores in excess of 90%. This suggests, however, that as much as 10% of the parse paths used in computing the maps reported here may contain one or several erroneous arcs. Our experience gives us reasonable assurance that these errors are minor and randomly distributed -- and have therefore not greatly skewed our competency maps. However caution and grammatical tuning will be necessary whenever ENGPARS or any parser is moved to a new domain of input.

Larger samples are needed. Sampling error is a more probable source of major error. In the data reported here, for example, a larger sample would allow us to control for topic variance, student age, student sex, and a host of other potential intervening diagnostic variables. Fortunately, computer parsing makes the analysis of large samples feasible.

Diagnostic parsers must be made user-courteous. In the end, grammars are evaluated not by their explanatory adequacy, nor even by their descriptive adequacy, but by their communicative adequacy. How well does the grammar communicate useful information to the student and to the teacher? ATNs have laid claim to "perspicuity", and we believe the grammar maps we have presented exhibit this virtue. But we authors undoubtedly find the maps more perspicuous because of our intimate familiarity with the megabyte of detailed computer code which lies behind them. Readers without our acquired ability to "read between the arcs" might honestly contest this claim of perspicuity.

The labels on our maps are necessarily short, and the underlying code is necessarily technical, so there is a need to make our grammar more expressive. An important finding of lambda grammar is that there is no one "best" grammar of a language. In the present case, the arcs and states could of our grammar could be extensively rearranged to highlight different features of English grammar. Indeed, Bates 1978 points out that an ATN grammar can be expressed with all arcs defined on one state, if the arcs are highly constrained. But the arrangement of arcs is not decorative. It is intrinsic to the grammar, and every such rearrangement is itself a major research project.

4.2 Opportunities.

Large sample studies are possible. The study of language teaching and learning has been inhibited by the difficulty of obtaining independent measures of L2 proficiency. Past efforts have ranged between broad measures (e.g., T-units, the measures in Table 1), and detailed measures (e.g., "grammatical morpheme" analyses). At either extreme, language research was ultimately hampered by the sheer volume of data which needed to be coded in order to obtain statistically stable analyses. The unreliability inherent in using multiple coders, and the individual tedium of coding natural language data by hand formed a powerful conspiracy against the data-based study of language.

Parser-analyzed data can integrate broad and detailed measures of learner language. Hundreds of variables can be automatically coded with machine-like reliability. The data presented here are based on only a small corpus, but it is eminently reasonable to consider extending this analysis to tens of thousands of essays so that a variety of other variables can be controlled. Fifteen years ago, researchers could never have proposed syntactic analysis on such a scale.

Parser analysis is suited to a wide range of research questions and techniques. Numerous applications of diagnostic parsing are readily imagined as variants of the preceding approach. Where every arc is a variable, routine factor analyses of learners' sentence patterns may reveal and/or classify students' learning styles and preferences. Alternatively, analyses such as the preceding might tell teachers more about the nature of the tasks and tests to which students are set. A particularly interesting variant of this theme involves parsing textbooks to compare their language against the language of students. And, returning to a scale of one, it is intriguing to imagine parser-assisted longitudinal studies.

Diagnostic parsing of language structures larger and smaller than the sentence is possible. We have alluded to dialogue and cohesion analysis, and, within limits, these are possible. The ENGPARS lexicon is organized as a semantic network, so metonymy-based analyses and other standard network-link analyses are feasible. On the other hand, pragmatic analyses which rely on world knowledge will always best be left to the teacher.

Various languages and dialects can be analyzed with common technology. As discussed above, grammatical fine-tuning will always be needed. Languages live and change, and grammar code must, too. More revolutionary revision of the grammar code is needed to create grammars of language variants like child language, dialogue or conversation, and dialect. Insofar as one student's idiosyncracies are another teacher's errors, the development of specialized grammars is a way to approach the problem of performing computer-conducted error analysis. Fortunately, many of the tools used in the ENGPARS system have proved useful in diverse domains. For example, we have already used the GPARS system to construct Chinese and Russian systems similar to ENGPARS.

Microcomputers make parser analysis available at the classroom level. At a classroom or program level, one can imagine teachers assigning new students to groups on the basis of competency map results, selecting reading materials whose parse maps exhibit features the different groups ought to focus on, pairing students for peer teaching whose competency maps are complementary, and evaluating progress against maps of expected competency.

5.0 References.

Allen, James. Natural Language Understanding. Menlo Park, CA: Benjamin/Cummings, 1987.

Allen, John. TLC-LISP/86. Redwood Estates, CA: The LISP Company, 1985.

Bates, Madeleine. "The Theory and Practice of Augmented

Transition Networks." In Natural Language Communication with Computers, Ed. L. Bolc. New York: Springer Verlag, 1978.

Loritz, Donald. 1990a. Ms. Pluralbelle. Arlington, VA: HC Enterprises.

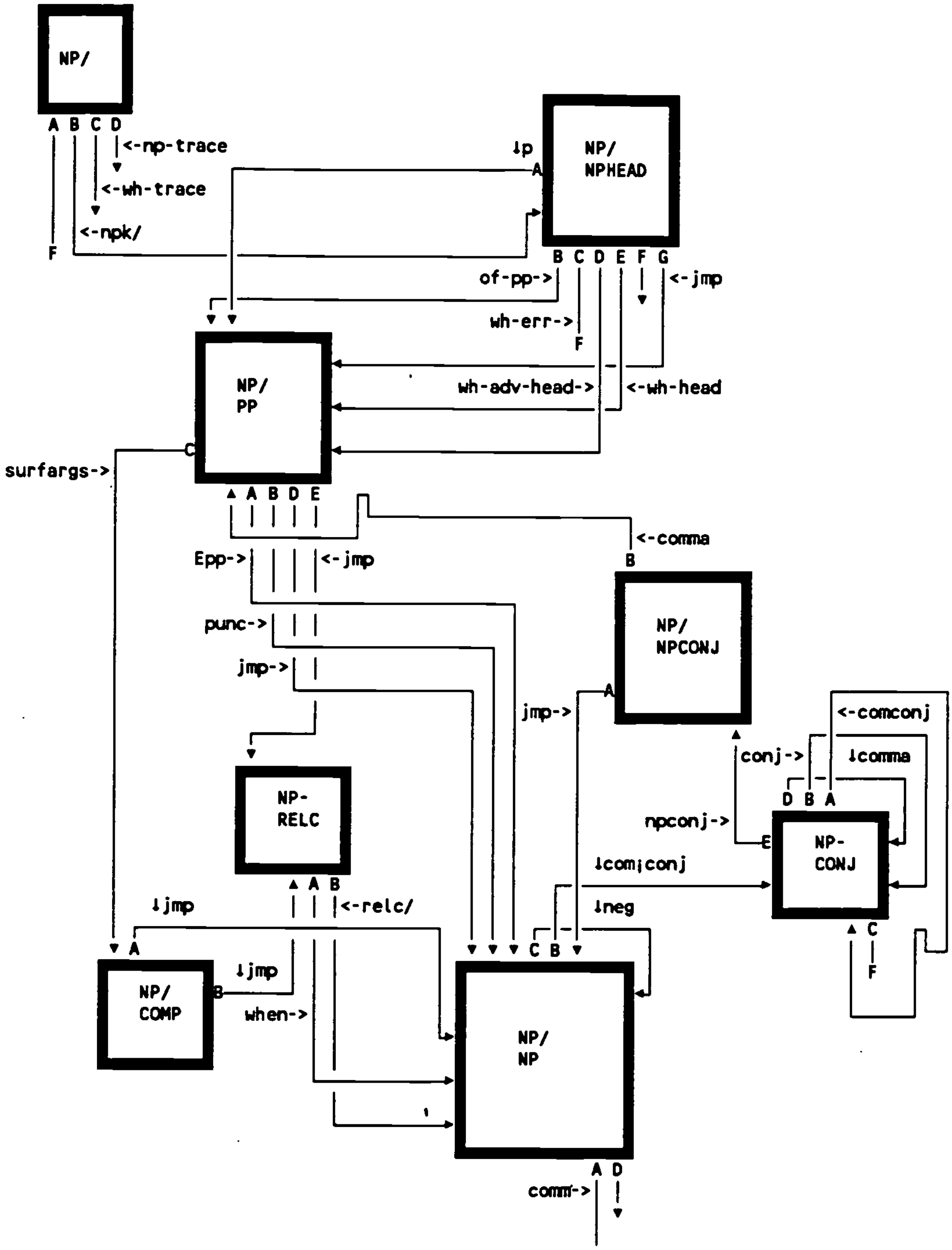
Loritz, Donald. 1990. Computerized checking of deaf students' English syntax. Georgetown Unversity, Language Research Laboratories Report #RX2500-950-1A. Washington, DC.

Mellish, C. S. 1989. "Some Chart-Based Techniques for Parsing Ill-formed Input." Proceedings of the Association for Computational Linguistics.

Wagner, R. TLC-LISP/386. Minneapolis, MN: Seeker Technologies, 1989.

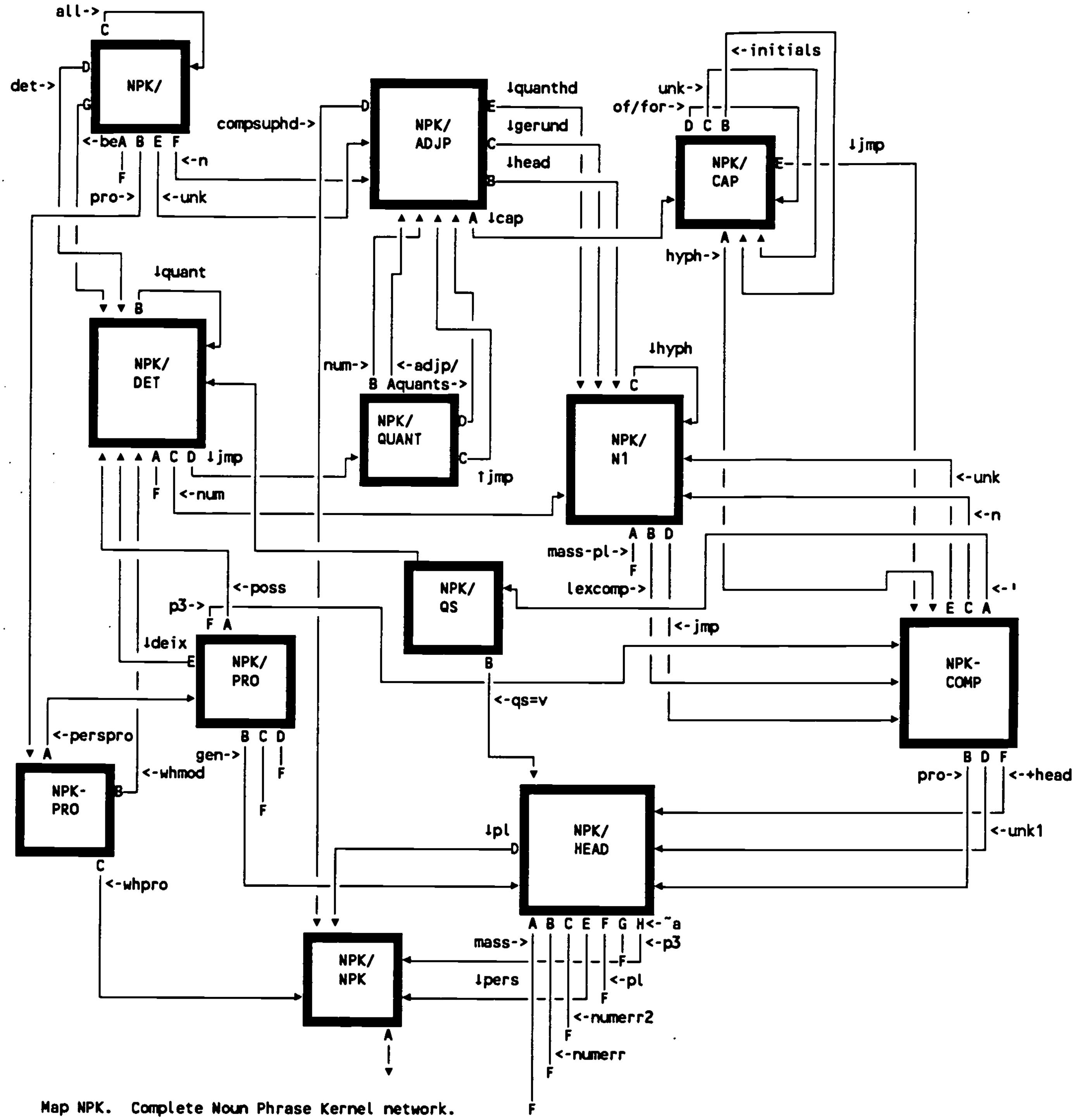
Winograd, T. Language as a Cognitive Process. Vol. 1: Syntax. Reading, MA: Addison-Wesley, 1983.

6.0 Complete Grammar Maps.



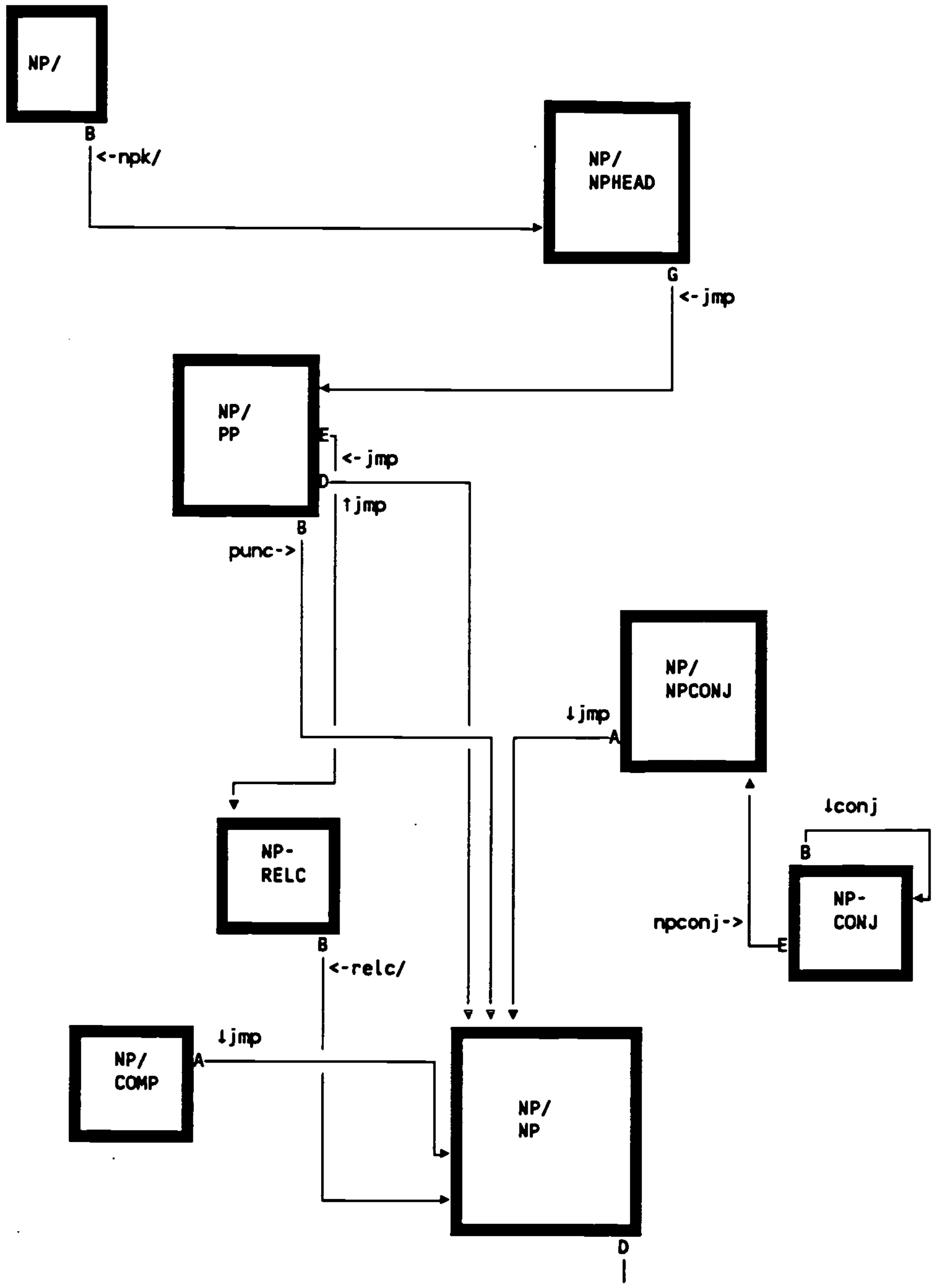
Map NP. Complete Noun Phrase network.

BEST COPY AVAILABLE

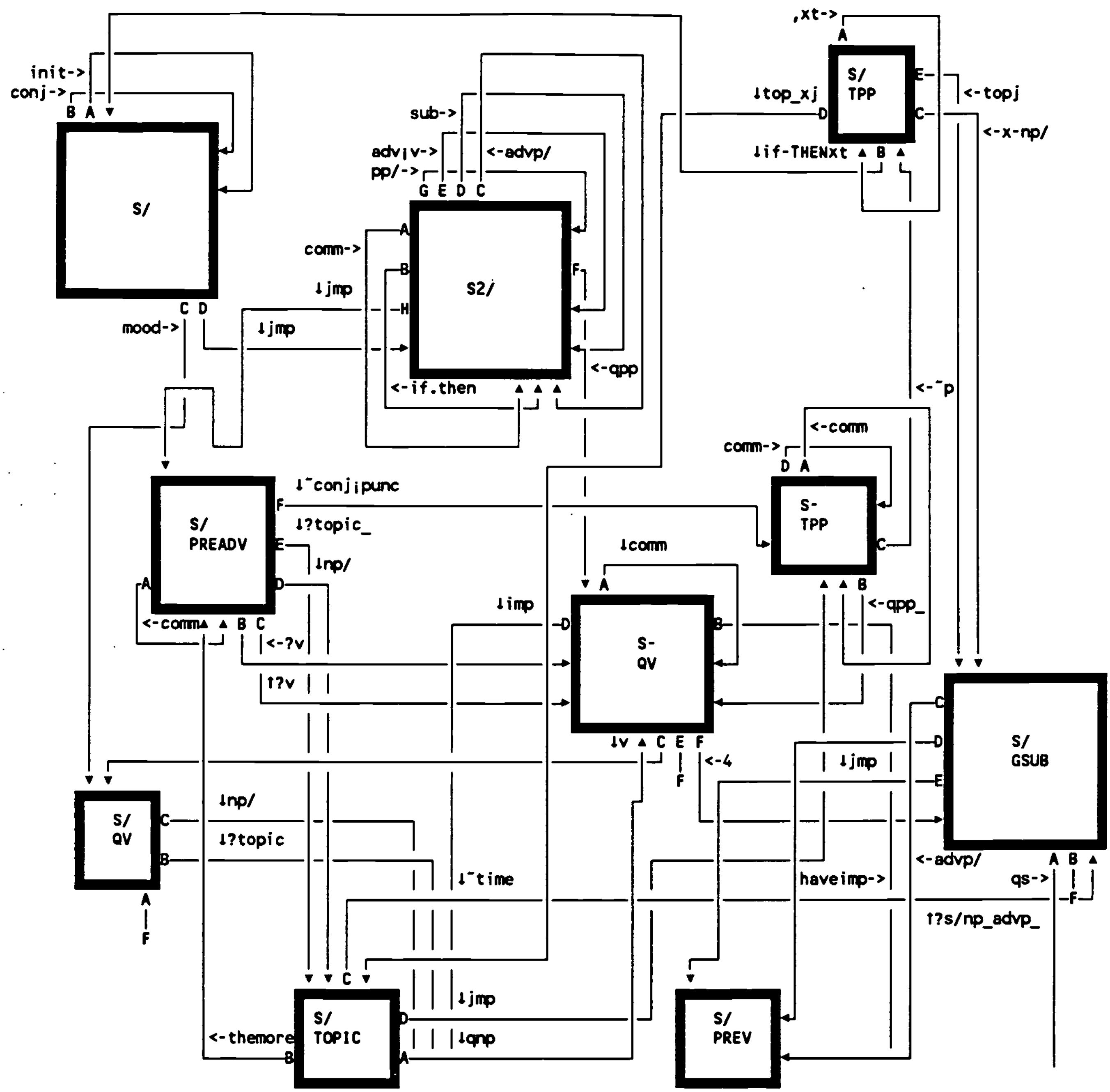


Map NPK. Complete Noun Phrase Kernel network.

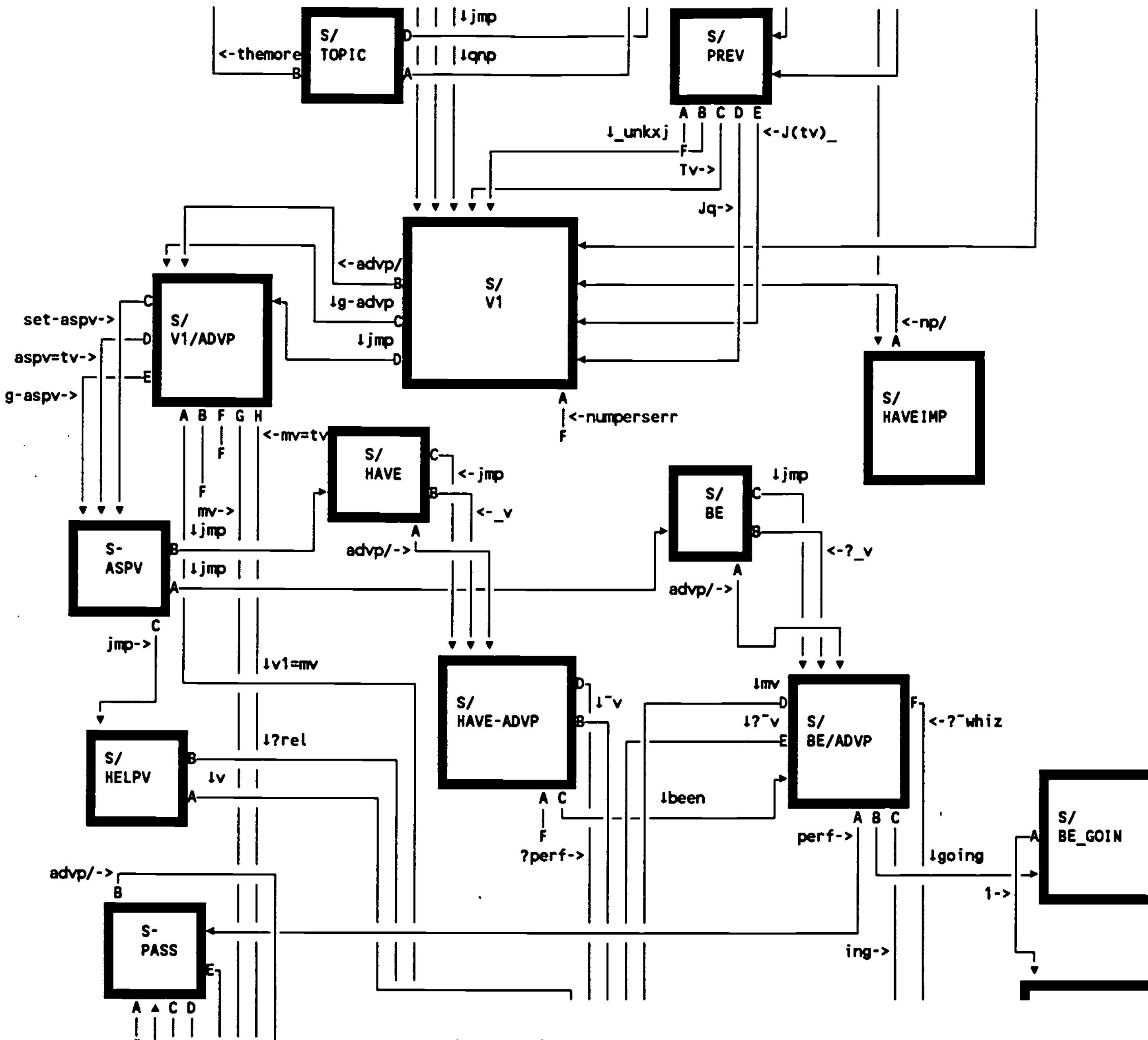
BEST COPY AVAILABLE



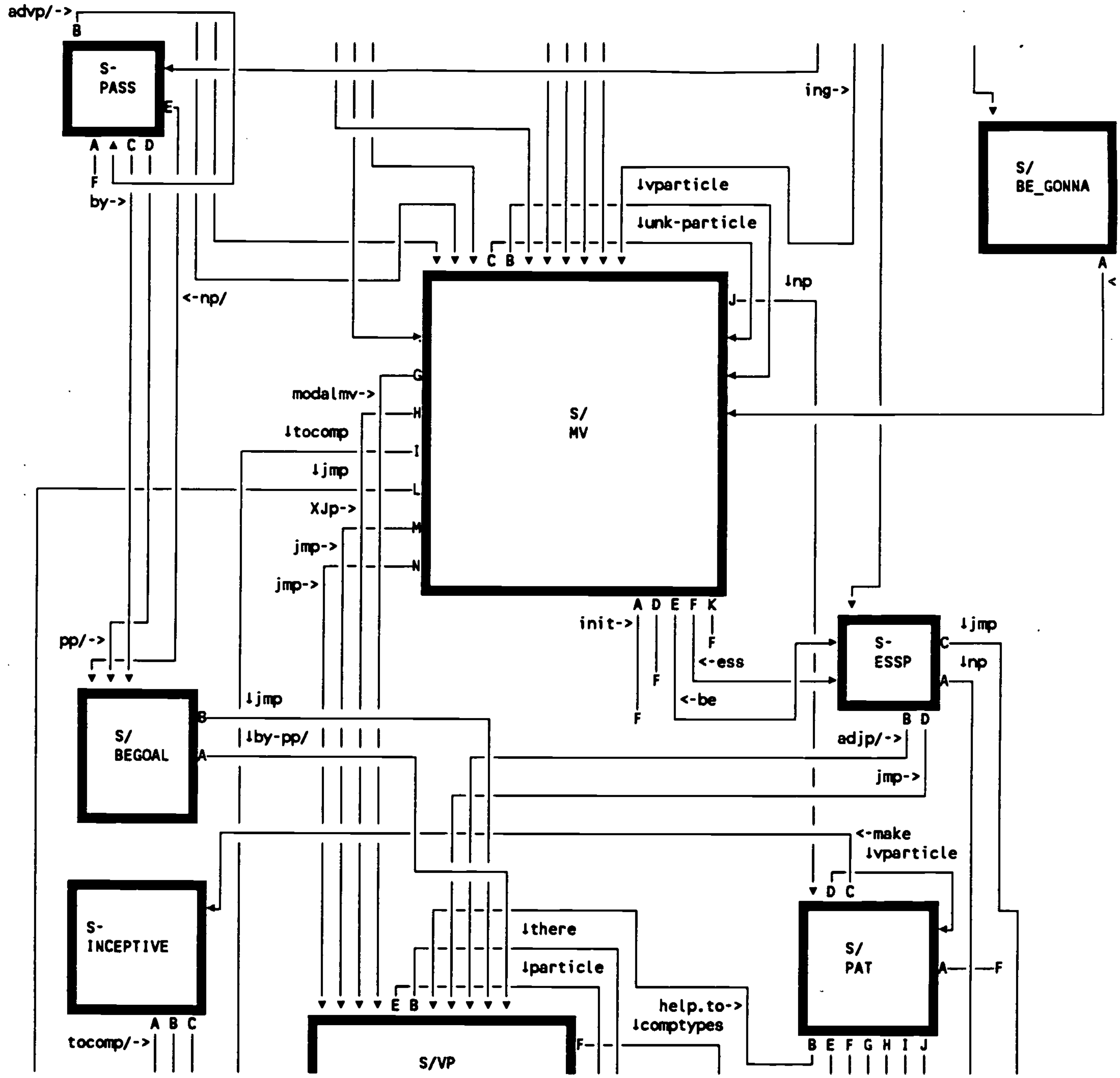
Map NPpart. Noun Phrase grammar of students not passing college entrance essay exam. Arcs mapped where n >= 5 from a sample of 233 NPs.



Map S (a). Complete S network.

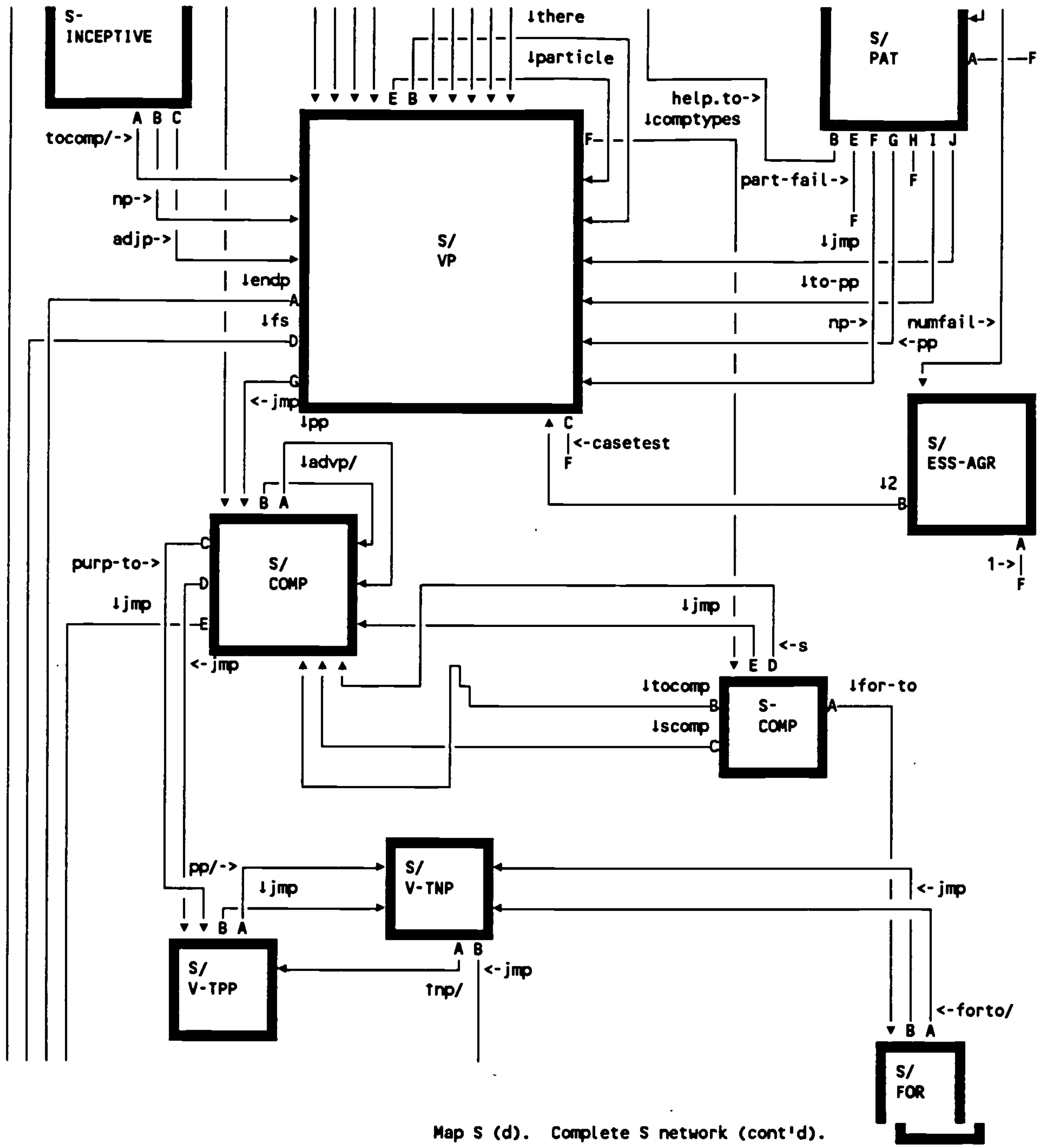


Map S (b). Complete S network (cont'd).



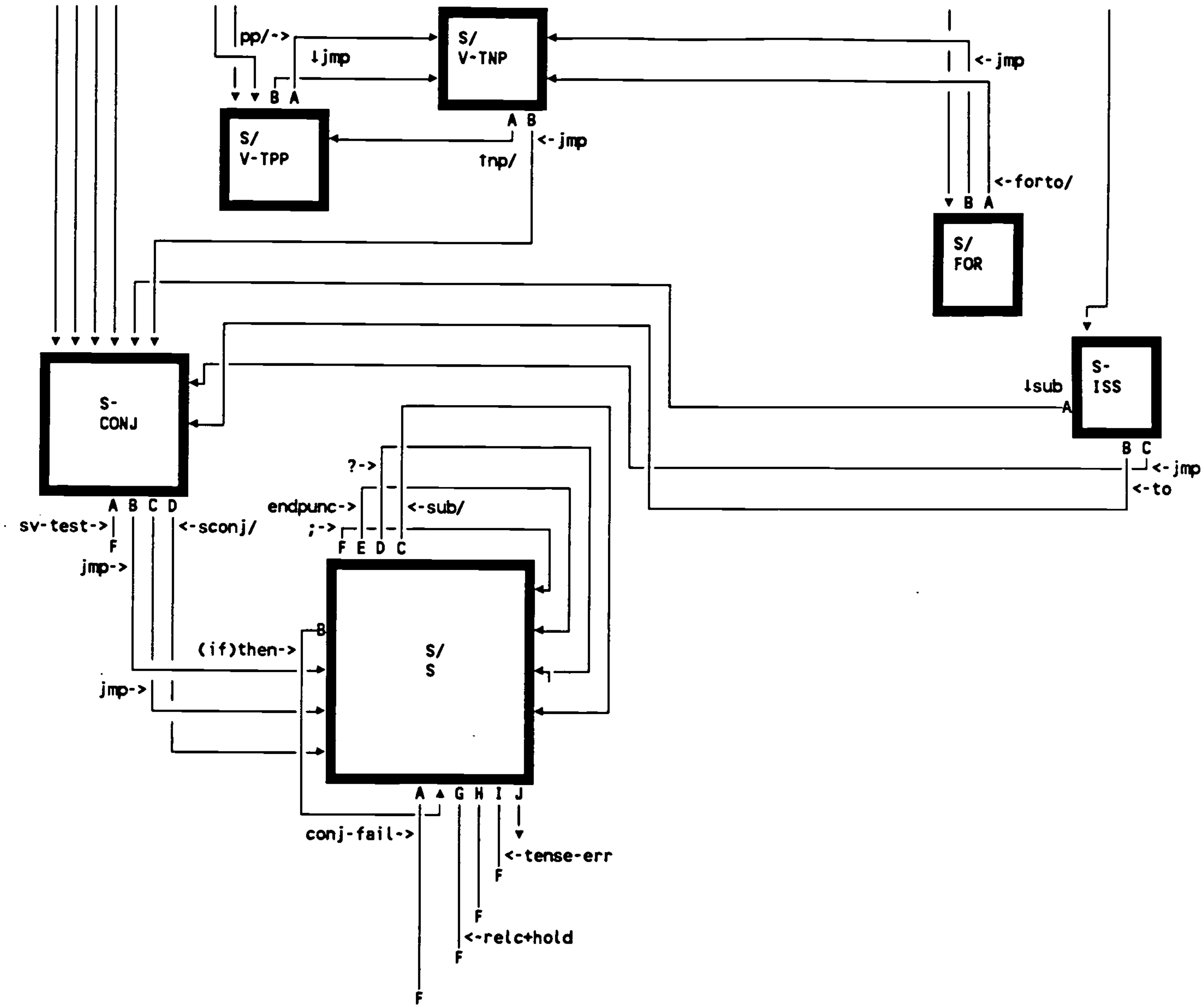
Map S (c). Complete S network (cont'd).

BEST COPY AVAILABLE

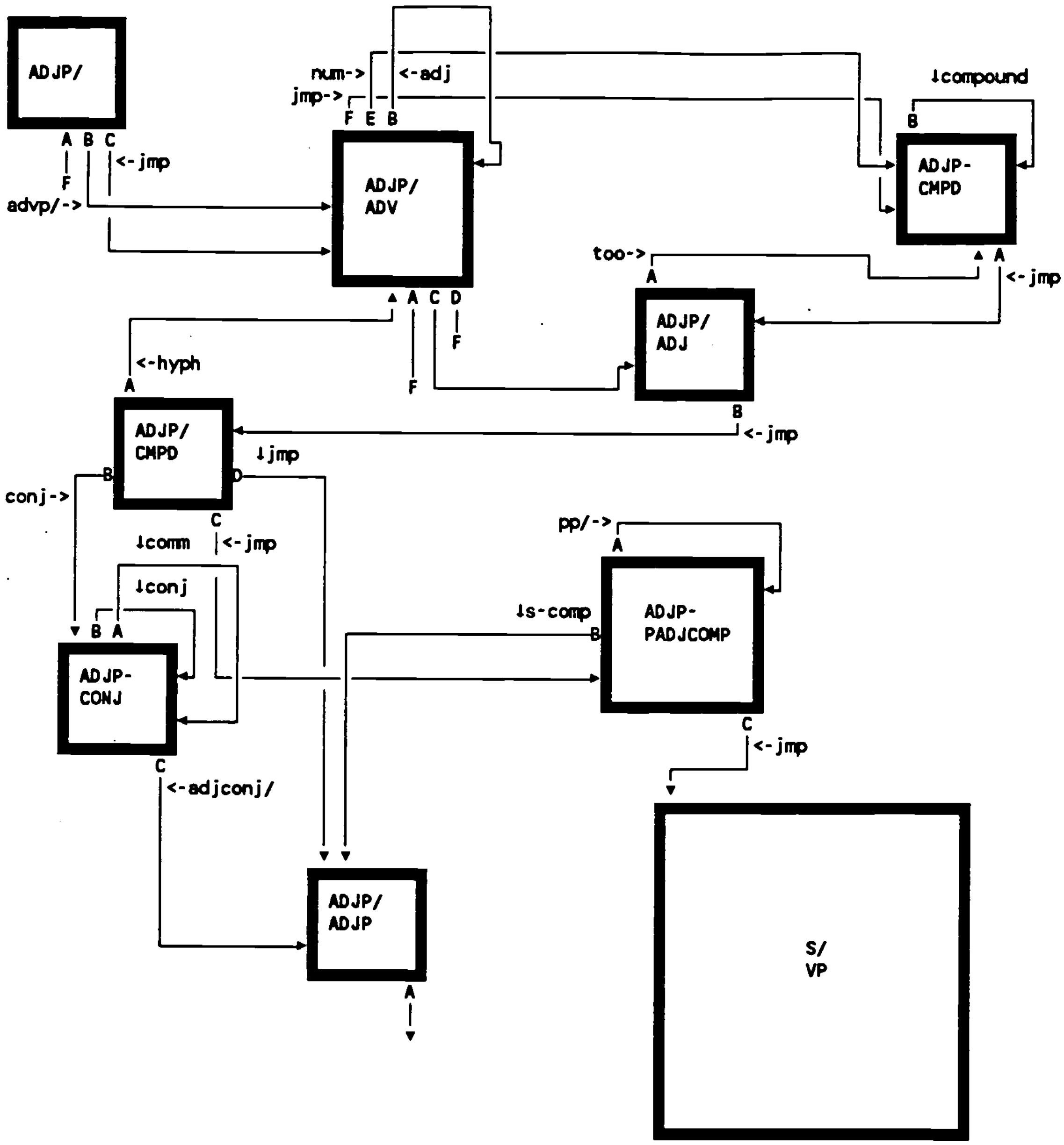


Map S (d). Complete S network (cont'd).

BEST COPY AVAILABLE

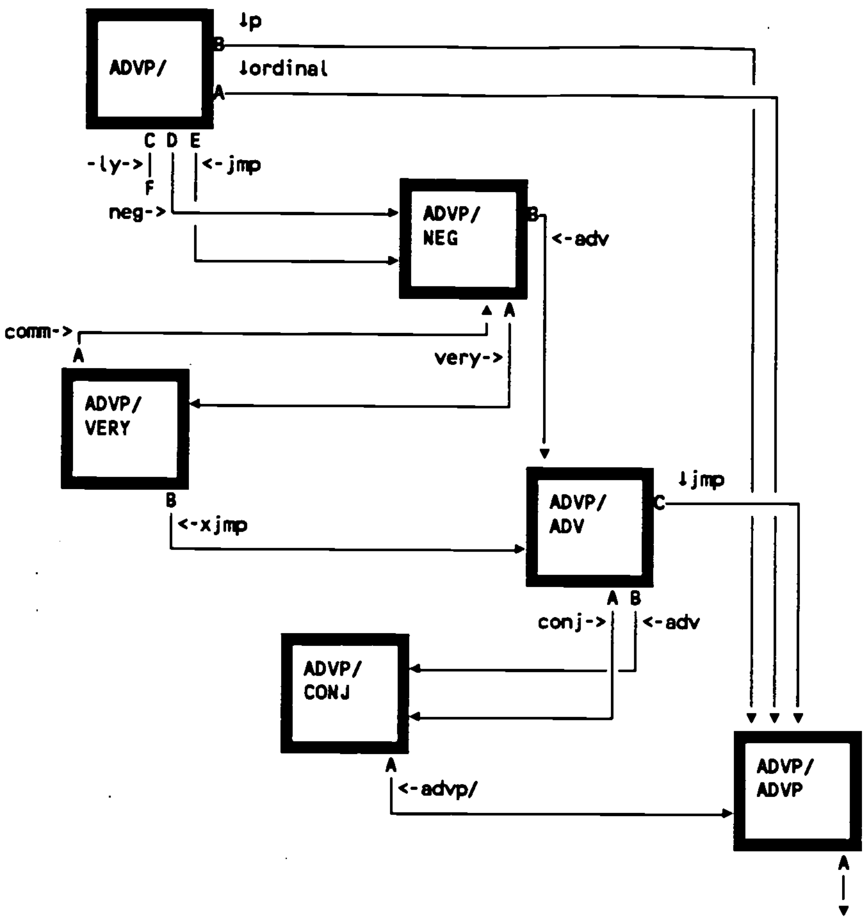


Map S (e). Complete S network (cont'd).



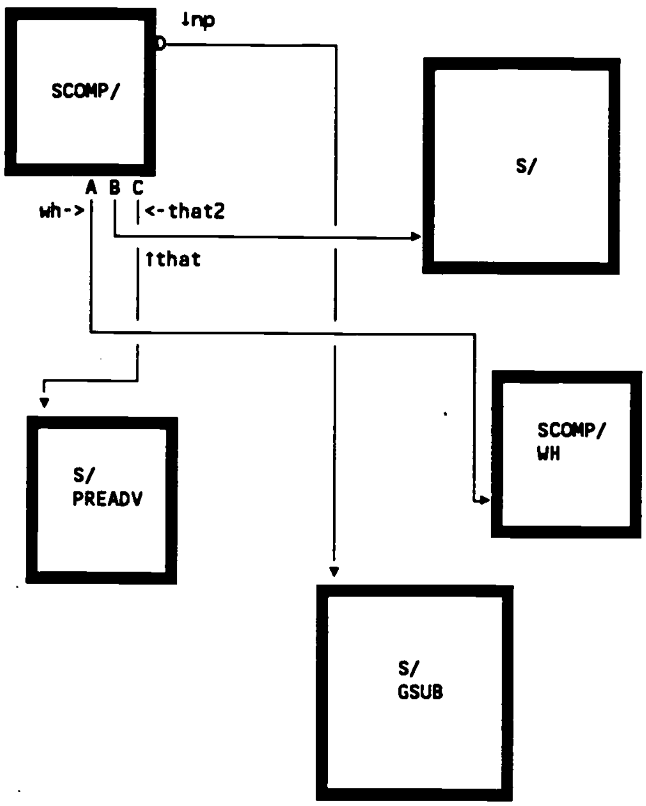
Map ADJP. Complete ADJective Phrase network.

BEST COPY AVAILABLE



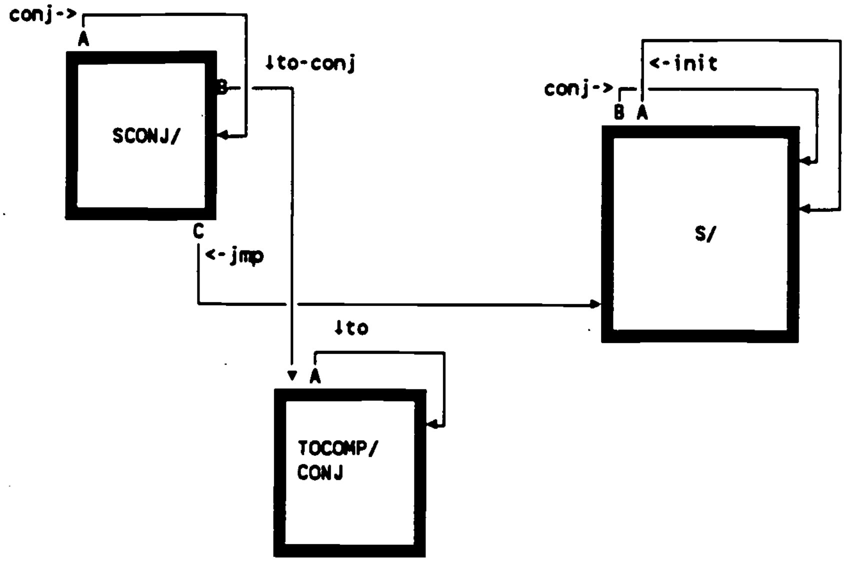
Map ADVP. Complete ADVerb Phrase network.

BEST COPY AVAILABLE



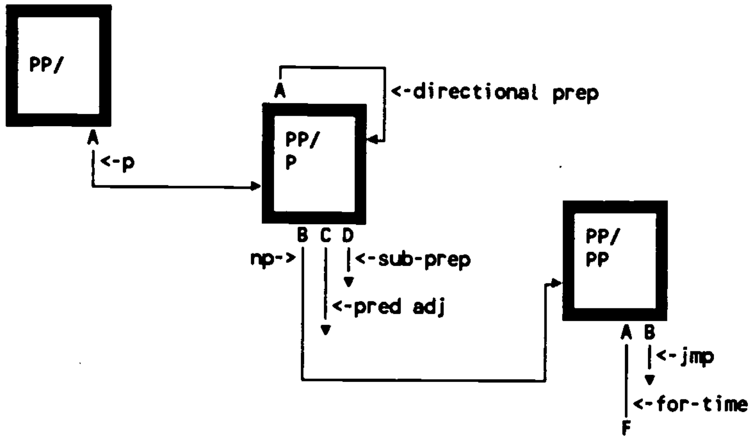
Map SCOMP. Complete Sentence COMPlement network.

BEST COPY AVAILABLE

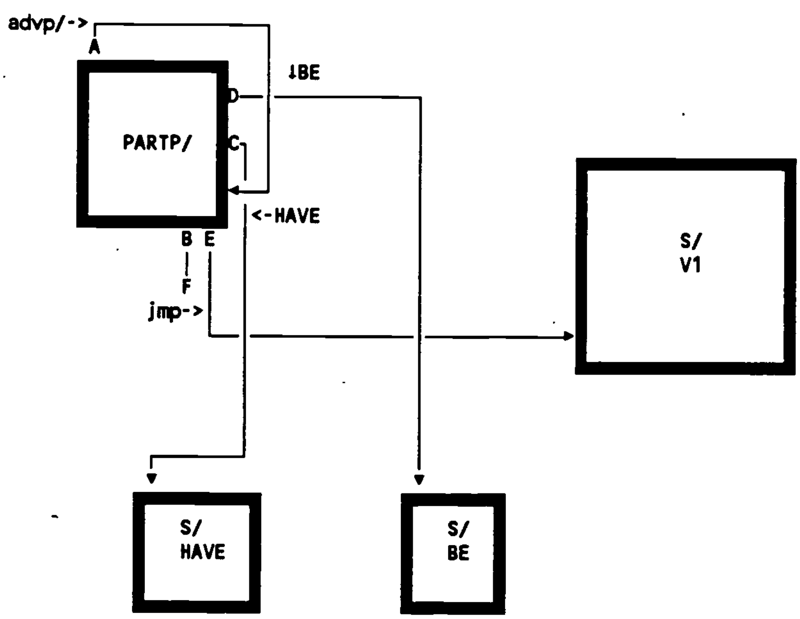


Map SCONJ. Complete Sentential CONJunct network.

BEST COPY AVAILABLE

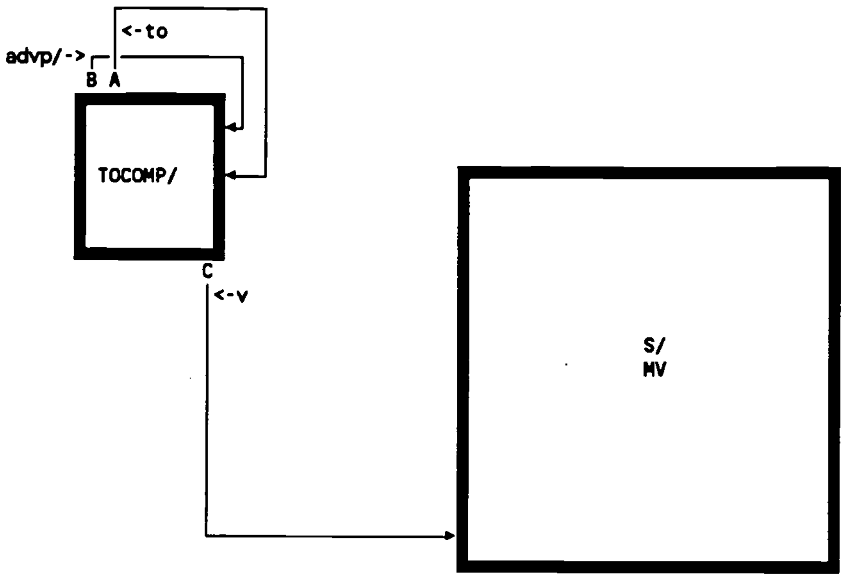


Map PP. Complete Prepositional Phrase network.

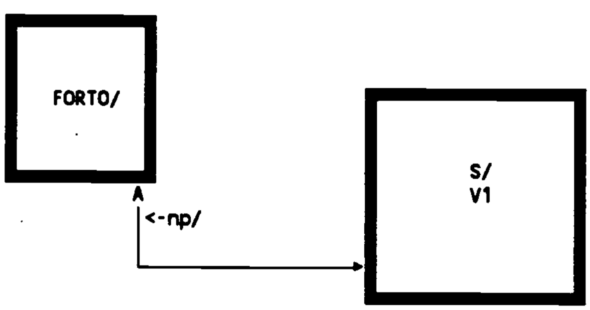


Map PARTP. Complete PARTicipial Phrase network.

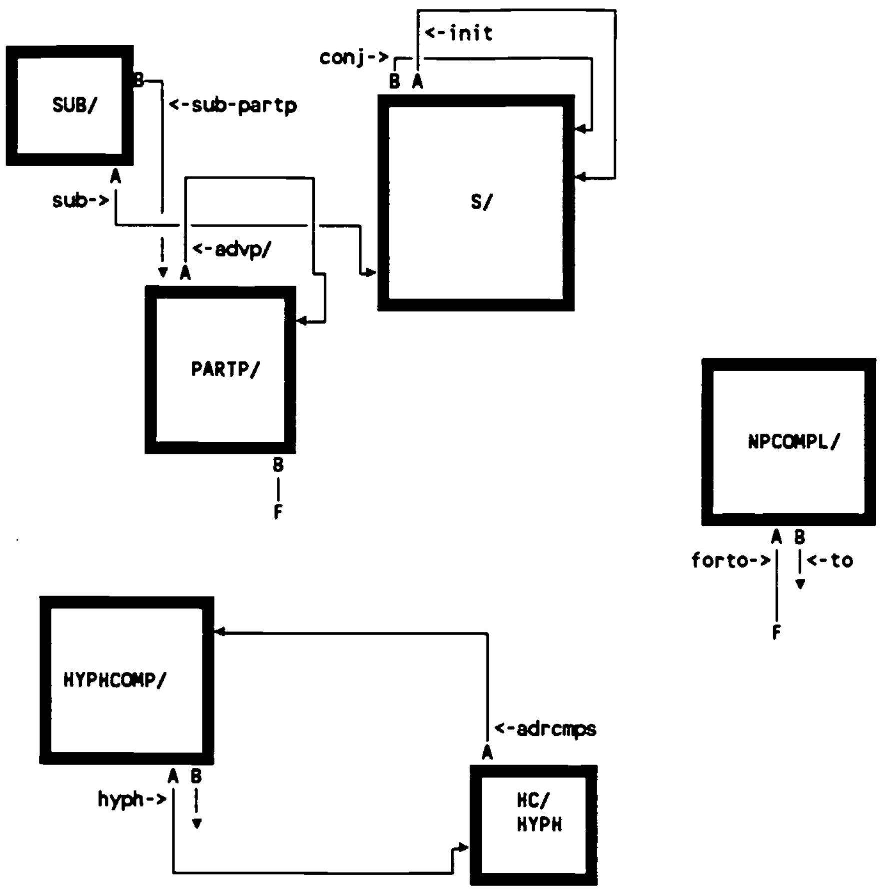
BEST COPY AVAILABLE



Map TOCOMP. Complete TO COMPlement network.

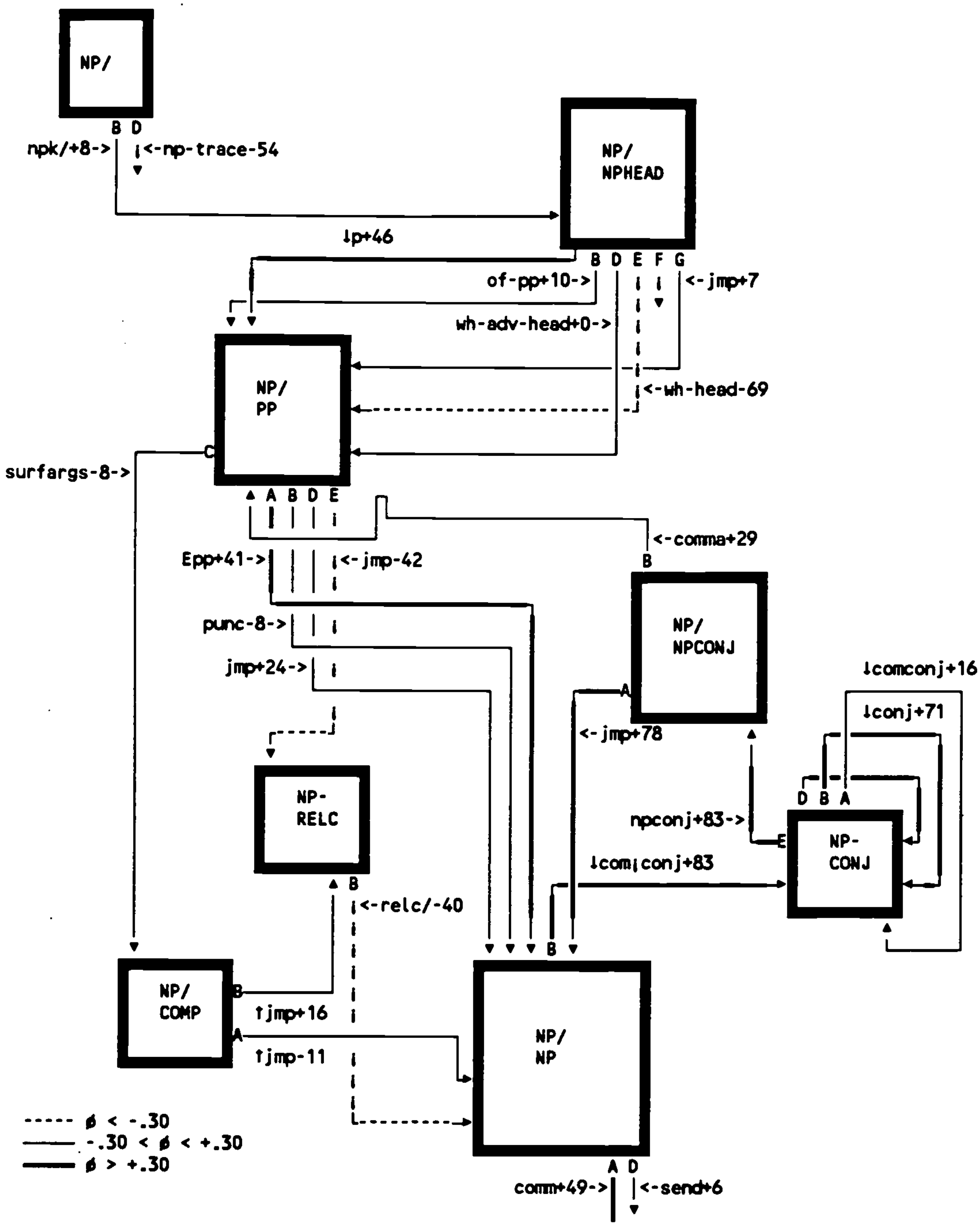


Map FORTO. Complete FOR TO complement network.

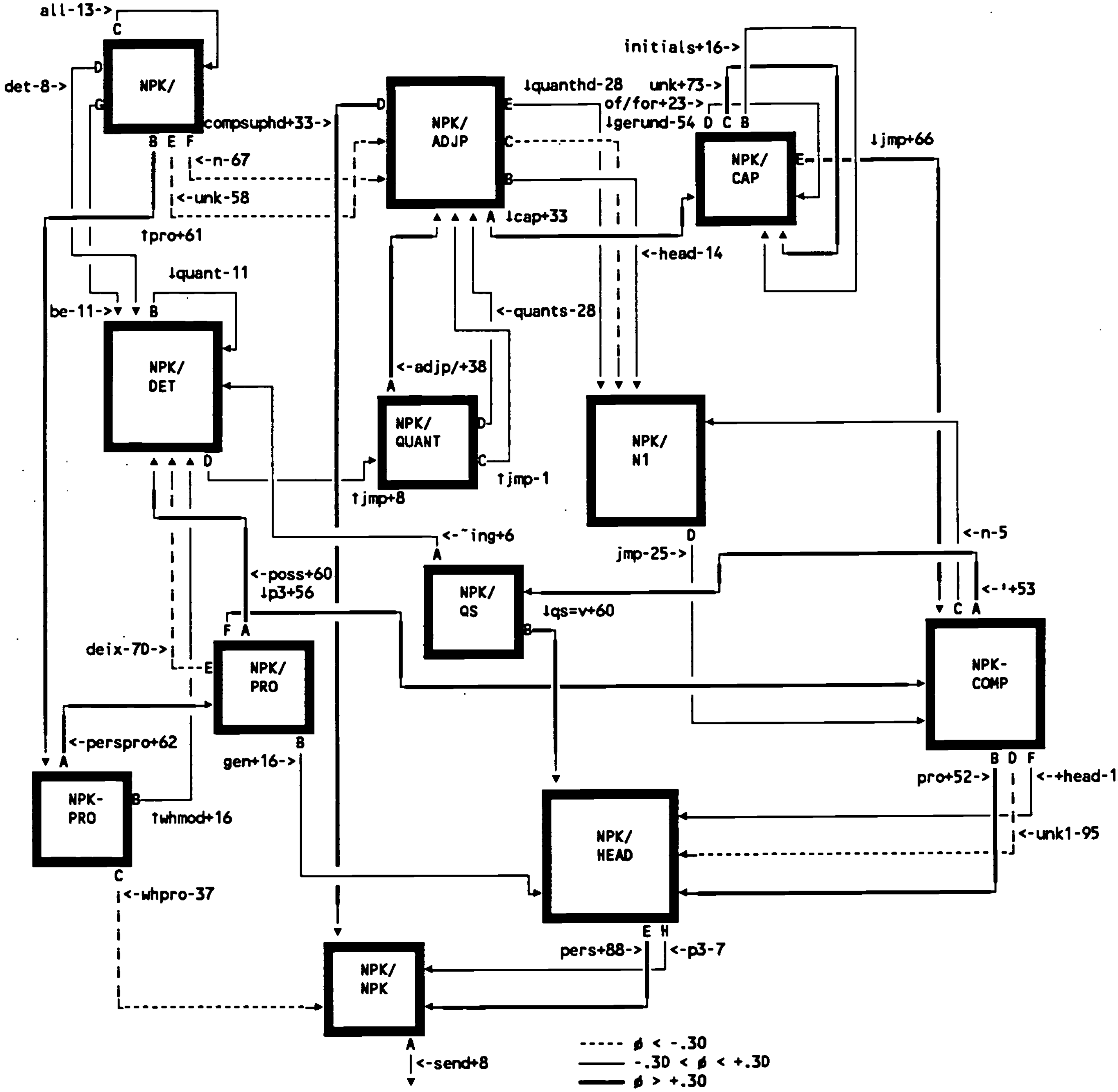


Map SUB. Complete SUBordinate clause network.
Map NPCOMPL. Complete Noun Phrase COMPLement network.
Map HYPHCOMP. Complete HYPHenated COMPound network.

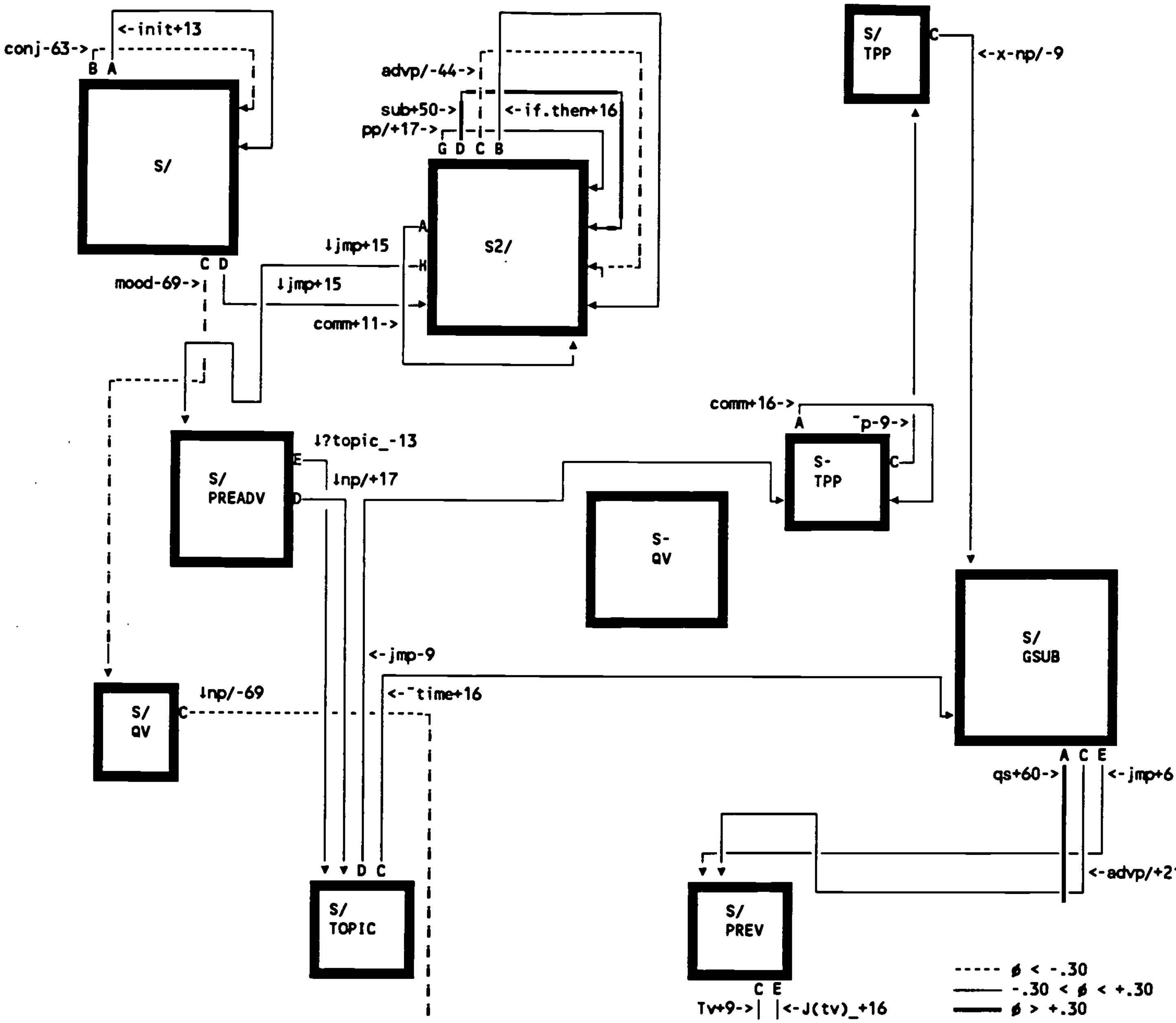
7.0 Differential Grammar Maps.



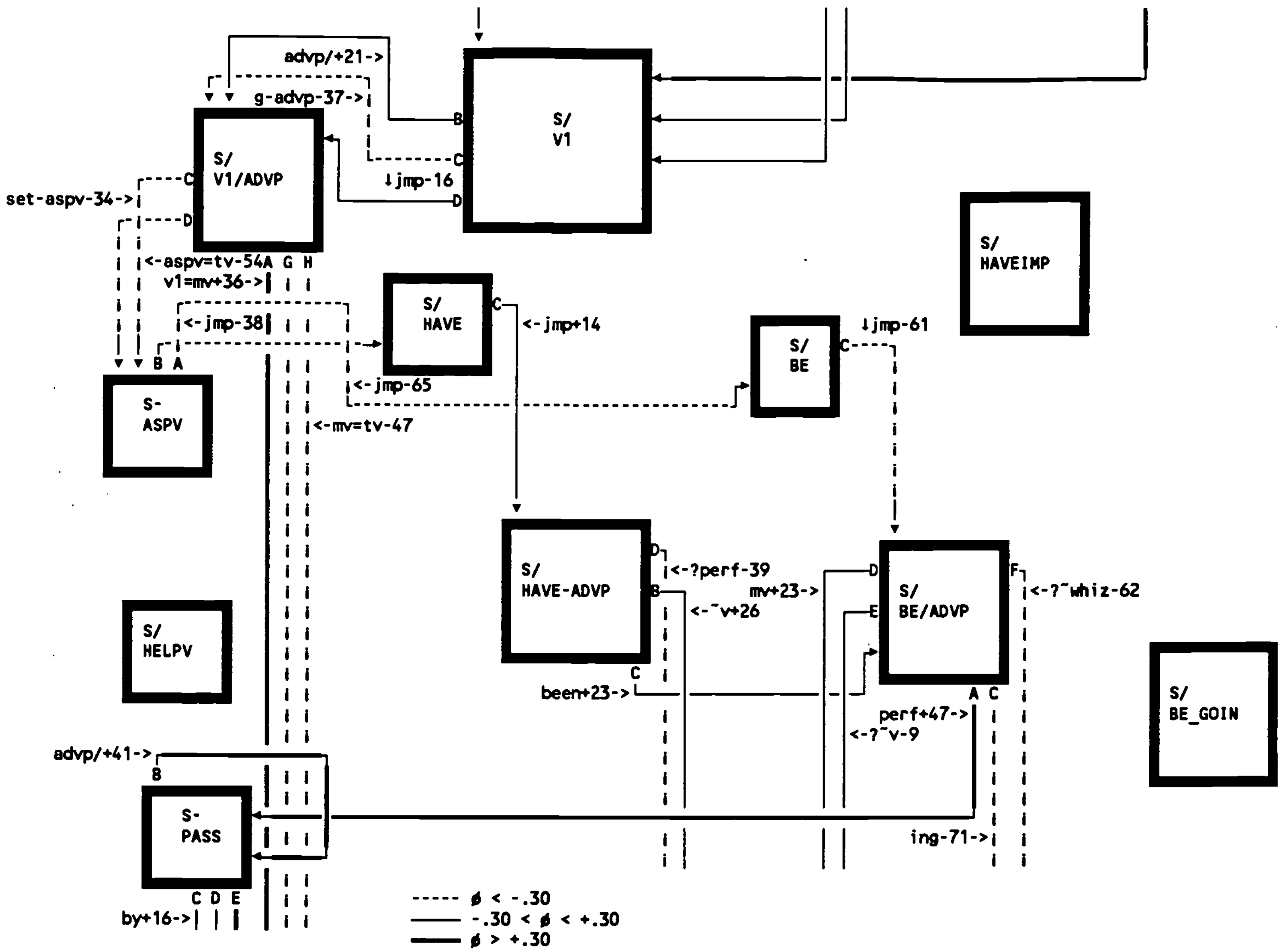
Map NPø. Differential Noun Phrase network



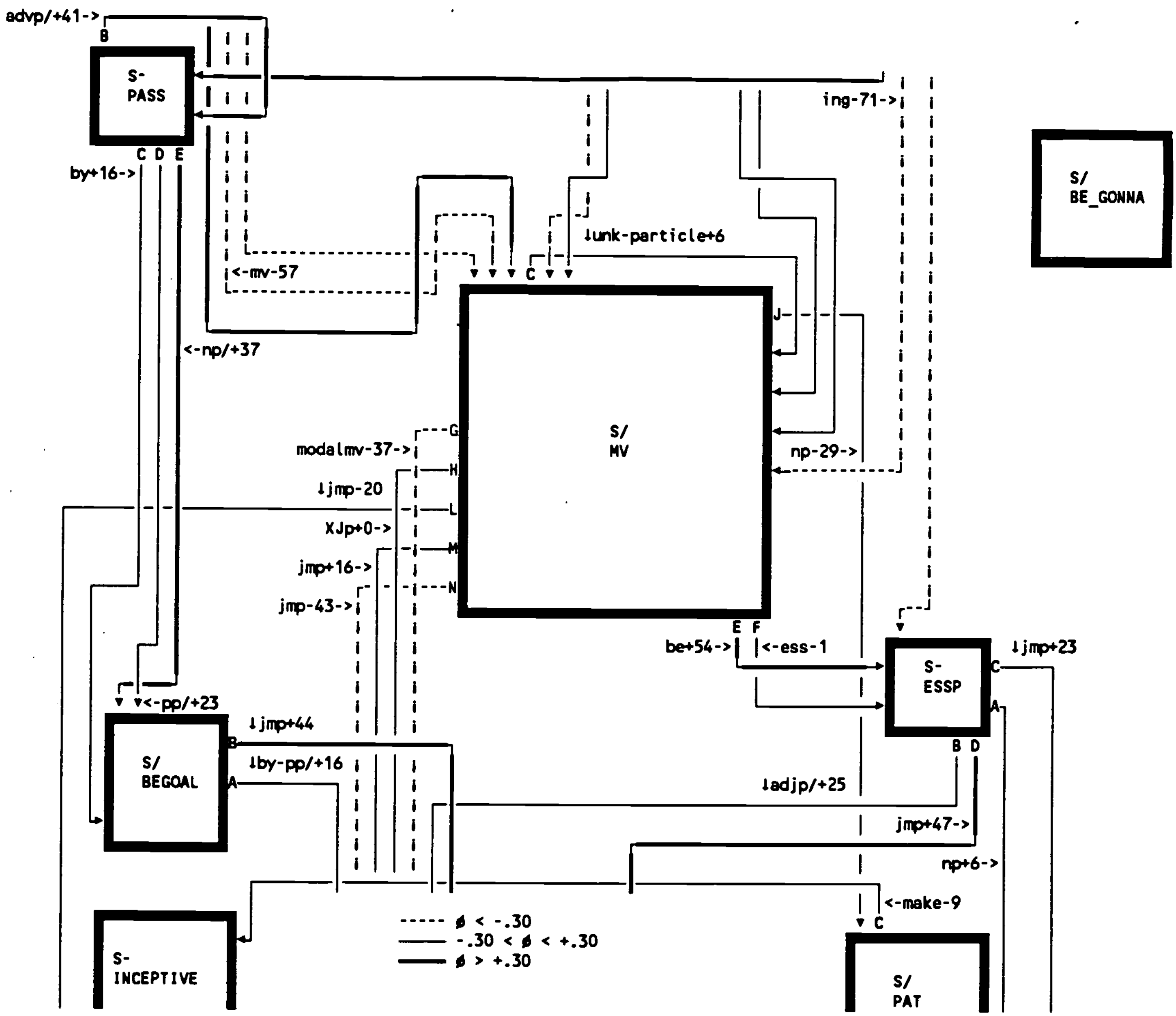
Map NPKø. Differential Noun Phrase Kernel network



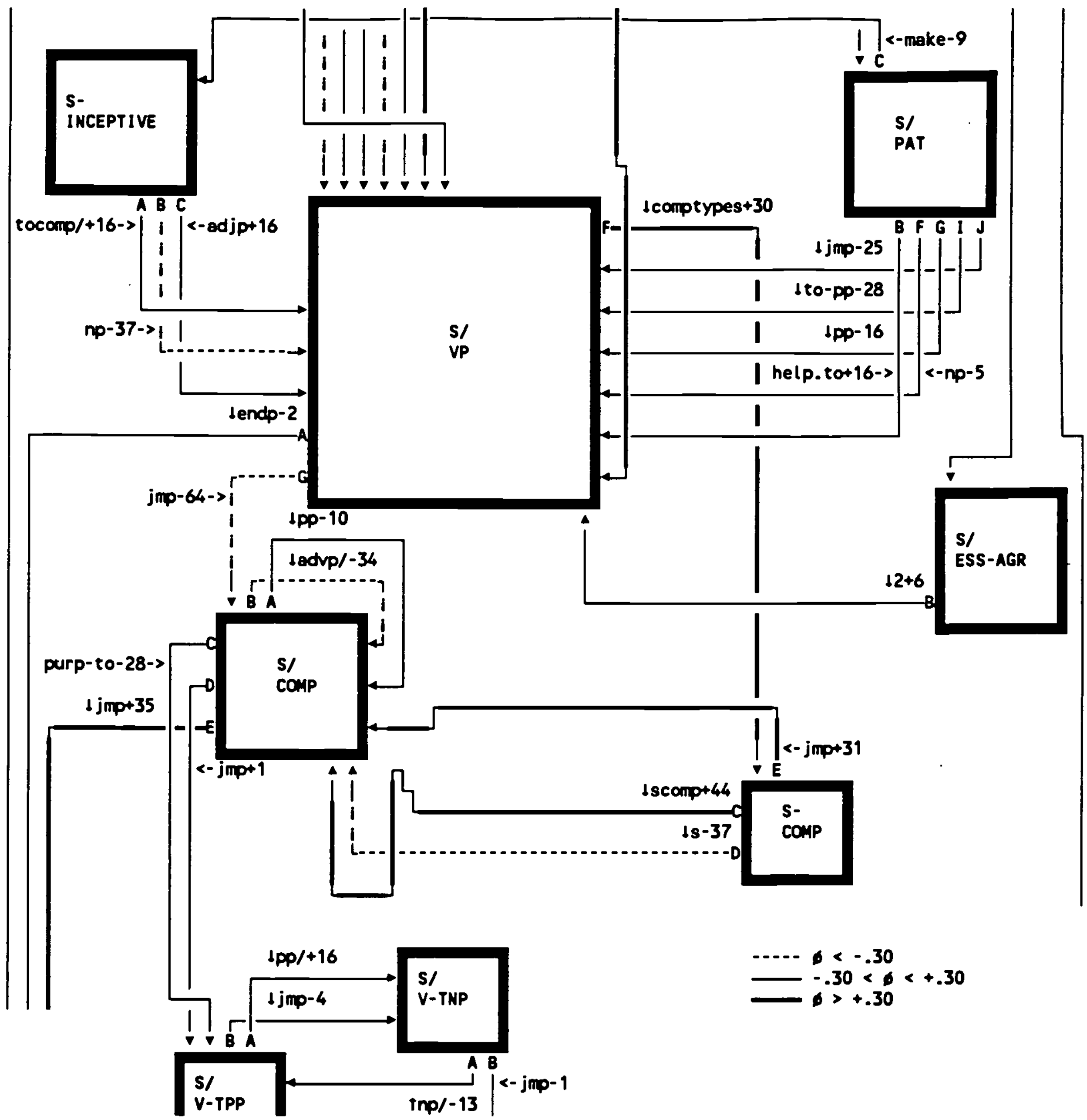
Map Sϕ (a). Differential S network



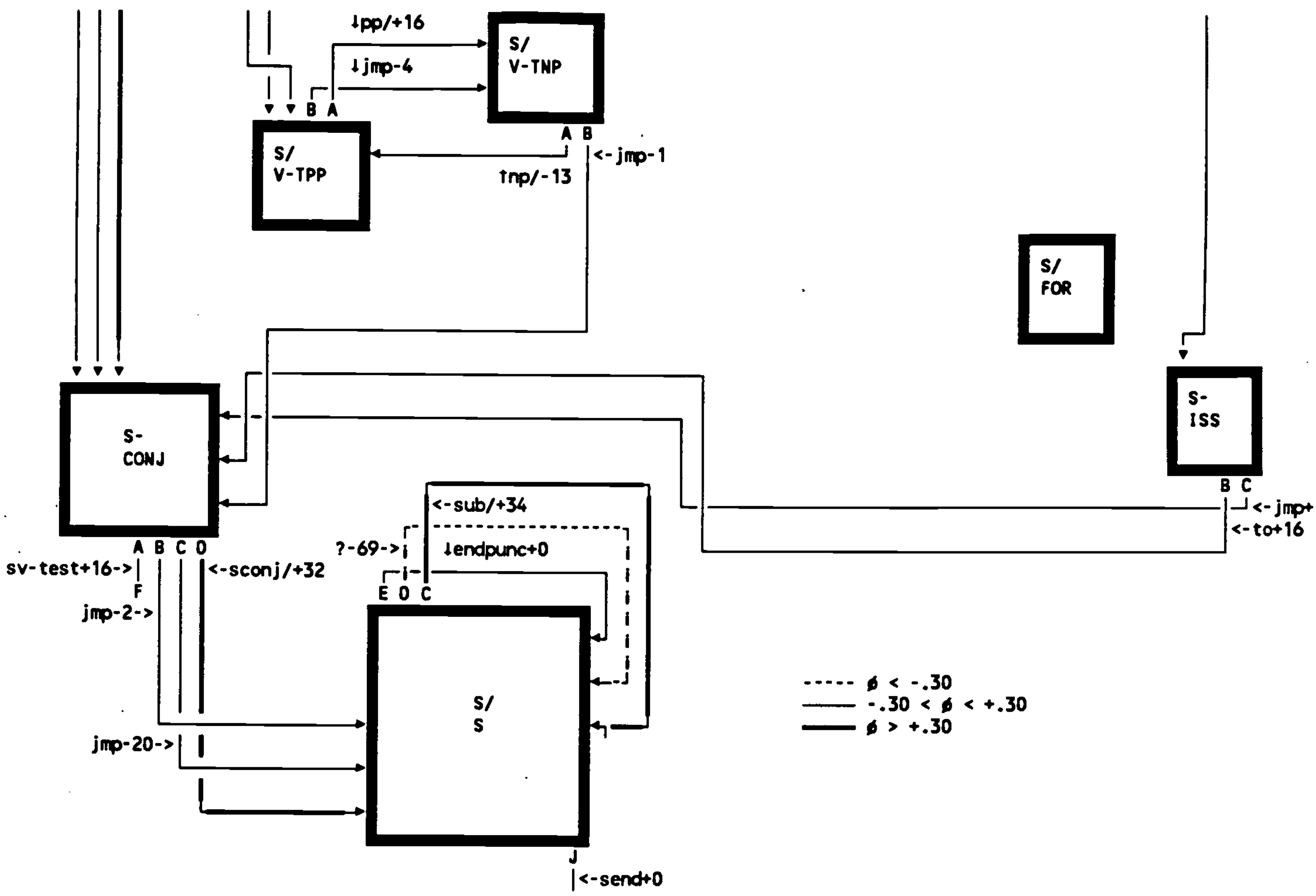
Map Sø (b). Differential S network



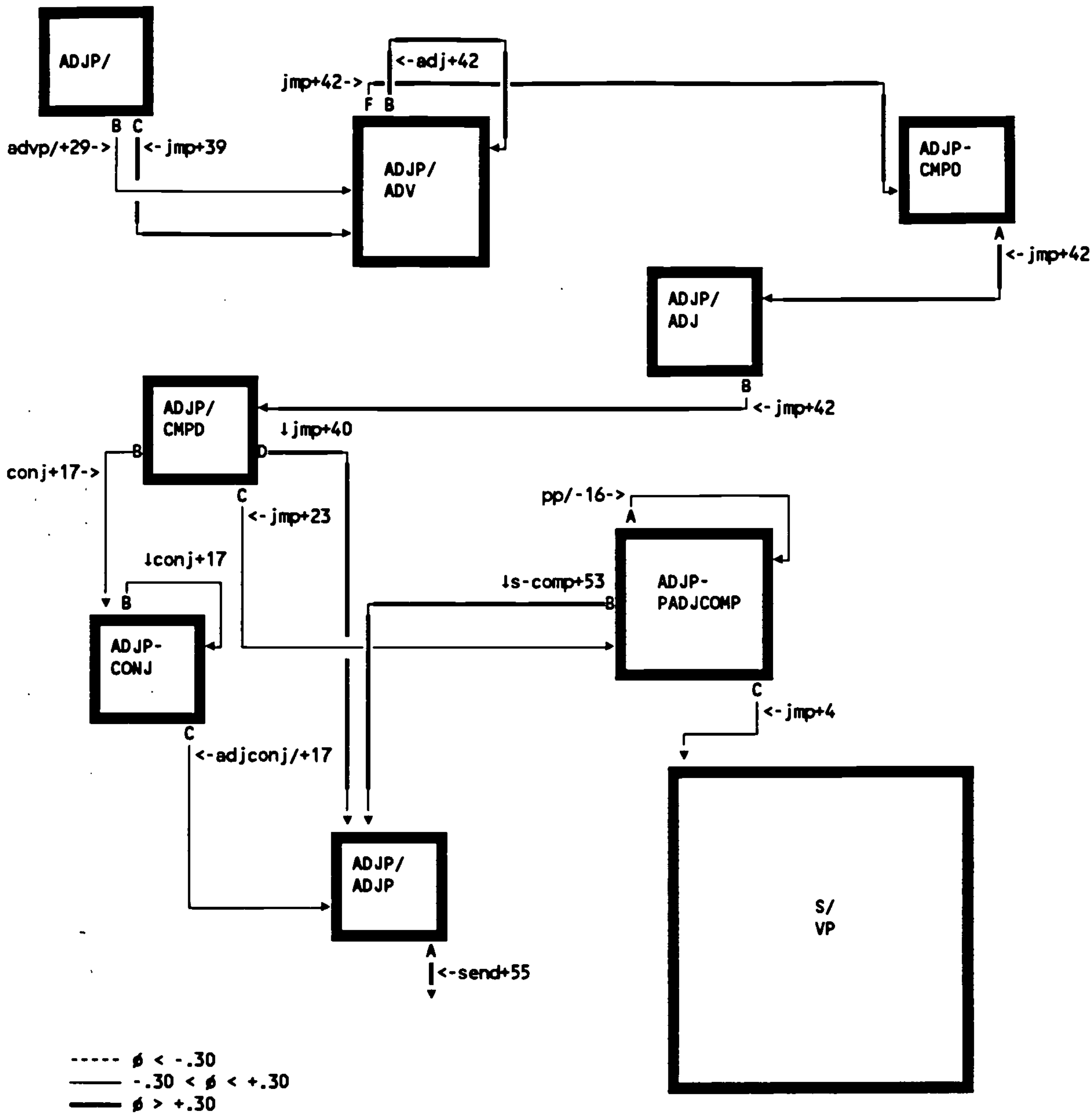
Map Sϕ (c). Differential S network.



Map Sϕ (d). Differential S network

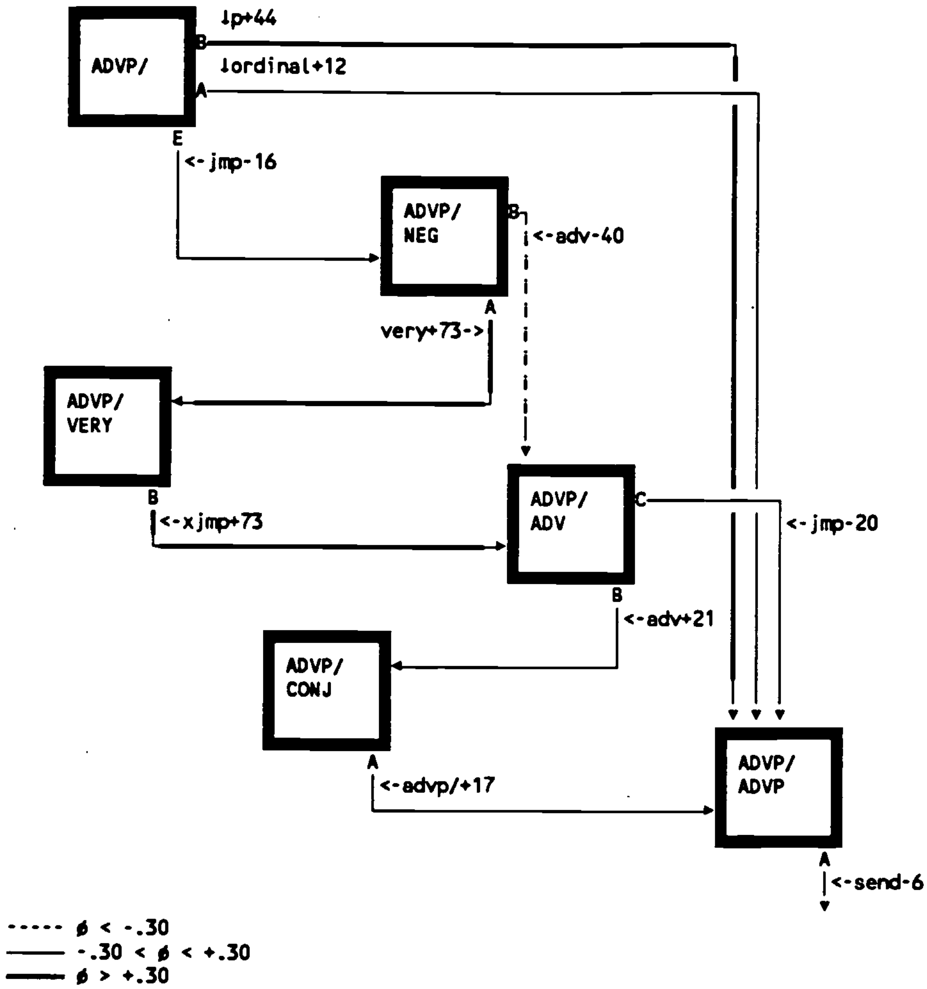


Map Sø (e). Differential S network



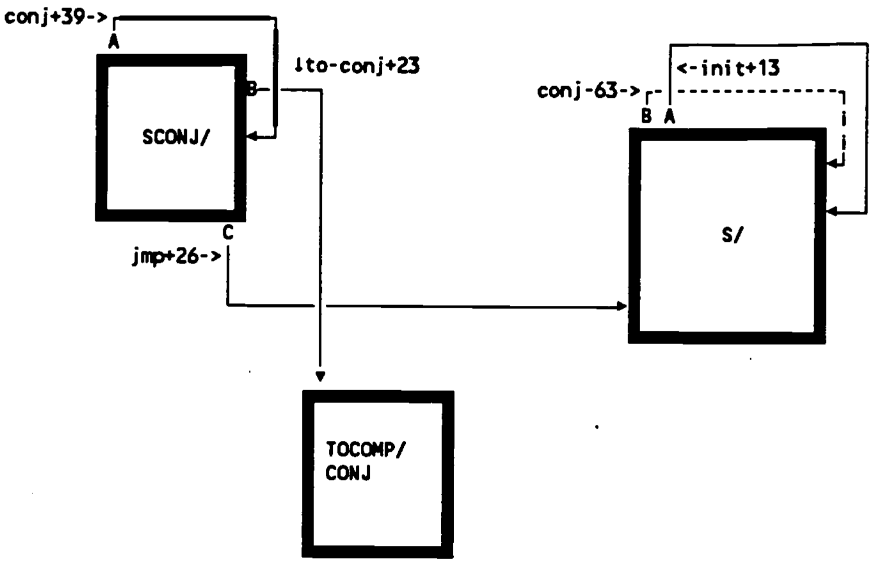
Map ADJPø. Differential ADJective Phrase network.

BEST COPY AVAILABLE

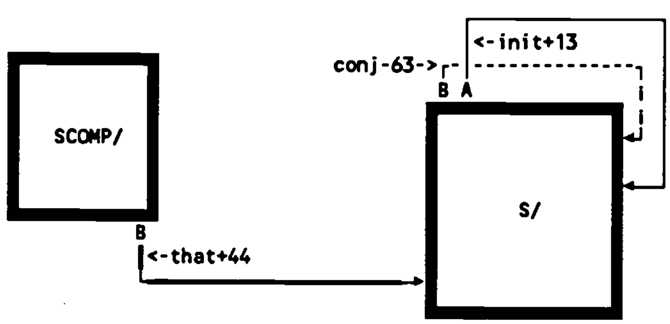


Map ADVPø. Differential ADVerb Phrase network.

BEST COPY AVAILABLE

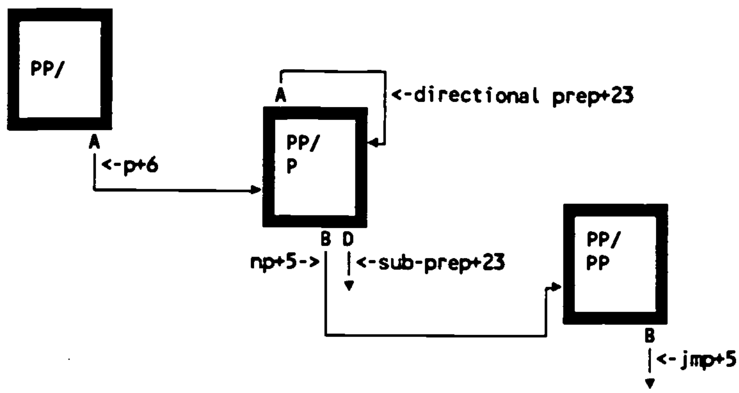


Map SCONJØ. Differential Sentence CONJunct network.

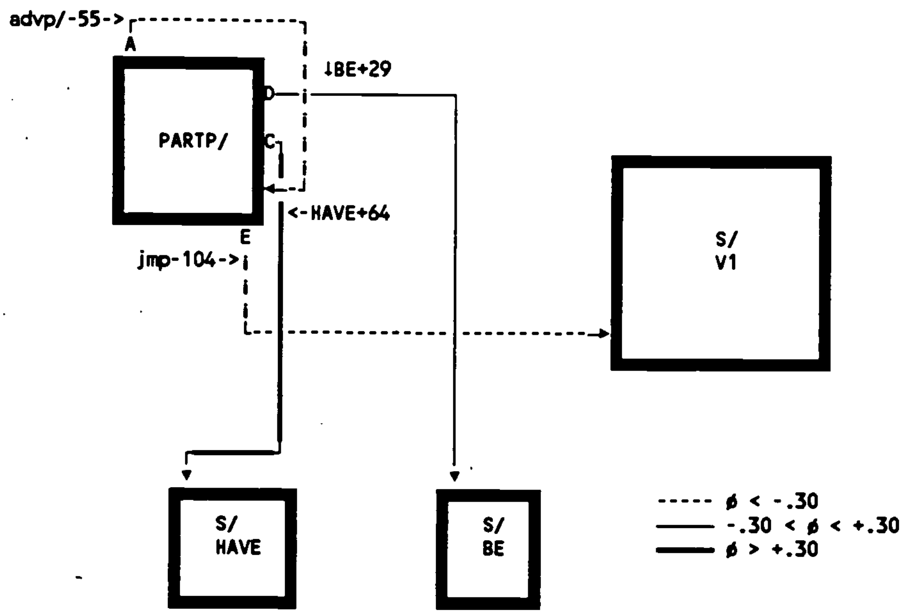


----- Ø < -.30
——— -.30 < Ø < +.30
——— Ø > +.30

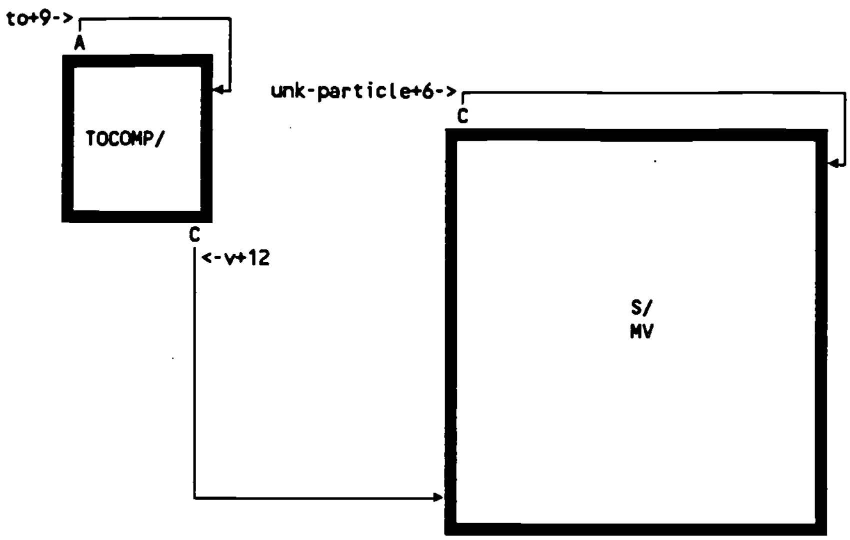
Map SCOMPØ. Differential Sentence COMPlement network.



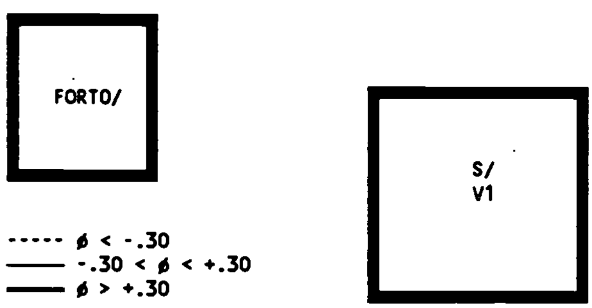
Map PPϕ. Differential Prepositional Phrase network.



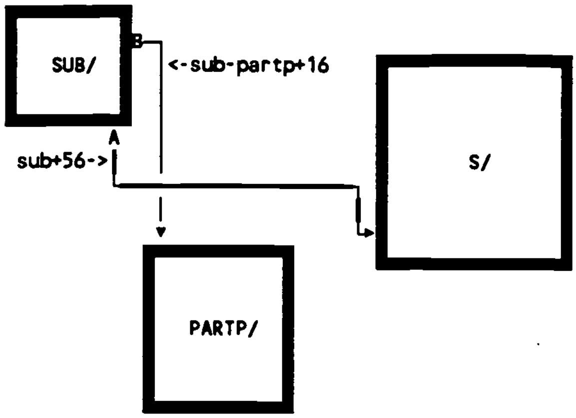
Map PARTPϕ. Differential PARTicipial Phrase network.



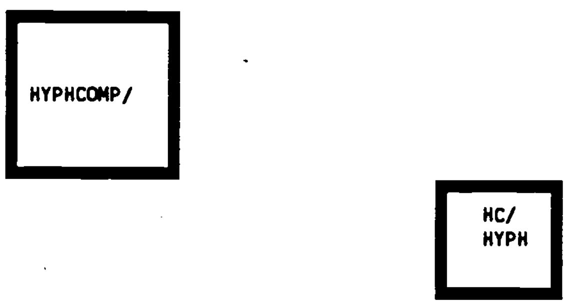
Map TOCOMPø. Differential TO COMPlement network.



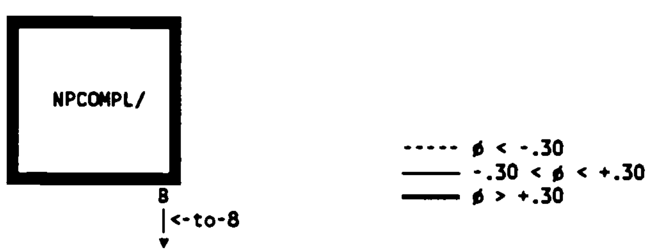
Map FORTOø. Differential FOR TO complement network.



Map SUBø. Differential SUBordinate clause network.



Map HYPHCOMPø. Differential HYPHenated COMPound network.



Map NPCOMPLø. Differential NP COMPLement network.

Generalized Transition Network Parsing for Language Study[1]

Donald Loritz
Department of Linguistics
Georgetown University
Washington, DC 20057, USA
loritz@guvax.bitnet

Abstract: A generalized transition network system, GPARS, is described particularly as it has been developed for the instructional parsing of deaf students' English. The GTN system extends the familiar Augmented Transition Network formalism by allowing top-down, bottom-up, depth-first, breadth-first, deterministic, and nondeterministic parsing strategies to be freely intermixed. These various strategies have also allowed the system to be used for parsing Chinese, Russian, and other languages.

Keywords: parsing, syntax, computer-augmented instruction, ATN, augmented transition networks, CALL, computer-assisted language learning, CAI, computer-assisted instruction, IBM PC, MS-DOS.

Donald Loritz [Ed.D., Applied Psycholinguistics, Boston University] is Assistant Professor of Computational Linguistics at Georgetown University. He has published articles on instructional parser design and instructional digitized speech processing.

INTRODUCTION

Nature unquestionably intended for a first language to be learned with at a mother's knee, and it would be nice if student-teacher ratios could be lowered so that second languages could be learned in the same way. But society seldom is as patient with its students as a mother is with her children. Learning a natural language is a slow, difficult, often tedious and, therefore, expensive process. Consequently, many second language teaching methodologies have been developed to make the second language learning process more cost-effective.

Traditional, grammar-based second language instruction has sought to lower the cost of language instruction by enabling the student to self-correct through the application of grammar rules. The central problem with this method has been that it interposes the requirement to learn an intermediary third language, grammar, between the learner's first and second languages.

[1]Supported in part by the U.S. Department of Education, Office of Postsecondary Education [G008740399], Secretary's Discretionary Fund [G008720150], and Office of Special Education and Rehabilitative Services [H180P80020].

For the past several years, I have been working on the development of instructional parsing systems which can partially excuse teachers from the mechanical task of error correction and students from the obstructive task of grammar study. An important secondary objective was that the system be implemented on affordable technology. Affluent students who can afford a human second language tutor have no need of a mechanical tutor.

My students and I have built laboratory systems for a variety of languages, but the immediate objective of the research reported here has been to develop an instructional parser for English. The first working student version of the Ms. Pluralbelle system has now been tested and qualified against the compositions of college-bound deaf learners of English. Ms. Pluralbelle currently achieves 90% accuracy in grammatically checking compositions randomly selected from this corpus. Now that the main system has been built and tested, other objectives can be pursued. These include the diagnostic analysis of the parsing results of students' written language and the implementation of the system for other languages.

In this paper, I will particularly describe the GPARS parsing engine which underlies Ms. Pluralbelle, and how it has evolved to meet these several objectives across a variety of languages. [The GPARS system is implemented in LISP. Because LISP makes heavy use of parentheses, textual asides like this will be enclosed in brackets. When necessary, LISP words will be capitalized or parenthesized.]

DEVELOPMENT HISTORY

The GPARS system originated as "Henry Higgins", a digital intonation display for English [Loritz, 1983], and "Miss Fidditch", a small, grammar-checking Augmented Transition Network [ATN] parser for English [Loritz, 1984]. Both were implemented for 8Mhz, 16MB Apple II computers, but when Apple effectively abandoned the Apple II's RISC architecture, the systems were ported to the IBM PC architecture.

On the PC, Miss Fidditch was first implemented in IQ-LISP. Although originally conceived as an English as a Second Language system, prototype systems for Chinese and Russian were funded first. This created the practical need to design a system which was generalized and "universal" in its capacity to accommodate very different languages. In 1986, Miss Fidditch's ATN interpreter was extended to the current generalized transition network [GTN] design, and ported to TLC-LISP/86 [John Allen, 1978, 1985].

With the development of grammars for other languages, the English parsing system was distinguished as ENGPARS, and the ESL learner system was renamed "Ms. Grundy". The system is now being ported to TLC-LISP/386 [Wagner, 1989], and the ESL learner system has been ineluctably renamed Ms. Pluralbelle [Loritz, 1989]. I shall henceforth use "GPARS" to refer to the most general system underlying my several English, Russian, Chinese and other parsers.

I shall use "ENGPARS" to describe the English parser (of principal concern here), and "Ms. Pluralbelle" only to reference specifics of the ENGPARS student-user interface.

THE ATN FORMALISM

The basic ATN formalism was originally selected for several reasons. First, it yields compact, fast grammars -- still essential if a system is to be implemented on affordable computers.

Second, because the ATN formalism is equivalent in power to a Turing machine, it places minimal constraints on the final form of the grammar.

This freedom has been criticized on the grounds that good engineering selects the least powerful parsing engine necessary for a particular task. I rejected this criticism on several grounds. First, it is apparent that the human brain is a massively parallel processor and that language is naturally a massively parallel process. Language learning and teaching are therefore tasks for which even a Turing machine is seriously underpowered.

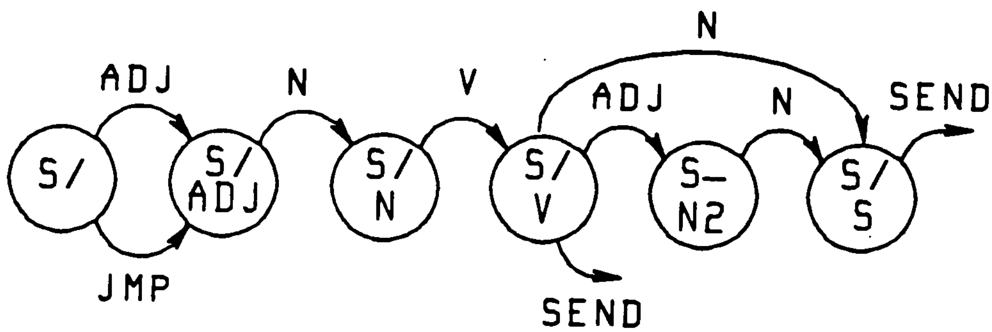
Second, this seemed especially desirable in the early 1980s because the "ill-formed input" of language learners had not been [and still has not been] widely-studied computationally. Neither had extensive computational research yet been done on a variety natural languages other than English.

Finally, parallel models of mind (Grossberg, 1969, et seq.) emphasize patterns of perception and behavior, in contrast with rule-governed approaches. Insofar as ATN grammars emphasize pattern, they present themselves as a congenial medium for expressing variance and invariance in language.

Types of transition networks

Transition networks are often presented as being of three types: basic, recursive, and augmented, but it is more useful to recognize eight types of transition network parsers, each characterized by a distinctive feature: elementary, optional, backtracking, full backtracking, structured, recursive, local, and augmented. The features of all eight types are present in a GTN, but all are not necessarily present in any given ATN, so I will briefly review them here. Readers desiring a more detailed introduction to ATNs are referred to Bates [1978], Winograd [1984] and Allen [1987].

Elementary and optional networks. An Elementary Transition Network begins in a start state (e.g., S/ in Network Grammar 1).



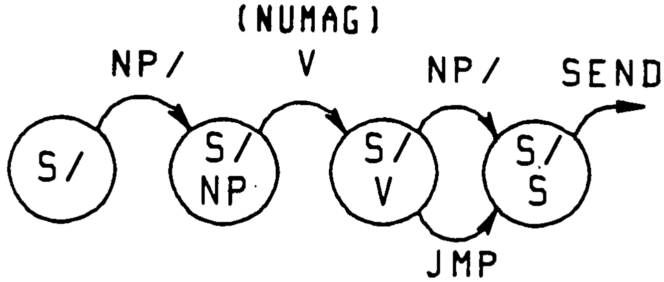
Network Grammar 1. An optional transition network.

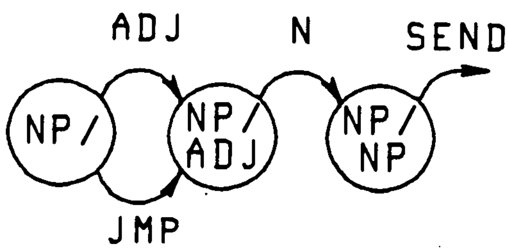
If the first input word of the sentence being parsed matches or otherwise satisfies some condition(s) on the first path or "arc" leaving the state, analysis proceeds "to" the next state and resumes with the second input word of the sentence. Such an arc is called a "to arc". For example, if we were parsing the sentence "Small children expend effort", the adjective "Small" would satisfy the to-arc "adj", and analysis would proceed to the state S/ADJ on the word "children".

A "jump arc" does not advance the analysis to the next word of the input. Without conditions, a jump arc makes the conditions of any preceding arcs **optional**. Jump arcs greatly increase the compactness of transition network grammars. Simple addition of the JMP arc to S/ allows Grammar 1 to allow both "Small children expend effort" and "Children expend effort".

Backtracking. "Backtracking" occurs when a parser makes a mistake. For example, parsing the sentence "Economy tickets cost less money", a parser following Grammar 1 might first jump to state S/ADJ and parse "economy" as a noun and "tickets" as a verb. When the following words cannot be matched, the parser must backtrack to state S/ and accept "economy" as an adjective. The system must "remember" everything it knew when it first visited the backtrack state. Usually the backtrack state is not the start state, so this can entail considerable bookkeeping.

Structured and recursive networks. Grammar 1 can be **structured** in the same manner that subroutines structure computer programs. By factoring the noun phrase specifications beginning at S/ and S/V out of Grammar 1, we arrive at the "structured" Network Grammar 2.





Network Grammar 2. A structured network grammar.

Now the sentence network "calls" the subnetwork NP/ from states S/ and S/V. It can match the same sentences as Grammar 1, but it is more concise.

The same mechanisms and variables can be used to parse the NP/ network as are used to the S/ network. Indeed, we must at times **recursively** parse the S/ network to capture embedded sentences. Recursion means that we re-use the mechanisms and variables of a network. To use the same variables without overwriting them, we must save their previous values on a stack, and, indeed, on multiple stacks.

Full backtracking. Both structuring and recursion complicate backtracking. In Grammar 2, consider the case of

i. $[\text{Economy}]_{NP}$ $[\text{tickets}]_{V}$ $[\text{cost}]_{NP}$ less money.

When "less money" cannot be parsed, we may have already exited NP/NP after parsing "cost". We must then backtrack fully through the $NP/_{cost}$ network to the state S/V, and from there back into the $NP/_{economy}$ net through NP/ADJP back to the first NP/ of the parse. **Full backtracking** referes to a parsers capability to backtrack in to already-parsed constituents.

It is possible to design transition network parsers in which backtracking is confined to the current network, but, as described below, this is not without other costs.

Local and augmented networks. Many natural language errors are **local** in scope. For example, in

ii. Three ticket costs thirty dollars

the number disagreement between "three" and "ticket" would occur wholly within the NP/ network of Grammar 2. It could be detected by comparing several local variables, say ADJ_NUM and N_NUM.

On the other hand,

iii. $[[\text{Tickets}]_{NP}$ costs thirty dollars$]_{S}$

illustrates a non-local disagreement. In Grammar 2, the pertinent variables, [call them N_NUM and V_NUM] would exist in separate networks [NP/ and S/]. The **augmented** transition network [Thorne, Bratley, and Dewar, 1968; Bobrow & Fraser, 1969; Woods, 1070] added a special class of variables called "registers" to the local transition network. Registers could be used to resolve such "long-distance dependencies".

I call ATN grammars "binary" because they force the grammar writer to attend to the pairwise relationship between two states linked by an arc. Like programming in low-level assembly code, this approach produces optimally fast and concise grammars, but at a corresponding cost in scholarly effort.

Generalizing the ATN Formalism

After all of the preceding features have been implemented, ATNs are still often syntax-centered, nondeterministic, right-branching, depth-first, top-down, rule-driven parsers. Generalized transition network parsers, in addition to other features, allow the preceding parsing strategies to be intermixed with cascaded, deterministic, ambiramiform, breadth-first, bottom-up, and data-driven strategies. Since ATN parsers are potential Turing machines, such generalizations have always been available. Some, if not all, have been implemented and reported previously [e.g., Kaplan, 1973; Woods, 1980]. The GPARS parser is called a GTN parser to distinguish it from those ATNs which less fully exploit the potential power of the ATN formalism.

Cascaded morphological parsing and lexical ambiguity. As noted above, English has a simple morphology, so morphological parsing has received little attention in the English-dominated parsing literature. In virtually all other languages, however, morphological parsing presents significant problems. In the case of inflected and agglutinating languages like Russian and Japanese, nearly all syntactic information can and must be recovered by parsing the morphological structure of words. Even the morphologies of "isolating", putatively uninflected languages can introduce serious indeterminacies into a parse. For example, in my Chinese system, Xuejiu, the morphological parser must decide whether several input ideographs should be taken as a single polysyllabic word, or whether five or six concatenated pinyin syllables should be decomposed into two or more smaller words. The latter task is especially nondeterministic because tone marks are customarily omitted in pinyin, making polysemy rampant.

Ambiguity is arguably the single biggest obstacle to syntactic parsing by computer, and in uninflected languages like English and Chinese polysemy is the greatest source of ambiguity. Parsers may choose either serial or pseudo-parallel approaches to resolving polysemy. In serial approaches, individual word senses are tried one at a time. If the parse blocks, the parser backtracks [frequently through the syntax, back into the morphological analysis] to try the next sense. In pseudo-parallel approaches, all senses are put on an ACTIVE_SENSE_LIST. All tests are applied to all senses on the list and senses which fail are deleted from the list. The pseudo-parallel approach can give good results where only one or two features of a sense are tested by the grammar [e.g., part-of-speech, number]. As tests and features increase, however, the cost of parallel testing can soon exceed the cost of backtracking.

GPARS uses a serial approach and relies on sense-ordering to minimize backtracking. For example, move__locact is ordered before move__suggest in the ENGPARS lexicon because it is by far the more frequently-used sense. ENGPARS accordingly tries the sense move__locact first. Similarly, compound nouns like New York City are ordered before new. If simple lookahead does not find the collocation York City directly following New in the input, the sense can be immediately rejected.

To accommodate nondeterministic morphological parsing and polysemy, the GPARS morphological parsing mechanism is fully cascaded into the syntactic system so that mixed parsing strategies and full backtracking can be maintained across morphosyntactic boundaries. GPARS morphological grammars take the same binary, GTN form described above. In the morphological context, however, many functions like (to) and (jmp) must be string functions rather than terminal and non-terminal symbolic functions. The GPARS system implements such functions, *mutatis mutandis*, within a separate morphological and syntactic closures.

Deterministic parsing. The cost of backtracking is limited in GPARS systems by several mechanisms. First, instructional parsers must be exceptionally tightly-constrained because of the incidence of learner errors. Secondly, GPARS tries to optimize backtracking speed through maximal use of TLC-LISP's native-code-supported control-stack and dynamic binding. Thirdly, GPARS implements a set of deterministic "cut functions": (xto), (xjmp), and (xseek). If an "x-arc" fails, the entire state fails. All subsequent arcs leaving the present state are ignored.

With these mechanisms, GPARS systems rarely back up more than three words unless a learner error is encountered. Since learner errors are to be expected in instructional parsers, several additional mechanisms have been implemented to constrain backtracking. First, well- and ill-formed phrase lists [described below] speed parsing by eliminating the need to reparse phrases which have already been parsed or rejected. Nevertheless, when a learner error occurs toward the end of a long and complex sentence, "backthrashing" can occur: The error can force the parser to backtrack repeatedly, searching nearly the entire grammar for a [nonexistent] combination of rules which will accept the input sentence. When the ratio of forced backtrackings to words parsed exceeds a backthrashing threshhold, GPARS aborts the parse.

LR[k] parsers [Knuth, 1965] eliminate backtracking by using a small "shift stack", and allowing the parser to look ahead k input units. This makes LR[k] grammar very efficient for applications like compiler design. Marcus [1980] developed such a "deterministic" LR[k] parser for parsing English.

If, however, backtracking is retained without lookahead, the Marcus/LR[k] parser becomes similar to a "shift-reduce" parser [Allen, 1987, pp. 166 ff.; Sato, 1988]. In shift-reduce parsers [the standard parser design for algebraic expressions], parsing begins as in any network parser. But where an ATN would immediately assign the just-parsed constituent to a role in the final parse tree structure, a shift-reduce parser holds the just-parsed constituent on a separate "shift stack" until the parser can look ahead [an arbitrary distance, with backtracking allowed]. After looking ahead, the parser can assign the just-parsed constituent's role more deterministically. Winston [1984] refers to this general strategy as "Wait-And-See Parsing" [WASP]. GPARS uses WASP mechanisms for parsing ambiramiform structures.

Parsing Ambiramiform Structures. The English possessive is a left-branching construction in an otherwise right-branching language. I call sentences like iv "ambiramiform".

iv. [E1] John's mother's cousin's brother is Fred.

The possessive construction in iv could be generated by a rule like v.

v. NP --> NP + 's + NP

Unfortunately, v is left-recursive. It traps the parser in an endless recursive loop before the 's term of the rule is ever reached. GPARS solves this by not (send)ing from NP/NP if * is bound to 's. Instead, the just-parsed NP [e.g., John's] is pushed onto a "shift stack", the original (seek 'np/) environment [e.g., E1] is reinstantiated, and the NP headed by mother is parsed. Before this next NP is closed, the shift stack is inspected, found to be non-empty, and John's is popped into the current NP as a modifier of mother. The process can be reapplied recursively. The actual GPARS implementation of the shift stack becomes somewhat more complicated by the stack asynchrony described in the next section, but it is accomplished with a simple (shift x) directive in the grammar.

Nested possessives are rare in English, and English parsers can usually parse them with kludges. Chinese, however, is a predominantly right-branching language whose relative clauses all branch left. Not only is the Chinese relative clause more common than the English possessive, it is also more complex. It was essential to develop shift-reduce mechanisms for Chinese, and this has made it convenient to use them for English and other languages as well.

Breadth-first parsing. English learners avoid using relative clauses [Schachter 1974]. Consequently, it is inefficient for ENGPARS to (seek 'relc/) in every NP. English relative clauses can begin with virtually any part of speech, and the search can be a long one. A rigorously top-down parser without full backtracking must seek a relative clause in every NP. Otherwise, once an NP is closed and popped from the control stack, there is no way to subsequently backtrack into the NP network. To allow premature, "provisional SENDs" the GPARS (seek) function copies its remaining tests and actions into a virtual state, and pushes this virtual state onto a NEXT_STATE_STACK. This allows the NP network to exit before a relative clause has been sought by jumping to the virtual state which is popped from NEXT_STATE_STACK. The main control stack continues to store the original calling state. If the "virtual" analysis fails, the parser can backtrack through the control stack into the provisionally-sent NP to (seek) a relative clause. As a consequence, however, the system control stack and the NEXT_STATE_STACK become desynchronized. GPARS uses parse tree registers to keep control synchronized.

Well-formed phrase lists and charts. GPARS systems normally keep a "well-formed phrase list". When a network SENDs, the registers of the just-parsed constituent are preserved on the well-formed list. Later, after backtracking, constituents which were previously parsed correctly do not need to be reparsed. [In the ambiramiform parsing described above, the "shift stack" also functions like an auxiliary well-formed phrase list.] Without a well-formed list, sentences with conjunctions and prepositions are particularly prone to "backthrashing" because of the many possible attachment points these parts of speech can have.

The GPARS well-formed phrase list mechanism also conserves stack space. When a well-formed phrase is parsed, its computation history can be popped from the stack. [This is an important consideration when running in the segmented architecture of the 808x where the system control stack is limited to 64K bytes.]

It is important to note that a phrase may be parsed, but it still may not be well-formed. If a phrase contains polysemous words whose senses have not all been examined by the parser, GPARS sets a dynamically-scoped well-formed register to NIL: The stack is not unwound [popped] and the phrase is not entered into the well-formed list. Similarly, if a phrase has been "provisionally sent" by the breadth-first strategy described above, the stack should not be unwound. In this case, it is the grammarian's responsibility to prevent unwinding by performing a (setr 'wf nil) within the grammar.

In addition to the well-formed phrase list, GPARS also implements an ill-formed phrase list. As mentioned previously, English relative clauses can begin with many different parts of speech, and a long search may be required before a search of the English relative clause network can be abandoned. The same is unfortunately true of the ubiquitous noun phrase. By also maintaining an "ill-formed phrase list", hypotheses rejected once in the course of a parse can be rejected out-of-hand after subsequent backtracking. An ill-formed register controls the ill-formed list in the same way the well-formed register controls the well-formed list.

Bottom-up parsing. A "chart" is a well-formed phrase list which holds all successfully-sent phrases, regardless of whether they contribute to the final parsed structure. Bottom-up parsers work by combining the phrases of a chart into successively higher order phrases. For example, NPs and ADVPs might be combined into clauses, and clauses subsequently combined into sentences. The GPARS system does not adopt a fundamentally bottom-up design because, for present objectives, it is too expensive to parse phrases which will never contribute to the final structure. Nevertheless, GPARS can maintain a "chart" for use when execution speed is not a factor. The full chart would be particularly useful for error analysis along lines proposed by Mellish [1989].

A major limitation of many traditional ATNs and top-down, nonbinary parsers is that they do not efficiently parse free word order languages. Although GPARS is not fundamentally bottom-up, the use of appropriate network designs, register swaps, shifts, and provisional sends at least makes GPARS sufficiently bottom-up to comfortably accommodate the free word order of Chinese and Russian.

In the case of Chinese, vi - viii are semantically equivalent, meaning Zhangsan ate [the] eggplant.

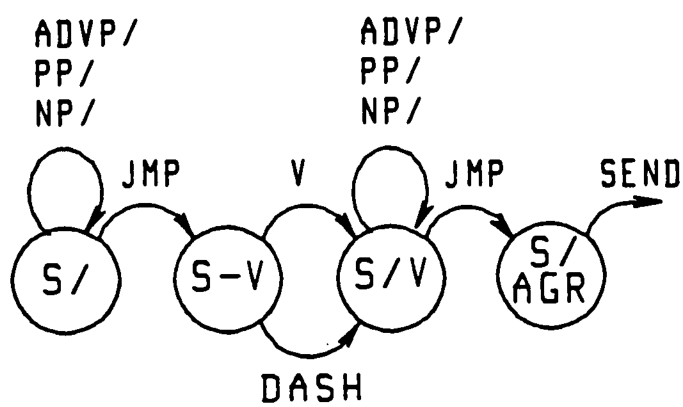
vi.	Zhangsan	chigwole	qiezi.
	Ag	V	Pat
	Zhangsan	ate	eggplant
vii.	Zhangsan	qiezi	chiguole.
	Ag	Pat	V
	Zhangsan	eggplant	ate
viii.	Qiezi	Zhangsan	chigwole.
	Pat	Ag	V
	Eggplant	Zhangsan	ate

On the basis of syntax alone, there is no way for the parser to avoid interpreting viii like vii and producing ix:

ix.	Qiezi	Zhangsan	chigwole.
	Ag	Pat	V
	Eggplant	Zhangsan	ate

However, upon detection of a case-frame error in the semantically-anomalous parse of ix, the GPARS Chinese parser can simply swap the contents of the agent and patient registers.

Russian presents even freer word order, and virtually all major constituents of a Russian sentence can be permuted. Our solution here has been to employ a "bottom-up" network similar to Network Grammar 3.



Network Grammar 3. Abstract of the RUSPARS S-network.

Network Grammar 3 admits major sentence constituents in virtually any order, but only admits one verbal nucleus to each sentence. The S/AGR state at the end of our actual RUSPARS network is a many-state cascaded subnetwork. It functions much like the functional component of a lexical-functional parser to reconcile the sentence's parsed, heavily-inflected constituents with their governing verbs.

Data-driven parsing. Many lexical items behave idiosyncratically. Compare, for example,

x. She has personality.
xi. She has a cute personality.

but,

xii. *She has cute personality.
xiii. ?She has exceptional personality.

It appears that <u>personality</u> could be minimally restricted to require a determiner, at least when it is modified by a monosyllabic adjective. Rather than burden the grammar with such details, a DEMON is marked in the lexical entry of <u>personality</u> [Listing 1].

```
(personality character trait
             n
             ( ... )
             ( ...
                (demon (t (eq state 'nphd/nphd)
                          (or (not (word (getr 'head)
                                        'personality) )
                              (not (getr 'adj) )
                              (getr 'det) ) )
                       (a (jmp state) ) )
                )
             ()
             )
```

Listing 1. A lexical demon.

FURTHER RESEARCH

Instructional parsers can contribute significantly to more efficient humanistic language study, but to do so they must yield in-depth parses despite the additional ambiguities inherent in learners' error-prone language. Moreover, they should do so on the smallish computers which teachers and students can afford. Transition networks yield compact grammars and fast parsers which can be generalized to answer this challenge for a variety of natural languages.

In the last few years, there has also been increased scientific interest in the parsing of ill-formed input. GPARS currently only supports a largest-left-corner strategy similar to that described in Weischedel & Sondheimer [1983]. Many more sensitive error-identification strategies are possible [e.g., Mellish 1989]. As microcomputers become more powerful, it will become increasingly cost-effective to implement more sophisticated error-handlers.

The greatest need facing all efforts at more sophisticated instructional parsing is for larger and more detailed parser-useable dictionaries. Research on machine-translating standard, printed dictionaries into machine-useable form is an important and promising current research topic [cf. Neff & Boguraev, 1989]. Despite the best MT results, years of research may still be needed before the myriad nuances of words like personality are adequately coded.

Finally, the output of competent parsers must also be carefully analyzed to facilitate better understanding of the language learning process. No matter how sophisticated our understanding of the parallel processes of thought become, we will have to communicate that understanding serially: Serial parsers and computational grammars represent our most highly-evolved means of communicating our evolving understanding. In the end, the fruits of computer-assisted language learning research could equal or exceed the direct instructional contributions of instructional parsers.

REFERENCES

Allen, James. Natural Language Understanding. Menlo Park, CA: Benjamin/Cummings, 1987.

Allen, John. The Anatomy of LISP. Englewood, NJ: Prentice-Hall, 1978.

Allen, John. TLC-LISP/86. Redwood Estates, CA: The LISP Company, 1985.

Bates, Madeleine. "The Theory and Practice of Augmented Transition Networks." In Natural Language Communication with Computers, Ed. L. Bolc. New York: Springer Verlag, 1978.

Bobrow, D.G. and B. Fraser. "An Augmented State Transition Network Analysis PrOcedure. IJCAI, 1 (1969), 557-67.

Fillmore, C.J. "The Case for Case." In Universals in Linguistic Theory, Ed. E. Bach and R. Harms. New York: Holt, Rinehart, and Winston, 1968, 1-90.

Grossberg, S. "On the Serial Learning of Lists." Mathematical Biosciences, 4 (1969), 201-253.

Grossberg, S. "The adaptive self-organization of serial order in behavior: speech, language, and motor control." In Pattern Recognition by Humans and Machines, Ed. E. C. Schwab and Howard C. Nusbaum. Orlando: Academic Press, 1986, 187-295.

Grossberg, S. "How Does a Brain Build a Cognitive Code?." Psychological Review, 87 (1980), 1-51.

Kaplan, R. "A General Syntactic Processor." In Natural Language Processing, Ed. R. Rustin. New York: Algorithmics Press, 1973.

Knuth, D. E. "On the Translation of Languages from Left to Right." Information and Control 8 (1965), 607-639.

Loritz, D. "A Digitized Speech Laboratory for under $2,000." In Methods II, Ed. K. Odwarka. Cedar Falls, IA: U. of

Northern Iowa, 1983.

Loritz, D. "The Parsing of Learner Sentences." Demonstration presented at the 18th Annual TESOL Convention. Houston, TX, March 1984.

Loritz, D. Ms. Pluralbelle. Arlington, VA: H. C. Enterprises. 1989.

Marcus, M. A Theory of Syntactic Recognition for Natural Language. Cambridge, MA: The MIT Press, 1980.

Mellish, C. S. "Some Chart-Based Techniques for Parsing Ill-formed Input." Proceedings of the Association for Computational Linguistics. 1989, 102-109.

Neff, M. S. and B. K. Boguraev. "Dictionaries, Dictionary Grammars and Dictionary Entry Parsing." Proceedings of the Association for Computational Linguistics. 1989, 91-101.

Sanders, A. and R. Sanders. "Syntactic Parsing: A Survey." Computers and the Humanities, 23,1 (1989), 13-30.

Sato, Paul T. "A Common Parsing Scheme for Left- and Right-Branching Languages. Computational Linguistics 14, 1 (1988), 20-30.

Schachter, J. "An Error in Error Analysis." Language Learning 24 (1974), 205-14.

Thorne, James, P. Bratley, and Harnish Dewar. "The Syntactic Analysis of English by Machine." Machine Intelligence 3. New York: Elsevier, 1968.

Wagner, R. TLC-LISP/386. Minneapolis, MN: Seeker Technologies, 1989.

Weischedel, R. and N.K. Sondheimer. "Meta-rules as a Basis for Processing Ill-Formed Input." AJCL 9, 3-4 (1983), 161-177.

Winograd, T. Language as a Cognitive Process. Vol. 1: Syntax. Reading, MA: Addison-Wesley, 1983.

Winston, Patrick Henry. Artificial Intelligence (Second Edition). Reading, MA: Addison-Wesley, 1984.

Woods, W.A. "Transition Network Grammars for Natural Language Analysis." Communications of the ACM, 13 (1970), 591-606.

Woods, W.A. "Cascaded ATN Grammars," AJCL, 6, 1 (1980), 1-12.


17 89

NOTICE

REPRODUCTION BASIS

☐ This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☑ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").