INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106
Obermeier, Klaus Karl

TEMPORAL INFERENCES IN COMPUTATIONAL LINGUISTIC INFORMATION PROCESSING

The Ohio State University

Ph.D. 1984

University Microfilms International

300 N. Zeeb Road, Ann Arbor, MI 48106

Copyright 1984 by Obermeier, Klaus Karl

All Rights Reserved
TEMPORAL INFERENCES IN
COMPUTATIONAL LINGUISTIC INFORMATION PROCESSING

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Klaus Karl Obermeier, M.A.

* * * * *

The Ohio State University
1984

Reading Committee:
Arnold M. Zwicky
B. Chandrasekaran
Michael Geis

Approved By

[Signature]
Adviser
Department of Linguistics
Copyright by
Klaus K. Obermeier
1984
DEDICATION

to

Arnold and Ann Zwicky
VITA

February 5, 1954 . . . . . . . Born – Landshut, West–Germany
1973 . . . . . . . . . . . . . . Abitur, Hans–Carossa–Gymnasium, Landshut, West–Germany
1973–1975 . . . . . . . . . . . . Student, University of Regensburg
1975–1976 . . . . . . . . . . . . Student, University of Nuernberg
1976–1978 . . . . . . . . . . . . Exchange Scholar, University of Kansas
1978–1984 . . . . . . . . . . . . Teaching Associate, Department of Linguistics, The Ohio State University, Columbus, Ohio
1980 . . . . . . . . . . . . . . M.A., The Ohio State University, Columbus, Ohio
1980 . . . . . . . . . . . . . . Staatsexamen, The University of Regensburg, West–Germany
1983–1984 . . . . . . . . . . . . Research Scientist, Battelle Memorial Institute, Columbus, Ohio

PUBLICATIONS

"Die Bedeutung des Zweifels als Grundlage des Wissens in der Existenzphilosophie."


FIELDS OF STUDY

Major Field: Linguistics

Studies in Psycholinguistics: Professors Melissa Bowerman and Robert Fox

Studies in Pragmatics: Professors Arnold M. Zwicky, Herbert Brekle, Philip Luelsdorff, and Sally Sedelow

Studies in Artificial Intelligence: Professor B. Chandrasekaran

Studies in Syntax/Semantics: Professors David R. Dowty and C. K. Oh
# LIST OF FIGURES

| Figure 1-1: | Sample Text for Case No. 172556 | 7 |
| Figure 1-2: | Syntactic Annotation for Sentence [1] in Figure 1-1 | 8 |
| Figure 1-3: | A [Simplified] Sample Output of the Representation for Sentences [1], [2], and [3] from Figure 1-1 | 9 |
| Figure 2-1: | Functional Perspective | 24 |
| Figure 2-2: | Dominant Problem Solving Paradigm | 31 |
| Figure 2-3: | A NLP With Distributed Problem Solving Specialists | 32 |
| Figure 2-4: | Stratified Model | 35 |
| Figure 2-5: | Blackboard Model | 36 |
| Figure 2-6: | Differences in Processing Strategies | 37 |
| Figure 3-1: | The Morphological Analyzer | 46 |
| Figure 3-2: | Control in the Morpho-analyzer | 48 |
| Figure 3-3: | Parse-Procedure | 50 |
| Figure 3-4: | X-bar structures from [Jackendoff 77, 17] | 52 |
| Figure 3-5: | X-bar Grammar for GROK | 53 |
| Figure 6-1: | The two interface components of a NL program | 98 |
| Figure 6-2: | Storage of medical information | 99 |
| Figure 6-3: | Modular versus Nonmodular Systems | 102 |
| Figure 7-1: | Lexical Access of Sentence [1] in Figure 1-1 | 107 |
| Figure 7-2: | Trace of Sentence [1] in Figure 1-1 | 109 |
| Figure 7-3: | Final Knowledge Representation of Event | 109 |
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDICATION</td>
<td>ii</td>
</tr>
<tr>
<td>VITA</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td>1 NLP - A Sample Problem.</td>
<td>4</td>
</tr>
<tr>
<td>1.1 A Sample Text.</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Scenario.</td>
<td>6</td>
</tr>
<tr>
<td>1.3 Problems.</td>
<td>10</td>
</tr>
<tr>
<td>1.3 Solutions.</td>
<td>12</td>
</tr>
<tr>
<td>1.3 Outline</td>
<td>13</td>
</tr>
<tr>
<td>2 GROK - A NLPS for Medical Texts.</td>
<td>15</td>
</tr>
<tr>
<td>2.1 Design of GROK -</td>
<td></td>
</tr>
<tr>
<td>2.1.1 Grammatical Representation of Objective Knowledge</td>
<td>17</td>
</tr>
<tr>
<td>2.1.2 The Flow of Control</td>
<td>20</td>
</tr>
<tr>
<td>2.1.3 The System.</td>
<td>23</td>
</tr>
<tr>
<td>2.2 Implementation</td>
<td>26</td>
</tr>
<tr>
<td>2.2.1 The Representation of Knowledge</td>
<td>26</td>
</tr>
<tr>
<td>2.2.1.1 Frames</td>
<td>27</td>
</tr>
<tr>
<td>2.2.1.2 Production Rules</td>
<td>29</td>
</tr>
<tr>
<td>2.2.2 The Use of Knowledge</td>
<td>31</td>
</tr>
<tr>
<td>2.3 Issues</td>
<td>33</td>
</tr>
<tr>
<td>2.3.1 Pragmatic Parsing</td>
<td>33</td>
</tr>
<tr>
<td>2.3.2 Temporal Pragmatics</td>
<td>38</td>
</tr>
<tr>
<td>3 Parser</td>
<td>45</td>
</tr>
<tr>
<td>3.1 Morphological Module</td>
<td>46</td>
</tr>
<tr>
<td>3.2 Lexicon</td>
<td>47</td>
</tr>
<tr>
<td>3.3 Syntactic Module</td>
<td>50</td>
</tr>
<tr>
<td>3.4 X-Bar Syntax</td>
<td>51</td>
</tr>
<tr>
<td>3.5 Comparison to Other Parsers.</td>
<td>52</td>
</tr>
</tbody>
</table>

vii
8.2 Methodological Contributions .................................................. 121
8.3 State of Implementation .......................................................... 122
9 Conclusion ..................................................................................... 125

BIBLIOGRAPHY ............................................................................... 128

APPENDIXES
A. ELISP Code for Parsing Procedure .............................................. 144
B. EFRL Code for Frame Structures ................................................. 167
C. Computer Examples ..................................................................... 174
INTRODUCTION

Just as the domain of biology includes something like all living things (for which a strict definition is probably impossible outside of biological theory), so the domain of cognitive science may be something like "knowing things," or, as George Miller [Miller 84] colorfully dubbed it, the "informavores." [Pylyshyn 84, xi]

Natural Language Processing [NLP] is that field of study within cognitive science which deals with linguistic behavior. Research of linguistic behavior has proceeded within the theoretical and methodological confines of the disciplines which concern themselves with language: linguistics, Artificial Intelligence [AI], psychology and philosophy.

NLP has been a controversial issue for AI and linguistics: AI research was said to be practical, but untheoretical, whereas research within linguistics was said to be theoretical, but impractical [Goodwin 82, 249].

Research in linguistics over the past twenty years has been under the influence of the Chomskian paradigm of generative grammar which tried to provide a rigorous and concise description of Language. The goal of Chomsky and his followers was to decide on formal grounds if a (grammatical) sentence belonged to a language or not. According to Chomsky's theory, the linguistic system can be divided into a native speaker's competence and performance. The realm of generative grammar is to study the linguistic competence of an idealized speaker, i.e., the rules which determine the grammaticality of a particular sentence. Linguistic performance, viz. how language is comprehended and produced belonged in the realm of psycholinguistics.

Research in AI has been based on the assumption that language should be studied by looking at linguistic processes observed in human linguistic behavior. Schank summarized
the goal of computational NLP: "Natural language researchers within AI thus had to come to grips with the fact that they would have to build their own theories of the linguistic process." [Schank 81, 4] Since humans are still the only organisms which understand language with all its complexities, NL research within AI has been dominated by the modelling of linguistic performance.

The gap between AI and linguistics widened when it came to determining what phenomena of language understanding should be investigated and how they could be tested. For the Chomskian linguist, hypotheses about language were confined to phenomena within single sentences; for the AI researcher, hypotheses about language encompassed all aspects of cognitive processing. Consequently, linguists could establish well-defined formal criteria for testing their hypotheses on carefully selected fragments of English, whereas the AI researchers had to resort to the Turing test or a vague notion of understanding as in the following excerpt:

A computer understands a subset of English if it accepts input sentences which are members of this subset and answers questions based on information contained in the input. [Bobrow 68, 146]

Criticism on Chomsky's syntactically oriented theory arose, however, from within linguistics. After a short and unsuccessful attempt by the generative semanticists to extend Katzian lexical semantics, Montague and later Barwise and Perry proposed formal semantic accounts for fragments of English. Montague grammar has led to an impasse because his strategy seems to have been "to devise a construct in the formal language for each construct in the natural language" [Brady 83, xxii] which thus far has not been applied to a significant fragment of any natural language.

The current state of affairs in NLP is characterized by on-going changes both within linguistics and AI. In linguistics, an increasing number of researchers follow Gazdar's contention that NL can be adequately described in terms of a CF-grammar [Gazdar 81] which has led to the GPSG framework. Generalized phrase structure rules provide a useful mechanism for computational information processing. In their methodology, linguists now appeal to discourse principles where they had tried to use convoluted
syntactic rules and regulations before [Sag 84, Zwicky 84]. In general, the trend of linguistic research points away from Chomskian or Montague-style competence theories [Goodwin 82, 243].

In AI, research is under way to provide criteria of how to evaluate computational theories within cognitive science [Doyle 83, Pylyshyn 84] and the implemented knowledge representation techniques [Touretzky 84]. These endeavors will lead to a better understanding of the programs and their theoretical significance. AI can draw on results from the current investigation of linguistic methodology, viz., what information processing tasks the various modules of a system perform. AI will also profit from the change in linguistics from transformational grammar to phrase structure grammar which is computationally more tractable.

My dissertation contributes to the interdisciplinary study of language by providing the NL researcher with a Natural Language Processing System [NLPS] for processing descriptive text, which is driven by pragmatic principles and based on a generally accepted syntactic theory, viz., X-bar syntax [Jackendoff 77]. Moreover, I have provided a knowledge-representation scheme for dealing with temporal information within a well-defined type of text.
CHAPTER 1

NLP - A SAMPLE PROBLEM

This dissertation describes the development of a natural language [NL] processing system [NLPS] for analyzing written text. The analysis, i.e. output, of the NLPS is a data structure which serves as the input to an expert system. The ultimate goal is to allow the user of the expert system to enter data into the system by means of NL text which follows the linguistic conventions of English.

The particular domain chosen to illustrate the underlying theory of such a system is that of medical descriptive texts which deal with patients' case histories of liver diseases. The texts are taken unedited from the Journal of the American Medical Association. The information contained in those texts serves as input to PATREC, an intelligent database assistant for MDX, the medical expert system. The objectives of this research are twofold, whereby the system described above is meant to be a particular implementation of a general NLP which could be used for a large variety of domains.

The first objective is to argue for a knowledge-based approach to NL processing in which the parsing procedure is driven by extra linguistic knowledge. The second objective is to provide a theory for processing temporal information contained in a given text.

This research relates to three areas of cognitive science: linguistics, psycholinguistics and artificial intelligence.

1 I use the term Natural Language Processing System to refer to the active agent - human or machine - in the understanding process, to make clear that my discussion is limited to cognitive activities involving NL use as an example of a cognitive problem-solving task.
- It provides the *linguist* with parsing techniques for text analysis which incorporate syntax, semantics, pragmatics, and knowledge about the domain.

- It provides a "mental model" of written language comprehension for the *psycholinguist*.

- It provides the researcher in *artificial intelligence* with a knowledge representation scheme for natural language processing.

This introductory chapter presents a general discussion of a sample text to show what the capabilities of the system are, as well as how it performs the given task. In the following chapters, details of the design and implementation of my system as well as its theoretical importance for the study of language are given.

1.1. A SAMPLE TEXT

The user of my NLPS can enter a text of the format given in figure 1-1\(^2\). The texts which the NLPS accepts are descriptive for a particular domain. The information-processing task consists of the analysis of linguistic information into datastructures which are chronologically ordered by the NLPS.

The first module of the program analyzes each single word by accessing a lexical component which assigns syntactic, semantic, and conceptual features to every word. The second module consists of a bottom-up parser which matches the output from the lexical component to a set of augmented phrase structure rules\(^3\). The third module consists of a knowledge base which contains the domain-specific information as well as temporal knowledge. The knowledge base is accessed during the processing of the text in conjunction with the augmented phrase structure rules.

---

\(^2\) The numbering on the sentences is only for ease of reference in the following discussion and does not appear in the actual text.

\(^3\) The augmentation consists of rules which contain knowledge about morphology, syntax, and the particular domain in which the NLPS is operating. These rules are used for interpreting the text, in particular, ambiguities, as well as for generating the final output of the NLPS.
The output of the program includes a phrase-structure representation as given in figure 1-2, and a knowledge representation as provided in figure 1-3. The phrase-structure representation provides the syntactic analysis of the sentence within the framework of X-bar syntax (See 3.4. for a brief discussion of the theory and my syntactic output). The resulting knowledge representation of my NLPS consists of a series of events which are extracted from the text and chronologically ordered by the NLPS based on the stored knowledge the system has about the domain and general temporal relations. The final knowledge representation which my NLPS generates is the input to the expert system or its database specialist. The final output of the expert system is a diagnosis of the patient.

1.2. SCENARIO

The comprehension of a descriptive text requires various types of knowledge: linguistic knowledge for analyzing the structure of words and sentences; "world knowledge" for relating the text to our experience; and, in the case of technical texts, expert knowledge for dealing with information geared toward the specialist. I contend in my dissertation that the comprehension of technical, descriptive text is simply a conversion of information from one representation into another based on the knowledge the NLPS has.

If a doctor were given a patient's case history [see figure 1-1] , he would read the text and try to extract the salient pieces of information which are necessary for his diagnosis. In this particular text type, he would be interested in the sign, symptoms, and laboratory data, as well as the medical history of the patient. The crucial point hereby is the temporal information associated with the occurrences of these data. In general, he would try to cluster certain abnormal manifestations to form hypotheses which would result in a coherent diagnosis. The clustering would be based on the temporal succession of the information in the text. Each manifestation of abnormalities I will refer to as an "event". Each event is defined and related to other events by means of temporal information explicitly or implicitly provided in the text. An important notion which I
1. This 80-year-old Caucasian female complained of nausea, vomiting, abdominal swelling, and jaundice.

2. She had diabetes mellitus, treated with insulin for six years before admission.

3. She had had ill-defined gastrointestinal complaints for many years and occasional episodes of nausea and vomiting three years previously.

4. Four weeks before admission she developed pain across the upper abdomen, radiating to the flanks.

5. She also complained of shooting precordial pains and palpitation with slight exertional dyspnea.

6. Increasing weakness and constipation developed, with bowel movements every second or third day.

7. Two weeks before admission she lost her appetite and began to vomit frequently immediately after meals.

8. There was little nausea and no hematemesis.

9. She recognized no retained food in the vomits.

10. Her abdomen became prominent, and one week before admission jaundice was noticed.

11. The urine became dark, but the stools did not change in color.

12. The patient lost 20 pounds in weight the month before admission.

Figure 1-1: Sample Text for Case No. 172556
This partial parse of the sentence follows Jackendoff's X-bar theory [Jackendoff 77], which is discussed in 3.4.; roman numerals indicate the number of bars assigned to each phrase. Comments to the parse were made after the actual run of the program.
EVENT1

SYMPTOM: NAUSEA/VOMIT/ABDOMINAL/SWELLING/JAUNDICE
KEY EVENT: ADMISSION
DURATION: ADMISSION

EVENT2

SYMPTOM: DIABETES MELLITUS
KEY EVENT: ADMISSION
RELATION TO KEY EVENT: 6 YEARS BEFORE
DURATION: SIX YEARS

EVENT3

SYMPTOM: GASTROINTESTINAL COMPLAINT
KEY EVENT: ADMISSION
RELATION TO KEY EVENT: MANY YEARS
DURATION: MANY YEARS

EVENT4

SYMPTOM: NAUSEA/VOMIT
KEY EVENT: ADMISSION
RELATION TO KEY EVENT: 3 YEARS BEFORE
DURATION: INTERMITTENT

Figure 1-3: A [Simplified] Sample Output of the Representation for Sentences [1], [2], and [3] from Figure 1-1
use in my program is that of a key event. "Events are organized around _key events_ (which are domain-specific – in the medical domain, some of the important ones are 'admission', 'surgery', 'accident', etc.), so that other events are typically stated or ordered with respect to these key events" [Mittal 82].

The purpose of my program is to simulate the chronological analysis of events in patient’s case histories which in turn will be the input to the expert system. Moreover, the general applicability of such an analysis for different domains are discussed.

### 1.3. PROBLEMS

The inherent problems of text comprehension from an information processing viewpoint are how to deal with the foremost problems in computational NLP (e.g., ambiguity, anaphora, ellipsis, conjunction), including the foremost problems in temporal information processing (e.g., implicit time reference, imprecision of reference).

In particular, in the sample text in figure 1–1, the following computational linguistic problems arise while parsing the sentences:

- a NLPS has to distinguish the different uses of inflectional morphemes (e.g., _ing, ed_). The suffix _ing_ attached to a verb indicates the gerund in (1), a present participle in (4), and a gerundive in (5). It can also lead to global ambiguities as seen in the well-known example _Flying planes can be dangerous_. The NLPS has to analyze the words and their specific context to determine the correct analysis of the sentences.

- a NLPS has to successfully disambiguate word-class ambiguities as seen in (1) where the word _female_, interpreted in isolation, can be either an adjective or a noun. The NLPS has to analyze the phrase-structure of the sentence and the context in which a word is used to determine to which class it belongs.

---

5 Events and the function of key events are further discussed in 4.4.2.

6 Cf. 8.1.2.

7 For a detailed analysis and discussion of the computational linguistic problems see 8.1; for a thorough discussion of a sample parse see 2.1.2. and chapter 7.
a NLPS has to distinguish between different types of conjunctions: the conjunction and combines either noun phrases as in (1), or clauses as in (3). The NLPS has to identify the syntactic, semantic, and conceptual features of the words to determine how to conjoin them according to their occurrence in a sentence.

- a NLPS has to interpret deictic references (e.g., this in (1)) as well as anaphoric references (e.g., she, there). The NLPS has to select the proper antecedent from the context to analyze anaphoras correctly.

The computational linguistic problems require a parser which has access to linguistic as well as extra-linguistic knowledge. While knowledge-based parsing has been proposed in particular by Schank [Schank 77] and his students, there is still an ongoing debate of what types of knowledge are necessary for a NLPS and how the knowledge should be represented and accessed. In particular, the discussion between AI researchers and linguists, who investigate language understanding, focuses on how much syntactic knowledge is necessary for a NLPS to comprehend natural language, and when syntactic knowledge should be used.

Within AI and computational linguistics, not many theories have been proposed for the processing of temporal information. In particular, a theory of how a NLP can comprehend temporal relations in a written text is still missing. In my dissertation, I present a theory for processing temporal information in a NLPS for a well-defined class of technical descriptive texts. The texts deal with a specific domain and tasks which require the processing of linguistic information into a chronological order of events. The problems for processing the temporal information contained in the text include:

- a NLPS has to work with implicit temporal information. Although in (1), (5), (6), (8), and (9), no explicit temporal reference is present, the NLPS has to detect the implied information from the context and the extra-linguistic knowledge available.

- a NLPS has to work with fuzzy information. The reference to for many years in (3) is fuzzy, and yet a NLPS has to relate it to the chronology of the case.

- a NLPS has to order the events in their chronology although they are not temporally ordered in the text.
1.4. SOLUTIONS

My solution to the problems discussed in the previous section lies within the computational paradigm as opposed to the Chomskyan generative paradigm. The computational paradigm focuses on how the comprehension processes are organized whereas within the generative paradigm, linguistic performance is of less importance for a linguistic theory than linguistic competence. Within the computational paradigm, the representation and use of extralinguistic knowledge is a major part of studying linguistic phenomena, whereas generative linguists separate linguistic phenomena which fall within the realm of syntax from other cognitive aspects [Winograd 83, 21].

Functionality is the central theoretical concept upon which the design of my NLPS rests. What is important for comprehending language is the function of an utterance in a given situation. Words are used for their meaning, and the meaning depends on the use in a given context. The meaning of a word is subject to change according to the context, which is based on the function of the words that make up the text. Therefore, my approach to building a NLPS focuses on modeling the context of a text in a particular domain. I am primarily concerned with the relationship between writer–text–reader, rather than with the relationship between two sentences. The use of the context for parsing requires a knowledge representation of the domain, and the type of text, in addition to linguistic and empirical knowledge.

In contradistinction to NLPSs which use syntactic information first [Robinson80, Thompson 81], and which possibly generate unnecessary structural descriptions, my system uses higher level information (e.g., domain, text–type) before and together with usually a smaller amount of syntactic information. In my system, the syntactic information selects between contextually plausible interpretations of the text — syntax acts as a filter for the NLPS.⁸

⁸ A detailed comparison between my NLPS and others is given in 4.5.
In contradistinction to NLPSs which use conceptual information first [Schank 77, Wilks75] my system, partially due to the limited information processing task and the particular domain, starts out with a small knowledge base and builds up datastructures which are used subsequently in the processing of the text. The knowledge base of my system contains only the information it absolutely needs, whereas Schankian scripts have problems with when to activate them and when to exit them. Besides these theoretical issues, methodological differences between my approach and other systems are apparent.

Modularity is the key methodological concept in my NLPS. However, by itself, modularity does not require strictly linear processing strategies: "we should bear in mind the happy compatibility of theories of static modularity with processing models incorporating high degrees of dynamic interaction" [Brady 83, xix]. Modularity in my system means to identify different types of problem solving mechanisms in different modules of the NLPS. The interaction between the various modules is governed by an explicit control structure. Within the computational paradigm, linguistic information processing can also be explained modularly by separating computational and representational issues. The major advantage of a modular system lies in the possibility of formulating constraints for individual modules which are of explanatory value.

1.5. OUTLINE

My investigation of these issues proceeds as follows:

In chapter 2, I provide an overview of the system as well as a discussion of my approach to parsing and temporal information processing.

---

9 Further discussion is given in 2.2.1.1.

10 For further discussion of methodological issues, see 8.2.
In chapter 3, the parsing components are described in detail, and my technique is compared to extant parsing systems with respect to the linguistic, computational, and cognitive theory being used.

Chapter 4 contains a discussion of temporal information processing within my system.

In chapter 5, the notion of "inference" is discussed. Temporal on-line inferences for building up a knowledge representation are distinguished from temporal inferences which operate on the knowledge representation.

In chapter 6, the applicability of my system as NLPS for MDX is discussed.

Chapter 7 contains an annotated trace of a sample parse.

Chapter 8 contains a summary of the contributions to the study of language in my dissertation.

In chapter 9, the results of my dissertation are discussed with respect to general issues of computational NLP.
CHAPTER 2

GROK -
A NLPS FOR MEDICAL TEXTS

In this chapter, a technical discussion of my NLPS is given. I first discuss the
general requirements of a NLPS for written texts, before analyzing in detail the design
and implementation of my NLPS.

The NLPS for a particular type of written text within a specific domain must be
capable of converting information from its linguistic representation into a conceptual
representation which the expert system can use for its specific problem-solving task.
The information-processing task of the NLPS requires a knowledge representation which
allows the system to compute the information given in the text, more specifically, to
keep track of things mentioned in the text. When the NLPS analyzes the text, it has
to have knowledge about

- the grammatical rules of the language, e.g., morphological and syntactic rules;
- the domain, e.g., domain-specific definitions of concepts and events;
- the text-type, e.g., descriptive text;
- the purpose of the analysis, e.g., the format of the final representation; and
- general concepts, e.g. "time" and "space".

Based on these knowledge sources, the NLPS has the following five characteristics:

1) Interaction of Knowledge Sources: To adequately convert all the information
provided in the text into the final datastructure, the system has to draw on linguistic
and extralinguistic knowledge. The interaction between the knowledge sources divides the
linguistic information processing task into a representational and a computational component – how the knowledge is stored and how it is retrieved are separate issues.

(2) **Types of Problem Solving**: NLPSs ordinarily are designed to solve a single problem in a well-defined area, e.g., ship maintenance, geological exploration. If the problem requires only a monostatal approach (as opposed to a modular one), the NLPS can merge either the discourse knowledge with the database knowledge (e.g., *LUNAR* [Woods 72]), or the linguistic with the domain knowledge (e.g., *LADDER* [Hendrix 78], *PLANES* [Waltz 78]). The modular systems process information on separate levels. The modules operate on different types of information using different types of problem solving (e.g., classifying and chronologically ordering events require two different types of problem solving).

(3) **Constraint Utilization**: NLPSs constructed within a limited time frame cannot encompass an extensive amount of "world knowledge", nor for that matter, can they solve the "AI problem" of how to build "intelligent" machines. They can be constructed to utilize the inherent constraints of a given task within a specified domain. The utilization of these inherent constraints indicates which of the problem-solving types are specific to a domain as well as which general problem solving types will be useful. Restricting the amount of knowledge adds an explanatory constraint to the underlying theory. Defining and limiting the necessary concepts and processes for the information processing task is part of the explanation of a particular phenomenon.

(4) **Coping with Ambiguity**: In NLPSs, ambiguity stems from implicit, incomplete, or incompatible information. NLPSs resolve ambiguities either by statistical means, or by using redundant knowledge from other sources.

---

11 Further discussion on extant modular NLPS is given in 6.1.; modularity is also discussed in 2.5.

12 Examples for using statistical methods to resolve uncertainty caused by incomplete information in the parsing process are *DIAGRAM* or *EPISTLE*, which resolve the combinatorial explosion of generated parses using a metric for selecting optimal parses. Since they use only syntactic knowledge initially, the information they can draw on is limited and thus causes uncertainty in the final analysis.
(5) **Extensibility and Maintenance**: NLPSs should be extensible to other domains and accommodate the addition of new knowledge. The modularity of knowledge is necessary for maintaining and updating the system.

### 2.1. DESIGN OF GROK - GRAMMATICAL REPRESENTATION OF OBJECTIVE KNOWLEDGE

NLPSs require a large knowledge base. There have been few attempts at developing a methodology for constructing a NLPS in a "principled way" because of the following problem:

It is well-known that the interpretation of natural language discourse can require arbitrarily detailed world knowledge and that a sophisticated natural language system must have a large knowledge base. But heretofore, the knowledge bases in natural language systems have either encoded only a few kinds of knowledge - e.g., sort hierarchies - or facts in only very narrow domains [Hobbs 84, 283].

Hobbs suggests a methodology for setting up a knowledge base by postulating that "facts are selected for the knowledge base by determining what facts are linguistically presupposed by a text in the domain of interest" [Hobbs 84, 283]. I selected the facts for the medical sublanguage from Sager's *Linguistic String Project* [Hirschman 83]. I adopted her methodology for selecting sublanguage classes of facts for the clinical sublanguage:

The sublanguage of clinical reporting, as evidenced in a typical patient document, conveys information about the patient's state and medical actions taken in connection with that state. The word classes for this material were developed by examining and comparing sets of words occurring in particular syntactic environments [Hirschman 83, 29-30].

To illustrate the selection process, consider the sentences (1) and (2):

---

13 I define "fact" as an entity relevant for an information processing task.

14 "We define sublanguage here as the particular language used in a body of texts dealing with a circumscribed subject area [often reports or articles on a technical speciality or science subfield] in which the authors share a common vocabulary and common habits of word usage" [Hirschman 83, 27].
(1) The patient developed fever.

(2) The patient had jaundice.

From this pattern, two word-classes for the clinical sublanguage can be formed: the class "sign/symptom" in object position, the class "report-symptom" for the verbs. For my particular NLPS, I use only about ten word-classes as facts for my knowledge base (as opposed to Sager who uses about 60 sublanguage-specific word-classes). In my NLPS, I extend Hobbs' principle in that facts are also selected by determining what the output structure is. After defining the task of my NLPS, I select the facts based on what I expect from the domain (e.g., liver disease is a thing expected from the domain); I organize the facts, based on what I know about the particular text-type (e.g., patient case histories); and I add the facts which are needed for my target representation (e.g., events connected to a "key event")16. I assume that an event is a basic fact for temporal information processing. I have added "event" as a fact to my knowledge base as a domain-independent concept which is necessary for temporal processing17.

The final step in the design is to outline the control structure of the NLPS which determines how the knowledge is used.

15 Each sublanguage class can contain different parts of speech: the word-class "Exam-Test", defined as "test or technique used by doctor during physical exam", includes percussion, palpable, touch [Hirschman 83, 76]

16 For the definition of "key event" see 1.2

17 Cf. 4.2.2.
2.1.1. Facts - Selection and Organization for the Knowledge Base

In chapter 1, I characterized the task of a doctor reading a patient's case history as finding key domain concepts (e.g., sign, symptom, laboratory data), relating them to temporal indicators (e.g., seven years ago), and ordering the events resulting from assigning temporal indicators to key concepts with respect to a "key event" (e.g., at admission, at surgery). In the sample text in figure 1-1, the first sentence, given in (3),

(3) This 80-year-old Caucasian female complained of nausea, vomiting, abdominal swelling, and jaundice.

requires the following domain concepts:

Patient: person identified by age, sex, and profession, whose signs, symptoms, and laboratory data will be given.

Symptoms: manifestations of abnormalities reported by the patient. Certain symptoms have to be further defined: swelling needs a characterization as to where it occurs. Pain can be characterized by its location, intensity, and nature (e.g., "shooting").

Signs: abnormalities found by the physician such as fever, jaundice, or swelling.

Whether "fever" is a sign or a symptom is indicated by the verb. Therefore, the verbs have features which indicate if the following is a sign or a symptom. There are no explicit temporal indicators in (3), except the tense marker on the verb. The doctor, however, knows that case histories ordinarily use "admission" as a reference point.

(4) She had diabetes mellitus, treated with insulin for six years before admission.

The sentence in (4) requires a temporal concept "year" in conjunction with the numerical value "six"; it also requires the concept "duration" to represent the meaning of for. The "key event" at admission is mentioned explicitly and must be recognized as a concept by the system.
After selecting the facts on the basis of about 35 case descriptions as well as previous research of the medical sublanguage [Hirschman 83], I organized them into schemas on the basis of what is known about the particular text type. In [Bonnet 79, 80], a medical summary is characterized as "a sequence of episodes that correspond to phrases, sentences, or groups of sentences dealing with a single topic. These constitute the model and are represented by schemas" [Bonnet 79, 80]. Schemas for the medical domain in Bonnet's system are $PATIENT-INFORMATION$ (e.g., sex, job), $SIGNS$ (e.g., fever, jaundice). In my NLP, I use the schemas $REPORT-SIGN$, $REPORT-SYMPTOM$, $REPORT-LAB-DATA$, $PATIENT-INFO$. Each of my schemas indicates "who reports, what to whom, and when". The $REPORT-SYMPTOM$ scheme has the following elements: verb(unknown), subject(patient), object(symptom), indirect object (medic), time(default is admission).

After selecting the facts on the basis of the domain, and organizing them on the basis of the text-type, I add one fact for putting the information into the target representation. The target representation consists of a temporal indicator attached to a domain-specific fact – what I had referred to in 1.2 as "event". The event structure contains the following elements: name of domain-specific concept, reference point, duration (known or unknown), relation to reference point (e.g., before, after).

2.1.2. The Flow of Control

In addition to domain-specific knowledge, a person reading a text also uses his linguistic knowledge of the English grammar. The problem for a NLPS is how to integrate linguistic and extralinguistic knowledge. The dominant paradigm in computational linguistics uses syntactic and morphological information before considering extralinguistic knowledge; if extralinguistic knowledge is used at all.

Considering syntactic knowledge before any other type of knowledge has the following

---

18 I use ten types of domain-specific facts: sign, symptom, lab data, body-part, etc.. I use six temporal facts: month, year, day, week, duration, period, i.e., "for how long".
problems which are avoided if enough contextual information can be detected by the
knowledge base of the NLP:

- global ambiguities cannot be resolved (e.g., Visiting relatives can be boring)
- word-class ambiguities (e.g., bank) and structural ambiguities cause multiple
  parses (e.g., I saw the man on the hill with the telescope).

Moreover, psycholinguistic experiments have shown [Marslen-Wilson
75, Marslen-Wilson 78, Marslen-Wilson 80] that the syntactic analysis of a sentence
does not precede higher level processing but interacts with semantic and pragmatic
information. These findings are, to some extent, controversial, and not accepted by all
psycholinguists.

In my system, knowledge about the domain, the text-type, and the target
representation is used before and together with syntactic information. The syntactic
information helps to select the interpretation of the sentence. Syntax functions as a
filter for processing information. It selects the constituents of a sentence, and groups
them into larger "chunks", called phrases. The phrase types noun phrase [NP] and verb
phrase [VP] contain procedures to form concepts (e.g., "abdominal pain"). These
concepts are combined by function specialists. Function specialists consists of procedures
attatched to function words (e.g., prepositions, determiners), inflectional morphemes, and
boundary markers (e.g., comma, period)

Technically, I distinguish between phrase specialists and function specialists. The phrase
specialists interact with extralinguistic knowledge to determine which concepts are
expressed in a text, the function specialists determine locally what relation these
concepts have to each other. So in general, the phrase specialists are activated before
the function specialists.

To illustrate this process, consider the sentence The patient complained of shooting
pain across the flanks for three days before admission. The NP-specialist combines the
and patient into a phrase. The central processing component in the sentence is the
VP-specialist. Its task is it to find the verb–particle construction (complain of), and the object (e.g., shooting pain). The VP-specialist also looks at the syntactic and semantic characteristics of complain of. It notes that complain of expects a symptom in its object position. The expectation of a symptom invokes the schema "report-symptom". At this point, the schema could fill in missing information, e.g., if no subject had been mentioned, it could indicate that the patient is the subject. The schema identifies the current topic of the sentence, viz., "symptom".

The NLP encounters the word shooting next. The word does not have any further specification besides being used as an adjective here. The head noun pain points to a more complex entity "pain" which expects further specifications (e.g., location, type) It first tries to find any further specifications within the already analyzed part of the NP. It finds shooting and adds this characteristic to the entity "pain". Since "pain" is usually specified in terms of its location, a place adverbial is expected. Upon the entry of across, the entity "pain" includes "across" as a location marker, expecting as the next word a body-part. The next word, flank is a body-part, and the "pain" entity is completed. Note here, that the attachment of the preposition was done by the information contained in the knowledge base.19

The next word for is a function word which can indicate duration. To determine which adverbial for introduces, the system has to wait for the information from the following NP-specialist. After the numeric value "three", the temporal indicator "day" identifies for as a duration marker.

Explicit temporal indicators such as day, week, or month, under certain conditions introduce new events. As soon as the NLP verifies that a temporal indicator started an event, it fills in the information from the "report-xxx" schema. The new event representation includes the sign, symptom, or laboratory data, and the temporal indicator. The last two words in the sample sentence before admission, provide the

19 Cf. 8.1.2.3. about a discussion of PP-attachment.
missing information as to what "key event" the newly created event is related to.\(^{20}\)

2.1.3. The System

The NLPS consists of the following highly interacting modules:

- a morphological analyzer for inflectional analysis as well as for morphosemantic analysis of medical words;

- a dictionary subdivided into suppletion, function and stem forms; the latter having semantic/conceptual features attached to them;

- a syntactic parser that determines the phrase and clause structure of a sentence on the basis of the information stored in the knowledge base of my NLPS, but without completing the parse tree for the whole sentence;

- a knowledge base consisting of generic patient data that contains information about signs, symptoms, and laboratory data that aid in establishing a concrete patient history for the incoming case. It is accessed before each phrase has been analysed by the parser and the semantic/conceptual pointers on the word(s) have guided the parse to the pertinent frame; and

- inference mechanisms which handle in particular temporal information before storing the results in the currently forming patient history frame.

An overview of the system modules is given in figure 2-1.

In spite of this high modularity, the flow of control within the system is not stratified, going from one level to the next; rather it is interactive between levels. The modularity of a system does not necessarily lead to a linear flow of control (e.g., stratification of levels), whereby the modules are accessed in a predetermined sequence. As a consequence of the stratification of levels [Winograd 83] in NLPSs, the syntactic analysis of a sentence would precede the semantic analysis which, in turn, would precede the pragmatic analysis. The behavior of a stratified program is contrary to evidence from some psycholinguistic experiments [Marslen-Wilson 80] which show that the syntactic analysis of a sentence does not precede higher level processing but rather interacts with semantic and pragmatic information.

\(^{20}\) A trace of my program of a sample sentence is given in chapter 7 and Appendix C
Figure 2-1: Functional Perspective
In my system, the order and frequency of accessing the modules depends on the input. The modularity of my system facilitates its maintenance. The interaction between modules and within the elements of the modules, is compatible to processes found in the psycholinguistic studies mentioned previously.

The analysis of the input proceeds from the text level to the morpheme level. Each separated linguistic unit helps to control the conceptual representation of information contained in the text. In brief, information processing is done in three stages: segmentation, analysis, and representation of information. The present program (implemented in ELISP and EFRL on a DEC20/60) performs the following tasks:

- determine the phrase structure for sentences of a text from the Journal of the American Medical Association [Cryer 77]. The text consists of a case history of a patient.21

- transform the linguistic information into a conceptual representation.

- draw inferences that account for the context and world knowledge related to the domain of liver diseases (in particular choleostasis).

- create and update a database containing the pertinent information related to the current patient data.

- provide patient information for PATREC, the database system for the diagnostic expert system MDX [Chandrasekaran 83]

In its present form, the program contains information of the patient record system PATREC [Mittal 81] which has the capacity of an "intelligent" data base specialist (i.e., could draw inferences and was not limited to verbatim data retrieval). It goes beyond PATREC in that it takes NL input and considers linguistic information.

---

21 The epistemological/theoretical perspective can be summed up as follows: language is used for reasoning (inferences); reasoning is based on knowledge (representation); knowledge is acquired mostly through language (use). The underlying assumption is based on Wittgenstein's dictum that the "meaning" of the language is its use.

22 The sample text analyzed in this dissertation was chosen randomly from about 35 case descriptions. Although the sample grammar and lexion is limited to this one sample case, the goal is to be able to analyze any other text within the domain of liver diseases provided the proper grammatical and lexical extensions are made.
2.2. IMPLEMENTATION

The implementation of a knowledge base requires the modelling of knowledge in terms of computer data structures. The two basic implementational issues for the knowledge base are:

(1) techniques to represent knowledge (e.g., production rules, logic, semantic nets, frames and analogue representation),

(2) techniques to use the knowledge (e.g., inference engine, distributed knowledge problem solving)

2.2.1. The Representation of Knowledge

The importance of knowledge representation for the success of a system is discussed in [Woods 83], where a distinction is drawn between "expressive adequacy", i.e., what the system can "understand" and "say", and "notational efficacy", i.e., how powerful the notation is.

The four most commonly used knowledge representation schemes include frames, production rules, logic, and semantic networks. I discuss in some detail frame-theory and production rules, and point out their advantages for my system without going into a discussion of logic and semantic networks.

In general, the form of representation (e.g., logic, frames) is not independent of its use, if "use" refers to the semantics of the representation (i.e., what the representation scheme does and allows one to do with it). For instance, when logicians abide by the Modus Ponens, they disagree psychologists who claim that the Modus Ponens is not part of our "natural" reasoning capability [Wason 72], i.e., Modus Ponens is a tool for reasoning which we learn to use.

The symbols of a language represent the primitives of a theory. Arriving at the symbols is a theory, and arriving at their combinations is a meta-theory of how to use
these symbols. Consequently, the form of the representation is only independent from its use, if "use" refers to the representation employed with respect to the domain or the task, but never to the semantics of the representation.

2.2.1.1. Frames

Minsky suggested frame theory [Minsky 76] as a technique, to represent background information for knowledge-based systems. In frame theory, the knowledge base is decomposed into "chunks" of knowledge which are datastructures that represent stereotypical situations.

When one encounters a new situation (or makes a substantial change in one's view of the present problem) one selects from memory a structure called a Frame. This is a remembered framework to be adapted to fit reality by changing details as necessary [Minsky 76].

In NLPs, frames are used to represent linguistic as well as extralinguistic knowledge. A frame consist of slots and fillers which store the elements of a particular concept. A frame contains slots, and slots contain fillers. The slot of one frame can point to another frame with slots of its own. A room can be represented as a concept ("frame") consisting of the elements ("slots") "door", "window", "floor". A noun phrase can be represented as a concept consisting of the elements "determiner", "adjective", "noun". Each of these elements can have further components ("fillers").

If a particular concept is mentioned during text processing, the corresponding frame for this concept is activated. The information stored in the frame can then be used for processing. Frames can either be interpreted as "a static datastructure about one stereotyped view" [Charniak 75], or as "organizing the processes of retrieval and inference which manipulate the stored representations" [Hayes-Roth 79]. The manipulation of data by frames is done by "triggers", which are procedures attached to frames, slots, or fillers. If during the information processing, a frame or one of its subparts is added, deleted, or accessed, the "triggers" activate routines which further manipulate the data.

In general, frames are stereotypical representations of entities that have the following characteristics [Minsky 76, Winograd 77, Roussopoulos 79]:
explicitness: entities, relations and attributes are made explicit; in the absence of disconfirming evidence, a default value is adopted;

- triggering: procedures can be attached to frames and executed automatically (e.g., "IF-NEEDED", "IF-ADDED");

- inheritance: frames are conceptually related allowing entities to be inherited from higher up in the hierarchy; and

- modularity: the data base is organized in clearly distinct components.

The drawbacks of using frames for a theory on how information is processed relate to how much knowledge should be stored in frames, what criteria are used to activate them, and if the notion of frames is just "little more than a cumbersome convention for the listing of facts" [Dresher 76, 356].

The idea of having a "precompiled" knowledge structure as background information during language processing is theoretically very useful because NLPs could rely on it for their tasks. However, as a psychological theory for human information processing it is less attractive for two reasons. If we had knowledge-structure-like frames stored for easy retrieval, "a lot less human discourse should occur than actually occurs" [Brown 83, 240], since most of the information would be present in our "frames", and nobody would have to bother making it explicit. Secondly, although stereotypical knowledge can be expected, it cannot be guaranteed. Discourse producers "make their discourse reflect this fact, and present the information in a form which serves as a reminder for those who already know and as an instruction for those who do not" [Brown 83, 240].

Another important issue for frame-systems is the decision as to which frame should apply in a given context. If multiple frames are possible, what criteria should be used for activating one. In brief, problems that have to be dealt with while using frames are:

- which frames are relevant for a given situation?

- how many different frames are needed?
I have used frames successfully in my system without encountering any of the drawbacks for the following reasons:

(a) The facts in the particular type of medical text can be succinctly defined and organized. The text structure is very rigid and stereotyped; only a few clearly distinguishable schemas need to be represented. Therefore, the two major problems with frames do not arise in the text type I worked on.

(b) The task of my system is clearly defined. The target representation is transparent based on the frame-approach. The implementation still preserves the theoretical division of the various knowledge sources. The frame-representation is both notationally efficient, and expressively adequate.23

(c) My theory is based on the controversial assumption that text comprehension in a particular domain requires higher level "pre-compiled" concepts (e.g., "pain", "event") which the NLPS can use. The use of a higher level concept does not presuppose that there is no need for primitive decomposition at times. The use of higher level concepts is simply a "short name for a primitive decomposition" [Charniak 78, 192].

2.2.1.2. Production Rules

A. Newell proposed production rules within a theory of how humans reason [Newell 72]. Production rules, also known as "IF-THEN rules" or "Situation-Action rules", consist of an antecedent, representing a pattern which is matched by a simple procedure to input features, and a consequent which determines the action once the left side pattern is matched. The antecedent can consist of a number of conditions (AND/OR) that have to be met if the rule should be "fired". In Newell's theory, production rules are contained in the short-term memory. The production rules are tried in an iterative control mechanism residing in the short-term memory until the left side of the rule is matched, which in turn causes the rule to "fire" and the "action" on the right side of the rule to apply.

23 The actual code is given in Appendix B.
Production systems contain a set of production rules, an algorithm for pattern matching, and a working memory. The conditions on every production rule are matched against the current data input in the working memory. If the conditions do not apply, a new rule is tried. If a condition matches the left side of a rule, the action proposed on the right side changes the working memory which may cause further rules to match. The cycle of trying to match the production rules continues until all rules are tried.

The knowledge representation in production systems is modular, since every production rule contains a separate piece of knowledge. Modularity is a useful feature for knowledge representation schemes because it facilitates the change, addition, and deletion of knowledge from the system. For larger systems, however, the modularity provided by production rules is inefficient if all rules have to be matched against every input.

Positive aspects of production systems include the uniformity of their structure, the modularity of their independent pieces of knowledge and their extensibility. The major disadvantage stems from the combinatorial explosion due to the numerous possibilities of rules that can apply. Every rule has to be considered as a potential candidate during each cycle of pattern matching. This problem can be reduced by using organizing principles (e.g., rule ordering, rule grouping, or explicit heuristics).

In my system, production rules are used for organizing the knowledge stored in the frames into the target representation. A combinatorial explosion does not occur since the number of rules are limited, and the rules are grouped into separate components. The specialists in my system are coded in terms of production rules since they have to react to changing patterns. A positive aspect of production rules is how easily they can be modified.24

24 For the actual code of these rules, see Appendix A.
2.2.2. The Use of Knowledge

The most common approach to NLP in computational linguistics, is to separate the knowledge from its use. In those systems, the knowledge base represented by frames, production rules, or other techniques, is separated from the problem solver as represented in figure 2-2.

![Diagram of Problem Solver and Knowledge Base](image)

Figure 2-2: Dominant Problem Solving Paradigm

Most NLPSs in computational linguistics use the approach of separating the problem-solver from the knowledge base. Their knowledge base consists mostly of rules which determine the phrase structure of the sentence. A radically different approach has been proposed in [Rieger 79] under the label "word-expert-parser". In Rieger's system, every word expert has its own knowledge base and problem solving capacity.\(^\text{25}\)

In my NLPS, the problem solving is controlled by a local control mechanism whereby each expert or specialist independently determines which knowledge source it needs to access next. Conceptually, I distinguish between two different types of specialists: phrase specialists which determine how NPs and VPs are constructed, and function specialists which determine the relation between the NP-specialists and the VP-specialists. The specialists include production-rules which apply whenever the NLP encounters a condition for applying a production-rule. The phrase specialists enable the program to access the facts stored in the frames. A flow diagram of the system is given in figure 2-3. If a frame is accessed by a production rule, the particular frame based on the information stored in its triggers may take control of the parsing procedure.

The knowledge base of my NLPS also contains domain-specific (e.g., signs, symptoms) as well as domain-independent knowledge (e.g., temporal indicators). On the basis of

\(^{25}\) cf 3.5.1.2.
Figure 2-3: A NLP With Distributed Problem Solving Specialists
the domain-specific knowledge, the NLPS determines the current topic of the sentence. The possible topics correlate to the domain-specific schemas. The NLPS attaches temporal indicators to every schema it encounters in the text. The information from the schema and its temporal indicator are stored in a separate event frame. Each of the event-frames is then connected to a "key-event" frame.\(^{26}\)

2.3. ISSUES

2.3.1. Pragmatic Parsing

The process of parsing can be described from two different points of view: on the one hand, parsing programs are characterized by what (linguistic or computational) theory they are based on; on the other hand, they could be classified in terms of their output. Two types of parsing procedures can be distinguished:

- syntactic parsing: the program uses only syntactic information and provides an annotated surface structure of the sentence as output. It is mostly restricted to the processing of sentences in isolation;

- pragmatic parsing\(^{27}\): the program uses information from more than one sentence. A pragmatic parsing program has the capability to establish connections between entities mentioned in the text it processes in order to show its "understanding" in the form of possible inferences it can draw and questions it can answer.\(^{28}\) Pragmatic parsers differ in two aspects from parsers that are limited to syntactic or semantic information: (a)

\(^{26}\) The theory upon which the design of my NLPS rests, will explained in detail in 2.3., and in the following chapters.

\(^{27}\) I introduce the notion of pragmatic parsing in my dissertation to refer to a parsing technique which uses contextual knowledge in its control structure. In particular, I use this notion to separate explicitly NLPSs which use contextual - sometimes also referred to as "semantic" information [e.g., Schank 75] - from NLPSs which use model-theoretic semantics or first-order logic.

\(^{28}\) As Zadeh puts it: "The ability to summarize is an acid test of the ability to understand, which in turn is a test of intelligence and competence" [Zadeh 84, 306].
they consider contextual information; (b) they can rely on the additional information they contain for their analysis.

For linguists and researchers in AI, "parsing" is a controversial issue. In general, linguists, especially under the tutelage of Chomsky, submit that syntactic processing is prior to higher level linguistic processing [Marcus 80]. Linguists who propose a semantic theory for parsing ordinarily argue that the semantic representation of a sentence is equal to its logical form. Both groups stipulate that their respective level of processing is "autonomous".

For the syntactician who works within the Chomskyan paradigm of Generative Transformational Grammar, autonomy means that for analyzing a sentence no semantic information is necessary (e.g., [Marcus 80]). Parsers that are based on autonomy of one form or another are written to "bootstrap". Bootstrapping here means that such programs perform only within the various levels, and do not integrate various knowledge sources. They "bootstrap" themselves to higher level processing.

In contradistinction to these bootstrap parsers, AI-researchers on natural language, most notably Schank [Schank 75, Schank 77], adopt the approach of what I call "overhead parsing". "Overhead" is the amount of postulated predetermined knowledge structure. The term is meant to suggest that the complexity of these structures is not necessary for parsing. Overhead parsers are based on the assumption that parsing is a memory-based process and that it relies on fixed knowledge representation. Parsing is then reduced to a process of matching low-order elements (found in a sentence) into higher order, prefabricated, stereotypical structures. The most significant problem with this approach is to determine which frame to access and under what conditions.

The common drawback of bootstrap and overhead parsers is their stratification of

---

29 Cf. arguments presented in 2.2.1.1
Schematically, the stratification of levels as seen in figure 2-4, includes a linear access of the levels involved. The resulting system is "brittle" [Winograd 83, 20] in that after the processing is completed on one level, the information represented on one level (e.g., morphology) can no longer be accessed, even if necessary.

Figure 2-4: Stratified Model

---

Stratification means that levels do not interact.
The arguments against stratified systems do not apply to modular systems in general. Modular systems work under the assumption that a complex problem-solving task makes use of different sources of knowledge. For language processing we can discern various knowledge sources, ranging from phonological to pragmatic components. The modularity of a system does not necessarily lead to a linear flow of control whereby the sources are accessed in a predetermined sequence. A viable alternative is the "blackboard" approach as used in HEARSAY II, which can be schematically represented as in Figure 2-5: various knowledge sources communicate via a blackboard, a control structure that contains the output from the various knowledge sources for a given problem.

![Blackboard Model](image)

Figure 2-5: Blackboard Model

Differences between bootstrap and overhead parsers lie in their focus on what knowledge source determines the information processing. For the Schankian approach, conceptual structures in the form of frames and scripts determine the understanding process, whereas for the proponents of Chomsky's theory, syntactic information governs the parsing process. Bootstrap parsers encounter problems with their grammatical rules, whereas overhead parsers have problems with rules that work on their knowledge representation. In both cases, the solution to their problems is to constrain their rule applications.

One goal of constraining rule applications could be a deterministic form of processing.
Deterministic parsing requires procedures that neither build up unnecessary structures nor require backtracking. Bootstrap parsers, especially PARSIFAL [Marcus 80], are limited in that syntactic information is not sufficient for comprehensive parsing. Bootstrap parsers generate annotated surface structures, or, in the case of GPSG, an intermediate logic representation. Overhead parsers are unfocused in that they allow for the instantiation of conceptual structures that may not be needed in a particular instance.\footnote{In GROK, the focus of processing is neither on the low level of the actual text, nor on the high level of the conceptual organization of the knowledge base, but on the process of transferring information from the text to the knowledge base\footnote{The knowledge base is discussed in chapter 2; its EFRL code is found in Appendix B.} as shown in Figure 2-6.}

In GROK, the focus of processing is neither on the low level of the actual text, nor on the high level of the conceptual organization of the knowledge base, but on the process of transferring information from the text to the knowledge base\footnote{It is problematic for such a program to determine which frame or script to instantiate at any given time. The notorious restaurant script does not have to invoked if a restaurant is mentioned only in passing, or for other reasons than food consumption.} as shown in Figure 2-6.

![Figure 2-6: Differences in Processing Strategies](image)

The information processing for written texts is determined by
- the perspective of the writer;
- the perspective of the reader;
- their linguistic knowledge;

\begin{itemize}
\item \textit{TEXT} \quad \longrightarrow \quad \textit{KNOWLEDGE REPRESENTATION}
\item \textit{BOOTSTRAP} \quad \uparrow \quad \textit{GROK} \quad \uparrow \quad \textit{OVERHEAD}
\end{itemize}
- their empirical knowledge available ("common sense");
- the sublanguage of the text; and
- the co-text.

The reader tries to reconstruct the writer's message from the text. If the text is highly structured (as in textbooks and tests), the perspective of the reader and writer are not very different, since the reader expects certain information in a certain form: the genre of the text determines the expectations of the reader. These expectations make the reader access his empirical knowledge in a focussed way: if there is mention of pain in a medical text, the options of understanding pain in this specific context are limited to interpreting it as the manifestation of some disorder. Moreover, the co-text consisting among other things of the meaning of a specific verb that changes the current theme focus will limit the term even further to a symptom.  

The information transfer can be characterized as follows: the writer and reader rely on a common basis of linguistic and empirical knowledge. Their perspective on a given text is focussed by their common understanding about types of texts (in a wider sense, types of discourse including on a more abstract level the intentions of the writer and reader). The specific text type helps the reader to interpret the message by means of the domain knowledge which the text requires.

2.3.2. Temporal Pragmatics

Current theories about time can be distinguished with respect to their ontological, methodological, and representational characteristics. Ontological differences, for example, arise when time is interpreted either as a series of real numbers or as as sequence of intervals. Methodological differences stem from the data that should be considered. In most tense-logical treatments of time, only sentences in isolation are investigated. Other theories are based on discourse structures or the speech situation. A further

---

33 The verb complain suggests that the patient is communicating a subjective disorder, viz., a symptom. This information is embedded in the verb and not in the noun itself since some nouns can be symptoms or signs (e.g., fever).
methodological consideration is the choice of semantic versus pragmatic approaches. Representational differences relate to what the theorist perceives to be an adequate representation for the temporal information embedded in a sentence/text. The representation of the temporal information can be based on a formal language, a time specialist, or conceptual structures.34

Differences in the definition of meaning distinguish model-theoretic semantics from procedural semantics. Procedural semantics35 is associated with the idea that meaning arises from computations; simplistically stated, programs are meanings. Identifying the meaning of the utterance with procedures does not produce adequate theories of meaning. "Procedures are devices linking truth conditions to perception" [Woods 81, 302]. In [Smith 82], Smith contends that procedural semantics has to have a declarative counterpart in an adequate theory of linguistic or computational semantics. He proposes to combine the operational interpretation of meaning in computer science with the declarative or referential use of meaning in linguistic studies. Programs which consist of the codification of an algorithm, in my opinion, express relations that have meaning, and are not the meaning itself.

Model-theoretic semantics of first-order logic and "possible world" semantics (e.g., Montague Grammar) are based on Frege's assumption that the reference of a sentence is its truth value - a theory of meaning is a theory of truth conditions. Model-theoretic semantics of first-order logic and "possible world" semantics fall short of explaining temporal phenomena. Since both theories work under the assumption that the meaning of a sentence can be studied independently from what the sentence refers to [Johnson-Laird 81, 114], temporal relations which depend on the reference situation (e.g., a while later), cannot be handled by these theories.

---

34 Formal languages are ordinarily used in logic; a time specialist, as used in [Kahn 77], is a program which can handle time relations without any further knowledge about the domain.

35 "In general, the procedural semantics approach is a paradigm or a framework for developing and expressing theories of meaning rather than being a theory of meaning itself" [Woods 81, 302].
Further evidence for the necessity of having access to the reference situation is provided by Kamp [Kamp 83, 255] where the differences in truth conditions between the French imparfait and passe simple are discussed with the conclusion that "the factors which determine the use of the imparfait and passe simple can only be explained at the level of discourse representation" [Kamp 83, 255]. Kamp's discourse representations rely on the reference situation which is not included in first-order logic or "possible world" semantics.

The problem of providing truth conditions for sentences in isolation has led in previous treatments of tense logic to discussions on what parameters (e.g., moment of speech, event, reference) should be introduced, and what their number should be [Prior 67, Reichenbach 47]. In my approach, however, the referent points are found in the discourse itself, thus reflecting the changing temporal focus within the discourse. The reference points differ from domain to domain. Their function is similar to Reichenbach's reference time [Reichenbach 47] which is discussed in detail in chapter 4.

Context consideration and building discourse representations lead to a pragmatic theory that is related to research done by Hans Kamp [Kamp 79, Kamp 83]. "These representations, which the recipient of a discourse is made to construct in response to the verbal inputs he receives have the form of event structures" [Kamp 83]. Since event structures are built up from the discourse, they do not deal with truth or real-world events. If each sentence contains one event, the total number of sentences makes up the discourse structure. Discourse representations in Kamp's theory represent an intermediate step between the syntactic analysis and the semantic interpretation of temporal information embedded in a text.

In AI, the few approaches to temporal processing either downplay the function of
NL\textsuperscript{36} or propose a temporal logic [McDermott 82, Allen 83]. An alternative to temporal logics is the notion of a "time specialist", an "idiot savant" [Kahn 77, 87] that is a domain independent procedure for general temporal information processing.

The overall ideas of GROK as they relate or differ from the extant theories and systems are introduced by looking at four major issues concerning temporal processing.

- temporality: how is an event defined in the system; how is temporal information treated vis-a-vis the whole system. What search algorithms or inference procedures are provided?

- organization: are events organized on a time line, by key events, calendar dates, before/after chains?

- problems: how is imprecision, fuzziness, and incompleteness of data handled?

- testing: how can the system be tested; by queries, proofs, etc.? Does it have a consistency checker?

In GROK, I use an interval-based approach to temporal information processing. An event is defined as an entity of finite duration. As in [Kamp 79, 377], event structures are transformed into instants by the Russell-Wiener construction.\textsuperscript{37}

In GROK, the NLPS processes temporal information by first associating a concept with a temporal reference, then evaluating the extension of this event. This evaluation considers syntactic (e.g., adverbials) and pragmatic information (current time focus). Each event is represented in the knowledge base with information about when, for how long, and what occurred.

\textsuperscript{36} In [Kahn 77], the required form of "stilted language" is given along with the explanation that this transformation has not altered the content of the interaction. The sentences I was born January 25, 1952. When I was a few weeks old I had an operation. have to be translated into (time-of(beg of life) (date)1952 1 25) (fuzz small) (time-of(end of operation) (beg of life) (fuzzy-amount) (a few weeks)).

\textsuperscript{37} Further discussion is provided in chapter 4.
The parser while analyzing the sentences, orders these events according to a "key-event". The single events contain information about the temporal indicator which is attached to a domain-specific fact. The single events are connected to the respective "key-event". "key-events" are domain-specific. In general, I stipulate that every domain has a limited number of such "key-events" which provide the "hooks" for the temporal structure of a domain-specific text.

GROK differs from logical theories in that it deals with discourse structures and their conceptual representations, not with isolated sentences and their truth value. It is different from Kahn's time specialist in that it uses domain knowledge and "knows" about temporal relations of a particular domain. The domain-specific temporal knowledge also sets it apart from Allen's [Allen 83] temporal inference engine approach. The structure of the tempus frame is not domain specific.

GROK differs from Kamp's discourse structures in that it uses the notion of reference intervals that are based on conventional temporal units (e.g., day, week, month, year) to organize single events into a chronological order.

The assumptions of this dissertation are:

(a) NL can be processed realistically by a deterministic algorithm which can be interpreted as a mental model.

(b) Temporal information processing is adequately explained only in a pragmatic theory that captures the duality of interval and point-based representation of time.

(c) NL can be processed efficiently by a set of integrated linguistic and extralinguistic knowledge sources.

A realistic NLPS tries to emulate human behavior; although working parsers do not necessarily have to be realistic, the degree of realism achieved provides much needed insight into the psychology of linguistic processing [Kac 82].
A deterministic NLPS, in contradistinction to systems with parallel processing or backtracking, stipulates (a) that a human NLPS makes irrevocable decisions during processing and (b) that humans are not unconstrained wait-and-see-parsers [Kac 82]. A necessary corollary to these stipulations is found in [Marcus 80], in which the phenomenon of "garden path" sentences can be explained as follows: whereas parsing takes place on an unconscious level, consciously applied procedures are needed to recover from "garden path" sentences.

An algorithm has two elements: logic and control. For NLPS logic subsumes the grammatical formalism applied and the form knowledge is represented. Control deals with search and rule invocation. Logic\[^{38}\] makes claims that pertain to theoretical linguistics, and cognitive science in general; whereas the issue of control is a matter of implementation.

A mental model provides an internal representation of the state of affairs that are described in a given sentence [Johnson-Laird 81]. In spite of the impossibility of determining what exactly this mental representation for a sentence might be and the unresolvable trade-off between what goes into the representation and what is covered by the processes [Anderson 76], the important point about these models "is not their phenomenal or subjective content, but their structure and the fact that we possess procedures for constructing, manipulating and interrogating them" [Johnson-Laird 81].

A pragmatic theory focuses on the interaction of linguistic and empirical knowledge in a given context. In my interpretation of a pragmatic theory, the referents of certain expressions determine the meaning of other expressions. My contention is based on the arguments found in [Johnson-Laird 81, 114-117]:

- "The principle underlying the interpretation of sentences is that a listener often has recourse, not to selectional restrictions, but to inferences based on factual knowledge about referents" [Johnson-Laird 81, 115].

\[^{38}\] "Logic" here means rule-governed and systematic principles which can be used to describe phenomena. It does not imply any preference of how to formalize the reasoning process.
"The entailments of sentences containing such relations as on the right of, in front of, at, near evidently depend on properties of the reference situation" [Johnson-Laird 81, 117].

Inferences depend on properties of the reference situation. Thus inferences based on deictic or indexical elements of expressions cannot be explained by decomposition [Smith 74, Schank 75], semantic networks [Collins 72] or meaning postulates [Kintsch 74], since these theories are based on the assumption that meaning is autonomous in the sense that it is not dependent on the reference situation [Johnson-Laird 81, 117].

A fundamental question for a theory of temporal relations is whether time is interpreted in terms of intervals or with reference to points. Following [Russell 56], I assume that events are basic and that intervals and points in time are defined in terms of events. Since intervals might "collapse" into point-events and vice versa, this duality has to be captured in a theory. In GROK, as in [Kamp 79], the Russell-Wiener construction is used to go from events to instants.

The theses in this dissertation's assumptions (a) and (b) relate to theoretical issues, whereas in (c) methodological claims are made. Understanding NL requires linguistic and empirical knowledge. The integration of these different types of knowledge is represented by a process of interacting knowledge sources based on the flow of information and control. This methodological standpoint is in sharp contrast to systems designed on the basis of autonomous modules accessed in strict sequential fashion going from one level to the next without "hand-shaking". Apart from being computationally inefficient due to the "ripple" effect they cause when modified [Winograd 83], they are also psychologically unnatural, since some experiments have shown [Marslen-Wilson 75] that higher level processing begins before a whole sentence is read. Thus, syntactic processing of the whole sentence does not temporally precede semantic or pragmatic processing modules but interacts with them. Ultimately, this methodology should come closer to reconciling the two diametrically opposed opinions on endeavors in AI, that systems are either efficient or natural [Winograd 83].
CHAPTER 3
PARSER

The present approach is based on the assumption that the solution to the foremost problems in computational NLP (e.g., ambiguity, anaphora, ellipsis) lies between the extremes of rule-based and word-based parsing. Words, according to their functional and semantic information, are the primary ingredients a NLPS employs to "understand" language. The NLPS has a number of strategies [Clark 77] that are based on the structural, functional, and conceptual information a word contains.

The overall "game-plan" during language processing is determined by a bottom-up mechanism that integrates the various knowledge sources triggered by the single words. This combination of bottom-up processing and triggering procedures to locally disambiguate prevents both combinatorial explosions of grammar rules and procedural lexical entries that are too complex. At the same time, it is supported by a number of psycholinguistic experiments [Tanenhaus 84]. Although I cannot provide any new psycholinguistic evidence for my theory, it is consistent with the current research results.

This chapter provides a detailed technical analysis of (1) the morphological module, which consists of a morphological analyzer and lexical access procedure; (2) the lexicon; (3) the syntactic module; and, (4) a comparison to current parsers. Parsing refers to the transposition of information into a representational structure that a computer program can use for further processing.

39 The actual source code is given partially in the appendix; utility functions are omitted; an outline of the user-interface including the menu containing the commands is provided.
3.1. MORPHOLOGICAL MODULE

The function of the morphological module is to analyze the morphemic structure of the words in an input sentence and to access the lexicon to retrieve the information contained in it.

In its implemented form, the morphological component fully analyzes inflectional morphemes by separating them from the stem-forms and also handles suppletive forms. Another task performed is a morphosemantic interpretation for medical jargon, along the lines of research found in [Pacak 80]. Other morphological processes (e.g., derivational morphology) are not analyzed since they are not the main focus of this project. Furthermore, the morphological analyzer for this system does not, in itself, make any far-reaching psycholinguistic claims, as can be found in the more ambitious program KIMMO that analyzes languages of different morphological complexity (Finnish versus English) with the same speed, thus emulating human performance (which does not depend on the complexity of morphological rules [Karttunen 83]).

![Figure 3-1: The Morphological Analyzer](image)

---

40 This is done by simply adding the most commonly encountered suffixes on medical terms to the analyzing procedure; words of Greek and Latin origin are translated in the lexicon into their English equivalents. The main reason for this is to forego a full-fledged morphosemantic account of the medical jargon, which is not a concern of this project.
The morphological module is called from the top level of the program. It analyzes each word of a sentence sequentially by first trying to look up its features in the lexicon. Lexical information is divided into a hash list for function words, suppletive forms, and word stems. After attempting to find the current word in the suppletion table, the analyzer looks for it in the stem table; if it fails there, inflectional suffixes are removed before the separated stem is again sent to the stem table. The order of processing is crucial to allow the morphological component to analyze words like *sing* and *early* correctly. The morphological analysis of medical terms (e.g., *-itis*) is done immediately before looking for the inflectional endings. A flowchart for this control-structure is given in figure 3-1.

Accessing the lexicon in this way is simply a computational convenience, since the low-level morphological processing can be separated from the higher level processing. Theoretically, the lexical processor is an autonomous module in this system. This assumption is supported by [Forster 79] and is the starting point for claims made in [Tanenhaus 84] that the lexical processing module provides access to all readings of an (ambiguous) word.

3.2. LEXICON

The interaction of linguistic and conceptual levels makes the important supposition that memory, containing conceptual information, is separate from the lexicon that contains the linguistic information and pointers to conceptual structures. This assumption is based on the fact that there is no one-to-one mapping between memory, as a mechanism for conceptualizations, and the lexicon, as a mechanism for labeling entities [Carbonell 79]. Evidence for the separation of lexicon and memory is found in the following facts:

- the same concept in one language has different labels in another language (e.g., *in* <English> translates into *dans* en <French>). If concepts and words

\(^{42}\) I am aware of the crucial difference between human and computer memory; the former due to its physiological make-up is much more given to "graceful degradation" of information, whereas the latter is not.
Figure 3-2: Control in the Morpho-analyzer
for the concepts were the same psychological entity, thinking would be
determined by language. Since we can use concepts that do not have a
considering word in our language, there must be a separation between the
linguistic and the conceptual level.

- humans retrieve more information from a sentence than is explicitly
  provided [Kintsch 74]. If our memory consisted of word-concept equivalents,
  we would not be able to make conceptual inferences but only linguistic ones.
  Combining concepts to make inferences, as shown in these experiments, goes
  beyond word associations.

- we use different names for the same concept (e.g., car, automobile)

- we distinguish between empirical and linguistic truth (e.g., Carter was the
  50th president. A bachelor is an unmarried man).

- we understand constructions that are semantically noncompositional (e.g.,
  idioms)

- we do not retrieve direct information from a lexical entry but simply access
  it and check that it has some semantic information [Johnson–Laird 81].

The structure of the lexical entries is based on the experiments and observations that
words are the focal points in the parsing process [Clark 77]; they provide the NLP
with three types of information: structural, functional, and conceptual. The structural
information consists of syntactic and semantic features that help "drive" the parsing
process; functional information helps to determine the role a word plays in a given
sentence, both syntactically and semantically (e.g., passive, dative); conceptual
information links the linguistic component during the parsing process to the knowledge
base if it has to be accessed to store or retrieve information. A schematic data-entry
in the lexicon is given below:

```
WORD (<STRUCTURAL INFO> <FUNCTIONAL INFO> <CONCEPTUAL INFO>)
```

The structural information slot contains labels for word-classes and semantic features;
the functional slot has condition-action rules that monitor the parse; the conceptual slot
has pointers to the frame-structured knowledge base. Each word in the lexicon has
these types of information attached.
3.3. SYNTACTIC MODULE

The parsing function consists of an augmented phrase structure grammar. This is done in a bottom-up, left-to-right procedure that uses a context-free grammar [Harman 63, Gazdar 81, Pullum 83]. While each word is processed, the slots containing the functional and conceptual information are activated if needed. Depending on the ambiguity of a word, different provisions are made to handle it; if a word belongs to different word-classes (e.g., attack, on), the structural slot is left empty; thus, the parse stops and the functional "demon" is triggered to help determine the word-class, by either looking ahead or back. Processing a word involves three parallel procedures that provide information on three different levels. Control is focused on the structural level; if needed, control is passed to the other levels. A schematic representation of the parsing control structure is given in figure 3-3.

![Figure 3-3: Parse-Procedure](image)

After each phrase, the parse accesses the knowledge base, and, if necessary the temporal heuristics component. If no temporal information is given, defaults or previous temporal referents are retrieved and used during the inferencing process.

The top-level contains a list of phrase structures that have the form of CF rewrite rules. At no point during the sentence parse is there a complete syntactic derivation. It is important to point out that the system parses directly into knowledge structures.
without postulating primitives on a submorphemic level, as in the Schankian approach to NLP [Schank 75, Schank 82].

The information attached to the lexical items allows the parsing procedure to use conceptual information if there is no rule in the grammar for a particular construction. Therefore, the capability to handle elliptic or possibly incomplete input is built into the parser. The parser becomes robust since it has the option to resort to the conceptual level for parsing if the syntactic information is not available, either because the grammar does not include a particular rule, or because the input is faulty.

3.4. X-BAR SYNTAX

Grammatical theories that are primarily based on syntax deal with the relationships between parts of speech and the syntactic categories. In traditional grammar three major divisions were recognized in the syntax: the word, the phrase, and the sentence. In modern linguistics, most notably in Generative Transformational Grammar, researchers suggested that, based on empirical evidence, there exists a grammatical category between the lexical and the phrasal level.

In my system, I adopt a form of X-bar syntax that corresponds to [Jackendoff 77] in which more than one phrasal projection of lexical categories is recognized.

In the following model, I have restricted my grammar to categories with two bars because (a) I deal with the analysis of phrases only and not full sentences, (b) the fewer levels of structure, the faster the response time (recursion). The two-bar framework used, captures "a linguistically significant generalization of the structures associated with major categories" [Jackendoff 77, 17], is schematically given in figure 3-4.

Computationally, this approach has the advantage of indicating to the conceptual procedures if the constituent has a pre- or postmodifier, whereby X2 accesses categories that go in front of X, and X1 activates procedures that follow X.
The grammar itself allows for processing complex sentences (e.g., subordinate clauses, relative clauses, multiple embeddings). My grammar fares well if it encounters garbled input, since the processing of the conceptual information precedes the linguistic processing. A part of the algorithm for the syntactic processing has also been used in a project which required a broad-coverage case-grammar for parsing New York Times articles in a particular domain [Obermeier 84a].

3.5. COMPARISON TO OTHER PARSERS

Parsing is a "formally defined process" that deals with abstract mechanisms for "applying structure determining rules to input strings" without considering the nature of the output representation [Wilks 83, 13]. Parsers, in the narrow sense of the word, perform the initial task of assigning structural descriptions to sentences or, on an even more elementary level, simply test for the grammaticality of the input; in a wider sense, they encompass also the process of how the knowledge is represented and used. In general, three types of NL programs can be distinguished: recognition-programs determine how well-formed a given input string is; parsing programs that assign structural description to sentences on the basis of a grammatical formalism; NLPs that transform linguistic information into conceptual representations. During the parsing process the linguistic input is "delinearized" and represented in a derivation tree to show the relations between the constituents of the sentence [Barr 81, 256]. The
CONSTITUENT-LIST
(ADV1 ADV2 N1 N2 V2 V1 P1)

GAP-LIST
((BOUNDARY PRO))

ADV1
(((ADV) P1))

ADV2
(((NJM) (N) (ADV)))

N1
(((N) (OF) (N)))

N2
(((DET) N=) (N*) (ADJ# N1)
((DET) ADJ= N*) (ADJ= N*)
((NEGATION) (ED) N=)
((NEGATION) N=) ((QU) N=)
((QU) (NJM) (BOUNDARY) (NJM)
(N)) ((N/JM) N*))

V2
(((HAVE) V1) ((ADV) V-PART)
((V-ASPECT) V1)
((DO) (NEGATION) (V)) ((V-INCHOATIVE) (TO) V1))

V1
(((V-PART (N)) ((V) N=) ((V) N2))
(((V) (ADV))
(((V) (ADJ)) ((V))))

P1
(((PREP) N2))

N=
(((N) N=) ((N)))

ADJ#
(((ADJ) ADJ*) ((ADJ)))

V-PART
(((V) (PART)))

Figure 3-5: X-bar Grammar for GROK

43 CONSTITUENT-LIST refers to the type of constituents recognized by this grammar; GAP-LIST refers to features that (a) cause a constituent break, (b) are not contained in the phrase structure rules, and (c) have a set of procedures attached which enable them to control the parsing locally.
information contained in the derivation tree is then used to interface with more elaborate conceptual structures feasible for higher level processing.

NL programs have been classified with regard to the underlying linguistic theory being used [Winograd 83], the computational techniques applied [Barr 81], the way knowledge is represented [Winograd 72], what type of knowledge is used (e.g., syntactic, empirical, conjectural), or simply according to their applications in specific domains [Gevarter 83, 11]. An assessment of their psychological validity has been of less importance so far.

Computationally, parsers are recursive pattern matchers that map sentences onto syntactic patterns. Early language programs were based on the idea that parsers can use recurring linguistic patterns in the sentence without an explicit grammatical formalism. During sentence analysis, the recognition procedure looks for a match out of a fixed number of patterns, and, if successful, re-arranges the input according to another pattern, similar to template-matching programs in other areas of AI (e.g., vision).

NL pattern-matching programs without a grammatical basis proved to be limited, in terms of linguistic as well as conceptual coverage of input and output. They are useful if only partial analysis is required, or if other components of the system can make up for the loss of syntactic information. In most of these programs, only recognition, and not parsing or understanding of the input, is achieved. Further extensions of nongrammatical pattern-matchers are the recognition and substitution of idioms, the conversion of words into their canonical form, and grouping strategies to limit the size of the pattern to be matched. The basic concept of pattern-matching programs of this nongrammatical kind has been further developed and used in semantic grammars.

Historically, a distinction can be made between the early NL programs that were based on keywords and pattern matching, i.e., domain-specific heuristics (e.g., BASEBALL [Green 63], ELIZA [Weizenbaum 66], PARRY [Colby 74], SAD–SAM, STUDENT), or on indexing schemes (e.g., PROTO–SYNTEx–1) without deductive power, and systems that were limited in their inferencing capabilities (e.g., SIR [Raphael 67],
DEACON, CONVERSE, TLC). The current NLPS are primarily knowledge-based and have some reasoning power [Schank 77, Schank 81, Winograd 72, Brady 83].

Parsers are often evaluated on the basis of their linguistic or conceptual coverage [Tennant 81]. Since most of the NL systems are designed for different reasons (e.g., scientific or commercial), comparisons based on performance criteria seem to be of little use. Thus, I focus my discussion of how the present system fits into the spectrum of parsing techniques in general, and how it compares to individual systems in particular. The class of systems chosen share a number of conceptual, theoretical and task-oriented characteristics, i.e., left-to-right processors that use multiple knowledge sources and that make extensive use of their knowledge base. For determining the differences and similarities of the current system to other approaches I limit the comparison to the following three issues:

1. the linguistic theory being used; 2. the computational theory (i.e., algorithm) being implemented, and 3. ceteris paribus, the cognitive theory behind the system. The following discussion includes programs that are characteristic for stressing one of these criteria over another. The division into parsing, knowledge base and reasoning of existing NL systems is admittedly arbitrary; it is done here to emphasize the differences between the current approach and previous methods used in NL programs.

3.5.1. Linguistic Theory

Linguistic theories focus on what elements are used, how they function and what type of rules make up a grammatical formalism. Linguistic approaches to parsing are distinguished on the basis of their emphasis on autonomous syntactic analysis before higher level processing, as opposed to pragmatic parsing which includes – in its extreme form – a direct mapping of the linguistic input string onto knowledge structures.

---

44 For a discussion of more general evaluation criteria see [Obermeier 84b, Obermeier 84c]

45 For a more comprehensive treatment of extant NLPS see [Obermeier 84d]
3.5.1.1. Syntax-oriented Approaches

The definition of formal grammars in terms of the Chomsky hierarchy had considerable impact on NLP, since (1) if the type of language a system has to account for is understood, a parsing algorithm can be determined easily; and (2) computational complexity is closely related to formal language theory [Hopcroft 69]. The present system follows in principle Gazdar’s contention that NL can be adequately described in terms of a CF-grammar [Gazdar 81]. This position is in contrast to Chomsky’s [Chomsky 63] assumption that NL has to be described by a more powerful grammar than CFGs. Transformational grammars meanwhile have turned out to be too powerful, since they can generate any recursively enumerable set which means that the recognition problem is not decidable. Since "only decidable systems are reliably parsed" [Wilks 83, 14] and TGs were not decidable systems originally, they were not suited for computational implementation.

In Chomsky’s Standard Theory, the mapping of surface into deep structures is too abstract for a viable implementation. Wilks pointed out that this problem arose from the deep-to-surface orientation of ST that resulted in an ad hoc set of heuristics that linked surface structures to deep structure representations [Wilks 83, 15].

The intrinsic problems with computationally implementing the Chomskyan paradigm is its emphasis on stating universal constraints on grammar declaratively; whereas, in a program, these constraints would have to be procedurally stated. This problem led Marcus [Marcus 80] to the theoretical motivation for his parser that a deterministic parser is a theoretically adequate computational equivalent to the Extended Standard Theory.

A host of other problems made researchers reconsider Phrase Structure Grammars.

---

46 This includes CF, context-sensitive and recursive sets.

47 Many are related to the subleties or transformations that made various supplementary devices [e.g., co-indexing, filters, movement constraints], as well as a complex interaction process necessary.
Although linguists had argued convincingly against Chomsky's dismissal of PSGs and Finite State Grammars [Harman 63], these types of grammars have only recently been the focus of many linguistic studies. In the computational paradigm, PSGs are attractive because they are relatively easy to implement in the form of context-free rules which correspond closely to the well-known production rules [Davis 77] in computer programming.

CF grammars are decidable and have a known algorithm [Earley 70] that provides a time limit for sentence recognition: recognition time is in the worst case proportional to the cube of the string length. My system in this thesis consists of grammar rules that are filtered by syntactic and semantic features attached to each word in the lexicon. Using these features does not change CFness, since the current grammar can be replaced by an equivalent CF grammar using a larger number of basic categories. For the grammar to be context-free, the conceptual information attached to each word does not ordinarily "override" decisions made by the CF algorithm. This information could be used in case the structural information is not sufficient to interpret the sentence. Thus this parser is robust in that it can recover from faulty input or from wrong parsing alternatives.

A common characteristic of most parsers is that they are based on rewrite rules for sentence-level parsing. This assumption lead to fairly successful NL programs, from simple keyword and pattern matching [Colby 74, Green 63], to syntactic analysis [Marcus 80, Woods 73, Thompson 81], and is embodied even in higher level NLPS. In spite of their different linguistic and computational grounding, these systems promote the idea that NL processing is best described by rules that deal with larger sentence constituents. Consequently, the words of a language are reduced to tokens that are relevant for comprehension only as part of sentence- and concept-level rules [Rieger 79].

---

48 Another issue here is what type of language is described by a grammar that uses conceptual information for recognizing sentences.
Evidence against this rule-based approach has led to the theory of word-expert parsing that takes the word as the basic linguistic unit; linguistic knowledge is distributed amongst a group of procedural experts that diagnose the word's contextual usage [Rieger 79]. Arguments in favor of this approach are (1) words have a rich linguistic and conceptual structure; psycholinguistic evidence [Marslen-Wilson 78] shows that lexical access is a deciding factor in language comprehension; (2) it is unlikely that language can be reduced to a number of rewrite rules; the multiple applicability of rules that are similar requires a control structure that is internal to the sentence in order to monitor the sequences of constituents. It is plausible to embody this control structure within the single constituents; (3) word sense ambiguities cannot plausibly be captured by existing rewrite systems [Rieger 79].

The current system tries to combine the two extremes: a CF grammar guides the bottom-up parsing process, whereas each word has a number of distributed knowledge sources that monitor the parse.

In contradistinction to other syntactically oriented parsers, the present system does not perform a complete syntactic analysis before it goes on to higher level processing.

3.5.1.2. Semantic Parsing

The central role of meaning in understanding language has led some researchers, in AI [Schank 75] as well as in linguistics [Fillmore 68], to rely on semantic rather than syntactic considerations in linguistic processing. Researchers working within the framework of semantic parsing do not deny the need for any structural processing; they simply reverse the order of importance between syntax and semantics.

Most of the extant systems built within this paradigm are based on Schank's notion of conceptual dependency and scripts [Schank 75, Schank 81]. Without going into a discussion of CDs, I briefly discuss the role of low-level parsing in the most recent

---

49 These include parsers written for transformational grammars [Marcus 80], case grammars [Bruce 75], as well as ATNs [Woods 70].
systems based on these notions: FRUMP [DeJong 74], IPP [Lebowitz 83], and
BORIS [Dyer 83].

These systems share a reliance on top-down predictions to recognize and analyze data, as well as a minimization of syntactic processing. The differences between them are based on where the predictive power is located. BORIS relies heavily on procedural definitions that are attached to the words in form of CDs, whereas IPP draws primarily on the memory structure for a given story. The individual, complex word-definitions in BORIS are replaced in IPP by a set of generic predictions generated from the memory structure [Lebowitz 83, 367]. In Rieger's Word Expert Parser, the level of complexity for the word definitions is even higher than for BORIS.

IPP uses word definitions if no top-down predictions are found, whereas BORIS relies exclusively on a "production-system mechanism for all processing" [Lebowitz 83, 369]. This emphasis on a text-driven control structure also sets IPP apart from FRUMP. In FRUMP, the program tries first to establish a script, and then starts processing the text looking for the pieces relevant to the previously established script. Moreover, FRUMP requires an additional level of processing to determine CDs.

The current system is similar to IPP in that it is not totally dependent on the top-down predictions; in the present program, the word definitions form an integral part of the procedural flow-of-control. Basic differences between my system and FRUMP and BORIS are that they rely on CDs and have rigid predictive-style control structure.

Another difference is based on the notion of expectation-driven parsing that is used in all three systems (its origin is Schank's early work on scripts [Schank 77]). Expectation-driven parsing has been used successfully for limited domains in past projects [Schank 81]. However, it stipulated a huge amount of background information in the form of elaborate memory structures, formalized scripts [Schank 77], and frames [Minsky 76].

Matching every sentence against a rigid memory structure seems to require as much
effort on the part of the comprehender as memorizing and retrieving whole sentences. A reason for building up complex memory structures could be a fallacious analogy between the human memory and that of a computer. It is adequate to have incoming information matched against a rigid data structure for computers.

For humans, there is no analogy to this type of memory, since the nervous system follows different principles (e.g., activation and inhibition of impulses). Thus, expectation-driven parsing as done in the past has to be modified as to what intermediate steps are necessary to explain the facile and efficient use of language by humans. Since humans are good at predicting and poor at remembering [Schank 77], they process language primarily on the basis of the immediate contextual information in the sentence.

3.5.2. Computational Theory

Computationally, parsers can be differentiated on the basis of their strategies (e.g., backtracking versus parallel processing; top-down versus bottom-up) and the techniques (e.g., ATN, charts, Horn clauses) they use. Since there are already several thorough state-of-the-art descriptions on the subject [Winograd 83, Barr 81, Sparck 83], I summarize only a few important points distinguishing my approach from others.

Whereas my parsing algorithm and techniques for low-level processing are not novel, the interface to, and the interaction with, the knowledge base is a new way of dealing with the disadvantages of both top-down and bottom-up parsers. Therefore, such parsing schemes as ATNs, chart parsers and definite clause grammars are different from my approach because they provide (in most cases) a syntactic representation before using semantic or pragmatic information. NLPSs from Yale and SRI, however, are more on the line of my project and warrant a more detailed discussion.
3.5.2.1. Augmented Transition Networks

Based on [Woods 70], ATNs were originally conceived as parsing techniques and not theories of language. Nowadays, ATNs have developed into a grammar-like concept, especially for computational linguists, and are treated on a par with Transformational Grammars [Winograd 83].

A network consists of an arc labeled by a lexical category (e.g., Noun, Verb). While processing a given sentence, the pattern matcher embedded in the program "traverses the network" until a current word is matched with the respective arc. Recursive transition networks have arcs that may consist of phrases; if activated, control is transferred to the subnetwork (PUSH to a subnetwork), and after termination returned (POP) to the main net. These procedures increase the power of the grammar considerably. ATNs proper have the ability to change the word order, as well as insert/delete elements.

Registers are used to store condition flags and to retain temporary structures in global storage. Arbitrary conditions on the arcs make it possible to go beyond checking only adjacent elements. It is immediately obvious that these conditions make a much more efficient parsing possible than earlier transition networks that did not have the capability to use registers; e.g., in a sentence, a form of be before the main verb can set a trigger to check for a preposition by, which in turn can corroborate evidence for a passive or an active sentence.

ATNs use an expectation-driven, top-down strategy, whereby each arc contains expectations of what to expect as the next element. In case of an unsuccessful parse, ATNs have to backtrack, i.e., go back to the point in the parse where there had been a choice between alternatives. To minimize the back-up, heuristic parsing is used by some parsers. To avoid parsing phrases time and time again (in cases of multiple backtracking), once a well-formed substring is found, a flag is set. Chart Parsers achieve the same effect by keeping track of the well-formed substrings already parsed.
Sometimes numerical values are attached to arcs in order to assign priorities to various options. This splitting of arcs on the basis of numerical values adds to the ordinary depth-first search of ATNs a breadth-first dimension.

A problem arises with ungrammatical sentences for which there are no provisions made within the ATN framework [Charniak 81]. If a construction is encountered by the program that does not have a structural description in the implemented grammar, the program stops.

An ATN-type system that is conceptually related to the syntactic information processing of my parsing procedure is SHRDLU, a program that simulated a robot and accepted queries related to moving blocks displayed on a screen. Written in MICRO-PLANNER, a LISP-based programming language, Winograd tried to show that in order to be able to process language, the system must integrate information about syntax, semantics and pragmatics. Meanings are embodied in procedures that are activated whenever appropriate. Procedures are called to parse the constituents of a sentence. These procedures to call the subroutines correspond to the arcs of an ATN, the various levels in the program correspond to conditions and actions, and program variables correspond to registers [Winograd 83]. Apart from the good performance of the system as a whole, the parser fares poorly with respect to complex constructions, quantifiers, and conjunctions. Although conceived in a top-down, right-left technique, the parser uses neither backtracking nor parallel processing. If in doubt, the parser relies on semantic procedures that guide the parse. Should the parse fail nevertheless, the parser uses directed back-up, subsequently pursuing the next best choice. Apart from program-specific heuristics, multiple-source knowledge from the reasoning and semantic component aids the parser in determining referents for pronouns and deictic references.

As has been pointed out [Wilks 75], the success of this system is largely due to its limited domain and to ad hoc heuristics at various points in the processing of input. Using multiple sources for guiding the parse, however, is one of the most useful
techniques in arriving at a viable NLP program; by the same token, SHRDLU was the first deterministic NL program using semantic criteria to back up the syntactic processing.

3.5.2.2. Chart Parsers

Active chart parsers are based on two ideas [Winograd 83]: the use of charts as an indexing procedure for constituents; and the use of active edges for partial constituents. A chart is a data structure that contains an up-to-date record of the constituents or phrases found in the sentence during the parsing process. One of the problems arising from multiple sentence representations is thus avoided: once a well-formed substring is recorded in the chart-structure, its constituents do not have to be re-parsed (as was the case in some ATN programs).

Chart parsers make use of bottom-up and parallel parsing. In the Powerful Parser developed by Kay [Kay 67], the bottom-up parser (in the original version) scans from right to left while at the same time, grammatical rewrite rules apply to the constituent. If more than one parse is possible, all alternatives are recorded independently. To reduce the combinatorial overhead, Kay designed the concept of a chart as outlined above. Kay's Powerful Parser provides for each rule to find matching sequences of edges in the chart and to insert new edges. The chart makes the procedures too powerful. Since the interaction of the various rule-schemata is hard to control, any rule can insert new elements into the chart. A very efficient way of preventing the uncontrolled application of rules is to use flags for blocking rules from applying at certain points. The blocking of rules leads to a "ripple" effect, since now the rules have to be written with this special option in mind. The usefulness of chart-rewrite parsers is best summarized by [Winograd 83]: "In general, systems based on chart rewrite parsers contain grammars of Byzantine complexity that are quite difficult to understand and modify, although some of them have obtained a reasonably broad coverage of English after many years of work".

The earliest implementation of a chart parser was found in the REL system under
development since 1967 [Thompson 69]. It was originally written in assembler code for
an IBM360. It consists of a data base and an English parser. The chart rewrite
algorithm contains a set of semantic transformations and features for verbs similar to
the case grammar formalism.

The current interest in chart parsers stems mostly from the recently developed
MCHART. The goal of MCHART is "to provide a tool which allows the expression
and manipulation of the grammar in terms determined by the linguist for linguistic
reasons" [Thompson 81]. It consists of three modules for the user to manipulate: (1)
a signal table for determining rule invocation strategy; (2) a functional interface for
grammatical formalism; (3) a multilevel agenda for determining search
strategy [Thompson 81].

The system preserves the separation, found in chart parsers, between linguistic
formalism, search strategy, and rule invocation— a separation that promotes clarity and
modularity of the system. MCHART is compatible with the present system. The major
difference lies in the decision as to how to select various interpretations of possible
constituent structures. The present program does not compute all possible readings of a
sentence before deciding on one, but instead tries to interpret the sentence ultimately
on the basis of conceptual information. Furthermore, the present program uses a
combination of top–down and bottom–up strategies to limit the search space.

3.5.2.3. Definite Clause Grammars

With the advent of PROLOG [Clocksin 81], a programming language based on
symbolic logic, the relationship between parsing and deduction has been used successfully
in working on NL systems [Dahl 81, Pereira 80, Pereira 83]. A program is considered
to be a set of axioms; the process of computation consists of proving these axioms.
PROLOG consists of a pattern matching mechanism referred to as "unification" that
operates on general record structures. The data structures in PROLOG are called
"clauses". They can be compared to procedures containing variables, each of which
contains a "head" (the initial entry point) and a "body" that consists of "goals" (i.e.,
procedure calls.\(^{50}\)

PROLOG procedures are activated by providing a goal to the program, which will be subsequently matched against the head of a clause. Trying each clause in a linear order, the process of "unification" is successful if the goal and the head of the clause "match", i.e., appropriate values for the variables in the procedure are found. If no match is found, the system backtracks to the point in the program of the last match and explores alternatives. If more than one match is found, the system provides "non-determinate" results, i.e., lists the possibilities. Since the PROLOG-10 system consists of a macroexpansion facility that "translates a CF grammar into a logic program that functions as a recursive-descent parser" [Shapiro 83, 639], the grammar for a particular language can be input directly into the PROLOG clause form. An assessment of DCGs has to be based on a distinction between the capacities of the programming language and the linguistic formalism.

The linguistic side of DCG is very similar to the computational implementation of this grammar: the (context-free) grammar is axiomatized in "definite clauses", a subset of first-order logic. Thus, parsing is similar to executing a finite set of proof procedures. As has been pointed out [Pereira 83], the parsing algorithms are "on-line" that apply all the constraints applicable to a rule immediately after the rules has applied, whereas other parsing techniques are based on "off-line" parsing strategies and depend on two passes, the second pass applying all the constraints after the rules have applied.

Although the proof procedures can be implemented efficiently, problems similar to those for top-down, backtrack parsing procedures are found [Pereira 83]. The relationship between parsing and definite-clause deduction brings out the following issues:

- how to recognize derived clauses that are no longer applicable in an Earley-style parsing;

\(^{50}\) As with LISP, PROLOG does not overtly have machine-oriented constructs of reference/assignment (e.g., pointers).
- how to construct a restricted formalism for the distinct derived clauses; and
- how to characterize off-line parsable grammars theoretically [Pereira 83].

DCGs use a special form of chart parsing that might lend itself, due to the compatibility between linguistic and computational means, to a more powerful, perhaps more efficient, way of constructing NL interfaces. At this point, however, DCGs are at best in an experimental stage that might or might not develop into a viable alternative to LISP-based parsing. The only extant, though very limited, implementation of a PROLOG-based parser was done for Spanish [Dahl 81].

3.5.3. Cognitive Theory

The question of how well a NL system emulates human behavior (as a test of theories of human language processing) does not affect the evaluation of computational theories. In this section I discuss briefly a few extant systems that deal with psychological reality in the parsing process.

3.5.3.1. Deterministic Parsing

PARSIFAL was conceived as a computational implementation of Chomsky's EST. Its basic claim is that a deterministic parser can provide a theoretical base for the universal constraints that Chomsky's work postulates. "Deterministic" refers to the fact that Marcus' parser does not build up unnecessary structures during the parsing process, i.e., the parser constructs only one representation. Marcus' strong hypothesis originally claimed that syntactic parsing mechanisms alone suffice to simulate the linguistic behavior of a human. His parser operated with a limited look-ahead\(^{51}\). As became clear, a number of cases required semantic support for successful parses.

Deterministic parsing employs various techniques which set it apart from other parsers:

\(^{51}\) Computationally, the current constituent along with the two following constituents would be loaded into a three-cell buffer. The length of the buffer would then limit the look-ahead, and explain the "garden-path" phenomenon.
- there is no backtracking or parallelism necessary
- the parser uses limited look-ahead
- new nodes are created in bottom-up fashion
- the parser tries to operate primarily on syntactic information.

PARSIFAL consists of (1) a grammar interpreter, (2) a push-down stack for incomplete constituents ("Active Node Stack") which have to be dominated by a nonterminal node, (3) three to five buffers which contain constituents whose grammatical function is still unknown. The parser moves from left to right pushing an item onto the active node stack to suspend the formation of a constituent while the buffer cells always contain the rightmost nodes which the parser will process next. If a word gets into a buffer, the grammar, consisting of a set of production rules, determines if a new constituent has to be formed. If a new constituent is required it is put into the active node stack which grows downward. The interpreter looks at the contents of only the three buffers as well as at the bottom of the active node stack.

The current implementation of PARSIFAL shows that deterministic parsers based only on syntactic information cannot handle global ambiguities or garden-path sentences [Milne 83]. In both cases, semantic information is needed.

PARSIFAL includes three distinct features:
- it tries to prevent a combinatorial explosion of parses
- it excludes backtracking
- it clearly shows the limits of syntax.

A serious critique on Marcus' parser is found in [Church 80]: according to Marcus, one preposition takes up one buffer space (out of a total of three). A whole NP takes up only one buffer space since NPs are clearly marked by "leading edges". The criticism is based on the idea of having leading edges as justification for filling up a
buffer space in the case of NPs, whereas PPs, although they are also clearly marked by a leading edge, viz., the preposition, take up two buffer spaces. If a PP would take up one buffer, the notorious example in (5)

(5) The horse raced past the barn fell\textsuperscript{52} would not be a "garden-path" sentence since fell would appear in the three-cell buffer while raced is being processed. Further counterevidence can be found in [Briscoe 83, 63] where a number of "garden-path" sentences are presented that are processed as regular sentences by PARSIFAL.

The idea of deterministic parsing is crucial for a psychologically motivated parsing mechanism. PARSIFAL does not live up to Marcus' claims, mostly because of its autonomous syntactic processing. Additional pragmatic and semantic information is needed to have a deterministic parser that emulates language processing without delay, as shown in [Marslen-Wilson 78]. In the present system, the pragmatic information precludes a build-up of unnecessary structure. The significance of the foci in parsing the sentences in my program points towards a pragmatic solution to the few problems with deterministic parsing.

3.5.3.2. Syntactic Parsing Preferences

A class of parsers is built around syntactic preference principles which — according to psycholinguistic research — are used by native speakers to disambiguate syntactically ambiguous sentences. Consider (6):

(6) John carried the groceries for Mary.

The prepositional phrase [PP] for Mary can either modify the NP the groceries, or the entire VP. The two competing syntactic preference principles which predict these two interpretations are Kimball’s Right Association principle [Kimball 73], and Frazier and Fodor’s Minimal Attachment principle [Frazier 79].

\textsuperscript{52}Milne contends [Milne 83] that according to his experiments involving these experiments, most native speakers have problems with interpreting these sentences. His findings shed some doubt on the validity of introducing garden-path sentences as supporting evidence for a theory.
The Right Association principle predicts that native speakers interpret PPs or adverbs as modifiers of their rightmost constituent, whereas the Minimal Attachment principle predicts that PPs are attached to the VP to minimize the number of nodes. If the two principles are in conflict, Minimal Attachment, according to [Frazier 79], takes precedence over Right Association. Both principles are employed in a host of NLPS under slightly different names, the best known are Shieber's Shift–Reduce Parsing Technique [Shieber 83], and Frazier and Fodor's Sausage Machine [Frazier 79].

The syntactic preference principles are inadequate for the following reasons:

- The meaning of a sentence depends on its context, thus pragmatic principles have to prevail. Since pragmatic principles can conflict with syntactic principles, the NLPS has to be able to decide which type of principle to adopt. This conclusion is not in line with NLPS which a priori decides to follow syntactic principles.

- There exists a list of apparent counterexamples to Minimal Attachment (e.g., Sue wanted the dress for Mary [Schubert 84, 249]) as well as Right Association (e.g., John discussed the girl that he met with his mother [Schubert 84, 248]).

A solution to these inherent problems for parsers which work on syntactic preferences lies in the adoption of pragmatic principles which precede syntactic decisions. If pragmatic principles are always dominant, syntactic preference principles are superfluous in the NLPS's initial encounter with the text.

The pragmatic principles used in my NLPS are not obtained through simplistic generalizations of the syntactic principles. They are obtained from examining a large number of context types.

---

53 Consider the rules VP→V NP PP, and NP→NP PP for my sample sentence. If the PP is directly attached to the VP, no further nodes occur in the structural description of the sentence [Schubert 84, 247]. Further discussion of PP-attachment is provided in §1.2.3.

54 Cf. [Schubert 84] for further references
CHAPTER 4
TEMPORAL INFORMATION PROCESSING

4.1. PHENOMENA

This chapter deals with the two fundamental questions "what is the function of time in written text?", and, "how can it be represented by a computer program?"

Time has only one dimension; different times are not simultaneous but successive. Different events can overlap. Temporal information helps to organize events/objects; it provides a "reference frame" for the discourse. Different temporal phrases can refer to the same event; e.g., today, one day after my birthday, October 27.

Events, in spite of their extension in time, function as reference points in the discourse structure; e.g., the day after surgery.

Temporal information is provided by single linguistic units (e.g., nouns, verbs, adverbs), in and by larger linguistic components (e.g., sentences, texts), and by the factual context in which a discourse is set.

Temporal relations are expressed by calibrated dates and pseudo-dates. Both temporal relations as well as references can be "fuzzy"; e.g., two weeks ago, two weeks before admission.

Temporal references can be expressed in many different ways. Events that function as referent points demonstrate the duality of time with respect to its point-based versus interval-based nature; e.g., at admission.

Temporal inferences cannot be based on exclusively point-based or interval-based
systems; otherwise turning the radio on would comprise three different state of affairs: on, off, neither–on–nor–off.

The observations made above lead me to the conclusion that processing temporal information in discourse requires from the NLP a focusing mechanism in addition to a feasible representation scheme.\textsuperscript{55}

Temporal relations are commonly expressed with reference to fixed intervals (e.g., hour, day, year). These intervals determine the time-frame the discourse proceeds in. If there has been reference to a year, the expression "a while later" is understood in relation to the introduced time interval (e.g., year). These time intervals are anchored by a reference time (e.g., February 1; at admission) that is commonly viewed as a point in time. The reference time/point can be interpreted as the focus of the discourse. Once it is introduced, it is always used as a default value for relational expressions (e.g., five years previously). In domain–specific discourse, the reference point is ordinarily understood, even before explicitly introduced.\textsuperscript{57}

4.2. ISSUES

The processing of temporal information in my program is based on the separation of two levels: on the linguistic level, the program tries to extract from the temporal indicators the temporal information directly; on the conceptual level, the program integrates the information from co-text and context into the representation in the form of the tempus frame. Temporal information is provided:

- explicitly, by linguistic markers and their surface interaction (e.g., in Seven years before admission, jaundice was noticed, the preposition before and the verb tense count as linguistic markers);

\textsuperscript{55} See 2.6 for a functional explanation of the focusing mechanism, and 6.3 for the interaction between focus and inference.

\textsuperscript{56} The importance of focus in discourse has been extensively discussed for anaphoric references [Sidner 83, Grosz 81].

\textsuperscript{57} In the domain of patient histories, the initial reference point is always admission; everything else is related to it.
- implicitly, by the (linear) text structure (e.g., the following events are interpreted as having occurred one after the other only on the basis of how they are ordered in the discourse: The patient had a heart attack. Jaundice was noted).

- implicitly, by discourse relations based on world knowledge. (e.g., in the following sentences, we understand that the death followed the heart attack because of our world knowledge: He died. He suffered a heart attack.)

Parsing, the process used to transform linguistic information into a format that a computer can work with, is a subcomponent of discourse processing that interacts with these three knowledge sources.

Sentences consist of words and their meanings that provide information (e.g., concepts and their relations) and control (e.g., focus of the discourse) which is established by the discourse participants. Each word in a sentence has many interpretations depending on the empirical and linguistic contexts as well as on various connotations. Humans understand language without going through the whole list of possible meanings for words and sentences. Control (i.e., focus) refers to the capacity of limiting the number of possible meanings by the participants of a discourse according to their intentions. Focusing is a dynamic process that on the one hand establishes what the discourse is about, on the other hand, it determines the information contained in the discourse.

Discourse is progressing towards a communicative (i.e., shared) goal: understanding a given text. The two major driving forces are the focus on the domain-specific concepts in the sentence, and the focus on the temporal relations.

Language understanding does not encompass all information available at all times. Language understanding is a selection process. It relies on knowledge sources. Knowledge sources depend on memory and control structures. As we read a text, we make choices, start hypotheses, seek goals, and, above all, make decisions based on the

---

58 Linguistic markers and world knowledge also provide other than temporal information; the fact that sentences are linearly ordered, however, can be a means of expressing temporal interaction between events.
information available at any given time. In computational terms, the reader determines the search space based on expectations about the subject area of the text, selects information operators, and focuses on the necessary information indicators (e.g., time and topic).

Linguistic information is processed on different levels concurrently. Language understanding starts on the word level by selecting and identifying word-concept relations, grouping them into larger "chunks", and context-dependently matching these "chunks" to higher order memory structures.59

In psycholinguistic terms, morphological, syntactic, semantic and pragmatic information interacts; in computational terms, each level of decision-making changes the search space and the control structure; based on new information being processed. The algorithm for this focusing process has to consider the event-driven control structure by establishing (focal) data structures that "anchor" the control procedures. The focusing process originally determines what is being focussed on, while after the focus is established it determines what is being said.

4.3. THE APPROACH

The issues in temporal information processing hinge primarily on contextual considerations with respect to the macro- and microstructures of discourse. Therefore, a pragmatic approach is used to represent temporal relations in NLP (as opposed to a semantic or syntactic solution). The following reasons illustrate the benefits of our current approach:

---

59 A sentence is analyzed by the X-bar rules given in 3.4. Each word has a set of semantic features. One of these features may indicate that the word refers to a significant concept within the particular domain. If it does, the frame access procedure which applies to each phrase, instantiates the concept which eventually will be hooked onto the predefined, prototypical event structure. After the structures are built up and matched they are finally stored in memory. Each processing stage changes the predisposition of the next one. While some processes go on simultaneously, others follow certain sequences. The complexity of interaction arises from having to determine at what point during the processing various competing knowledge sources have to be invoked.
1. The deictic use of tenses: as Partee points out, sentences like (7) refer to a specific point/period in the past that is dependent on the discourse in which it is used.

(7) I didn't turn off the stove.

Since tense operators are used in relation to an indefinite point in time, the sentence above could be evaluated in tense logic only if for any time in the past the stove was turned off or not. By the same token, the negation of the sentence causes further complications since it does not imply that at no point in time the stove hadn't been turned off; but, that at a specific point in time this was the case. The use of indexes is necessary to capture this reading, thus pointing to a pragmatic solution.

2. The imperfective paradox: as Bennett [Bennett 78, 14] points out, it is necessary to include the notion of a "sentence being true at an interval in time" in an analysis of tense and aspect to solve the imperfective paradox. Since the sentence in (8) allows the inference in (9), we have to consider the context of the sentence to decide whether the inference is warranted (e.g., in the case of John was building a house before he died, it is not warranted). A pragmatic solution is called for.

(8) John was building a house.

(9) John built a house.

3. The nonlinguistic element: depending on our world knowledge, we understand sentences like the following as expressing different time relations between the events:

(10) John ate a doughnut and drank a cup of coffee.

(11) John drank a martini and ate scampi.

---

Aspectual issues are crucial while dealing with discourse structure [Dry 82, Dowty 82]; due to their complexity and the nature of my discourse domain, however, they are only mentioned in passing.
John went to church and to K-Mart.

Even if the sequencing of events is clear, the duration of the events is not clearly marked by linguistic means; it is open to context-sensitive interpretation; moreover, we interpret the perceived temporal order of a text not "simply by virtue of the times that the discourse asserts events to occur or states to obtain, but rather also in terms of the additional larger intervals where we sometimes assume them to occur and obtain" [Dowty 82]. Again, pragmatic considerations play an important role in the understanding process.

4. Temporal anaphora: phoric references necessitate a well-defined discourse structure. The interplay of tense and time adverbials as well as the sequencing of events by tenses and aspectral classes of the predicates go beyond the sentence boundaries [Hinrichs 81].

4.4. TIME

The fundamental question before proposing a theory on temporal relations consists of whether to view time as consisting of intervals or points. Should temporal relations be described as points without our extensions, or should they be interpreted as events that depend on our experience?

My decision to adopt an interval-based approach for my system is not justified on ontological grounds in my dissertation. It is a result of my conviction that events that we take part in or witness have their extension in time. Events are only secondarily used as reference points.

The function of time in understanding language has been treated as just another

---

61 There are other philosophical theories on time that are more abstract, e.g., Kant's mental category, Poincare's theoretical convention, Heidegger's phenomenological interpretation.

62 I agree with Kamp [Kamp 79, Hinrichs 81] that point-based theories might stem from an uncritical transfer from physics to philosophy.
linguistic phenomenon (like quantification or scope), or just another logical phenomenon (like tense operators). In linguistics and logic, the issues addressed do not take into account the all-encompassing nature of time with respect to discourse understanding. Time is at the very center of all other problems in linguistics and logic. The approach I take to temporal information processing in this dissertation is based on the thesis that the event frame\(^63\) is established by temporal indicators to represent temporal information. Temporal information provides the structure for the discourse. Temporal information includes:

- when something happened;
- how long it lasted; and
- how events are related to one another in time.

Temporal indicators include:

- explicit dates;
- relational terms; and
- fuzzy expressions.

The first indicator requires only the recognition of a linguistic marker (e.g., October 5, 1982); the second requires the identification of linear structure (e.g., before–after); the third requires inferences based on discourse relations as in (13), and world knowledge as in (14):

(13) **Two years ago, I met him in Paris; a while later he died. Two weeks ago, I met him in Paris; a while later he died**;

(14) **Two years ago he had a broken leg; he was fine two months later. Two years ago he had an ear cut off; he was fine two months later**.

---

\(^63\) The event frame, in my theory, is the conceptual and computational construct which together with a temporal tracking procedure provides a hierarchical structure for storing temporal information in the knowledge base.
Temporal references are expressed in and by sentences. The temporal information that can be extracted from any single time indicator in isolation has to be integrated in the whole discourse. Although the parsing itself is an interactive process between knowledge base and the syntactic/semantic component, temporal relations between and within the discourse units have to be established by pragmatic procedures, since the establishment of temporal relations involves the new piece of information in the text with respect to the discourse as a whole. The interaction of the various levels of processing and the various knowledge sources is illustrated by a discussion of the following examples. Three levels of processing are identified:

- the linguistic level, containing explicit markers;
- the discourse level, establishing implicit sequences of actions; and
- the conceptual level, determining world-knowledge relations.

On the linguistic level, explicit temporal markers are used to convey a time frame for the discourse (e.g., tense, adverbs, prepositions, conjunctions). In (15) and (16), the tense markers make explicit the temporal relations in the sentences.

(15) John took out the garbage. He had cleaned his house.

(16) John had cleaned the house. He took out the garbage.

On the discourse level, a sequencing of the events could be achieved by simply ordering the sentences; instead of using tense markers, the same effect as in (16) is found in (17).

(17) John took out the garbage. He cleaned his house.

It is usually understood in discourse that the order of events reflects their actual occurrence (in literary parlance: time "moves" forward in the second sentence). This observation does not apply when events happen concurrently, or, if the second sentence is in the progressive, regardless of the predicate type.
The problem of concurrent events is analyzed by the program in terms of \textit{aktionsarten} of the verbs that are classified into classes (e.g., \textit{accomplishments}, \textit{achievements}, \textit{states}, \textit{activities}) by [Vendler 67]. However, as [Verkuyl 72, Dowty 79] point out, the classification might also depend on what kind of NP or PP the verb takes; thus, in (18), the activity verb in (19) can turn into an accomplishment verb due to the following PP:

(18) \textit{Mary walked to church.}

(19) \textit{Mary walked.}

On the conceptual level, event sequences suggested by the discourse structure can be interpreted with regard to their "logical" order. In (20), our knowledge about the world renders it impossible for someone to die and then have a heart-attack.

(20) \textit{He died. He had a heart attack.}

Along these lines, narrative techniques (e.g., historical present) can suspend tense markings to attain certain effects (e.g., suspense) while relying on the Natural Language Processor to adjust the time frame.

The various relations between events (e.g., overlap, precedence) are mostly established relying on the empirical knowledge the NLPS has.

The parsing of fuzzy relations and events of implied duration relies more on the NLPS's knowledge about events and their stereotypical structure than on explicit linguistic markers.

From these examples it becomes apparent that an analysis of temporal references has to start on the word level to determine semantic properties of time referents within the sentence and, then has to continue on the discourse level. Temporal information is processed in my program in three steps: the linguistic analysis of the sentence, its
representation in the event frames, and its subsequent integration process in terms of possible inferences.

4.4.1. Linguistic Analysis

In the past, the linguistic analysis of time has been based on syntactic and/or semantic categories [Leech 69]. The semantic division (e.g., duration, frequency) often cuts across the syntactic categories for time (e.g., adverbials, tense, aspect). A pragmatic account of time relations primarily deals with the function and expression of temporal information within a given context. The pragmatics of time in English is established in the program by (a) determining what establishes the event sequence; (b) deciding what the temporal markers are and how they interact; (c) finding out how new information is integrated. The discussion of these issues leads to a set of features that can be used by the system for representation and reasoning purposes.

It will be shown that the event frames are anchored by a temporal reference point that is extracted from the cotext, or the empirical and linguistic context. The dimensional orientation (e.g., past, present, future) of the event frame is established by markers on the verb and by the particular aspects that verbs express (e.g., inchoative, continuative); this dimensional orientation is interpreted by the system as "background" (e.g., adverbials, conjunctions). Temporal markers as well as the temporal meaning of individual words make up the "foreground" of the text. On the text level, the linear structuring of events is an implicit marker that can be used to set up the event frame. The integration of new information is done with respect to these levels of temporal information carriers: word, sentence, discourse. The complexity of the integration process stems from interacting grammatical and semantic categories, as well as from the multiple relations between events.

[^64]: I provide this "algebra" only for past events since the particular type of text I am mostly concerned with is characterized by dealing with past, aspectually not marked events/states.
4.4.2. Event Frame

The event frame can be described as the background information necessary to classify events/states/actions in terms of past, present, or future. It also includes aspect and inchoative, continuative, conclusive, and iterative aspects of verbal meaning. The following discussion deals exclusively with past events and various features of verbal meaning.

On the sentence level, the tense of the verb ordinarily\(^{65}\) determines the time reference. The use and meaning of various tenses relates three temporal connectors: speech time, event time, and reference time [Reichenbach 47]. In the theory of tense that Reichenbach proposed, temporal information processing is completely characterized only by introducing a reference time in addition to the time of speech and the time an event is said to have taken place. The speech time is defined as the time of utterance; the event time refers to the time the reported event takes place.

The notion of reference time that Reichenbach introduced captures the fact that once an event appears in a text, it can henceforth be used as a quasi-speech time, or a stepping-stone for structuring the ensuing discourse. In (21), the previous event time yesterday can be re-interpreted as reference time by using the past perfect:

\[(21) \text{Yesterday he died. He had lived a happy life.}\]

The crucial point that Reichenbach makes in [Reichenbach 47, 288]\(^{66}\) can be interpreted as an emphasis on the pragmatic, i.e., contextual determination of the reference point by the co-text (i.e., the collection of sentences in the text which does not contain the current focus) [Dijk 72, 39]

\(^{65}\) Exceptions are mostly found in the present tense markers (e.g., I leave tomorrow)

\(^{66}\) The determination which time point is used as reference point in the sentence Peter had gone, "is rather given by the context of speech. In a story, for instance, the series of events recounted determines the point of reference, which in this case is in the past, seen from the point of speech; some individual events lying outside this point are then referred, not directly to the point of speech, but to this point of reference determined by the story."
On the word level, the inchoative (e.g., begin to suffer, start to hurt) and continuative (e.g., keep complaining, continue to hurt) aspects of the verb phrase contribute to the formation of the event frames. The gradual onset or end of an event is used to determine the period of time in which the event took place.

On the discourse level, the program uses the linear sequence of sentences to integrate data concepts into the temporal structure of the knowledge base. Information contained in sentences without explicit time references is integrated into the tempus frame by using the global time focus as reference interval. In our sample text, are numerous examples in which the time interval of the previous sentence functions as the current time interval. The time focus also marks only temporal changes. In the sentence Four weeks passed, the time focus changes without having a data item attached to it.

4.4.3. Time Relations

On the sentence level, temporal indicators establish time relations. Temporal indicators are
- time adverbs
- conjunctions
- prepositional phrases
- tense.

The temporal indicators stand in anaphoric relationships to the co-text which go across sentence boundaries [Hinrichs 81]. The tense marking in my text is not as significant as the other indicators.

On the word level, the temporal information depends on the individual conceptualization of terms (e.g., the words fever, death or jaundice conjure up different extension in time for different people), and the type of verb used (e.g., activity,

---

67 Issues related to verbs and temporal processing are not discussed extensively in my dissertation.
achievement). In a medical text, the mentioning of fever five weeks before admission may not explicitly be followed by its time of termination. Defaults on the medical concepts could indicate in the data base that there was an episode of fever five weeks before admission. Since fever does ordinarily subside after a few days, a possible default for fever is one week. Other concepts (e.g., diabetes) take up a longer time period and often cannot be limited as to their endpoints.

On the discourse level, temporal conjunctions and the interaction of the temporal relators within the event frame establish the temporal context for new information to be integrated. On the discourse level, global parameters for focus which indicate domain-specific concepts are used to process the incoming information.

4.4.4. The Integration Process

The interaction of various grammatical and semantic categories determines the temporal complexity of the text. The system integrates incoming temporal information by relating events to each other on the basis of lexical features which indicate duration, frequency, and the beginning and end of events.

Each processing step changes the predisposition for the next action the system has to take. On a low level, the system identifies the lexical components and makes the first choice in determining the referent by accessing the knowledge base; grouping these words into phrases is the second decision; every further step will influence the decision process, so that its final outcome is based on the sequence of choices the system made.
CHAPTER 5
INFERENCES

5.1. WHAT IS AN INFERENCE

Inferences are distinguished from logical deductions in that they may be "true" or "false" and that they are context-sensitive. In general, an inference is a piece of information derived from pieces of given information. The given information can either be a set of propositions or a set of statistical data which led to the conclusions on the basis of what is believed to be the case. The first class of inferences I call qualitative inferences, whereas the second class I refer to as quantitative inferences.

In the cognitive sciences, two ways of describing the reasoning process of inferencing are distinguished: bottom-up versus top-down. The former method is also referred to as data-driven or event-driven reasoning since the inferences are drawn on the basis of current input; the latter is goal-driven or expectation-driven since it is primarily concerned with a specific goal. It is often necessary to combine both ways of reasoning.

A final distinction that I would like to draw is that between inferences based on discourse structures and those based on conceptual structures. In my system, conceptual structures are based on a "closed-world" assumption similar to formal deductive systems that work within a finite number of schemata, as mentioned in [Barwise 83, Gibson 68]; whereas linguistic structures represent an infinite number of meanings that depend on and can be recognized in the context. The meaning of a term determines its extension; it is not a matter of the speaker's psychological state [Putnam 75].

In contradistinction Schank's assumption that the "amorphous" nature of inference generation is a "reflex response" [Schank 74], I contend that linguistic inferences are
extremely structured and predictable within a system that uses pragmatic knowledge. The active use of contextual knowledge sources (e.g., co-text, intentions, sublanguage) distinguishes my approach from the static use of knowledge structures (e.g., stereotypes) in Schankian systems.

The theoretical implications of my distinction between a conceptual and a linguistic level of reasoning could be an explanation for the dilemma of how humans can reason based on only incompletely specified knowledge. While the reasoning process works with plausibility assumptions (e.g., at a given time t, assume p in the absence of counter-evidence), it is based on systematic inference procedures that employ detected schemata. My model combines these counteracting forces by assuming that, on the linguistic level, inferences about partially specified situations are made on the basis of pragmatic information. These inferences are context-dependent and ordinarily do not follow first-order logic. In experimental studies [Wason 72], logically naive speakers of English show that people do not have "a natural syllogistic machinery accessible to consciousness" [McDermott 82, 104]. If inferences were always context-dependent, rational behavior would be hard to explain since no general process could be found. I contend that the difference between rational performances lies in the ability to discern 'what is said' and 'what is meant'. The differences in reasoning could be explained by attributing an homogenous level of conceptual structures to all speakers of language, whereas an heterogenous level of lingustic structures depending on the speakers' attunement.

5.2. INFERENCE AS A PROCESS

The task of the reader is to process the literal form of a message to get to the intended meaning. The inferential process requires a certain amount of time as common

---

68 Schemata are cognitive and socially defined representations [Bartlett 32].
experience and psychological tests indicate. During the understanding process, the NLPS establishes "missing links" [Brown 83, 257] which are distinguished as to whether they require additional processing time. If the identification of missing links does not require additional processing time, as experiments described in [Sanford 81] indicate, they are hypothesized to be part of the knowledge representation and as such are "automatically" activated. If the identification of a missing link requires additional processing time, as experiments indicate [Haviland 74], they are hypothesized to be based on "bridging assumptions" and therefore are nonautomatic. An obvious conclusion would be to refer to as "inferences" only those processes which require additional processing time, and distinguish them from automatic connections. The characterization of automatic connections based on the knowledge representation causes the following problems:

- Some connections are either automatic or nonautomatic depending on the individual speaker.
- Some connections cannot be described by a decomposition of meaning.

The issues in studying inferences are succinctly summarized as follows:

We have argued against equating inferences with any form of connection.

69 In [Clark 77], the inference based on the two sentences Mary got some picnic supplies out of the car. The beer was warm consists of the "bridging assumption" that The picnic supplies mentioned include some beer [Brown 83, 257].

70 In the examples (a) Mary dressed the baby. The clothes were made of pink wool, (b) Mary put the baby's clothes on. The clothes were made of pink wool, informants did not take longer in understanding the second sentence in either (a) or (b), although they had to "fill in" a missing link (e.g., Dressing is related to clothes) in (a).

71 In the example given in [Haviland 74], the reaction times for the two sets of sentences (a) Mary got some beer out of the car. The beer was warm, and (b) Mary got some picnic supplies out of the car. The beer was warm, were measured revealing that for some people beer is part of picnic supplies and thus is an automatic connection, while for others it is not and therefore nonautomatic.

72 In [Brown 83, 265], the authors argue against the idea that "when a verb like dress is encountered, this will evoke from memory a representation which contains slots for a variety of entities implied in the meaning of the verb, such as clothing" [Sanford 81, 108], by pointing out that a number of connections would be redundant if these slots were activated whenever a particular concept is mentioned.
between sentences in a text. We have emphasised that inferences are connections people make when attempting to reach an interpretation of what they read or hear. We have also suggested that the more interpretive 'work' the reader (hearer) has to undertake in arriving at a reasonable interpretation of what the writer (speaker) intended to convey, the more likely it is that there are inferences being made. The problem with this view is that it leaves 'inferencing' as a process which is context-dependent, text-specific, and located in the individual reader (hearer).

While we believe that this is the correct view and that it is, in principle, impossible to predict the actual inferences a reader will make in arriving at an interpretation of a text, we may be able to make predictions regarding particular aspects of individual texts which readers will generally have to interpret on the basis of inference [Brown 83, 265–266].

The implications of this argument for my research are twofold: (1) a model of inferences can make predictions only about the procedures that are part of the inferencing process, and not about the actual inferences. (2) A model of inferences can be tested only by using plausible comprehension criteria that are generally acceptable (e.g., question-answering).

In general, a model for the inferencing process is dependent on the knowledge sources and their control structure. In discourse analysis, the knowledge sources are the intentions of the NLA, the text type, and the co-text. I distinguish two types of inferencing procedures: (1) on-line inferences that are made to "understand" the text in order to form a representation of it; (2) inferences in a query mode that are made to "interpret" the representation of the text. The former are inferences based on the linguistic input that result in a conceptual representation; the latter are inferences based on the conceptual representation that result in a linguistic output. The notion of linguistic primacy of the reasoning process attributes the individual differences between speakers to their facility to use and comprehend language.

---

73 This corresponds to Johnson-Laird's comments on the significance of mental models that "it is not their phenomenal or subjective content, but their structure and the fact that we possess procedures for constructing, manipulating and interrogating them" [Johnson-Laird 81].
In the following discussion about the various knowledge sources that are responsible for the production and comprehension of a text, these sources are considered separately only for the purpose of our discussion. In the program, sources interact as structural and lexical information is being processed.

5.3. FOCUS AND INTENTION

The reader of a text relies on his knowledge of particular text types in order to decide what and how information is to be processed. The reader also takes into account that the author's intentions\textsuperscript{74} are sometimes different from his own. In a highly structured domain-specific text (e.g., medical histories) the intentions of the reader and author are congruent since the author can espouse his ideas only within limits established by the audience. In a narrative, other criteria apply.

In general, texts contain various themes that the author elaborates on. The notion of theme or topic has been interpreted either as a constituent in a sentence [Grimes 75, 337], or as a proposition [Keenan 76, 380]. Both interpretations suffer from a much too narrow definition of the term. The topic of a text is not adequately represented by a single constituent. In a sentence like Mary hated John for leaving her, the topic cannot be localized in any one constituent, but in the meaning of the sentence as a whole. By the same token, it is more than a proposition that constitutes the topic here.

In my approach, the topic is defined as a number of foci that are determined by the author's and reader's intentions and by the text type. If the text requires a rigid structure, author's and reader's intentions are secondary to the text structure. The text type "prescribes" the foci which also reflect the intentions of the author and the reader.

\textsuperscript{74} Intensions refer to the conveyed meaning embedded in a text; they do not imply any philosophical issues. The author's "conveyed meaning" is the reader's intention in the case of highly structured technical texts.
Focussing is the process that determines what is central to a text, as well as what is perceived by the readers to be the content of the text [Grosz 81]. It is a self-defining process. The interaction of all the knowledge sources requires an algorithm that determines what knowledge becomes relevant at which point in processing. The initial problem for the reader is one of selection: finding the new information in the text on the basis of what she knows already. The reader has to (a) determine what is relevant, and (b) which knowledge sources he should use. Both tasks can be achieved by first finding the topic of the discourse, then activating/accessing knowledge which is relevant if such a knowledge is present.

In [Sidner 83], the focus within a sentence is determined by an algorithm that first tries various constituents as candidates for serving as the focus of the sentence. The default focus is the VP of the sentence. This approach is problematic as pointed out earlier since most foci cannot be determined just on the basis of constituents or propositions of a sentence.

In [Grosz 81], the focus is more domain dependent. Within the domain of assembling objects, certain tools and configurations of parts become important at various points in the discourse. Grosz' particular study is in the area of dialogue. She tries to find procedures that keep track of the foci established by the interlocutors.

Both studies [Sidner 83, Grosz 81], investigate aspects of discourse understanding by using foci to control the information process. "Focus" here means an agreement between the speaker and listener to establish common denominators for exchanging information.

In my study, I propose two types of foci: a temporal focus which allows the interlocutors to share a common event structure, and a thematic focus which is determined by the specific text or sublanguage in which the discourse is set. My

---

75 The studies emphasize areas of research that are different from mine. My study is about domain-specific text understanding by a NLPS, and neither about a dialogue situation, nor about finding the focus in any domain.
interpretation of the thematic foci (and procedures for finding them) as the intentions of the NLPS and the author of the text in a domain-specific situation is the starting point for analyzing the discourse. My contention is that in a domain-specific text, (e.g., case histories of patients), the intentions of the NLPS and the author are determined by the text type itself. For other forms of discourse (e.g., narrative), the intentions of author and NLP are different and not determined by the text type.

5.4. INFERENCES AND TEXT TYPE

The formation of inferences while reading a text are facilitated by the reader's prior knowledge of the characteristics of a given text type. Studies of medical summaries [Bonnet 79] have shown that from the sequences of schemas the reader builds up an expectation of what information comes next. If the structure of a schema is interrupted by a temporary topic shift, the reader can detect a possible inference.

Evidence from studies of medical texts [Bonnet 79, Hirschman 83] as well as psychological findings about memory [Bartlett 32, Collins 72] support my contention that the components of a specific text and its structure help the reader to process the information by realizing the author's intentions and their representations. The information about text components and text structure can be put into a program. If the program can process the text and retrieve the information when asked to do so, it provides supporting evidence for stipulating particular components and their structure.

In my theory, a domain-specific text is defined as an ordered set of topics which are

---

76 Medical summaries are highly structured and can be broken down into topics which are represented by schemas. Topics include symptoms, signs, and laboratory data which are mentioned in a certain order in the text. "Schema" indicates that these topics are psychological entities that lead the NLPS to expect information related to single schemas as well as related to the schematic structure of the text.

77 An example of a topic shift is the mentioning of the ordering of a laboratory test (which is a topic by itself) while discussing within the topic "past-history" the fact that the person drinks a lot of alcohol. If the result of this test is not given, the reader may assume that it is negative and that the initial lead (e.g., liver damage) should be disregarded.
represented by schemas. The topics are expressed in a particular form of language called a "sublanguage".

We define sublanguage here as the particular language used in a body of texts dealing with a circumscribed subject area (often reports or articles on a technical specialty or science subfield) in which the authors of the documents share a common vocabulary and common habits of word usage. As a result, the documents display recurrent patterns of word co-occurrence that characterize discourse in this area and justify the term sublanguage. [Hirschman 83, 26]

Whereas on a higher level of processing, the topical structure facilitates the understanding of a text; on a lower level it is the specific sublanguage. Domain-specific discourse uses a restricted grammar and has fewer ambiguities. In the medical context, the fragment no liver refers to a test and means no liver felt. However, as has been pointed out in [Hirschman 83], syntactic processing is necessary even for a limited domain, especially to determine modifier-host relations, as the differences in the following examples show: Had fever 1 day prior to admission versus Had 1 day fever prior to admission.

5.5. TEMPORAL INFERENCES

Temporal inferences differ from thematic inferences in many respects. The impact of these differences on representation and reasoning with respect to temporal information is discussed in detail in this and the following section of this chapter.

- Temporal inferences are used for the representation of events. Temporal information structures the collection of the events described in a text. Thematic inferences depend on the NLPS's temporal structure.

- Temporal inferences are used for reasoning about events. After the temporal information is represented, to answer any queries, the NLPS has to use the relational knowledge pertinent for the respective query to be answered.

- Temporal inferences also depend on the nature of the events. The duration and causal relations between individual events shape a single temporal inference.

- Temporal inferences are not domain-dependent. Thematic inferences depend on the particular domain the NLPS is in. In the domain, a special expertise
Temporal information is necessary to deal with thematic reasoning processes. Temporal information processing does not change the conceptual framework. It might be necessary to consider different time units (e.g., in computer science msecs are used, whereas in archeology, the units are larger).

Temporal information is provided on all levels of processing a text. The time units (e.g., year, month) express the perceptual differences related to the passing of time. The perceived differences allow humans to organize data. Events are occurrences of data at a given moment in time. The organization of data into events and the chronological representation of events depend on a particular reference point from which the event structure is organized. The vantage or reference point changes during a type of discourse. In technical texts, it ordinarily remains the constant. In a patient's case history, significant events are used as reference points. Rather than referring to June 15 and August 20, the author of the text uses at admission and at surgery. The significant events are therefore used by the author as well as by the NLPS to "anchor" the event structure. The event structure establishes the time frame which is based on time intervals.

During text processing, the NLP has to keep track of the reference point and the changing time units. Another problem is to keep the time frame for the discourse consistent. In particular, how are relational and fuzzy references related to the chronology of the discourse?

The knowledge about reference points for particular texts is necessary for the NLP to find the relevant anchors for the time frame of the discourse. In the following sentences, the differences in finding a suitable reference point is apparent.

In (22), admission is a reference point which is used to anchor the time interval "six weeks".

---

78 The reference points within a text are ordinarily the "key events". With regard to "key events", I contend in my theory that they provide the temporal "hooks" for the comprehension of domain-specific texts.
(22) Six weeks before admission, the patient had stomach aches. In (23), "meals" is not a reference point for anchoring the discourse because the patient's history is not reported with respect to when the meals take place.

(23) Two hours after meals, the patient had stomach aches. Two hours after meals identifies more closely the event of having stomach aches and not how their occurrence relates to the overall event structure of the discourse. The event of having stomach aches two hours after meals has to be embedded in a time interval within the time frame.

The selection of the reference point can be domain specific (e.g., technical texts), or context specific (e.g., narratives). Structural criteria (e.g., the NP which follows after or before are not suitable as (19) and (20) show. Therefore, the NLPS has to resort to knowledge about the domain.

My program finds the time intervals including their numerical or fuzzy values, instantiates the generic event interval frame before inserting the occurrence of a data item (e.g., jaundice)\(^79\) into the particular instantiation of the generic event interval frame.

The focus is always a certain time interval (e.g., month, year) connected to a reference point (e.g., admission, surgery). The control structure of the program keeps track of the current temporal and thematic foci. The foci are needed to support the instantiation process. If fuzzy references (e.g., a while later, after that) are used, they are interpreted (a) with respect to the established time-interval (e.g., year, hour), and (b) with respect to their significance (e.g., the exact time period between reference event and related event might not matter).

\(^79\) These data items are classified in terms of possible foci within a specific domain. Jaundice is a symptom. Symptoms are foci in the domain of medical descriptions. They are represented in generic frames that contain slots for further specification of the individual concept. Specifications are expressed in terms of intensity of the symptom, when first noted by the patient, etc..
Computationally, the time focus and topic focus are established by an algorithm that considers (a) the specific text, (b) the first sentence(s) of a text, and (c) the text type. The final result of processing the text is the event as illustrated in figure 1–3.

Each focus is kept in a stack which compares potential new temporal and thematic information. After low-level processing in terms of words and phrases is completed, each phrase is searched for corresponding frame representations of relevant concepts that are considered themes and time references. In the case of patient histories, possible topics are signs, symptoms, lab data, and patient data.

The temporal focus is based on given reference intervals (e.g., year, day) that are used with respect to a reference point. In our case, admission is normally the reference point; in other types of text the reference point has to be extracted from the first sentence(s). In patient histories, this method would also lead to choosing admission as the reference point, since it is almost always mentioned in the first sentence.

5.6. ON-LINE INFERENCES

Information storage and retrieval are two different operations of the NLPS. Storing incoming information involves an organization of reported events. Inferences are drawn that enable the NLPS to process information on-line which results in a representation of knowledge. Retrieving information involves a reasoning with the previously organized events.

The distinction between the two types of inferences results from the fact that humans do not establish all possible relations between events when the information comes in, but that they process information for abstracting knowledge first. The representation of knowledge is therefore a different process from the reasoning with knowledge.

---

80 An analogy to these processes is the distinction between short term and long term memory. Storing the information involves processes in STM which lead to an abstraction of the information subsequently stored in LTM.

81 If this were the case, processes referred to as "remembering", "recall" and "reasoning", which take considerable time as numerous psychological experiments have shown, would be performed without delay.
On-line inferences rely on previous knowledge about the text type, foci, structural knowledge and conceptual configurations. Temporal on-line inferences rely on the structure of the time frame, the time intervals which are combined in the time frame, the temporal foci, and the linguistic indicators.

Linguistic indicators include temporal knowledge about (1) the items in the lexicon (what is the extension in time of events referred to as fever or jaundice, etc.), (2) the relations on the phrase level (how do adverbs and prepositional phrases interact), and (3) their order of occurrence in the text. The temporal information expressed by the linguistic indicators is chosen with respect to the time focus. Each time focus corresponds to an interval which is integrated into the event frame.

Temporal on-line inferences include:

- Finding the reference interval for each event. If the sentence contains more than one event, what determines the proper temporal reference for the individual event.

- Monitoring the reference interval. If a sentence does not have an explicit temporal reference, the program has to determine which reference interval applies.

- Integrating "fuzzy" references (e.g., a while later, after that). If a sentence contains only relational expressions that connect two sentences, the program has to determine how to establish the proper relation and subsequent representation of the information.

The result of temporal on-line inferences is a feasible representation of events which can be used for reasoning about the relationships between events. Each time a new temporal focus is encountered by the program, a new event frame is instantiated. If thematic frames are instantiated, they are connected to the respective event frame. The connection takes place usually at boundaries in the sentence or at the end of a sentence. The specific data concept is also inserted as a slot into the generic temporal interval frame. The information is stored in two ways:

- The instantiated event frames contain a slot with the instantiated data concept. Numerical or fuzzy values are stored in the "Reference-to-Key-Event" slot. At this point, no explicit chronological ordering procedure is
invoked. The ordering of events takes place when a query is put to the system. The type of information processing related to the instantiation of event frames allows the system to answer questions relating to temporal criteria (e.g., what signs were present a year before admission?).

- The name of the data item is placed as a slot onto the generic temporal frame (e.g., year) together with the numerical or fuzzy value. The answer procedure invoked by the query asking for occurrences of specific data items can access this information (e.g., when did jaundice occur?).

The storage of the information at the stage of on-line processing consists of a knowledge representation without explicitly ordering the events. Before a text is processed, the data base contains only the generic frames. After the text is processed all the instantiated frames have been added to the data base. They are at that point connected via the event structure generated during processing. Temporal inferences are made on two levels. At the on-line level they use lexical, syntactic, semantic, and pragmatic information to provide a knowledge representation. In the query mode, temporal inferences involve calculations based on the knowledge representation, and on heuristics which apply to fuzzy temporal references.

---

82 In the current implementation, the system chooses admission as the reference point if none is mentioned in the first sentence. The reference point also functions as a focus for a temporal interval in which data items are found. If my sample text is run through the system again, the time interval of the last sentence replaces the time interval "admission" which was identified in the first run as the temporal focus in the first sentence.
CHAPTER 6
APPLICATION - MEDICAL EXPERT SYSTEM MDX

6.1. EXPERT SYSTEMS

The design of expert systems and NL programs has become an active area of research in Artificial Intelligence, a branch of science that tries to explain what intelligence is, as well as to find out what is required to produce "intelligent" machines. Based on the classic distinction made by McCarthy and Hayes [McCarthy 69] between an epistemological and a heuristic part of the AI problem, most expert systems show a dichotomy between the representation and the processing of knowledge. Research in NLP adds a third – a linguistic dimension – to their dichotomy [Waltz 83].

Expert systems are computer programs that are capable of performing tasks ordinarily requiring a well-trained specialist in a given domain. In their performance, they go beyond the mere alphanumeric data processing of traditional computer programs by exhibiting a reasoning capacity derived from an individual knowledge base and a set of heuristics, or rules of thumb, that were previously gathered from a human expert. Expert systems are distinguished from traditional computer programs by their ability (1) to perform problem-solving tasks based on judgmental and uncertain knowledge; (2) to explain their reasoning process; and (3) to acquire and assimilate new information.

The user of an expert system has the following options in storing and accessing the information in the system: (1) use of the system's programming language; (2) menu; (3) query language; (4) graphic query language; (5) natural language front end.

Although formatted approaches [see (1)–(4) above] to information processing have the advantage of being precise, unambiguous, and powerful, they achieve these virtues only at a very low level of abstraction.
The purposes of this chapter are to discuss (1) practical aspects of NL processing systems (e.g., tasks, feasibility, evaluation); and (2) theoretical implications of extant programs (e.g., typology, theoretical bias); as well as (3) to demonstrate how my pragmatic parsing technique that uses co-text and context fares in the MDX environment.

Scientific study of NL has shown the complexity of language as a cognitive process [Winograd 83], whereas commercial endeavors have demonstrated its usefulness. NL systems developed within the AI community have relied heavily on matching linguistic input into predefined knowledge representations [Schank 81, Schank 82] while linguists have focused on matching the input into a predefined linguistic representation [Winograd 72, Marcus 80, Gazdar 81]. Both approaches suffer from relying too heavily on a predetermined representation of knowledge (conceptual versus linguistic) without considering the use of pragmatic knowledge that makes understanding easy for humans.

The designer of NL programs is faced with two interface problems: the user interface and the NL/expert system interface. The user interface depends on the tasks (e.g., information storage and retrieval) and particular domains (e.g., medicine, chemistry) and is therefore clearly defined. The NL/expert system interface involves on the one hand a separation of linguistic knowledge proper from expert knowledge, while on the other hand requires the integration of these two types of knowledge. Schematically, the various knowledge aspects of the NL system can be represented as in figure 6–1.
Figure 6-1: The two interface components of a NL program
The tasks of a NL front end involve storing and retrieving information. The usefulness and "user-friendliness" of natural language interfaces with respect to information storage are illustrated in figure 6-2. The input of data into a system in the form of the system's programming language requires, even from the casual user, an understanding of the data structures used. Another drawback is the error rate both in terms of physically storing the information as well as preformatting it. For reasons of efficiency and accuracy NL input is to be preferred.

\[
\text{(AT ADMISSION)}
\]
\[
\text{JAUNDICE}
\]
\[
\text{(PRURITUS PERSISTENT)}
\]
\[
\text{VOMIT}
\]
\[
\text{(WTLOSS (AMOUNT (LB 16.0)))}
\]
\[
\text{(PAIN F ABDOMEN)}
\]
\[
\text{E}
\]
\[
\text{((3 MON) BEFORE ADMISSION)}
\]
\[
\text{(WTLOSS (AMOUNT (KG 7)))}
\]
\[
\text{HEARTBURN}
\]
\[
\text{BELCHING}
\]
\[
\text{E}
\]
\[
*\text{EOF}*
\]

A 39-year-old man was admitted to the hospital because of jaundice. He was well until three months previously, when he began to have heartburn and frequent belching; he reduced his food intake and lost 7 kg in weight...

Pruritus occurred almost immediately after the operation, with anorexia, nausea and occasional vomiting.

Figure 6-2: Storage of medical information
Information storage and retrieval each requires a different strategy of NLP since the linguistic input is structurally different in both cases. For storing the information contained in a piece of text, the program can rely on explicit, structural (i.e., syntactic) information. For retrieving the information, the system faces two problems: ungrammatical input and the unknown intentions of the user ("stonewalling"). Since the retrieval of information relies on how the information is stored, the configuration of the stored knowledge and how this configuration is constructed is considered to be an integral part of the retrieval process.

When humans retrieve information they ordinarily want fast and easy access without going through the formalities of proper syntax (e.g., Employees on vacation versus List all employees that are on vacation). Research in the area of sublanguages [Kittredge 82] has shown that domain-specific discourse uses a restricted grammar and has fewer ambiguities. In the medical context, the fragment no liver refers to a test which means no liver felt. However, as has been pointed out in [Hirschman 83], syntactic processing is necessary even for a limited domain, especially to determine modifier-host relations, as the differences in the following example shows: Had fever 1 day prior to admission versus Had 1 day fever prior to admission.

By the same token, users do not want to explain to the system at great length what their intentions are for retrieving a particular type of information. A typical query could be: Who makes more than $50,000 a year. Depending on how the data base is set up possible answers could be: All vice-presidents or John Smith, Paul Morris. This example shows that the power of the NL retrieval system does not lie exclusively in how many complex syntactic constructions it can handle but also in how it can effectively mediate between the user and the data base. The usefulness of a NL front end depends on it being able to handle a large variety of queries, and to accommodate the user who is not familiar with the information potential of the data base.

After these functional aspects are dealt with, structural aspects that have to be considered pertain to the systems architecture [Obermeier 84a, Obermeier 84e]:
modularity, extensibility, transportability, robustness. (1) Modularity: the enormous amount of information stored in the program requires modular systems to maintain, debug, and extend it. (2) Extensibility: after a system's pilot stage, the possibility to upgrade depends on a sound underlying theoretical orientation. (3) Transportability: the high costs of developing a NL system and the often needed access to multiple software facilities make this criterion important. (4) Robustness: because of the problems that stem from human/machine interaction (e.g., faulty or incomplete input), the system has to be able to recover from deficient input.

The feasibility of a NL system certainly cannot be determined for any data base per se unless we had a complete classification of all databases as well as a classification of NLPSs. For expert systems, however, NL interfaces (1) increase the usability for (casual) users who do not want to learn the stringent and limiting format of a query language; (2) increase productivity by providing a tool for optimally accessing and manipulating information in natural language; (3) allow the user to make the queries more precise, since he is not limited to a few commands or to cumbersome search mnemonics; (4) enable all users on different levels of authorization to access the information directly without going through intermediaries.

For the person maintaining the NLPS, the usefulness of a system depends on how well it is structured; for the user, on how well it functions. Any information-retrieval system that is interactive and relies heavily on query commands would become more efficient and cost-effective by using a NL front end. Although formatted retrieval methods (e.g., use of system's programming language, menu, special query language) have the advantage of being precise, unambiguous, and powerful; they achieve these virtues only at a very low level of abstraction. Without the capacity to access information by means of NL, the user either has to understand how the knowledge has been stored and what data structures are available, or has to be content with the inflexibility of the query structure provided.

A NL front end consists of a computational and a conceptual component. The former
has to deal with bookkeeping and scheduling (e.g., search and rule invocation), the latter with grammatical formalisms and knowledge representation. NL processing programs are characterized by the extent of their linguistic and conceptual coverage, and the relative primacy of the different processing levels found in each system. An understanding of the conceptual differences between the systems is best achieved by looking at (a) whether there is a separation within the system into modules (e.g., linguistic, conceptual, domain-specific components), and (b) how the modular systems differ in their control structure. In figure 6-3 an overview of modular versus nonmodular systems is given. This distinction is equivalent to special-purpose versus general-purpose systems as found in [Moore 81]. Since NL interfaces are domain specific and thus, in a sense, special purpose systems, I prefer to use a more structurally oriented terminological distinction of extant systems.

**NONMODULAR SYSTEMS**

<table>
<thead>
<tr>
<th>LUNAR  [Woods 72]</th>
</tr>
</thead>
<tbody>
<tr>
<td>LADDER [Hendrix 78]</td>
</tr>
<tr>
<td>PLANES [Waltz 78]</td>
</tr>
</tbody>
</table>

**MODULAR SYSTEMS**

<table>
<thead>
<tr>
<th>DIALOGIC [Grosz 82]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHLIQA 1 [Bronnenberg 80]</td>
</tr>
<tr>
<td>TQA [Damerau 80]</td>
</tr>
<tr>
<td>INTELLECT [Harris 77]</td>
</tr>
<tr>
<td>SAM [Schank 75]</td>
</tr>
<tr>
<td>TDUS [Robinson 80]</td>
</tr>
<tr>
<td>ASK [Thompson 83]</td>
</tr>
<tr>
<td>EXPLORER [Lehnert 82]</td>
</tr>
<tr>
<td>SHRDLU [Winograd 72]</td>
</tr>
</tbody>
</table>

Figure 6-3: Modular versus Nonmodular Systems
In general, the NL front end consists of three types of knowledge: "linguistic knowledge, knowledge of the domain of discourse, and knowledge of the organizational structure of the database" [Boguraev 82]. The nonmodular systems are characterized by either merging discourse with database knowledge (e.g., LUNAR [Woods 72]), or merging linguistic and domain knowledge in their grammars (e.g., LADDER [Hendrix 78], PLANES [Waltz 78]). The modular systems are characterized by separating different levels of processing. The ways in which the flow of control between the levels is set up and the primary focus of the parsing is defined, lead to a theoretical and practical distinction between NL processing strategies.

Most modular systems that are based on a generic syntactic analyzer use domain-dependent semantics (e.g., TQA [Damerau 80], PHLIQA 1 [Bronnenberg 80]). In TDUS [Robinson 80] as well as in a system described in [Boguraev 82], the transition between syntactic and semantic interpretation is accomplished by invoking a translator after the sentence is syntactically analyzed. A variation on this method is found in DIALOGIC [Grosz 82] where domain-independent semantic operators are triggered during the syntactic parsing.

6.2. THE TASKS OF GROK

A medical assistant [MA] receives "raw" data from the patient. He then extracts from this data information what will be relevant for the MD's diagnosis. The MA draws lower level conclusions in the form of trends based on numerical values or judgmental reasoning (e.g., if the patient has symptoms of dyspepsia, only this term will be used by the assistant and not the single symptoms). In the anamnesis of the patient, the MA has already included the previously established syndromes or diagnoses in addition to the patient subjective symptoms.

The process of extracting information from the patient is parallel to the clustering of

83 The term "discourse knowledge" refers to the pragmatic information stored in the knowledge base; whereas database knowledge refers to the content of the database.
information into topics by the MA; e.g., if the patient is an alcoholic, the MA very likely mentions SGOT values in his description of the patient. The pre-processing of information in the form of clustering by the MA allows the MD to see connections from the immediate context.

6.3. PATREC

The existing program for handling patient records (PATREC) is atomistic in its knowledge representation and management, especially with respect to temporal information. It uses the notion of "event" as the temporal characteristic of an object. Events are organized into episodes by key-events [Mittal 82]. The key event is ordinarily admission. All other events are organized around admission. PATREC tries to order events in a time graph after all other events have been input. Two major differences between PATREC and GROK are discussed below. In PATREC, a definition hierarchy stores the input according to the relations between various events. From the definition hierarchy, PATREC creates a time graph, and finally applies simple inference rules to order the events. GROK tries to model the on-line process of establishing a temporally valid knowledge representation. It does not go through various stages of processing, building up structures which might have to be changed.

An important difference is the inference procedures. While PATREC orders all relations before it knows what questions are being asked, it does not distinguish between on-line processing and answering queries. During the on-line phase it stores the information in a definition hierarchy without attempting to construct a viable knowledge representation. This feature of PATREC may be a result of using stylized input which lends itself to the neglect of actual on-line processes.

---

84 A third difference is the way the data is input; in PATREC a stylized format is used that consists of an event cluster and its corresponding temporal description; in GROK, English sentences can be used to input data.
6.4. DISCOURSE STRUCTURE OF THE CASE DESCRIPTIONS

The discourse structure found in the case descriptions of the *Journal of the American Medical Association* is rigid; certain themes and topics form a pattern that the program can use. The initial topic consists of the general information about the patient (e.g., sex, profession, age). The second topic deals with symptoms noted by the patient. Symptoms are subjective observations of disorders by the patient. The third topic involves the signs noted by the doctor. Signs are objective observations of the patient's behavior made by the physician. The fourth topic comprises the history of the patient (e.g., previous illnesses, family history). The last topic includes an extensive discussion of laboratory data and results from examinations.
CHAPTER 7
THE PROGRAM

GROK is a functioning program which is implemented in ELISP and EFRL on a DEC20/60. A brief description of the environment, and the code itself is provided in appendixes A and B. A longer trace of the program with self-explanatory comments is reproduced in appendix C. The design and implementation of GROK is discussed in 2.1. and 2.2., respectively. In this chapter, I step through one sample sentence from my corpus in figure 1-1 to illustrate the working of my NLPS.

The program performs the following tasks during analysis of the text:

- access the lexicon
- analyze the words morphologically
- invoke the specialists
- assign a partial phrase structure to the sentence
- find and instantiate the relevant frames
- monitor the interaction between the various knowledge sources
- store the results of the text analysis in the form of events in the knowledge base.

In figure 7-1, a trace of the lexical access processing is given. The lexical procedure assigns syntactic, functional, and semantic features to the words. At the same time, it analyzes the morphological characteristics of the words.

The output of the lexical module is analyzed by the bottom-up parser in terms of
the specialists. Each phrase specialist uses its syntactic knowledge which consists of the phrase structure rules of the X-bar grammar, and its factual knowledge which depends on the incoming words.

The NLPS encounters the word this and activates the NP-specialist. The adjective eighty-year-old has the feature "age" attached to it. The knowledge base knows that "age" is part of the frame which contains the information about the patient, stored in a patient frame. The slot for "age" in the patient frame is then filled, and so are the slots for "race" and "profession". The NP-specialist hands over control to the VP-specialist when it gets to the word complain.

The VP-specialist checks first to see if there are any particles following the verb. It

---

85 Specialists are discussed in 2.1. and 2.2.

86 Cf. 3.3. and 3.4.

87 Cf. 3.4.

88 Cf. 2.1.1.
finds of, and based on the information on the verb complain, GROK expects the domain-specific schema "report-symptom". The initial task of the VP-specialist consists in finding the schema for the current sentence. If the schema is found, the information contained in the schema provides defaults or expectations for the next step in the analysis. In my example in figure 7-2, the schema "report-symptom" contains the information that the subject of the sentence must be a patient; the object, a symptom. The schema also expects a default value for the key-event.89

The schema instantiates the event frame which will contain the symptom and the appropriate temporal indicators. One example of the final representation is given in figure 7-3. The schema indicates that the following NP which contains an abnormality will be a symptom, and has to be placed into the instantiated event frame. Therefore, the next complex NP will be put into the frame-slot "symptom" in the first event frame of this particular patient history.

The boundary specialists apply numerous checks to the elements being conjoined. The and-analysis considers the possibility of joining clauses or phrases. It first checks if there has been an event, a time-focus or a domain-concept established. It also checks if the NP specialist or the VP specialist has been activated immediately before the conjunction. It compares the word-class of the previous and the following words, and decides correctly that the and conjoins two parts of an NP.

After the sentence is processed, it is possible to check the phrase-structure assignments which were made during the processing. An annotated parse of the sentence at hand is given in figure 1-2.

During the analysis process, the program keeps track of the current thematic and temporal foci which are predefined by the particular domain (e.g., medicine), or their generality (e.g., time). The program stores the foci for all the sentences it encounters in

89
Cf 1.2.
This eighty-year-old Caucasian female complained of nausea vomit abdominal swelling jaundice.

Figure 7-2: Trace of Sentence [1] in Figure 1-1

(FASSERT EVNT1
(SYMPTOM (VALUE ((((((NIL NAUSEA)VOMIT)ABDOMINAL SWELL)JAUNDICE)
(KEY-EVENT (VALUE (ADMISSION))))
(DURATION (VALUE (ADMISSION))))
(CLASSIF (VALUE (INDIVIDUAL))))
(TYPE (VALUE (EVENT))))

Figure 7-3: Final Knowledge Representation of Event

Once a new event frame or domain-specific frame is instantiated, GROK can use the information associated with each event frame (e.g., duration, key-event), together with the information from the domain-specific frame (e.g. the pain frame contains slots for specifying the location, intensity, and type of pain) to interpret the text.
CHAPTER 6

CONTRIBUTIONS TO THE STUDY OF LANGUAGE

The single most important virtue of programming should not come from a finished program itself, or what it does, but rather from the business of developing it. Indeed, the aim should be neither to simulate human behavior—often a species of dissimulation—nor to exercise artificial intelligence, but to force the theorist to think again. [Johnson-Laird 80, 110]

In this chapter, I summarize the contributions of my dissertation to the study of language from theoretical and methodological points of view. In the last section of this chapter, I discuss the inherent limitations and possible extensions of my theory and program.

8.1. THEORETICAL CONTRIBUTIONS

The research for my dissertation included three major areas of cognitive science that concern themselves with language: linguistics, artificial intelligence, and—to the extent that my program is consistent with current psychological experiments and theories—psycholinguistics.

A major theoretical contribution by a NLPS is to introduce computability into a theory for NLP. Computability provides an important constraint for any theory which has led some of the AI research "to actually anticipate certain developments in linguistic theory and philosophy" [Brady 83, 36]. Besides computability, my research has contributed to the theory of parsing, to an understanding of general linguistic phenomena (e.g., anaphora), and to the representation and processing of temporal knowledge.
8.1.1. Parsing

Computational linguists consider parsing as a primarily structural process, whereas AI researchers emphasize the conceptual\textsuperscript{90} side of parsing [Schank 75, Rieger 79].\textsuperscript{91} Both groups, however, agree that structural, viz., syntactic information cannot be separated from conceptual, viz., semantic or pragmatic information when it comes to a NLPS which "understand" language. My parsing algorithm points to a solution of this impasse. It incorporates enough domain-specific and general knowledge so that the parsing procedure can be driven by conceptual information, while at the same time employing a descriptively adequate theory of syntactic processing.\textsuperscript{92} The theoretical significance of my parsing algorithm lies in its use of syntactic information. Syntactic information selects between contextually plausible interpretations of the text.

My parsing algorithm not only supports the prevalent theories of semantic parsing put forth in AI, but also provides a sound linguistic theory for the necessary syntactic information processing. My knowledge-based parsing algorithm furthermore suggests to the computational linguist a viable extension of the well-known parsing techniques: strictly top-down or bottom-up processing.

The state of the art in parsing in computational linguistics distinguishes two techniques, top-down and bottom-up processing.\textsuperscript{93} A strictly top-down parser tries to match the grammar rules against the input, starting at the topmost level and recursively moving toward lower levels of sentence structure until the individual word classes are combined.

\textsuperscript{90}The term "conceptual" refers to making use of concepts represented in the knowledge base, and not to Schankian "Conceptual Dependencies".

\textsuperscript{91}Cf. 2.3. and 3.5.1.

\textsuperscript{92}Cf 3.4.

\textsuperscript{93}For a more detailed discussions of the techniques and extant systems, see 3.5.
There are at least three disadvantages of top-down parsers: \(^{94}\) (1) if a particular rule cannot be matched, the parser has to backtrack; thus repeated analysis of the same constituent is possible;

(2) if garbled input is encountered and no rule is found, the parser is forced to stop; procedures to relax the grammar are then needed to finish parsing the sentence;

(3) if a multitude of parses are encountered, a separate module has to decide which is the best parse.

Bottom-up parsers which start to combine the lowest level elements (i.e., words) first, then build larger constituents, \(^ {95}\) also have severe limitations:

(4) they produce numerous spurious or useless parses, since the pattern matcher is not goal directed; and

(5) the correctness of the parse can only be assessed after all the parses are built.

In GROK, these five problems are avoided, based on my theory of pragmatic parsing: \(^ {96}\)

(1) My parsing technique does not include backtracking to find the correct parse because it uses limited look-ahead. Word-class ambiguities are often resolved, if a parsing technique allows for looking ahead to the end of a phrase [Marcus 80]. Global ambiguities or structural ambiguities can be resolved in my system because it

\(^{94}\) Cf. 3.5.2.1.

\(^{95}\) Cf. 3.5.2.2.

\(^{96}\) Cf 2.3.3.
uses extralinguistic knowledge to establish the individual phrase structure of a sentence.  

(2) If garbled input is encountered, my NLPS can still process the information based on the immediate interface to the knowledge base. Before the syntactic procedures try to establish a phrase, the NLP tries to find the conceptual information. Therefore, even if there were no syntactic rule for the parse, the NLPS would still be able to recover from the erroneous input.

(3) and (4) My NLPS is deterministic. It accesses the knowledge sources immediately. Since the control mechanism is based on local specialists, the NLP can determine after each processing step which of the options to pursue.  

(5) The support of the knowledge base for making decisions during the parsing process, assures that the resulting parse is compatible with the information stored in the knowledge base. A decision as to how to continue a particular parse is made during, and not after processing a sentence.

8.1.2. Linguistic Phenomena

The explanatory adequacy of my theory is best demonstrated by showing how the theory handles foremost problems in (computational) NLP (e.g., conjunctions, anaphora).

97 Cf. 2.1.

98 Cf. 2.1.2.

99 Cf. 3.3.
8.1.2.1. The Conjunction \textit{and} 

Conjunctions are still subject to intense investigation by computational linguists since no algorithm handles all of them.

It is my contention that for processing conjunctions, syntax is of much less importance than the semantic and pragmatic information found in the sentence. This observation is in line with recent publications on the complementary role of syntax and discourse interpretation for constructions like Gapping and VP-deletion [Zwicky 84].

Conjunctions separate events or objects. Conjoined phrases can be reduced by different types of ellipsis. Temporal information can be relevant in the following forms:

- Two conjoined events are related to one temporal reference (e.g., \textit{Tom went to church and Joe lost his keys on Friday}).

- Two conjoined events are related to two different temporal references (e.g., \textit{Tom went to church on Friday and Joe lost his keys on Saturday}).

- Two conjoined events are related to a temporal reference and an anaphoric reference (e.g., \textit{Tom went to church and Joe lost his keys a while later}).

Further complications arise when a conjunction is used twice, with a different function each time. Consider (24) from my sample text (see Figure 1-1):

\begin{quote}
(24) She had had ill-defined gastrointestinal complaints for many years and occasional episodes of nausea and vomiting three years previously.
\end{quote}

In this sentence, the first occurrence of \textit{and} indicates a separation of two events, the second occurrence a conjunction of NPs. In (25), another sentence within the same corpus, the conjunction \textit{and} does not separate two events, but the first event falls within the time reference of the previous sentence while another temporal reference is introduced after the conjunction.

\begin{quote}
(25) Her abdomen became prominent, and one week before admission jaundice was noticed.
\end{quote}
In (25), the temporal reference does not hold for the entire sentence but only for part of it. If a NP conjunction is used, however, the temporal reference at the end of the sentence is relevant for the entire sentence (e.g., (26)).

(26) The patient complained of jaundice and increasing weakness three months before admission.

Another complication is introduced if the sentence (e.g., (28)) does not have an explicit temporal reference but instead, the temporal reference from the previous (e.g., (27)) sentence has to be interpreted as the reference for the current sentence.

(27) Four weeks before admission she developed pain across the upper abdomen, radiating to the flanks.

(28) She also complained of shooting precordial pains and palpitation with slight exertional dyspnea.

Sentence (29) introduces the problem of mentioning a time reference which modifies the previous phrase. The phrase every second or third day does not refer to weakness or constipation, but modifies bowel movements. The occurrence of bowel movements is still within the time frame of the four weeks before admission.

(29) Increasing weakness and constipation developed, with bowel movements every second or third day.

If there are conjunctions in the sentence, the following algorithm applies at the time the conjunction is encountered:

- If there is (in the current sentence) a new temporal and thematic focus, link the thematic to the temporal focus, i.e., instantiate an event.
- If there is a new thematic focus, wait till the next boundary to find a possibly new temporal focus.
- If there is a new temporal focus, wait till the next boundary to find a possibly new thematic focus.
In this theory, the boundary markers are used as indicators for the NLPS to determine if thematic and temporal information must be linked. The boundary markers indicate that new information can be expected. This algorithm shows how a NLPS can detect that a sentence can have two different events, or that two sentences can refer to one event. A sentence which only contains a change in the time reference (e.g., Three more days passed)\textsuperscript{100} can also be processed by my model; in this case, only the temporal focus changes.

A further complication arises in (30) where two NPs containing one temporal reference occur:

(30) She complained of jaundice five weeks and three days before admission.

In this sentence, the importance of the NP specialist\textsuperscript{101} becomes apparent. If the NLPS followed the above algorithm, the sentence would be interpreted as having two distinct events. The NP specialist, however, stays in control and suspends the "and-specialist" until the adverbial before admission is encountered.\textsuperscript{102} In this example, the conjunction can be interpreted correctly only if the NLPS has some knowledge about temporal indicators and how they can be conjoined.\textsuperscript{103}

\textsuperscript{100} See [Hinrichs 81, 5] where this case is pointed out but not further discussed: "Weitere Faelle waeren zu beruecksichtigen, etwa die Rolle von Nominalphrasen in Subjektposition, die Zeitintervalle bezeichnen".

\textsuperscript{101} Cf. 2.1.2.

\textsuperscript{102} See the last example in Appendix C for a trace of a similar sentence.

\textsuperscript{103} GROK cannot disambiguate [21]. However, it constructs a plausible reading.
8.1.2.2. Anaphora

The treatment of anaphora in discourse theories has been a primary concern for analysts [Sidner 83]. In [Sidner 83], two possibilities are discussed for determining the referents of anaphora: (1) exhaustive search in the co-text to find the referent, or (2) a search guided by the foci found in the text. While in [Sidner 83], a heuristic for interpreting anaphoras is given, the author does not discuss the issue from the lowest level of processing. In my approach to parsing, the pronouns, and anaphoras in general, are specialists which trigger a verification procedure to determine their referents. If the parse finds a pronoun, it tries to establish a referent from the foci of the immediate context or the knowledge base. Consider the sentences in (31):

(31) Three days before admission, the patient had jaundice. Two weeks before that, nausea occurred.

The that-specialist knows from the NP specialist invoked by two weeks that that refers to an event. The previous event three days before admission becomes the reference event for the second sentence. The "that" specialist has to make a three-way decision if it is invoked since that can be a relative pronoun, a complementizer, or a deictic element. The treatment of function words and anaphoras as "specialists" with localized control structure allows the NLPS to interpret ambiguities without building up unnecessary structural connections during parsing.\textsuperscript{104}

8.1.2.3. Prepositional Phrase Attachment

In 3.5.3.2., I discuss the shortcomings of syntactic preference principles for parsing. My parsing algorithm suggests a viable solution to the problem of PP attachment. Consider the sentences

(32) I saw the man on the hill with the telescope.

\textsuperscript{104} Cf. 2.1.2.
(33) I saw the man at the corner with the cane.

(34) The patient complained of pain across the abdomen at admission.

All three sentences contain two PPs which could be attached, from a syntactic viewpoint, in three different ways. The invocation of syntactic preference principles clearly does not explain or describe the meanings of the sentences. The sentence in (32) is ambiguous only because we know about telescopes, hills, and active agents. The sentence in (33) is not ambiguous for the same reason. The sentence in (34) also requires special knowledge about the world and the particular domain of medicine.

The NLPS, without having access to domain and world knowledge, could not interpret any of these sentences correctly. What is needed, is a parsing algorithm which allows the pragmatic knowledge to guide the parsing procedure. Pragmatic knowledge contains, for instance, a concept of pain and the concomittant concept of the location of pain, its intensity, etc. If this knowledge is available to the parsing procedure, the parse is successful without oscillillating between possible interpretations.

The treatment of conjunctions and anaphoras in my NLPS demonstrates the advantages of my parsing algorithm for computational linguistics. My parsing theory, however, provides some evidence for general linguistics to consider higher level discourse structures when resolving ambiguities caused by conjunctions and anaphoras. Thus, part of my research can be used to support more psychologically oriented theories of text understanding as discussed in [Sag 84, Zwicky 84].

8.1.3. Text Processing

The other theoretical contribution of my dissertation to the study of language consists of a psycholinguistically plausible model of the processes and representation of the information contained in a text\(^{105}\). The modelling of the knowledge sources which go

\(^{105}\) I quoted Johnson-Laird [Johnson-Laird 81] who emphasizes that is is not the specific representation of a sentence that is interesting but "the fact that we possess procedures for constructing, manipulating and interrogating them".
into the comprehension process and their interaction led to a theory about how a reader can achieve "understanding" of a text in the medical domain.

For my theory, I interpret the notion of "focus" to include domain-specific concepts (e.g., sign and symptoms in the medical domain), as well as general concepts (e.g., time) whose relevance is dependent on the type of text in question. A NLPS can "comprehend" a well-structured, technical text\textsuperscript{106} if it can detect the relevant foci during text processing. Whereas the general foci may well be limited in number, the number of foci necessary to comprehend different texts, or even prose and poetry may be large. If this hypothesis could be borne out, a theory of text processing includes the study of foci and their relation to each other coupled with a syntactic processor which selects foci based on their placement in the sentence.\textsuperscript{107}

8.1.4. Temporal Information

Temporal information in a text is conveyed by explicit temporal indicators, implicit temporal relations based on what one knows about written texts (e.g., "time moves forward"), and "key events".\textsuperscript{108}

Temporal indicators in a sentence are not of equal importance. The tense marking on the verb has been the least influential for filling in the event structure. For the program, the most important information sources are adverbials.

The linear sequence of sentences also contributes to the set-up of the configurations of events. My program could make use of two generally known heuristics: time moves forward in a narrative if not explicitly stated otherwise; the temporal reference of the subordinate clause is ordinarily the same as that in the main clause.

\textsuperscript{106} Cf. 5.4.

\textsuperscript{107} Cf. 5.3. and 5.4.

\textsuperscript{108} Cf 2.3.4. and 4.4.
"Key events" are significant since they are used to relate events\(^{109}\) to one another. In my theory of text processing, key events build up the temporal structure of a text. If key events for other domains can be identified, they could be used to explain how a NLPS can "comprehend" the texts of the domain in question.

The representation of temporal information is significant in my theory. I define an event as the result of the assignment of a temporal value to a domain-specific concept. The structure of an event is generalizable to other domains. An event\(^{110}\) consists of a concept, a key event, a relation to key event, and a duration. In the medical domain, the instantiated event contains information about how long, and when a symptom or sign occurred, and what the key event of the instantiated event was.

Apart from the temporal issue, my research has shown that, if the domain and the task of the NLPS are sufficiently constrained, the use of frames as a knowledge representation scheme is efficient in implementing the system\(^{111}\). In my program, I have used individual frames to represent single concepts. These concepts later access the domain-specific knowledge representation which include the foci of a particular domain. Together with the temporal indicators, the information from the knowledge representation is then transferred to the topmost event frame. Procedures are then used to relate various event frames to each other. The restrictions and checks on the instantiation of the individual frames preclude an erroneous activation of a frame.

The viability of this approach shows that the idea of stereotypical representation of information is useful for NLPS if properly constrained. My program checks for the accessibility of the various levels of the knowledge representation whenever new information is coming in. This multilayer approach constrains the instantiation of the event frame sufficiently in order to prevent erroneous event instantiation.

\(^{109}\) Cf. 4.4.2.

\(^{110}\) Cf. 4.2.2.

\(^{111}\) Cf. 2.2.1.1.
8.2. METHODOLOGICAL CONTRIBUTIONS

The major methodological contribution a NLPS can make is the "explicit concern with partitioning explanations of linguistic behavior into representational and computational components" [Brady 83, 37]. In as much as such a partitioning in the system mirrors a natural "divisions of labor", we gain insight into the structure of the phenomenon \textit{in toto}, i.e., what the individual modules are in terms of their internal structure, what knowledge they contain, and their interacting mechanisms.

The interaction of syntactic, semantic, and pragmatic information in the comprehension process requires a control structure which invokes the knowledge sources at the appropriate point during the processing. The focus on control structure versus specific theories is a major difference in the methodology of AI and that of linguistics. The problem with specific theories for different levels of linguistic structure lies in combining them to obtain a cognitive theory of language understanding.\footnote{Therefore, the most promising methodology for studying cognition is to start with a balanced theory of the cognitive task to be explained. That task may be limited, and the special theories implemented in the experts may be simplified. But balance requires that the theory account for all the experts involved, and all the control structure relating them [Goodwin 82, 261].} The primary task of working toward a cognitive theory is to integrate available theories by means of a viable control structure. It follows from the emphasis on control structure that stronger and more complete special theories of syntax, case semantics, and the like may not be necessary for language understanding. (This is not at all the same thing as saying that humans do not embody stronger and more complete special theories for syntax than those available; nor that stronger theories would be useless; nor that linguists should not work on them.) More necessary may be additional theories of aspects of the problem as yet largely untouched, such as turn-taking or world knowledge about the behavior of liquids, together with a control structure that somehow knows which special theories to believe and how to interpret their results. [Goodwin 82, 261]

A common criticism of AI-based theories is their holistic approach "allowing no effective decomposition of the phenomenon at all" [Goodwin 82, 263]. In my system, the emphasis is on high modularity and a transparent control structure. Although my
system consists of autonomous modules, these modules interact during information processing.

In my dissertation, I follow the idea of integrating various linguistic theories by using an implementation of psychologically plausible processes as control structures. None of the linguistic theories I have used is novel. The contribution to the study of language lies primarily in the formation of a control structure that uses pragmatic information to access the linguistic knowledge sources which consist of restricted special theories (e.g., on the syntactic level, I used a form of X-bar syntax).

My approach combines the modularity found in linguistic theories with the holistic, psychological motivation of AI theories. My methodology produces a theory that is supported by a mental model of how temporal information can be processed. In addition, my theory is supported by a working program that makes it possible to test my contentions and to prove them wrong if processing new data turns out to be impossible.

AI research was said to be practical, but untheoretical, whereas research within linguistics was said to be theoretical, but impractical [Goodwin 82, 249]. In GROK, it has become apparent that neither has to be the case. My theory of temporal information processing profits from the practicality of a working, and fairly extensible program, while at the same time adhering to an adequate linguistic theory, and a powerful knowledge representation scheme.

8.3. STATE OF IMPLEMENTATION

GROK is a highly exploratory program. The limitations of the current implementations are in three areas: (1) the parser itself does not provide the capability of a chart parser since it will not give different structural interpretations of an ambiguous sentence of the type in (35):

(35) President Reagan swears in his new cabinet.
This type of structural ambiguity, where one constituent can belong to two or more different constructions, would not be detected. The shortcoming, however, could easily be removed by using a chart parsing algorithm that provides the possible structural interpretations of a sentence.

(2) the knowledge base does not have a fully implemented frame structure. Each generic frame has a certain number of slots that define the concept. A generic concept (e.g., sign) must have slots which contain possible attributes of the specific frame (e.g., where is the sign found; how severe is its manifestation). These slots have not yet been implemented. The number of frames is strictly limited to the temporal frames and a few exemplary generic frames necessary to process the text.

(3) the range of phenomena is limited. Only "before-admission" references are recognized by the system. Furthermore, slots that prevent the inheritance of events of limited durations are not yet in place.

Further research of temporal information processing must proceed in three areas.

In the area of psychology, a series of tests should be designed to test the claims made in my dissertation, especially with regard to the idea of an event structure as used in my dissertation.

In the area of linguistics, the theory should be extended to other temporal issues. In particular, tense and aspect should be incorporated in my theory. The verb and its function in representing and relating events requires further study. Different types of discourse have to be investigated. Intentions and how to discern them in a given text warrant intensive investigation. Structural ambiguities require an extension of the parsing module.

In the area of AI, different problem-solving mechanisms should be investigated. Extant systems could be used in comparative studies after GROK is extended to other domains. General evaluation criteria for the performance of extant systems and GROK have to be established.
In general, GROK is still in a developmental stage at which a number of phenomena have yet to be accounted for through an implementation.
CHAPTER 9

CONCLUSION

A race of hyperintelligent pan-dimensional beings once built themselves a gigantic supercomputer called Deep Thought to calculate once and for all the Answer to the Ultimate Question of Life, the Universe and Everything.

For seven and a half million years, Deep Thought computed and calculated, and in the end announced that the answer was in fact Forty-two — and so another, even bigger, computer had to be built to find out what the actual question was. [Adams 82, 1-2]

My dissertation aims for an integration of insights gained from linguistic, psychological, and Al-based research to provide a pragmatic theory and cognitive model of how temporal inferences can be explained within the framework of computational information processing. A pragmatic theory focuses on the information from the context (e.g., co-text, discourse situation, intentions of interlocutors) to explain linguistic behavior.

I have shown how an integration of linguistic and extralinguistic knowledge achieves a form of comprehension, where comprehension is characterized as a conversion of information based on knowledge from one representation into another. I have also shown how this approach leads to a parsing technique which avoids common pitfalls, and, at the same time, is consistent with results in psycholinguistic research. I have furthermore shown that such a procedural approach is a basis for an event-based theory for temporal information processing.

In particular, this dissertation shows the shortcomings of the orthodox rule-based approach to language processing which reduces words to tokens in a larger context while overemphasizing the role of the phrase and sentence level. It does this by providing a temporal knowledge representation and algorithms for processing pragmatic information
which are applicable to a wider range of phenomena than most of the notable computational NL theories within the field of AI [Schank 81, Rieger 79, Wilks 75], or linguistics [Marcus 80].

The research done for this dissertation lies within the computational paradigm [Winograd 83]. It focuses on the notion of process, i.e., on a step-by-step model of natural language processing [Schank 81]. The developed theory for NL use in my dissertation is demonstrated in a cognitive model. Cognition determines the behavior of an entity in a complex environment. Cognitive abilities allow the entity to recognize the underlying system of the complex environment and function according to the detected system. Language is a system for communication. Cognition determines the behavior of the language user. If a program functions consistently within this system of communication, it shows how cognitive abilities interact but not what they are. The activity of knowing entails for the program, as well as for the human, linking linguistic and conceptual structures in order to be able to communicate. The program and the human perform two activities at the same time: understanding the language and understanding based on language.

Understanding is primarily a linguistic process that is contingent on plausibility, representability, and computability of language. "Meaning and the constellation of mental attitudes that exhibit it, are manufactured products. The raw material is information", which is "an objective commodity, something whose generation, transmission, and reception do not require or in any way presuppose interpretive processes" [Dretske 80, vii]. We understand an utterance because we are familiar with the conventional meaning that is contained in the information.

Language is more than a manifestation of the reproducible understanding process; it is evidence for "creative" cognitive processes, viz., "thoughts". The understanding process requires a selection among alternatives of meaning. By the same token, thinking requires an organization of cognitive constructs. The difference between understanding and thinking is not qualitative, but rather quantitative. While understanding something, we
limit our selections of alternatives to the immediate context and its implications, whereas thinking requires recourse to all of our cognitive resources. These remarks are not meant to limit intelligent behavior to elaborate search strategies. They are meant to distinguish understanding from thinking while pointing to commonalities between these processes [Obermeier 84c].

Language is a manifestation of thinking; thinking is not mediated by language, rather it is translated into language. The understanding process relies on the linguistic transfer of "thoughts". The proper decoding and encoding of the meaning of a linguistic utterance requires a rule system for the translation process together with a representation scheme for the cognitive process. The relation between representation and cognition is explained in the following excerpt:

What makes it possible for humans (and other members of the natural kind informavores) to act on the basis of representations is that they instantiate such representations physically as cognitive codes and that their behavior is a causal consequence of operations carried out on these codes. Since this is precisely what computers do, my proposal amounts to a claim that cognition is a type of computation [Pylyshyn 84, xiii].
BIBLIOGRAPHY

[Adams 82] Adams, D.
The Restaurant at the End of the Universe.

[Allen 83] Allen, J.F.
Maintaining Knowledge about Temporal Intervals.
CACM 26, 1983.

[Anderson 76] Anderson, J.R.
Language, Memory and Thought.
Erlbaum, 1976.

[Barr 81] Barr, A. and Feigenbaum, E.
The Handbook of Artificial Intelligence, Volume I and II.

[Bartlett 32] Bartlett, F.C.
Remembering: a study in experimental and special psychology.
Cambridge University Press, 1932.

[Barwise 83] Barwise, J. and Perry, J.
Situations and Attitudes.

[Bennett 78] Bennett, M. and Partee, B.
Toward the Logic of Tense and Aspect.

[Bobrow 68] Bobrow, D.
Natural Language Input for a Computer Problem-solving System.
Semantic Information Processing.

[Boguraev 82] Boguraev, B.K. and Sparck Jones, J.
How to drive a database front end using general semantic information.
1982
University of Cambridge, Computer Laboratory, Technical Report No 32.
[Bonnet 79] Bonnet, A.
Understanding Medical Jargon as if it were Natural Language.

[Brady 83] Brady, M. and Berwick, R.C.
Computational Models of Discourse.

[Briscoe 83] Briscoe, E.J.
Determinism and its implementation in PARSIFAL.
Automatic Natural Language Processing.
Ellis Horwood, 1983.

[Bronnenberg 80]
Bronnenberg, W.J.H.J.
The question answering system PHLIQA 1.
In Bolc (editor), Natural language question answering systems.

[Brown 83] Brown, G., Yule, G.
Discourse Analysis.

[Bruce 75] Bruce, B.
Case Systems for Natural Language.
Artificial Intelligence 6, 1975.

[Carbonell 79] Carbonell, J.
Towards a self-extending Parser.
1979
Proceedings from the 17th Annual Meeting of the ACL, University of San Diego, La Jolla, Ca.

[Chandrasekaran 83]
Chandrasekaran, B. and Mittal, S.
Conceptual Representation of Medical Knowledge for Diagnosis by Computer: MDX and Associated Systems.
1983
Advances in Computer, Vol. 22.

[Charniak 75] Charniak, E.
Organization and inference in a frame-like system of common-sense knowledge.
Theoretical Issues in Natural Language Processing.
BBN, 1975.
[Charniak 78] Charniak, E.
With Spoon in Hand this Must be the Eating Frame.
In \textit{TINLAP-2}. 1978.

[Charniak 81] Charniak, E.
1981
Proc. of the Seventh IJCAI.

[Chomsky 63] Chomsky, N.
Formal properties of grammars.
\textit{Handbook of Mathematical Psychology, Vol. 2}.

[Church 80] Church, K.W.
On memory limitations in natural language processing.
1980
Report MIT/LCS/TR-245, Laboratory of Computer Science, MIT.

[Clark 77] Clark, H., Clark, E.
\textit{Psychology and Language}.

[Clocksin 81] Clocksin, W.F., Mellish, C.S.
\textit{Programming in Prolog}.
Springer-Verlag, 1981.

[Colby 74] Colby, K. et al.
Pattern-matching rules for the recognition of natural language dialog expressions.

[Collins 72] Collins, A.M., Quillian, H.R.
How to make a language user.
In Tulvin, E., Donaldson, W. (editors), \textit{Organization and Memory}.

[Cryer 77] Cryer, P.E. and Kissane, J.M.
Obstructive jaundice in a patient with polycystic disease.

[Dahl 81] Dahl, V.
Translating Spanish into Logic through Logic.
[Damerau 80] Damerau, F.J.
The transformational question answering (TQA) system: description, operating experience, and implications.
1980
Report RC8287, IBM.

[Davis 77] Davis, R., King, J.
An Overview of Production Systems.

[DeJong 74] DeJong, G.
Prediction and substantiation: a new approach to natural language processing.

[Dijk 72] Dijk, T.A. van.
*Some aspects of text grammars.*
Mouton, 1972.

[Dowty 79] Dowty, D.
*Word Meaning and Montague Grammar.*

[Dowty 82] Dowty, D.
The Effects of Aspectual Class on the Temporal Structure of Discourse: Semantics or Pragmatics?
1982
Paper presented at the OSU conference on the Semantics of Tense and Aspect in Discourse.

[Doyle 83] Doyle, J.
*Some Theories of Reasoned Assumptions. An essay in rational psychology.*

[Dresher 76] Dresher, B. and Hornstein, N.
On some supposed contributions of artificial intelligence to the scientific study of language.
*Cognition* 4, 1976.

[Dretske 80] Dretske, F.
*Knowledge and the Flow of Information.*
[Dry 82]  Dry, H.
The movement of narrative time.
1982
ms.

[Dyer 83]  Dyer, M.G.
In-Depth Understanding.

[Earley 70]  Earley, J.
An efficient context-free parsing algorithm.

[Fillmore 68]  Fillmore, C.
The Case for Case.
Universals in Linguistic Theory.

[Forster 79]  Forster, K.I.
Levels of processing and the structure of the language processor.
In Cooper, W.E., Walker, E.C. (editors), Sentence Processing:

[Frazier 79]  Frazier, L. and Fodor, J.
The Sausage Machine: a new two-stage parsing model.

[Gazdar 81]  Gazdar G.
Generalized Phrase Structure Grammar.
The Nature of Syntactic Relations.
Reidel, 1981.

[Gevarter 83]  Gevarter, W.B.
An overview of computer-based natural language processing.

[Gibson 68]  Gibson, J.J.
The Senses Considered as Perceptual Systems.

Artificial Intelligence and the study of language.
[Green 63] Green, B et al.  
BASEBALL: An automatic question answerer.  
*Computers and Thought.*  

[Grimes 75] Grimes J.E.  
The thread of discourse.  
Mouton, 1975.

[Grosz 81] Grosz, B.  
Focusing and description in natural language dialogues.  
*Elements of discourse understanding.*  

[Grosz 82] Grosz, B.  
DIALOGIC: a core natural-language processing system.  

[Harman 63] Harman, G.  
Generative Grammar without transformation rules. A defense of phrase structure.  

[Harris 77] Harris, L.  
User-oriented data base query with the Robot natural language query system.  

[Haviland 74] Haviland, S., Clark, H.H.  
What's new? Acquiring new information as a process in comprehension.  

[Hayes–Roth 79] Hayes–Roth, B. and Hayes–Roth, F.  
A cognitive model of planning.  

[Hendrix 78] Hendrix, D.G et al.  
Developing a natural language interface to complex data.  
*ACM Transaction on Database Systems,* 1978.

[Hinrichs 81] Hinrichs, E.  
Temporale Anaphora im Englischen.  
1981  
Staatsarbeit, Universitaet Tuebingen.
[Hirschman 83]
Hirschman, L. and Sager, N.
Automatic Information Formatting of a Medical Sublanguage.
In Kittredge (editor), Sublanguage. deGruyter, 1983.

[Hobbs 84]
Hobbs, J.
Building a large knowledge base for a natural language system.

[Hopcroft 69]
Hopcroft, J.E., Ullman, J.
Formal Languages and their Relations to Automata.

[Jackendoff 77]
Jackendoff, R.

[Johnson–Laird 80]
Johnson–Laird, P.N.
Mental models in cognitive science.

[Johnson–Laird 81]
Johnson–Laird, P.N.
Mental Models of Meaning.
In Joshi, A. Webber, B. Sag, I (editor), Elements of Discourse Understanding. Cambridge University Press, 1981.

[Kac 82]
Kac, M.B.

[Kahn 77]
Kahn, K. and Gorry, G.A.
Mechanizing Temporal Knowledge.
Artificial Intelligence 9, 1977.

[Kamp 79]
Kamp, H.
Events, Instants and Temporal Reference.

[Kamp 83]
Kamp, H. and Rohrer, C.
Tense in Texts.
In Baeuerle, R., Schwarze, C., Stechow, A., von (editors), Meaning, Use and Interpretation of Language. Walter de Gruyter, 1983.
[Karttunen 83] Karttunen, L.
KIMMO: A General Morphological Processor.
1983
Texas Linguistic Forum 22, Dept. of Linguistics, The University of Texas at Austin.

[Kay 67] Kay, M.
Experiments with a powerful parser.
1967
Proc. 2nd Int. COLING #10.

[Keenan 76] Keenan E.O., Schieffelin, B.
Topic as a discourse notion.
Subject and Topic.

[Kimball 73] Kimball, J.
Seven principles of surface structure parsing in natural language.

[Kintsch 74] Kintsch, W.
The Representation of Meaning in Memory.
Erlbaum, 1974.

[Kittredge 82] Kittredge, R.
Sublanguages.
AJCL 8:79–84, 1982.

[Lebowitz 83] Lebowitz, M.
Memory-Based Parsing.
Artificial Intelligence, 1983.

[Leech 69] Leech, G.N.
Towards a Semantic Description of English.

[Lehnert 82] Lehnert, W. and Shwarts, S.
EXPLORER: A Natural Language Processing System for Oil Exploration.
1982
Cognitive Systems Research Report No. 5.

[Marcus 80] Marcus, M.
Theory of Syntactic Recognition for Natural Language.
[Marslen-Wilson 75]
Marslen-Wilson, W.D.
Sentence perception as an interactive parallel process.

[Marslen-Wilson 78]
Marslen-Wilson, W. and Welsh, A.
Processing interactions and lexical access during word recognition in continuous speech.

[Marslen-Wilson 80]
Marslen-Wilson, W. and Tyler, L.
The temporal structure of spoken language understanding: the perception of sentences and words in sentences.
*Cognition* 8, 1980.

[McCarthy 69] McCarthy, J. and Hayes, P.J.
Some philosophical problems from the standpoint of artificial intelligence.
*Machine Intelligence* 4.

[McDermott 82]
McDermott, D.
A temporal Logic for Reasoning about Processes and Plans.

[Miller 84]
Miller, G.A.
Informavores.
*The Study of Information: Interdisciplinary Messages.*

[Milne 83]
Milne, R.
*Resolving lexical ambiguity in a deterministic parser.*

[Minsky 76]
Minsky, M.
A framework for Representing Knowledge.
*The Psychology of Computer Vision.*

[Mittal 81]
Mittal, S. and Chandrasekaran, B.
PATREC: A data base system that has domain–specific knowledge to answer questions.
1981
AI–working papers, CIS–Dept., OSU.
[Mittal 82] Mittal, S.
Event-based Organization of Temporal Databases.
1982
Proceedings from the 4th National Conference of the Canadian Society
for Computational Studies of Intelligence, Saskaton, Canada.

[Moore 81] Moore, R.C.
Practical Natural Language Processing by Computer.
1981
SRI Technical Note 251.

*Human Problem Solving.*

[Obermeier 84a]
Obermeier, K.
Natural Language Front-ends for Expert Systems.
New York.

[Obermeier 84b]
deHilster, D. and Obermeier, K.
Linguistic Processing for a Cybernetic Model.
In *Proceedings from the Conference on Intelligent Systems and
Rochester, Michigan.

[Obermeier 84c]
Obermeier, K.

[Obermeier 84d]
Obermeier, K.
*Natural Language Parsing for Database Generation. The State-
of-the-Art of Extant, Syntax-based Parsers With Broad
Grammatical Coverage.*
[Obermeier 84e]
Obermeier, K.
Arlington, Virginia.

[Pacak 80]
Pacak, M.G., Norton, L.M., Dunham, G.S.
Morphosemantic Analysis of -ITIS Forms in Medical Language.

[Pereira 80]
Pereira, F. and Warren, D.
Definite Clause Grammars for Language Analysis- A Survey of the Formalism and a Comparison with Augmented Transition Networks.
*Artificial Intelligence* 13, 1980.

[Pereira 83]
Pereira, F and Warren, D.
Parsing as Deduction.
1983
Proceedings of the 21st Annual Meeting of the ACL.

[Prior 67]
Prior, A.
*Past, Present and Future.*

[Pullum 83]
Pullum, G.
Context-freeness and the computer processing of human language.

[Putnam 75]
Putnam, H.
The Meaning of 'Meaning'.
University of Minnesota Press, 1975.

[Pylyshyn 84]
Pylyshyn, Z.W.

[Raphael 67]
Raphael, B.
SIR, a computer program for semantic information retrieval.
*Semantic Information Processing.*
[Reichenbach 47]
Reichenbach, H.
*Elements of Symbolic Logic*.

[Rieger 79]
Rieger, C. and Small, S.
Word Expert Parsing.

[Robinson 80]
Robinson, A.E. et al.
Interpreting Natural-Language Utterances in Dialogs about Tasks.
1980
SRI Technical Note 210.

[Roussopoulos 79]
Roussopoulos, N.
CSDL: A Conceptual Schema Definition Language for the Design of
Data Base Application.

[Russell 56]
Russell, B.
On Order in Time.
*Logic and Knowledge*.
Allen and Unwin, 1956.

[Sag 84]
Sag, I., Hankamer, J.
Toward a Theory of Anaphoric Processing.

[Sanford 81]
Sanford, A.J., Garrod, S.C.
*Understanding Written Language*.

[Schank 74]
Schank, R. and Rieger, Ch. III.
Inference and the Computer Understanding of Natural Language.

[Schank 75]
Schank, R.
*Conceptual Information Processing*.
North Holland, 1975.

[Schank 77]
Schank, R. and Abelson, R.
*Scripts, Plans, Goals, and Understanding*.
[Schank 81] Schank, R.C. and Riesbeck, C.K.
*Inside Computer Understanding. Five Programs Plus Miniatures.*

[Schank 82] Schank, R.C.
*Dynamic Memory. A Theory of Reminding and Learning in Computers and People.*

[Schubert 84] Schubert, L.K.
On Parsing Preferences.
In *COLING84.* 1984.

[Shapiro 83] Shapiro, E.Y.
The 5th generation project – a trip report.

[Shieber 83] Shieber, S.M.
Sentence Disambiguating by a Shift–Reduce Parsing Technique.

[Sidner 83] Sidner, C.L.
Comprehension of definite anaphora.
*Computational MOdels of Discourse.*

Comparative processes in semantic memory.

[Smith 82] Smith, B.
Language and computational semantics.
In *20th Annual Meeting of the ACL.* 1982.

[Sparck 83] Sparck Jones, K., Wilks, Y.
*Automatic Natural Language Parsing.*
Ellis Horwood, New York, 1983.

[Tanenhaus 84] Tanenhaus, M.K., Carlson, G., Seidenberg, M.S.
Do Listeners Compute Linguistic Representations?
[Tennant 81] Tennant, H.
*Natural Language Processing. An Introduction to an Emerging Technology.*
Petrocelli, 1981.

[Thompson 69] Thompson, F. et al.
REL: a rapidly extensible language system.
1969
Proc. of the 24th Nat. Conf. of the ACM.

[Thompson 81] Thompson, H.
Chart Parsing and rule schemata in PSG.
1981
19th Ann. ACL.

[Thompson 83] Thompson, B.H. and Thompson, F.B.
Introducing ASK, a Simple Knowledgeable System.
In *Conference on Applied Natural Language Processing.* ACL, 1983.

[Touretzky 84] Touretzky.
*The Mathematics of Inheritance Systems.*

[Vendler 67] Vendler, Z.
*Linguistics in Philosophy.*

[Verkuyl 72] Verkuyl, H.J.

[Waltz 78] Waltz, D.
An English language question answering system for a large relational database.

Artificial Intelligence: An Assessment of the State-of-the-Art and Recommendations for Future Directions.

*Psychology of Reasoning and Content.*
[Weizenbaum 66]
Weizenbaum, J.
ELIZA—A Computer Program for the study of Natural Language
Communication between man and machines.
CACM 9, 1966.

[Wilks 75]  Wilks, Y.
An intelligent analyzer and understander of English.
CACM 18, 1975.

[Wilks 83]  Wilks, Y., Sparck Jones, K.
Introduction: a little light history.
In Sparck Jones, K., Wilks, Y. (editors), Automatic Natural Language

[Winograd 72]  Winograd, T.
Understanding Natural Language.

[Winograd 77]  Winograd, T. and Bobrow, S.
An Overview on KRL: a Knowledge Representation Language.

[Winograd 83]  Winograd, T.
Language as a Cognitive Process.
Addison–Wesley, 1983.

[Woods 70]  Woods, W.
A Transition Network for natural language analysis.
CACM 13, 1970.

An experimental parsing system for transition network grammars.
Natural Language Processing.

An experimental parsing system for transition network grammars.
Natural Language Processing.

Procedural Semantics as a theory of meaning.
In Joshi, A., Webber, B., Sag, I. (editors), Elements of Discourse
What's important about knowledge representation?
*IEEE Computer*, 1983.

[Zadeh 84] Zadeh, L.
Coping with the imprecision of the real world.

[Zwicky 84] Zwicky, A.
On Appealing to Discourse Interpretation.
APPENDIX A

ELISP CODE FOR PARSING PROCEDURE

The code provided here includes all the functions necessary to access the lexicon, parse the sentence, and access the frames. It is listed in alphabetical order of the functions as they are in the actual file in my directory on the DEC20/60. Since GROK is embedded into a larger program in my directory, I have omitted the functions which are not used in GROK.

The last set of functions are the Convert-Word-to-Number functions which transform number words from the text into arabic numerbers.

The macroexpansions for the DO, LET, FOR functions are taken from [Schank 81], and are not listed here.

The ELISP functions

DEMORPHOLOGIZE
PHIL-GRAMMAR
MARK-FEATURES
CONVERT-WORD-TO-NUMBERS

are adapted from a program described in [Obermeier 84a], and were originally coded by David de Hilster. The other pieces of code are my own.

(DV *CURRENT-EVENT* NIL)
(DV *CURRENT-MED-TYPE* NIL)
(DV *CURRENT-TIME-FOCUS* NIL)
(DV *EVENT-STACK* NIL)
(DV *INFO-BUS* (NIL))
(DV *KEY-EVENT* NIL)
(DV *KEY-EVENT-FLAG* NIL)
(DV *MARKED-FEATURE-SENTENCE* NIL)
(DV *NODE-FLAG* NIL)
(DV *NODE-STACK* NIL)
(DV *NUMERIC-VALUE* 3.)
(DV *PARSE-TREE* NIL)
(DV *PARTICLE-FLAG* NIL)

(DE *PERIOD*-SPECIALIST (CONSTITUENT)
  (MSG T "period-specialist invoked")
  (POP *MARKED-FEATURE-SENTENCE*)
  (COND [[EQUAL *CURRENT-MED-TYPE* NIL] (SETQ *CURRENT-MED-TYPE* 'SIGN-SYMPTOM)])
  (COND [[AND [EQUAL *CURRENT-TIME-FOCUS* NIL]
               [EQUAL (CAR *EVENT-STACK*) NIL]]
         (SETQ *CURRENT-TIME-FOCUS* (CAR (FGET 'REPORT-SYMPTOM 'TIME 'SVALUE)))])
  (COND [[AND [NOT (MEMB *CURRENT-TIME-FOCUS* '(DAY WEEK MONTH YEAR))]
               [NOT (EQUAL *CURRENT-TIME-FOCUS* NIL)]]
         (SETQ *KEY-EVENT* *CURRENT-TIME-FOCUS*)]
  (COND [[EQUAL *CURRENT-EVENT* NIL]
         (//INSTANTIATE 'EVENT)
         (PUSH *EVENT-STACK* *CURRENT-EVENT*)]
  (COND [[AND [EQUAL *CURRENT-TIME-FOCUS* NIL]
               [NOT (EQUAL (CAR *EVENT-STACK*) NIL)]
         (SETQ *MEM*
              (FGET (CADR *EVENT-STACK*) 'RELATION-TO-KEY-EVENT 'SVALUE))
         [(NOT (EQUAL *REL-TO-KEY-EVENT* NIL))
          (FPUT *CURRENT-EVENT* 'RELATION-TO-KEY-EVENT 'SVALUE *NUMERIC-VALUE*)
          (FPUT *CURRENT-EVENT* 'RELATION-TO-KEY-EVENT 'SVALUE *REL-TO-KEY-EVENT*])]
  (FPUT *CURRENT-EVENT* 'KEY-EVENT 'SVALUE *KEY-EVENT*)]
  (COND [[EQUAL *CURRENT-TIME-FOCUS* NIL]
         (SETQ *TEM* (FGET (CADR *EVENT-STACK*) 'DURATION 'SVALUE))
         (FPUT *CURRENT-EVENT* 'DURATION 'SVALUE *TEM*)]
  [T (FPUT *CURRENT-EVENT* 'DURATION 'SVALUE *NUMERIC-VALUE*)
   (FPUT *CURRENT-EVENT* 'DURATION 'SVALUE *CURRENT-TIME-FOCUS*)]
  (FPUT *CURRENT-EVENT* *CURRENT-MED-TYPE* 'SVALUE *SIGNSYMPTOMS*)]
  (COND [[(NOT (EQUAL (FNAME? 'EXPECTATION) NIL)) (FERASE EXPECTATION)]
      (COND [[(NOT (EQUAL (FNAME? 'LOCATION) NIL)) (FERASE LOCATION)]
        (COND [[(NOT (EQUAL (FNAME? 'BODY-PART) NIL)) (FERASE BODY-PART)]
          (COND [[(NOT (EQUAL (FNAME? 'DIRECTION) NIL)) (FERASE DIRECTION)]])]
  (DV *REL-TO-KEY-EVENT* NIL)
(DV ADJ* (((ADJ) ADJ*) ((ADJ))))

(DE ADJ*-ANALYSIS (ADJ*-CHUNK)
  (PROG (TEMP TERM)
    (SETQ TEMP ADJ*-CHUNK)
    (FOR (PIECE IN TEMP)
      (DO (PROG (SEARCH-SPACE TOKEN TYPE)
        (COND [[EQUAL 'ADJ* (CAR PIECE)]
          (SETQ SEARCH-SPACE (CADR PIECE))]
        [T (SETQ SEARCH-SPACE PIECE)])
        (ACCESS-FRAMES SEARCH-SPACE))))))

(DE ADV-SPECIALIST (CONSTITUENT)
  (MSG "adv spec" T)
  (POP *MARKED-FEATURE-SENTENCE*)
  (PUSH *NODE-STACK* CONSTITUENT))

(DV ADV1 (((ADV) P1))))

(DV ADV2 (((NUM) N= (ADV))))

(DE ALL-IN-P (ITEM-LIST OBJECT-LIST)
  (COND [[NULL ITEM-LIST) NIL]
    [(NULL (CDR ITEM-LIST)) (INP (CAR ITEM-LIST) OBJECT-LIST)]
    [T (AND [INP (CAR ITEM-LIST) OBJECT-LIST]
      [ALL-IN-P (CDR ITEM-LIST) OBJECT-LIST)]))

(DE AMBIG-VN (CONSTITUENT)
  (MSG T "noun-verb ambiguity found - disambiguating..." T)
  (PROG (T1 T2)
    (SETQ T1 (REVERSE (MAPCAR 'CONSP CONSTITUENT)))
    NEXT-ITEM
    (COND [[EQUAL (CAR T1) NIL) (RETURN NIL)]
      [T (SETQ T2 (CAR T1))
        (SETQ T1 (CDR T1))
        (COND [[EQUAL (CAR T2) 'ED)
          (SETQ *CURRENT-CONSTITUENT*
            (CONS (CAR *CURRENT-CONSTITUENT*) '(ED V)))]
        [(AND [EQUAL (CAR T2) 'ING] [EQUAL *NODE-FLAG* 'NP])
          (ING CONSTITUENT)]
        [(EQUAL (CAR T2) 'ING)
          (SETQ *CURRENT-CONSTITUENT*
            (CONS (CAR *CURRENT-CONSTITUENT*) '(ING V))
            (ING *CURRENT-CONSTITUENT*)])])

(DE ANALYZE-PHRASE (PARSED-SENTENCE)
  (PROG (CURRENT-PHRASE PHRASE-STACK)
    (SETQ PHRASE-STACK PARSED-SENTENCE)
    NEXT-PHRASE
    (SETQ CURRENT-PHRASE (CAR PHRASE-STACK))
    (COND [[NULL CURRENT-PHRASE) (RETURN (MSG "DONE"))]
      [(EQUAL 'NP (CAR CURRENT-PHRASE)) (PRINT (CADR CURRENT-PHRASE))]
      [(EQUAL 'VP (CAR CURRENT-PHRASE)) (PRINT (CADR CURRENT-PHRASE))]
      [POP PHRASE-STACK]
      [GO NEXT-PHRASE]])

(DE AND-ANALYSIS (CONSTITUENT)
(MSG T "and-analyses started" T)
(PDP *MARKED-FEATURE-SENTENCE*)
(COND [(INTERSECTION (CAR *NODE-STACK*) '(DAY WEEK MONTH YEAR))
  (SETQ *TEMP-CONJ* (CAAR *NODE-STACK*))
  (MSG T "temp-con" *TEMP-CONJ*))]
(PUSH *NODE-STACK* CONSTITUENT)
(COND [(AND [NOT (EQUAL *CURRENT-TIME-FOCUS* NIL)]
  [NOT (EQUAL *SIGNSYMPTOMS* NIL)])
  (COND [(AND [NOT (MEMB *CURRENT-TIME-FOCUS* '(DAY WEEK MONTH YEAR))
    [NOT (EQUAL *CURRENT-TIME-FOCUS* NIL)])
    (SETQ *KEY-EVENT* *CURRENT-TIME-FOCUS*)])
  (COND [(EQUAL *CURRENT-EVENT* NIL)
    (/INSTANTIATE 'EVENT)
    (SETQ *CURRENT-EVENT* (CAR *FRAMES*))
    (PUSH *EVENT-STACK* *CURRENT-EVENT*)]]
  (COND [(NOT (EQUAL *REL-TO-KEY-EVENT* NIL))
    (FPUT *CURRENT-EVENT* 'RELATION-TO-KEY-EVENT
      'SVALUE
      *NUMERIC-VALUE*)
    (FPUT *CURRENT-EVENT* 'RELATION-TO-KEY-EVENT
      'SVALUE
      *CURRENT-TIME-FOCUS*)
    (FPUT *CURRENT-EVENT* 'RELATION-TO-KEY-EVENT
      'SVALUE
      *REL-TO-KEY-EVENT*)]
  (FPUT *CURRENT-EVENT* 'KEY-EVENT 'SVALUE *KEY-EVENT-FLAG*)
  (FPUT *CURRENT-EVENT* 'KEY-EVENT 'SVALUE *KEY-EVENT*)
  (FPUT *CURRENT-EVENT* 'DURATION 'SVALUE *NUMERIC-VALUE*)
  (FPUT *CURRENT-EVENT* 'DURATION 'SVALUE *CURRENT-TIME-FOCUS*)
  (FPUT *CURRENT-EVENT* 'CURRENT-MED-TYPE* 'SVALUE *SIGNSYMPTOMS*)
  (SETQ *SIGNSYMPTOMS* NIL)
  (SETQ *CURRENT-EVENT* NIL)
  (SETQ *CURRENT-TIME-FOCUS* NIL)])]
(DE ASSESS-PAIN NIL
  (/INSTANTIATE 'PAIN-FRAME)
  (SETQ PAIN-FEATURE (CAR *FRAMES*))
  (PUSH *SIGNSYMPTOMS* PAIN-FEATURE)
  (FPUT PAIN-FEATURE 'LOCATION 'SVALUE 'UNKNOWN)
  (FPUT PAIN-FEATURE 'BODY-PART 'SVALUE 'UNKNOWN)
  (FPUT PAIN-FEATURE 'DIRECTION '$IF-NEEDED '(POST-EXPECTATION))
  (SETQ EXPECTATION-TARGET 'DIRECTION)
  (FPUT PAIN-FEATURE 'INTENSITY '$SVALUE 'UNKNOWN)
  (COND [(INTERSECTION (CADR *NODE-STACK*) *FRAMES*)
    (MSG T "val"
      (SETQ VALU (CAR (INTERSECTION (CADR *NODE-STACK*) *FRAMES*))))
    (MSG T "fra" (SETQ FRA (CAR (FGET VALU 'SLOT-OF '$VALUE))))
    (FGET &DEL FRA VALU)
    (FPUT FRA VALU 'SVALUE (CAADR *NODE-STACK*)))]
(DE AVAILABLE-PATIENT-INFO NIL
  (MSG "THE FOLLOWING INFORMATION IS AVAILABLE ON THE CURRENT PATIENT" T)
  (MSG "SEX OF THE PATIENT IS : " (FGET 'GENERAL-INFO 'SEX '$VALUE') T)
(MSG "PROFESSION OF THE PATIENT IS : ")
  (FGET 'GENERAL-INFO 'PROFESSION 'SVALUE)
T)
(MSG "AGE OF THE PATIENT IS : " (FGET 'GENERAL-INFO 'AGE '$VALUE) T)
(MSG "RACE OF THE PATIENT IS : ")
  (FGET 'GENERAL-INFO 'RACE '$VALUE) T))

(DV BNP (DET QU NUM ADJ N))

(DE BOTTOM-UP (MARKED-FEATURE-SENTENCE)
  (PROG (T1 T2)
    NEXT-CONSTITUENT
      (COND [(EQUAL (CAR MARKED-FEATURE-SENTENCE) NIL) (RETURN)]
        [T (SETQ *CURRENT-CONSTITUENT* (CAR MARKED-FEATURE-SENTENCE))
          (BUILD-PHRASE (CAR MARKED-FEATURE-SENTENCE))
          (SETQ MARKED-FEATURE-SENTENCE (CDR MARKED-FEATURE-SENTENCE))
          (GO NEXT-CONSTITUENT)]))

(DE BP (SENTENCE)
  (SETQ *SIGNSYMPTOMS* NIL)
  (SETQ *NODE-STACK* NIL)
  (SETQ *NODE-FLAG* NIL)
  (SETQ *CURRENT-TIME-FOCUS* NIL)
  (SETQ *NUMERIC-VALUE* NIL)
  (SETQ *EXPECTED-VALUE* NIL)
  (SETQ *CURRENT-EVENT* NIL)
  (SETQ *TEMP-COND* NIL)
  (BOTTOM-UP (SETQ *MARKED-FEATURE-SENTENCE* (MARK-FEATURES SENTENCE))))

(DE BUILD-PHRASE (CONSTITUENT)
  (COND [(INTERSECTION CONSTITUENT BNP) (NP-SPECIALIST CONSTITUENT)]
    [(INTERSECTION CONSTITUENT BVP) (VP-SPECIALIST CONSTITUENT)]
    [(MEMB 'VN CONSTITUENT) (AMBIG-VN CONSTITUENT)]
    [(MEMB 'PRO CONSTITUENT) (PRO-SPECIALIST CONSTITUENT)]
    [(MEMB 'PREP CONSTITUENT) (PREP-SPECIALIST CONSTITUENT)]
    [(MEMB 'ADV CONSTITUENT) (ADV-SPECIALIST CONSTITUENT)]
    [(MEMB 'TO CONSTITUENT) (TO-SPECIALIST CONSTITUENT)]
    [(MEMB 'THAT CONSTITUENT) (MSG "boundary")]
    [(MEMB 'AND CONSTITUENT) (AND-ANALYSIS CONSTITUENT)]
    [(MEMB '*COMMA* CONSTITUENT)
      (PUSH *NODE-STACK* CONSTITUENT)
      (POP *MARKED-FEATURE-SENTENCE*)]
    [(MEMB '*PERIOD* CONSTITUENT) (*PERIOD*-SPECIALIST CONSTITUENT)]
    [T (MSG "end of sentence")]))

(DV BVP (V))

(DE CHANGE-FEATURES (CHANGE-LIST FEATURES)
  (PROG NIL
    NEXT-CHANGE
      (COND [(NULL CHANGE-LIST) (RETURN FEATURES)]
        [(EQUAL (CAAR CHANGE-LIST) '+)
          (SETQ FEATURES (UNION FEATURES (CDAR CHANGE-LIST))))
        [(EQUAL (CAAR CHANGE-LIST) '-)
          (SETQ FEATURES
(DE CHANGE-FEATURES-PART-TWO (LIST-1 LIST-2)
  (COND [(OR (NULL LIST-1) (NULL LIST-2)) NIL]
         [(NULL (CDR LIST-2)) (REMOVE (CAR LIST-2) LIST-1)]
         [T (CHANGE-FEATURES-PART-TWO (REMOVE (CAR LIST-2) LIST-1) (CDR LIST-2))]))

(DE CHANGE-I-TO-Y (WORD)
  (COND [(NULL (GETHASH WORD STEM-HASH-TABLE))
          (READLIST (REVERSE (CONS 'Y (CDR (REVERSE (EXPLODE WORD))))))]
          [T WORD]))

(DE CHECK-FOR-SIGN-SYMPTOM NIL
  (FGET-NEED-NEW 'REPORT-SYMPTOM 'CLASSIF '$VALUE))

(DE CONSTITUENT-ANALYSIS (PHRASE)
  (FOR (CONSTITUENT IN PHRASE)
    (DO (MSG T "CURRENT CONSTITUENT IS " CONSTITUENT T)
        (LET (CONSTITUENT (FOCUS CONSTITUENT))))))

(DV CONSTITUENT-LIST (ADV1 ADV2 N1 N2 V2 V1))

(DM CREATE-HASH-TABLE (BODY)
  (LIST 'SETQ (CADR BODY) (LIST 'MAKE-EQUALT-HASH-TABLE)))

(DE DEMORPHOLOGIZE (WORD LAST-ENDING-TRIED)
  (PROG (ENDING STEM)
            (SETQ ENDINGS '(ED D ING ES S /'S EN N ITIS LY))
         NEXT-UNTRIED-ENDING
            (COND [(NULL LAST-ENDING-TRIED) (GO NEXT-ENDING)]
                  [(NOT (EQUAL (CAR ENDINGS) LAST-ENDING-TRIED))
                   (POP ENDINGS)
                   (GO NEXT-UNTRIED-ENDING)]
                  [POP ENDINGS]
         NEXT-ENDING
            (SETQ STEM (GET-STEM-FROM-ENDING (CAR ENDINGS) WORD))
         (COND [(NOT (NULL STEM)) (RETURN (LIST STEM (CAR ENDINGS)))]
               [(NOT (NULL ENDINGS)) (POP ENDINGS) (GO NEXT-ENDING)]
               [T (RETURN (LIST WORD))])))

(DE DURATION-PREP-ANALYSIS (PREP)
  (SETQ SPEC-OF-DURATION (CADR PREP))
  (MSG T "SPEC" T SPEC-OF-DURATION T)
  (PROG (TEMP DURATION) (PUSH *TIME-PREP-STACK# 'DURATION)))

(DE ED (CONSTITUENT) (MSG T " ed-speelal1st activated " T))

(DE ERRORS NIL
  (MSG T "THE FOLLOWING ERROR MESSAGES ARE POSSIBLE" T T T
        **** WORD NOT IN DICTIONARY ***** " T T
        **** NO PSG-RULE FOUND *********** " T T))

(DE FILL-HASH-TABLE (HASH-LIST DICTIONARY)
  (PROG NIL)
NEXT-PUTHASH
  (COND 
    [(NULL HASH-LIST) (RETURN 'OK)])
  (PUTHASH (CAAR HASH-LIST) (CDAR HASH-LIST) DICTIONARY)
  (POP HASH-LIST)
  (GO NEXT-PUTHASH))

(DE FIND-PARTICLE NIL
  (PROG (T1 T2)
    (COND
      [(EQUAL (INTERSECTION (FSLOTS (CAR (INTERSECTION *FRAME* *CURRENT-CONSTITUENT*)))
               (CADR *MARKED-FEATURE-SENTENCE*)) NIL)
       (RETURN)]
      [T (FGET-NEED-NEW
           (CAR (FGET (CAR (INTERSECTION *FRAME* *CURRENT-CONSTITUENT*))
                     (CAR (INTERSECTION (FSLOTS (CAR (INTERSECTION *FRAME* *CURRENT-CONSTITUENT*))
                               *MARKED-FEATURE-SENTENCE*))
                     'VALUE))
           'CLASSIF
           'VALUE)
           [SETQ *PARTICLE-FLAG* 'YES]]
      [POP *MARKED-FEATURE-SENTENCE*]
      [POP *MARKED-FEATURE-SENTENCE*])
    )
  (RETURN))

(DE FIND-SYMPTOM NIL
  (MSG T " symptom-frame invoked: looking for symptom" T)
  (SETQ *CURRENT-MED-TYPE* 'SYMPTOM))

(DE FOCUS (CONSTITUENT)
  (PROG (THEME-FOCUS NEW-THEME-FOCUS OLD-THEME-FOCUS TIME-FOCUS OLD-TIME-FOCUS)
    (SETQ THEME-FOCUS (CAR *THEME-FOCUS-STACK*))
    (COND 
      [(EQUAL THEME-FOCUS (CDAR CONSTITUENT)) (MSG T "SAME-THEME-FOCUS")
       [(MEMBER THEME-FOCUS CONSTITUENT)
        (SETQ THEME-FOCUS (CAR (MEMBER THEME-FOCUS CONSTITUENT)))]
      [(PUSH THEME-FOCUS 'THEME-FOCUS-STACK*)
       (MSG T "CURRENT-THEME-FOCUS" T (CAR 'THEME-FOCUS-STACK*)]
      [SETQ TIME-FOCUS (CAR 'TIME-FOCUS-STACK*)]
      (COND [(EQUAL TIME-FOCUS (CDAR CONSTITUENT)) (MSG T "SAME-TIME-FOCUS")
               [(MEMBER TIME-FOCUS CONSTITUENT)
                (MSG T "NEW-TIME-FOCUS" T
                (SETQ NEW-TIME-FOCUS (CAR (MEMBER TIME-FOCUS CONSTITUENT))))]
      (RETURN CONSTITUENT))]

(DE FOCUS-MONTH NIL
  (MSG T " current time focus : month" T))

(DE FOCUS-WEEK NIL (MSG T " current time focus : week" T))

(DE FOCUS-YEAR NIL (MSG T " current time focus : year" T))
(DE FORMAT (SENTENCE)
 (PROG (LOW-LEVEL-SENT FEAT-SENT SAVE-FEAT-SENT CONST-LIST PHIL-INFO)
   (SETQ CONST-LIST CONSTITUENT-LIST)
   (SETQ FEAT-SENT (MARK-FEATURES SENTENCE))
   (SETQ SAVE-FEAT-SENT FEAT-SENT)
 NEXT-PIECE-OF-SENTENCE
   (COND [[(NULL FEAT-SENT) (RETURN (REVERSE LOW-LEVEL-SENT))]
    [(ALL-IN-P (CDAR FEAT-SENT) GAP-LIST)
     (PUSH LOW-LEVEL-SENT (CAR FEAT-SENT))
     (POP FEAT-SENT)
     (GO NEXT-PIECE-OF-SENTENCE)])
 NEXT-CONSTITUENT
   (SETQ PHIL-INFO (PHIL-GRAMMAR (EVAL (CAR CONST-LIST)) FEAT-SENT))
   (COND [[(NULL CONST-LIST) (GO NEXT-PIECE)]
    [(NULL PHIL-INFO) (POP CONST-LIST) (GO NEXT-CONSTITUENT)]
    (PUSH LOW-LEVEL-SENT (LIST (CAR CONST-LIST) PHIL-INFO))
    (SETQ FEAT-SENT *INFO-BUS*)
    (SETQ CONST-LIST CONSTITUENT-LIST)
    (GO NEXT-PIECE-OF-SENTENCE)])
 NEXT-PIECE
   (MSG T "****** NO PSG-RULE FOUND *******")

(DV FUNCTION-HASH-LIST ((AFTER PREP) (ALSO ADV) (AND BOUNDARY) (*COMMA= BOUNDARY) (*PERIOD= BOUNDARY) (BUT BOUNDARY) (BEFORE PREP) (FOR PREP) (HER PRO) (IN PREP) (NO NEGATION) (NOT NEGATION) (OR BOUNDARY) (OF PREP) (SHE PRO) (THE DET) (THIS DET) (WITH PREP) (ACROSS LOCATION PREP) (TO OPERATOR)))

(DV GAP-LIST (BOUNDARY PRO OPERATOR PREP))

(DE GET-STEM-FROM-ENDING (ENDING WORD)
 (PROG (ENDING-LIST WORD-LIST)
   (SETQ ENDING-LIST (REVERSE (EXPLODE ENDING)))
   (SETQ WORD-LIST (REVERSE (EXPLODE WORD))))
 NEXT-LETTER
   (COND [[(NULL ENDING-LIST) (RETURN (READLIST (REVERSE WORD-LIST)))]
    [(NOT (EQUAL (CAR ENDING-LIST) (CAR WORD-LIST))) (RETURN NIL)]
    (POP ENDING-LIST)
    (POP WORD-LIST)
    (GO NEXT-LETTER)])

(DE GETHASH-P (KEY HASH-TABLE)
 (SETQ «INFO-BUS» (GETHASH KEY HASH-TABLE))
 (COND [[(NULL «INFO-BUS») NIL] [T T]])

(DEF GROK NIL
 (MSG T "************ ************ ************  ***  ***  ***
   "************ ************  ************  ***  ***  ***
   "***  ***  ***  ***  ***  ***  ***  ***  ***  ***
   "***  ***  ***  ***  ***  ***  ***  ***  ***  ***
   "***  ***  ***  ***  ***  ***  ***  ***  ***  ***
   "***  ***  ***  ***  ***  ***  ***  ***  ***  ***
   "**  ***  ***  ***  ***  ***  ***  ***  ***  ***
   "************  ************  ************  ***  ***
   "************  ************  ************  ***  ***")

 (MSG T "HELLO AND WELCOME TO GROK" T "THE" T "GRAMMATICAL" T)
"REPRESENTATION OF" T "OBJECTIVE" T "KNOWLEDGE" T "SYSTEM" T T
(MSG "THE FOLLOWING COMMANDS ARE IMPLEMENTED AND READY TO USE" T T
"ERRORS: GIVES YOU A LIST OF POSSIBLE ERRORS MESSAGES" T T
"MARK-FEATURES: GIVES YOU THE LEXICAL AND MORPHOLOGICALLY
ANALYZED FORM OF THE INPUT" T T T)

(DE IN-P (ITEM GIVEN-LIST)
(COND [[(NULL ITEM) NIL] [(INP ITEM GIVEN-LIST) T] [T NIL]])

(DE ING (CONSTITUENT)
(COND [[(MEMB 'ADJ (CAR *NODE-STACK*))
            (SETQ SEMANTIC-FEATURE
                  (CONS 'G (CONS 'N (CDR *CURRENT-CONSTITUENT*))))]
            (SETQ *CURRENT-CONSTITUENT*
                  (CONS 'N (CDR *CURRENT-CONSTITUENT*)) SEMANTIC-FEATURE))
            (MSG T "Ing specialist activated:" *CURRENT-CONSTITUENT* T)
            (NP-SPECIALIST (REVERSE (CDR (REVERSE *CURRENT-CONSTITUENT*))))]
         [AND [MEMB '*COMMA* (CAR *NODE-STACK*)]
               [MEMB '*CCMMA* (CADR *MARKED-FEATURE-SENTENCE*)]
               [INTERSECTION (CADR *NODE-STACK*) BNP]]
            (SETQ SEMANTIC-FEATURE (CONS 'N (CDR *CURRENT-CONSTITUENT*)))
            (SETQ *CURRENT-CONSTITUENT*
                  (CONS (CAR *CURRENT-CONSTITUENT*) SEMANTIC-FEATURE))
            (MSG T "Ing specialist activated:" *CURRENT-CONSTITUENT* T)
            (NP-SPECIALIST (REVERSE (CDR (REVERSE *CURRENT-CONSTITUENT*))))]
         [T (NP-SPECIALIST (REVERSE (CDR (REVERSE CONSTITUENT))))])

(DE LINGUISTIC-SPECIALISTS (PARSED)
(COND [[(NULL PARSED) (MSG "DONE")]
            [(EQUAL 'N1 (CAR PARSED))
               (N1-SPECIALIST (CADR PARSED))
               (PUSH "PHRASE-LABEL-STACK* 'N1")
            [T (FUNCTION-WORD-OR-MARKER-SPECIALIST PARSED)]]

(DE LIST-NIL (ITEM) (COND [[(NULL ITEM) NIL] [T (LIST ITEM)]]))

(DE LOOK-FOR-BODY-PART NIL
(//INSTANTIATE 'SWELL-FRAME)
(MSG T "looking for body-part ..." T)
(SETQ SWELL-FEATURE (CAR *FRAMES*))
(SETQ *SIGNSYMPTOMS* (LIST *SIGNSYMPTOMS* SWELL-FEATURE))
(FPUT SWELL-FEATURE 'BODY-PART 'SVALUE 'UNKNOWN)
(COND [(INTERSECTION (CAR (NODE-STACK)) *FRAMES*)
  (SETQ VALU (CAR (INTERSECTION (CAR (NODE-STACK)) *FRAMES*)))
  (SETQ FRA (CAR (FGET VALU 'SLOT-OF 'SVALUE)))
  (FGETADEL FRA VALU)
  (MSG T "....found body-part mentioned within this phrase...")
  (CAADR (NODE-STACK)))
]
)(DE MARK-FEATURES (SENTENCE)
  PROG (NEW-SENTENCE STEM ENDING DEMORPH-INFOS STEM-INFOS)
  (PUSH SENTENCE 'DUMMY-POP-OFF*)
  NEXT-WORD
  (POP SENTENCE)
  (COND [(NULL SENTENCE) (RETURN (REVERSE NEW-SENTENCE))]
    [(NUMBERP (CAR SENTENCE))
      (PUSH NEW-SENTENCE (LIST (CAR SENTENCE) 'QUANT))
      (GO NEXT-WORD)]
    [(GETHASH-P (CAR SENTENCE) FUNCTION-HASH-TABLE)
      (PUSH NEW-SENTENCE (CONS (CAR SENTENCE) *INFO-BUS*))
      (GO NEXT-WORD)]
    [(GETHASH-P (CAR SENTENCE) SUPPLETION-HASH-TABLE)
      (SETQ STEM-INFOS (GETHASH (CAR *INFO-BUS*) STEM-HASH-TABLE))
      (PUSH NEW-SENTENCE
        (CONS (CAR *INFO-BUS*) (UNION STEM-INFOS (CDR *INFO-BUS*))))
      (GO NEXT-WORD))
    (SETQ ENDING NIL)
  ]
)(DE DEMORPHOLOGIZE-IT
  (SETQ DEMORPH-INFOS (DEMORPHOLOGIZE (CAR SENTENCE) ENDING))
  (SETQ STEM (CAR DEMORPH-INFOS))
  (SETQ ENDING (CAR DEMORPH-INFOS))
  (COND [(EQUAL (CAR (LAST (EXPLODE STEM))) 'I)
      (SETQ STEM (CHANGE-I-TO-Y STEM))]
    [(GETHASH-P STEM STEM-HASH-TABLE)
      (SETQ FEATURES *INFO-BUS*)
      (COND [(IN-P ENDING '(ED D))
        (SETQ FEATURES
          (APPEND FEATURES (CONS (CAR *INFO-BUS*) (UNION STEM-INFOS (CDR *INFO-BUS*))))
          (T (SETQ FEATURES (APPEND FEATURES (CONS (CAR *INFO-BUS*) (UNION STEM-INFOS (CDR *INFO-BUS*) (CDR STEM-INFOS)))))))]
      (SETQ STEM (CHANGE-I-TO-Y STEM))]
    [T (SETQ FEATURES (APPEND FEATURES (CONS (CAR *INFO-BUS*) (UNION STEM-INFOS (CDR *INFO-BUS*))))))]}
(DE MATCH-FEATURES-P (SUB-FEATURES FEATURES)
  (PROG NIL
    NEXT-SUB-FEATURE
      (COND 
        [(NULL SUB-FEATURES) (RETURN T)]
        [(EQUAL (CAR (EXPLODE (CAR SUB-FEATURES))) '-)
          (COND 
            [(INP (READLIST (CDR (EXPLODE (CAR SUB-FEATURES)))) FEATURES) (RETURN NIL)])
            [(NOT (INP (CAR SUB-FEATURES) FEATURES)) (RETURN NIL)])
          (POP SUB-FEATURES)
          (GO NEXT-SUB-FEATURE))]

  (DV N* (((N) N*) ((N))))
  (DV N1 (((N) (OF) (N))))

  (DE N1-ANALYSIS (CHUNK) (MSG "N1" T CHUNK))
  (DV N2 (((DET) N*) (N*) (ADJ* N1) ((DET) ADJ= N*) ((NEGATION) (ED) N*) ((NEGATION) N*) ((QU) N*) ((QU) (NUM) (BOUNDARY) (NUM) (N)) ((NUM) N*)) ((NUM) N*))

  (DE N2-ANALYSIS (PHRASE)
    (FOR (CONSTITUENT IN PHRASE)
      (DO (MSG T "CURRENT CONSTITUENT IS " CONSTITUENT T)))
    (PROG (HEAD TERM MOD TEMP)
      (SETQ HEAD (LAST PHRASE))
      (MSG T "HEAD" HEAD)
      (SETQ TOKEN (CAR (LAST (CADAR HEAD))))
      (MSG T TOKEN)
      (COND [(MEMBER TOKEN *FRAMES*)
        (SETQ TERM (CAR (FGET TOKEN 'SLOT-OF 'SVALUE)))
        (SETQ HEAD (HEAD-SEARCH-SPACE))
        (SETQ TOKEN (FGET&DEL TERM TOKEN))
        (FPUT TERM TOKEN 'SVALUE TEMP)])

  (DE NP-SPECIALIST (CONSTITUENT)
    (POP *MARKED-FEATURE-SENTENCE*)
    (COND [(NOT (EQUAL CONSTITUENT NIL))
      (COND [(EQUAL *NODE-FLAG* 'NP)
        (PUSH *NODE-STACK* CONSTITUENT)
        (MSG T "np-specialist active : " CONSTITUENT T)]
        [(NOT (EQUAL *NODE-FLAG* 'NP))
          (MSG T "np-specialist activated :" CONSTITUENT T)
          (SETQ *NODE-FLAG* 'NP)
          (PUSH *NODE-STACK* CONSTITUENT))]]
    (COND [(NOT (MEMB 'G CONSTITUENT))
      (MSG T "G CONSTITUENT")]}
(PROG (T1 T2)
  (COND [(EQUAL (CAR T1) NIL) (RETURN)]
    [T (SETQ T2 (CAR T1)) (SETQ T1 (CDR T1)) (EVAL T2)])
  (GO NEXT-ITEM))))

(PROG (T1 T2)
  (COND [(MEMB 'EXPECTATION *FRAMES*)
    (SETQ T1 (CAR (FGET 'EXPECTATION 'SLOT-OF '$VALUE')))
    (COND [(MEMB (CAR (FGET T1 'EXPECTATION '$VALUE)) CONSTITUENT)
      (SETQ *EXPECTED-VALUE* (CAR CONSTITUENT))
      (MSG T " currently expecting : " *EXPECTED-VALUE* T)
      (FGETEDEL T1 EXPECTATION-TARGET)
      (FPUT T1 EXPECTATION-TARGET '$VALUE *EXPECTED-VALUE*'))
    [(MEMB 'SIGNSYMPTOM CONSTITUENT)
      (SETQ *SIGNSYMPTOMS* (LIST *SIGNSYMPTOMS* (CAR CONSTITUENT)))
    ]
    [(AND [NOT (EQUAL *TEMP-CONJ* NIL)]
      [NOT (EQUAL *TEMP-CONJ* (CAR CONSTITUENT))]
      [INTERSECTION CONSTITUENT '(WEEK YEAR DAY MONTH)]
      (SETQ *CURRENT-TIME-FOCUS* (CAR CONSTITUENT))
      (SETQ *TEMP-CONJ* NIL)
      (SETQ *CURRENT-EVENT* (CAR *EVENT-STACK*))
    ]
    [(AND [INTERSECTION CONSTITUENT '(YEAR WEEK DAY MONTH)]
      [NOT (EQUAL *CURRENT-EVENT* NIL)]
      (FPUT *CURRENT-EVENT* 'TIME '$VALUE *CURRENT-TIME-FOCUS*)
    ]
    [(INTERSECTION CONSTITUENT 'WEEK]
      (SETQ *CURRENT-TIME-FOCUS* (CAR CONSTITUENT))
    ]
    [(INTERSECTION CONSTITUENT ‘FRAMES*]
      (SETQ T1 (CAR (INTERSECTION CONSTITUENT))
      (SETQ T2 (CAR (FGET T1 'SLOT-OF '$VALUE')))
      (COND [(EQUAL (CAR (FGET T2 T1 '$VALUE')) 'UNKNOWN)
        (FGETEDEL T2 T1)
        (FPUT T2 T1 '$VALUE (CAR CONSTITUENT))]
      [T (FGETEDEL T2 T1) (FPUT T2 T1 '$VALUE (CAR CONSTITUENT))]
    ]
  ]
)

(PROG (T3 T4)
  (COND [(AND [INTERSECTION CONSTITUENT ‘(YEAR WEEK DAY MONTH)]
    [MEMB 'NUM (CADR *NODE-STACK*)]
    (SETQ *NUMERIC-VALUE* (CONVERT-WORD-TO-NUMBER (CAADR *NODE-STACK*)))
  ]
    [(AND [INTERSECTION CONSTITUENT ‘(YEAR WEEK DAY MONTH)]
    [MEMB 'FUZZY (CADR *NODE-STACK*)]
    (SETQ *NUMERIC-VALUE* (CAADR *NODE-STACK*)))
  ]
    [(AND [MEMB *KEY-EVENT* CONSTITUENT]
      [MEMB (CAADR *NODE-STACK*) '(BEFORE AFTER)]
    ]
    [(SETQ *REL-TO-KEY-EVENT* (CAADR *NODE-STACK*))]
  ]
)

(DE NUM-ANALYSIS (NUMBER)
  (PROG (NUM) (SETQ NUM NUMBER) (SETQ *NUM-VALUE-STACK* NUM)))

(DE P1-ANALYSIS (PHRASE)
  (FOR (CONSTITUENT IN PHRASE)
    (DO (MSG T "CURRENT CONSTITUENT IS " CONSTITUENT T)))

(DE PARSE (SENTENCE)
(PROG (LOW-LEVEL-SENT FEAT-SENT SAVE-FEAT-SENT CONST-LIST PHIL-INFO)
  (SETQ CONST-LIST CONSTITUENT-LIST)
  (SETQ FEAT-SENT (REVERSE SENTENCE))
  (SETQ SAVE-FEAT-SENT FEAT-SENT)
  NEXT-PIECE-OF-SENTENCE
    (COND [(NULL FEAT-SENT) (RETURN (REVERSE LOW-LEVEL-SENT))]
      [(ALL-IN-P (CDAR FEAT-SENT) GAP-LIST)
       (PUSH LOW-LEVEL-SENT (CAR FEAT-SENT))
       (POP FEAT-SENT)
       (GO NEXT-PIECE-OF-SENTENCE))]
  NEXT-CONSTITUENT
    (SETQ PHIL-INFO (PHIL-GRAMMAR (EVAL (CAR CONST-LIST)) FEAT-SENT))
    (COND [(NULL CONST-LIST) (GO NEXT-PIECE)]
      [(NULL PHIL-INFO) (POP CONST-LIST) (GO NEXT-CONSTITUENT)]
      (PUSH LOW-LEVEL-SENT (LIST (CAR CONST-LIST) PHIL-INFO))
      (SETQ FEAT-SENT *INFO-BUSY*)
      (SETQ CONST-LIST CONSTITUENT-LIST)
      (GO NEXT-PIECE-OF-SENTENCE))
  NEXT-PIECE
    (MSG T "***** NO PSG-RULE FOUND *****" T))

(DE PARSE-OLD-VERSION (SENTENCE)
  (PROG (LOW-LEVEL-SENT FEAT-SENT SAVE-FEAT-SENT CONST-LIST PHIL-INFO)
    (SETQ CONST-LIST CONSTITUENT-LIST)
    (SETQ FEAT-SENT (MARK-FEATURES SENTENCE))
    (SETQ SAVE-FEAT-SENT FEAT-SENT)
    NEXT-PIECE-OF-SENTENCE
      (COND [(NULL FEAT-SENT) (RETURN (REVERSE LOW-LEVEL-SENT))]
        [(ALL-IN-P (CDAR FEAT-SENT) GAP-LIST)
         (PUSH LOW-LEVEL-SENT (CAR FEAT-SENT))
         (POP FEAT-SENT)
         (GO NEXT-PIECE-OF-SENTENCE))]
      NEXT-CONSTITUENT
        (SETQ PHIL-INFO (PHIL-GRAMMAR (EVAL (CAR CONST-LIST)) FEAT-SENT))
        (COND [(NULL CONST-LIST) (GO NEXT-PIECE)]
          [(NULL PHIL-INFO) (POP CONST-LIST) (GO NEXT-CONSTITUENT)]
          (PUSH LOW-LEVEL-SENT (LIST (CAR CONST-LIST) PHIL-INFO))
          (SETQ FEAT-SENT *INFO-BUSY*)
          (SETQ CONST-LIST CONSTITUENT-LIST)
          (GO NEXT-PIECE-OF-SENTENCE))
      NEXT-PIECE
        (MSG T "***** NO PSG-RULE FOUND *****" T))

(DE PARSE-TEXT (TEXT)
  (FOR (SENTENCE IN TEXT)
    (DO (MSG T "CURRENT SENTENCE IS " SENTENCE T)
      (LET (SENTENCE (PARSE SENTENCE))
        (MSG T "PHRASE-PARSE IS ")
        (SPRINT SENTENCE 2.))))

(DE PATIENT NIL
  (MSG "THE FOLLOWING INFORMATION IS AVAILABLE ON THE CURRENT PATIENT" T)
  (MSG "SEX OF THE PATIENT IS:" (FGET 'GENERAL-INFO 'SEX 'SVALUE) T)
  (MSG "PROFESSION OF THE PATIENT IS:" (FGET 'GENERAL-INFO 'PROFESSION 'SVALUE) T)
  (MSG "AGE OF THE PATIENT IS:" (FGET 'GENERAL-INFO 'AGE 'SVALUE) T)
(MSG "RACE OF THE PATIENT IS:"
        (FGET 'GENERAL-INFO 'RACE '$VALUE)
        T))

(DE PHIL-GRAMMAR (CFG-LIST FEAT-SENT)
    (PROG (GRAM-SENT GRAM-SENT-INFO CFG SAVE-FEAT-SENT SYMBOL-S)
        (SETQ SAVE-FEAT-SENT FEAT-SENT)
        (SETQ CFG (CAR CFG-LIST))
        NEXT-ALTERNATIVE
            (SETQ SYMBOL-S (CAR CFG))
            (COND ((AND [NULL FEAT-SENT] [NOT (NULL CFG)]))
                (SETQ GRAM-SENT NIL)
                (SETQ FEAT-SENT SAVE-FEAT-SENT)
                (SETQ CFG (POP CFG-LIST))
            )
            [(NULL CFG)
                (SETQ *INFO-BUS* FEAT-SENT)
                (RETURN (REVERSE GRAM-SENT))
            ]
            [(CONSP SYMBOL-S)
                (COND ((MATCH-FEATURES-P SYMBOL-S (CAR FEAT-SENT))
                    (PUSH GRAM-SENT (CONS SYMBOL-S (CAR FEAT-SENT)))
                    (POP CFG)
                    (POP FEAT-SENT))
                )
                [(NULL (CDAR FEAT-SENT))
                    (PUSH GRAM-SENT (CONS SYMBOL-S (CAR FEAT-SENT)))
                    (POP CFG)
                    (POP FEAT-SENT)]
                )
                [ T (SETQ GRAM-SENT NIL)
                    (SETQ FEAT-SENT SAVE-FEAT-SENT)
                    (SETQ CFG (POP CFG-LIST)))]
            )
            [(T (SETQ GRAM-SENT-INFO (PHIL-GRAMMAR (EVAL SYMBOL-S) FEAT-SENT)))
                (COND ((NULL GRAM-SENT-INFO))
                    (SETQ GRAM-SENT NIL)
                    (SETQ CFG (POP CFG-LIST))
                    (SETQ FEAT-SENT SAVE-FEAT-SENT))
                [ T (PUSH GRAM-SENT (CONS SYMBOL-S GRAM-SENT-INFO))
                    (SETQ FEAT-SENT *INFO-BUS*)
                    (POP CFG)]
            ]
        (GO NEXT-ALTERNATIVE)))

(DE PHRASE-LABEL (PARSED)
    (COND [(NULL PARSED) (MSG "DONE")]
        [(EQUAL 'N1 (CAR PARSED)) (MSG T "FOUND N1" (N1-ANALYSIS (CADR PARSED)))]
        [(EQUAL 'N2 (CAR PARSED))
            (MSG T "FOUND N2" (TERSE-N2-ANALYSIS (CADR PARSED)))]
        [(EQUAL 'P1 (CAR PARSED)) (MSG T "FOUND P1" (P1-ANALYSIS (CADR PARSED)))]
        [(EQUAL 'ADV1 (CAR PARSED))
            (MSG T "FOUND ADV1" (ADV1-ANALYSIS (CADR PARSED)))]
        [(EQUAL 'ADV2 (CAR PARSED))
            (MSG T "FOUND ADV2" (ADV2-ANALYSIS (CADR PARSED)))]
        [(EQUAL 'V1 (CAR PARSED))
            (MSG T "FOUND V1" (TERSE-V1-ANALYSIS (CADR PARSED)))]
        [(EQUAL 'V2 (CAR PARSED))
            (MSG T "FOUND V2" (TERSE-V2-ANALYSIS (CADR PARSED)))]
        [(T (MSG T "FOUND GAP" T (GAP-ANALYSIS (CAR PARSED)))]
    )

(DE PHRASE-SPECIALISTS (PARSED)
    (COND [(NULL PARSED) (MSG "DONE")])
[[EQUAL 'N1 (CAR PARSED)) (TERSE-N1-ANALYSIS (CADR PARSED)) (PUSH *PHRASE-LABEL-STACK* 'N1)]
[[EQUAL 'N2 (CAR PARSED)) (TERSE-N2-ANALYSIS (CADR PARSED)) (PUSH *PHRASE-LABEL-STACK* 'N2)]
[[EQUAL 'P1 (CAR PARSED)) (TERSE-P1-ANALYSIS (CADR PARSED)) (PUSH *PHRASE-LABEL-STACK* 'P1)]
[[EQUAL 'ADV1 (CAR PARSED)) (TERSE-ADV1-ANALYSIS (CADR PARSED)) (PUSH *PHRASE-LABEL-STACK* 'ADV1)]
[[EQUAL 'ADV2 (CAR PARSED)) (TERSE-ADV2-ANALYSIS (CADR PARSED)) (PUSH *PHRASE-LABEL-STACK* 'ADV2)]
[[EQUAL 'V1 (CAR PARSED)) (TERSE-V1-ANALYSIS (CADR PARSED)) (PUSH *PHRASE-LABEL-STACK* 'V1)]
[[EQUAL 'V2 (CAR PARSED)) (TERSE-V2-ANALYSIS (CADR PARSED)) (PUSH *PHRASE-LABEL-STACK* 'V2)]
[T (GAP-ANALYSIS PARSED)])

(DE PLACE-NUM-VALUE (NUM))
(MSG T "NUM-VAL IS" T (CONVERT-WORD-TO-NUMBER NUM))
(SETQ *TIME-REF-VALUE* (CONVERT-WORD-TO-NUMBER NUM)))

(DE POST-EXPECTATION NIL)
(MSG T "expec" T)
(FPUT PAIN-FEATURE 'EXPECTATION '$VALUE 'BODY-PART))

(DE PREP-SPECIALIST (CONSTITUENT))
(MSG T "prep-specialist invoked : » CONSTITUENT)
(SETQ *NODE-FLAG* 'PREP)
(PROG (T1 T2)
  (COND [[EQUAL *PARTICLE-FLAG* 'YES]
         (SETQ *CURRENT-CONSTITUENT* (CONS (CAR *CURRENT-CONSTITUENT*) '(PART)))
         (SETQ *PARTICLE-FLAG* NIL)
         (PUSH *NODE-STACK* *CURRENT-CONSTITUENT*)]
       [T (PUSH *NODE-STACK* CONSTITUENT)])
  (COND [T (MEMB 'FOR CONSTITUENT) (SETQ *DURATION-FLAG* 'YES)])
  (RETURN))
(POP *MARKED-FEATURE-SENTENCE*))

(DE PREVIOUS-DIAGNOSES NIL)
(PROG (T1 T2 T3)
  (COND [T (NOT (MEMBER 'PREVIOUS-DIAGNOSIS *FRAMES*))
         (MSG T "NO PREVIOUS DIAGNOSIS AVAILABLE")]
       [T (MEMBER 'PREVIOUS-DIAGNOSIS *FRAMES*) (GO ON)])
  ON (SETQ T1 (FGET 'PREVIOUS-DIAGNOSIS 'SLOT-OF '$VALUE))
  (SETQ T2 (FGET (CAR T1) 'PREVIOUS-DIAGNOSIS '$VALUE))
  (COND [T (EQUAL T1 NIL) (MSG T "NO PREVIOUS DIAGNOSIS AVAILABLE")]
       [T (MSG T "THE FOLLOWING DIAGNOSIS IS AVAILABLE" "

T}
(CAR (FGET (CAR T1) 'TIME-REF-VALUE 'VALUE))
""
(CAR (FGET (CAR T1) 'TYPE 'VALUE))
"(S)"
"BEFORE ADMISSION THERE WAS"
(CAR (FGET (CAR T1) 'PREVIOUS-DIAGNOSIS 'VALUE))
T]]]))

(DE PRO-SPECIALIST (CONSTITUENT)
(PROG (T1 T2)
  (MSG T "pronoun-specialist activated: " CONSTITUENT T)
  (POP *MARKED-FEATURE-SENTENCE*)
  (PUSH *NODE-STACK* "CURRENT-CONSTITUENT*)))

(DE PROCESS (SENTENCE) (SPLIT-PHRASES (PARSE SENTENCE)))

(DE PROCESS-TEXT (TEXT)
  (FOR (SENTENCE IN TEXT)
    (DO (MSG T "CURRENT SENTENCE IS " SENTENCE T)
         (LET (SENTENCE (BP SENTENCE)) (SPRINT SENTENCE 2.))))))

(DE READ-CASE (CASE)
  (FOR (SENTENCE IN CASE)
    (DO (MSG T "CURRENT SENTENCE IS " SENTENCE T)
         (LET (SENTENCE (PARSE SENTENCE))
              (MSG T "PHRASE-PARSE IS ")
              (SPRINT SENTENCE 2.))))))

(DE READ-PHRASE (SENTENCE)
  (FOR (PHRASE IN SENTENCE)
    (DO (MSG T "CURRENT PHRASE IS " PHRASE T)
         (LET (PHRASE (PHRASE-LABEL PHRASE))))))

(DE RELATION-PREP-ANALYSIS (PREP) (MSG "THIS IS REL-PREP-ANAL")))

(DE SPLIT-PHRASES (SENTENCE)
  (FOR (PHRASE IN SENTENCE)
    (DO (LET (PHRASE (LINGUISTIC-SPECIALISTS PHRASE)))))

(DE SS NIL (MSG))

(DV STEM-HASH-LIST ((ABDOMEN N BODY-PART) (ABDOMINAL BODY-PART ADJ)
                      (ADMISSION N) (APPETITE N) (BOWEL N BODY-PART)
                      (CAUCASIAN ADJ RACE) (CHANGE N V) (COLOR N) (COMPLAIN V
                        ((FGET-NEED-NEW (QUOTE COMPLAIN) (QUOTE CLASSIF)
                          (QUOTE $VALUE)))))
                      (COMPLAINT N SIGNSYMPTOM) (CONSTIPATION N SIGNSYMPTOM)
                      (DARK ADJ COLOR) (DAY N) (DEVELOP V (FGET-NEED-NEW
                        (QUOTE DEVELOP) (QUOTE SVALUE)))
                      (DIABETES N SIGNSYMPTOM) (DYSPNEA N SIGNSYMPTOM)
                      (EPISODE N) (EVERY QU ADJ) (EXERTIONAL ADJ)
                      (FEMALE N SEX) (FLANK N BODY-PART) (FOOD N)
                      (FOUR NUM) (EIGHTY-YEAR-OLD ADJ AGE)
                      (FREQUENT ADJ) (GASTROINTESTINAL ADJ BODY-PART)
                      (HEMATEMESIS N SIGNSYMPTOM) (ILL-DEFINED ADJ)
                      (IMMEDIATE ADJ) (INCREASE V) (INSULIN N MEDICIN)
                      (LITTLE ADJ) (MANY FUZZY DET) (MEAL N)
                      (MELLITUS N SIGNSYMPTOM)

(DV SUPPLETION-HASH-LIST ((HAS HAVE (ED)) (LOST LOOSE (ED)) (Said SAY (ED)) (WAS BE PAST) (WENT GO PRET) (WERE BE PRET) (BECAME BECOME PRET) (BEGAN BEGIN PRET) (DID DO PRET)))

(DV T1-1 (THIS EIGHTY-YEAR-OLD CAUCASIAN FEMALE COMPLAINED OF NAUSEA *COMMA* VOMITING *COHWA* ABDOMINAL SWELLING *COMMA* AND JAUNDICE *PERIOD*))

(DV T1-10 (HER ABDOMEN BECAME PROMINENT *COMMA* AND ONE WEEK BEFORE ADMISSION JAUNDICE WAS NOTICED *PERIOD*))

(DV T1-11 (THE URINE BECAME DARK *COMMA* BUT THE STOOLS DID NOT CHANGE IN COLOR *PERIOD*))

(DV T1-12 (THE PATIENT LOST TWENTY POUNDS IN WEIGHT THE MONTH BEFORE ADMISSION))

(DV T1-2 (SHE HAD DIABETES MELLITUS *COMMA* TREATED WITH INSULIN FOR SIX YEARS BEFORE ADMISSION *PERIOD*))

(DV T1-3 (SHE HAD HAD ILL-DEFINED GASTROINTESTINAL COMPLAINTS FOR MANY YEARS AND OCCASIONAL EPISODES OF NAUSEA AND VOMITING THREE YEARS PREVIOUSLY *PERIOD*))

(DV T1-4 (FOUR WEEKS BEFORE ADMISSION SHE DEVELOPED PAIN ACROSS THE UPPER ABDOMEN *COMMA* RADIATING TO THE FLANKS *PERIOD*))

(DV T1-5 (SHE ALSO COMPLAINED OF SHOOTING PRECORDIAL PAINS AND PALPITATION WITH SLIGHT EXERTIONAL DYSPEA *PERIOD*))

(DV T1-6 (INCREASING WEAKNESS AND CONSTIPATION DEVELOPED *COMMA* WITH BOWEL MOVEMENTS EVERY SECOND OR THIRD DAY *PERIOD*))

(DV T1-7 (TWO WEEKS BEFORE ADMISSION SHE LOST HER APPETITE AND BEGAN TO VOMIT FREQUENTLY IMMEDIATELY AFTER MEALS *PERIOD*))
(DV T1-8 (THERE WAS LITTLE NAUSEA AND NO HEMATEMESIS *PERIOD*))

(DV T1-9 (SHE RECOGNIZED NO RETAINED FOOD IN THE VOMITUS *PERIOD*))

(DV TEXT1 ((THIS EIGHTY-YEAR-OLD CAUCASIAN FEMALE COMPLAINED OF NAUSEA
  *COMMA* VOMITING *COMMA* ABDOMINAL SWELLING *COMMA* AND JAUNDICE
  *PERIOD*) (SHE HAD DIABETES MELLITUS *COMMA* TREATED WITH
  INSULIN FOR SIX YEARS BEFORE ADMISSION *PERIOD*) (SHE HAD HAD
  ILL-DEFINED GASTROINTESTINAL COMPLAINTS FOR MANY YEARS AND
  OCCASIONAL EPISODES OF NAUSEA AND VOMITING THREE YEARS
  PREVIOUSLY *PERIOD*) (FOUR WEEKS BEFORE ADMISSION *COMMA* SHE
  DEVELOPED PAIN ACROSS THE UPPER ABDOMEN *COMMA* RADIATING TO
  THE FLANKS *PERIOD*) (SHE ALSO COMPLAINED OF SHOOTING PRECORDIAL
  PAINS AND PALPITATION WITH SLIGHT EXERTIONAL DYSPEPSIA *PERIOD*)
  (INCREASING WEAKNESS AND CONSTIPATION DEVELOPED *COMMA* WITH
  BOWEL MOVEMENTS EVERY SECOND OR THIRD DAY *PERIOD*) (TWO WEEKS
  BEFORE ADMISSION SHE LOST HER APPETITE AND BEGAN TO VOMIT
  FREQUENTLY IMMEDIATELY AFTER MEALS *PERIOD*) (THERE WAS LITTLE
  NAUSEA AND NO HEMATEMESIS *PERIOD*) (SHE RECOGNIZED NO RETAINED
  FOOD IN THE VOMITUS *PERIOD*) (HER ABDOMEN BECAME PROMINENT
  *COMMA* AND ONE WEEK BEFORE ADMISSION JAUNDICE WAS NOTICED
  *PERIOD*) (THE URINE BECAME DARK *COMMA* BUT THE STOOLS DID NOT
  CHANGE IN COLOR *PERIOD*) (THE PATIENT LOST TWENTY POUNDS IN
  WEIGHT THE MONTH BEFORE ADMISSION *PERIOD*)))

(DE TIME NIL

(Prog (Ref-infO Temp)
  (Msg "At admission the following info is available: ")
  (Setq Ref-infO (Car (Fget 'admission 'Instances 'Svalue)))
  (Cond [(Equal Ref-infO Nil) (Msg T "No info available" T)]
        [T (Go next-frame)])

Next-frame
  (Cond [(null Ref-infO) (Return Nil)]
        [T (Setq fram (Fget (Car Ref-infO) 'Instances 'Svalue))
           (Msg fram T)
           (Setq Ref-infO (Cdr Ref-infO))
           (Go next-frame)])

(DE TO-SPECIALIST (constituent)
  (Msg "to-spec: " constituent T)
  (Cond [(intersection (Cadr 'marked-feature-sentence*) BNP)
          (Setq *current-constituent* '(to prep direction))
          (Msg T "this ok" T)
          (Setq T2 (Car (Fget 'direction 'slot-of 'Svalue)))
          (Msg "t2 p000?" T2 T)
          (Fget-need-new T2 'direction 'Svalue)])

(DE TRANS-INTRANS NIL (Msg "trans" T))

(DE UPDATE (NEW-ENTRY)
  (Setq parserfns
    (Remove NIL (Insert new-entry parserfns))
    (Dskout "parser")))

(DE UPDATE-HASH-TABLES NIL
  (Fill-hash-table function-hash-list function-hash-table)
  (Fill-hash-table stem-hash-list stem-hash-table))
(FILL-HASH-TABLE SUPPLETION-HASH-LIST SUPPLETION-HASH-TABLE)

(DV V-PART ((V) (PART))))

(DV V1 ((V-PART (N)) ((V) (ADV)) ((V) (ADJ)) ((V)) (V-INCHIOATIVE) (TO) V1))

(DV V2 (((HAVE) V1) ((ADV) V-PART) ((V-ASPECT) V1) ((DO) (NEGATION) V) (V)) (V-INCHIOATIVE) (TO) V1))

(DE VP-SPECIALIST (CONSTITUENT)
 (MSG T "vp-specialist activated : " CONSTITUENT T)
 (COND [(NOT (EQUAL *NODE-FLAG* 'VP)) (SETQ *NODE-FLAG* 'VP)])
 (PROG (T1 T2)
    (SETQ T1 (REVERSE (MAPCAR 'CONSP CONSTITUENT)))
    NEXT-ITEM
    (COND [(EQUAL (CAR T1) NIL) (RETURN)]
          [T (SETQ T2 (CAR T1)) (SETQ T1 (CDR T1)) (EVAL T2)])
    (GO NEXT-ITEM))
 (PUSH *NODE-STACK* *CURRENT-CONSTITUENT*)
 (POP *MARKED-FEATURE-SENTENCE*)
)

(DE ZAP (ENTRY) (SETQ PARSERFNS (REMOVE ENTRY PARSERFNS)) (UPDATE))

(NOCOMPILE (DEFV PARSERFNS (*CURRENT-EVENT* *CURRENT-MED-TYPE* *CURRENT-TIME-FOC
    US* *EVENT-STACK* *INFO-BUS* *KEY-EVENT* *KEY-EVENT-FLAG* *MARKED-FEATURE-SENTEN
    CE* *NODE-FLAG* *NODE-STACK* *NUMERIC-VALUE* *PARSE-TREE* *PARTICLE-FLAG* *PERIO
    D* SPECIALIST *REL-TO-KEY-EVENT* ACCESS-FRAMES ACCESS-QUERY ADJ* ADJ*-ANALYSIS A
    DV-SPECIALIST ADV1 ADV1-ANALYSIS ADV2 ADV2-ANALYSIS ALL-IN-P AMBIG-VN ANALYZE-PH
    RASE AND-ANALYSIS ASSESS-PAIN AVAILABLE-PATIENT-INFO BNP BOTTOM-UP BP BUILD-PHRA
    SE BVP CHANGE-FEATURES CHANGE-FEATURES-PART-TWO CHANGE-I-TO-Y CHECK-FOR-SIGNSYM
    TOM CONSTITUENT-ANALYSIS CONSTITUENT-LIST CREATE-HASH-TABLE DEMON-DROP DEMORPHOL
    OIZE DET-ANALYSIS DURATION-PREP-ANALYSIS ED ERRORS FILL-HASH-TABLE FIND-PARTICL
    E FIND-SUBJECT FIND-SYMPTOM FOCUS FOCUS-MONTH FOCUS-WEEK FOCUS-YEAR FORMAT FUNCT
    ION-HASH-LIST FUNCTION-WORD-DR-MARKER-SPECIALIST GAP-ANALYSIS GAP-LIST GET-STEM-
    FROM-ENDING GETHASH-P GROK IN-P ING LINGUISTIC-SPECIALISTS LINK-FRAMES LIST-INFO
    LIST-NIL LOOK-FOR-BODY-PART MARK-FEATURES MATCH-FEATURES-P N* N*-ANALYSIS N1 N1
    -ANALYSIS N2 N2-ANALYSIS NP-SPECIALIST NUM-ANALYSIS P1 P1-ANALYSIS PARSE PARSE-O
    LD-VERSION PARSE-TEXT PATIENT PHIL-GRAMMAR PHRASE-LABEL PHRASE-SPECIALISTS PLACE
    -NUM-VALUE POST-EXPECTATION PREP-SPECIALIST PREVIOUS-DIAGNOSES PRO-SPECIALIST PR
    OCCESS PROCESS-TEXT QUERY QUERY-FRAMES READ-CASE READ-PHRASE RELATION-PREP-ANALYS
    )
)
(DE BASE-NUMBER-P (NUMBER)
  (COND [(INP NUMBER '((1000. 1000000. 1000000000.)))]))

(DEFUNCT CONVERT-WORD-TO-NUMBER (WORD)
  (WITH (LIST-SPLIT)
    (COND [(NULL WORD) NIL]
      [(EQUAL WORD 'ZERO) 0.]
      [(EQUAL WORD 'ONE) 1.]
      [(EQUAL WORD 'TWO) 2.]
      [(EQUAL WORD 'THREE) 3.]
      [(EQUAL WORD 'FOUR) 4.]
      [(EQUAL WORD 'FIVE) 5.]
      [(EQUAL WORD 'SIX) 6.]
      [(EQUAL WORD 'SEVEN) 7.]
      [(EQUAL WORD 'EIGHT) 8.]
      [(EQUAL WORD 'NINE) 9.]
      [(EQUAL WORD 'TEN) 10.]
      [(EQUAL WORD 'ELEVEN) 11.]
      [(EQUAL WORD 'TWO) 12.]
      [(EQUAL WORD 'THIRTEEN) 13.]
      [(EQUAL WORD 'FOURTEEN) 14.]
      [(EQUAL WORD 'FIFTEEN) 15.]
      [(EQUAL WORD 'SIXTEEN) 16.]
      [(EQUAL WORD 'SEVENTEEN) 17.]
      [(EQUAL WORD 'EIGHTEEN) 18.]
      [(EQUAL WORD 'NINETEEN) 19.]
      [(EQUAL WORD 'TEN) 20.]
      [(EQUAL WORD 'THIRTY) 30.]
      [(EQUAL WORD 'FORTY) 40.]
      [(EQUAL WORD 'FIFTY) 50.]
      [(EQUAL WORD 'SIXTY) 60.]
      [(EQUAL WORD 'SEVENTY) 70.]
      [(EQUAL WORD 'EIGHTY) 80.]
      [(EQUAL WORD 'NINETY) 90.]
      [(EQUAL WORD 'HUNDRED) 100.]
      [(EQUAL WORD 'THOUSAND) 1000.]
      [(EQUAL WORD 'MILLION) 1000000.]
      [(EQUAL WORD 'BILLION) 1000000000.]
      [(INP '- (EXPLODE WORD))
        (SETQ LIST-SPLIT (STRING-SUBSET 'WORD))
        (PLUS (CONVERT-WORD-TO-NUMBER (CAR LIST-SPLIT))
                (CONVERT-WORD-TO-NUMBER (CADR LIST-SPLIT))))]]))

(DEFUNCT CONVERT-WORDS-TO-NUMBER (PHRASE)
  (PROG (PREVIOUS CURRENT COLLECTOR SUM)
    (SETQ PHRASE (REMOVE 'AND (REMOVE-CHARACTER '/, PHRASE)))
    (SETQ COLLECTOR 0.)
    (SETQ SUM 0.)
    (SETQ CURRENT (CONVERT-WORD-TO-NUMBER (POP PHRASE)))
    (COND [(BASE-NUMBER-P CURRENT) (SETQ SUM CURRENT) (GO NUMBER-NODE)]
          [(EQUAL CURRENT 100.) (SETQ COLLECTOR CURRENT) (GO HUNDRED-NODE)]
          [(NULL CURRENT) (RETURN (PLUS SUM COLLECTOR))]
          [(BASE-NUMBER-P CURRENT) (GO BASE-NUMBER-NODE)])
\begin{verbatim}
([ (EQUAL CURRENT 100.)
  (SETQ COLLECTOR (TIMES PREVIOUS 100.))
  (GO HUNDRED-NODE))
(SETQ COLLECTOR (SQUASH COLLECTOR CURRENT))
(GO NUMBER-NODE)
HUNDRED-NODE
  (SETQ PREVIOUS CURRENT)
  (SETQ CURRENT (CONVERT-WORD-TO-NUMBER (POP PHRASE)))
  (COND [(NULL CURRENT) (RETURN (PLUS SUM COLLECTOR))]
         [(NOT (BASE-NUMBER-P CURRENT))
          (M-ADD CURRENT COLLECTOR)]
         (GO NUMBER-NODE))
BASE-NUMBER-NODE
  (M-ADD (TIMES CURRENT COLLECTOR) SUM)
  (SETQ COLLECTOR 0.)
  (SETQ CURRENT (CONVERT-WORD-TO-NUMBER (POP PHRASE)))
  (COND [(NULL CURRENT) (RETURN SUM)]
         [(BASE-NUMBER-NODE)]
          
(DE LIST-SUBSET (SUB-LIST GIVEN-LIST)
  (PROG (SAVE-SUB-LIST LEFT MATCH-LIST)
      (SETQ SAVE-SUB-LIST SUB-LIST)
      NEXT-GIVEN-LIST-LETTER
        (COND [(NULL GIVEN-LIST) (RETURN NIL)]
               [(NOT (EQUAL (CAR SUB-LIST) (CAR GIVEN-LIST))) (GO PUSH-POP)]
               (SETQ MATCH-LIST GIVEN-LIST))
      NEXT-SUB-LIST-LETTER
        (POP SUB-LIST)
        (POP MATCH-LIST)
        (COND [(NULL SUB-LIST)
               (RETURN (LIST (REVERSE LEFT) SAVE-SUB-LIST MATCH-LIST))]
               [(NOT (EQUAL (CAR SUB-LIST) (CAR MATCH-LIST)))
               (SETQ SUB-LIST SAVE-SUB-LIST)]
               (GO PUSH-POP))]
      PUSH-POP
        (PUSH LEFT (POP GIVEN-LIST))
        (GO NEXT-GIVEN-LIST-LETTER))
      (GO NEXT-SUB-LIST-LETTER))
      (DM M-ADD (BODY)
      (LIST 'SETQ (CADDR BODY) (LIST 'PLUS (CADR BODY) (CADDR BODY))))
      (DE REMOVE-CHARACTER (CHARACTER WORD-LIST)
        (REMOVE NIL
          (MAPCAR '(LAMBDA (WORD)
            (SQUASH-LIST (REMOVE CHARACTER (EXPLODE WORD))))
           WORD-LIST))
      (DE SQUASH (ATOM1 ATOM2)
        (COND [(NULL ATOM1 ATOM2]
               [(NULL ATOM2 ATOM1]
               ([T (READLIST (APPEND (EXPLODE ATOM1) (EXPLODE ATOM2)))])))
      (DE SQUASH-LIST (GIVEN-LIST)
        (PROG (NEW-LIST)
          NEXT-ATOM)
      
\end{verbatim}
(COND [[(NULL GIVEN-LIST) (RETURN NEW-LIST))]
(SETP NEW-LIST (SQUASH NEW-LIST (CAR GIVEN-LIST))]
(POP GIVEN-LIST)
(GO NEXT-ATOM))]

(DE STRING-SUBSET (ITEM STRING)
 (MAPCAR #'(LAMBDA (SPLIT) (SQUASH-LIST SPLIT))
 (LIST-SUBSET (EXPLODE ITEM) (EXPLODE STRING))))

(NOCOMPILE (DEFV WORDNUMFNS (BASE-NUMBER-P CONVERT-WORD-TO-NUMBER CONVERT-WORDS-
 TO-NUMBER LIST-SUBSET M-ADD REMOVE-CHARACTER SQUASH SQUASH-LIST STRING-SUBSET)))

"Y
APPENDIX B

EFRL CODE FOR FRAME STRUCTURES

The code for the frames is written in EFRL, a version of FRL. EFRL has a cross-indexing facility for the slots of the frames and an inconsistency checker.

{(;

The following frames are saved in this file and in this order.
*DUMMY* *FRAME* *SLOT* *VALUE* ABOUT AGE COMPLAIN EVENT
GENERAL-INFO INDIR-OBJECT MEDICT NUM-VALUE
OBJECT OF ORIGIN-OF-PAIN PAIN-FRAME
PROFESSION QUALITY-OF-PAIN
RACE REF-POINT LOCATION-OF-PAIN REPORT-SYMPTOM
SEVERITY-OF-PAIN SEX
SUBJECT TEMPORAL-INFO TENSE TIME TIME-UNIT TO TREAT VERB
WITH YEAR MONTH WEEK REPORT-SIGN REPORT-LAB
---------------------------------------------

(FASSERT *DUMMY*
 (TYPE($VALUE(* FRAME*))
  ($SIF-ADDED(//#ERASE-DUMMY FRAME: VALUE:)))))

(FASSERT *FRAME*
 (TYPE($SIF-REMOVED((FREMOVES VALUE:
                             (QUOTE INSTANCES)
                             (QUOTE $VALUE)
                             (FNAME FRAME:))))
  (NAME($LAST-NAM(FROOO)))
  (INSTANCES($VALUE(*DUMMY*)))
  (MEDICT))))

(FASSERT *SLOT*
 (INSTANCES($VALUE(NUM-VALUE)
             (PROFESSION)
             (SEX)
             (RACE)
             (AGE)
             (WITH)
             (VERB)

167
(FASSERT *VALUE*
  (TYPE($VALUE(MEDICT)))
  (CSLOTS($VALUE(VALUE-OF)))
  (VALUE-OF))

(FASSERT ABOUT
  (SLOT-OF($VALUE(COMPLAIN)))
  (TYPE($VALUE(*SLOT*))))

(FASSERT AGE
  (SLOT-OF($VALUE(GENERAL-INFO)))
  (TYPE($VALUE(*SLOT*))))

(FASSERT COMPLAIN
  (OF($VALUE(REPORT-SYMPTOM)))
  (ABOUT($VALUE(UNKNOWN)))
  (TO($VALUE(UNKNOWN)))
  (CLASSIF($IF-NEEDED((FIND-PARTICLE))))

(FASSERT EVENT
  (REF-POINT($VALUE(UNKNOWN)))
  (RELATION-TO-REF-POINT($VALUE(UNKNOWN)))
  (DURATION($VALUE(UNKNOWN))))
(CLASSIF($VALUE(GENERIC)))
(NAME($LAST-NAM(EVT000))))

(FASSERT GENERAL-INFO
(CLASSIF($VALUE(GENERIC)))
(PROFESSION($VALUE(NOT-KNOWN)))
(SEX($VALUE(NOT-KNOWN)))
(RACE($VALUE(NOT-KNOWN)))
(AGE($VALUE(NOT-KNOWN))))

(FASSERT INDIR-OBJECT
(SLOT-OF($VALUE(REPORT-SYMPTOM)))
(TYPE($VALUE(*SLOT*))))

(FASSERT MEDICT
(TYPE($VALUE(*FRAME*)))
(CSLOTS$COMM("set of characteristic slots")
  "set of characteristic slots")
  "set of characteristic slots")
  "set of characteristic slots")
  "set of characteristic slots")
  "set of characteristic slots")
  "set of characteristic slots")
  "set of characteristic slots")
  "set of characteristic slots")
  "$VALUE(*SLOT*)))
  ($VALUE(*VALUE*))
  ($IF-REMOVED((/#CLEANUP-DELETED-FR VALUE: FRAME:)))
))

(FASSERT NUM-VALUE
(SLOT-OF($VALUE(WEEK))
  (MONTH)
  (YEAR)))
  (TYPE($VALUE(*SLOT*)))))

(FASSERT OBJECT
(SLOT-OF($VALUE(REPORT-SYMPTOM))))
  (TYPE($VALUE(*SLOT*)))))

(FASSERT OF
(SLOT-OF($VALUE(COMPLAIN))))
  (TYPE($VALUE(*SLOT*))))

(FASSERT ORIGIN-OF-PAIN
(SLOT-OF($VALUE(PAIN-FRAME)))
(TYPE($VALUE(*SLOT*))))

(FASSERT PAIN-FRAME
(CLASSIF($IF-NEEDED((ASSESS-PAIN))))
(NAME($LAST-NAM(PAINOOO)))
(QUALITY-OF-PAIN($VALUE(UNKNOWN)))
(ORIGIN-OF-PAIN($VALUE(UNKNOWN)))
(DIRECTION-OF-PAIN($VALUE(UNKNOWN)))
,LOCATION-OF-PAIN($VALUE(UNKNOWN)))
(AFFECTED-BODY-PART($VALUE(UNKNOWN)))
(TEMPORAL-INFO($VALUE(UNKNOWN)))

(FASSERT PROFESSION
(TYPE($VALUE(*SLOT*)))
(SLOT-OF($VALUE(GENERAL-INFO))))

(FASSERT QUALITY-OF-PAIN
(TYPE($VALUE(*SLOT*)))
(SLOT-OF($VALUE(PAIN-FRAME))))

(FASSERT RACE
(TYPE($VALUE(*SLOT*)))
(SLOT-OF($VALUE(GENERAL-INFO))))

(FASSERT REF-POINT
(SLOT-OF($VALUE(EVENT))))
(TYPE($VALUE(*SLOT*))))

(FASSERT LOCATION-OF-PAIN
(TYPE($VALUE(*SLOT*)))
(SLOT-OF($VALUE(PAIN-FRAME))))

(FASSERT REPORT-SYMPTOM
(CLASSIF($IF-NEEDED((FIND-SYMPTOM))))
(NAME($LAST-NAM(REPOOOO)))
(VERB($VALUE(UNKNOWN)))
(SUBJECT($VALUE(PATIENT)))
(OBJECT($VALUE(SYMPTOM)))
(INDIR-OBJECT($VALUE(MEDIC)))
(TENSE($DEFAULT(PAST)))
(TIME($DEFAULT(ADMISSION))))
(FASSERT SEVERITY-OF-PAIN
  (TYPE(SVALUE(*SLOT*)))
  (SLOT-OF(SVALUE(PAIN-FRAME))))

(FASSERT SEX
  (TYPE(SVALUE(*SLOT*)))
  (SLOT-OF(SVALUE(GENERAL-INFO))))

(FASSERT SUBJECT
  (TYPE(SVALUE(*SLOT*)))
  (SLOT-OF(SVALUE(REPORT-SYMPTOM))))

(FASSERT TEMPORAL-INFO
  (TYPE(SVALUE(*SLOT*)))
  (SLOT-OF(SVALUE(PAIN-FRAME))))

(FASSERT TENSE
  (TYPE(SVALUE(*SLOT*)))
  (SLOT-OF(SVALUE(REPORT-SYMPTOM))))

(FASSERT TIME
  (SLOT-OF(SVALUE(REPORT-SYMPTOM)))
  (TYPE(SVALUE(*SLOT*))))

(FASSERT TIME-UNIT
  (SLOT-OF(SVALUE(EVENT)))
  (TYPE(SVALUE(*SLOT*))))

(FASSERT TO
  (SLOT-OF(SVALUE(COMPLAIN)))
  (TYPE(SVALUE(*SLOT*))))

(FASSERT TREAT
  (WITH(SVALUE(REMEDY)))
  (CLASSIF($IF-NEEDED((FIND-PARTICLE)))))

(FASSERT VERB
  (TYPE(SVALUE(*SLOT*)))
  (SLOT-OF(SVALUE(REPORT-SYMPTOM))))
(FASSERT WITH
  (SLOT-OF($VALUE(TREAT)))
  (TYPE($VALUE(+SLOT*))))

(FASSERT YEAR
  (NUM-VALUE($VALUE(12)))
  (CLASSIF($IF-NEEDED((FOCUS-YEAR)))))

(FASSERT MONTH
  (TYPE($VALUE(YEAR)))
  (NUM-VALUE($VALUE(4)))
  (CLASSIF($IF-NEEDED((FOCUS-MONTH)))))

(FASSERT WEEK
  (TYPE($VALUE(MONTH)))
  (NUM-VALUE($VALUE(7)))
  (CLASSIF($IF-NEEDED((FOCUS-WEEK)))))

(FASSERT REPORT-SIGN
  (CLASSIF($IF-NEEDED((FIND-SIGN))))
  (NAME($LAST-NAM(REPOOO)))
  (VERB($VALUE(UNKNOWN)))
  (SUBJECT($VALUE(MEDIC)))
  (OBJECT($VALUE(SIGN)))
  (TENSE($DEFAULT(PAST)))
  (TIME($DEFAULT(ADMISSION))))

(FASSERT REPORT-LAB
  (CLASSIF($IF-NEEDED((FIND-LAB))))
  (NAME($LAST-NAM(REPOOO)))
  (VERB($VALUE(UNKNOWN)))
  (SUBJECT($VALUE(MEDIC)))
  (OBJECT($VALUE(LAB)))
  (TENSE($DEFAULT(PAST)))
  (TIME($DEFAULT(ADMISSION))))

{; -------------------------------}
The following frames are saved in this file and in this order.
SWELL-FRAME
-----------------------------}

(FASSERT SWELL-FRAME
  (NAME($LAST-NAM(SWEL00)))
  (CLASSIF($IF-NEEDED((LOOK-FOR-BODY-PART)))))

{; -------------------------------}
The following frames are saved in this file and in this order.
DEVELOP
-----------------------------}
(FASSERT DEVELOP
    (INTRANS(SUBJECT($VALUE(SIGNSYMPTOM)))))
(TRANS(OBJECT($VALUE(SIGNSYMPTOM))))
    (SUBJECT($VALUE(PATIENT)))))
(CLASSIF($IF-NEEDED((TRANS-INTRANS))))
APPENDIX C
COMPUTER EXAMPLES

[PHOTO: Recording initiated Wed 21-Nov-84 12:42AM]

>>> EFRL, Revised 7-12-83
>>> FRL-NGDE*** Type H for help, Q to quit
F>bp ti-1

np-specialist activated : (THIS DET)
np-specialist active : (EIGHTY-YEAR-OLD ADJ AGE)
np-specialist active : (CAUCASIAN ADJ RACE)
np-specialist active : (FEMALE N SEX)

vp-specialist activated : (COMPLAIN V ((FGET-NEED-NEW (QUOTE COMPLAIN) (QUOTE CLASSIF) (QUOTE $VALUE))) (ED))

ed-specialist activated

symptom-frame invoked: looking for symptom

prep-specialist invoked : (OF PREP)
np-specialist activated : (NAUSEA N SIGNSYMPTOM)

noun-verb ambiguity found - disambiguating...
ing specialist activated : (VOMIT N SIGNSYMPTOM (ING))

np-specialist active : (VOMIT N SIGNSYMPTOM)
np-specialist active : (ABDOMINAL BODY-PART ADJ)

vp-specialist activated : (SWELL V (FGET-NEED-NEW (QUOTE SWELL-FRAME) (QUOTE CLASSIF) (QUOTE $VALUE)) (ING))
ing specialist activated : (SWELL G N (FGET-NEED-NEW (QUOTE SWELL-FRAME) (QUOTE CLASSIF) (QUOTE $VALUE)) (ING))
np-specialist activated : (SWELL G N (FGET-NEED-NEW (QUOTE SWELL-FRAME) (QUOTE CLASSIF) (QUOTE $VALUE)))

looking for body-part ...
....found body-part mentioned within this phrase...ABDOMINAL

and-analysis started

np-specialist active : (JAUNDICE N SIGNSYMPTOM)

period-specialist invoked
NIL
F>pp evnt1

(FASSERT EVNT1
  (SYMPTOM $(VALUE (((((NIL NAUSEA) VOMIT) SWEL1) JAUNDICE))))
  (DURATION $(VALUE (ADMISSION) (NIL)))
  (KEY-EVENT $(VALUE (ADMISSION)))
  (CLASSIF $(VALUE (INDIVIDUAL)))
  (TYPE $(VALUE (EVENT))))

F>bp t1-2

pronoun-specialist activated: (SHE PRO)

vp-specialist activated : (HAVE V (ED))

ed-specialist activated

np-specialist active : (DIABETES N SIGNSYMPTOM)

np-specialist active : (MELLITUS N SIGNSYMPTOM)

noun-verb ambiguity found - disambiguating...

prep-specialist invoked : (WITH PREP)

np-specialist activated :(INSULIN N MEDICIN)

prep-specialist invoked : (FOR PREP)

np-specialist activated :(SIX NUM)

np-specialist active : (YEAR N (FGET-NEED-NEW (QUOTE YEAR) (QUOTE CLASSIF) (QUOTE $VALUE)) (SS))

prep-specialist invoked : (BEFORE PREP)

np-specialist activated :(ADMISSION N)

period-specialist invoked
NIL
F>pp evnt2

(FASSERT EVNT2
  (SYMPTOM $(VALUE (((((NIL DIABETES) MELLITUS))))
  (DURATION $(VALUE (YEAR) (6)))
  (KEY-EVENT $(VALUE (ADMISSION)))
  (RELATION-TO-KEY-EVENT $(VALUE (BEFORE) (YEAR) (6)))
  (CLASSIF $(VALUE (INDIVIDUAL)))
  (TYPE $(VALUE (EVENT)))))
F>bp t1-3

pronoun-specialist activated: (SHE PRO)

vp-specialist activated: (HAVE V (ED))

ed-specialist activated

vp-specialist activated: (HAVE V (ED))

ed-specialist activated

np-specialist activated: (ILL-DEFINED ADJ)

np-specialist active: (GASTROINTESTINAL ADJ BODY-PART)

np-specialist active: (COMPLAINT N SIGNSYMPTOM (SS))

prep-specialist invoked: (FOR PREP)

np-specialist activated: (MANY FUZZY DET)

np-specialist active: (YEAR N (FGET-NEED-NEW (QUOTE YEAR) (QUOTE CLASSIF) (QUOTE $VALUE)) (SS))

and-analysis started

temp-conYEAR

np-specialist active: (OCCASIONAL ADJ)

np-specialist active: (EPISODE N (SS))

prep-specialist invoked: (OF PREP)

np-specialist activated: (NAUSEA N SIGNSYMPTOM)

and-analysis started

noun-verb ambiguity found - disambiguating...

np-specialist active: (VOMIT VN SIGNSYMPTOM)

np-specialist active: (THREE NUM)

np-specialist active: (YEAR N (FGET-NEED-NEW (QUOTE YEAR) (QUOTE CLASSIF) (QUOTE $VALUE)) (SS))

adv spec

period-specialist invoked
NIL
F>pp evnt3

(FASSERT EVNT3
  (SYMPTOM ($VALUE ((NIL COMPLAINT)))))
  (DURATION ($VALUE (YEAR) (MANY))))
(KEY-EVENT (SVALUE (ADMISSION) (NIL)))
(RELATION-TO-KEY-EVENT (SVALUE (BEFORE) (YEAR) (MANY)))
(CLASSIF (SVALUE (INDIVIDUAL)))
(TYPE (SVALUE (EVENT))))

F>pp evnt4

(FASSERT EVNT4
  (SYMPTOM (SVALUE (((NIL NAUSEA) VOMIT))))
  (DURATION (SVALUE (INTERMITTENT))))
  (KEY-EVENT (SVALUE (ADMISSION))))
  (RELATION-TO-KEY-EVENT (SVALUE (BEFORE) (YEAR) (3)))
  (CLASSIF (SVALUE (INDIVIDUAL)))
  (TYPE (SVALUE (EVENT))))

F>bp t1-4

np-specialist activated : (FOUR NUM)

np-specialist active : (WEEK N (FGET-NEED-NEW (QUOTE WEEK) (QUOTE CLASSIF)
  (QUOTE $VALUE)) (SS))

  current time focus : week

prep-specialist invoked : (BEFORE PREP)
np-specialist activated : (ADMISSION N)

pronoun-specialist activated: (SHE PRO)

vp-specialist activated : (DEVELOP V (FGET-NEED-NEW (QUOTE DEVELOP)
  (QUOTE CLASSIF) (QUOTE $VALUE)) (ED))

ed-specialist activated

TRANS

np-specialist activated : (PAIN N (FGET-NEED-NEW (QUOTE PAIN-FRAME)
  (QUOTE CLASSIF) (QUOTE $VALUE)))

valDEVELOP

prep-specialist invoked : (ACROSS LOCATION PREP)
np-specialist activated : (THE DET)

np-specialist active : (UPPER ADJ LOCATION)
np-specialist active : (ABDOMEN N BODY-PART)

vp-specialist activated : (RADIAT V (ING))
to-spec: (TO OPERATOR)

this ok
t2 p000?PAIN1

expec

np-specialist activated : (THE DET)
np-specialist active : (FLANK N BODY-PART (SS))

  currently expecting : FLANK

period-specialist invoked
NIL
F>pp evnt5

(FASSERT EVNT5
  (SYMPTOM (SVALUE ((PAIN1))))
  (DURATION (SVALUE (WEEK) (4))))
  (KEY-EVENT (SVALUE (ADMISSION))))
  (RELATION-TO-KEY-EVENT (SVALUE (BEFORE) (WEEK) (4))))
  (CLASSIF (SVALUE (INDIVIDUAL))))
  (TYPE (SVALUE (EVENT))))

F>pp pain1

(FASSERT PAIN1
  (DIRECTION (SVALUE (FLANK)) (SIF-NEEDED ((POST-EXPECTATION))))
  (BODY-PART (SVALUE (ABDOMEN))))
  (LOCATION (SVALUE (UPPER))))
  (INTENSITY (SVALUE (UNKNOWN))))
  (CLASSIF (SVALUE (INDIVIDUAL))))
  (TYPE (SVALUE (PAIN-FRAME))))

F>bp t1-5

pronoun-specialist activated: (SHE PRO)
adv spec
vp-specialist activated : (COMPLAIN V ((FGET-NEED-NEW (QUOTE COMPLAIN)
  (QUOTE CLASSIF) (QUOTE $VALUE))) (ED))

  ed-specialist activated
    symptom-frame invoked: looking for symptom

prep-specialist invoked : (OF PREP)
vp-specialist activated : (SHOOT V (ING))

np-specialist activated : (PRECORDIAL ADJ BODY-PART)

np-specialist active : (PAIN N (FGET-NEED-NEW (QUOTE PAIN-FRAME) (QUOTE
CLASSIF) (QUOTE $VALUE)) (SS))

valBODY-PART
fnpAIXN2
  and-analysis started

np-specialist active : (PALPITATION N SIGNSYMPTOM)

prep-specialist invoked : (WITH PREP)
np-specialist activated : (SLIGHT ADJ)

np-specialist active : (EXERTIONAL ADJ)
np-specialist active : (DYSPNEA N SIGNSYMPTOM)

period-specialist invoked
NIL
F>pp evnt6

(FASSERT EVNT6
  (SYMPTOM ($VALUE (((PAIN2) PALPITATION) DYSPNEA))))
  (DURATION ($VALUE (((WEEK 4)))))
  (KEY-EVENT ($VALUE (ADMISSION))))
  (RELATION-TO-KEY-EVENT ($VALUE (((BEFORE WEEK 4)))))
  (CLASSIF ($VALUE (INDIVIDUAL))))
  (TYPE ($VALUE (EVENT))))

F>bp ' (she complained of nausea for six years and three months
>before admission *period*)

pronoun-specialist activated: (SHE PRO)

vp-specialist activated : (COMPLAIN V (((FGET-NEED-NEW (QUOTE COMPLAIN) (QUOTE CLASSIF) (QUOTE $VALUE)) (ED))

  ed-specialist activated

  symptom-frame invoked: looking for symptom

prep-specialist invoked : (OF PREP)
np-specialist activated : (NAUSEA N SIGNSYMPTOM)

prep-specialist invoked : (FOR PREP)
np-specialist activated : (SIX NUM)

np-specialist active : (YEAR N (((FGET-NEED-NEW (QUOTE YEAR) (QUOTE CLASSIF) (QUOTE $VALUE)) (SS)))

  and-analysis started

temp-conYEAR
np-specialist active : (THREE NUM)

np-specialist active : (MONTH N (((FGET-NEED-NEW (QUOTE MONTH) (QUOTE CLASSIF) (QUOTE $VALUE)) (SS)))

prep-specialist invoked : (BEFORE PREP)
np-specialist activated : (ADMISSION N)

period-specialist invoked
NIL
F>pp evnt7

(FASSERT EVNT7
(SYMPTOM (SVALUE (NIL) ((NIL NAUSEA))))
(DURATION (SVALUE (MONTH) (3) (YEAR) (6)))
(KEY-EVENT (SVALUE (ADMISSION) (NIL)))
(RELATION-TO-KEY-EVENT (SVALUE (BEFORE) (MONTH) (3) (YEAR) (6)))
(CLASSIF (SVALUE (INDIVIDUAL)))
(TYPE (SVALUE (EVENT))))

[PHOTO: Recording terminated Wed 21-Nov-84 12:45AM]