The Propositional Analysis System

Version 1.0

January 15, 1987

Althea A. Turner

Department of Psychology

and

Institute of Cognitive Science

University of Colorado

Boulder, CO 80302

This research was supported by the Air Force Human Resources Laboratory, Brooks Air Force Base, Texas.

TABLE OF CONTENTS

SUMMARY

A theory of language comprehension and memory has been developed by Kintsch and his colleagues (Kintsch, 1974; Kintsch and van Dijk, 1978; van Dijk and Kintsch, 1983). According to this theory, language is broken down into a propositional representation during comprehension. This theory of comprehension is briefly described. It is possible to analyze written information into a a set of propositions which embody its meaning. The method for deriving a propositional representation is detailed, using a number of examples.

A computer system has been developed to aid propositional analysis. This system runs on the IBM PC family under the DOS operating system. It allows a user to create propositions by identifying propositional elements in the text or specifying embedded propositions. The user is able to concentrate on the meaning of the passage rather than dividing attention between the passage and the bookkeeping involved in performing a propositional analysis. The system is menu-based, and each menu is described in detail.

INTRODUCTION

One of the most complex tasks people do is to understand language. With what appears to be little effort, we are able to derive information from the complex system of symbols that is natural language. Researchers in machine intelligence would be delighted if they were able to build a computer system that could understand language as well as the average child. The meaning in most texts or utterances seems obvious to us. Yet how we are able to take the symbols in a language system and convert them into a representation of its meaning is not at all clear.

One proposal is that when we understand language, we employ a set of complex processes to build a network of semantic units, called **propositions**†. This network of propositions represents the meaning that is understood. Language understanding, then, consists of deriving a set of interrelated propositions from incoming information -- whether it be from a novel, a conversation with a friend, a scientific article, a political speech, a technical manual, or a newspaper article -- and integrating this set of propositions with knowledge we already have in memory. Some of the information may be lost, some may be misunderstood, but much of it is retained in memory and becomes part of the knowledge used for interpretation in the future.

Similarly, the propositional content of memory can be transformed back into language. This requires that the language producer make choices as to the information that should be included, the order in which to produce the ideas, the vocabulary and syntactic forms to be used, rhetorical considerations, and so on. While language production is an interesting and important topic, the focus of the present work is on the derivation of a propositional representation of meaning from natural language discourse. Therefore, production is mentioned here briefly to be complete, but will not be discussed further.

Information that is expressed in oral or written language can be analyzed into its semantic units. The propositional network that results from this analysis can be used for a variety of purposes. Some of these purposes are scientific while others are more applied. But because language is so rich and varied, some method of expressing its meaning more formally is important to any of these purposes. On the scientific side, propositional analysis has been used for such varied purposes as exploring issues about language comprehension, the theoretical modeling of comprehension and memory, and comparing educational strategies. More applied uses have been to examine computer programmers' abilities and writing technical manuals.

A number of systems of discourse analysis have been developed (e.g. Kintsch, 1974; Kintsch and van Dijk, 1978; Meyer, 1975, 1983; Anderson, 1976, 1983; Frederiksen, 1975, 1977; Graesser, 1981; Graesser and Goodman, 1983). Each system has its strengths and weaknesses. The present work is based on the system developed by Kintsch and his colleagues. This system has been widely used for a variety of reasons. One is that it is relatively easy to learn and use. Several practical manuals have been developed to explicate this method of propositional analysis (Turner and Greene, 1978; Bovair and Kieras, 1983). It requires no special tools to perform, but has been accomplished using pencil and paper. It is not subject to the problems of machine

---

†As used here, a proposition is taken to be an intensional unit, corresponding roughly to the meaning of a declarative sentence. Propositions are a form of conceptual representation in a cognitive model of language comprehension. Although it is not necessary that propositions are expressible only in natural language, as opposed to images or other possible semiotic systems, the discussion here is limited to propositions as they are expressed in natural language.

In linguistics and much of philosophy, a proposition is defined as the meaning of a sentence, and hence provides a conceptual or intensional semantics. Most modern logic defines the proposition in referential, denotational, or extensional terms, that is, propositions, or the sentences expressing them, are related to truth values, and can be true or false. In some more recent theories of logics, truth values have been made relative, and a proposition is the set of possible worlds in which it is true. A more sophisticated definition is that a proposition is a function from possible worlds to truth values. For a more complete discussion, see van Dijk and Kintsch (1983).

parsers, i.e. that all information, including each and every word used, must be programmed into a computer before an analysis can be done. Two people can achieve reasonable agreement as to the propositional content of a discourse after minimal training.

This is not to suggest, however, that it is a trivial process. Some sensitivity to language use is necessary. The bookkeeping aspects of an analysis of a text longer than a few paragraphs can become overwhelming. Errors can be introduced easily and are easy to overlook. Altering or correcting some part of an analysis can be viewed as more trouble than the additional accuracy is worth. A computer based system to aid in propositional analysis will be described below. This system eliminates many of the above problems, as well as introducing significant improvements over the pencil and paper method of propositional analysis.

PROPOSITIONAL REPRESENTATION AND THE COMPREHENSION OF NATURAL LANGUAGE

The following discussion of propositional representation and the theory of language comprehension is based on the theory developed by Kintsch and his colleagues (Kintsch, 1974; Kintsch and van Dijk, 1978; van Dijk and Kintsch, 1983). Further information on this theory can be obtained by consulting these references. Additional information about this method of propositional analysis can be found in Turner and Greene (1978) and Bovair and Kieras (1983).

When we understand language, we employ a set of complex processes to build a network of semantic units, called **propositions.** Each proposition represents a single idea. This network of propositions represents the meaning that is understood. Each proposition consists of a set of concepts. The first of these is the **predicate** of the proposition. The others are the proposition's **arguments**. A predicate asserts a relationship among its arguments. This relationship may express an action, a qualification, a temporal ordering, or any other conceptual relation possible among a set of concepts or propositions.

A given predicate can be described as a **frame**, which includes a set of slots for arguments. Each slot represents a particular semantic **case** which specifies the role and properties of its arguments. For example, the predicate **GIVE** will have the following case slots for arguments:

AGENT: the instigator of the action GIVE, i.e. the one who gives;
GOAL: the goal of the action GIVE, i.e. to whom the object is given;
OBJECT: the object of the action GIVE, i.e. what is given.

The propositional frame for this predicate is GIVE [AGENT, GOAL, OBJECT]. During understanding, this propositional frame would be filled out with specific information from the discourse. For example, the sentence "John gave Bill a marble" would be represented as "give [John, Bill, marble]." In some sentences, arguments for some of the case slots may be absent. For example, in the sentence "Bill was given a marble," the person who gave the marble to Bill is not specified.

The words used can be considered to be a surface form, or **type**, of an underlying concept. The concept is represented in a proposition by a **token**, a word or set of words which stand for the more abstract semantic concept. A given predicate can be expressed by a variety of words or sets of words. For example, the sentence "John presented Bill with a marble" can be analyzed into the proposition "give [John, Bill, marble]." The word "presented" is the type for the token word "give," which represents the abstract concept. Just as a predicate can be expressed in many ways, a proposition can have various surface realizations in natural language. To return to the example above, "Bill was presented with a marble by John," "A marble was given by John to Bill," and "John gave a marble to Bill" are all equivalent forms of the proposition "give [John, Bill, marble]."

As a discourse is understood, a representation of its meaning is produced as a result of the comprehension process. This process has components which work from the actual words used, from previous knowledge, and from strategies and purposes. The final understanding a person acquires from a discourse is a mixture of propositions derived directly from the discourse, more general propositions produced through deletion, abstraction, and construction processes, propositions resulting from the integration of new propositions with other propositions, inferences, and any previous knowledge that is necessary to complete a coherent and well integrated network. This complete network of propositions represents the meaning understood from the discourse by a particular language user.

The network of propositions which are directly expressed in the discourse is often referred to as the **microstructure**. The set of propositions which make up the high level, more general representation of the ideas in a discourse is its **macrostructure**. These propositions could be important microstructure propositions which are explicitly expressed in the discourse. Alternatively, these macrostructure propositions can be derived from the microstructure propositions through a set of strategies which allow the new information to be transformed or relevant previous knowledge to be included. A set of macro-operators can be applied to selectively delete or abstract certain types of information (Turner, McCutchen, and Kintsch, 1986; Turner and McCutchen, in preparation), resulting in propositions that are more general than the original information. A macrostructure proposition can be the result of integrating previous knowledge with the information presented in the discourse, causing previously known propositional content to be included in the representation of the discourse or new propositions to be created through inferential processes. High level strategies which guide comprehension will also help to determine the propositions which become part of the macrostructure.

Some propositions that are part of the a person's understanding of a discourse are not strictly derived from the discourse itself. Instead, these propositions are derived from the knowledge the person already had. The information presented in the discourse must be related to what is already known to be understood. A model of the local situations or facts from the discourse is constructed during comprehension from memories of similar or related situations or facts. This model has been referred to as the **situation model** (van Dijk and Kintsch, 1983). The situation model contains the information that was left implicit in the discourse. It provides access to a coherent model of the discourse which includes relevant but unstated information and is used in further understanding of the remainder of the discourse. Since what each person knows about a topic is likely to be idiosyncratic, the situation model will be the source of the greatest individual differences in the representation of the meaning of a discourse.

Propositional analysis allows information in a discourse to be more formally specified. Equivalent ideas can be represented by the same proposition, regardless of their surface form. One advantage of representing information in this way is that it allows the structure of the ideas contained in a discourse to be examined. The semantic structure of a discourse is determined by the connections between ideas. These may vary in number, type, and strength, depending on factors related to the discourse and to the person understanding it. The order in which ideas are presented, the difficulty of the vocabulary and syntax used, and rhetorical devices that indicate importance and focus are some examples of the factors in the discourse affecting its representation in memory. The comprehender's abilities, previous knowledge, and purposes are some human variables which could affect the representation.

Another advantage of propositional analysis is that it is easier to discover ideas that underlie the meaning of a discourse but that are not expressed. Sometimes these potential inferences are useful in bridging gaps between expressed ideas. At other times, they are important assumptions or abstractions that are necessary for a high level understanding of the expressed ideas. During comprehension, these inferences must be derived, and that process adds to the difficulty of the discourse.

By itself, propositional analysis is a method of delineating meaning semi-formally. However, when combined with a process model, the combination provides a way to predict how a discourse will be received by a comprehender. The analysis can help identify information that a comprehender should already know before it is possible to understand a particular discourse. Patterns of information recall from a discourse as well as parts of a discourse that will be most difficult for a person with poor comprehension abilities can be predicted. In addition, it can make it possible to restructure the information to be presented so that it is more easily comprehended.

Some Simple Examples of Propositions

Before getting into more complicated examples, let us look at a few examples of propositions from simple sentences or sentence fragments. The convention for writing propositions used here is to present a predicate followed by a set of arguments. The arguments are enclosed in square brackets ([]) and separated by commas, if there is more than one argument. If a predicate does not have any arguments, it is followed by empty square brackets. In some cases the predicate is not directly expressed in the phrase or sentence, but must be inferred. Some predicates are standard tokens which denote basic relationships. Some examples of these are "number-of," which indicates the relationship of numerosity (the number of something that can be counted) and "possess," which indicates the relationship of possession or belongingness (not simple ownership).

The first set of examples are phrases or sentences expressing a single proposition. These examples are simple and will not be explained further.

A red marble	color-of [marble, red]
Jason's shoes	possess [Jason, shoes]
some pennies	quantify [pennies, some]
two pennies	number-of [pennies, two]
the car's tire	part-of [car, tire]
a stick of wood	consist-of [stick, wood]
the sheep in the pasture	location:in [sheep, pasture]
the Pacific Ocean	name-of [ocean, Pacific]
apples and oranges	conjunct:and [apples, oranges]
breathing	breathe [ ]
John played with the ball.	play [John, ball]
Mary bought a balloon.	buy [Mary, balloon]
The wind opened the door.	open [wind, door]
The earthquake caused the tsunami.	cause [earthquake, tsunami]

In the next set of examples, several propositions are expressed in each sentence.

Jason has black shoes.	possess [Jason, color-of [shoes, black]]
The tire of the car was flat.	flat [part-of [car, tire]]
The sheep were in the lush pasture.	location:in [sheep, qualify [pasture, lush]]
The wind opened the door suddenly.	manner [open [wind, door], sudden]
The agent said the flight was late.	say [agent, late [flight]]
The cliff was sixty feet high.	extent-of [possess [cliff, height], number-of [feet, sixty]]
The newspaper cost 50 cents.	cost-of [newspaper, number-of [cents, fifty]]
Six to ten inches of snow are predicted.	predict [extent-of [snow, [range-of [inches, six, ten]]]
John was breathing deeply and slowly.	manner [breathe [John], conjunct:and [deep, slow]]
If it rains, the game will be cancelled.	condition:if [rain [ ], cancel [game]]
Frank runs hard to improve his running speed.	purpose-of [manner [run [Frank], hard], improve [Frank, rate-of [run [Frank]]]]
The story was about a prince and a pauper.	topic-of [story, conjunct:and [prince, pauper]]
Moonlight glistened on the high peaks.	location:on [glisten [moonlight], quantify [peaks, high]]
The satellite moves from west to east in an inclined elliptical orbit.	location:in [move [satellite, west, east], incline [shape-of [orbit [satellite], ellipse]]]
Continents are thought to drift like icebergs on the ocean.	contrast:like [drift [continents], location:on [drift [icebergs], ocean]]
Although the computer is broken, it can still be used as a paperweight.	concession [broken [computer], possible [intensify [use-for [computer, paperweight], still]]]

A Simple Example of a Sentence Analysis

To illustrate the process of deriving the propositions from text that is more complicated than those above, we will analyze the following sentence in detail:

> Wegener proposed a theory of continental drift when geologists were beginning to find conventional theories of continental permanence inadequate.

The words in the text are clues to the underlying meaning. They may be very direct translations or subtle indications. The elements of the propositions are **tokens** which are represented by the **types** of the actual words used in the sentence. Not all words are types for concepts, however. Some words (or parts of words) may indicate which concepts should fill which cases in a predicate frame. Other words may indicate that a proposition should be embedded as arguments of another proposition. Some words are necessary for the sentences to be syntactically well formed.

Tense information, such as the past progressive tense of the verb "to begin," is generally ignored in an analysis of meaning, unless it contributes to the semantic content, as when it is used to indicate temporal order or time frame. Although many predicates are verbs (e.g. "propose," "begin," "find"), predicates may be derived from many other parts of speech, such as adjectives, adverbs, and conjunctions.

Action and state verbs: The initial step in an analysis is to identify predicates. A proposition must have a predicate, thus each predicate identified will define a proposition. The simplest ones to identify are lexical verbs, those verbs that carry meaning about actions or states. In cases where the predicate is a verb, the predicate token is the infinitive form of the verb. The verbs in the passage below are indicated in boldface, and the first propositions are defined for the sentence.

> Wegener **proposed** a theory of continental drift when geologists were **beginning** to **find** conventional theories of continental permanence inadequate.

propose [ ]
begin [ ]
find [ ]

Coordinators: Text coordinators are relatively easy to identify as well. This class of words defines relationships which bind propositions into more complex ideas. There are a number of classes of coordinators, such as time, location, cause, purpose, condition, or concession. Only one example of a coordinator is in this sentence, the word "when," which expresses a temporal relationship. Since many words used to express time can be used to express other types of coordination, the prefix "time:" is added to the predicate to assure that its meaning is not ambiguous. (Hereafter words previously identified as types will be underlined; newly identified types and tokens will be presented in boldface.)

> Wegener proposed a theory of continental drift **when** geologists were beginning to find conventional theories of continental permanence inadequate.

propose [ ]
time:when [ ]
begin [ ]
find [ ]

Adjectives and adverbs: Adjectives and adverbs can present some difficulty. The relationship between a modifier and its object is often inferred. A common predicate specifying general modification is "qualify," which indicates that an object has a property or attribute. The adjective "conventional" is an example of the predicate "qualify," and it will be the second argument of this proposition. (The first is the object of the modification, in this case "theories of continental permanence." These arguments will be filled in later.)

Sometimes a modifier is actually a predicate, as in the case of the adjective "inadequate." Other times, the noun that is modified expresses an action or state. Both instances of the modifier "continental" are examples of words which modify actions. The concept "continents" expressed by the word "continental" is actually an argument of the proposition defined by the concept "drift", i.e. continents drift.

> Wegener proposed a theory of continental **drift** when geologists were beginning to find **conventional** theories of continental **permanence** **inadequate**.
>
> propose [ ]
> **drift [ ]**
> time:when [ ]
> begin [ ]
> find [ ]
> **qualify [ ]**
> **permanent [ ]**
> **inadequate [ ]**

Miscellaneous: Finally, the text can be scanned for miscellaneous predicates that have been missed. In this sentence, the only predicates that have been omitted are those represented by the phrase "theory of." The exact relationship here must be inferred. We will use the token "topic-of" to indicate that something is about something else, as in "a theory about drifting continents" or "a theory about permanent continents." The function word "of" indicates that there is a relationship between the preceding and following text elements. The predicate "topic-of" will be attached to this word in the text, since there is no more obvious type for this token.

> Wegener proposed a theory **of** continental drift when geologists were beginning to find conventional theories **of** continental permanence inadequate.
>
> propose [ ]
> **topic-of [ ]**
> drift [ ]
> time:when [ ]
> begin [ ]
> find [ ]
> qualify [ ]
> **topic-of [ ]**
> permanent [ ]
> inadequate [ ]

Anchoring propositions: The token "topic-of" was attached to the word "of" in the sentence. Although the word "of" does not convey this meaning, it is the point in the text at which a person who is reading the sentence knows that a new idea is required. If the sentence has been "Wegener proposed a theory when ... ," there would have been no requirement to predicate what the theory he proposed was about. (This is not to say that some readers would not have wondered or even hypothesized what his theory was about, only that the words in the sentence do not direct the reader to form this idea.) But because the sentence reads "Wegener proposed a theory of ... ," we as comprehenders expect that the topic of his theory will appear next. This point in the text at which a proposition should be formed is called the **anchor** of a proposition.

The anchors of the other propositions in this sentence are more obvious. Each word which was underlined above is the anchor of a proposition. Notice also that the order in which the propositions are listed is determined by the order of the anchors in the sentence.

Filling in the arguments: Once the predicates have been identified, the arguments can be placed into the slots of each predicate. For simplification of this example, we will keep the discussion of the precise slot labels, or cases, to a minimum. Instead, we will informally indicate the role of an argument in the proposition. Inclusion of the slot labels in the propositions, while rendering the analysis more complete, is not necessary and can make a proposition more difficult to apprehend.

A predicate frame generally has from one to four or five slots, with two slots by far the usual case. A predicate and its argument slots together express a single idea. A slot may be optional, in that it may not be filled with a value in each expression of the predicate in natural language. However, each slot in a predicate frame should define an element that contributes in an integral way to the single idea represented by the predicate.

An argument may assume a value that is a single concept or one that is more complex, that is, a value that is itself a proposition. We will look first at the instances of arguments that are single concepts. In the "propose" predicate frame, one slot answers the question "who proposed," and it is filled by the name "Wegener." A "theory" (or "theories") is something that has a topic. The objects that drift are "continents," as are the objects that are "permanent." Those who "begin" and who "find" are "geologists." Something has the attribute of "conventionality." Note that "conventional" is both an anchor for a proposition and an argument in it. It is not a predicate and an argument. It is not possible for a concept to be both a predicate and an argument of that predication. However, it is possible in certain circumstances for a concept to fill more than one slot in a predicate frame (e.g. "John loves himself" ⇒ "love [John, John]").

> **Wegener** proposed a **theory** of **continental** drift when **geologists** were beginning to find **conventional theories** of **continental** permanence inadequate.
>
> propose [**Wegener**]
> topic-of [**theory**]
> drift [**continents**]
> time:when [ ]
> begin [**geologists**]
> find [**geologists**]
> qualify [**conventional**]
> topic-of [**theories**]
> permanent [**continents**]
> inadequate [ ]

Embedded propositions: At this point, most of the words in the sentence have been used in a proposition. However, the ideas represented by the propositions are not complete. The rest of the arguments in the propositions representing this sentence's meaning are propositions. In this case, the embedded proposition is subordinated to the predicate. This does not necessarily mean that the embedded meaning is less important than the meaning of the whole, rather that the meaning of the embedded proposition is a part of the meaning expressed by the other idea. Again, the propositions that are embedded as arguments can be determined by asking a question which defines the argument slot. For example, what did Wegener propose? -- A theory about drifting continents. He did not simply propose a theory, but a particular theory about a particular topic.

> Wegener proposed a theory of continental drift when geologists were beginning to find conventional theories of continental permanence inadequate.
>
> time:when [propose [Wegener, topic-of [theory, drift [continents]]], begin [geologists, find [geologists, inadequate [qualify [topic-of [theories, permanent [continents]], conventional]]]]]

Writing a proposition list: When propositions are written as they are above, their interrelationships are transparent. However, it is difficult to access the individual ideas. Another way to write out the list requires each proposition to be assigned a label by which it can be referenced. Then embedded propositions are indicated by inserting their labels in the argument slots. Below ,each proposition is assigned a number prefixed by "P." This label is used for reference in an argument slot.

> P1 propose [Wegener, P2]
> P2 topic-of [theory, P3]
> P3 drift [continents]
> P4 time:when [P1, P5]
> P5 begin [geologists, P6]
> P6 find [geologists, P10]
> P7 qualify [P8, conventional]
> P8 topic-of [theories, P9]
> P9 permanent [continents]
> P10 inadequate [P7]

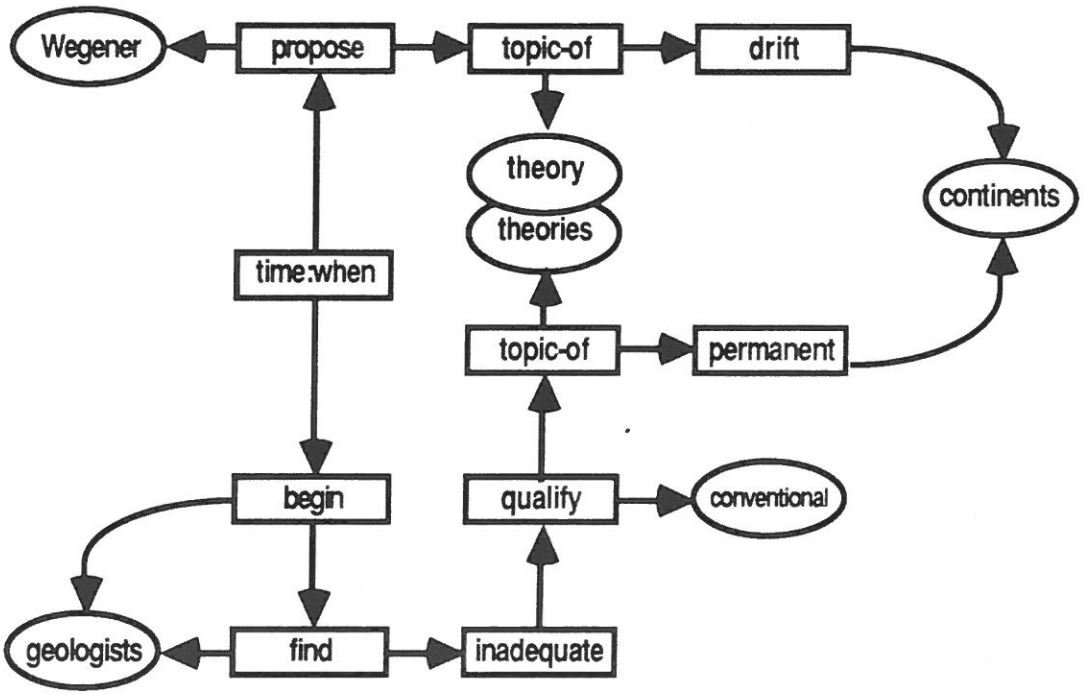
Of course, the exact way a proposition is written out is merely a formalism. The ideas that are embodied by the different formats are the same. People who want to use a propositional representation in a LISP program may choose to represent a proposition as a set of token words separated by spaces and enclosed by parentheses. The only way that the predicate is distinguished in this format is by its initial position within the list of concepts. Proposition numbers can be included as part of the same list or within a list containing the proposition list, as illustrated below.

```
(P1 (propose Wegener P2 ) )
(P2 (topic-of theory P3 ) )
(P3 (drift continents ) )
(P4 (time:when P1 P5 ) )
(P5 (begin geologists P6 ) )
(P6 (find geologists P10 ) )
(P7 (qualify P8 conventional ) )
(P8 (topic-of theories P9 ) )
(P9 (permanent continents ) )
(P10 (inadequate P7 ) )
```

Another way to write a list of propositions highlights the predicate frames and the structure of the ideas. The cases, or slot labels, are listed in parentheses after each element.

```
time:when
    propose (EVENT)
        Wegener (AGENT)
        topic-of (OBJECT)
            theory (OBJECT)
            drift (TOPIC)
                continents (OBJECT)
    begin (REFERENCE EVENT)
        geologists (AGENT)
        find (OBJECT)
            geologists (AGENT)
            inadequate (OBJECT)
                qualify (OBJECT)
                    topic-of (OBJECT)
                        theories (OBJECT)
                        permanent (TOPIC)
                            continents (OBJECT)
                    conventional (QUALIFICATION)
```

Finally, since a proposition representation is actually a network, not a list, a graphic representation can depict interrelationships in the network more effectively than a textual list. In the graph below, argument concepts are represented by ellipses; predicate concepts by rectangles; and argument slots as the directional arcs between a predicate and an argument concept. A proposition is the predicate (the rectangle), its case frame (all arcs exiting from the rectangle), and its arguments (all ellipses pointed to by the arcs).



Identifying Concepts

Concepts are the building blocks from which propositions are constructed. A concept is a distinct entity that can be manipulated in many contexts. Frequently the placement of a concept into the context of a specific propositional network will make it become more concrete. It may lose the full range of meaning, and take on the properties of a specific instance of the more abstract concept. For example, the general concept of "dog" includes a range of sizes, colors, and breeds. But in the context of the story of Lassie, "dog" will be used as an instance of the concept "dog," limited to a particular collie dog belonging to a particular boy named Timmy.

Concepts in turn are built from networks of propositions. A concept is an idea formed from the combination of propositions that define its characteristics, relationships, etc. Unlike propositions, concepts can not be asserted or negated. It does not make sense to negate the concept "dog," although it is perfectly acceptable to negate the proposition that Morris the Cat is a dog.

It is not necessary to decompose a concept into its components in order to use it. A complex concept appears to be used as easily or almost as easily as a simpler concept. Definitional information can become available if it becomes relevant. Precisely when and how far one decomposes a concept into component parts is an interesting debate (Fodor, Garret, Walker, & Parkes, 1980), but beyond the scope of the present discussion.

Words are labels used for manipulating concepts in natural language. A **type** is a word or set of words used as an entry point into a concept; the more abstract **token** for a concept is a handle which identifies a part of knowledge that the concept embodies. Different types for the same concept may focus on different aspects of the concept, and thus be to some extent equivalent in meaning. Similar concepts would share a high percentage of their propositional information.

Actions, states, or other relational concepts can become predicates in propositions. Objects and properties can be used to fill argument slots in propositions. The part of speech of a type is an imperfect indicator to whether it will be used as a predicate or an argument. Verbs are almost always predicates. Auxiliary verbs are usually not represented at all. Adjectives and adverbs are usually arguments, although there are a fair number of cases when they are predicates. These adjectives and adverbs predicate actions (e.g. "drifting continents") or states (e.g. "a knowing look").

Nouns are usually arguments, but there are cases where they are nominalized predicates. A nominalization occurs when a concept that is normally a dynamic action is treated as a static concept, as in the case of "drift" in "continental drift." Some words can be used in more than one part of speech. The context in which such words occur determines whether it will be a predicate or argument. An example of this is the word "tax." This type stands for several concepts, for example: the idea of a government imposing a levy for services rendered; the idea of making serious demands on one's resources; and the idea of money given to a government for its support. Ideally, a different token should be used to disambiguate these three concepts. However, pragmatically, this is only necessary when the concept would be ambiguous otherwise.

Prepositions are more complex. They serve many functions, some of which involve explicit or implicit meaning and others which indicate a case assignment for the word or words following it. Prepositions following verbs can change the meaning of the verb. In these cases, the token should be represented by a combination of the infinitive form of the verb and the preposition, or by another verb which denotes the meaning of the verb-preposition combination. An example of this is the verb "carry," which alone can denote the someone moves while supporting something, such as in the sentence "John carried the trophy." If we change the sentence by adding the preposition "off," "John carried off the trophy," the predicate concept is altered. Its meaning would now be better represented by "win [John, trophy]" rather than by "carry [John, trophy]."

Another example is the verb "introduce." In its most frequent sense, it means that someone presents someone or something to someone or some group. Another meaning is indicated when the verb is followed by the preposition "into." Now it means that something that was not present before was brought into and established in some place, as in "Merinos were introduced into Australia." The proposition expressed by this sentence would be better represented by "introduce-into [Merinos, Australia]" than by "introduce [Merinos, Australia]."

Frequently, prepositions contain a semantic component. Some common classes of meaning for prepositions are location (e.g. to, at, on, in), time (e.g. in, for, as, by), manner (e.g. with, by, in), purpose (to, for), cause (as, since). Note that in the examples given, there is quite a bit of overlap between the categories. The complete meaning is derived from the preposition and its context.

The preposition "of," the most common preposition in English, is the most extreme case of meaning that is derived from its context. Some examples would show the range of possibilities for the semantic aspect of this adaptable preposition.

the sheep of Australia	possess [Australia, sheep]
a wheel of solid wool	consist-of [wheel, qualify [wood, solid]]
the sin of envy	isa [sin, envy]
a bag of marbles	contain [bag, marbles]
the steppes of Asia	location:in [steppes, Asia]
the city of Paris	name-of [city, Paris]
John was one of the sailors.	element-of [sailors, John]
Alexander died of a fever.	cause [die [Alexander], possess [Alexander, fever]]
the sonnets of Shakespeare	write [Shakespeare, sonnets]
the easterly component of the satellite's velocity	velocity [satellite, east]

Predicates

The predicate is the pivotal concept of a proposition. It presents the nature of the relationship that is asserted and specifies its elements. Predicates can be actions, states, temporal or locative relationships, qualifications, coordinators among propositions, or any other relationship that is possible among a set of concepts or propositions.

Predicate frames: Each predicate has a frame of argument slots which are filled with particular values when an instance of the predicate is instantiated. A given predicate can only take as many arguments as are allowed for in its frame of argument slots. The modal number of arguments in a predicate frame is two, but this can vary between one and four or five. An exception to this is conjunction. Since this relationship predicates the conjoining of sets of concepts or propositions, there may be any number of arguments to this predicate.

The slots can be labelled to indicate the role taken by a concept or proposition that might fill it. A slot might have conditions that must be met by any potential candidates before they can be placed in it. For example, the predicate "give" has the frame "GIVE [AGENT, GOAL, OBJECT]." A candidate for the agent slot must be capable of performing the action "give." Thus in the sentence "The floor gave a marble to Bill," the word "floor" would not meet normal requirements for the Agent slot, and this sentence would be marked as nonsense at best.

Fillmore (1968) proposed a set of slot labels that have been used in propositional analysis and that have been fairly robust for predicates expressing actions or states. These slot names are **AGENT**, **OBJECT**, **EXPERIENCER** (sometimes called **PATIENT**), **GOAL**, **SOURCE**, and

INSTRUMENT. Many other slot labels are possible, for example, **EVENT**, **VALUE**, **QUALIFICATION**, **LOCATION**, **CATEGORY**, and **INSTANCE**.

Standard predicates: A set of standard predicate tokens has been developed to enable the underlying nature of equivalent types to be expressed. Some of these are predicate tokens that can stand of a variety of equivalent types. Others are predicate classes that enable the similarities of propositions within a class to be included as part of the propositions. By presenting a set of standard predicates, several things are accomplished. First, some standardization among proposition lists produced by different individuals is encouraged. Second, certain propositions become easier to specify quickly and easily. Finally, we believe that propositions with the same predicate or with predicates in the same predicate class will be treated in similar ways during comprehension.

Certain predicate relationships are basic to the nature of language. Often there are many ways to express them. Relationships like possession (**possess**), in the sense of belongingness, or negation (**negate**) are among the earliest learned predicates. Qualification of objects (**qualify**) or actions (**manner**) are basic predications. Labelling (**name-of**) is inherent in the nature of language. Other standard predicates that are used as tokens for simple and frequently encountered relationship are **consist-of**, to be used when the composition of an object is specified; **part-of**, which predicates that one of its arguments is a part of the other; **intensify**, to indicate that its argument should be stressed; and **hedge**, for fuzzing concepts or assertions.

Quantitative relationships are represented in several standard predicates. The predicate **quantify** is used if some quantitative measure can be taken of the argument in its Object slot, but its value is indefinite. If the exact number of the argument in the Object slot is given, the predicate **number-of** is used. If a exact extent of the argument is given, whether the extent is a proportion or a measure, the predicate used is **extent-of**. Sometimes a range of values is given along a particular dimension. In this case, **range-of** is used.

Some standard predicates are not predicates per se, but classes of predicates. The semantic relationships of members of these classes are held together by a common thrust. The dimensions of **time** and **location** (physical space) are two predicate classes that seem to have a special status in that actions, states, and objects in the physical world can be placed within them. Another class of predicates is **conjunction**, predicating that its arguments are joined together in some way. **Contrast** is an important class of predicates, allowing distinctions to be made between all sorts of things. Other classes are possible and perhaps will be added in the future.

Many of the higher order coordination functions are represented as standard predicates. In some systems of discourse analysis, these predicates may hold special status. Predications of **cause**, **purpose**, **condition**, and **concession** are relationships that glue together groups of propositions into segments that are important parts of narratives or expository prose. A related predicate, in that it is also frequently important to a discourse, is the **topic-of** predicate, which asserts that a particular discourse or other communication is about some topic.

Some standard predicates are referential in nature. That something exists (**exist**) is rarely asserted directly, but existential propositions are sometimes included in an analysis to be complete. The relationship of identity is expressed using the predicate token **reference**. If the relationship predicated is hierarchical such that one argument is a superordinate of the other, the predicate token **isa** is used.

It should be pointed out here that the introduction of standard tokens for some predicate relationships does not imply that all predicates could eventually be subsumed into some standard category. The class of all predicate concepts is an open class; new relationships can be invented or old ones transformed at any time. Without postulating semantic decomposition of predicates during comprehension, it is difficult to see how the class of predicates could be closed, and thus subject to an exhaustive accounting.

A Few Examples of Longer Analyses

Several longer passages are presented below. These examples are left for the reader to peruse.

Saint:

In the request to canonize the "Frontier Priest," John Newmann, bishop of Philadelphia in the nineteenth century, two miracles were attributed to him in this century. In 1923, Eva Benassi, dying from peritonitis, dramatically recovered after her nurse prayed to the bishop. In 1949, Kent Lenahan, hospitalized with two skull fractures, smashed bones, and a pierced lung after a traffic accident, rose from his deathbed and resumed a normal life after his mother prayed ardently to John Newmann.

P1 part-of [P2, P11]
P2 request [P3]
P3 canonize [P5]
P4 reference [John-Newmann, Frontier-Priest]
P5 isa [P4, bishop]
P6 location:in [P5, Philadelphia]
P7 time:in [P5, P8]
P8 number-of [century, nineteen]
P9 number-of [miracles, two]
P10 attribute [P9, John-Newmann]
P11 time:in [P10, P12]
P12 reference [century, current]
P13 time:in [P19, 1923]
P14 dying [Eva-Benassi]
P15 cause [P16, P14]
P16 possess [Eva-Benassi, peritonitis]
P17 manner [P18, dramatic]
P18 recover [Eva-Benassi, P15]
P19 time:after [P17, P21]
P20 possess [Eva-Benassi, nurse]
P21 pray [P20, bishop]
P22 time:in [P38, 1949]
P23 hospitalize [Kent-Lenahan]
P24 cause [P31, P23]
P25 possess [Kent-Lenahan, P29]
P26 number-of [P27, two]
P27 fracture [skull]
P28 smash [bones]
P29 conjunct:and [P26, P28, P30]
P30 pierce [lung]
P31 time:after [P23, P32]
P32 qualify [accident, traffic]
P33 rise [Kent-Lenahan, P34]
P34 possess [Kent-Lenahan, deathbed]
P35 conjunct:and [P34, P36]
P36 resume [Kent-Lenahan, P37]
P37 qualify [life, normal]
P38 time:after [P35, P41]
P39 possess [Kent-Lenahan, mother]
P40 pray [P39, John-Newmann]
P41 manner [P40, ardent]

Australian sheep:

Australia has more sheep than any other country. It produced about thirty percent of the world's wool. The part of Australia where the largest number of sheep are raised is in the region receiving ten to twenty inches of rain a year. This semiarid region covers fully one third of Australia. Most of Australia's sheep are in the eastern part of the continent, but some are also in the southwest.

Three fourths of Australia's sheep are pure merinos. They are popular because of the large amount of semiarid land. The Merino thrives on the grasses and low bushes which grow on semiarid plains. They are known for their heavy fleeces of fine quality wool which bring a high price from textile manufacturers. Merinos were introduced into Australia only five years after the first settlement and have been important ever since.

The remaining fourth of Australia's sheep are of mixed breeds which are raised both for meat and wool. The crossbreeds are raised especially in the moist regions where pastures are planted in good grass. The lambs are fattened and sold to meat packers. Some of the mutton is exported, but much is sold in markets in Australia. The wool, which is coarser than Merino wool, is sold to carpet manufacturers.

P1 possess [Australia, sheep]
P2 contrast:more [P1, P3]
P3 any-other [country, Australia]
P4 produce [Australia, P7]
P5 hedge [P6, about]
P6 number-of [percent, thirty]
P7 extent-of [P8, P5]
P8 possess [world, wool]
P9 part-of [Australia, part]
P10 location:where [P11, P9]
P11 contrast:most [P12]
P12 raise [sheep]
P13 reference [P10, P14]
P14 part-of [Australia, P18]
P15 receive [region, P17]
P16 range-of [inches, ten, twenty]
P17 extent-of [rain, P16]
P18 time:during [P15, year]
P19 reference [P20, P13]
P20 qualify [region, semiarid]
P21 cover [P20, P24]
P22 intensify [P23, fully]
P23 number-of [thirds, one]
P24 extent-of [Australia, P22]
P25 contrast:most [P27, P26]
P26 possess [Australia, sheep]
P27 location:in [sheep, P28]
P28 part-of [continent, east]
P29 but-also [P27, P31]
P30 quantify [P26, some]
P31 location:in [P30, P32]
P32 part-of [continent, southwest]
P33 number-of [fourths, three]
P34 extent-of [P35, P33]
P35 possess [Australia, sheep]
P36 consist-of [P34, P37]
P37 intensify [Merinos, pure]
P38 qualify [P37, popular]
P39 cause [P38, P40]
P40 quantify [P41, large]
P41 qualify [land, semiarid]
P42 thrive-on [Merinos, P43]
P43 conjunct:and [grass, P44]
P44 quantify [bushes, low]
P45 grow [P43]
P46 location:on [P45, P47]
P47 qualify [plains, semiarid]
P48 know-for [Merinos, P49]
P49 possess [Merinos, P51]
P50 quantify [fleeces, heavy]
P51 consist-of [P50, P52]
P52 qualify [wool, fine-quality]
P53 sell [P49, P54, P55]
P54 quantify [price, high]
P55 manufacture [textile]
P56 introduce-into [Merinos, Australia]
P57 intensify [P58, only]
P58 number-of [years, five]
P59 time:after [P56, P60, P57]
P60 qualify [settlement, first]
P61 conjunct:and [P59, P63]
P62 qualify [Merinos, important]
P63 time:ever-since [P62, P56]
P64 remaining [P66, P34]
P65 number-of [fourths, one]

P66 extent-of [P67, P65]
P67 possess [Australia, sheep]
P68 consist-of [P64, P69]
P69 mix [breeds]
P70 raise [P69]
P71 conjunct:both-and [meat, wool]
P72 purpose-of [P70, P71]
P73 raise [crossbreeds]
P74 intensify [P75, especially]
P75 location:in [P73, P76]
P76 qualify [regions, moist]
P77 location:where [P78, P76]
P78 plant [pastures, P79]
P79 qualify [grass, good]
P80 fatten [lambs]
P81 conjunct:and [P80, P82]
P82 sell [lambs, P83]
P83 pack [meat]
P84 quantify [mutton, some]
P85 export [P84]
P86 concession [P85, P89]
P87 quantify [mutton, much]
P88 sell [P87]
P89 location:in [P88, P90]
P90 location:in [markets, Australia]
P91 contrast:coarser [wool, P92]
P92 possess [Merinos, wool]
P93 sell [P91, P94]
P94 manufacture [carpet]

Wegener's Theory of Continental Drift:

Wegener proposed a theory of continental drift when geologists were beginning to find conventional theories of continental permanence inadequate. Previous continental drift theories held that some catastrophic event initiated continental displacement. In contrast, Wegener proposed that the same forces that produce great folded mountain ranges, displaced the continents. He presented evidence from such a range of sciences that his theory could not easily be ignored.

In the late Paleozoic era, according to Wegener's hypothesis, all the continents were part of one huge landmass, Pangaea, occupying half the Earth's surface. The other half was covered by the primeval Pacific Ocean. Wegener presented evidence that in the Jurassic period Pangaea began to break into fragments, and the weaker oceanic rock yielded to allow the continents to drift apart like icebergs in water.

The concept of continental drift first occurred to Wegener as he contemplated the apparent fit of the coastlines of the Atlantic Ocean. He tested this fit using the edges of the continental shelf as the boundary between continents and oceans. He postulated that the continental blocks retain the approximate outlines they acquired during the breakup of Pangaea. If the younger, Tertiary folded mountains could be flattened, the pieces could be reassembled into one large continent partially flooded by shallow seas.

P1 propose [Wegener, P2]
P2 topic-of [theory, P3]
P3 drift [continents]
P4 time:when [P1, P5]
P5 begin [geologists, P6]
P6 find [geologists, P10]
P7 qualify [P8, conventional]
P8 topic-of [theories, P9]
P9 qualify [continents, permanent]
P10 qualify [P7, inadequate]
P11 contrast:previous [P13, P1]
P12 drift [continents]
P13 topic-of [theories, P12]
P14 hold [P11, P17]
P15 qualify [P16, some]
P16 qualify [event, catastrophic]
P17 initiate [P15, P18]
P18 displace [continents]
P19 contrast:in-contrast [P14, P20]
P20 propose [Wegener, P21]
P21 reference [P22, P26]
P22 cause [forces, P24]
P23 qualify [P25, great]
P24 fold [P23]
P25 consist-of [ranges, mountains]
P26 displace [forces, continents]
P27 present [Wegener, P28]
P28 derive-from [evidence, P30]
P29 cause [P27, P33]
P30 quantify [sciences, many]
P31 possess [Wegener, theory]
P32 could [P34]
P33 negate [P32]
P34 manner [P35, easy]
P35 ignore [P31]
P36 time:in [P45, P37]
P37 qualify [P38, late]
P38 name-of [era, Paleozoic]
P39 hypothesize [Wegener, P36]
P40 quantify [continents, all]
P41 part-of [P40, P42]
P42 number-of [P43, one]
P43 quantify [landmass, huge]
P44 name-of [P42, Pangaea]
P45 occupy [P44, P46]
P46 extent-of [P47, half]
P47 possess [earth, surface]
P48 contrast:other [P49, P46]
P49 extent-of [P47, half]
P50 cover [P51, P48]
P51 qualify [P52, primeval]
P52 name-of [ocean, Pacific]
P53 present [Wegener, P54]
P54 show [evidence, P55]
P55 time:in [P59, P56]
P56 name-of [period, Jurassic]
P57 begin [Pangaea, P58]
P58 breakup [Pangaea, fragments]
P59 conjunct:and [P57, P63]

P60 contrast:weaker [P61, continents]
P61 qualify [rock, oceanic]
P62 yield [P60]
P63 allow [P62, P65]
P64 drift [continents]
P65 manner [P64, apart]
P66 contrast:like [P64, P67]
P67 drift [icebergs]
P68 location:in [P67, water]
P69 topic-of [P70, concept]
P70 drift [continents]
P71 qualify [P72, first]
P72 think [Wegener, P69]
P73 time:as [P71, P74]
P74 contemplate [Wegener, P75]
P75 manner [P76, apparent]
P76 fit [P77, P77]
P77 possess [P78, coastlines]
P78 name-of [ocean, Atlantic]
P79 test [Wegener, P76]
P80 use [Wegener, P81, P84]
P81 possess [P82, edges]
P82 possess [continental, shelf]
P83 purpose-of [P80, P79]
P84 location:boundary-between [continents, oceans]
P85 postulate [Wegener, P87]
P86 qualify [blocks, continental]
P87 retain [P86, P90]
P88 qualify [outlines, approximate]
P89 acquire [P86, P88]
P90 time:during [P89, P91]
P91 breakup [Pangaea]
P92 condition:if [P96, P98]
P93 contrast:younger [P94, P90]
P94 qualify [P95, Tertiary]
P95 fold [mountains]
P96 could [P97]
P97 flatten [P93]
P98 could [P99]
P99 reassemble [P102, pieces]
P100 number-of [P101, one]
P101 quantify [continent, large]
P102 extent-of [P103, partial]
P103 flood [P104, P100]
P104 quantify [seas, shallow]

Sonnet from Romeo and Juliet:

Romeo: If I profane with my unworthiest hand
This holy shrine, the gentle sin is this,
My lips, two blushing pilgrims, ready stand
To smooth that rough touch with a tender kiss.

Juliet: Good pilgrim, you do wrong your hand too much,
Which mannerly devotion shows in this:
For saints have hands that pilgrims' hands do touch,
And palm to palm is holy palmers' kiss.

Romeo: Have not saints lips, and holy palmers too?

Juliet: Ay, pilgrim, lips that they must use in prayer.

Romeo: O then, dear saint, let lips do what hands do,
they pray. Grant thou, lest faith turn to despair.

Juliet: Saints do not move, though grant for prayers' sake.

Romeo: Then move not while my prayer's effect I take.

P1 condition [P2, P15
P2 profane [Romeo, P7]
P3 use [Romeo, P5, P2]
P4 part-of [Romeo, hand]
P5 contrast:unworthiest [P4]
P6 reference [P7, Juliet]
P7 qualify [shrine, holy]
P8 qualify [P9, gentle]
P9 sin [Romeo, P2]
P10 reference [P8, P2]
P11 part-of [Romeo, lips]
P12 reference [P11, P13]
P13 quantify [P14, two]
P14 blush [pilgrims]
P15 stand-ready [P12, P16]
P16 smooth [P12, P17]
P17 reference [P18, P2]
P18 qualify [P19, rough]
P19 touch [Romeo, P7]
P20 qualify [P16, P21]
P21 qualify [P22, tender]
P22 kiss [P12, P7]
P23 qualify [pilgrim, good]
P24 reference [pilgrim, P23]
P25 wrong [Romeo, P26]
P26 part-of [Romeo, hand]
P27 intensify [P25, too-much]
P28 reference [P27, P29]
P29 qualify [devotion, proper]
P30 show [P35, P28]
P31 cause [P35, P27]
P32 part-of [saints, hands]
P33 part-of [pilgrims, hands]
P34 touch [P33, P32]
P35 conjunct:and [P34, P37]
P36 location:adjacent [palm, palm]
P37 reference [P36, P39]
P38 qualify [palmers, holy]
P39 kiss [P38]
P40 part-of [saints, lips]
P41 query [Romeo, Juliet, P40]
P42 conjunct:and [P41, P43]
P43 query [Romeo, Juliet, P46]
P44 qualify [palmers, holy]
P45 part-of [P44, lips]
P46 conjunct:too [P45, P40]
P47 affirm [Juliet, P42]
P48 reference [pilgrim, Romeo]
P49 conjunct [saints, P44]
P50 must [P51]
P51 use [P49, lips, P52]
P52 pray [P49]
P53 condition [P50, P56]
P54 qualify [saint, dear]

P55 reference [P54, Juliet]
P56 allow [Juliet, P58]
P57 do [lips]
P58 contrast:same [P57, P59]
P59 do [hands]
P60 pray [lips]
P61 grant [Juliet, P60]
P62 conjunct:lest [P61, P63]
P63 change [faith, despair]
P64 negate [P65]
P65 initiate [saints]
P66 concession [P64, P68]
P67 grant [saints]
P68 purpose-of [P67, P69]
P69 possess [someone, sake]
P70 pray [someone]
P71 condition [P68, P74]
P72 move [Juliet]
P73 negate [P72]
P74 time:while [P73, P76]
P75 pray [Romeo, effect]
P76 take [Romeo, P75]

Satellite Ground Traces:

On a flat map of the earth (Mercator Projection) satellite ground traces appear to have different shapes than on a sphere. The ground trace for an object in an inclined circular or elliptical orbit appears as a sinusoidal trace with North - South limits equal to the inclination of the orbital plane.

The fun starts when you start the earth rotating. When you do this, visualizing a satellite's ground trace becomes more complex. As we said earlier, a point on the equator moves from west to east more rapidly than do points north and south of the equator. Satellites in a circular orbit travel at a constant speed. But when orbits are inclined to the equator, the component of satellite velocity which is effective in an easterly or westerly direction varies continuously throughout the orbital trace.

As the satellite crosses the equator, its easterly or westerly component of velocity is its instantaneous total velocity times the cosine of its angle of inclination. When it is at the most northerly or southerly portion of its orbit, its easterly or westerly component is equal to its total instantaneous velocity. To put it in simpler terms, a satellite in a circular or nearly circular orbit is not moving as fast in an easterly or westerly direction at the equator as it is at its most northerly or southerly point.

In elliptical orbits only the horizontal (to the Earth's surface) velocity component contributes to the ground trace. To further complicate things, the ground trace is changed because the inertial or absolute speed of the satellite varies throughout the elliptical path.

Fortunately, most of the orbits, with a few special exceptions, we deal with are nearly circular and have a fairly low altitude (within 400 to 600 NM). This eases the problem considerably.

Because the ground trace of a satellite is dependent upon the relative motion between the satellite and the earth, the visualization of ground tracks becomes quite complicated. Earth rotation causes each successive track of a satellite in a near earth orbit (400 NM or less) to cross the equator at a point which is west of the preceding track. This phenomena is referred to as "regression of the nodes."

P1 location:on [P6, P3]
P2 qualify [map, flat]
P3 possess [earth, P2]
P4 qualify [projection, Mercator]
P5 name-of [P3, P4]
P6 trace [satellite, ground]
P7 appear [P6, P8]
P8 contrast:different [shapes, P1, P9]
P9 location:on [P6, sphere]
P10 trace [P11, ground]
P11 location:in [object, P12]
P12 incline [P14]
P13 qualify [orbit, circular]
P14 conjunct:or [P13, P15]
P15 qualify [orbit, elliptical]
P16 appear [P10, P17]
P17 qualify [P18, sinusoidal]
P18 trace [ ]
P19 extent-of [P17, P20]
P20 range-of [north-south, P21, P21]
P21 incline [P22]
P22 possess [orbit, plane]
P23 start [fun]
P24 time:when [P23, P25]
P25 start [you, P26]
P26 rotate [earth]
P27 time:when [P30, P25]
P28 visualize [you, P29]
P29 trace [satellite, ground]
P30 become [P28, P31]
P31 contrast:more [complex, P28]
P32 say [we, P36]
P33 time:earlier [P32]
P34 location:on [point, equator]
P35 move [P34, west, east]
P36 contrast:more [speed, P35, P37]
P37 move [P39]

P38 location:north [points, equator]
P39 conjunct:and [P38, P40]
P40 location:south [points, equator]
P41 location:in [satellites, P42]
P42 qualify [orbit, circular]
P43 move [P41]
P44 rate-of [P43, constant]
P45 concession [P44, P46]
P46 time:when [P54, P47]
P47 incline [orbits, equator]
P48 velocity [satellite, P49]
P49 operate [component, P51]
P50 qualify [direction, east]
P51 conjunct:or [P50, P52]
P52 qualify [direction, west]
P53 vary [P48]
P54 time:frequency [P55, continuous]
P55 time:during [P53, P56]
P56 trace [orbit]
P57 time:when [P58]
P58 cross [satellite, equator]
P59 qualify [component, east]
P60 conjunct:or [P59, P61]
P61 qualify [component, west]
P62 velocity [satellite, P60]
P63 extent-of [P62, P67]
P64 time:during [P66, instant]
P65 quantify [component, total]
P66 velocity [satellite, P65]
P67 multiply [P64, P68]
P68 cosine [P69]
P69 angle-of [P70]
P70 incline [satellite]
P71 time:when [P83, P72]
P72 location:at [satellite, P76]
P73 contrast:most [P75]
P74 contrast:most [P77]
P75 location:toward [P78, north]
P76 conjunct:or [P73, P74]
P77 location:toward [P78, south]
P78 part-of [orbit]
P79 possess [satellite, P81]
P80 qualify [component, east]
P81 conjunct:or [P80, P82]
P82 qualify [component, west]
P83 extent-of [P79, P85]
P84 quantify [component, total]
P85 time:during [P86, instant]
P86 velocity [satellite, P84]
P87 purpose-of [P88, P89]
P88 restate [P71, P95]
P89 simplify [P71]
P90 location:in [satellite, P92]
P91 qualify [orbit, circular]
P92 conjunct:or [P91, P93]
P93 hedge [P94, nearly]
P94 qualify [orbit, circular]
P95 contrast:less [speed, P97, P103]
P96 move [P90]
P97 qualify [P96, P99]
P98 qualify [direction, east]
P99 conjunct:or [P98, P100]
P100 qualify [direction, west]
P101 location:at [P97, equator]
P102 move [P90]
P103 location:at [P102, P107]
P104 contrast:most [P106]
P105 contrast:most [P108]
P106 location:toward [point, north]
P107 conjunct:or [P104, P105]
P108 location:toward [point, south]
P109 location:in [P111, P110]
P110 qualify [orbits, elliptical]
P111 intensify [P115, only]
P112 horizontal [component, P113]
P113 part-of [earth, surface]
P114 velocity [P112]
P115 contribute [P112, P116]
P116 trace [ground]
P117 intensify [P118, further]
P118 complicate [P121, things]
P119 trace [ground]
P120 change [P119]
P121 cause [P126, P120]
P122 qualify [P125, inertial]
P123 conjunct:or [P122, P124]
P124 qualify [P125, absolute]
P125 possess [satellite, speed]
P126 vary [P123]
P127 time:during [P126, P128]
P128 qualify [path, elliptical]
P129 qualify [P130, fortunate]
P130 contrast:most [P136, P133]
P131 quantify [P132, few]
P132 qualify [exceptions, special]
P133 deal [we, orbits]
P134 hedge [P135, nearly]
P135 qualify [orbits, circular]
P136 conjunct:and [P134, P137]
P137 possess [orbits, P138]
P138 hedge [P139, fairly]

P139 quantify [altitude, low]
P140 location:within [altitude, P141]
P141 range-of [NM, 400, 600]
P142 ease [P136, problem]
P143 quantify [P142, considerable]
P144 cause [P146, P152]
P145 trace [satellite, ground]
P146 depend [P145, P147]
P147 relative [P148, P149]
P148 move [satellite]
P149 move [earth]
P150 visualize [P151]
P151 trace [ground]
P152 become [P150, P153]
P153 hedge [P154, quite]
P154 complicate [P150]
P155 rotate [earth]
P156 cause [P155]
P157 each [P158]
P158 successive [P159]
P159 trace [P160]
P160 location:in [satellite, P161]
P161 location:near [orbit, earth]
P162 range-of [NM, 400, P163]
P163 less [400]
P164 cross [P157, equator]
P165 location:at [P164, P166]
P166 location:west [point, P167]
P167 precede [P168]
P168 trace [ ]
P169 refer [phenomena, P170]
P170 regress [nodes]

THE PROPOSITIONAL ANALYSIS SYSTEM

The Propositional Analysis System is a program that can be used to analyze text into a propositional representation. It runs on IBM Personal Computers, under the PC-DOS or MS-DOS operating system, version 3.0 or later. The Propositional Analysis System should run on any IBM compatibles and may run with lower versions of DOS.The system must be configured as an ANSI device. On the IBM family of personal computers, this is done by including a file, CONFIG.SYS, on the root directory of the drive from which it boots. This file must contain a line "DEVICE=[d:][path]ANSI.SYS", where the file ANSI.SYS is located as indicated by the [d:][path]. This file is included on the DOS diskette.

To run the program, put the diskette with the file PROP.EXE in drive a: and type "prop<CR>."

A passage to be analyzed should be on a file. Any special control characters, such as formatting characters, will be ignored. Currently, the maximum number of characters allowed in a text file is 6144. This is roughly related to the number of bytes in a file. Any tab characters will be expanded to spaces. Any sequence of more than one blank line will be reduced to a single blank line. Leading and trailing blank lines will be deleted. Lines with more than eighty characters will be split so that no line is greater than eighty characters. No words greater than 50 characters are allowed. At the beginning of the first analysis session for a passage, a text file containing the passage is read by the Propositional Analysis System, which processes it into an internal format. If there are more than 6144 processed characters, the remaining information in the text file is ignored. After this, the passage may be analyzed.

Each session can be saved in a system file. This system file allows the user to stop an analysis at any given point, save the information about the current state of the analysis in a system file, and resume at some later time at the same point. Once an analysis has begun, the text may not be edited. If changes must be made to a text, they must be made outside of the Propositional Analysis System. The analysis of the text must be redone for the changed text. No plans exist at this time to allow a text to be edited within the Propositional Analysis System.

The Propositional Analysis System is a menu based system. It uses three display windows: the text window, the proposition window, and the command window.

The command window consists of the bottom three lines of the display. The menu of options is displayed here, along with any error or information messages. Menu options are chosen as indicated below in SELECTING A MENU CHOICE. At any given point, a small set of explicit options are available on a "menu" of choices. At times, a selection displayed may not be valid. If so, a message will appear explaining why that selection is inappropriate. You may then make another selection.

The text window appears at the top of the screen and displays eight lines of the text. The text cursor is a reverse video area within the text window which indicates the current word in the text. The cursor moves by whole words within this window as described below in the MOVE TO TEXT WORD MENU. Currently, words are defined according to rather simplistic rules about word contents and boundaries. It is not possible to alter the way the words are defined in the text window at this time.

The proposition window contains twelve propositions in the center section of the display. The proposition cursor is a reverse video area within the proposition window which indicates the current proposition. This cursor includes an entire proposition. Access to the parts of a proposition are possible through the EDIT PROPOSITION MENU. The proposition cursor may be moved as described in the MOVE TO PROPOSITION MENU.

Selecting a Menu Choice

To select a menu choice, press the key indicated in highlighted letters to the left of the desired choice. These keys are usually function keys; except in the case of ESC (the ESCAPE key). In general, ESC is reserved for exiting a menu. In cases where it is desired to undo a selection in progress, pressing ESC will generally undo the selection and return to the parent menu. For example, if a user was in the middle of the MULTIPLE WORD MENU, and decided that a multiple word concept was not in order, pressing ESC will cause the multiple word concept to be undone and the user to be returned to the previous menu.

The menu choice that is in reverse video is the current default menu choice. By pressing ENTER, this selection is chosen. This allows operations which are fairly typical to be entered faster.

Each menu choice contains a single capital letter. Typing that letter will allow the user to move the default choice to the nearest item with that capital letter. A letter that is not represented among the choices will have no effect. It is still necessary to type an ENTER to select the item. Currently, this is an inefficient way to use the menu, since use of the function keys is more direct.

Anchoring Propositions

At times the token for a predicate or an argument is not represented directly in the text. When the predicate is not directly represented, a point in the text to which the proposition is to be anchored must be selected. One reasonable criteria to use in selecting an anchor is to determine the point in the text at which the predicate concept of the proposition is invoked. The anchor point has practical implications as well: the location of a proposition in a proposition list is determined by its anchor in the text.

Default Extensions for File Names

Certain type of files used or created by the Propositional Analysis System have default extensions as follows: "SYS" for system files, "PRP" for proposition files, and "SUM" for summary analysis files. Nothing is assumed about the extension of the text file. Other types of files that might be created have no default extension at this time, although they might in the future.

In addition to defaults for extensions, the Propositional Analysis System assumes that the default name for a system, proposition, or summary file is the same as that of the original text or system file for that analysis. It is possible to override this default by editing the suggested file name. Although it is possible to change these defaults, it is recommended that the default extensions be used if at all possible.

The Available Memory Indicator

At times, a reverse video number may appear in a box at the lower right hand corner of the screen. This number represents the number of bytes in memory available to be allocated for propositions. If this number gets very small (say below 3,000), it may not be possible to read in a system file from a session. The only recourse at the moment is to try to break a large text into a series of smaller ones. A text of 500 words or less should not be a problem. A solution to this limitation will be available in later versions of the Propositional Analysis System. It will include a way to combine several system files into one. In the meantime, if the available memory indicator appears on the screen, save a system file from the session. Do a separate analysis on the text that has not yet been analyzed.

The Monitor File

When using the Propositional Analysis System, a file is created which contains all user keystrokes. This file, called "MONITOR.DAT," is saved at regular intervals. If something goes wrong, such as a program crash, or if the program is exited without saving a system file, the Monitor file can be used to recover most, if not all, of the session. If the system file read in for a particular session was altered during the session, recovery may not be possible. If the original system file for the session exists as a backup file (it will have a "~YS" extension), this file should be renamed so that it has the "SYS" extension and used in the recovery attempt. However, if it does not, or if the system file was repeatedly saved during the session, DO NOT ATTEMPT A RECOVERY.

To recover from a monitor file, when invoking the program, include the argument "-r" (e.g. "prop -r"). If you have renamed a monitor file for future recovery, include the argument "-r<filename>", where <filename> is the renamed file (e.g. "prop -rjunk.dat" if the recovery file is named junk.dat). A backup for the monitor file will exist in most occasions. If it is necessary to recover from the previous session, include the argument "-rMONITOR.~AT."

MAIN MENU

F1	Read information	Go to READ INFORMATION MENU to read in raw text or a previous session saved as a system file.
F2	Analyze text	Go to ANALYZE TEXT MENU to do a propositional analysis of a previously entered text. This may not be selected unless a text is available already using the READ INFORMATION MENU.
F3	Print	Go to PRINT INFORMATION MENU to select and print certain types of information. This may not be selected unless a text is available already using the READ INFORMATION MENU. Note: you must have a printer connected to the computer to use this menu option.
F4	Save information	Go to SAVE INFORMATION MENU to select and save certain types of information. This may not be selected unless a text is available already using the READ INFORMATION MENU.
F5	Use utilities	The selections under this menu choice are under development.
ESC	eXit	Exit the propositional analysis program and return to the operating system. If changes have been made to an analysis, the program will ask if the changed analysis should be saved. An answer of 'y' will save the analysis in a system file; an answer of 'n' will cause the program to exit without saving.

READ INFORMATION MENU

F1 read Text file

Read in a text file containing the passage to be analyzed. This file must already exist; there is no way to enter text from within the propositional analysis program or to change the text once it has been entered into the propositional system. The program will prompt for the name of the file. If this file does not exist, the program will indicate this and prompt for another a file name. The previously entered file name will appear in reverse video. This may be edited (see EDITING INFORMATION MENU) or erased.

A file must be less than 6144 characters, including at the end of each line. Currently it is recommended that text files be kept considerably shorter than this. No word may be longer than 50 characters. Lines longer than 80 characters will be divided.

F2 read System file

Read in a system file containing information about an analysis from a previous session with the Propositional Analysis System. The program will prompt for the name of the file. If this file does not exist, the program will indicate this and prompt for another file name. The previously entered file name will appear in reverse video. This file name may be edited (see EDITING INFORMATION MENU) or erased.

A system file is not in ASCII format. It cannot be edited, altered, or created, except through the use of the program. When a file that is not a legitimate system file is input as a system file, the most likely result is a fatal error, **RDSYS_DICT Allocation Error: If changes have been made, save them.** (see PROGRAM ERROR MESSAGES - ALL FATAL). In the event that this message is received when reading a file, exit without saving.

ESC eXit

Return to the MAIN MENU.

ANALYZE TEXT MENU

F1	select Predicate	Select a predicate (see SELECT PREDICATE MENU).
F2	select Argument	Select an argument for the current proposition (see SELECT ARGUMENT MENU). Arguments can be selected for a proposition only after its predicate has been identified.
F3	Delete prop	Delete a proposition (see DELETE PROPOSITION MENU). At least one proposition must have been identified before this can be selected.
F4	Edit prop	Edit a proposition (see EDIT PROPOSITION MENU). At least one proposition must have been identified before this can be selected.
F5	Move to prop	Move to a proposition (see MOVE TO PROPOSITION MENU). At least two propositions must have been identified before this can be selected.
F6	Move to word	Move to a word in the text (see MOVE TO TEXT WORD MENU).
ESC	eXit	Return to the MAIN MENU.

SELECT PREDICATE MENU

F1	Move to predicate	Move to a word in the text that is to be the predicate for a new proposition.
F2	Select this word	Select the current text word to be the predicate for a new proposition.
F3	select > 1 Word	Select a set of words in the text to be the predicate for a new proposition. This set of words will begin with the current text word and move forward in the text from that point (see MULTIPLE WORD MENU). The proposition will be anchored to the current word in the text (see ANCHORING PROPOSITIONS).
F4	Enter predicate	Enter a string that is to be the predicate for a new proposition (see EDITING INFORMATION). The proposition must be anchored to a word in the text (see ANCHORING PROPOSITIONS).
F5	select Standard	Select a standard predicate to be the predicate for a new proposition (see SELECT STANDARD PREDICATE MENU). The predicate must be anchored to a word in the text (see ANCHORING PROPOSITIONS).
ESC	eXit	Return to the ANALYZE TEXT MENU.

SELECT STANDARD PREDICATE MENUS

The list of standard predicates below is not necessarily complete. Suggestions as to further candidates for this list are welcome. By presenting a set of standard predicates, several things are accomplished. First, some standardization among proposition lists produced by different individuals is encouraged. Second, certain propositions become easier to specify quickly and easily. Most standard predicates are tokens to be used for a variety of equivalent types. Some standard predicates are actually predicate classes.

When a standard predicate is a predicate class, it will require further specification. At this time, there are four predicates classes in the Propositional Analysis System: "time," "location," "contrast," and "conjunct." The particular type is added to the class, so that the predicate is placed in its appropriate context. This is particularly important when the type is a word or set of words that may be used to indicate the different semantic relationships, depending on the class.

Currently, the program will allow a single text word, the anchor (see ANCHORING PROPOSITIONS), to be combined with predicate classes. To indicate a multiple word predicate in a predicate class, it is necessary to edit the combined predicate further (see EDIT PROPOSITION MENU).

First Menu

F1	qualify	The predicate "qualify" takes two arguments: an OBJECT case and a QUALIFICATION case. It indicates that the first argument is modified in some way by the second, or possesses the quality, characteristic, or property expressed by the second.
F2	possess	The predicate "possess" takes two arguments: an AGENT case and an OBJECT case. It indicates that the first argument owns or possesses the second, or that there is some way in which the second argument "belongs" to the first (e.g., "Australia's sheep").
F3	time	The predicate "time" is a predicate class. It indicates the placement of its first argument along the temporal dimension. For example, the temporal relationship of "on a particular day" (e.g., "on Wednesday") could be indicated by "time:on". The first argument must be a proposition which entails an event, or a concept that stands for an event. The second argument depends on the exact nature of the temporal predicate. It may be a proposition that entails another event, an absolute temporal point, such as "yesterday," or a temporal frequency. Certain predicates may have a third argument, indicating elapsed time span or a second temporal reference point.
F4	location	The predicate "location" is a predicate class. It indicates the placement of its first argument along the spatial dimension. For example, the spatial relationship of "on top of" (e.g., "on a desk") could be indicated by "location:on". The first argument of a locative predicate is an OBJECT case. The second is a REFERENCE POINT case. Both arguments must be capable of existing in spatial coordinates. A third argument may be necessary in some cases, such as when there are two reference points or when a distance between the reference points is given.
F5	manner	The predicate "manner" takes two arguments: an EVENT case and a MANNER case. The first argument must be a proposition entailing an event or action. The second must further specify the way in which the first is done or happens. Generally, the second argument would answer a question as to how the first is done.
F6	isa	The predicate "isa" takes two arguments: an INSTANCE case and a CATEGORY case. It indicates that the first argument is a member of a hierarchically superordinate category.
F7	reference	The predicate "reference" takes two arguments: a CURRENT REFERENCE case and a PAST REFERENCE case. It indicates that the two arguments have the same referent.
F8	Next menu	Move to second menu of standard predicates.
ESC	eXit	Return to the SELECT PREDICATE MENU without selecting a standard predicate.

Second Menu

F1	quantify	The predicate "quantify" takes two arguments: an OBJECT case and a QUANTIFICATION case. The quantity specified is indefinite (e.g., "some" or "a bit"). The dimension quantified may be magnitude, size, volume, area or length.
F2	number-of	The predicate "number-of" takes two arguments: an OBJECT case and a QUANTIFICATION case. The quantity specified is definite.
F3	extent-of	The predicate "extent-of" takes two arguments: an OBJECT case and a PROPORTION case. The quantity specified is a proportion (e.g., "two thirds of the country") or a degree (e.g., "a sixty foot cliff").
F4	part-of	The predicate "part-of" takes two arguments: a PART case and a WHOLE case. The relation indicated is that the first argument is a clearly defined part of the second (e.g., "the cow's tail" or the wheel of the car").
F5	consist-of	The predicate "consist-of" takes two arguments: an OBJECT case and a COMPOSITION case. The first argument is composed of the second.
F6	exist	The predicate "exist" takes a single argument: an OBJECT case. It predicates an object's existence.
F7	name-of	The predicate "name-of" takes two arguments: an OBJECT case and a NAME case. The object specified in the first argument has a proper name, which is generally the second argument (e.g., "the boy, John Doe"), although it is sometimes a combination of the two (e.g., "Lake Michigan" or "San Francisco Bay").
F8	Next menu	Move to third menu of standard predicates.
ESC	eXit	Return to the SELECT PREDICATE MENU without selecting a standard predicate.

Third Menu

F1	cause	The predicate "cause" takes two arguments: a CAUSE case and an RESULT case. The two arguments are propositions which entail events, actions or states.
F2	purpose-of	The predicate "purpose-of" takes two arguments: an AGENT case and a RESULT case. The first argument has as its intention the bringing about of the second argument. Both arguments are propositions which entail events actions or states, although the first argument is rarely a state.
F3	contrast	The predicate "contrast" is a predicate class which takes two arguments: a CONTRAST case and a STANDARD case. Its two arguments are contrasted along a particular dimension, as indicated by the second part of the contrast predicate (i.e., the anchor word from the text). The first argument is held in contrast to the standard of the second. Its arguments are propositions or concepts which can be expanded into a proposition that is parallel to the other argument.
F4	conjunct	The predicate "conjunct" is a predicate class, which includes relationships of conjunction or disjunction. Its second part may differ in intensity from a loose, informal conjunction to a stressed one, such as "both a and b." It takes two or more OBJECT cases.
F5	concession	The predicate "concession" takes two cases: an ASSERTION case and a CONCESSION case. Both arguments are propositions. The first argument sets up a point. The second argument yields an explicit or implicit disputed contention arising from the first argument.
F6	negate	The predicate "negate" takes a single argument, a proposition which is negated.
F7	topic-of	The predicate "topic-of" takes two arguments: a TOPIC case and a DISCOURSE case. This predicate indicates macro-level information. The discourse may be any level of discourse, from spoken to written to nonverbal communication.
F8	Next menu	Move to first menu of standard predicates.
ESC	eXit	Return to the SELECT PREDICATE MENU without selecting a standard predicate.

SELECT ARGUMENT MENU

F1	Move to argument	Move to a word in the text that is to be an argument for the current proposition.
F2	Select this word	Select the current text word to be an argument for a new proposition. The argument will be inserted in the final argument position, if other arguments already exist, unless another position has been indicated (see "Position argument" below).
F3	select > 1 Word	Select a set of words in the text to be an argument for the current proposition. This set of words will begin with the current text word and move forward in the text from that point (see MULTIPLE WORD MENU).
F4	Enter argument	Enter a string that is to be an argument for the current proposition (see EDITING INFORMATION).
F5	embed Prop	Select a proposition to be embedded as an argument of the current proposition. At least one other proposition must exist to make this selection.
F6	Position argument	Indicate the position within the set of arguments for the current proposition that is to be the position for the next argument selected. If this is not chosen, the argument will be inserted as the last argument.
F7	Move to prop	Move to a proposition for which an argument is to be embedded. At least one other proposition must exist to make this selection.
ESC	eXit	Return to the ANALYZE TEXT MENU.

MULTIPLE WORD MENU

F1	Include word	Include the current word (in reverse video) in the multiple word concept and move to the next word.
F2	Exclude word	Do not include the current word (in reverse video) in the multiple word concept but move to the next word.
F3	Last word	Include the current word (in reverse video) in the multiple word concept and return this multiple word concept to the previous menu. This could be the SELECT PREDICATE MENU, SELECT ARGUMENT MENU, or the REPLACE PREDICATE MENU.
F4	Remove last word	Remove the previous word (in reverse video) from the multiple word concept, if it was included. Move to the previous word.
F5	Cancel multiword	Return to the previous menu without a multiple word concept. The previous menu could be the SELECT PREDICATE MENU, SELECT ARGUMENT MENU, or the REPLACE PREDICATE MENU.
ESC	eXit	Return the multiple word concept, if there is one, to the previous menu. The previous menu could be the SELECT PREDICATE MENU, SELECT ARGUMENT MENU, or the REPLACE PREDICATE MENU.

EMBED PROPOSITION MENU

F1	Move to prop	Move the proposition cursor to the proposition that is to be embedded in the current proposition.
F2	select prop Num	Select a proposition by number to be embedded in the current proposition. The proposition will actually be embedded with confirmation by choosing the "Select this prop" option after "select prop Num."
F3	Select this prop	Select this proposition to be embedded in the current proposition.
ESC	eXit	Return to the previous menu without embedding any proposition. The previous menu could be the SELECT PREDICATE MENU or the SELECT ARGUMENT MENU.

DELETE PROPOSITION MENU

F1	Delete this prop	Delete the current proposition (in reverse video). If this proposition is embedded in other propositions, the program will indicate this. If the user still wants to delete it, any instances of the embedded proposition must be replaced or removed. If the user does not choose to do so, the proposition will not be deleted.
F2	delete Number	Indicate a proposition by number that is to be deleted. Deletion will not occur without confirmation by choosing the "Delete this prop" option after "delete Number."
F3	Move to prop	Move to a proposition that is to be deleted. Deletion will not occur without confirmation by choosing the "Delete this prop" option after "Move to prop."
ESC	eXit	Return to the ANALYZE TEXT MENU.

EDIT PROPOSITION MENU

F1	edit Predicate	Edit the predicate of the current proposition (see EDITING INFORMATION MENU).
F2	edit Argument	Edit the argument of the current proposition (see EDITING INFORMATION MENU).
F3	Replace predicate	Replace the predicate of the current proposition (see REPLACE PREDICATE MENU).
F4	Replace argument	Replace an argument from the current proposition (see REPLACE ARGUMENT MENU).
F5	Move to prop	Move to a proposition that is to be edited (see MOVE TO PROPOSITION MENU).
F6	Remove argument	Remove an argument from the current proposition. The program will request which argument should be removed. The default (chosen by ENTER) is the last argument in the proposition. An ESCAPE will return to the EDIT PROPOSITION MENU without removing an argument.
ESC	eXit	Return to the ANALYZE TEXT MENU.

REPLACE PREDICATE MENU

F1	Move to new word	Move to a word in the text that is to be the replacement for the current proposition.
F2	Select this word	Select the current text word to be the replacement for the current proposition.
F3	select > 1 Word	Select a set of words in the text to be the replacement predicate for the current proposition. This set of words will begin with the current text word and move forward in the text from that point (see MULTIPLE WORD MENU). The proposition will be anchored to the current word in the text (see ANCHORING PROPOSITIONS).
F4	Enter predicate	Enter a string that is to be the replacement predicate for current proposition (see EDITING INFORMATION). The proposition must be anchored to a word in the text (see ANCHORING PROPOSITIONS).
F5	select Standard	Select a standard predicate to be the replacement predicate for the current proposition (see SELECT STANDARD PREDICATE MENU). The predicate must be anchored to a word in the text (see ANCHORING PROPOSITIONS).
ESC	eXit	Return to the EDITING PROPOSITION MENU.

REPLACE ARGUMENT MENU

F1	Move to new word	Move to a word in the text that is to be the replacement argument for the current proposition.
F2	Select this word	Select the current text word to be the replacement argument for the current proposition.
F3	select > 1 Word	Select a set of words in the text to be the replacement argument for the current proposition. This set of words will begin with the current text word and move forward in the text from that point (see MULTIPLE WORD MENU).
F4	Enter argument	Enter a string that is to be the replacement argument for the current proposition (see EDITING INFORMATION).
F5	embed Prop	Select a proposition to be embedded as the replacement argument of the current proposition. At least one other proposition must exist to make this selection.
F6	Position argument	Indicate the position within the set of arguments for the current proposition that is to be replaced. If a position is not explicitly indicated by selecting this option, the last argument in the proposition will be replaced.
ESC	eXit	Return to the EDITING PROPOSITION MENU.

MOVE TO TEXT WORD MENU

➡	Move Right	Move to the next word in the text. If the current word is the last word, the text cursor will not be moved.
⬆	Move Up	Move to a word above the current text word. If the text cursor is on cursor is on the first line, the text cursor will not be moved.
Home	First Word	Move to the first word in the text.
PgUp	Beginning of Line	Move to the first word on the current line. If the cursor is already at the first word in the text, the text cursor will not be moved. If it is on the first word of some other line, the text cursor will be moved to the first word of the previous line.
⬅	Move Left	Move to the previous word in the text. If the current word is the first word, the text cursor will not be moved.
⬇	Move Down	Move to a word below the current text word. If the current text word is on the last line, it will not be moved.
End	Last Word	Move to the last word in the text.
PgDn	End of Line	Move to the last word on the current line. If the cursor is already at the last word in the line, the text cursor will move to the last word of the next line. If the cursor is on the last word in the last line, the text cursor will not be moved.
ESC	eXit	Return.

MOVE TO PROPOSITION MENU

↑	Up a Proposition	Move to the previous proposition, if there is one.
Home	First Proposition	Move to the first proposition.
PgUp	Up a Window	Move to the proposition twelve before the current one, if one exists. If not, move to the first proposition.
→	Go to Proposition	Go to a proposition to be indicated by number. The number must be a legitimate proposition number.
↓	Down a Proposition	Move to the next proposition, if there is one.
END	Last Proposition	Move the the last proposition.
PgDn	Down a Window	Move to the proposition twelve after the current one, if one exists. If not, move to the last proposition.
ESC	eXit	Return.

PRINT INFORMATION MENU

Note: to use the PRINT options, a printer must be attached to the computer running the Propositional Analysis System, and the printer must be on-line.

F1	print Text	Once a text file has been input into the program, it is available to be printed.
F2	print Prop list	Once there is at least one proposition, a list of the propositions currently identified can be printed. This list will be preceded by the text.
F3	print Analysis	Once a text file has been input into the program, information about the propositional analysis can be printed. Currently, this information includes the following: the total number of words and the number of unique words in the text, the number of words in the dictionary (which includes words that are in propositions but not in the text), the number of propositions, the total number of arguments in all propositions, and the list of propositions. For each proposition, its anchor word and all propositions in which that proposition is embedded are listed. If there are no propositions derived, only the first three items will be printed.
ESC	eXit	Return to the MAIN MENU.

SAVE INFORMATION MENU

F1	save System file	Once a text file has been input into the program, a system file can be created from it. This file contains information which will allow a text to be analyzed in multiple sessions.
F2	save Prop list	Once there is at least one proposition, a list of the propositions currently identified can be saved in ASCII format in a file.
F3	save Analysis	Once a text file has been input into the program, information about the propositional analysis can be saved in an ASCII format in a file. Currently, the file contains the total number of words and the number of unique words in the text, the number of words in the dictionary (which includes words that are in propositions but not in the text), the number of propositions, the total number of arguments in all propositions, and the list of propositions. For each proposition, its anchor word and all propositions in which that proposition is embedded are listed. If there are no propositions derived, only the first three items will be written to the file.
ESC	eXit	Return to the MAIN MENU.

USE UTILITIES MENU

F1	Save predicates	Save predicate information for use with another file.
F2	Use pred info	Use predicate information that was saved from another analysis.
F3	Merge pred info	Merge predicate information from two files. These files must already exist. They should have been created either with Save predicates or Merge pred info. The first file will be altered. Three scratch files are used: "junk1.tmp," "junk2.tmp," and "junk3.tmp." Files with these names will be destroyed if they are on the target disk. This option may still have some problems.
F4	Chunk proposition	Indicate process chunk boundaries for an analysis. The analysis should be complete before selecting this option. This option may have some problems still, so save the analysis before using it.
F5	Toggle bell	Allow the beep which accompanies many messages to be silenced if it is on or turned on if it is off.
ESC	eXit	Return to the MAIN MENU.

EDITING INFORMATION MENU

HOME	Move to the first character in the edit field.
END	Move immediately after the last character in the edit field.
⬅	Move a character left in the edit field, if not at the first character.
➡	Move a character right in the edit field, if not at the position immediately after the last character character.
BS	Delete the previous character in the edit field, if not at the first character.
DEL	Delete the current character in the edit field, if not immediately after the last character.
INS	Toggle the mode of insertion between insert and overstrike. Initially, the mode is set to insertion.
<Printable character>	If there is room for another character in the edit field, the character is inserted.
<SPACE>	If there is room for another character in the edit field, the space character is inserted.
⬆	Clear all characters from the edit field.

ERROR MESSAGES

Non-fatal Error Messages

A proposition cannot be embedded in itself.
An attempt was made to embed a proposition in itself.

Cannot analyze: no text
Option "Analyze text" in MAIN MENU cannot be selected until a text has been selected and read in using "Read information."

Cannot chunk: no propositions
Option "Save predicates" in USE UTILITIES MENU cannot be selected until at least one proposition has been derived from a text.

Cannot delete: no propositions
Option "Delete prop" in ANALYZE TEXT MENU cannot be selected until at least one proposition has been derived from a text.

Cannot edit: no propositions
Option "Edit prop" in ANALYZE TEXT MENU cannot be selected until at least one proposition has been derived from a text.

Cannot edit argument: no arguments in this proposition
Option "edit Argument" in EDIT PROPOSITION MENU cannot be selected unless the current proposition has at least one argument.

Cannot move: only one proposition
Option "Move to prop" in ANALYZE TEXT MENU, SELECT ARGUMENT MENU, EMBED PROPOSITION MENU, DELECTE PROPOSITION MENU, or EDIT PROPOSITION MENU cannot be selected until at least one proposition has been derived from a text.

Cannot move above first line in text
An attempt was made to move the text cursor before the first line in the text.

Cannot move before first word in text
An attempt was made to move the text cursor before the first word in the text.

Cannot move past last line in text
An attempt was made to move the text cursor past the last line in the text.

Cannot move past last word in text
An attempt was made to move the text cursor past the last word in the text.

Cannot print: no text

Option "Print information" in MAIN MENU cannot be selected until a text has been selected and read in using "Read information."

Cannot print proposition list: no text

Option "print Prop list" in PRINT INFORMATION MENU cannot be selected until at least one proposition has been derived using "Analyze Text."

Cannot read predicate information: no text

Option "Chunk proposition" in USE UTILITIES MENU cannot be selected until at least one proposition has been derived from a text..

Cannot remove argument: no arguments in this proposition

Option "Remove argument" in EDIT PROPOSITION MENU cannot be selected unless the current proposition has at least one argument.

Cannot save: no text

Option "Save information" in MAIN MENU cannot be selected until a text has been selected and read in using "Read information."

Cannot select argument: no propositions

Option "select Argument" in ANALYZE TEXT MENU cannot be selected until at least one proposition has been derived from a text.

Cannot write predicate information: no propositions

Option "Save predicates" in USE UTILITIES MENU cannot be selected until at least one proposition has been derived from a text.

Cannot write proposition list: no propositions

Option "Save propositions" in SAVE INFORMATION MENU cannot be selected until at least one proposition has been derived from a text.

Circularity test failed: P<number> cannot be embedded in P<number>.

Embedded propositions cannot be edited.

An embedded proposition is not a string that can be edited, but a pointer to another proposition. To change a predicate that is a proposition, use "Replace predicate" in the EDIT PROPOSITION MENU. To change an argument that is a proposition, first use "Remove argument" in the EDIT PROPOSITION MENU. Then use "select Argument" in the ANALYZE TEXT MENU.

File contains a word of more than 50 characters.
The text file contains at least one word that is larger than the maximum allowable word length.

File contains more than 80 consecutive nonblank characters.
A line was read from a text file which contained at least eighty consecutive nonblank characters.

File contains no recognizable words.
An attempt was made to read an empty text file or a text file with no character strings recognizable as words.

Multiple word too long
Multiple word concepts are limited to 50 characters.

P<number> cannot be deleted if it is embedded in P<number>.
In order to delete a proposition, all references to it in other propositions must be removed. This can be accomplished by replacement or by removal.

PROP: invalid option <character>
An invalid option was entered on the command line when the program was invoked.

Usage: prop -r<recovery file>
Invalid information was entered on the command line when the program was invoked.

Write fault error writing device PRN
Abort, Retry, Ignore?
This is a DOS error message that may occur if you attempt to print without a printer attached to the computer or with the printer off line. If you get this message, typing "a" will cause the Propositional Analysis System to be aborted.

File Errors

Error writing MONITOR.DAT: no further recovery possible

An error was encountered in writing the file "MONITOR.DAT." (see The Monitor File). This could mean that there was not enough free disk space to continue saving keystrokes or that an error occurred when writing keystrokes to the file. If this happens and you would like to continue saving your keystrokes, save your system file on another disk drive or floppy, exit, and make room on the drive which contains the file prop.exe. Then you can reenter the Propositional Analysis System, read in your system file, and continue.

<filename> cannot be opened for reading.

The recovery file could not be opened. This probably means that the file does not exist.

<filename> does not exist.

The file to be opened for reading cannot be found. A new name must be entered (see EDITING INFORMATION MENU), or ESC pressed, to exit without entering a file name.

<filename> already exists. Do you want to replace it?

A file by the designated name has been found. A response of 'y' will replace the file with the new one. In some cases, the old file may be saved as a backup. A response of 'n' will allow the entry of another file name.

<filename> cannot be opened for writing.

The file could not be opened for writing. This should not happen.

File not written: insufficient disk space

The disk did not have enough free space to complete writing a file. If this happens when you are trying to write to a floppy disk, try to put in a floppy disk with more free space.

File not written: error in writing to disk

An error was encountered in writing the file. If this happens when you are trying to write to a floppy disk, try to put in another floppy disk.

Try another name: <filename>

The previously entered file name is available for editing or replacing (see EDITING INFORMATION MENU).

Fatal Program Error Messages

If you get any of the following errors, please copy the following files to a diskette: the original file(s) as input, the new system file (if any), and MONITOR.DAT. Then please contact Dr. Althea Turner, (303) 492-6655.

INS_LEX_INFO: No match on type

DEL_LEX_INFO: No match w in type

DEL_LEX_INFO: No match p in type

GET_WORD: Word level too large

REMOVE_TT: Proposition not found

If you get any of the above errors, some aspect of memory appears to have been compromised.

<function name> Allocation Error: If changes have been made, save them.

No memory is available for further processing. If changes have been made, save them in a system file. It would be best not to replace another file but to create a new one, in case there are problems in the new system file. It is also possible that an attempt was made to read a file as a system file that was not one.

<function name> Release Error: If changes have been made, save them.

An error was made when memory was released. If changes have been made, save them in a system file. It would be best not to replace another file but to create a new one, in case there are problems in the new system file.

WEBSTER: <word>|<> not found in dictionary, mode=<0|1>. Save system file."

The dictionary appears to have been compromised. If changes have been made, save them in a system file. It would be best not to replace another file but to create a new one, in case there are problems in the new system file.

REFERENCES

Anderson, J. R. (1976). Language, memory, and thought. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Bovair, S., & Kieras, D. E. (1983). A guide to propositional analysis for research on technical prose. In B. K. Britton and J. B. Black (Eds.), Understanding expository prose. Hillsdale, NJ: Erlbaum.

Fillmore, C. J. (1968). The case for case. In E. Bach & R. T. Harms (Eds.), Universals of linguistic theory. New York: Holt, Rinehart & Winston.

Frederiksen, C. H. (1975). Representing logical and semantic structure of knowledge acquired from discourse. Cognitive Psychology, 7, 371-458.

Frederiksen, C. H. (1977). Semantic processing units in understanding text. In R. O. Freedle (Ed.), Discourse processes; Advances in research and theory. Norwood, NJ: Ablex.

Graesser, A. C. (1981). Prose comprehension beyond the word. New York: Springer-Verlag.

Graesser, A. C. and Goodman, S. M. (1983). How to construct conceptual graph structures. In B. K. Britton and J. B. Black (Eds.), Understanding expository prose. Hillsdale, NJ: Erlbaum.

Kintsch, W. (1974). The representation of meaning in memory. Hillsdale, NJ: Erlbaum.

Kintsch, W. & van Dijk, T. A. (1978). Toward a model of text comprehension and production. Psychological Review, 85, 363-394.

Meyer, B. J. F. (1975). The organization of prose and its effects on memory. Amsterdam: North-Holland.

Meyer, B. J. F. (1983). Prose analysis: Purposes, procedures, and problems. In B. K. Britton and J. B. Black (Eds.), Understanding expository prose. Hillsdale, NJ: Erlbaum.

Fodor, J. A., Garret, M. F., Walker, E. C., & Parkes, C. H. (1980). Against definition. Cognition, 8, 263-367.

Turner, A. A., McCutchen, D., & Kintsch, W. (1986, November). The generation of macropropositions during comprehension. Paper presented at the Twenty-seventh Annual Meeting of the Psychonomics Society, New Orleans, LA.

Turner A. A., & McCutchen, D. (in preparation). Reduction and abstraction processes operating during comprehension.

Turner, A. A., & Greene, E. (1978). Construction and use of a propositional text base. JSAS Catalogue of selected documents in psychology, MS 1713.

van Dijk, T. A., & Kintsch, W. (1978). Strategies of discourse comprehension. New York: Academic Press.