A THEORY AND SIMULATION OF MACROSTRUCTURE

Sheryl R. Young

Department of Psychology
University of Colorado

Abstract

A cyclical process theory of macrostructure is presented which addresses the LTM representations of macropropositions, micropropositions and their relationship. The process model is simulated by a computer program called a Simulation of Text On-line Processing (STOP), which enables theory specific predictions to be derived for any text.

The theory describes two hierarchical memorial representations, one continaing macropropositions and one containing micropropositions. The microstructure is hypothesized to be composed of many hierarchically structured components, or subtrees, each containing micropropositions related to a specific macroproposition.

Recall of micropropositions is hypothesized to be a function of four components: 1- The importance of the macroproposition pointing to the subtree continaing the microproposition; 2- The level within the subtree hierarchy occupied by the micropropositions; 3- The number of processing cycles a microproposition participates in; and 4- the interaction of subtree and level within a subtree.

The theory was tested by comparing the predictions generated by STOP for three texts with the incidental recall protocols. All four hypotheses were verified.

It is well documented that people do not recall everything they read. Rather, some elements of a text are consistently recalled more frequently than other aspects. The portions of a text which are recalled best are, in some sense, the "important elements" (Newman, 1939; Gomulicki, 1956; Johnson, 1970; Brown & Smiley, 1977). Given that one goal of a good reader is to derive a highlevel synopsis of the relevant aspects of a text, we must ask how to characterize what is important or relevant.

Empirical evidence points to the usefulness of characterizing the importance within a text in two ways: by its microstructure, and by its macrostructure. A text's microstructure is a representation of the facts presented in a text. It is composed of the sequence of propositions underlying the series of sentences contained in the discourse (vanDijk, 1977). Propositions refer to the units in memory into which text is encoded (McKoon & Ratcliff, 1980). Usually, propositions are represented as a predicate (property or relationship) followed by a list of arguments (items which are related via the predicate). Propositions are hierarchically organized in a long-term memory representation of the text base which is called the microstructure. The hierarchical structuring of propositions in LTM is performed with the constraint that the sequences be linearly coherent. One aspect of coherence involves reference to similar or related entities. However, coherence is also determined conceptually via relation to the discourse topic and main ideas of the text.

One of the most widely supported text structure principles used to explain why some textual elements are recalled better than others can be called the microstructure levels principle. The principle is based using text grammars to generate hierarchically structured representations of the importance of micropropositions. The principle claims that micropropositions which are located at high levels of a microstructure hierarchy or coherence graph have a higher probability of being recalled than items located at lower levels of the microstructure. In other words, the probability of a microproposition being recalled is a function of the level occupied by the particular microproposition in the microstructure. Micropropositions located at the upper levels of the microstructure hierarchy should have a higher recall probability than micropropositions located at the lower levels of the microstructure.

This explanation has been empirically supported in studies which investigate recall of micropropositions within either isolated sentences or single, short paragraphs (Meyer & McConkie, 1973; Kintsch, 1972, 1974; Meyer, 1975; Kintsch & Keenan, 1973; McKoon, 1977; Kintsch, Kozminsky, Streby, McKoon & Keenan, 1975). In these experiments, the higher in the microproposition hierarchy a microproposition occurred, the more likely it was to be recalled, as the principle predicts. The general finding in all of these studies is that the recall probabilities associated with micropropositions located at Levels 1 - 4 decrease as level in the microstructure increases, but asymptote at Level 4. However, it must be noted that the textual materials employed in these experiments were very short. As the length of a text increases, the number of levels in the overall microstructure increases. Furthermore, the rules for generating microstructures (Kintsch & vanDijk, 1978; Miller & Kintsch, 1980) are sensitive to where a microproposition is located in a text. Micropropositions which occur at the beginning of a text tend to be placed at higher levels in the microstructure than micropropositions occurring later in a text. It is hard to believe that people do not recall items which occur after the first few sentences of a text. However, these microstructure findings may well be justified if one assumes that the materials in these studies were subsumed under a single topic. It is therefore proposed that the microstructure levels effect will persevere only when constrained within a single local topic contained within a text. Thus, in lenghtly texts which contain numerous local topics, level within the microstructure should only predict recall of micropropositions if one creates a separate microstructure representation for the micropropositions associated with each local topic.

A second body of evidence suggests that the local topics and overall meaning of a text can be characterized at a global level. The global structure which represents the overall meaning of a text has been called the macrostructure (Bierwisch, 1965; vanDijk, 1972; Kintsch & vanDijk, 1975; vanDijk, 1977; Kintsch & vanDijk, 1978). Macrostructures are formed from main ideas, or macropropositions, which are propositions usually not stated in the text, but inferred from sequences of micropropositions. Essentially, macropropositions capture the meaning of a series of micropropositions.

There are four rules which have been developed to explain how a macroproposition is created to capture the essence of many micropropositions (vanDijk, 1977). The first rule is generalization, where a superordinate concept is substituted for a series of micropropositions can each be considered its subsets. The second macro-rule is deletion. The deletion rule deletes irrelevant propositions from a series of propositions. The third macro-rule, integration, selects special micropropositions which are substituted for a series of microspopositions which are conditions, components or consequences of the special microproposition to be incorporated into the macrostructure. The fourth macro-rule is construction. If there is an idea or proposition such that a series of micropropositions are normal conditions, components or consequences of this idea, then the idea is included in the macrostructure in place of the series of micropropositions. Thus, the macrorules guide the creation of the macropropositions which are then ordered via coherence to form the macrostructure.

There is much evidence to support the assertion that readers derive a high level synopsis of a text's gist, or a macrostructure representation of a text's meaning (Bierwisch, 1965; vanDijk, 1972, 1975, 1977; Kintsch & vanDijk, 1978;

vanDijk & Kintsch, 1977; Kieras, 1981a, 1981b, 1982; Masson, 1983; Singer, 1983; Young, Angell & Bourne, 1984; Kintsch, 1977; Mandler & Johnson, 1977; Rumelhart, 1975). Early studies (Bierwisch, 1965; vanDijk, 1972, 1975, 1977; vanDijk & Kintsch, 1975; Rumelhardt, 1975; Mandler & Johnson, 1977) concentrate on the macrostructures created during the comprehension of a story. These experiments support the assumptions that: (1) Stories can be divided into a sequence of episodes, where each episode consists of an exposition, a complication, and a resolution, (2) each of these textual components can be summarized into a proposition (macroproposition). Additionally, the sequence of episodes can be hierarchically described (Kintsch, 1977; Rumelhardt, 1975; Mandler & Johnson, 1977), and the hierarchical descriptions agree woth subject ratings and summaries for both normal and scrambled texts (Kintsch, 1977; Kintsch & vanDijk, 1975; Johnson, 1970). Additionally, developmental differences have been found in the ability to use macro-rules (Brown, 1981; Brown & Day, 1983; Brown & Smiley, 1978). Furthermore, empirical evidence suggesting ways in which mature readers determine what is macro-relevant has been obtained (Kieras, 1980, 1981a, 1981a, 1982; Collin, Brown & Larkin, 1980). Kieras (1980) has shown that subjects use an initial mention strategy to determine the topic of a passage. People look for clues regarding the main idea of a passage in the initial sentence. Kieras (1982) further refined this strategy by demonstrating that subjects select a candidate passage topic and attempt to verify their hypothesis by checking the passage for confirmation.

As indicated by the bodies of evidence discussed above, most research efforts have addressed a singular type of analysis and have failed to consider possible interdependencies between microstructure, and macrostructure. However, there is a great deal of evidence pointing to the importance of characterizing a text at each of these levels of analysis. Furthermore, the predominant notion in cognitive psychology is that the reader can only process a limited amount of information at each point in time. Given multiple representations of a text and numerous knowledge based interactions, it is encumbent upon researchers to derive a text processing theory which describes the formation of multiple text representations during the encoding process.

There have been a few attempts to describe the encoding processes responsible for creating text representations and explaining the differential importance of textual elements. Meyer (1977) reported evidence to suggest that items which are percieved as being important are more thoroughly encoded during processing. In a similar vein, Frederiksen (1972, 1975) has proposed that superior recall of important textual elements can be explained within a levels of processing approach (Craik & Lockhart, 1972). Frederiksen, like Meyer, claims that important elements are more deeply processed or encoded in the information processing system.

The Kintsch and vanDijk (1978) model of text processing attempts to describe the differential encoding of important textual elements. Their model describes the memorial representations of both micropropositions and macropropositions. The theory centers on the assumption that working memory has a limited capacity. From this basic premise, a cyclical theory is evolved, where during a single processing cycle only a small segment of text is input to working memory. These items are processed and then the next segment of text is input. In this theory, each segment of text input to working memory is composed of micropropositions. The segments or chunks are determined by surface structure boundaries (Aaronson & Scarborough, 1977). Each chunk of micropropositions is checked for coherence upon input. This process involves looking for argument overlap among the propositions in the input chunk and propositions held over from previous cycles in the working memory buffer. If a chunk is judged to be coherent, it is input to long term memory.

If coherence cannot be established, a reinstatement search of all previously processed propositions is conducted. A reinstatement search is designed to look at all previously processed micropropositions held in long term memory to see if there is overlap between any of them and the unconnected microproposition(s) in the current input chunk. If coherence is established, the unconnected micropropositions from the current input chunk are connected to a level below the already processed micropropositions they overlap with. When coherence cannot be established via reinstatement searches, a bridging inference is performed.

Following the processing of each chunk of propositions, some propositions are selected to be held over in the working memory buffer. Only the held over propositions are available for connecting new incoming material with already processed information. Kintsch and vanDijk postulated that items are selected on the basis of their level in the short term memory structure created during the processing of each chunk of information. The most important items are believed to be located at the highest levels of the STM structure. Thus, Kintsch and van Dijk postulate that a subject traces the STM tree structure by choosing the highest level microproposition first and then selecting level 2 propositions on the basis of recency.

Kintsch and vanDijk also address the processing of macropropositions. Their first assumption is that macroprocessing is independent of microprocessing. The macroprocessing portion of their theory is believed to be hierarchical. However, these investigators fail to discuss the nature of the macrorepresentation in memory, aside from declaring that it too must be coherent. Rather, they state that macro-operators are applied in several cycles, so that items chosen to be macrorelevant early on in processing might not be macrorelevant in later cycles. Unfortunately, these investigators do not explain where in the cyclical processing cycle macroprocessing takes place, nor do they specify the nature of the macrostructure representation aside from denoting that some items are selected to be macrorelevant on more than one occasion.

The microstructure portion of the Kintsch and vanDijk model was simulated by Miller and Kintsch (1980). In the process of creating the simulation, Miller and Kintsch made a few important additions to the Kintsch and vanDijk model. First, by contrasting the predictions of the simulation to subject recall protocols, the simulation had the best fit to the data when buffer size was set at two micropropositions. Second, although buffer size was set, the buffer was believed to be flexible. By allowing the buffer to stretch to a size of 3, Miller and Kintsch provided for the inclusion of embedded propositions in the working memory buffer. Therefore, complete ideas were held over. Last, as Kintsch and vanDijk hypothesized that free resources were allocated to increasing buffer size, Miller and Kintsch allowed buffer size to be expanded by one slot on the first processing cycle to accommodate the hypothesis that readers have more free resources when beginning a task.

In this paper I will present a theory which describes the formation of both macrostructure and microstructure memorial representations during encoding. Additionally, the relationship between microstructure and macrostructure processing and memorial representation is explicitly addressed. The proposed theory is implemented by a computer simulation which is capable of deriving specific, theory based predictions for any text input to the simulation. Furthermore, the predictions derived by the simulation for a number of texts are empirically tested in the studies described later. The theory presented takes as a given the process model of microprocessing developed by Kintsch and vanDijk and Miller and Kintsch and adds to it a macrostructure which influences the short term memory operations applied to the microstructure. The macrostructure is hypothesized to affect the recall of micropropositions. Relationships between the macrostructure and microstructure are depicted as "tree" and "subtree" relationships. The theory is described next, and later simulated and empirically tested, with all modifications to the Miller and Kintsch (1980) simulation of Kintsch and vanDijk's microprocessing theory explicitly discussed.

The Theory of Macrostructure

The heuristics described in this theory are hypothesized to drive the processing of a text, taking into consideration the limitations of working memory. These principles are text based processing heuristics which are hypothesized to occur during the encoding of a text. They addresses the interaction of microprocessing and macroprocessing and describe principles which guide the formation of a macrostructure and a microstructure during the encoding of a text. By including provisions for a LTM macrostructure, the theory is capable of describing the comprehension of fairly long texts (up to perhaps 10 or more pages). The Miller and Kintsch (1980) model can be thought of as an accurate processing description of reading behavior on short, paragraph length texts. As such, we can think of the Miller and Kintsch model as a special case of this model, when this model is applied to short texts containing one major main idea. The theory does not address the complex interaction of knowledge and macroprocessing or microprocessing, or the derivation of macropropositions from micropropositions.

Basic Assumptions

The theory taks as a given the processing assumptions of the Kintsch and vanDijk (1978) model of text processing, but adds four assumptions which are not shared by Kintsch and vanDijk. Kintsch and vanDijk assume that multiple long term memory representations are created upon the reading of a text. These representations are created during encoding, and thus a cyclical processing model is assumed. Chunks of input propositions are created via surface structure boundaries. Each chunk is internally processed for coherence. Coherence between chunks is established by maintaining propositions from previously processed chunks in a working memory buffer, so that new propositions may be connected to them. There are four basic assumptions underlying the theory of macroprocessing which are not held in common with the Kintsch and vanDijk model of text comprehension. These are: (1) An initial macrostructure is created during the encoding of a text, (2) Macro-operators are applied only once to each set of micropropositions and are not applied to macropropositions, (3) The last macroproposition derived is maintained in the working memory buffer, and (4) Each macroproposition points to a specific microproposition which is located at the top level of hierarchy of micropropositions, or subtree, contained in the microstructure and is related to the macroproposition. These four assumption are critical to the proposed theory, and are explained next.

The first assumption is that the macrostructure memorial representation described in the theory is postulated to be the initial macrostructure a subject forms while encoding a text. Thus, the LTM representation of the macrostructure is a product of short term memory text processing. While there are circumstances where a reader might perform a reorganization of the macrostructure as a result of later input to the information processing system, the criteria for performing a reorganization are not well understood. Furthermore, preliminary data on the reorganization process (Bransford & Johnson, 1973) indicate that subjects only perform incomplete reorganizations when faced with radically conflicting input. As there is much evidence to suggest that subjects form a representation of the main points of a text while encoding (Kieras, 1980, 1982; Collins, Brown & Larkin, 1980; Guindon & Kintsch, 1984; Young, Angell & Bourne, 1984; Young, 1984), the theory of macrostructure addresses the on-line formation of macrostructures.

The second assumption concerns the application of macro-operators. Macropropositions are hierarchically organized in the long term memory macrostructure representation of the text. The differential importance of macropropositions to the gist of a text is represented by the level a macroproposition occupies in the macrostructure. The higher a macroproposition's location in the macrostructure hierarchy, the greater its importance to the overall gist of the text. Thus, macro-operators are believed to operate only on micropropositions, and only once on a set of micropropositions. However, the theory does not address the knowledge based constraints which guide the application of macro-rules.

This approach can be contrasted with Kintsch and van Dijk's cyclical application of macro-operators. Kintsch and van Dijk acknowledged that macropropositions vary in their importance to the overall gist of a text. To account for this fact, they applied macro-operators on more than one occasion. Thus, in their model, some macropropositions are deemed to be macrorelevant on more than one occasion. However, Kintsch and van Dijk fail to specify when in the processing cycle macro-operators are applied. Since all processing is hypothesized to occur in working memory, the idea that macro-operators are applied repeatedly on the same information implies that all previously derived macropropositions must be retrieved from the LTM macrostructure and be reprocessed. The hierarchical approach used in the following model accounts for differences in importance among macropropositions by hypothesizing a hierarchically organized LTM macrostructure which is created while encoding portions of text cyclically. By using a levels approach to macrostructure, differences in the ability of a macroproposition to elicit recall of associated micropropositionsis accounted for, as described in the fourth assumption.

All active text processing and selection of important micropropositions for holdover in the short term memory buffer occurs in working memory. The third assumption relates to the holdover selection process. It states that the last macroproposition added to the macrostructure is maintained in the working memory buffer until a new macroproposition is derived. Thus, two propositions are maintained in the working memory buffer to assist in the processing of new information, the last macroproposition added to the macrostructure and the most recent microproposition which is located at the high level in STM. The microproposition which is held over in STM is the item located at the highest level of STM which was not involved in the derivation of the last macroproposition.

As each incoming chunk of propositions is input to the system, it is connected to previously processed items on the basis of the micropropositions selected for storage in the working memory buffer. Because the items held over from previous processing cycles are the only information remaining in the working memory buffer, proper selection is crucial for efficient processing. Thus, by including the last macroproposition in the working memory buffer an emphasis is placed on local topic reference during microstructure coherence processing.

The fourth assumption holds that each macroproposition has a pointer to an associated microproposition. This assumption is based on the "tree" and "subtree" concepts mentioned earlier. It envisions a hierarchically structured microstructure tree. This tree contains a number of subtrees. Each subtree is hierarchically organized, and is composed of a group of micropropositions which are directly related to a specific macroproposition. As macropropositions are derived from a limited number of micropropositions, the most important microproposition involved in the derivation of the macroproposition is the root node of the subtree.

The model assumes that the micropropositions related to each macroproposition vary in their importance to the macroproposition. This is represented by hierarchical subtrees. In the theory, the microproposition which is pointed to by its associated macroproposition is the microproposition which was most important in deriving the macroproposition. Propositions located just below the pointed-to microproposition in the subtree hierarchy are more closely related to the main idea than are items located at lower levels of the subtree hierarchy.

The strength of the pointers between macropropositions and micropropositions is not constant. Rather, their strength is a function of the level in the macrostructure the macroproposition occupies. Specifically, the more important a macroproposition is to the overall gist of a text, the stronger will be its pointer to an associated microproposition. The strength of a pointer has recall implications. Namely, more of the body of micropropositions associated with an important macroproposition will be recalled than micropropositions associated with a less important macroproposition.

Creation of the Macrostructure

There are three basic rules involved in the creation of a hierarchical macrostructure tree from macropropositions. The first macroproposition derived is always placed at the root node of the macroproposition tree. In a well written text, the first macroproposition is usually representative of the main theme addressed in the text. However, even in cases where the above assumption does not apply, evidence suggests that subjects select a candidate main idea of a text from either the first sentence or the first paragraph of the text (Kieras, 1980; 1982; Collins, Brown & Larkin, 1980).

Following the initial selection of a macroproposition to represent the main idea or topic of the text, additional macropropositions derived later in processing are added to the macrostructure hierarchy tree on the basis of argument overlap. Argument overlap means that the two macropropositions have some common element between then, or refer to the same person or event. Argument overlap, or referential coherence accounts for a reader's ability to relate old information with new information. This method also allows for some causality information to be preserved, as micropropositions represent causality when it is explicitly stated in the text. However, meaning and inference are also important in determining coherece. Meaning relations have not been used here nor in Kintsch and vanDijk to ease simulation. Furthermore, referential coherence is objective and has been shown to be a reasonable approximation of the relations readers form while reading by Kintsch and vanDijk (1978) and Miller and Kintsch (1980).

The following are the heuristics used for adding a newly derived macroproposition to the macrostructure. If the new macroproposition overlaps with more than one macroproposition which is already stored in memory, the new macroproposition is attached to the most important existing macroproposition. Should the new macroproposition overlap with more than one macroproposition of equal importance, then degree of overlap is accessed and the new macroproposition is connected to the most closely related macroproposition. Should the degree of overlap be equal, then the new macroproposition is attached to the macroproposition derived from the most recently input micropropositions. The rationale for this rule is that processing limitations only permit a limited number of macropropositions to be retrieved at any one point in time. Thus, retrieval of macropropositions is conducted in a top down manner. The top level macroproposition is the first macroproposition to be retrieved from LTM. Following its reinstatement into STM, it is assessed for overlap with the new macroproposition. By connecting incoming macropropositions to the highest level possible in the tree, importance to the main theme is emphasized above order of incoming information. As the macrostructure represents the overall gist of the text, we can model this process by connecting macropropositions in the macrostructure in a manner which represents their overall importance to the theme. In this way, the rationale underlying Kintsch and vanDijk's cyclic application of macro-operators is preserved. As additional macropropositions can be added to any level except Level 1 in the macrostructure, differential importance of macropropositions is preserved without reapplying macro-operators at arbitrary points in the on-line processing of a text.

If an incoming macroproposition does not share any common referents with the macropropositions already in the macrostructure, it is placed at a level beneath the lowest level in the tree and is marked with a property which indicates that it is not connected to the other macropropositions in the macrostructure. The implications of these rules for recall will be discussed shortly.

Relationship between Micro and Macrostructures

The relationship between the macrostructure and the microstructure is rather complex. However, there are two basic ways the macrostructure influences the microstructure. First, the macrostructure is hypothesized to influence the formation of the microstructure. The manner in which this influence occurs is that the most recently input macroproposition (input to the macrostructure) influences which micropropositions are held over in the buffer. Since it is impossible to process an entire text at one time, chunks of micropropositions are input to working memory. These are checked for coherence with prior input. Thus, which propositions are maintained in the buffer is of extreme importance to the creation of an overall coherent micro-representation of a text. By maintaining the last macroproposition input to the macrostructure in working memory, local topic continuity between chunks of micropropositions is emphasized.

As Miller and Kintsch (1980) determined that a buffer size of two provided the best fit of their simulation to the data, their buffer size was adopted in this model. Thus, in addition to maintaining a copy of the most recently input macroproposition in the buffer, one additional microproposition is selected for holdover. This microproposition must be the most recent, high level microproposition not involved in deriving the last macroproposition. In order to preserve meaning, if the microproposition selected for holdover contains embedded micropropositions, the buffer has been made flexible to accommodate one embedded microproposition. However, if the microproposition selected for holdover contains more than one embedded microproposition, then only the most recent of the embedded micropropositions is maintained in the working memory buffer.

On any processing cycle when a new macroproposition is input to the macrostructure, the microproposition candidates for holdover in the buffer are resticted to micropropositions which have not been held over from previous processing cycles where another macroproposition was guiding holdover. This provision is made to strengthen the relationship between microstructure subtrees and macropropositions, described in depth below.

The second way the macrostructure affects the microstructure is that the microstructure is hypothesized to be composed of many subtrees, each of which are associated with an individual macroproposition. Each macroproposition contains a pointer to a subtree, or the portion of the microstructure which contains facts which support and/or are related to the macroproposition.

It will be recalled that the strength of the pointer between a macroproposition and its associated microproposition subtree is a function of the importance of the macroproposition to the overall gist of the text. Thus, the more important a macroproposition is, the greater the number of associated micropropositions that will be recalled. The strength of these pointers is hypothesized to decay at equal rates over time.

Several factors are hypothesized to have a significant effect on the probability of recalling an individual microproposition. First, the subtree in which a microproposition is contained should influence the probability of recalling a microproposition. Items which are contained in subtrees of the microstructure which are associated with macropropositions located high up in the hierarchy should have a higher recall probability than micropropositions contained in subtrees which are associated with macropropositions located at lower levels of the macrostructure hierarchy.

Second, the level a microproposition occupies within a given subtree of the microstructure should influence its probability of being recalled. Micropropositions which are located high up in the microstructure subtree are closely related to the macroproposition pointing to that subtree of facts. Thus the probability of recalling a microproposition located high up in a subtree of the microstructure should be greater than the probability of recalling a microproposition located at lower levels of a microstructure subtree. Third, the number of processing cycles a microproposition participates in should increase the strength of its trace in LTM. Therefore, the probability of recalling a microproposition should increase as the number of processing cycles it participates in increases. Lastly, there should be an interaction between the importance of the macroproposition pointing to the subtree in which it is contained and the level within the subtree occupied by the microproposition. The level within a subtree effect should diminish as the importance of the macroproposition pointing to the subtree decreases.

Description of a Processing Cycle

The model takes as input a list of micropropositions representing the ideas in the text. These are grouped on the basis of surface structure boundaries into input units which are called chunks. On the first input cycle, subjects select a candidate main idea or topic for the text. This macroproposition derived by the subject is placed at Level 1, or the highest part of the macrostructure hierarchy. Furthermore, the most significant microproposition used in the formation of the macroproposition is placed at Level 1 of the microstructure. This microproposition is operationally defined as the microproposition which shares the most commont referents with the macroproposition. All additional micropropositions are connected to the microproposition at Level 1 via argument overlap. When all of the micropropositions in the first input chunk have been connected to one another either via argument overlap or inference making procedures, the structure created in STM is transferred to LTM, creating the beginning of the LTM representation of the microstructure. Following the transfer of micropropositions to LTM, a copy of the macroproposition created during this initial input processing is maintained in a working memory buffer along with two additional micropropositions. In this cycle, buffer size is expanded to three. This special provision for the first processing cycle is based on the assumption made by Miller and Kintsch (1980) that the reader has more free resources to allocate at the beginning of a task. These additional resources are allocated to increasing the size of the working memory buffer by one slot. The two micropropositions held over on the first processing cycle are selected on the criterion that they were not used in deriving a macroproposition. For all micropropositions satisfying this criterion, two are chosen which are located highest up in the STM representation of the last processing cycle, and are the most recent micropropositions. By selecting items for holdover in the working memory buffer, constraints are placed on the manner in which chunks of propositions can be related. Namely, relations between micropropositions are maintained by functional recency and local topic continuity. Additionally, all items which are maintained in the buffer for additional processing will have increased trace strength in LTM, according to the principles proposed by Atkinson and Shiffrin (1968).

On all additional processing cycles, an input chunk of micropropositions is connected to the items in the buffer via argument overlap. However, if micropropositions are found which do not share common referents with the last macroproposition, the microproposition held over in the buffer, or other micropropositions which have been connected in STM, a reinstatement search is conducted. This search is performed in a top down, breadth first manner. Therefore, if micropropositions are encountered which are unrelated to the local topic, a search of all previously processed micropropositions is performed. The top down nature of the reinstatement search ensures that the subject initially checks for common referents in the micropropositions related to the overall topic of the text, or macroproposition 1. Thus, if local topic reference fails, overall topic reference is assessed. When reinstatement searches fail to find micropropositions which share common referents with the unconnected micropropositions, it is assumed that the unconnected micropropositions are related to previous input, but a bridging inference is necessary to determine the nature of the relationship. Thus, an inference making procedure is initiated.

Following coherence processing in STM and the concurrent transfer of items not already represented in the LTM microstructure to LTM, the items in the input chunk are searched for possible macrorelevance (i.e. the macro-operators are applied if the necessary knowledge based constraints are satisfied). If the preconditions are satisfied and one of the macro-rules is applied, a macroproposition is created, and added to the macrostructure. This macropropositions has a pointer to the most important microproposition from which it was formed. Then propositions are selected for hold over in the buffer. If no additional macropropositions were derived from the most recent input chunk of micropropositions, the last macroproposition added to the macrostructure is selected for holdover in the buffer. However, if a macroproposition was derived, then it is maintained in the buffer. In addition, the buffer maintains one microproposition for additional processing. As stated earlier, this microproposition must satisfy at least three criteria to be held over. First it must not have been one of the micropropositions from which the held over macroproposition was derived, second it must be located at an upper level of the STM representation of the input chunk, and third it must be the most recent upper level microproposition satisfying the first two criteria. Then short term memory is cleared of all items except those chosen for holdover. This cycling continues until the entire text has been processed.

In the theory, the LTM representations of macrostructure and microstructure are described as trees, with the most important propositions located at the tree root node. The next section describes the operationalization of the structural ideas and cyclical processing assumptions embodied in the theory of macrostructure as they were implemented in a computer simulation.

The Simulation

This section describes the operationalization of the theory of macrostructure using a computer simulation written in INTERLISP. The simulation enables one both to implement the model's heuristics and derive both structural and processing predictions for each individual proposition contained in any text. To illustrate its operation, Table 1 presents the dribble file resulting

---

Insert Table 1 about here

---

from applying the simulation to a portion of one of the texts used in the experiment described later. The complete text can be found in Appendix 1 and the total dribble file produced during the run can be found in Appendix 2.

Formulation of the Simulation.

This program is a modification and extension of the Miller and Kintsch (1980) formalization of the Kintsch and vanDijk (1978) theory of text comprehension. Miller and Kintsch designed a microstructure coherence program which adds a chunk of micropropositions to the contents of the STM buffer and create a LTM representation of a text's microstructure. Their simulation took as input a list of propositions which had been presegmented. At the beginning of the first cycle, the user was asked to choose the most important microproposition from those contained in the first input chunk. The proposition supplied was placed at the top of the microstructure trees in both STM and LTM. Miller and Kintsch used the superordinate microproposition provided by the user to simulate macroprocesses. Following the user's selection of a superordinate microproposition, the program added the propositions remaining in the input chunk to STM. This process is performed by connecting micropropositions to the superordinate microproposition on the basis of argument overlap (i.e. common

Table 1

Illustrative Processing Cycle

The sentence "The lack of oxygen is the leading suspect in the reduced catch of valuable food fish" will be processed by the simulation in order to illustrate its operation.

At the beginning of this processing cycle, STM contains the following items:

```
Level 1:
   P27 points to (P28 P34) and represents
     (P27 (RESULT DAMAGE FISHERIES P28))
   Level 2:
      P28 points to NIL and represents
        (P28 (CALL REAGAN CLEAN-UP))
      P34 points to NIL and represents
        (P34 (EXTEND DAMAGE AREA SUSQUEHANA POTOMAC))
```

Additionally, the last macroproposition added to macro-memory was (M3 (INTEGRATE P27 P28)).

```
The input for this cycle is:
     (P43 (EQUAL OXYGEN DEPLETION LACK-OF OXYGEN))
     (P44 (SUSPECT P44.1))
     (P44.1 (BECAUSE LACK-OF OXYGEN P45))
     (P45 (REDUCE CATCH))
     (P46 (OF CATCH FISH))
     (P47 (MOD FOOD FISH))
     (P48 (MOD VALUABLE FOOD))
```

The program attempts to add these micropropositions to STM using the argument overlap routines. However, it is discerned that none of the items in the inputset share common referents with the last macroproposition or the most recent, held over microproposition. Thus, a LTM reinstatment search of the microstructure is conducted. The following is the output produced by the program in its attempt to build a coherent STM representation of the propositions contained in the inputset.

```
Propositions (P43 P44 P44.1 P45 P46 P47 P48) cannot be
added to STM: search LTM for a new starting proposition.
Do a LTM search for P43:
The LTM search succeeded: P1 can be reinstated via P43.
Bump P1's cycle counter.
Nothing from *INPUTSET* was placed, but the LTM search
  succeeded: P1 was found.
This counts as a reinstatement search.
Put P43 at level 2, pointed to by P1.
The proposition (P43 (EQUAL OXYGEN DEPLETION LACK-OF OXYGEN))
    overlaps with (P1 (FOUND WATER OXYGEN)).
Put P44.1 at level 2, pointed to by P1.
The proposition (P44.1 (BECAUSE LACK-OF OXYGEN P45)) overlaps
    with (P1 (FOUND WATER OXYGEN)).
Put P44 at level 3, pointed to by P44.1.
The proposition (P44 (SUSPECT P44.1)) overlaps with
```

(P44.1 (BECAUSE LACK-OF OXYGEN P45)).
Put P45 at level 3, pointed to by P44.1.
The proposition (P45 (REDUCE CATCH)) overlaps with
(P44.1 (BECAUSE LACK-OF OXYGEN P45)).
Put P46 at level 4, pointed to by P45.
The proposition (P46 (OF CATCH FISH)) overlaps with
(P45 (REDUCE CATCH)).
Put P47 at level 5, pointed to by P46.
The proposition (P47 (MOD FOOD FISH)) overlaps with
(P46 (OF CATCH FISH)).
Put P48 at level 6, pointed to by P47.
The proposition (P48 (MOD VALUABLE FOOD)) overlaps
with (P47 (MOD FOOD FISH)).

Next the program searches for new macropropositions. M4, as a result of oxygen depletion, the fish catch was reduced, is found to be a product of the micropropositions in the current inputset. Thus, it is added to the LTM macrostructure. M4 is a result of applying the INTEGRATION macro-rule to propositions P44.1, P45 and P46. The LTM macrostructure contains M1 at Level 1, M2 at Level 2, and M3 at Level 3. The program does the following:

The macroproposition (M4 (INTEGRATE P44.1 P45 P46))
is being added to macro-memory.
It overlaps with macropropositions (M1 M2).
Put M4 at level 2 in MACRO
M4 has value (INTEGRATE P44.1 P45 P46) and points to P44.1
in the microproposition tree.
Put M4 at level 2, pointed to by M1.
Buffer size is now 4.

The contents of STM are now added to the LTM microstructure representation by first retrieving the level occupied in the microstructure of P1, the proposition located at the top of the STM representation. Then, all items located at Level 2 in STM are assessed to see if they are already represented in the microstructure. As they are not, they are placed at the level below the level occupied by P1, and connected to P1. The remaining micropropostions in STM are added to the LTM microstructure in the same manner. Then, micropropositions are selected for holdover in the working memory buffer. As the last macroproposition added to the macrostructure was M4 (INTEGRATE P44.1 P45 P46) propositions P44.1, P45 and P46 are retained in the buffer. Then, an additional microproposition is selected which is part of the inputset added during the current cycle, as a new macroproposition was derived. The following is a copy of what the program outputs when determining what to maintain in the working memory buffer:

Apply the leading edge strategy.
Retain proposition P44.1 at level 1.
Proposition P45 is embedded in P44.1.
Retain proposition P45 at level 2.
Retain proposition P46 at level 3.
Retain proposition P43 at level 1.
At the end of cycle 9, STM contains:
Level 1:
P43 points to NIL
P44.1 points to (P45)
Level 2:
P45 points to (P46)
Level 3:

P46 points to NIL
The last macroproposition added to macro-memory was
(M4 (INTEGRATE P44.1 P45 P46)).

reference and shared arguments among propositions). First, the program searches the remaining propositions for items which overlap with the superordinate microproposition. Then, any remaining micropropositions are searched to see if they share a common referent with items located at Level 2. The micropropositions located at Level 2 are those which shared a common referent with the superordinate microproposition. The items located at Level 2 are searched for overlap with remaining micropropositions by order of recency. This matching procedure continues until either all the items in the input chunk have been connected.

Following this, the argument overlap process is called to build a connected representation of the micropropositions in the input chunk in LTM. First, the subject supplied superordinate microproposition is placed at Level 1 in LTM. Then the argument overlap process operates as described above. Following the connecting of items in the input chunk, some propositions are selected for holdover in the STM buffer. These items are therefore processed in more than one cycle. Only the propositions selected for holdover are available to connect the next input chunk of micropropositions to the last. Miller and Kintsch empirically determined that a buffer size of 2 is a reasonable approximation of the number of items a subject maintains in th working memory buffer. However, they reasoned that subjects resources are not as heavily taxed on the first processing cycle as on additional processing cycles. Thus, they expanded the size of the working memory buffer on the first cycle to 3. Miller and Kintsch used the following rules to select items for holdover. First, retain only those items located at high levels in the STM representation of the microstructure. In other words, first consider the Level 1 microproposition for holdover, then Level 2 etc. Miller and Kintsch used a recency rule. For example, if there are many items located at Level 2 in STM and the buffer has one remaining slot, the Level 2 items would be sorted in order of recency and the most recent proposition would be selected for holdover. For all other processing cycles, the input chunk of propositions are processed for argument overalp as in cycle 1. First, micropropositions which overlap with the items held over in the buffer from previous cycles are connected in STM. Then micropropositions sharing common referents with the micropropositions just input are added to the STM structure. Micropropositions are processed in this manner until all item in the input chunk are connected. Additionally, Miller and Kintsch included the Kintsch and vanDijk reinstatement and inference provisions in their model for cases when some or all the the propositions contained in an input chunk could not be connected in STM. They implemented a reinstatement procedure (searching the LTM representation of the microstructure for an item which shared a common referent with one of the unconnected propositions in the input chunk) in a top down, breadth first manner. If such a proposition was found, STM processing ceased and the matching routine was called on LTM. However, when the reinstatement search failed, an inference making procedure was simulated by constructing a new LTM graph. Following this, the argument overlap routine was called on LTM to transfer the STM contents. This routine operated on all items in LTM. Unfortunately, when this procedure is applied to lengthly texts, the top down, breadth first search often results in connecting new items to items which were not contained in the current input chunk or held over from previous processing cycles. Then items in STM were selected for holdover as described for cycle 1. Although buffer size was set at 2, Miller and Kintsch allowed for flexibility in the buffer size to accommodate embedded propositions and therefore include complete ideas. If one of the micropropositions selected for holdover contained one or more embedded propositions, they were added to the working memory buffer. If on later evaluation buffer size was greater than 3, the less recent embedded micropropositions were deleted until a buffer size of 3 was obtained. A flowchart of the Miller and Kintsch simulation can be seen in

Figure 1.

Insert Figure 1 about here

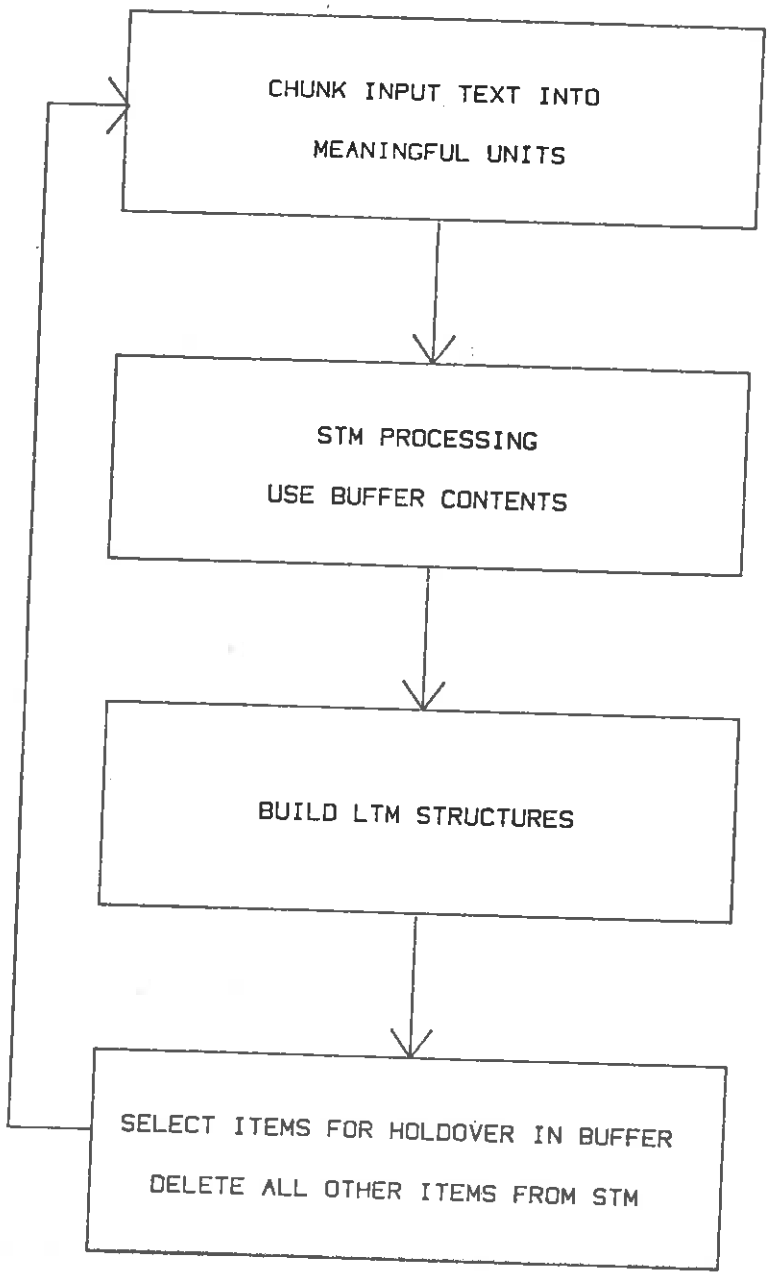
Simulation of Text On-line Processing (STOP).

While based on the work of Miller and Kintsch, the simulation designed to implement the theory of macrostructure (STOP) has altered the Miller and Kintsch simulation in many important ways which will be described at the end of this section. STOP is a processing simulation designed to create multiple interrelated LTM representations of a text. Specifically, STOP creates a LTM macrostructure representation of a text, and an interrelated LTM representation of the microstructure by simulating the encoding operations believed to occur during STM processing. As described in the previous section, recall of micropropositions is not dependent upon level in the microstructure alone, but rather level in the macrostructure of the associated macroproposition. Thus, the LTM representation of microstructure and macrostructure are not independent. Each macroproposition included in the macrostructure has a pointer to a microproposition in the microstructure. The propositions located beneath the pointed-to microproposition is the group of facts related to the particular macroproposition. The following is a description of the simulation used to implement the theory of macrostructure. The program uses a general pattern matcher and does not employ knowledge. This enables it to derive descriptors for all micropropositions and macropropositions for any text input to the simulation.

STOP takes two lists as input. One list contains micropropositions derived by the rules specified in Bovair and Kieras (1981) and chunked according to surface structure constituents. The second list contains macropropositions

Figure 1

Flowchart for Miller and Kintsch Simulation



which can be derived from either a knowledge constrained application of vanDijk's (1977) macro-rules or obtained from subject text summaries.

STOP uses macropropositions which are represented in a somewhat general form. The predicate states the macro-rule used in generating the macroproposition and the arguments are the numbers of the micropropositions to which the rule was applied. Thus, an example of a macroproposition might be (M7 (GENERALIZATION P7 P9 P11)). This means that macroproposition 7 is a generalization of micropropositions 7, 9, and 11. The argument list for each microproposition is ordered by importance to the gist of the macroproposition. Here, importance is operationally defined in terms of common referents between the macroproposition as represented in words and the arguments of the micropropositions used in forming it. It should be noted that the actual macropropositions a reader forms while reading are a function of both his/her familiarity with the subject matter and his/her goals. For this reason, texts upon which to test the model's predictions were very carefully selected so that it was most unlikely that the reader would possess an extensively developed knowledge base of similar or related information. Furthermore, the subject was provided with recall instructions to standardize reading goals.

At the beginning of the first cycle, the buffer and both long term memory structures are empty. The complete list of micropropositions is read into a queue, and the macroproposition list is read into a list. Then the first chunk of micropropositions is removed from the front of the queue and placed in an input chunk. Following this, all macropropositions in the macroproposition list are evaluated by their argument lists to determine which ones share common arguments with the chunk of microproposition in the input chunk. As macropropositions are represented by the macro-rule used to generate the macroproposition followed by the micropropositions to which the rule was applied, the search for macropropositions sharing arguments with the micropropositions in a chunk is actually a search of each macroproposition's argument list. The macroproposition created from micropropositions occurring first in the text is placed at the root node of the LTM macrostructure tree, or Level 1. Then, the processed macroproposition is removed from the list of all remaining macropropositions. Additionally, a pointer is established between the macroproposition (located at Level 1 in the macrostructure) and the first microproposition in its argument list. By establishing a pointer between the macroproposition and an associated microproposition, it is implied that the microproposition being pointed to was most important in the creation of the macroproposition. This macroproposition is returned to the main program as the last macroproposition added to memory.

As seen in Figure 2, following the selection of a macroproposition to be

----

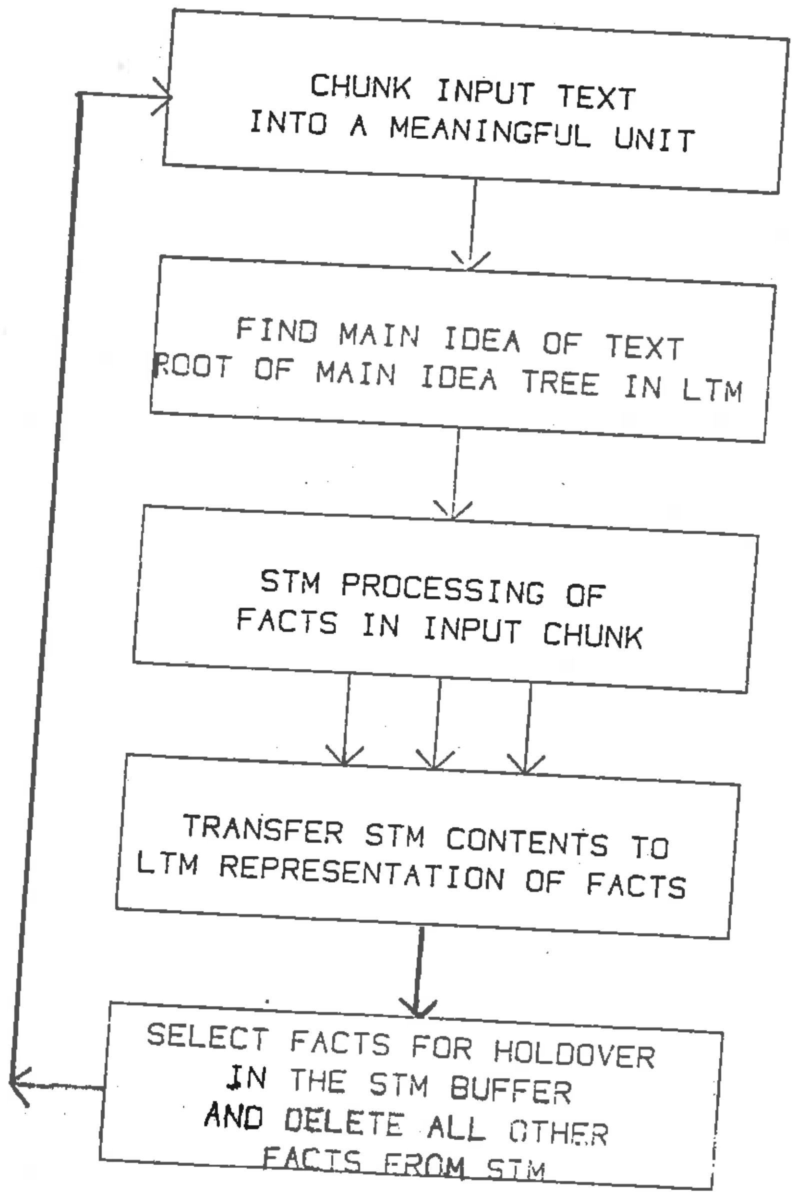
Insert Figure 2 about here

----

placed at Level 1 in the macrostructure, the program begins building a long term memory representation of the microstructure by use of a working memory buffer. First, it chooses a microproposition from the input chunk to be placed at Level 1 of both the working memory buffer and long term memory microstructure trees. The microproposition chosen is the one pointed to by the macroproposition at the macrostructure tree root node. This microproposition is then removed from the input chunk and the rest of the propositions in the chunk are added to the STM tree on the basis of argument overlap with the microproposition at the microstructure tree root node. All propositions which have arguments in common with the microproposition at Level 1 in the microstructure tree, or are embedded within this superordinate microproposition are placed at Level 2 of the microstructure tree and connected to the microproposition at Level 1. These are then removed from the input chunk. Following this, all micropropositions

Figure 2

Flowchart for Cycle #1



remaining in the input chunk are compared with each of the propositions at Level 2 in order of recency for argument overlap as operationally defined above. If they share common referents with the propositions located at Level 2, they are placed at Level 3 and connected to the proposition they have common arguments with at Level 2 or are embedded within. Again, connected micropropositions are removed from the input chunk. It should be noted that if a proposition overlaps with more than one proposition at Level 2 (in this instance) the microproposition is connected to the most recent Level 2 microproposition it overlaps with. This comparison process proceeds until all propositions in the input chunk have been connected to the tree representation in STM.

Following the above procedures, the contents of STM are transferred to the LTM microstructure representation. The Level 1 microproposition is located at Level 1 in LTM. Then all micropropositions connected to the Level 1 microproposition in STM and located at Level 2 in STM are placed at Level 2 in LTM with a pointer between them and the Level 1 microproposition. Then each of the Level 2 micropropositions in STM is assessed to determine if they point to any subordinate micropropositions in STM. If so, their Level 3 subordinates are placed at Level 3 in LTM. This process continues until all of the micropropositions in STM are evaluated to see if they point to any subordinates and the subordinates are transferred, or in other words, until all items in STM have a representation in LTM. Then the working memory buffer is emptied of all micropropositions except those chosen to be held over so as to assist in processing the next chunk of micropropositions.

Two special provisions are made in determining what is held over in the working memory buffer for the first processing cycle. First, buffer size is expanded so that one macroproposition and two micropropositions can be maintained. Buffer size is dynamically determined on the first cycle by adding 2 to the number of arguments in the macroproposition. The macroproposition is maintained in the working memory buffer by maintaining its microproposition argument list. Thus, all micropropositions which were contained in STM during the just completed processing cycle are scanned to see if they are also a member of the argument list of the first macroproposition added to memory. If so, they are maintained in the working memory buffer along with any propositions which may be embedded in them. Then two additional micropropositions are selected for holdover on the first processing cycle by the following procedure. First, these micropropositions cannot be arguments of the macroproposition being maintained. Second, the group of micropropositions satisfying the above criteria are assessed for their level in STM. The two micropropositions which are located at the highest levels of STM are selected. If this fails to narrow the holdover candidates down to two because a number of possible candidates are located at the same level in STM, then the micropropositions which were the most recent are selected.

A second special provision is made for embedded micropropositions. Because micropropositions containing embedded propositions are usually incomplete ideas, buffer size has been made flexible to accomodate the additional of one embedded microproposition, should it exist. When a microproposition is embedded in the micropropositions held over in the working memory buffer it too is added to the buffer. This provision is actualized by maintaining all embedded micropropositions in the buffer However, if upon later evaluation it is revealed that the buffer size exceeds the permitted holdover allowance by more than one proposition, then embedded micropropositions are deleted from the buffer in order of primacy. Deletion is performed in accordance with the following rules. First, any proposition which is also an argument of the last macroproposition input to the macrostructure is <u>never</u> deleted from the temporary buffer. The only items which can be deleted are micropropositions embedded in the micropropositions which do not represent the last macroproposition. These are deleted on the basis of primacy, so that the remaining embedded microproposition is also the most recently input proposition. All micropropositions which are retained in the buffer are tagged with a cycles property. This property represents the increased trace strength of the microproposition in LTM as a function of the additional processing it received in STM.

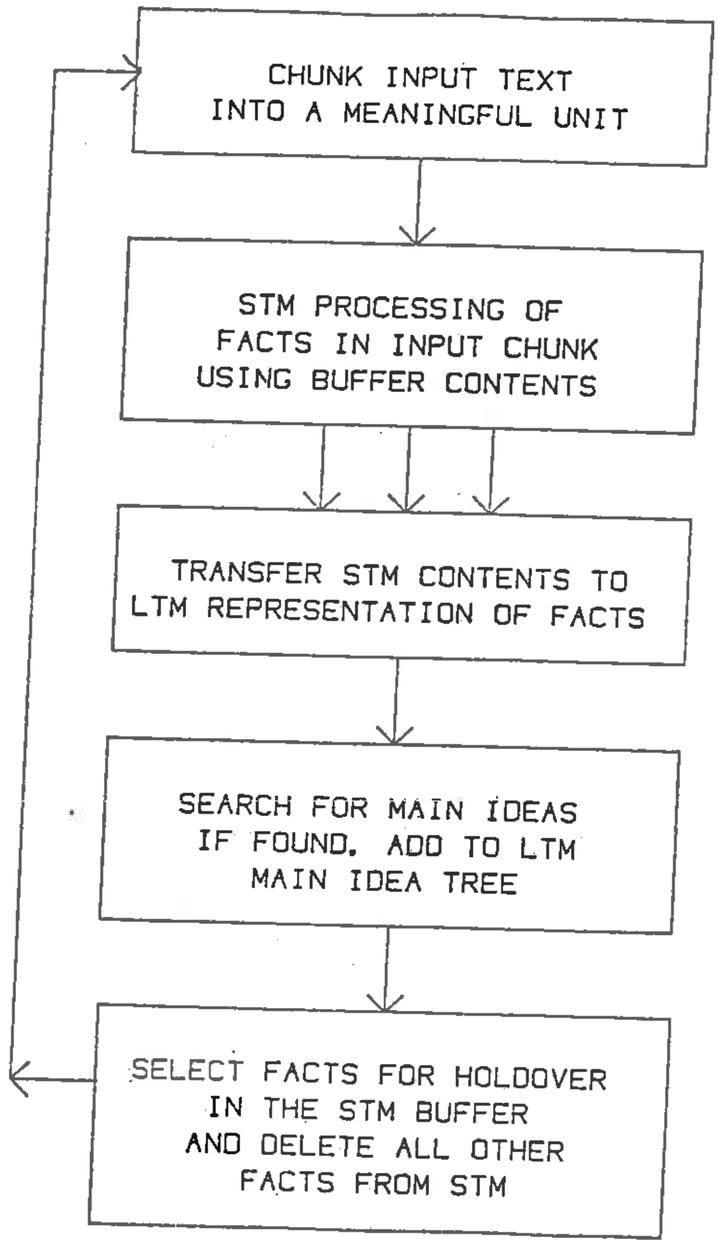
For all cycles following the preliminary one, the program flows as described below (see Figure 3). First, the input chunk is filled with micropropositions. Then the micropropositions in the input chunk are added to the microstructure by first processing them for argument overlap in STM with the micropropositions held over from the previous cycle in the working memory buffer. Then unconnected micropropositions are connected to the new micropropositions which were connected to the held over items. This process continues as described above for the first cycle.

Insert Figure 3 about here

If there exist micropropositions which cannot be added to the structure in STM because they share no elements in common with any of the propositions in STM, the model performs two operations to connect the remaining propositions. As described in Miller and Kintsch (1980), the first step taken to correct the problem is a long term memory reinstatement search. This alternative searches previously processed microproposition contained in the LTM microstructure to see if the unconnected micropropositions overlap with them. This search is conducted in a top down, breadth first manner. If an unconnected microproposition does overlap with previously processed micropropositions, the processed microproposition is reinstated into working memory and the unconnected microproposition is connected to it and removed from the input chunk. If some items from the input chunk were connected in STM prior to conducting a

Figure 3

Flowchart for all Cycles Except #1



reinstatement search, the items which were connected are transferred to LTM by the procedure described later. Then all micropropositions remaining in the input chunk are placed on the top of the stack of all unprocessed micropropositions, and the holdover selection process is bypassed, as the reinstated microproposition is placed at Level 1 of STM. If however, no items from the input chunk had been connected prior to conducting a successful reinstatement search then the buffer is emptied, the cycle properties of all held over micropropositions is decremented by one, and the reinstated proposition is placed at Level 1 of STM and processing continues. In all cases, when a microproposition is reinstated from LTM its cycle property is incremented by one. The combination of maintaining the last macroproposition in the working memory buffer and conducting reinstatement searches in a top down, breadth first manner implies that when micropropositions are encountered they are first assessed for local topic continuity. If this fails, a common referent search is conducted by first looking at the overall text topic, followed by later local topics. The reason this can be implemented in a top down search is because the LTM microstructure is sensitive to when an item was input.

If the reinstatement procedure fails to find any propositions sharing common referents with any of the unconnected micropropositions remaining in the input chunk, then a second problem solving mechanism must be employed. This involves the construction of a bridging inference. Unfortunately, the model at present does not specify what this inference is. Rather, it notes its occurrence and begins a new subtree for the propositions remaining in the chunk stack. The selection of a superordinate microproposition for placement at the root node of the new subtree is similar to the initial selection procedure because both are determined by the arguments of a macroproposition. However, in this case, selection is based on the last macroproposition added to the macrostructure. If there is no overlap between the micropropositions forming the argument list of the last macroproposition and the unconnected propositions in the input chunk, then all microproposition arguments of all macropropositions in the macroproposition list are evaluated to see if they overlap with the remaining micropropositions in the chunk stack. If so, a macroproposition is added to the macrostructure and the microproposition listed first in its argument list becomes the Level 1 microproposition in the new LTM tree.

It is hypothesized that subjects process information at the micro level while simultaneously searching for new main ideas to add to the macrostructure. Thus, following microstructure processing in the STM buffer, and prior to transferring the items in STM to the LTM microstructure, the remaining macropropositions in the macroproposition list are searched to see if any of their arguments overlap with the micropropositions in the current input chunk. If so, they are added to the macrostructure.

The rules for adding new macropropositions to the macrostructure are as follows. If a macroproposition shares arguments with only one other macroproposition in the macrostructure, then it is connected to that macroproposition and is located at the level below the level occupied by the macroproposition already in the macrostructure. If however, a macroproposition overlaps with more than one item already in the macrostructure and these items are located at different levels, the incoming macroproposition is connected to the item located closest to the root node (highest up in the tree). Thus, a top dwon search is employed to connect incoming macropropositions to the most important macroproposition sharing a common referent. This heuristic is implemented by retrieving the Level 1 macroproposition in the macrostructure and evaluating it for overlap. The evaluation for common referents is conducted by comparing all of the argument lists of the micropropositions which serve as the arguments for the macroproposition with the argument lists of all micropropositions in the argument list of the incoming macroproposition. If the evaluation reveals no overlap, then Level 2 macropropositions are retrieved and evaluated, and so on. Should a macroproposition overlap with more than one item already in the macrostructure and both items are located at the same level, then the incoming macroproposition is connected to the item which shares the greatest number of common referents with the incoming macroproposition. Thus, when a new macroproposition shares common referents with two macropropositions of equal importance, it is connected to the one it is most closely related to. The number of common referents is determined by evaluating all the argument lists of each of the micropropositions listed in each macroproposition's argument list. If the degree of overlap is determined to be equal (both macropropositions share an equal number of common referents with the incoming macroproposition) then the incoming macroproposition is connected to the macroproposition whose argument list overlaps with the most recent argument in the incoming macroproposition. Thus, if a new macroproposition overlaps with two macropropositions of equal importance with which it shares equal relationships with, then it is connected to the most recent of these. This means that if incoming macroproposition X overlaps with macropropositions Y and Z, the microproposition argument lists of Y and Z are ordered by primacy of initial input and the macroproposition with the highest numbered microproposition argument is selected to point to the incoming macroproposition, X. When an incoming macroproposition does not overlap with any of the items already contained in the macrostructure representation of the text, then it is placed at the lowest possible level in the tree and is marked with a special property. By placing the item at the lowest level of the tree, we ensure that its recall probability is lower than that of all other macropropositions. Furthermore, its probability of eliciting the recall of associated micropropositions is also lower than that of all other micropropositions, as microproposition recall is a function of the level the associated macroproposition occupies in the macrostructure.

Should the search for macropropositions in the processing cycle succeed, then the new macroproposition is added to the macrostructure according to the above rules and deleted from the macroproposition list. Furthermore, the last macroproposition added to the macro-representation is returned to the main program. However, if the search for macropropositions fails, then no change is made to the last macroproposition stored in the main processing loop. Following this, the micropropositions in STM are added to the LTM microstructure.

The procedure for adding micropropositions to the LTM microstructure representation is as follows. First, the Level 1 microproposition is located in LTM and its level in the LTM microstructure is noted. Then all Level 2 micropropositions in STM are assessed to see if they are already represented in the LTM microstructure. If not, they are added to LTM at 1 + the level occupied by the microproposition located at Level 1 in STM. Thus if P36 was located at Level 12 in the LTM microstructure and located at Level 1 in STM, then any item located at Level 2 in STM which was not already represented in LTM would be placed at Level 13 in LTM. Additionally, a pointer is placed between the two items. The above procedure continues until all the items in STM have a LTM representation in the microstructure. Then, micropropositions are selected for holdover in the working memory buffer.

The procedure used to determine buffer size is to count the number of arguments to the last macroproposition added to the macrostructure and add one to that number. Then the micropropositions which form the argument list of the last macropropostion are maintained along with one additional microproposition as described for cycle 1. The embedding provisions described for cycle 1 also apply on all other cycles. However, on all other cycles buffer size is initially set to 1 + the number of arguments in the last macroproposition. Furthermore, if a new macroproposition was descerned during the current processing cycle, the holdover heuristics are resticted to selecting an additional microproposition which was a member of the current input set to assure that subtrees contain only micropropositions related to a macroproposition.

When the program has completed the processing of a text, it prints out a number of microproposition and macroproposition properties which describe the processes an individual item participated in as well as where it is located in the LTM representations of the text. First, the program prints a list of the root nodes of all the microproposition subtrees it created during processing. Second, the level of the macroproposition pointing to the subtree root node is printed as well as the macrostructure hierarchy. Thus STOP print the level each macroproposition occupies and which macropropositions are its siblings. Third, a list of all micropropositions which were maintained in the working memory buffer for at least one additional processing cycle is printed along with the number of times the micropropositions were held over or reinstated. Fourth, the program produces a list of the micropropositions located at each level in the LTM microstructure as well as the micropropositions each microproposition points to or is superordinate to. Finally, STOP produces general statistics for the text. It prints out the number of reinstatement searches conducted, the number of inferences performed, the number of times the working memory buffer was stretched, and the number of times the working memory buffer was overloaded.

Contrasts Between STOP and the Miller and Kintsch Simulation.

The first major difference between STOP and the Miller and Kintsch simulation is that in addition to building a macrostructure representation of a text, STOP has deleted dependence upon user interaction. Rather than having the user provide a superordinate microproposition at the beginning of the program, STOP utilizes the first macroproposition to guiding the selection of a superordinate microproposition. Basically, the first process performed is the selection of a candidate main idea which represents the topic of the text. This main idea is synthesized from the micropropositions in the first input chunk. As macropropositions are represented by the macrorule used to generate the macroproposition followed by the micropropositions to which it was applied, the first microproposition to which the rule was applied is used as the superordinate item in the microstructure representation.

The second major modification to the Miller and Kintsch simulation involves the manner in which items from the current input chunk are connected to the LTM microstructure representation of the text. Miller and Kintsch used the same matching procedure which was used to process the items in STM when creating the LTM microstructure representation. However, as the number of micropropositions in a text increases, the probability of finding many items in LTM which overlap with a new microproposition increases drastically. Because the Miller and Kintsch argument overlap routine searches for overlapping items in a top down manner, frequently an item X which was connected to item Z in STM gets connected to item Y in LTM because X overlaps with both Y and Z and Y is located higher up in the microstructure. To model the transfer from STM to LTM, the approach of applying the pattern matching routines to LTM was abandoned. Rather, a transfer routine was developed which found where in the LTM representation of the microstructure the top most microproposition in STM was located. Then it connected all items at Level 2 in STM to the LTM representation of this item, provided that these Level 2 micropropositions were not already included in the LTM representation of the microstructure.

As a result of developing the transfer routine, the manner in which Miller and Kintsch implemented reinstatement searches had to be altered. Their heuristic was to transfer all items in the input chunk to LTM (including those which could not be connected in STM) provided that the reinstatement search succeeded. Since their pattern matching routine was not applied to LTM in this simulation, the following heuristics were used. First, as in Miller and Kintsch, if none of the items in the input chunk could be connected to the items held over in the STM buffer and a reinstatement search was successful, the reinstated proposition was placed at Level 1 in the STM buffer. However, all items previously held over were discarded from STM when a successful reinstatement search occurred. Then, the STM pattern matching routine was evoked and processing continued. However, when some of the items in the input chunk had been connected to the propositions held over in the buffer, a different heuristic was employed. The new propositions which had been connected were transfered to the LTM representation of the microstructure. Those incoming micropropositions which could not be held over were placed back on the stack and became the next chunk of propositions to be input for processing. Additionally, the normal holdover which would have taken place following the transfer of items to LTM was eliminated, and the proposition which was reinstated was placed in the buffer to connect the rest of the propositions to.

The final modification of the Miller and Kintsch simulation involved the selection of micropropositions to be held over in the working memory buffer. As opposed to finding 2 to 3 micropropositions located at Levels 1 and 2 in STM, the following heurisitc was employed in this model. The first item selected for holdover was the last macroproposition added to the macrostructure, or local topic. Then an additional microproposition which was not an element upon which the macrorule used in the last macroproposition was applied was chosen. The choice of this item was dependent upon whether a new macroproposition was added during the processing cycle, as describe in the theory of macrostructure.

The Experiment

Several testable predictions are made by the theory of macrostructure. First, recall of micropropositions is influenced by the subtree in which the microproposition is contained. Second, the level within a subtree occupied by a microproposition should affect its probability of being recalled. Third, as the number of cycles a microproposition is held over in the short term memory buffer increases, recall probability should increase. Fourth, subtree and level within a subtree should interact. Specifically, the lowest levels of subtrees associated with very unimportant macropropositions should have considerably lower recall probabilities than the lowest levels of subtrees associated with more important macroproprepositions. In this experiment, all four predictions were tested. Texts were input into the simulation and the descriptors generated by STOP were compared with those from empirical data. More specifically, a number of texts were input to STOP in order to derive descriptive measures of each proposition's level in the microstructure, subtree it was contained in, level within a subtree, and number of processing cycles. These measures were then compared with the recall protocols generated by a sample of subjects.

A reading rate manipulation was included to investigate whether subjects adopted a strategy of selectively devoting resources to processing main ideas under limited reading time conditions, a suggest by the work of Young, Angell, and Bourne (1984) and Young (1984).

Following the recall task, subjects ranked the importance of select micropropositions which were associated with macropropositions of varying importance. The rationale underlying this portion of the study was to provide an additional means of evaluating the relative importance of macrostructure level and microstructure level in recall. Specifically, the micropropositions used in the ranking task were located at Level 1, or the root node, of various subtrees located at different levels within the microstructure. Four micropropositions were ranked for each text analyzed by STOP. Two of the micropropositions for each text were associated with macropropositions of secondary importance, while the remaining two micropropositions were associated with macropropositions of tertiary importance. One microproposition associated with each of the macrostructure levels was located near the top levels of the overall microstructure hierarchy, while the second was located near the bottom of the hierarchy. Thus, macrostructure importance and microstructure level were crossed.

Method

Subjects. All ninety subjects were recruited from a newspaper ad placed in the Colorado Daily, the University of Colorado's student newspaper. Each subject was paid $5.00 for participation in the study. Subjects were randomly divided into two groups of equal size.

Design. There were two parts to the study. Part one used a within proposition design. One group of subjects read at 350 msec/word, while the other group read at 125 msec/word to see if subjects adopted a strategy of devoting processing resources to the derivation and identification of main ideas under limited reading time conditions, facilitating a "macro-levels effect". Rates were determined by having a random sample of 10 pilot subjects report comfortable reading rates as well as comprehensible but uncomfortable rates. This was especially important as it was necessary for all subjects to be able to comprehend in the fast condition. Thus, one within proposition variable was employed, Reading Rate. Additionally, four between proposition variables were used, Text, Subtree, Level within a Subtree, and Number of Processing Cycles. Therefore, a 2 X 3 X 3 X 4 X 3 mixed design was employed in part 1. Order of target text presentation was counterbalanced within each reading rate group of subjects by using a 3 X 3 Latin Square.

In part two, all subjects in each reading rate condition ranked four micropropositions per target text for importance. Micropropositions which were the root nodes of subtrees associated with Level 2 and Level 3 macropropositions were chosen as items for rank ordering. Specifically, one of the micropropositions associated with Level 2 macroprositions was located high up in the microstructure tree and the other was located at a much lower level. The same microstructure considerations applied to micropropositions associated with Level 3 macropropositions. Additionally, all micropropositions were equated for number of processing cycles so that the cycling factor, k, was held constant. Thus, a 2 X 3 X 2 X 2 mixed design was employed. Reading rate group was crossed with the three within subject variables of text, macrostructure level and microstructure level.

Materials. Three target texts were used in this experiment. Texts were selected by having thirty voluntary undergraduate cognitive psychology students summarize six candidate target texts. Each text was five paragraphs long, of moderate difficulty and adapted from essays published in the magazines High Technology, Omni, or Discover. Texts were approximately two double spaced typed pages in length. Subjects read each of the six essays and wrote a six line summary of each text immediately following each reading. Order of texts was counterbalanced by using a 6 X 6 Latin Square. Subjects read the texts at a self determined reading rate. All main ideas mentioned by subjects for each text were listed and the number of subjects who mentioned each main idea was tabulated. Then two criteria were used to narrow the number of texts. First, texts which contained more than 15 different main ideas were eliminated from consideration as target texts. Using this criterion, two of the six candidate texts were eliminated. The four remaining texts were input to the simulation using macropropositions which were chosen by selecting macropropositions which were mentioned by at least 15 of the 30 subjects. Next, the macrostructure generated by the simulation for each text was considered. For purposes of the rating task, each text was required to have at least two main ideas located at Levels 3 or below in the macrostructure tree. This procedure resulted in the elimination of one of the four remaining candidate texts. Three remaining texts which satisfied all criteria were chosen as the target texts. Texts were propositionalized according to the guidelines of Bovair and Kieras (1981). Each text contained approximately 125 micropropositions and 8 macropropositions. The texts, their micropropositions and macrostructures can be seen in Appendices 1,and 3 - 10.

Thirteen additional distractor texts were used in this experiment. Each text was an essay taken from the same pool of texts contained in either High Technology, Omni, or Discover magazines. All texts averaged two double spaced typed pages in length.

Procedure. Each subject was asked to read the 16 texts (three targets and 13 distractors). Each text was presented one clause at a time (RSVP clauses) for the amount of time per word alloted the subject given his/her respective group times the number of words in each clause. RSVP clauses were used to ensure that subjects in both reading rate groups read all of the textual material, as evidence suggests that rapidly reading subjects are more likely to view all parts of a text when small amount of text are presented to a stationary position in the viewing field (Potter, Kroll & Harris, 1980). However, the size of the presentation unit can also have a profound effect on comprehension (Young, 1984). When single words are presented to a stationary position in the viewing field (RSVP words) comprehension is degraded when presentation rate is very rapid (i.e. 600 wpm) and pauses aren't inserted at sentence boundaries (Masson, 1983). On the other hand, RSVP clauses promote integration of textual material and does not degrade comprehension relative to reading normally presented text (Young, Angell & Bourne, 1984; Young, 1984).

Texts 1 - 10 were not target texts. Rather, they were used to stabilize any possible strategies the subject might be employing and to allow the subject to adjust to reading clauses on a CRT screen. The next three texts (numbers 11 - 13) were the target texts. The last three texts served as additional distractors. Texts were presented at either 350 msec/word or 125 msec/word. Subjects were informed about limited reading time and were told to read carefully in order to answer questions which might be posed. Two questions about the contents of texts 1 - 10 were included. These appeared after each text but not immediately following each reading. To illustrate, after subjects had read texts 1 through 3 they were asked two questions about story 1 only. Questions about essay 2 were posed after subjects read essay 4. Subjects read texts 5 and 6 and then answererd questions about text 3. This type of procedure was used throughout the experiment, such that subjects were required to answer questions about filler texts after reading target texts. It should be noted that subjects were never asked questions about the target texts which they were later required to recall. All questions posed related to the first ten distractor texts. In the actual test of recall, an incidental recall protocol was collected for each of the three target texts (texts 11 - 13) Recall tests were administered following completion of text 16.

In sum, subjects read 10 filler texts followed by the three target texts and finally the last 3 filler texts. In between target texts, subjects answered questions about distractor texts 1 - 10 at their own rate. Following this, subjects were asked to recall the three target texts in order of occurrence. Subjects were unaware of the fact that they would be asked to recall the three target texts until they had completed reading all 16 essays.

The recall directions were as follows: "Now we would like you to recall everything you can remember about the essay on _____ (one word topic). Please try to be as thorough and comprehensible as possible. Now please turn to the next page of your question booklet and use this page to write down everything you remember about the ____ text." Finally, subjects were instructed to follow the directions at the top of each of the remaining pages in the question booklet. The remaining pages each contained directions for rank ordering the four statements (propositions) contained on each page. A rank of 1 indicated that that the statement was most important to the gist of the text while a rank of 4 indicated that the subject felt the statment was the least important.

Results and Discussion

Analysis of Recall Data. The recall data obtained in part 1 of this study were analyzed in two ways. First, a one way, between propositions analysis of variance was conducted on the variable microstructure level. This analysis was performed to assess the validity of the microstructure levels hypothesis on the textual materials used in the present study. Second, a 2 X 3 X 3 X 4 X 3 mixed analysis of variance was performed to assess the within proposition effect of reading rate, and the between proposition effects of target text, subtree, level within a subtree, and number of processing cycles, respectively, on the probability of recalling a microproposition. These later four items had been generated by the computer simulation, STOP, which was an implementation of the theory of macrostructure. It should be noted that empty cells result from the interactions of subtree by level within a subtree by processing cycles, text by level within a subtree by processing cycles, text by subtree by level within a subtree by processing cycles, and the five way interaction. Therefore, the three way interaction of subtree by level within a subtree and processing cycles was deleted from the analysis and added to the error term as well as all three way interactions involving text, and all four and five way interactions.

The one way analysis of variance designed to assess the relevance of the microstructure hypothesis under the specific conditions delineated by this experimental design indicated that the effect of level in the microstructure was significant, $F(15,719) = 8.63$, $p < .001$. As shown in Table 2, the recall

----

Insert Table 2 about here

----

probabilities for each level of the microstructure are statistically equivalent with the exception of Levels 1 and 2 in the microstructure. Tukey HSD tests reveal that Level 1 in the microstructure significantly differs from all other levels of microstructure at the .05 significance level. Additionally, micropropositions located at Level 2 of the microstructure differ significantly from those located at Levels 4 through 11, $p < .05$. As the microproposition located at Level 1 in the microstructure is also the microproposition pointed to by the Level 1 macroproposition, this finding is to be expected according to the theory of macrostructure proposed herein, as the highest recall probabilities should be associated with micropropositions located at Level 1 of subtrees associated with the most important macropropositions.

Thus, the microstructure hypothesis is not a sufficient explanation for the data obtained in this study. The effect of microstructure only varied for the Level 1 and 2 micropropositions, rather than at all levels, as the microstructure hypothesis suggests. As the microproposition located at Level 1 in the microstructure is also the proposition located at Level 1 of the subtree of micropropositions related to the most important macroproposition.

A 2 X 3 X 4 X 3 X 3 mixed analysis of variance was performed on the variables reading rate, subtree, level within a subtree, cycles and text to test the four hypotheses of interest. To test hypotheses 1, the probability of

Table 2

Microstructure Level Effects

Level in Microstructure	Mean Recall Probability	Significance
1	.7033	**
2	.4091	*
3	.2345	
4	.2091	
5	.1929	
6	.1710	
7	.1421	
8	.1393	
9	.1710	
10	.1959	
11	.1968	
12	.2257	
13	.1450	
14	.0200	
15	.0550	

** differs significantly from levels 2-15
* differs significantly from levels 4-15

recalling a microproposition increases as the macroproposition associated with the subtree containing the microproposition increases in its importance to the overall thrust of a text, the effect of subtree was assessed. Findings indicated that the effect of subtree was significant, $F(2,568) = 25.23$, $p < .0001$. The probability of recalling a microproposition given the subtree in which it was contained increased as a function of the importance of the macroproposition pointing to the subtree , as denoted in Figure 4. The mean

----

Insert Figure 4 about here

----

probability of recalling a microproposition associated with a Level 1 macroproposition was .26, while the probability of recalling micropropositions associated with Level 2 and 3 macropropositions was .18 and .15, respectively. Tukey tests applied to this effect reveal that micropropositions located in subtree 1's have a significantly higher probability of being recalled that micropropositions located in Level 2 and 3 subtrees, $p < .05$. Figure 4 shows the means associated with each subtree for each reading rate. As the effect of subtree was not moderated by a rate by subtree interaction, $F(2,568) = 0.14$, $p > .87$, the only reading rate differences to be observed is the overall recall probability associated with each subtree.

Hypothesis 2 proposed that the level within a subtree is an important predictor of recall probability. Findings indicated that the level a microproposition occupied within a subtree was significant, $F(3,568) = 47.92$, $p < .0001$. The closer a proposition was to the root node, or Level 1, of a subtree, the higher its probability of being recalled (see Figure 5). Tukey

----

Insert Figure 5 about here

Figure 4

Effect of Subtree

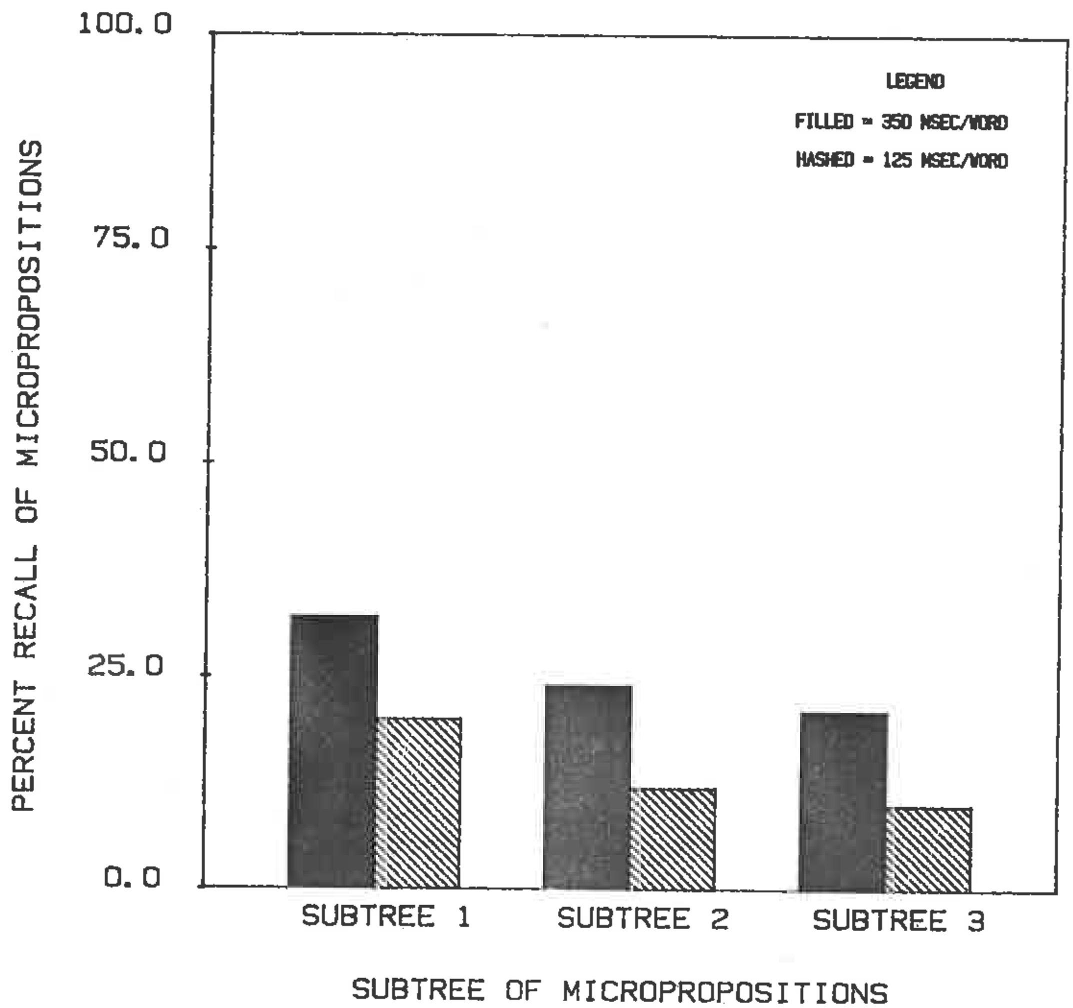
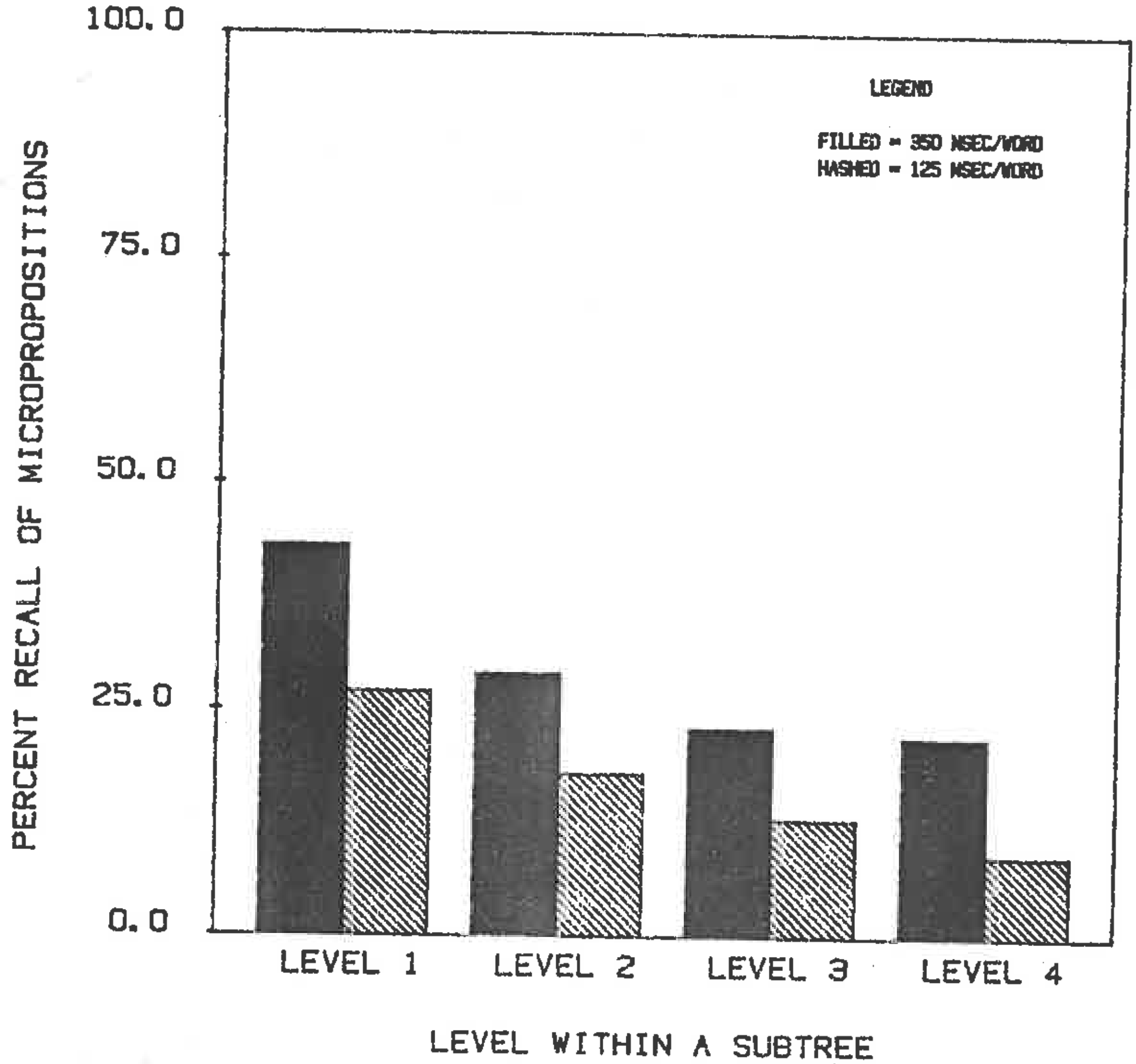


Figure 5

Effect of Level Within a Subtree



analyses revealed that the probability of recalling a microproposition located at Level 1 in a subtree differed significantly from the probability of recalling a microproposition located at Levels 2, 3, or 4 of a subtree, $p < .05$. Also, micropropositions located at Level 2 of a subtree have a significantly higher probability of being recalled than micropropositions located at levels 3 or below, $p < .05$. As the distance between the root node of the subtree and the level a microproposition occupied within the subtree increased, its probability of being recalled decreased. As seen in Figure 5, the effect of level a microproposition occupies within a subtree of facts associated with a given macroproposition does not vary with respect to reading rate, $F(3,568) = 1.39$, $p > .25$.

Hypothesis 3 proposed that as the number of processing cycles a microproposition participates in increases, its probability of being recalled increases. Findings indicated that the number of processing cycles a proposition participated in was a significant factor in its probability of being recalled, $F(2,568) = 61.60$, $p < .0001$. The mean recall probabilities were .15, .21, and .40 for propositions participating in 1, 2, and 3 or more processing cycles, respectively, (see Figure 6). The results of Tukey tests revealed that

----

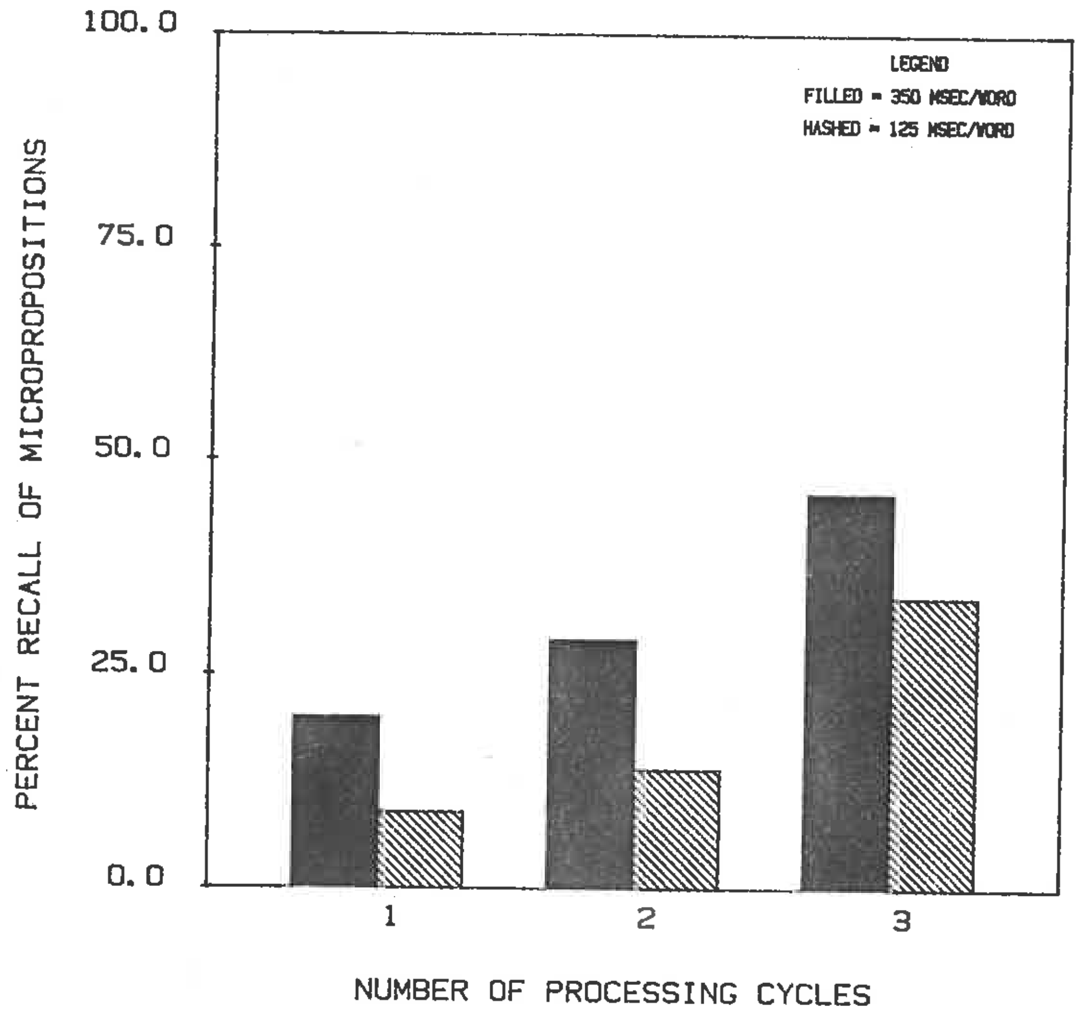
Insert Figure 6 about here

----

all differences between number of processing cycles was significant at the .05 level. As indicated by Figure 6, the relative increase in recall probability associated with each additional processing cycle a proposition participated in, differed only in magnitude when reading rate is considered, $F(2,568) = 1.62$, $p > .20$.

Figure 6

Effect of Processing Cycles



The reading rate manipulation was included to determine whether the levels effect associated with subtree and level within a subtree would be maintained at rapid reading rates, as suggested by the findings of Young, Angell and Bourne (1984). Overall, findings indicated that the effect of reading rate was very significant, $F\ (1,568) = 128.66$, $p < .0001$. Subjects who read at fast rates (125 msec/word) recalled propositions at an average probability of .13, while subject who read at fairly normal rates (350 msec/word) recalled propositions at an average probability of .25. While subjects recalled fewer micropropositions when reading at rapid rates, Figures 4, 5, and 6 indicate that the patterns of recall associated with rapid reading did not vary from the patterns manifested at slower reading rates. The increasing recall probability associated the increasing importance of the macroproposition associated with a subtree persisted at rapid reading rates. This finding suggests that when subjects are required to read at rapid rates, they adopt a reading strategy of devoting processing resources to deriving main ideas. Additionally, the increase in recall probability associated with higher level within a subtree persisted at rapid reading rates, as did the processing cycle effect.

The probability of recalling a microproposition did not differ among texts, $F\ (2,568) = 1.53$, $p > .21$. Overall, the probabilities associated with recalling a proposition contained in the three target texts were .19, .22, and .18, respectively. However, there were a number of interactions with text. For the main effects already addressed, there were significant text interactions with both the effects of level within a subtree, $F\ (6,568) = 2.41$, $p < .05$, and number of processing cycles a microproposition participated in, $F\ (4,568) = 4.97$, $p < .01$. These significant interactions represent unknown sources of variability for which the current theory has no explanation. They are depicted in Tables 3 and 4.

Insert Tables 3 and 4 about here

The last hypothesis proposed that subtree and level within a subtree would interact as a decrease in recall probability was expected to be associated with descending levels within a hierarchical subtree would not be constant across subtrees associated with macropropositions of varying importance. The subtree by level within a subtree interaction was significant, $F\ (6,568) = 10.71$, $p < .0001$. As can be seen in Table 5, the probability of recalling a proposition is

----

Insert Table 5 about here

----

a function of both the level it occupies within a subtree and the importance of the macroproposition pointing to the subtree. These two factors interact such that the effect of level within a subtree varies as a function of the importance of the macroproposition pointing to the subtree. The simple effects analyses of level within a subtree for each level of subtree were significant for all subtrees, $F\ (3,154) = 18.12$, $p < .0001$, $F\ (3,443) = 4.52$, $p < .01$, $F\ (3,117) = 8.53$, $p < .0001$, for subtrees associated with Level 1, 2 and 3 macropropositions, respectively. However, the magnitude is smaller for subtrees associated with macropropositions of secondary importance relative to subtrees associated with macropropositions of primary or tertiary importance. Specifically, for subtrees associated with Level 1 macropropositions, the probability of recalling a microproposition located at Level 1 of the subtree differs significantly from the probability of recalling micropropositions located at all lower levels of the subtree, $p < .05$. Also, micropropositions located at Level 2 of subtrees associated with Level 1 macropropositions have a significantly higher probability of being recalled than items located at or

Table 3

Effects of Level Within a Subtree by Text by Rate

Level	350 Msec per Word Mean Recall Probability	125 Msec per Word Mean Recall Probability	Marginal
	Text 1		
1	.46	.27	.37
2	.33	.18	.26
3	.21	.11	.16
4	.19	.08	.14
	Text 2		
1	.34	.27	.31
2	.32	.26	.29
3	.23	.18	.20
4	.25	.13	.19
	Text 3		
1	.52	.25	.39
2	.30	.19	.25
3	.24	.13	.19
4	.21	.07	.14

Table 4

Effects of Processing Cyles by Text by Rate

Cycles	350 Msec per Word Mean Recall Probability	125 Msec per Word Mean Recall Probability	Marginal
	Text 1		
1	.20	.10	.15
2	.27	.11	.19
3	.45	.30	.37
	Text 2		
1	.20	.11	.15
2	.35	.23	.29
3	.40	.38	.39
	Text 3		
1	.20	.08	.14
2	.26	.09	.18
3	.51	.33	.42

Table 5

Effects of Level Within a Subtree by Subtree by Rate

Level	350 Msec per Word Mean Recall Probability	125 Msec per Word Mean Recall Probability	Marginal
Subtree 1			
1	.78	.63	.70
2	.56	.44	.50
3	.31	.23	.27
4	.23	.10	.17
Subtree 2			
1	.37	.23	.30
2	.26	.15	.22
3	.22	.14	.18
4	.22	.08	.15
Subtree 3			
1	.41	.18	.30
2	.31	.16	.24
3	.15	.05	.10
4	.14	.09	.11

below Level 4 of the subtree. For subtrees associated with Level 2 macropropositions, the probability of recalling a microproposition located at Level 1 of the subtree differs significantly from the probability of recalling items located at Levels 2 and below of the subtree, $p < .05$. Subtrees associated with Level 3 macropropositions show that the probability of recalling items located at Levels 1 or 2 of the subtree differ from the probability of recalling items located at Levels 3 and below of the subtree, $p < .05$. These significant differences between means suggest that the reader is likely to recall items located at the top of all subtrees. However, as the macroproposition associated with a subtree decreases in its overall importance to the gist of a text, the probability of recalling items located at lower levels of the subtree decreases. Subtrees associated with Level 2 macropropositions asymptote at a much higher probability than those associated with Level 3 macropropositions. Additionally, this interaction was not modified by rate, $F(6,568) = 0.42$, $p > .86$.

The level within a subtree by number of processing cycles interaction was reliable, $F(6,568) = 3.60$, $p < .01$, As seen in Table 6, the recall

----

Insert Table 6 about here

----

probabilities of micropropositions decrease as their level within a subtree decreases for micropropositions participating in one and two processing cycles. Additionally, the processing cycles effect is maintained at all levels within a subtree. However, the level within a subtree effect does not hold for items which participate in three or more processing cycles. Specifically, micropropositions which participate in three processing cycles and are located at Level 4 in a subtree of the microstructure have recall probabilities which exceed expectation. This may be due to the fact that occasionally, a

Table 6

Effects of Level Within a Subtree by Cycles by Rate

Level	350 Msec per Word Mean Recall Probability	125 Msec per Word Mean Recall Probability	Marginal
	Processing Cycles = 1		
1	.47	.19	.33
2	.26	.13	.19
3	.20	.11	.15
4	.19	.08	.13
	Processing Cycles = 2		
1	.34	.16	.25
2	.23	.10	.17
3	.28	.19	.23
4	.36	.09	.23
	Processing Cycles = 3		
1	.49	.33	.41
2	.49	.41	.45
3	.30	.20	.25
4	.57	.33	.45

microproposition which participates in the formation of a macroproposition and is not the most important microproposition in the macroproposition's formation is located at a level above the microproposition which is identified at the root node of a new subtree. Thus, the eight micropropositions responsible for this cell deviation would be left at Level 4 of the old subtree, yet cycled many times. Furthermore, this interaction was not modified by rate, $F\ (6,568) = 1.21$, $p < .29$.

In a similar vein, the subtree by processing cycles interaction was significant, $F\ (4,568) = 3.39$, $p < .05$. Again, as seen in Table 7, there are no

----

Insert Table 7 about here

----

cells which deviate from the general patterns expected from the processing cycles effect or the subtree effect. However, the magnitude of these differences does vary. The significance of this interaction is inflated due to the large number of degrees of freedom in the error term. Furthermore, this interaction is not moderated by reading rate, $F\ (4,568) = 1.22$, $p > .30$.

Regression analyses revealed that the combination of all factors assessed in the analysis of variance accounted for 71% of the variability in the probability of recalling propositions, $R2 = .71$. The adjusted R2's associated with the effects of subtree, level within a subtree, number of processing cycles, rate, subtree by level within a subtree, subtree by processing cycles, level within a subtree by processing cycles and subtree by level within a subtree by processing cycles were .063, .12, .11, .12, .054, .01, .02, and .056, respectively. Thus, reading rate and the factors hypothesized by the theory of macrostructure to affect the probability of recalling a microproposition collectively accounted for 55.3% of the total variance. The additional 15.7% accounted for by the total regression was due to the effects of interactions

Table 7

Effects of Subtree by Processing Cycles by Rate

Subtree	350 Msec per Word Mean Recall Probability	125 Msec per Word Mean Recall Probability	Marginal
	Processing Cycles = 1		
1	.20	.10	.15
2	.21	.10	.15
3	.14	.07	.10
	Processing Cycles = 2		
1	.37	.20	.28
2	.26	.11	.19
3	.27	.12	.20
	Processing Cycles = 3		
1	.64	.53	.58
2	.38	.29	.33
3	.40	.18	.29

with text and constants.

To overview the data analyses, significant results were obtained for all four hypotheses relating to microproposition recall. Namely, the probability of recalling an individual microproposition is a function of (1) the subtree in which it is contained, (2) the level it occupies within the subtree, (3) the number of processing cycles it participates in, and (4) the interaction of subtree and level within a subtree. Additionally, the reading rate findings indicated that subjects adopt a strategy of selectively devoting processing resources to deriving the main ideas of a text. Significant text effects were noted.

<u>Rank Order Data</u>. In the second phase of this study, subjects were asked to rank the importance of four micropropositions which were located at Level 1 of their respective subtrees. A four-point ranking scale was used, where a rank of 1 indicated most important and a rank of 4 indicated least important. This portion of the study was included to disambiguate any possible confoundings of the effects of the importance of a macroproposition pointing to a subtree and the position the subtree occupied within the overall microstructure hierarchy. Thus, two micropropositions which were the root nodes of subtrees associated with macropropositions of secondary importance were chosen from each of the three target texts. Of these, one was located high up in the overall microstructure hierarchy and the second was located in the lower protions of the hierarchies. Additionally, two micropropositions which were the root nodes of subtrees assocaited with macropropositions of tertiary importance were also chosen from each target text. Again, the microstructure level considerations described above were used here. The results of the rank ordering portion of the study converged nicely with the recall data. The data were analyzed using the Friedman Test for J Matched Groups. This nonparametric test was employed to

take into account the dependencies among ranks. The results of these analyses revealed that the effect of macrostructure level was significant (see Figure 7).

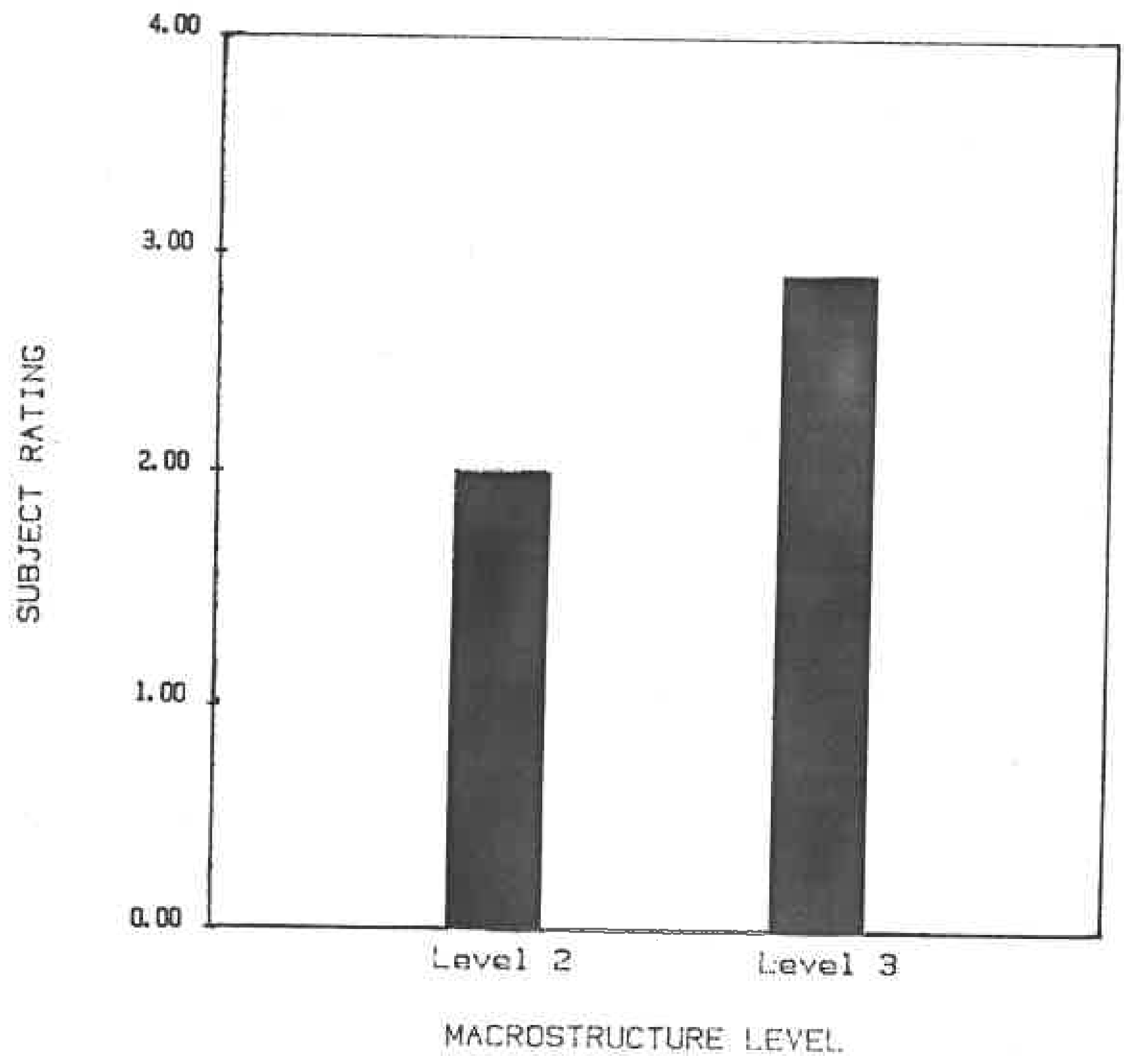
Insert Figure 7 about here

Specifically, the Chi-Square value for macroproposition level was 8,231.6, $p < .00001$. Specifically, the mean rank associated with macropropositions of secondary importance was 2.05, while the mean rank assocaited with macropropositions of tertiary importance was 2.95. Furthermore, it appears that this effect was not moderated by rate as the mean ranks associated with macropropositions of secondary importance are 2.10 and 2.01 for the regular and fast reading rate groups respectively. Additionally, the mean ranks assocaited with macropropositions of tertiary importance are 2.90 and 2.98 for the normal and rapid subjects, respectively.

The main effect for microstructure level, however, was not significant. The mean ranks associated with micropropositions located at high levels and low level of the overall microstructure hierarchy are 2.478 and 2.518, respectively. Additionally, these do not vary as a function of reading rate. The mean rank assocaited with micropropositions located at the upper levels of the hierarchy are 2.431 and 2.524 for the normal and fast readers, resepctively, while the mean ranks for micropropositions located at low levels of the hierarchy are 2.569 and 2.467. As the average rank for all texts was equal, this effect could not be assessed. Unfortunately, the testing of interactions with appropriate rank order statistics was not possible, as no appropriate rank order statisitics exist for analyzing more than one repeated measure. However there does not appear to be an interaction of microstructure level and macrostructure level, as can be seen in Figure 8.

Figure 7

Effect of Importance of Macropropositions on Ranking Task



Insert Figure 8 about here

There does appear to be an interaction between text and macrostructure level as can be seen in Figure 9. The rankings associated with Level 2 and

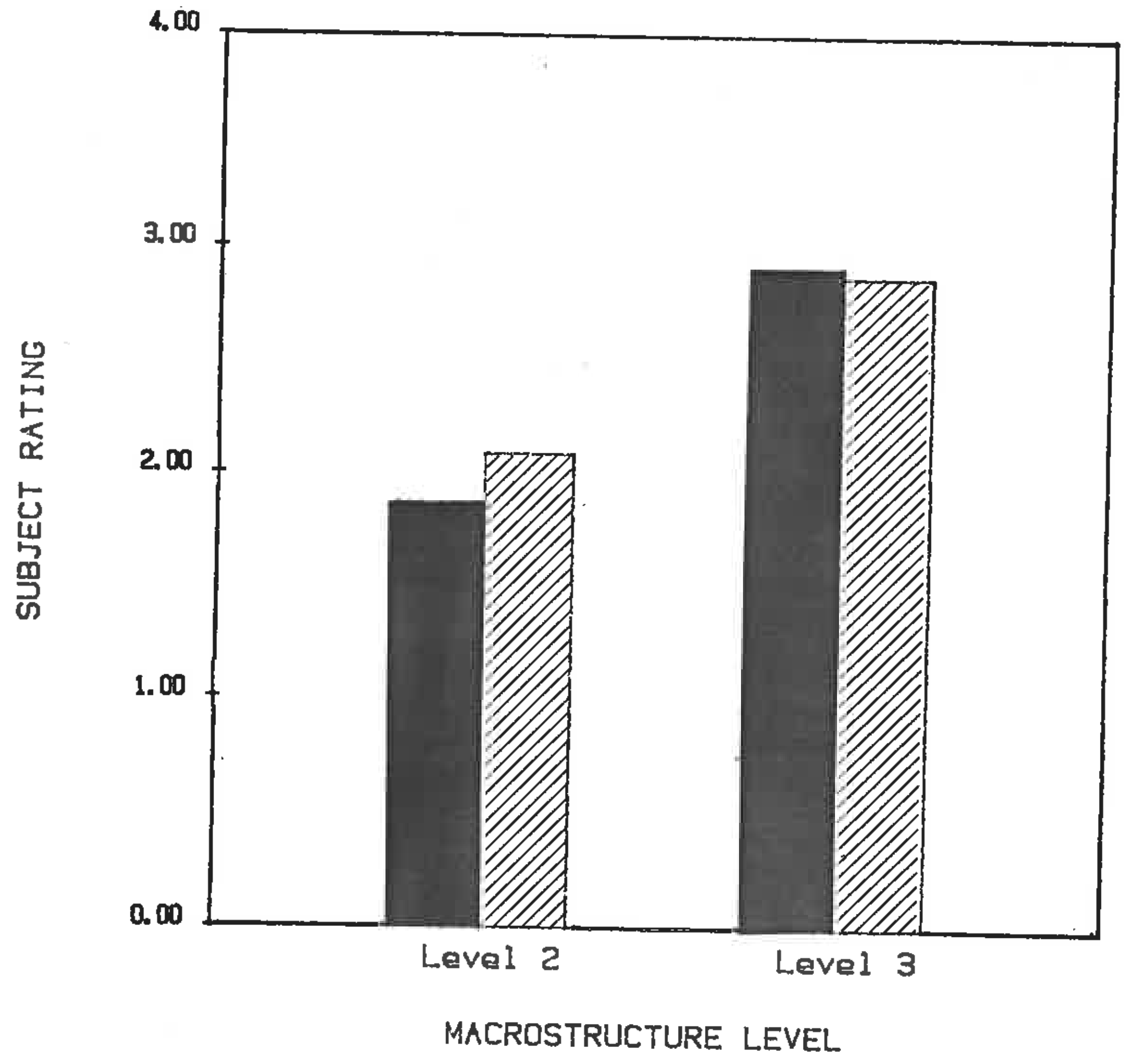
----

Insert Figure 9 about here

Level 3 macropropositions are most different on target text C, the Computer Crime text. In fact, these ratings are precisely what was expected. Propositions associated with Level 2 macropropositions were judged as much more important than propositions associated with Level 3 macropropositions by all 90 subjects. The rankings for target texts A and B (the Einstein and Chesapeake Bay texts, respectively) are not quite as clean as the rankings for target text C. However, the differences among macropropositions are significant for all texts. In each of target texts A and B, there was one microproposition which had a rating that diverged from that expected. In the Einstein text, the statement "There was a file on Einstein" received an average rank of 2.5. This statement is associated with a Level 3 macropropositions and is located at a low level in the overall microstructure. An explanation of this divergent ranking can be related to prior knowledge of the subject matter. Persons who were familiar with the McCarthy era would not be surprised at the fact that an FBI file was maintained on Albert Einstein. Therefore, text driven importance ratings would be expected. However, those subjects who were unfamiliar with the era would probably rate the above staement as being more important than a text based analysis would predict. Similarly, in target text B, the statement "Reagan ordered a cleanup" (of Chesapeake Bay) also received an average rank of 2.5. This is not surprising given President Reagan's environmental policies.

Figure 8

Macrostructure and Microstructure Effects on Ranking

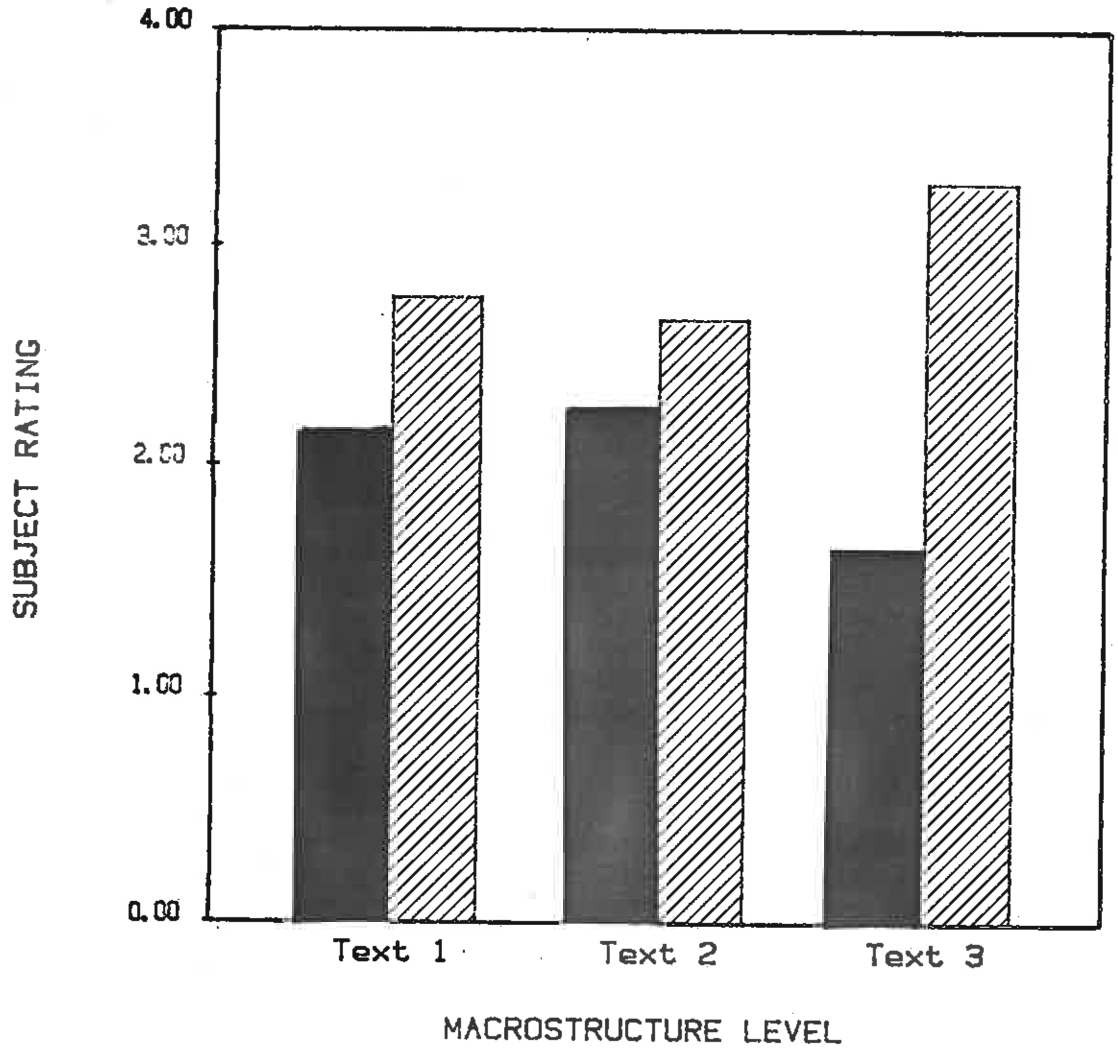


hashed = low in microstructure

filled - high in microstructure

Figure 9

Macrostructure Importance By Texts Effect on Ranking



Thus, the text by macroproposition level interaction can be attributed to prior knowledge about the subject matter which was not coded into the simulation predicting importance of macropropositions.

The results of the rank order analyses can be contrasted with the results obtained from the analysis of the recall protocols. In the ranking task, the effect of the importance of a macroproposition was highly significant. Only macropropositions of secondary and tertiary importance were included in the ranking task. In the recall data, the effect of subtree was also highly significant. However, Tukey analyses revealed that subtrees associated with macropropositions of secondary and tertiary importance did not differ significantly. Another factor which should be considered is that the items included in the ranking task were located at Level 1 of their respective subtrees. Therefore, the direct contrast of the subtree effects in these two tasks is not entirely appropriate, as the subtree effect in the recall data is summed over all levels within a subtree. A possible explanation of the discrepancy is that subjects are most likely to recall items directly associated with macropropositions, should the macroproposition be recalled at all. Additionally, the ranking data reflect subject's importance judgments.

Another contrast between tasks involves the effect of microstructure. There was no significant microstructure effect in the ranking data. This finding can be contrasted with the recall effects. Overall, the effect of microstructure was significant in the recall analysis. However, Tukey test applied to this effect revealed that micropropositions located at Levels 1 and 2 in the microstructure differed significantly from items located at lower levels.

General Discussion

The most significant and enduring finding in the text comprehension area is that people are more apt to recall "important" elements of a text than less important elements (Newman, 1939; Gomulicki, 1956; Johnson, 1970; Brown & Smiley, 1977). Hence, researchers have devoted considerable effort towards explaining this phenomenon.

Some researchers (Meyer & McConkie, 1975; Kintsch, 1972, 1974; Meyer, 1975; Kintsch & Keenan, 1973; Kintsch, Kozminsky, Streby, McKoon & Keenan, 1975) have explained differential importance of textual elements by postulating a hierarchically structured microstructure representation of textual items. The assertion made by these researchers is that subjects form a complete representation of a text (as indicated by cued recall studies). This representation is hypothesized to be hierarchical and items near the top of the hierarchy have a higher probability of being recall than items located at lower levels. This approach has been called the microstructure levels hypothesis. However, the studies used to support the microstructure levels hypothesis have employed very short texts. Furthermore, close examination of the hierarchy construction mechanisms reveals a sensitivity to when an item is input. By embedding hierarchical subtrees within the microstructure hierarchies employed by other researchers (Kintsch & vanDijk, 1978; Miller & Kintsch, 1980), the present work was able to eliminate sensitivity of recall predictions to when an item is input to the information processing system while preserving overall coherence among textual elements. Additionally, the present work was able to account for recall of textual material in long texts.

The evaluation of the microstructure levels hypothesis in the present work showed that the level a microproposition occupies within the microstructure is only a limited predictor of recall. Previous studies which supported the idea that microstructure level was a predictor of microproposition recall employed very short texts. If these texts are viewed as having a single main idea associated with them, then the data supporting the microstructure hypothesis can be viewed as a special case of the theory of macroprocessing. Indeed, the level a microproposition occupies within a subtree is a valid predictor of a subject's ability to recall the item. However, the microstructure hypothesis breaks down when applied to long texts containing numerous macropropositions.

Some researchers have asserted that the high recall probabilities associated with important textual elements can be attributed to "deeper" or more thorough encoding of textual elements which are perceived as being important (Meyer, 1975; Fredriksen, 1972, 1975). This explanation has been adopted in the work of Kintsch and vanDijk (1978), Miller and Kintsch (1980), and the present work. A working memory buffer was used as a holdover mechanism to maintain important points for additional processing and to create coherence among textual items input at different times by the investigators. The rationale underlying this approach was that items which are processed in working memory for long periods of time have stronger LTM traces than items processed for short periods of time (Atkinson & Shiffrin, 1968). By maintaining important textual elements in the working memory buffer for additional processing cycles, the memory traces associated with these items should be stronger and easier to recall than items not held over in the working memory buffer. Thus, by using a working memory buffer in conjunction with a cyclical processing model the encoding hypothesis proposed by Fredriksen and Meyer has been refined into a plausible process model.

The Kintsch and vanDijk (1978) and Miller and Kintsch (1980) work indicated that the number of processing cycles a microproposition participated in was a reasonably good predictor of microproposition recall. In the present work, number of processing cycles was again a reasonably good recall predictor. However, this variable alone accounted for only 11% of the variance in microproposition recall.

As cycling appears to be a good predictor of recall, one is led to ask what sorts of criteria are used to determine which items are important enough to warrant additional processing. Kintsch and vanDijk and Miller and Kintsch used a leading edge strategy whereby items located at upper levels of the hierarchy of elements created in working memory are maintained for additional processing cycles. In the present work, only one of the items selected for additional processing cycles was based on the Kintsch and vanDijk leading edge strategy. All additional items were selected on the basis of macrorelevance, which is not necessarily congruent with a leading edge strategy. As the data obtained pointed to the importance of an interaction between the microprocessing and macroprocessing components of the reading process, it seemed reasonable that recent macropropositions or main ideas should be included among the items selected for holdover in the working memory buffer.

Finally, research has suggested rules for formulating main ideas or macropropositions from the facts read in the prose (vanDijk, 1977), but no attempt has been made to describe the memorial representation of main ideas or possible relationships between macrostructure representations of main ideas and microstructure representations of micropropositions. Kintsch and vanDijk (1978) assert that main ideas differ in their overall relevance to the gist of a text. However, they attempt to explain this differential importance through multiple application of macro-operators. Basically, they claim that macro-operators are applied to sequences of micropropositions which meet the criteria for application. Later, however, these macro-operators are applied to various macropropositions. The macropropositions which have had macro-operators applied to them on more than one occasion are predicted to be more memorable than those participating in only one application of macro-operators, according to Kintsch and vanDijk. However, while these investigators raise the important issue of differential importance of main ideas, they fail to provide a plausible processing explanation for the phenomenon. The present theory was designed specifically to address the issue of differential importance of macropropositions and to provide a plausible processing explanation for the phenomenon. By implementing the theory as a simulation it also became necessary to hypothesize about the relationship between microstructure and macrostructure. Kintsch and vanDijk addressed these representations separately, primarily for reasons of simplification. Thus, in addition to providing a more plausible method for creating macrostructures, the present theory of macrostructure suggests a number of modifications to the original formulation of an on-line text processing model proposed by Kintsch and vanDijk.

The theory of macrostructure adopted a text-based approach to the text comprehension problem. The hierarchical structures created by implementing the theory of macrostructure with STOP are believed to be accurate descriptors of the memorial representations a subject creates while reading. As demonstrated by the experiment, the text driven analysis simulated by STOP accounted for 56% of the variance in subject recall across texts. However, the rank order data pointed to the importance of a subject's prior knowledge in determining the relative importance of text based macropropositions. Furthermore, a subject's reading goals can interact with what is deemed macrorelevant within a text. These findings lead one to examine the knowledge-based work done in the text processing area.

The knowledge-based work on text processing (Schank & Abelson, 1977; Cullingford, 1981) has primarily addressed the ways in which prior knowledge of event sequences guided the determination of what was important about a text. However, these researchers have failed to provide process models which integrate prior knowledge expectations with the derivation of macropropositions or the construction of a representation of a text base. One notable exception is Lehnert's work on BORIS (Lehnert, Dyer, Johnson, Yang, & Harley, 1983). BORIS parses words and sentences by searching the system's memory for connections between the input text and what the system knows so far. While BORIS does not create either a microstructure or macrostructure representation of a text, it does keep track of various senarios. However, BORIS is limited to analyzing narratives.

Thus, a major implication of the data obtained is that a complete analysis of text processing should incorporate a subject's prior knowledge and reading goals into the current theory, or a similar theory of macroprocessing. It is speculated that the incorporation of knowledge and goals into STOP would influence only what was deemed macrorelevant. The relationship between the macrostructure and microstructure would remain unchanged, as would the heuristics used in the on-line formation of the long term memory microstructure and macrostructure representations.

In summary, a text driven analysis was able to account for a significant portion of what subjects recall following a reading task. By implementing the approach, as specified in the theory of macrostructure, by STOP, predictive descriptors were obtainable from most any text.

REFERENCES

Aaronson, D. & Scarborough, H.S. (1977) Performance theories for sentence coding: Some quantitative models. Journal of Verbal Learning and Verbal Behavior, 16, 277-303.

Atkinson, R.C. & Shiffrin, R.M. (1968) Human memory: A proposed system and its control processes. In K.W. Spence & J.T. Spence (Eds.), The Psychology of Learning and Motivation: Advances in Research and Theory, Vol. 2. New York: Academic Press.

Bierwisch, M. (1965) Poetik und linguistik. In H. Kreuzer & R. Gunzenhauser (Eds.), Mathematik und Dichtung. Munich: Nymphenburger. English translation (1970) in D.C. Freeman (Ed.), Linguistics and Literary Style. New York: Holt, Rinehart & Winston.

Bovair, S. & Kieras, D.E. (1981) A Guide to Propositional Analysis for Research on Technical Prose. University of Arizona Technical Report No. 8.

Bransford, J.D. & Johnson, M.K. (1973) Considerations of some problems of comprehension. In W. Chase (Ed.), Visual Information Processing. New York: Academic Press.

Brown, A.L. (1981) Metacognition and reading and writing: The development and facilitation of selective attention strategies for learning from texts. In M.L. Kamil (Ed.), Directions in Reading: Research and Instruction. Washington, D.C.: The National Reading Conference.

Brown. A.L. & Day, J.D. (1983) Macrorules for summarizing texts: The development of expertise. Journal of Verbal Leraning and Verbal Behavior, 22, 1-14.

Brown, A.L. & Smiley, S.S. (1977) Rating the importance of structural units of prose passages: A problem of metacognitive development.

Child Development, 48, 1-8.

Brown, A.L. & Smiley, S.S. (1978) The development of strategies for studying texts. Child Development, 49, 1076-1088.

Collins, A., Brown, J.S. & Larkin, K.M. (1980) Inference in Text Understanding. In R.J. Spiro, B.C. Bruce & W.F. Brewer (Eds.) Theoretical Issues in Reading Comprehension. Hillsdale, N.J.: Erlbaum.

Craik, F. & Lockhart, R. (1972) Levels of processing: A framework for memory research. Journal of Verbal Learning and Verbal Memory, 11, 671-684.

Cullingford, R.E. (1981) SAM. In R.C. Schank & C.K. Reisbeck (Eds.), Inside Computer Understanding: Five Programs Plus Miniatures. Hillsdale, N.J.: Erlbaum.

Fredericksen, C.H. (1972) Effects of task-induced cognitive operations on comprehension and memory processes. In J.B. Carroll & R.O. Freedle (Eds.), Language Comprehension and the Acquisition of Knowledge. Washingfton, D.C.: Winston.

Fredericksen, C.H. (1975) Effects of context-induced processing operations on semantic information acquired from discourse. Cognitive Psychology, 7, 139-166.

Gomulicki, B.R. (1956) Recall as an abstractive process. Acta Psychologica, 12, 77-94.

Guindon, R. & Kintsch, W. (1984) Priming macropropositions: Evidence for the primacy of macropropositions in memory for text. Journal of Verbal Learning and Verbal Behavior, 23, 508-518.

Johnson, R.E. (1970) Recall of prose as a function of the structural importance of the linguistic units. Journal of Verbal Learning and Verbal Behavior, 9, 12-20.

Kieras, D.E. (1980) Initial mention as a signal to thematic content in technical passages. Memory and Cognition, 8, 345-353.

Kieras, D.E. (1981a) The role of major referents and sentence topics in the construction of passage macrostructure. Discourse Processes, 4, 1-15.

Kieras, D.E. (1981b) Component processes in the comprehension of simple prose. Journal of Verbal Learning and Verbal Behavior, 20, 1-23.

Kieras, D.E. (1982) A model of reader strategy for abstracting main ideas from simple technical prose. Text, 2, 47-81.

Kintsch, W. (1972) Notes on the structure of semantic memory. In E. Tulving & W. Donaldson (Eds.) Organization of Memory. New York: Academic Press.

Kintsch, W. (1974) The Representation of Meaning in Memory, Hillsdale, N.J.: Erlbaum.

Kintsch, W. (1977) On comprehending stories. In M.A. Just & P.A. Carpenter (Eds.), Cognitive Processes in Comprehension. Hillsdale, N.J.: Erlbaum.

Kintsch, W. & Keenan, J. (1973) Reading rate and retention as a function of the number of propositions in the base structure of sentences. Cognitive Psychology, 5, 257-274.

Kintsch, W., Kozminsky, E., Streby, W.J., McKoon, G., & Keenan, J. (1975) Comprehension and recall of text as a function of content variables. Journal of Verbal Learning and Verbal Behavior, 14, 196-214.

Kintsch, W. & vanDijk, T.A. (1975) Comment on se rappelle et on resume des histtoires. Languages, 40, 98-116.

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

Lehnert, W.G., Dyer, M.G., Johnson, P.N., Yang, C.J., & Harley, S. (1983)

BORIS -- An in-depth understander of narritives. *Artificial Intelligence*, *20*, 15-62.

Mandler, J.M. & Johnson, N.S. (1977) Remembrance of things parsed: Story structure and recall. *Cognitive Psychology*, *9*, 111-151.

Masson, M.E.J. (1983) Conceptual processing of text during skimming and rapid sequential reading. *Memory and Cognition*, *11*, 262-274.

McKoon, G. (1977) Organization of information in memory. *Journal of Verbal Learning and Verbal Behavior*, *16*, 247-260.

McKoon, G. & Ratcliff, R. (1980) Priming in item recognition: The organization of propositions in memory for text. *Journal of Verbal Learning and Verbal Behavior*, *19*, 369-386.

Meyer, B.J.F. (1975) *The Organization of Prose and its Effect on Memory*. Amsterdam: North Holland.

Meyer, B.F.J. (1977) The structure of prose: Effects on learning and memory and implications for educational practice. In R.C. Angerson, R. Shapiro, & W. Montague (Eds.), *Schooling and the Acquisition of Knowledge*. Hillsdale, N.J.: Erlbaum.

Meyer, B.J.F. & McConkie, G.W. (1973) What is recalled after hearing a passage. *Journal of Educational Psychology*, *65*, 109117.

Miller, J.R. & Kintsch, W. (1980) Readability and recall of short prose passages: A theoretical analysis. *Jounral of Experimental Psychology: Human Learning and Memory*, *6*, 335-354.

Newman, E.B. (1939) Forgetting of meaningful material during sleep and waking. *Americal Journal of Psychology*, *52*, 65-71.

Potter, M.C., Kroll, J.F. & Harris, C. (1980) Comprehension and memory in rapid sequential reading. In R.S. Nickerson (Ed.), *Attention and Performance VIII*. Hillsdale, N.J.: Erlbaum.

Rumelhart, D.E. (1975) Notes on schema for stories. In D.G. Brown & A.

Collins (Eds.) Representation and Understanding: Studies in Cognitive Science, New York: Academic Press.

Schank. R.C. & Abelson, R.P. (1977) Scripts, Plans, Goals, and Understanding. Hillsdale, N.J.: Erlbaum.

Singer, M. (1982) Comparing memory for natural and laboratory reading. Journal of Experimental Psychology: General, 111, 331-347.

vanDijk, T.A. (1972) Some Aspects of Text Grammars. The Hague: Mouton.

vanDijk, T.A. (1975) Recalling and Summarizing Complex Discourse. Mimeographed, University of Amsterdam.

vanDijk, T.A. (1977) Semantic macrostructures and knowledge frames in discourse comprehension. In M.A. Just & P.A. Carpenter (Eds.) Cognitive Processes in Comprehension. Hillsdale, N.J.: Erlbaum.

vanDijk, T.A. & Kintsch, W. (1977) Cognitive psychology and discourse. In W.U. Dressler (Ed.), Trends in Text Linguistics. Berlin and New York: DeGruyter.

Young, S.R. (1984) RSVP: A task, reading aid and research tool. Behavior Research Methods, Instruments and Computers, 16(2), 121-124a.

Young, S.R., Angell, L.S., Bourne, L.E.,Jr. (1984) Resource allocation in reading: An interactive approach. Submitted.

APPENDIX 1

CHESAPEAKE BAY TEXT

Ten percent of the water in Chesapeake Bay contains virtually none of the dissolved oxygen normally found in clean water. This is because every year the spring runoff flushes large amounts of fertilizer residues and other pollutants into the bay.

A research team that studies oxygen depletion reports that the amount of oxygen rich water has steadily decreased since the 1930's, when oxygen depletion occurred only in the months of July and August. Because of oxygen depletion, enough damage has now been done to seriously threaten the Chesapeake Bay commercial fisheries. This lead President Reagan to call for a "long, necessary effort to clean up" the Chesapeake last January.

The affected area extends from the Susquehanna River to below the mouth of the Potomac. All of the water inbetween these areas has been highly poluted and suffers from oxygen depletion.

The lack of oxygen is the leading suspect in the reduced catch of valuable food fish. All the deep water blue crabs are gone. Additionally, Virginia oystermen have reported hauling in "black bottoms" or foul smelling sediment containing only dead shellfish.

The pollution provides nutrients that support lage populations of microscopic plants. When the plants die, they sink to the bottom of the bay and begin to decay. From spring until fall, the process of decay consumes the bay's oxygen faster than it is replenished from the atmosphere.

APPENDIX 2

MICROPROPOSITIONS FOR CHESAPEAKE BAY TEXT

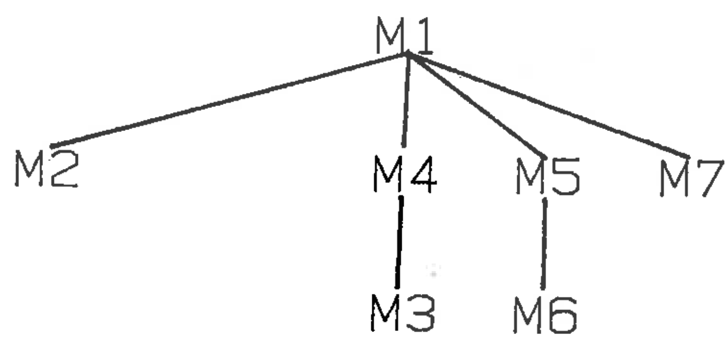
P0 (NOT P1)
P1 (FOUND WATER OXYGEN)
P2 (MOD DISSOLVED OXYGEN)
P3 (IN P2 CLEAN WATER)
P4 (WHERE P0 CHESAPEAKE BAY)
P5 (QUANTIFY 10% CHESAPEAKE BAY)
SENTENCE
P6 (FLUSH RUNOFF FERTILIZER)
P7 (FLUSH RUNOFF POLLUTION)
P7.1 (INTO RUNOFF CHESAPEAKE BAY)
P8 (MOD RUNOFF SPRING)
P9 (WHEN P6 P7 EVERY-YEAR)
P10 (BECAUSE P4 P7 P6)
SENTENCE
P11 (STUDY TEAM REPORTS)
P12 (TYPE-OF REPORTS OXYGEN DEPLETION)
P13 (ISA RESEARCH TEAM)
P14 (REPORT TEAM P17)
P15 (DECREASE OXYGEN WATER)
P16 (SINCE P15 1930)
P17 (STEADY P15)
CLAUSE
P17.1 (IN 1930 P18)
P18 (OCCUR OXYGEN DEPLETION P18.1)
P18.1 (IN JULY AUGUST)
P19 (ISA MONTH JULY)
P20 (ISA MONTH AUGUST)
P21 (ONLY P18)
SENTENCE
P21.1 (BECAUSE OXYGEN DEPLETION P22)
P22 (OCCUR DAMAGE FISHERIES)
P23 (THREATEN DAMAGE FISHERIES)
P24 (MOD P23 SERIOUSLY)
P25 (MOD CHESAPEAKE BAY FISHERIES)
P26 (MOD COMMERCIAL FISHERIES)
SENTENCE
P27 (RESULT DAMAGE FISHERIES P28)
P28 (CALL REAGAN CLEAN-UP)
P29 (MOD PRESIDENT REAGAN)
P30 (MOD LONG CLEAN-UP)
P31 (MOD NECESSARY CLEAN-UP)
P32 (MOD EFFORT CLEAN-UP)
P33 (WHEN P28 JANUARY)
P33.1 (MOD LAST JANUARY)
SENTENCE
P34 (EXTEND DAMAGE AREA SUSQUEHANA POTOMAC)
P35 (ISA RIVER SUSQUEHANA)

P36 (ISA RIVER POTOMAC)
P37 (QUALIFY MOUTH POTOMAC)
P37.1 (WHERE BELOW P37)
SENTENCE
P38 (IN POLLUTION WATER)
P39 (BETWEEN WATER SUSQUEHANA POTOMAC)
P40 (MOD HIGHLY POLLUTION)
P41 (QUALIFY ALL WATER)
P42 (SUFFER WATER OXYGEN DEPLETION)
SENTENCE
P43 (EQUAL OXYGEN DEPLETION LACK-OF OXYGEN)
P44 (SUSPECT P44.1)
P44.1 (BECAUSE LACK-OF OXYGEN P45)
P45 (REDUCE CATCH)
P46 (OF CATCH FISH)
P47 (MOD FOOD FISH)
P48 (MOD VALUABLE FOOD)
SENTENCE
P49 (ISA BLUE-CRAB FISH)
P50 (GONE BLUE-CRAB)
P51 (MOD DEEP-WATER BLUE-CRAB)
P52 (QUALIFY ALL BLUE-CRAB)
SENTENCE
P53 (ISA OYSTERMEN FISHERMAN)
P54 (REPORT OYSTERMEN P55)
P55 (HAUL OYSTERMEN BLACKBOTTOMS)
P56 (ISA BLACKBOTTOMS SEDIMENT)
P57 (MOD SMELLY SEDIMENT)
P58 (MOD FOUL SMELLY)
P59 (CONTAIN SEDIMENT FISH)
P60 (MOD SHELL FISH)
P61 (MOD DEAD FISH)
SENTENCE
P62 (PROVIDE POLLUTION NUTRIENT)
P63 (SUPPORT NUTRIENT PLANT)
P64 (MOD POPULATION PLANT)
P65 (MOD MICROSCOPIC PLANT)
P66 (QUANTIFY LARGE POPULATION)
SENTENCE
P67 (DIE PLANT)
P68 (WHEN P67 P69 P71)
P69 (SINK PLANT BOTTOM)
P70 (MOD BOTTOM BAY)
P71 (BEGIN DECAY P72)
P72 (DECAY PLANT)
SENTENCE
P73 (WHEN P75 SPRING FALL)
P74 (ISA PROCESS DECAY)
P75 (CONSUME DECAY OXYGEN)
P76 (MOD BAY OXYGEN)
P77 (FASTER P75 REPLENISH)

P78 (FROM ATMOSPHERE P79)
P79 (REPLENISH OXYGEN)

APPENDIX 3

MACROSTRUCTURE FOR CHESAPEAKE BAY TEXT



APPENDIX 4

DRIBBLE FILE FOR CHESAPEAKE BAY TEXT

```
Coherence graph generation: 10-OCT-84 18:12:12
Pruning strategy: LEADINGEDGE
Text propositions from DISK$USER1:<YOUNG.ILISP3.OK>BAY.MIC;13
Output to DISK$USER1:<YOUNG.ILISP3.OK>BAY.OUT;11
Statistics to BAY.STT
COPYRIGHT 1984 BY SHERYL R. YOUNG AND JAMES R. MILLER
Input for cycle 1:
     (P0 (NOT P1))
     (P1 (FOUND WATER OXYGEN))
     (P2 (MOD DISSOLVED OXYGEN))
     (P3 (IN P2 CLEAN WATER))
     (P4 (WHERE P0 CHESAPEAKE BAY))
     (P5 (QUANTIFY 10 CHESAPEAKE BAY))
Determine the best starting configuration.
Put M1 at level 1 in MACRO
M1 has value (INTEGRATE P0 P1 P4 P10 P7 P6) and points to P0
 in the microproposition tree.
Build a macrograph with proposition M1 at level 1.
Buffer size is now 7
Build a graph with proposition P0 at level 1.
Put P1 at level 2, pointed to by P0.
The proposition (P1 (FOUND WATER OXYGEN)) overlaps with (P0 (NOT P1)).
Put P4 at level 2, pointed to by P0.
The proposition (P4 (WHERE P0 CHESAPEAKE BAY)) overlaps with (P0 (NOT
P1)).
Put P2 at level 3, pointed to by P1.
The proposition (P2 (MOD DISSOLVED OXYGEN)) overlaps with (P1 (FOUND
WATER
OXYGEN)).
Put P3 at level 3, pointed to by P1.
The proposition (P3 (IN P2 CLEAN WATER)) overlaps with (P1 (FOUND
WATER OXYGEN))
Put P5 at level 3, pointed to by P4.
The proposition (P5 (QUANTIFY 10 CHESAPEAKE BAY)) overlaps with (P4
(WHERE P0
CHESAPEAKE BAY)).
Apply the leading edge strategy.
This is cycle 1: STM is expanded to 8 slots for this cycle only.
Retain proposition P0 at level 1.
     Proposition P1 is embedded in P0.
Retain proposition P1 at level 2.
Retain proposition P4 at level 2.
Retain proposition P5 at level 3.
Retain proposition P3 at level 3.
     Proposition P2 is embedded in P3.
Retain proposition P2 at level 3.
At the end of cycle 1, STM contains:
Level 1:
```

```
    P0 points to (P1 P4)
    Level 2:
       P1 points to (P2 P3)
       P4 points to (P5)
       Level 3:
          P2 points to NIL
          P3 points to NIL
          P5 points to NIL
The last macroproposition added to macro-memory was (M1 (INTEGRATE P0
P1 P4 P10
P7 P6)).
Input for cycle 2:
     (P6 (FLUSH RUNOFF FERTILIZER))
     (P7 (FLUSH RUNOFF POLLUTION))
     (P7.1 (INTO RUNOFF CHESAPEAKE BAY))
     (P8 (MOD RUNOFF SPRING))
     (P9 (WHEN P6 P7 EVERY-YEAR))
     (P10 (BECAUSE P4 P7 P6))
Add the cycle 2 propositions to STM.
Put P7.1 at level 3, pointed to by P4.
The proposition (P7.1 (INTO RUNOFF CHESAPEAKE BAY)) overlaps with (P4
(WHERE P0
CHESAPEAKE BAY)).
Put P10 at level 3, pointed to by P4.
The proposition (P10 (BECAUSE P4 P7 P6)) overlaps with (P4 (WHERE P0
CHESAPEAKE
BAY)).
Put P6 at level 4, pointed to by P7.1.
The proposition (P6 (FLUSH RUNOFF FERTILIZER)) overlaps with (P7.1
(INTO RUNOFF
CHESAPEAKE BAY)).
Put P7 at level 4, pointed to by P7.1.
The proposition (P7 (FLUSH RUNOFF POLLUTION)) overlaps with (P7.1
(INTO RUNOFF
CHESAPEAKE BAY)).
Put P8 at level 4, pointed to by P7.1.
The proposition (P8 (MOD RUNOFF SPRING)) overlaps with (P7.1 (INTO
RUNOFF
CHESAPEAKE BAY)).
Put P9 at level 4, pointed to by P10.
The proposition (P9 (WHEN P6 P7 EVERY-YEAR)) overlaps with (P10
(BECAUSE P4 P7
P6)).
Apply the leading edge strategy.
Retain proposition P0 at level 1.
Retain proposition P1 at level 2.
Retain proposition P4 at level 2.
Retain proposition P10 at level 3.
     Proposition P7 is embedded in P10.
The proposition pointing to P7 is not in HOLDOVER: move P7 to follow
P10.
```

Retain proposition P7 at level 4.
Proposition P6 is embedded in P10.
The proposition pointing to P6 is not in HOLDOVER: move P6 to follow P10.
Retain proposition P6 at level 4.
Retain proposition P7.1 at level 3.
At the end of cycle 2, STM contains:

```
Level 1:
   P0 points to (P1 P4)
   Level 2:
      P1 points to NIL
      P4 points to (P7.1 P10)
      Level 3:
         P7.1 points to NIL
         P10 points to (P6 P7)
         Level 4:
            P6 points to NIL
            P7 points to NIL
```

The last macroproposition added to macro-memory was (M1 (INTEGRATE P0 P1 P4 P10
P7 P6)).
Input for cycle 3:

```
     (P11 (STUDY TEAM REPORTS))
     (P12 (TYPE-OF REPORTS OXYGEN DEPLETION))
     (P13 (ISA RESEARCH TEAM))
     (P14 (REPORT TEAM P17))
     (P15 (DECREASE OXYGEN WATER))
     (P16 (SINCE P15 1930))
     (P17 (STEADY P15))
```

Add the cycle 3 propositions to STM.
Put P12 at level 3, pointed to by P1.
The proposition (P12 (TYPE-OF REPORTS OXYGEN DEPLETION)) overlaps with (P1 (
FOUND WATER OXYGEN)).
Put P15 at level 3, pointed to by P1.
The proposition (P15 (DECREASE OXYGEN WATER)) overlaps with (P1 (FOUND WATER
OXYGEN)).
Put P11 at level 4, pointed to by P12.
The proposition (P11 (STUDY TEAM REPORTS)) overlaps with (P12 (TYPE-OF REPORTS
OXYGEN DEPLETION)).
Put P16 at level 4, pointed to by P15.
The proposition (P16 (SINCE P15 1930)) overlaps with (P15 (DECREASE OXYGEN WATER
)).
Put P17 at level 4, pointed to by P15.
The proposition (P17 (STEADY P15)) overlaps with (P15 (DECREASE OXYGEN WATER)).
Put P13 at level 5, pointed to by P11.
The proposition (P13 (ISA RESEARCH TEAM)) overlaps with (P11 (STUDY

TEAM REPORTS
)).
Put P14 at level 5, pointed to by P11.
The proposition (P14 (REPORT TEAM P17)) overlaps with (P11 (STUDY TEAM REPORTS))
The macroproposition (M2 (INTEGRATE P15 P16)) is being added to macro-memory.
It overlaps with macropropositions (M1).
Put M2 at level 2 in MACRO
M2 has value (INTEGRATE P15 P16) and points to P15 in the microproposition tree.
Put M2 at level 2, pointed to by M1.
Buffer size is now 3.
Apply the leading edge strategy.
Retain proposition P15 at level 1.
Retain proposition P16 at level 2.
Retain proposition P12 at level 1.
At the end of cycle 3, STM contains:
Level 1:
   P12 points to NIL
   P15 points to (P16)
   Level 2:
      P16 points to NIL
The last macroproposition added to macro-memory was (M2 (INTEGRATE P15 P16)).
Input for cycle 4:
     (P17.1 (IN 1930 P18))
     (P18 (OCCUR OXYGEN DEPLETION P18.1))
     (P18.1 (IN JULY AUGUST))
     (P19 (ISA MONTH JULY))
     (P20 (ISA MONTH AUGUST))
     (P21 (ONLY P18))
Add the cycle 4 propositions to STM.
Put P18 at level 2, pointed to by P12.
The proposition (P18 (OCCUR OXYGEN DEPLETION P18.1)) overlaps with (P12 (TYPE-OF
 REPORTS OXYGEN DEPLETION)).
Put P17.1 at level 3, pointed to by P16.
The proposition (P17.1 (IN 1930 P18)) overlaps with (P16 (SINCE P15 1930)).
Put P18.1 at level 3, pointed to by P18.
The proposition (P18.1 (IN JULY AUGUST)) overlaps with (P18 (OCCUR OXYGEN
DEPLETION P18.1)).
Put P21 at level 3, pointed to by P18.
The proposition (P21 (ONLY P18)) overlaps with (P18 (OCCUR OXYGEN DEPLETION
P18.1)).
Put P19 at level 4, pointed to by P18.1.
The proposition (P19 (ISA MONTH JULY)) overlaps with (P18.1 (IN JULY AUGUST)).

Put P20 at level 4, pointed to by P18.1.
The proposition (P20 (ISA MONTH AUGUST)) overlaps with (P18.1 (IN JULY AUGUST)).
Apply the leading edge strategy.
Retain proposition P15 at level 1.
Retain proposition P16 at level 2.
Retain proposition P12 at level 1.
At the end of cycle 4, STM contains:
Level 1:
   P12 points to NIL
   P15 points to (P16)
   Level 2:
      P16 points to NIL
The last macroproposition added to macro-memory was (M2 (INTEGRATE P15 P16)).
Input for cycle 5:
      (P21.1 (BECAUSE OXYGEN DEPLETION P22))
      (P22 (OCCUR DAMAGE FISHERIES))
      (P23 (THREATEN DAMAGE FISHERIES))
      (P24 (MOD P23 SERIOUSLY))
      (P25 (MOD CHESAPEAKE BAY FISHERIES))
      (P26 (MOD COMMERCIAL FISHERIES))
Add the cycle 5 propositions to STM.
Put P21.1 at level 2, pointed to by P12.
The proposition (P21.1 (BECAUSE OXYGEN DEPLETION P22)) overlaps with (P12 (
TYPE-OF REPORTS OXYGEN DEPLETION)).
Put P22 at level 3, pointed to by P21.1.
The proposition (P22 (OCCUR DAMAGE FISHERIES)) overlaps with (P21.1 (BECAUSE
OXYGEN DEPLETION P22)).
Put P23 at level 4, pointed to by P22.
The proposition (P23 (THREATEN DAMAGE FISHERIES)) overlaps with (P22 (OCCUR
DAMAGE FISHERIES)).
Put P25 at level 4, pointed to by P22.
The proposition (P25 (MOD CHESAPEAKE BAY FISHERIES)) overlaps with (P22 (OCCUR
DAMAGE FISHERIES)).
Put P26 at level 4, pointed to by P22.
The proposition (P26 (MOD COMMERCIAL FISHERIES)) overlaps with (P22 (OCCUR
DAMAGE FISHERIES)).
Put P24 at level 5, pointed to by P23.
The proposition (P24 (MOD P23 SERIOUSLY)) overlaps with (P23 (THREATEN DAMAGE
FISHERIES)).
Apply the leading edge strategy.
Retain proposition P15 at level 1.
Retain proposition P16 at level 2.
Retain proposition P12 at level 1.

At the end of cycle 5, STM contains:
Level 1:
P12 points to NIL
P15 points to (P16)
Level 2:
P16 points to NIL
The last macroproposition added to macro-memory was (M2 (INTEGRATE P15 P16)).
Input for cycle 6:
(P27 (RESULT DAMAGE FISHERIES P28))
(P28 (CALL REAGAN CLEAN-UP))
(P29 (MOD PRESIDENT REAGAN))
(P30 (MOD LONG CLEAN-UP))
(P31 (MOD NECESSARY CLEAN-UP))
(P32 (MOD EFFORT CLEAN-UP))
(P33 (WHEN P28 JANUARY))
(P33.1 (MOD LAST JANUARY))
Add the cycle 6 propositions to STM.
Propositions (P27 P28 P29 P30 P31 P32 P33 P33.1)
cannot be added to STM: search LTM for a new starting proposition.
Do a LTM search for P27:
The LTM search succeeded: P22 can be reinstated via P27.
Bump P22's cycle counter.
Nothing from *INPUTSET* was placed, but the LTM search succeeded: P22 was found.
This counts as a reinstatement search.
Put P27 at level 2, pointed to by P22.
The proposition (P27 (RESULT DAMAGE FISHERIES P28)) overlaps with (P22 (OCCUR
DAMAGE FISHERIES)).
Put P28 at level 3, pointed to by P27.
The proposition (P28 (CALL REAGAN CLEAN-UP)) overlaps with (P27 (RESULT DAMAGE
FISHERIES P28)).
Put P33 at level 3, pointed to by P27.
The proposition (P33 (WHEN P28 JANUARY)) overlaps with (P27 (RESULT DAMAGE
FISHERIES P28)).
Put P29 at level 4, pointed to by P28.
The proposition (P29 (MOD PRESIDENT REAGAN)) overlaps with (P28 (CALL REAGAN
CLEAN-UP)).
Put P30 at level 4, pointed to by P28.
The proposition (P30 (MOD LONG CLEAN-UP)) overlaps with (P28 (CALL REAGAN
CLEAN-UP)).
Put P31 at level 4, pointed to by P28.
The proposition (P31 (MOD NECESSARY CLEAN-UP)) overlaps with (P28 (CALL REAGAN
CLEAN-UP)).
Put P32 at level 4, pointed to by P28.

The proposition (P32 (MOD EFFORT CLEAN-UP)) overlaps with (P28 (CALL REAGAN
CLEAN-UP)).
Put P33.1 at level 4, pointed to by P33.
The proposition (P33.1 (MOD LAST JANUARY)) overlaps with (P33 (WHEN P28 JANUARY)
).
The macroproposition (M3 (INTEGRATE P27 P28)) is being added to macro-memory.
It doesn't overlap with any existing macropropositions.
Macropropostion M3 cannot be added to macro memory: apply noconnect rule.
Put M3 at level 3 in MACRO
M3 has value (INTEGRATE P27 P28) and points to P27 in the microproposition tree.
Buffer size is now 3.
Apply the leading edge strategy.
Retain proposition P27 at level 1.
    Proposition P28 is embedded in P27.
Retain proposition P28 at level 2.
Retain proposition P33 at level 2.
At the end of cycle 6, STM contains:
Level 1:
  P27 points to (P28 P33)
  Level 2:
    P28 points to NIL
    P33 points to NIL
The last macroproposition added to macro-memory was (M3 (INTEGRATE P27 P28)).
Input for cycle 7:
    (P34 (EXTEND DAMAGE AREA SUSQUEHANA POTOMAC))
    (P35 (ISA RIVER SUSQUEHANA))
    (P36 (ISA RIVER POTOMAC))
    (P37 (QUALIFY MOUTH POTOMAC))
    (P37.1 (WHERE BELOW P37))
Add the cycle 7 propositions to STM.
Put P34 at level 2, pointed to by P27.
The proposition (P34 (EXTEND DAMAGE AREA SUSQUEHANA POTOMAC)) overlaps with (P27
(RESULT DAMAGE FISHERIES P28)).
Put P35 at level 3, pointed to by P34.
The proposition (P35 (ISA RIVER SUSQUEHANA)) overlaps with (P34 (EXTEND DAMAGE
AREA SUSQUEHANA POTOMAC)).
Put P36 at level 3, pointed to by P34.
The proposition (P36 (ISA RIVER POTOMAC)) overlaps with (P34 (EXTEND DAMAGE AREA
SUSQUEHANA POTOMAC)).
Put P37 at level 3, pointed to by P34.
The proposition (P37 (QUALIFY MOUTH POTOMAC)) overlaps with (P34 (EXTEND DAMAGE

AREA SUSQUEHANA POTOMAC)).
Put P37.1 at level 4, pointed to by P37.
The proposition (P37.1 (WHERE BELOW P37)) overlaps with (P37 (QUALIFY MOUTH
POTOMAC)).
Apply the leading edge strategy.
Retain proposition P27 at level 1.
Retain proposition P28 at level 2.
Retain proposition P34 at level 2.
At the end of cycle 7, STM contains:
Level 1:
   P27 points to (P28 P34)
   Level 2:
      P28 points to NIL
      P34 points to NIL
The last macroproposition added to macro-memory was (M3 (INTEGRATE P27 P28)).
Input for cycle 8:
     (P38 (IN POLLUTION WATER))
     (P39 (BETWEEN WATER SUSQUEHANA POTOMAC))
     (P40 (MOD HIGHLY POLLUTION))
     (P41 (QUALIFY ALL WATER))
     (P42 (SUFFER WATER OXYGEN DEPLETION))
Add the cycle 8 propositions to STM.
Put P39 at level 3, pointed to by P34.
The proposition (P39 (BETWEEN WATER SUSQUEHANA POTOMAC)) overlaps with (P34 (
EXTEND DAMAGE AREA SUSQUEHANA POTOMAC)).
Put P38 at level 4, pointed to by P39.
The proposition (P38 (IN POLLUTION WATER)) overlaps with (P39 (BETWEEN WATER
SUSQUEHANA POTOMAC)).
Put P41 at level 4, pointed to by P39.
The proposition (P41 (QUALIFY ALL WATER)) overlaps with (P39 (BETWEEN WATER
SUSQUEHANA POTOMAC)).
Put P42 at level 4, pointed to by P39.
The proposition (P42 (SUFFER WATER OXYGEN DEPLETION)) overlaps with (P39 (
BETWEEN WATER SUSQUEHANA POTOMAC)).
Put P40 at level 5, pointed to by P38.
The proposition (P40 (MOD HIGHLY POLLUTION)) overlaps with (P38 (IN POLLUTION
WATER)).
Apply the leading edge strategy.
Retain proposition P27 at level 1.
Retain proposition P28 at level 2.
Retain proposition P34 at level 2.
At the end of cycle 8, STM contains:
Level 1:
   P27 points to (P28 P34)

Level 2:
P28 points to NIL
P34 points to NIL
The last macroproposition added to macro-memory was (M3 (INTEGRATE P27 P28)).
Input for cycle 9:
(P43 (EQUAL OXYGEN DEPLETION LACK-OF OXYGEN))
(P44 (SUSPECT P44.1))
(P44.1 (BECAUSE LACK-OF OXYGEN P45))
(P45 (REDUCE CATCH))
(P46 (OF CATCH FISH))
(P47 (MOD FOOD FISH))
(P48 (MOD VALUABLE FOOD))
Add the cycle 9 propositions to STM.
Propositions (P43 P44 P44.1 P45 P46 P47 P48)
cannot be added to STM: search LTM for a new starting proposition.
Do a LTM search for P43:
The LTM search succeeded: P1 can be reinstated via P43.
Bump P1's cycle counter.
Nothing from *INPUTSET* was placed, but the LTM search succeeded: P1 was found.
This counts as a reinstatement search.
Put P43 at level 2, pointed to by P1.
The proposition (P43 (EQUAL OXYGEN DEPLETION LACK-OF OXYGEN)) overlaps with (P1
(FOUND WATER OXYGEN)).
Put P44.1 at level 2, pointed to by P1.
The proposition (P44.1 (BECAUSE LACK-OF OXYGEN P45)) overlaps with (P1 (FOUND
WATER OXYGEN)).
Put P44 at level 3, pointed to by P44.1.
The proposition (P44 (SUSPECT P44.1)) overlaps with (P44.1 (BECAUSE LACK-OF
OXYGEN P45)).
Put P45 at level 3, pointed to by P44.1.
The proposition (P45 (REDUCE CATCH)) overlaps with (P44.1 (BECAUSE LACK-OF
OXYGEN P45)).
Put P46 at level 4, pointed to by P45.
The proposition (P46 (OF CATCH FISH)) overlaps with (P45 (REDUCE CATCH)).
Put P47 at level 5, pointed to by P46.
The proposition (P47 (MOD FOOD FISH)) overlaps with (P46 (OF CATCH FISH)).
Put P48 at level 6, pointed to by P47.
The proposition (P48 (MOD VALUABLE FOOD)) overlaps with (P47 (MOD FOOD FISH)).
The macroproposition (M4 (INTEGRATE P44.1 P45 P46))
is being added to macro-memory.
It overlaps with macropropositions (M1 M2).
Put M4 at level 2 in MACRO

M4 has value (INTEGRATE P44.1 P45 P46) and points to P44.1
in the microproposition tree.
Put M4 at level 2, pointed to by M1.
Buffer size is now 4.
Apply the leading edge strategy.
Retain proposition P44.1 at level 1.
Proposition P45 is embedded in P44.1.
Retain proposition P45 at level 2.
Retain proposition P46 at level 3.
Retain proposition P43 at level 1.
At the end of cycle 9, STM contains:
Level 1:
P43 points to NIL
P44.1 points to (P45)
Level 2:
P45 points to (P46)
Level 3:
P46 points to NIL
The last macroproposition added to macro-memory was (M4 (INTEGRATE
P44.1 P45 P46
)).
Input for cycle 10:
(P49 (ISA BLUE-CRAB FISH))
(P50 (GONE BLUE-CRAB))
(P51 (MOD DEEP-WATER BLUE-CRAB))
(P52 (QUALIFY ALL BLUE-CRAB))
Add the cycle 10 propositions to STM.
Put P49 at level 4, pointed to by P46.
The proposition (P49 (ISA BLUE-CRAB FISH)) overlaps with (P46 (OF
CATCH FISH)).
Put P50 at level 5, pointed to by P49.
The proposition (P50 (GONE BLUE-CRAB)) overlaps with (P49 (ISA
BLUE-CRAB FISH)).
Put P51 at level 5, pointed to by P49.
The proposition (P51 (MOD DEEP-WATER BLUE-CRAB)) overlaps with (P49
(ISA
BLUE-CRAB FISH)).
Put P52 at level 5, pointed to by P49.
The proposition (P52 (QUALIFY ALL BLUE-CRAB)) overlaps with (P49 (ISA
BLUE-CRAB
FISH)).
Apply the leading edge strategy.
Retain proposition P44.1 at level 1.
Retain proposition P45 at level 2.
Retain proposition P46 at level 3.
Retain proposition P49 at level 4.
At the end of cycle 10, STM contains:
Level 1:
P44.1 points to (P45)
Level 2:
P45 points to (P46)

```
      Level 3:
         P46 points to (P49)
         Level 4:
            P49 points to NIL
The last macroproposition added to macro-memory was (M4 (INTEGRATE
P44.1 P45 P46
)).
Input for cycle 11:
     (P53 (ISA OYSTERMEN FISHERMAN))
     (P54 (REPORT OYSTERMEN P55))
     (P55 (HAUL OYSTERMEN BLACKBOTTOMS))
     (P56 (ISA BLACKBOTTOMS SEDIMENT))
     (P57 (MOD SMELLY SEDIMENT))
     (P58 (MOD FOUL SMELLY))
     (P59 (CONTAIN SEDIMENT FISH))
     (P60 (MOD SHELL FISH))
     (P61 (MOD DEAD FISH))
Add the cycle 11 propositions to STM.
Put P59 at level 4, pointed to by P46.
The proposition (P59 (CONTAIN SEDIMENT FISH)) overlaps with (P46 (OF
CATCH FISH)
).
Put P60 at level 4, pointed to by P46.
The proposition (P60 (MOD SHELL FISH)) overlaps with (P46 (OF CATCH
FISH)).
Put P61 at level 4, pointed to by P46.
The proposition (P61 (MOD DEAD FISH)) overlaps with (P46 (OF CATCH
FISH)).
Put P56 at level 5, pointed to by P59.
The proposition (P56 (ISA BLACKBOTTOMS SEDIMENT)) overlaps with (P59
(CONTAIN
SEDIMENT FISH)).
Put P57 at level 5, pointed to by P59.
The proposition (P57 (MOD SMELLY SEDIMENT)) overlaps with (P59
(CONTAIN SEDIMENT
 FISH)).
Put P55 at level 6, pointed to by P56.
The proposition (P55 (HAUL OYSTERMEN BLACKBOTTOMS)) overlaps with (P56
(ISA
BLACKBOTTOMS SEDIMENT)).
Put P58 at level 6, pointed to by P57.
The proposition (P58 (MOD FOUL SMELLY)) overlaps with (P57 (MOD SMELLY
SEDIMENT)
).
Put P53 at level 7, pointed to by P55.
The proposition (P53 (ISA OYSTERMEN FISHERMAN)) overlaps with (P55
(HAUL
OYSTERMEN BLACKBOTTOMS)).
Put P54 at level 7, pointed to by P55.
The proposition (P54 (REPORT OYSTERMEN P55)) overlaps with (P55 (HAUL
OYSTERMEN
```

BLACKBOTTOMS)).
Apply the leading edge strategy.
Retain proposition P44.1 at level 1.
Retain proposition P45 at level 2.
Retain proposition P46 at level 3.
Retain proposition P61 at level 4.
At the end of cycle 11, STM contains:

```
Level 1:
   P44.1 points to (P45)
   Level 2:
      P45 points to (P46)
      Level 3:
         P46 points to (P61)
         Level 4:
            P61 points to NIL
```

The last macroproposition added to macro-memory was (M4 (INTEGRATE
P44.1 P45 P46
)).
Input for cycle 12:

```
     (P62 (PROVIDE POLLUTION NUTRIENT))
     (P63 (SUPPORT NUTRIENT PLANT))
     (P64 (MOD POPULATION PLANT))
     (P65 (MOD MICROSCOPIC PLANT))
     (P66 (QUANTIFY LARGE POPULATION))
```

Add the cycle 12 propositions to STM.
Propositions (P62 P63 P64 P65 P66) cannot be added to STM: search LTM for a new
starting proposition.
Do a LTM search for P62:
The LTM search succeeded: P7 can be reinstated via P62.
Bump P7's cycle counter.
Nothing from *INPUTSET* was placed, but the LTM search succeeded: P7 was found.
This counts as a reinstatement search.
Put P62 at level 2, pointed to by P7.
The proposition (P62 (PROVIDE POLLUTION NUTRIENT)) overlaps with (P7 (FLUSH
RUNOFF POLLUTION)).
Put P63 at level 3, pointed to by P62.
The proposition (P63 (SUPPORT NUTRIENT PLANT)) overlaps with (P62 (PROVIDE
POLLUTION NUTRIENT)).
Put P64 at level 4, pointed to by P63.
The proposition (P64 (MOD POPULATION PLANT)) overlaps with (P63 (SUPPORT
NUTRIENT PLANT)).
Put P65 at level 4, pointed to by P63.
The proposition (P65 (MOD MICROSCOPIC PLANT)) overlaps with (P63 (SUPPORT
NUTRIENT PLANT)).
Put P66 at level 5, pointed to by P64.

The proposition (P66 (QUANTIFY LARGE POPULATION)) overlaps with (P64
(MOD
POPULATION PLANT)).
The macroproposition (M5 (INTEGRATE P62 P63)) is being added to
macro-memory.
It overlaps with macropropositions (M1).
Put M5 at level 2 in MACRO
M5 has value (INTEGRATE P62 P63) and points to P62 in the
microproposition tree.
Put M5 at level 2, pointed to by M1.
Buffer size is now 3.
Apply the leading edge strategy.
Retain proposition P62 at level 1.
Retain proposition P63 at level 2.
Retain proposition P65 at level 3.
At the end of cycle 12, STM contains:

```
Level 1:
   P62 points to (P63)
   Level 2:
      P63 points to (P65)
      Level 3:
         P65 points to NIL
```

The last macroproposition added to macro-memory was (M5 (INTEGRATE P62
P63)).
Input for cycle 13:

```
     (P67 (DIE PLANT))
     (P68 (WHEN P67 P69 P71))
     (P69 (SINK PLANT BOTTOM))
     (P70 (MOD BOTTOM BAY))
     (P71 (BEGIN DECAY P72))
     (P72 (DECAY PLANT))
```

Add the cycle 13 propositions to STM.
Put P67 at level 3, pointed to by P63.
The proposition (P67 (DIE PLANT)) overlaps with (P63 (SUPPORT NUTRIENT
PLANT)).
Put P69 at level 3, pointed to by P63.
The proposition (P69 (SINK PLANT BOTTOM)) overlaps with (P63 (SUPPORT
NUTRIENT
PLANT)).
Put P72 at level 3, pointed to by P63.
The proposition (P72 (DECAY PLANT)) overlaps with (P63 (SUPPORT
NUTRIENT PLANT))
Put P68 at level 4, pointed to by P67.
The proposition (P68 (WHEN P67 P69 P71)) overlaps with (P67 (DIE
PLANT)).
Put P70 at level 4, pointed to by P69.
The proposition (P70 (MOD BOTTOM BAY)) overlaps with (P69 (SINK PLANT
BOTTOM)).
Put P71 at level 4, pointed to by P72.
The proposition (P71 (BEGIN DECAY P72)) overlaps with (P72 (DECAY
PLANT)).

```
The macroproposition (M6 (GENERALIZE P68 P69 P67 P71))
 is being added to macro-memory.
It doesn't overlap with any existing macropropositions.
Macropropostion M6 cannot be added to macro memory: apply noconnect
rule.
Put M6 at level 4 in MACRO
M6 has value (GENERALIZE P68 P69 P67 P71) and points to P68
 in the microproposition tree.
Buffer size is now 5.
Apply the leading edge strategy.
Retain proposition P68 at level 1.
     Proposition P67 is embedded in P68.
Retain proposition P67 at level 2.
     Proposition P69 is embedded in P68.
The proposition pointing to P69 is not in HOLDOVER: move P69 to follow
P68.
Retain proposition P69 at level 2.
     Proposition P71 is embedded in P68.
The proposition pointing to P71 is not in HOLDOVER: move P71 to follow
P68.
Retain proposition P71 at level 2.
     Proposition P72 is embedded in P71.
The proposition pointing to P72 is not in HOLDOVER: move P72 to follow
P71.
Retain proposition P72 at level 3.
At the end of cycle 13, STM contains:
Level 1:
   P68 points to (P69 P71)
   Level 2:
      P67 points to (P68)
      P69 points to NIL
      P71 points to (P72)
      Level 3:
         P72 points to NIL
The last macroproposition added to macro-memory was (M6 (GENERALIZE
P68 P69 P67
P71)).
Input for cycle 14:
     (P73 (WHEN P75 SPRING FALL))
     (P74 (ISA PROCESS DECAY))
     (P75 (CONSUME DECAY OXYGEN))
     (P76 (MOD BAY OXYGEN))
     (P77 (FASTER P75 REPLENISH))
     (P78 (FROM ATMOSPHERE P79))
     (P79 (REPLENISH OXYGEN))
Add the cycle 14 propositions to STM.
Put P74 at level 3, pointed to by P71.
The proposition (P74 (ISA PROCESS DECAY)) overlaps with (P71 (BEGIN
DECAY P72)).
Put P75 at level 3, pointed to by P71.
The proposition (P75 (CONSUME DECAY OXYGEN)) overlaps with (P71 (BEGIN
```

```
DECAY P72
)).
Put P73 at level 4, pointed to by P75.
The proposition (P73 (WHEN P75 SPRING FALL)) overlaps with (P75
(CONSUME DECAY
OXYGEN)).
Put P76 at level 4, pointed to by P75.
The proposition (P76 (MOD BAY OXYGEN)) overlaps with (P75 (CONSUME
DECAY OXYGEN)
).
Put P77 at level 4, pointed to by P75.
The proposition (P77 (FASTER P75 REPLENISH)) overlaps with (P75
(CONSUME DECAY
OXYGEN)).
Put P79 at level 4, pointed to by P75.
The proposition (P79 (REPLENISH OXYGEN)) overlaps with (P75 (CONSUME
DECAY
OXYGEN)).
Put P78 at level 5, pointed to by P79.
The proposition (P78 (FROM ATMOSPHERE P79)) overlaps with (P79
(REPLENISH OXYGEN
)).
The macroproposition (M7 (INTEGRATE P75 P74 P77))
 is being added to macro-memory.
It overlaps with macropropositions (M1 M2 M4 M6).
Put M7 at level 2 in MACRO
M7 has value (INTEGRATE P75 P74 P77) and points to P75
 in the microproposition tree.
Put M7 at level 2, pointed to by M1.
Buffer size is now 4.
Apply the leading edge strategy.
Retain proposition P75 at level 1.
Retain proposition P74 at level 2.
Retain proposition P77 at level 2.
Retain proposition P79 at level 2.
At the end of cycle 14, STM contains:
Level 1:
   P75 points to (P77 P79)
   Level 2:
      P74 points to NIL
      P77 points to NIL
      P79 points to NIL
The last macroproposition added to macro-memory was (M7 (INTEGRATE P75
P74 P77))
End of file found.
The final LTM graph is completely connected.
32 propositions were held over:
P0: 2 cycles.
P1: 3 cycles.
P2: 1 cycles.
P3: 1 cycles.
```

```
P4: 2 cycles.
P5: 1 cycles.
P6: 1 cycles.
P7: 2 cycles.
P7.1: 1 cycles.
P10: 1 cycles.
P12: 3 cycles.
P15: 3 cycles.
P16: 3 cycles.
P22: 1 cycles.
P27: 3 cycles.
P28: 3 cycles.
P33: 1 cycles.
P34: 2 cycles.
P43: 1 cycles.
P44.1: 3 cycles.
P45: 3 cycles.
P46: 3 cycles.
P49: 1 cycles.
P61: 1 cycles.
P62: 1 cycles.
P63: 1 cycles.
P65: 1 cycles.
P67: 1 cycles.
P68: 1 cycles.
P69: 1 cycles.
P71: 1 cycles.
P72: 1 cycles.
At the end of cycle 14, LTM contains:
Level 1:
   P0 points to (P1 P4)
   Level 2:
      P1 points to (P2 P3 P12 P15 P43 P44.1)
      P4 points to (P5 P7.1 P10)
      Level 3:
         P2 points to NIL
         P3 points to NIL
         P5 points to NIL
         P7.1 points to (P6 P7 P8)
         P10 points to (P9)
         P12 points to (P11 P18 P21.1)
         P15 points to (P16 P17)
         P43 points to NIL
         P44.1 points to (P44 P45)
         Level 4:
            P6 points to NIL
            P7 points to (P62)
            P8 points to NIL
            P9 points to NIL
            P11 points to (P13 P14)
            P16 points to (P17.1)
```

```
P17 points to NIL
P18 points to (P18.1 P21)
P21.1 points to (P22)
P44 points to NIL
P45 points to (P46)
Level 5:
   P13 points to NIL
   P14 points to NIL
   P17.1 points to NIL
   P18.1 points to (P19 P20)
   P21 points to NIL
   P22 points to (P23 P25 P26 P27)
   P46 points to (P47 P49 P59 P60 P61)
   P62 points to (P63)
   Level 6:
      P19 points to NIL
      P20 points to NIL
      P23 points to (P24)
      P25 points to NIL
      P26 points to NIL
      P27 points to (P28 P33 P34)
      P47 points to (P48)
      P49 points to (P50 P51 P52)
      P59 points to (P56 P57)
      P60 points to NIL
      P61 points to NIL
      P63 points to (P64 P65 P67 P69 P72)
      Level 7:
         P24 points to NIL
         P28 points to (P29 P30 P31 P32)
         P33 points to (P33.1)
         P34 points to (P35 P36 P37 P39)
         P48 points to NIL
         P50 points to NIL
         P51 points to NIL
         P52 points to NIL
         P56 points to (P55)
         P57 points to (P58)
         P64 points to (P66)
         P65 points to NIL
         P67 points to (P68)
         P69 points to (P70)
         P72 points to (P71)
         Level 8:
            P29 points to NIL
            P30 points to NIL
            P31 points to NIL
            P32 points to NIL
            P33.1 points to NIL
            P35 points to NIL
            P36 points to NIL
```

```
                    P37 points to (P37.1)
                    P39 points to (P38 P41 P42)
                    P55 points to (P53 P54)
                    P58 points to NIL
                    P66 points to NIL
                    P68 points to NIL
                    P70 points to NIL
                    P71 points to (P74 P75)
                    Level 9:
                       P37.1 points to NIL
                       P38 points to (P40)
                       P41 points to NIL
                       P42 points to NIL
                       P53 points to NIL
                       P54 points to NIL
                       P74 points to NIL
                       P75 points to (P73 P76 P77 P79)
                       Level 10:
                          P40 points to NIL
                          P73 points to NIL
                          P76 points to NIL
                          P77 points to NIL
                          P79 points to (P78)
                          Level 11:
                             P78 points to NIL
At the end of cycle 14, MACRO contains:
Level 1:
   M1 points to P0 in the microproposition tree.
   M1 points to (M2 M4 M5 M7)
   Level 2:
      M2 points to P15 in the microproposition tree.
      M4 points to P44.1 in the microproposition tree.
      M5 points to P62 in the microproposition tree.
      M7 points to P75 in the microproposition tree.
      M2 points to NIL
      M4 points to NIL
      M5 points to NIL
      M7 points to NIL
      Level 3:
         M3 points to P27 in the microproposition tree.
         M3 points to NIL
         Level 4:
            M6 points to P68 in the microproposition tree.
            M6 points to NIL
This run required 14 input cycles, 0 reorganization attempts,
     3 reinstatements using 3 LTM searches, and 0 inferences
     using 0 LTM searches.
STM was stretched 0 times and overloaded 0 times.
```

APPENDIX 5

EINSTEIN TEXT

Beginning in 1950, FBI director J. Edgar Hoover placed Albert Einstein, the great physicist, under surveillance. Einstein arrived in the United States in 1932. Following his arrival, he expanded his writing to causes far beyond the subject of physics. His work included essays and pulic statements supporting civil liberties, disarmament, and racial equality.

Richard Alan Schwartz, an English professor at Florida International University in Miami discovered Einstein's FBI files. Using the Freedom of Information act, Schwartz was not only able to view the files, but was able to publish descriptions of the charges contained in them. He described the charges against Einstein as "a litany of horrors that make wonderful absurdist drama. Although the FBI was "never able to get any corroborating information", it ordered investigations of Einstein at the slightest provocation.

There were plenty of provacateurs. The FBI files contained 1,500 pages of accusations by people claiming to know that Einstein was involved in treasonous plots as diverse as a "push to take over Hollywood" and the development of a thought-controlling robot. One letter mentioned a death ray the scientist supposedly invented. Another acuser asserted that Einstein had participated in the 1932 Lindbergh kidnapping. Another charged that the physicist had a Berlin office in the 1920's and 30's which had been a front for Soviet spying.

The communist accusation was the most common charge against Einstein. One woman claimed to have evidence that Einstein was a communist which was based on the scientist's refusal to stand up while the German National Anthem was being played in 1930. But, not all of the mail was from crackpots. Radio commentator Walter Winchell forwarded a letter from a listener containing a long list of "commie fronts" to which Einstein was alledged to have belonged.

What to make of this? Schwartz claims that Einstein's accomplishments were so great that he inspired not only awe, but fear in others. He believes that the FBI wanted to believe that Einstein was guilty of something for this reason. Yet, all Einstein had done was to exercise his rights of free speech and political expression.

APPENDIX 6

MICROPROPOSITIONS FOR EINSTEIN TEXT

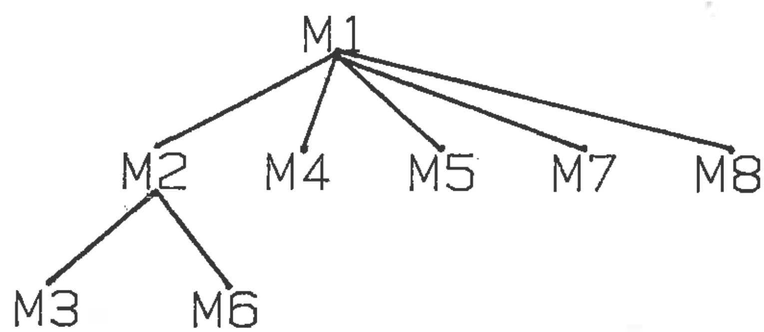
P1 (TIME P2 1950)
P2 (PLACED J-EDGAR-HOOVER P3)
P3 (SURVEILLANCE EINSTEIN)
P4 (ISA J-EDGAR-HOOVER FBI)
P5 (MOD FBI DIRECTOR)
P6 (ISA PHYSICIST EINSTEIN)
P7 (MOD PHYSICIST GREAT)
SENTENCE
P8 (ARRIVED EINSTEIN US)
P9 (TIME P8 1932)
P10 (AFTER P8 P11)
P11 (EXPANDED EINSTEIN WRITING P12)
P12 (INTO CAUSES)
P13 (TYPES-OF CAUSES P13.1)
P13.1 (BEYOND PHYSICS)
P14 (ISA SUBJECT PHYSICS)
SENTENCE
P15 (INCLUDED WRITING ESSAYS)
P16 (INCLUDED WRITING PUBLIC-STATEMENTS)
P17 (SUPPORT PUBLIC-STATEMENTS CIVIL-LIBERTIES)
P18 (SUPPORT PUBLIC-STATEMENTS DISARMAMENT)
P19 (SUPPORT PUBLIC-STATEMENTS RACIAL-EQUALITY)
SENTENCE
P20 (DISCOVERED SCHWARTZ FILES)
P21 (TYPE-OF FILES FBI)
P22 (MOD EINSTEIN FBI FILES)
P23 (MOD SCHWARTZ RICHARD-ALLEN)
P24 (ISA PROFESSOR SCHWARTZ)
P25 (TYPE-OF PROFESSOR ENGLISH)
P26 (WHERE PROFESSOR UNIVERSITY)
P27 (MOD UNIVERSITY FLORIDA-INTERNATIONAL)
P28 (WHERE UNIVERSITY MIAMI)
SENTENCE
P29 (USE SCHWARTZ FREEDOM-OF-INFO-ACT P30 P31)
P30 (VIEW SCHWARTZ FBI FILES)
P31 (PUBLISH CHARGES SCHWARTZ)
P32 (DESCRIBE SCHWARTZ CHARGES)
P33 (WHERE CHARGES FBI FILES)
SENTENCE
P34 (DESCRIBE SCHWARTZ P35 P36)
P35 (AGAINST CHARGES EINSTEIN)
P36 (ISA CHARGES HORRORS)
P37 (MOD HORRORS DRAMA)
P38 (MOD LITANY HORRORS)
P39 (MOD ABSURDIST DRAMA)
P40 (MOD WONDERFUL P39)

SENTENCE
P41 (ALTHOUGH P42)
P42 (NOT P42.1)
P42.1 (GET FBI INFORMATION)
P43 (TYPE-OF INFORMATION CORROBORATING)
P44 (ORDER FBI INVESTIGATION)
P45 (OF INVESTIGATION EINSTEIN)
P46 (WITH PROVOCATION INVESTIGATION)
P47 (MOD PROVOCATION SLIGHTEST)
SENTENCE
P48 (EXIST MANY PROVOCATION)
P49 (FROM PROVOCATION PEOPLE)
SENTENCE
P50 (CONTAINED FBI FILES ACCUSATION)
P51 (NUMBER-OF ACCUSATION 1500 PAGE)
P52 (FROM ACCUSATION PEOPLE)
P53 (KNOW PEOPLE P54)
P54 (INVOLVED EINSTEIN PLOTS)
P55 (TYPE-OF PLOTS TREASONOUS)
P56 (MOD PLOTS DIVERSE)
P57 (TYPE-OF PLOTS P58 P59)
P58 (TAKEOVER HOLLYWOOD)
P59 (DEVELOP ROBOT)
P60 (MOD ROBOT THOUGHT-CONTROLLING)
SENTENCE
P61 (WERE PAGE LETTER)
P62 (CONTAIN LETTER P63)
P63 (INVENTED EINSTEIN DEATH-RAY)
P64 (SUPPOSEDLY P63)
SENTENCE
P65 (ANOTHER ACCUSATION P66)
P66 (PARTICIPATED EINSTEIN KIDNAPPING)
P67 (MOD KIDNAPPING LINDBERGH)
P68 (WHEN P67 1932)
SENTENCE
P69 (ANOTHER CHARGE P70 P71)
P70 (HAD EINSTEIN OFFICE)
P71 (ISA FRONT OFFICE SOVIET)
P72 (WHERE OFFICE BERLIN)
P73 (HAD P70 1920 1930)
P74 (MOD SPYING SOVIET)
SENTENCE
P75 (WAS ACCUSATION)
P76 (MOD ACCUSATION SOVIET P77)
P77 (AGAINST CHARGE EINSTEIN)
P78 (MOD ACCUSATION COMMON)
P79 (MOD COMMON MOST)
SENTENCE
P80 (CLAIM P81)
P81 (POSSESS WOMAN EVIDENCE P83)
P82 (MOD WOMAN ONE)

P83 (ISA EINSTEIN SOVIET)
P84 (BASED-ON EVIDENCE P85)
P85 (REFUSE EINSTEIN STAND)
P86 (WHEN P85 P87)
P87 (PLAY ANTHEM)
P88 (MOD ANTHEM GERMAN-NATIONAL)
P89 (TIME-OF P85 1930)
SENTENCE
P90 (FROM ACCUSATION CRACKPOTS)
P91 (MOD ALL ACCUSATION)
P92 (NOT P91)
SENTENCE
P93 (FORWARD WALTER-WINCHELL LETTER)
P94 (ISA WALTER-WINCHELL RADIO-COMMENTATOR)
P95 (FROM LETTER LISTENER)
P96 (CONTAIN LETTER LIST)
P97 (MOD LONG LIST)
P98 (OF LIST FRONT)
P99 (MOD FRONT SOVIET)
P100 (BELONG EINSTEIN P99)
P101 (ALLEDGE P100)
SENTENCE
P102 (CLAIM SCHWARTZ P103)
P103 (BECAUSE P104 P106 P107)
P104 (HAVE ACCOMPLISHMENTS EINSTEIN)
P105 (MOD ACCOMPLISHMENTS GREAT)
P106 (INSPIRE EINSTEIN AWE)
P107 (INSPIRE EINSTEIN FEAR)
P108 (WHO P105 OTHERS)
SENTENCE
P109 (BELIEVE SCHWARTZ P110)
P110 (WANTED-TO-BELIEVE FBI P111)
P111 (GUILTY EINSTEIN SOMETHING)
P112 (WHY P110 P106 P107)
SENTENCE
P113 (EXERCISE EINSTEIN RIGHT)
P114 (MOD RIGHT FREE-SPEECH)
P115 (MOD RIGHT POLITICAL EXPRESSION)
P116 (ONLY P113)

APPENDIX 7

LTM MACROSTRUCTURE FOR EINSTEIN TEXT



APPENDIX 8

COMPUTER CRIME TEXT

Seven young computer hackers in Milwaukee got some unwelcome publicity by managing to tap into the computers at Los Alamos National Laboratories in New Mexico, where many of the nation's nuclear bombs are designed. Although the group of kids did not uncover any top secret weapons plans, thye caused dismay in government agencies, corporations, hospitals and other institutions across the country by spotlighting the vulnerability of all computerized information.

The members of the group, who ranged in age from 15 to 22, proved the contention of artificial intelligence expert Marvin Minsky that "computers are dumb machines, and people can still easily outwit them." The group of kids were not computer wiz kids or mad geniuses bent on cracking almost unbreakable electronic codes. Rather, they were just ordinary home computer hobbyists who used a few standard programming techniques to dial up private files. Among the other computerized data they tinkered with were the records of radiation therapy patients at manhattan's Sloan-Kettering Cancer Institute.

The group began their electronic raids a while back, but stepped up their raids after seeing the movie "War Games", where a clever young kid taps into the nation's defense system computer and begins playing a game of global thermonuclear war. Like the kid in the movie, all the group needed was a personal computer and a modem, a device which links the computer to a telephone. With this equiptment they could call up all the computers connected by a giant network known as telenet. Although each institution subscribing to telenet supposedly has its own security system, sneaking around that security proved to be only too simple.

At both Los Alamos and Sloan Kettering, the tampering was noticed quickly, and the FBI notified. Decoy files were then created in the data banks to try to trap the unauthorized users. Anonymous tips finally led the FBI to the group of kids from Milwaukee. As of recently, no charges have been brought against the group, but in the world of computers, the panic was on.

Donn Parker, a computer crime expert at SRI International in Menlo Park, California views the incident as part of a larger fad. There is an epidemic of malicious computer break-ins. Everybody's privacy is in jeopardy. Just about the only people not upset by the rumpus were the cryptographers who design high security computer codes. It seemed likely that their skills would be in great demand for years to come.

APPENDIX 9

MICROPROPOSITIONS FOR COMPUTER CRIME TEXT

P1 (RECEIVED HACKERS PUBLICITY)
P2 (NUMBER-OF HACKERS SEVEN)
P3 (MOD HACKERS YOUNG)
P4 (MOD HACKERS MILWAUKEE)
P5 (MOD PUBLICITY SOME)
P6 (TYPE-OF PUBLICITY UNWELCOME)
P7 (MANAGED HACKERS P8)
P8 (TAP HACKERS COMPUTER)
P9 (MOD COMPUTER LABORATORY)
P10 (MOD LABORATORY NATIONAL)
P11 (WHERE LABORATORY LOS-ALAMOS)
P12 (MOD LOS-ALAMOS NEW-MEXICO)
CLAUSE
P13 (IN LABORATORY P13.1)
P13.1 (DESIGN BOMB)
P14 (MOD BOMB NUCLEAR)
P15 (FOR BOMB NATION)
SENTENCE
P16 (ISA HACKERS GROUP)
P17 (ISA HACKERS KIDS)
P18 (NOT P18.1)
P18.1 (UNCOVER KIDS WEAPONS)
P19 (TYPE-OF WEAPONS SECRET)
P20 (MOD SECRET TOP)
CLAUSE
P21 (CAUSE HACKERS DISMAY)
P22 (TO DISMAY GOVERNMENT)
P23 (MOD GOVERNMENT AGENCIES)
P24 (TO DISMAY CORPORATIONS)
P25 (TO DISMAY HOSPITAL)
P26 (TO DISMAY OTHER INSTITUTION)
P27 (ACROSS INSTITUTION COUNTRY)
P28 (AT DISMAY VULNERABILITY)
P29 (MOD SPOTLIGHT VULNERABILITY)
P30 (OF VULNERABILITY INFORMATION)
P31 (TYPE-OF INFORMATION COMPUTER)
SENTENCE
P32 (WERE MEMBERS GROUP HACKERS)
P33 (AGE MEMBERS 15-22)
P34 (PROVED GROUP CONTENTION)
P35 (OF CONTENTION AI EXPERT)
P36 (ISA EXPERT MINSKY)
P37 (MOD MINSKY MARVIN)
P38 (ISA CONTENTION COMPUTER DUMB)
P39 (MOD DUMB MACHINE)
P40 (DUE-TO DUMB P41)
P41 (OUTWIT PEOPLE COMPUTER)

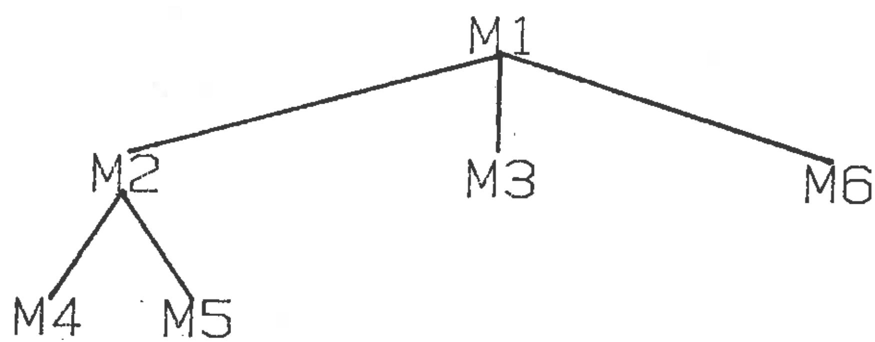
```
P42 (MOD P41 EASILY)
*SENTENCE*
P43 (ISA GROUP KIDS HACKERS)
 P44 (NOT P44.1)
P44.1 (ISA GROUP WIZ-KIDS)
P45 (TYPE-OF WIZ-KIDS COMPUTER)
P46 (NOT P46.1)
P46.1 (ISA GROUP GENIUSES)
P47 (TYPE-OF GENIUSES MAD)
P48 (BENT-ON GENIUSES P49)
P49 (CRACKING CODES)
P50 (TYPE-OF CODES COMPUTER)
P51 (MOD CODES UNBREAKABLE)
P52 (MOD P51 ALMOST)
*SENTENCE*
P53 (ISA GROUP HACKERS HOBBYISTS)
P54 (TYPE-OF HOBBYISTS COMPUTER)
P55 (MOD COMPUTER HOME)
P56 (MOD HOBBYISTS ORDINARY)
P57 (USED HOBBYISTS TECHNIQUES)
P58 (TYPE-OF TECHNIQUES PROGRAMMING)
P59 (MOD TECHNIQUES STANDARD)
P60 (MOD TECHNIQUES FEW)
P61 (FOR TECHNIQUES P62)
P62 (DIAL-UP FILES)
P63 (MOD FILES PRIVATE)
*SENTENCE*
P64 (TINKERED GROUP DATA)
P65 (TYPE-OF DATA COMPUTER)
P66 (MOD DATA MANY)
P67 (ISA DATA RECORDS)
P68 (TYPE-OF RECORDS PATIENT)
P69 (MOD PATIENT RADIATION)
P70 (MOD RADIATION THERAPY)
P71 (AT RECORDS INSTITUTE)
P71.1 (MOD INSTITUTE CANCER)
P72 (MOD INSTITUTE SLOAN-KETTERING)
P73 (WHERE SLOAN-KETTERING MANHATTAN)
*SENTENCE*
P74 (BEGAN GROUP HACKERS TAP RAIDS)
P75 (MOD ELECTRONIC TAP RAIDS)
P76 (WHEN P74 A-WHILE-BACK)
P77 (STEPPED-UP GROUP TAP RAIDS)
P78 (AFTER P79 P77)
P79 (SEE GROUP MOVIE)
P80 (MOD MOVIE WAR-GAMES)
*CLAUSE*
P81 (IN MOVIE KID P84)
P82 (MOD KID CLEVER)
P83 (MOD KID YOUNG)
P84 (TAP COMPUTER)
```

P85 (MOD COMPUTER DEFENSE)
P86 (MOD DEFENSE SYSTEM)
P87 (MOD DEFENSE NATION)
P88 (BEGINS MOVIE KID GAME)
P89 (OF GAME WAR)
P90 (TYPE-OF WAR THERMONUCLEAR)
P91 (TYPE-OF WAR GLOBAL)
SENTENCE
P92 (NEEDED GROUP MODEM COMPUTER)
P93 (MOD COMPUTER PERSONAL)
P94 (AS MOVIE KID P92)
P95 (ISA MODEM DEVICE)
P96 (LINK DEVICE COMPUTER)
P97 (LINK DEVICE TELEPHONE)
SENTENCE
P98 (WITH COMPUTER MODEM P99)
P99 (CALL GROUP COMPUTER)
P100 (MOD COMPUTER ALL)
P101 (CONNECT COMPUTER NETWORK)
P102 (MOD NETWORK TELENET)
P103 (MOD NETWORK GIANT)
SENTENCE
P104 (SUBSCRIBE INSTITUTION NETWORK)
P105 (MOD INSTITUTION EACH)
P106 (HAVE P105 SECURITY)
P107 (MOD SECURITY SYSTEM)
P108 (SNEAK SECURITY SIMPLE)
P109 (ONLY P108 SIMPLE)
P110 (MOD SIMPLE TOO)
SENTENCE
P111 (VIEW PARKER TAP INCIDENT)
P112 (MOD PARKER DONNA)
P113 (ISA EXPERT PARKER)
P114 (MOD EXPERT COMPUTER CRIME)
P115 (AT PARKER SRI-INTERNATIONAL)
P116 (IN SRI-INTERNATIONAL CAL)
P117 (MOD CAL MENLO-PARK)
P118 (PART INCIDENT FAD)
P119 (MOD FAD LARGE)
SENTENCE
P120 (EXIST EPIDEMIC TAP BREAK-INS)
P121 (MOD BREAK-INS MALICIOUS)
P122 (MOD BREAK-INS COMPUTER)
SENTENCE
P123 (JEOPARDIZE TAP BREAK-INS PRIVACY)
P124 (MOD PRIVACY EVERYONE)
SENTENCE
P125 (UPSET BREAK-INS MANY)
P126 (NOT P125 CRYPTOGRAPHER)
P127 (DESIGN CRYPTOGRAPHER CODES)
P128 (MOD CODES COMPUTER)

P129 (MOD CODES SECURITY)
P130 (MOD SECURITY HIGH)
SENTENCE
P131 (EXIST DEMAND CRYPTOGRAPHER SKILL)
P132 (MOD DEMAND GREAT)
P133 (EXIST DEMAND YEARS FUTURE)
SENTENCE
P134 (WAS NOTICE TAMPERING LOS-ALAMOS)
P135 (WAS NOTICE TAMPERING SLOAN-KETTERING)
P136 (WHEN NOTICE QUICKLY)
P137 (AFTER NOTICE P137.1)
P137.1 (CALL FBI)
SENTENCE
P138 (CREATE FILES BANKS)
P139 (MOD FILES DECOY)
P140 (MOD BANKS DATA)
P141 (TRAP FILES GROUP)
P142 (MOD GROUP UNAUTHORIZED-USERS)
SENTENCE
P143 (WERE TIPS FBI)
P144 (MOD TIPS ANONYMOUS)
P145 (LED TIPS P146)
P146 (TRAP GROUP)
P147 (MOD GROUP KIDS)
P148 (MOD GROUP MILWAUKEE)
P149 (EXIST PANIC COMPUTER WORLD)
SENTENCE

APPENDIX 10

LTM MACROSTRUCTURE FOR COMPUTER CRIME TEXT



M7

APPENDIX 11

HOW TO USE STOP

What it does

The coherence program is designed to give you a number of descriptors for any input list of micropropositions and macroproposition. First, the program prints out a list of all micropropositions (indexed by their microproposition number) which have been held over in the buffer for one or more processing cycles. The microproposition number is printed and the number of processing cycles it was held over in the buffer follows it. Each microproposition and cycle property is printed on a separate line.

The second list output by the coherence program is the overall long term memory representation of the microstructure. The program prints a level in the microstructure header followed by a list of microproposition numbers. Next to each microproposition number is a list of all micropropositions at the next level in the microstructure that the microproposition points to.

The third item output by the program is a list of macropropositions occuring at each level in the macrostructure. Additionally, all macropropositions which the macroproposition points to are listed. Furthermore, the microproposition which is the root node of the subtree of micropropositions associated with each macroproposition is printed. To derive subtrees of micropropositions, one must graph the overall microstructure according to the levels and points-to properties printed out in conjunction with the overall microstructure statistics. Next, each microproposition which is pointed to by a macroproposition should be indicated (I put a square around the microproposition) in the overall microstructure. Then, all micropropositions which this item points to and the micropropositions which its subordinants point to, etc. are designated at being a part of the subtree, unless one of the micropropositions is pointed to by a macroproposition. These subtrees can then be individually graphed so that level within a subtree can be determined.

The last information output by the program is a list of statistics. Included in the list are the number of reinstatement searches conducted, the number of inferences performed, the number of times the buffer was stretched, the number of times the buffer was overloaded, and the number of processing cycles.

What to Input

The program takes as input two lists. The program will prompt you for each list. The first list asked for is a list of micropropositions. The way to format such a list is to begin by putting a right parenthesis on the first line of your file. On each consecutive line, place a single microproposition or clause or sentence marker. Micropropositions should be constructed according to the guidelines of Bovair and Kieras (1980). The follwing are examples of micropropositions (additional examples can be found in the sample list, in the file named BAY.MIC).

(P10 (MOD SCHOOL RED))
(P15 (WHEN P12 YESTERDAY))
(P20 (WENT SHERYL STORE))

As the predicate of the microproposition is not used in the argument overlap process, it is important that the predicate describe the relation (i.e. modify or mod, time-of, before, etc.) or be a verb. When you encounter a clausal boundary in a long sentence or a sentence marker, the word CLAUSE or SENTENCE should be placd on a line in the file after the last microproposition included in the clause or sentence. The last item in your list of micropropositions should be a right parenthesis.

The list of macropropositions is formatted similarly to the list of microproposition. The first line in your file should be a left parenthesis and the last line should be a right parenthesis. There are no sentence or clausal markers in the macroproposition list. Macropropositions are input in general terms, where the predicate is the macro-rule used in generating the macroproposition and the arguments are the micropropositions to which the macro-rule was applied. For example (M6 (GENERALIZE P28 P20 P24)) would be an acceptable macroproposition. This macroproposition would imply that P28 was the most important microproposition used in its derivation. Here important is operationally defined as most closely resembling in gist. A sample file of macroproposition is called BAY.MAC.

<u>What to do</u>

The following is a list of questions you must answer before the program will execute.

1- List of propositions? Here you type in the name of the file which contains your microproposition list.

2- List of macropropositions? Type in the name of the file which contains your macropropositions.

3- Name of output file or output to terminal? Terminal is signified by TTY: If you wish the output to go to a file, just enter the file name. The output in question is the cycling process whereby the microstructure, macrostructure and subtrees are created.

4- Name of prediction file? Here you must specify a file name. The program will output all of the statistics and descriptors described above into this file.

A copy of the simulation can be obtained by sending $15 and writing:

Walter Kintsch
Institute of Cognitive Science
University of Colorado
Boulder, CO 80309