ED 031 094

EM 007 305

Protosynthex III (PSIII) is a language processing system developed as an (as yet inadequate) experimental vehicle for testing student responses, with a view to constructing a model of an automated tutor. A version of the PLANIT system was modified so that a human tutor could be used to make instructional decisions in response to students' constructed answers to questions presented by PLANIT, and the resulting interactions used as examples of the breadth of language that the language processor eventually must straddle. The project is based on two interdependent assumptions yet untested: that the model will be effective as a teaching instrument (it demands from the student greater subject mastery); and that it can be constructed as a working computer system. The automated model is fed with a correct answer which is just sufficient to satisfy the query--the canonical answer (CA). The model then takes the student response (SR) and compares it with the CA. There are five possible relationships between the CA and the SR, ranging from equipollent, through partially irrelevant, to totally irrelevant, and the model directs the student accordingly. The automated tutor is still in its infancy, and a great deal of work is foreseen before it becomes a useful system. (GO)

LINGUISTIC ANALYSIS OF CONSTRUCTED
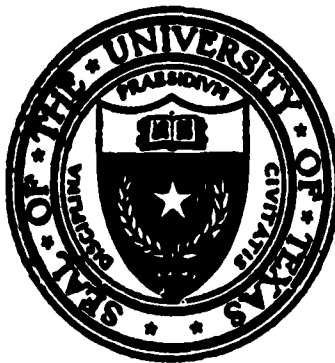
STUDENT RESPONSES IN CAI

by

Robert F. Simmons[*]

October 1968   TNN-86

THE UNIVERSITY OF TEXAS AT AUSTIN

COMPUTATION CENTER

AUSTIN

THE UNIVERSITY OF TEXAS AT AUSTIN

COMPUTATION CENTER

LINGUISTIC ANALYSIS OF CONSTRUCTED

STUDENT RESPONSES IN CAI

by

Robert F. Simmons*

,

*Professor Simmons is a member of the Department of Computer Sciences.

ABSTRACT

Any Computer Aided Instructional system that purports to deal
with natural English constructed responses requires a powerful natural
language processing system at its base. A tutorial decision model
for such a CAI system is presented and a set of example responses are
analyzed to demonstrate methods by which the language processor can
cope with constructed responses by students and generate minimally
correct tutorial interaction.

Linguistic Analysis of Constructed Student Responses in CAI

I. Introduction

Since July of 1968 the Systems Development Corporation has been engaged in a research program aimed toward the development of an experimental version of a computer aided instructional (CAI) system that can interact with students in a subset of natural English. The basic requirement of such a CAI system is a natural language processor that can successfully generate and semantically interpret a wide range of natural English constructions including sentences, fragments of sentences, questions and responses to questions. So far, a complete though shaky system for syntactic and semantic analysis, paraphrase, sentence generation, and question answering with respect to English sentences has been developed on the project. Detailed descriptions of the syntactic, semantic and logical approaches used in the language processor are to be found in Simmons et al. (1968a, 1968b), Burger et al. (1968) and Schwarcz et al. (1968). The overall research plan and design for the instructional system was described in Simmons and Silberman (1967).

In this paper, following a brief exposition of the notion of a natural language CAI system, the initial simplified decision model for the tutorial instruction system is described and a set of actual student responses are analyzed to demonstrate the methods by which the language processor can "understand" constructed responses of the student and generate minimally appropriate tutorial interaction.

II. The Natural Language CAI Notion

Various recent CAI systems typified by SDC; PLANIT,
(Feingold et al. 1966) offer some capability for dealing with re-
sponses constructed by the student as answers to system problems or
queries. Constructed responses have been found to be particularly
useful for teaching such formalized disciplines as logic, mathematics
or programming. In these areas, variations in form and content of
the constructed response can usually be dealt with by using an
algebraic evaluation system that can successfully recognize algebraic
equivalents of the required answer. However, if the constructed
response is allowed to be in the form of an English phrase, the
problem is vastly complicated by the lack of any comparable English
evaluation system for recognizing every meaning-preserving paraphrase
of an answer.

The difficulty in dealing with English paraphrases arises from
the basic flexibility inherent in natural languages. It is generally
the case that for any word or phrase in a natural language expression,
another sentence, word or phrase can be substituted to express almost
exactly the same meaning without changing the meaning of the larger
expression. With such a wide variety of expressive potential avail-
able, the lesson designer who tries to predict all correct variations
for a prescribed short English answer is doomed to failure.

Current systems recognize this difficulty and provide a few aids
for the lesson designer. PLANIT offers a logic for detecting and in-
noring misspellings, a root-form procedure to account for some varia-
tions due to inflectional suffixes, and a keyword facility for ignor-
ing all but significant content words in a student's response. ELIZA,
which has recently been claimed as a CAI system (Taylor 1968), includes
a pattern-operation logic. If a certain ordered pattern of words is
present, with or without regard for intervening words, control is passed
to a given operation such as the presentation of additional material or
the asking of further questions. If these approaches were further
augumented by the use of synonym dictionaries or thesauri, another step

could be taken to ease the handling of constructed English responses. However, our own experience with such non-analytic approaches to dealing with English meanings leads inescapably to the conclusion that they are and will remain hopelessly inadequate for the task. The lesson designer who wants constructed English responses--short or long--still requires a complete language processing system to recognize alternate expressions of the same meaning.

But, if he had a powerful language processing system in conjunction with his CAI machinery, he could do much more. He could develop a radically different concept of a CAI system, one that would minimize lesson programming requirements while it maximized individualized interactions with the student. The reason for this is that a language processor in order to recognize acceptable paraphrases of an answer, must deal successfully with meaning rather than with words. If it can deal with meanings, it can represent a lesson content as a set of meanings to be communicated to a student and it can measure the student's progress in terms of the amount of meaning that the student has absorbed from the lesson. If it could deal with paraphrases, answer questions and generate English statements then it could also interact with both student and lesson designer in relatively free English. Such a system could answer questions regarding lesson content whether they came from the student or from the instructor. It could generate English statements representing meanings in the lesson content. In evaluating student responses to questions, it could measure the student's knowledge of lesson content with regard to its own representation and use the discrepancy between the two as a basis for generating statements that the student could learn until the discrepancy disappeared.

In short, a CAI system based on a powerful language processor would soon lead to the design and construction of an automated tutor that could measure the position of a student with respect to the lesson content and use the discrepancy to generate materials that would allow the student to close the gap in a manner fitted to the student's own learning pattern. What we are suggesting is that an adequate

language processor must include many of the important symbol-meaning processing capabilities of a human. If these are available to a computer, they can be used in the manner that a human tutor uses them.

This line of thought and our research prog.am leading to the eventual design of an automated tutorial CAI system has been described in detail elsewhere (Simmons & Silberman, 1967). We have made appreciable progress in the development of a language processing system that can serve as an experimental vehicle for testing some of these ideas in a CAI environment. This vehicle, Protosynthex III, has also been described previously (Simmons et al. 1968b, Schwarcz et al. 1968). Protosynthex III (PSIII) is by no means the powerful language processor that would be required at the base of an automated tutor, but it is at least minimally sufficient to experiment with some actual student responses and so help develop additional knowledge for the eventual construction of an automated tutor.

As a first step in this direction, a version of the PLANIT system has been modified to use a live tutor to make instructional decisions in response to the students' constructed answers to questions presented by PLANIT. Protocols of interactions between human tutors and students have been collected to serve as samples of language exchanges that the language processor must eventually be able to encompass. A considerably simplified decision model of a tutorial system has been designed which uses the language processor to recognize discrepancies in student responses and to generate English statements that can help him to correct his errors. As this simplified model of the tutor is perfected, it will be embedded in PLANIT so that each time a verbal response is constructed by the student, PLANIT will use the tutorial decision system to understand, evaluate and shape the student response.

III. The Tutorial Decision Model

Our first approximation to an automated tutor requires the lesson designer to present his material as a sequence of text interspersed with or followed by queries to the student. These queries require short constructed responses from the student. For each query, the designer formulates a complete correct answer which contains only the necessary and sufficient information to answer the query. This complete answer is called the Canonical Answer (CA). The Student Response (SR), is taken by the language processor and tested to determine if it is an equivalent paraphrase of the CA. The SR is expected to vary widely in choice of vocabulary and phrasing from the CA and the language processor has the function of determining in what ways the meaning content of the SR corresponds to and differs from the content of the CA.

The five possible relations between the SR and the CA, exemplified by SR1 through SR5, are illustrated in the Venn diagrams of Figure 1. In Case 1, shown by the exact coincidence of a dashed outline with the solid outline representing the CA, there is no difference between SR1 and CA. SR1 is complete and correct. Case 2 is illustrated by SR2 where the dashed SR and the solid CA are completely disjoint. For this case, the student response is completely incorrect and irrelevant. SR3 shows the case of a partially correct answer that also contains some incorrect or irrelevant information. Case 4, illustrated by SR4 is another example of a partially correct response, but without any incorrect or irrelevant information.
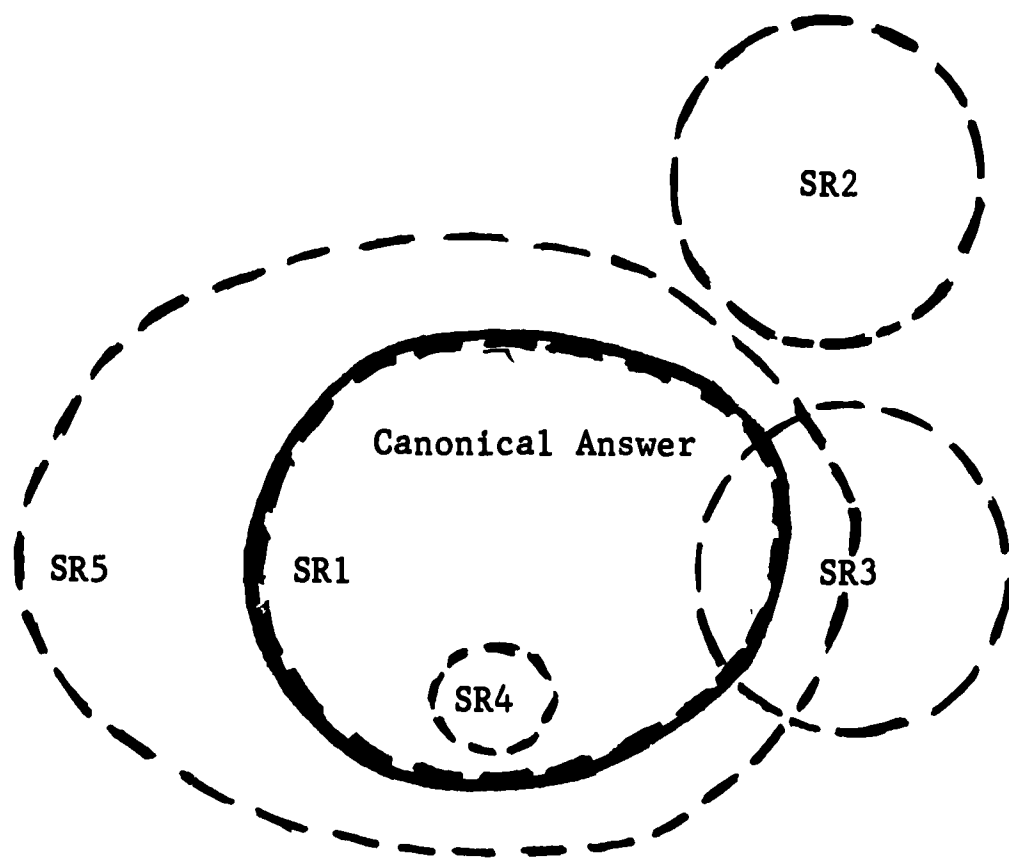
Figure 2. Comparison of Student Responses (SR)

with Canonical Answer (CA)

The final case, SR5, includes the entire correct answer along with some additional irrelevant material.

Our use of correct, incorrect and irrelevant is strictly with reference to the canonical answer which is defined as the only correct and complete (i.e. necessary and sufficient) information to answer the query. The categories of correctness and relevance into which an SR is classified dictate the tutorial decisions that the model makes. Table 1 summarizes these decisions showing the types of positive and negative reinforcement given to the student and the subsequent lesson materials provided for him.

For the completely correct and relevant Case 1, one of the positive feedback messages such as "OK" "That's completely right" etc. is generated and the next frame of the main sequence is presented. For SR3 and SR4 that are partially correct, the feedback tells the student that his answer is partly right and the language processor generates a new query from the part of the CA that the student has missed and uses that part as the new CA. For Case 2, where the SR completely misses the CA, the feedback is a negative statement such as "No, that's not at all correct" followed by a paraphrase of the original query. If Case 2 occurs twice in sequence, the relevant lesson information is presented again by the same query. For Case 5 which includes the complete CA but also some irrelevant material the feedback is a positive message such as, "This is the correct part of your answer, the rest is irrelevant". This message is followed by an English statement representing the CA.

Case	Correctness	Relevance	Feedback Type	Decision
I	Correct	Relevant	Positive	Continue Lesson
II	Incorrect	Irrelevant	Negative	Generate Do you remember SUBJ
III	Part Correct	Part Irrelevant	Positive Incomplete	Generate Do you remember (Omitted SUBJ)
IV	Part Correct	Relevant	Positive Incomplete	Generate Do you remember (omitted SUBJ)
V	Correct	Part Irrelevant	Positive Redundant	Generate This part of your answer is enough: (CA)

Table 1. Summary of Decision Model

In Cases II, III and V, the incorrect content of the answer is generally ignored and the student is brought back to the lesson content by generating new statements from the untouched portion of the canonical answer. We made the decision to ignore most incorrect content both because it simplified the tutorial model and because we believe that in most cases, working with incorrect content may have the effect of confusing the student. In this first tutorial model, our goal is to shape the student's responses toward the desired content by using positive and negative reinforcement supported by partial repetitions primarily in the form of hints generated from the CA.

It can be seen that this model greatly reduces the effort required from a lesson designer. Instead of having to predict for each query all categories of possible responses from the student and to decide ahead of time what to do for each, he is able to produce a main line of lesson content and allow the tutorial system to deal with individual variations in the student's response patterns. Although the tutorial system assumes a single main line of lesson content, once it is sufficiently developed to be embedded in a general purpose CAI system, there is no reason why the designer cannot include branching lessons at a level above the tutorial system. As the tutorial system is further developed it will probably also be possible for the lesson designer to program appropriate branchings depending on the categories of discrepancy between SR and CA.

The preceding description of the tutorial model is based on two assumptions that are still untested. Perhaps the more important of these has to do with the effectiveness of the model as a teaching instrument. Our faith in its eventual effectiveness is based on the belief that constructed responses demonstrate a greater mastery of subject content at a higher level of understanding than rote learning and that consequently, the material learned will transfer more easily to various applications. Certainly, the constructed response requires of the student not only that he remember the text that he has read, but that he also be able to generate English statements of his own that reflect the same content. The extent to which such a tutorial model may

in fact improve the effectiveness of CAI systems is one that can eventually be answered only by experimental comparisons. But such comparisons depend on the validity of the second assumption, which is that the tutorial model can actually be constructed as a working computer system.

It is with regard to the achievability of the natural language based tutorial system that our present line of research is mainly concerned. Many major linguistic and logical problems must be defined and resolved before our present language processor, PSIII, becomes an adequate vehicle to serve as a basis for the tutorial system. Several versions of the tutorial decision model will probably have to be programmed before any useful system is developed. However, the line of development will be seen to be fairly clear as we show in the next section, the linguistic and logical analyses that are made in order to represent the meanings that underly the phrases and fragments of English used in student responses.

IV. Analysis of Student Responses

The tutorial model imposes two difficult requirements on a language processor. These are first, that it be able to recognize and measure the extent to which any two English statements are equivalent paraphrases of each other and, second, that it be able to generate English statements that express the meaning of any student response or canonical answer. In order to meet these requirements, a language processor must be able to analyze an English phrase or sentence into an underlying logical structure of non-verbal objects that represent concepts or ideas. In conjunction with the analysis process, it must also have the capability of synthesizing, i.e. generating, English statements from these underlying structures to represent their meaning in English. A third requirement is implied that if two underlying logical or conceptual structures are equivalent in meaning, there should be rules of inference that can be used to transform one into the other.

To attain these requirements, a language processing system uses syntactic and semantic analysis functions to read and transform text into a deep structure of concepts whose defining attributes and whose relations with other concepts are made explicit in a well-defined, quantified logical structure. The deep conceptual structures we are using in our most recent versions of PSIII represent the meaning of an English statement as a verb or relational concept in various case relations to the nominal concepts of the sentence. In using this structure we are indebted to ideas developed by Fillmore in his recent papers, (Fillmore 1967, 1968) but we have modified his structure both for convenience in programming and to obtain consistency with our conceptual model as described elsewhere (Simmons et al. 1968a).

The resulting structure can be understood with reference to the example sentence, "The old man bought a boat from Tom." The main verb idea of the sentence is a buying, so this is the head structure. "The old man" is not only the subject of the sentence, but an animate, active person and is thus in the agentive case. "A boat" is the direct object of the verb, and the indirect object, "Tom," is technically

in the dative case. However, since the preposition "from" carries more information than the case "dative," we mark the case relation as "from". The general idea behind this approach of case relations is that in most natural languages, nominal concepts can be considered to be the objects of prepositions that mark their case-relations to verbs. In English such case-relations as subject, agent, object, dative, and instrumental are frequently not marked by an explicit preposition, but the information is carried instead by a term's position with relation to the verb or by characteristics (i.e. semantic features) of its meaning. (In some languages, for example Japanese, subject and object are explicitly marked by pre- or postpositional words.)

After analysis, the example sentence has the following structure:

Buy TENSE Past, AGT((Man MOD old) Q 1), OBJ (Boat Q 1),
FROM (Tom Q All).

A particular man distinquished by the MODifier, "old", and the article "the", is the subject and agent. The nominal "boat" is quantified (Q), by the article "a" as one of a class of such concepts.* The proper name "Tom" is considered (in this example) as signifying all of the concept of Tom as a particular person. Several alternative sentences would all have been analyzed to result in the same structure. For examples,

a) A boat was bought from Tom by the old man.

b) From Tom, the old man bought a boat.

c) The man who was old bought a boat from Tom.

etc.

---

* In most of the following examples, we will ignore quantification, which is a most difficult aspect of natural languages and one for which we are still attempting to work out a satisfactory approach. Also, the actual representation of a concept is a pointer to a particular sense of meaning for a word. However, to keep our exposition clear we will continue to use words to represent concepts in all examples.

Language processors based on this idea of structure may be weak or powerful depending on the depth of the conceptual structure and the generality of the rules for deduction. One structure is deeper than another if the first is able to assign identical structural descriptions to two English statements that mean the same thing when the other cannot. Let us consider two sentences that most people would agree are equivalent paraphrases of each other:

       1) Mary bought a boat from Tom.

and 2) Tom sold a boat to Mary.

One system might compute the following conceptual structures:

    1a. Buy TENSE Past, AGT Mary, OBJ boat, FROM Tom.

    2a. Sell TENSE Past, AGT Tom, OBJ boat, TO Mary.

A second might compute:

    1b. Exchange TENSE Past, TO Mary, FROM Tom, OBJECT Boat, FOR Value.

    2b. Exchange TENSE Past, TO Mary, FROM Tom, OBJECT Boat, FOR Value.

Since the second system analyzes the two sentences into identical structures, 1b and 2b, the second type of analysis results in a deeper structure than that of the first system that produced nonidentical structures 1a and 2a. Since the identity of meaning of the two sentences is immediately apparent in the 1b-2b analysis, the second language processor can be said to be more powerful than the first, (i.e. it accomplishes more work in its analysis than the first).

However, the first system can be made quite as effective for answering questions as the second by adding deduction rules such as the following:

    D1. Buy AGT X1, OBJ X2, FROM X3 = Sell AGT X3, OBJ X2, to X1.

    D2. Buy = Exchange FOR Value, TO $\langle$AGT$\rangle$.

    D3. Sell = Exchange FOR Value, FROM$\langle$AGT$\rangle$.

Rule D1 is a transformation of one structure pattern into an equivalent one that rearranges the variables X1, X2, and X3 in accordance with the rule. D2 and D3 define buy and sell in terms of the more abstract verb, "exchange", differentiated by the qualification "FOR Value" and the sub-stitution of a prepositional relation for the agentive noun. Thus by operating the rules D1 through D3 on structures 1a and 2a, the deeper structures of 1b and 2b result.

It is apparent that even deeper structures than 1b and 2b can be derived by the use of additional deductive rules. For example, "buy" and "sell" imply not just an exchange of objects, but an exchange of the property of ownership of the objects. Thus a deeper structure for the example sentence is as follows:

    1c. Exchange FOR (Value ATTRIBUTE Ownership)
                  OBJ (Boat ATTRIBUTE Ownership)
                  TO Mary
                  FROM John

Even deeper structures defining exchange in terms of two reciprocal changes in ownership from Time 1 to Time 2 between Mary and John can be developed. Each such deeper structure makes more explicit the inferences that can be made from the sentence. In human thought patterns, there is probably no deepest structure; but for any computer system there is probably an optimal depth for any given purpose.

Optimization of depth of structure for a computer language processor depends on how frequently questions that require subtle inferences are to be asked. If they are frequent, the deeper structure is desirable to avoid frequent and redundant application of the rules of inference. If, on the other hand, few deep inferences are required (as might be the case in a document retrieval system) a much shallower structure supported by many rules of inference will suffice. In either case, however, since there is no apparent limit to depth of structure, deductive inference rules that transform one structure into another are required by a language processor.

This apparent digression into an examination of depth of conceptual structures actually shows that equivalent paraphrases are statements that either have the same deep conceptual structure, or which can be transformed

into each other by some set of deductive rules. If two statements are not
equivalent paraphrases, the extent to which they agree can be measured by
the degree of similarity and difference in their conceptual structures and
in terms of the proportions of those structures that can be transformed into
each other.

Thus for the tutorial model, the language processor can measure the
paraphrastic equivalence of the student response and the canonical answer
in terms of the similarity of their respective deep conceptual structures.
The portion of the CA that is not equivalent or transformable (hence trans-
formationally equivalent) to the SR, then serves as a basis for generating
an appropriate hint or new query.

The syntactic and semantic processes the language processor uses to
transform from English into the deep conceptual structure are quite com-
plex and a detailed description would require more space than is available
in this paper. We will, however, attempt to describe briefly the essence
of the process for generating English from the conceptual structures, allow-
ing the interested reader to consult previous descriptions (Simmons et al. 1968b)
for detailed treatment.

Generation from structure to English string is accomplished by taking
each triple of structure as a possible left half of a generation rule. Let
us consider again example la below.

la. Buy TENSE Past, AGT Mary, OBJ (boat Q 1), FROM Tom.

Structure triples are, Buy TENSE Past, buy AGT Mary, boat Q 1, etc. The
first and third terms in each triple are expressed by a word or a morpheme
while the middle relational term may be expressed as a word or a function to
accomplish a combination of morphemes or to translate a first or third term
into a word. Associated with each element of structure is an appropriate
syntactic class; thus Buy-VP, TENSE-function, Past-ed, AGT-O, Mary-NP,
Q-function, 1-Determiner, etc.

A subset of generation rules sufficient to transform this structure into an English sentence is shown below.

```
NP Q Det    -  (TMOD(C,A))NP = a boat
VP TENSE Past - (TENSE(A,C))VP = bought
VP Prep NP  - (A B C ) VP = bought from Tom
VP OBJ NP   - (A C) Pred = bought from Tom a boat
PRED AGT NP - (C A) Sentence = Mary bought from Tom a boat.
```

Terms such as TMOD and TENSE are functions that are applied to the arguments in the following parentheses. Abbreviations such as Q for Quantifier, Det for Determiner, NP for Noun Phrase, VP for Verb Phrase, Pred for Predicate are common linguistic usage. The letters A, B, and C refer respectively to the first, second and third terms of the left half of the expression. The phrases following the equal sign after each rule show the result of the system's applying the rule and its functions to the structure to produce a string of English words.

The generator that applies these rules is a fairly simple routine that takes first the most deeply nested triple, derives its syntactic form, e.g. "NP Q Det" from "boat Q 1,", looks it up as a left half of the rule, applies the rule to convert the structure, and then continues from left to right with the remainder of the structure, until the structure is completely expressed in English or it fails for lack of an appropriate rule. If a rule contains a function such as TMOD or TENSE, this generation routine operates the function to obtain its result which then becomes part of the English string. Given any set of structures, the generator will produce one or more English sentences to communicate its meaning. With the use of certain introductory phrases which can be elements in rules and the appropriate choice of structures, this generator is an adequate first approximation to what is needed for the tutorial model.

Some Example Analyses of Student Responses

As mentioned earlier, a version of PLANIT was modified to allow a human tutor to accept free English responses from students and to make instructional decisions. Preliminary experiments with this system have given us a number of protocols chat show the text that was presented, the canonical answer, and various student responses. Several CAs and SRs so obtained will be analyzed in this section to illustrate particular aspects of the logic used by the language processor for determining the extent of paraphrase and for generating an appropriate instructional response.

Following the presentation of appropriate text material concerning the anatomy of the eye the first query in the lesson was:

"The primary image-forming function accomplished by the cornea and aqueous humor working together is to...? (COMPLETE THE SENTENCE)

The canonical answer (CAI) was:

"The function is to ..bend light to form an image on the retina.

The student response (SR1) was:

"(The function is to)...bend light rays for focussing on the retina.

The CA and SR are quite similar to a human reader but even after analysis, the concept structures as diagrammed in Figure 2, do not precisely match.

First, however, a few remarks to explain Figure 2. The CA contains three embedded sentences which can be expressed roughly as follows:

1. Function is to bend etc.
2. Bending of light is to form etc.
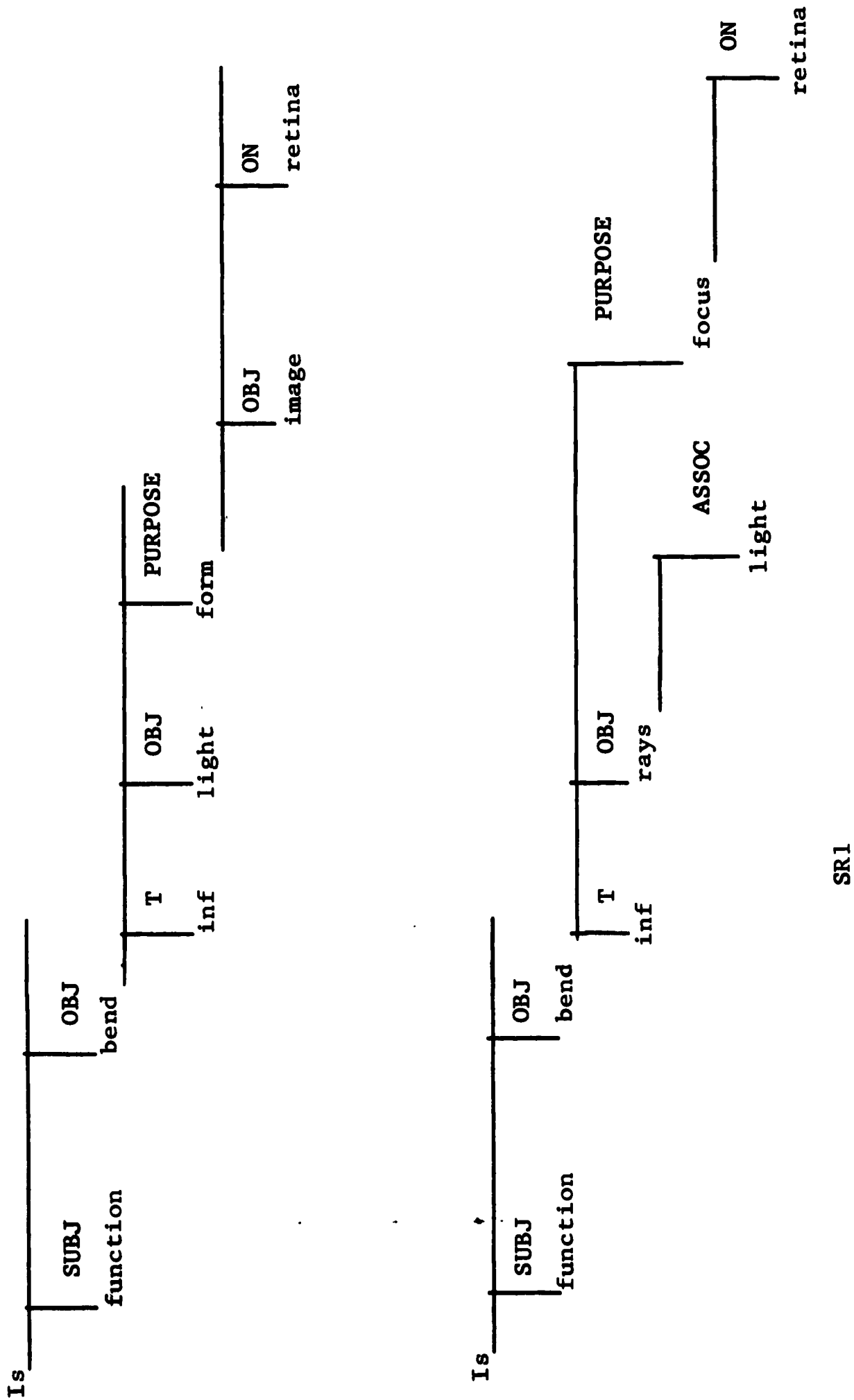3. Forming of image is on retina.

Figure 2. Diagrammatic Comparison of Canonical Answer and Student Response

The SR contains the following four embeddings:

1. Function is to bend etc.
2. Light rays are bent to focus etc.
3. Rays are associated with light.
4. Focussing is on retina.

Each of these embeddings corresponds to a downward step in the diagrams of Figure 2. With the exception of "rays ASSOC light" each is headed by a verb. Relational terms such as SUBJ, OBJ, T (for TENSE), PURPOSE, etc. are labels for the vertical lines in the diagrams. The diagrammatic presentation is exactly equivalent to the previous notation which used commas, parentheses and semicolons to show the nested nature of the relationships. For example, CA1 can also be expressed in that notation as shown below:

        Is SUBJ function, OBJ(Bend T inf, OBJ light,
            PURPOSE (Focus OBJ Image, ON retina).

A different arrangement using the same notation but eliminating parentheses is as follows:

        Is SUBJ Function,
            OBJ bend
                    T Inf
                    OBJ Light
                    PURPOSE Form
                                OBJ Image
                                On retina.

These various forms of notation are all equivalent expressions of the structure and will be used where appropriate as we continue the illustration.

The reader should note expecially in Figure 2 that where an embedded sentence is signified by a downward step (or by a parenthesis or a shift right in the other notations), everything in that embedding is an argument of the relation. Thus "bend" and the entire remaining structure of CA1 is in an OBJ relation to "Is".

Comparison to determine if SR1 is an equivalent paraphrase of CA1 is accomplished by examining each embedded structure as a subunit, from the most deeply embedded structure up to the top structure. Table 2 below shows the comparisons to be made.

Depth	CA1	SR1
3	Form OBJ Image, ON Retina;	Focus ON Retina;
2	Bend OBJ light, PURPOSE 3	Bend OBJ 2a, PURPOSE 3;
2a		Rays ASSOC light;
1	Is SUBJ Function, OBJ 2;	Is SUBJ Function, OBJ 2;

Table 2. Comparison of Embedded Structures
in CA1 and SR1

Starting with Depth 3, the language processor compares the two structures, Form OBJ Image, ON retina; and Focus ON retina. Since they do not match on "Form" and "Focus", these words are looked up in the system's dictionary to discover associated definitions,Def., (signified by =) and inference rules,Drul., (identified by⇒). As it applies each inference rule or definition, the system rewrites the structure and makes the comparison again. Table 3 summarizes the steps taken by the system, showing the inference rules used successfully and the result of applying each until, for this example, the SR is transformed to match the CA.

Depth 3 Comparison:

       Focus ON Retina vs. Form OBJ Image, ON retina.

Difference: Focus vs. Form OBJ Image

Def: Focus = Form OBJ (Point OR Image)

Difference: Form OBJ (Point OR Image) vs. form OBJ Image

Drul: (X1 = X2 R (X3 OR X4)) $\Rightarrow$ (X1 = X2 R X3)

                             or (X1 = X2 R X4)

Result: Form OBJ Image, ON retina.

Difference: 0

Depth 2 Comparison:

       Bend OBJ (rays ASSOC light), PURPOSE 3 vs. bend OBJ light, PURPOSE 3

Difference: Rays ASSOC light vs. light

Drul: X1 ASSOC X2 $\Rightarrow$ X2

Result: Bend OBJ light, PURPOSE 3

Difference: 0

Depth 1 Comparison:

       Is SUBJ function, OBJ 2 vs. Is SUBJ function, OBJ 2.

Difference = 0 = Case 1

Table 3. Computation of Similarities
Between SR1 and CA1

In the Depth 3 comparison, the first result is the difference between
"focus" in the SR and "form OBJ image" in the CA. The definition (Def) of
"focus" is next shown to be equivalent to the structure, "form OBJ (point
OR image)" so this equivalence serves as a deduction rule (Drul) that
allows the definition to be substituted for the word. The difference
between "form OBJ point or image and form OBJ image" still remains. Since
a definition containing or-ed elements implies a set of alternate definitions,
the deduction rule used is true and applicable (where X1, X2, X3, X4 and R)
are positional variables that can stand for any term). The result of apply-
ing this rule is to reduce the differences in Depth 3 to zero.

In the depth 2 comparison both SR1 and CA1 include the triple
"bend PURPOSE 3." The number, 3, refers to the Depth 3 structures which

have already been found to be transformationally equivalent. In Table 2 on page 19, the Depth 2 structure for the SR1 was "bend OBJ 2a, PURPOSE 3" where 2a referred to the structure, "rays ASSOC light." To clarify the comparison task, this structure has been used in Table 3 in the Depth 2 comparison. The result of this comparison is the difference between "rays ASSOC light" and "light." The relation ASSOC is symmetrical and also has the property of left collapsibility.* Therefore the deduction rule, "X1 ASSOC X2 $\Rightarrow$ X2" is applicable and the difference is reduced to zero. Finally in the Depth 1 comparison, the two structures are found to match exactly.

The whole computation shows that the SR fits Case 1 and should transfer control to the generator which produces a positive reinforcement and continues with the lesson. Subsequent illustrations will be presented briefly with comment limited to that needed to explain unusual situations.

The next query in the lesson was presented as follows:

The iris and pupil work together to...?

The canonical answer was:

Regulate amount of light entering the eye.

The student's response was:

Accommodate for light intensity.

Table 4 shows the embedded structures in the SR and CA and the detailed comparison of each depth level. The depth 3 comparison failed completely. The depth 2 comparison used a definition (DEF.) and a deduction rule (DRUL) successfully to discover that intensity of light means amount of light. The depth 1 comparison also failed, so the only correct portion of SR2 was "light intensity" which allows the substitution of the marker "CORRECT" in CA2 at the point that it called for "amount of light."

---

* Left collapsibility is a property that allows the head of certain linguistic constructions to substitute for the whole construction. It is defined as follows: If R is left collapsible, (X1 R X2) $\Rightarrow$ X1.

From the original query and the remainder of the CA the generator is now allowed to give the deepest structure "entering the eye" as a hint and to use the correct portion of the student's response and so generate the following two statements.

Light intensity is part of the answer, try this:

The iris and pupil work together to _____ intensity of light entering the eye.   (FILL IN THE BLANK).

In the example of Table 4, the student's use of "accommodate" shows a basic confusion which a tutor should be expected to note and correct. While our present design does not allow for this, we have eventual plans for a list of words and concepts significant to the lesson and when any one of these is used in an incorrect manner the tutor will first attempt to correct that concept, then return to the frame at hand.

Depth	CA2	SR2
3	enter SUBJ 2, OBJ eye;	None
2	light ASSOC amount;	light ASSOC intensity;
1	regulate OBJ 2.	accommodate FOR 2.

4a Embedded Structures in CA2 and SR2

Depth 3 Comparison:

NONE vs. enter SUBJ 2, OBJ eye;

DIFF: CA2

DRULES: NONE

DIFF: enter SUBJ2, OBJ eye;

Table 4   Treatment of SR2

Depth 2 Comparison:

> light ASSOC intensity; vs. light ASSOC amount

DIFF: Intensity vs. amount

DEF: Intensity = amount ASSOC (force OR energy)

DRUL: X1 ASSOC X2 = X1

Result: Intensity = Amount

DIFF: 0

Depth 1 Comparison

> accomodate FOR light vs. regulate OBJ light

DIFF: Accommodate FOR vs. regulate OBJ

DEF: accommodate
          = adjust OBJ lens, PURPOSE (see AT (distances MOD different))

DEF: regulate = bring OBJ (order OR uniformity)

DIFF: greater than before.

4b Comparison of Differences

Correct Portion: light ASSOC intensity

Remainder of CA: regulate OBJ (CORRECT SUBJ^{-1} (enter OBJ eye)).

CASE: 3

Generate: The iris and pupil work together to _____
                    intensity of light entering the eye.

4c. Response Generation

Table 4. (cont.) Treatment of SR2

The two preceding examples have illustrated response Cases 1 and 3, the completely true and partially true cases. The next illustration exemplifies Case 2 where there is no correspondence between the SR and CA. After appropriate text the following query was presented:

"Explain why the process of accommodation deteriorates with age."

The CA:

"Lens becomes increasingly rigid with advancing age."

The SR:

"The ciliary muscles are losing their ability to contract."

The language processor found no similarity between these two responses. Instead of simply repeating the query, as the present tutorial model allows, it is apparent that the system would be improved by a generation such as "That's not the answer, try again."

"The lens becomes...?"

Such a generation, taking the subject of the CA as a hint, would be easily possible in this case, but we don't yet know how general such a rule would be.

A case 4 imcomplete but correct partial answer is illustrated by the next example. In this case the query was:

"Despite their need for nutrients, can you think of any anatomical elements in the eye that should be isolated from the blood vessels?"

The CA was:

"Cornea, lens, vitreous and aqueous humors."

The SR was:

"The vitreous humor."

Since the CA has the syntactic structure of a simple list, the system can easily be expected to count the remaining members and generate:

"Vitreous humor is one correct answer, try again. There are three more, _____, _____, _____."

So far we have obtained only a single example of a Case 5 response where the SR has more information than the CA. The query was:

"How does the lens accommodate in human vision?"

The CA:

"Lens shape changed by ciliary muscles."

The SR:

"The ciliary muscles change the convexity of the lens."

The only comparison problem in this pair of responses was to find the correspondence between "lens ASSOC convexity" and "lens ASSOC shape." Part of the definition of convexity is "convexity SUP shape," so convexity implies shape but it is a particular curved shape so it carries more information than the CA required. The tutorial system design calls for generating:

"Lens shape changed by ciliary muscles, is the correct part of your answer; the rest is irrelevant."

This would be a rather surprising response which might be expected to arouse some bitterness in the student since convexity of the lens is certainly relevant to accommodation. It would obviously be more appropriate to accept this as a Case 1 correct response. However, exactly how to recognize and deal with Case 5 responses will have to wait upon our obtaining a larger sample of actual cases.

VI.   Discussion and Conclusion

We have discussed the problem of dealing with English constructed responses, presented the design for a tutorial system based on a natural language processor and illustrated its operation by the analysis of some example student responses. The PSIII language processor is sufficiently well developed that it can accept definitions, deductive rules and generated responses using a generation grammar. However, before we use this system for this purpose, many analyses of the type presented must first be made and the tutorial system must be programmed to account for their variations in form and content.

Although the system we are currently building is a large one with 4-8 million words of storage available to it, it is still a moot question as to whether we can supply a sufficiently large vocabulary, grammar and system of deductive rules to allow relatively free interaction with students. We believe that the design of a tutorial system is basically sound and that our methods for analyzing student responses are basically correct. Nevertheless, we are painfully aware at this stage of our research that there is a long hard path still ahead of us before our designs and techniques coalesce into a programmed system that will allow us to test our basic hypothesis that a tutorial system is superior in teaching effectiveness to other forms of CAI.

REFERENCES

1. Burger, J.F., Schwarcz, R.M., Simmons, R.F. Users Guide and Program Descriptions for Protosynthex III. System Development Corporation, Santa Monica, California, September, 1968, In Press.

2. Feingold, S.L. and Frye, C.H. Users Guide to PLANIT, Programming Language, for Interactive Teaching. System Development Corporation, Santa Monica, California, TM-3055/000/01, October, 1966.

3. Fillmore, C.J. A proposal concerning English prepositions. Monograph Series on Languages and Linguistics, No. 19, 1966, Georgetown University Institute of Languages and Linguistics.

4. Fillmore, C.J. The Case for Case. Pre-publication MSS. Linguistics Department, Ohio State University, Columbus, 1967.

5. Simmons, R.F., Burger, J.F., Schwarcz, R.M. A Computational Model of Verbal Understanding. Proceedings AFIPS, Fall 1968, In Press. Also, System Development Corporation, Santa Monica, California, SP-3132, April, 1968.

6. Simmons, R.F. and Burger, J.F. A Semantic Analyzer for English sentences. System Development Corporation, Santa Monica, California, SP-2987, January, 1968.

7. Simmons, R.F. and Silberman, H.F. A Plan for Research toward computer-aided instruction with Natural English. System Development Corporation, Santa Monica, California, TM-3623, August, 1967.

8. Schwarcz, R.M., Burger, J.F., and Simmons, R.F. A Deductive Question Answerer for natural language inference. System Development Corporation, Santa Monica, California, September, 1968, In Press.

9. Taylor, Edwin F., Ed. ELIZA; A Skimmable Report on the ELIZA Conversational Tutoring System. Educational Research Center, MIT, March, 1968.