Proceedings of the Annual Meeting of the Cognitive Science Society

From Children's Arithmetic To Medical Problem Solving An Extension of the Kintch-Greeno Model

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 9(0)

Authors

Groen, Guy J.

Jerney, Janos P.

Publication Date

1987

Peer reviewed

FROM CHILDREN'S ARITHMETIC TO MEDICAL PROBLEM SOLVING: AN EXTENSION OF THE KINTSCH-GREENO MODEL

Guy J. Groen & Janos P. Jerney
McGill University

ABSTRACT

It has been found that expert physicians use forward reasoning in diagnostic explanations of clinical cases. This paper shows that the Kintsch-Greeno model for solving arithmetic word problems, which assumes a forward chaining process, can be extended to explain this phenomena. The basic approach is to modify the lexicon and the schema structure of the existing simulation program while retaining the basic control structure. The principle modifications are in the structure of the schemata which make use of three slots: indicator, abnormality and consequence. As with the Kintsch-Greeno theory, the model proceeds by using these schemata to build super-schemata from the propositional representation of the problem text.

From Children's Arithmetic to Medical Problem Solving: An Extension of the Kintsch-Greeno Model

Guy J. Groen & Janos P. Jerney
McGill University

Recently, two approaches have evolved that specifically combine the methodologies of propositional analysis and protocol analysis. The first is the theory by Kintsch and Greeno (1985) of the processes used in solving algebraic and arithmetic word problems and the series of computer simulation programs that are based upon it (Dellarosa, 1986). Such problems pose a simple arithmetic problem as a story. The simulation begins with a propositional representation of the story which it transforms into a set of frame structures called proposition frames. On this basis, it builds "set frames" that represent each number specified in the story as a set of objects. It then builds a "superschema" that specifies the relational structure betweeen these sets. The presence of a satisfactory superschema triggers an appropriate algorithm that generates the answer. If an incomplete superschema has been created, then the program produces "intelligent guesses" about the answer. Crucial to this theory is the distinction (van Dijk & Kintsch, 1983) between a textbase and a situation model. The textbase is the semantic representation of the input text. The situation model is the representation of the knowledge required to solve the problem. In the simulation program, the textbase is the proposition frames. The situation model is the final superschema.

The second is an approach developed by Patel and Groen (1986) to study clinical reasoning in medicine. This involves transforming a propositional representation of a reasoning protocol into a semantic network, and deriving from this a set of production rules that are adequate to simulate the reasoning task. This was applied to study reasoning in a task involving the diagnosis of a case of acute bacterial endocarditis. The subjects were seven specialists in cardiology. It was shown that the diagnostic explanations of subjects making an accurate diagnosis could be accounted for in terms of a model consisting of pure forward reasoning (i.e. from data to diagnosis) through a network of causal-conditional rules, actuated by relevant propositions in the stimulus text. In contrast, subjects with inaccurate diagnoses tended to make use of a mixture of forward and backward reasoning, beginning with a high level hypothesis and proceeding in a top-down fashion to the propositions embedded in the stimulus text or to the generation of irrelevant rules.

It may seem that childrens' arithmetic word problems are a world apart from clinical cases in medicine. However, both involve situations in which the task is to make inferences from narrative discourse. The simulation program, with appropriate parameters, performs like an expert in its domain and has one feature that is extremely important for our purposes. It builds the superschema from the initial frames by means of a simple, well-known forward chaining procedure (Winston & Horn, 1981) on the basis of a set of production rules. Since the strongest finding by Patel and Groen was the use of forward reasoning by experts with accurate diagnoses, it seems reasonable to explore at a more precise level how the two approaches are related.

Groen and Patel (in press) have proposed that the van Dijk-Kintsch (1983) theory of comprehension can be combined with a generalization of the Kintsch-Greeno model and Dellarosa's program to yield a simulation model of expert diagnostic reasoning in medicine. Its important aspects would be as follows:

- The reading of a text results in the creation of a propositional textbase and a frame based situation model.
- Reasoning tasks result in an elaboration of the situation model.
- Expertise resides in the rules that develop the situation model. More satisfactory rules result in a more satisfactory situation model.
- An individual will apply these rules by a process of forward chaining until he or she is aware of an inaccurate or incomplete situation model. This accounts for the phenomenon of forward chaining by experts, but does not preclude forward chaining by the less than expert.
- An individual's awareness of an inaccurate or incomplete situation model will result in some form of backward chaining.
- The situation model (in a preliminary form) will also affect the structure of the textbase by means of various strategies specified by van Dijk and Kintsch, which would also affect the nature of the macropropositions. An expert situation model would result in higher level macropropositions.

The purpose of the research reported in this paper was to use these notions to develop a program that is capable of accounting for the results concerning forward reasoning obtained by Patel and Groen (1986). The basic strategy was to use ARITHPRO, Dellarosa's instantiation of the Kintsch and Greeno model as a shell with which to develop it. This simulation runs on a XEROX 1108/1186 computer. The program includes a rule base written in Interlisp and a lexicon written in LOOPS. Within the rule base, seperate rules exist for 1) building the proposition frames of the textbase, 2) building the set frames and superschemata of the situation model, 3) encoding the procedural knowledge for solving arithmetic problems, 4) encoding the procedural knowledge for converting one type of problem to another and 5) encoding the procedural knowledge for default solution strategies enabling "good" guesses (consistent with guesses young children would make) given incomplete information. The lexicon, on the other hand, relies on LOOPS' inheritance system to create a taxonomy of the words likely to be encountered in the problem text.

Since ARITHPRO is currently the most complete and accurate instantiation of the Kintsch-Greeno model, every attempt was made to work within the existing framework of the program. As such, nearly all of the simulation's control structure was taken directly from it. In a similar fashion, most of the rules for constructing the proposition frames of the textbase were retained though slightly modified. The similarity ends here, however, as none of the set building or arithmetic counting procedures present in the Kintsch-Greeno model are applicable to the domain of medical problem solving. Therefore, a completely new set of rules had to be constructed to build the schemata necessary for diagnosing cases of acute bacterial endocarditis. In addition to this, the lexicon was modified to incorporate the new words appropriate for this domain.

In constructing the situation model, the simulation relies on a single general data structure called an Abnormality schema which contains three slots: 1) the abnormality slot which is usually a physiological disorder (such as an emboli) or disease category, 2) the indicator slot which is generally a primary clinical indicator of the abnormality (such as transient blindness) and 3) the consequence slot which can either be a clinical or physiological consequence of the abnormality. In the current implementation, these slots appear to be adequate for encapsulating the information the program needs to use in order to arrive at an acceptable solution. However, considerations are being made to expand the schema structure in future implementations to allow for the explicit storage of additional information. This is facilitated by the existing rule structure which allows for a graceful incrementation. The structure of an Abnormality schema along with an example is given in Table 1.

Table 1.
Structure of Abnormality Schema with example

```
(<schema name>   (INDICATOR (: <indicator list>))
                 (CONSEQUENCE (: <consequence list>))
                 (ABNORMALITY (: <abnormality>)))

(EMBOLI     (INDICATOR (: BLINDNESS))
            (CONSEQUENCE (: HEMORRHAGE))
            (ABNORMALITY (: EMBOLI)))
```

Our simulation, like ARITHPRO, does not have a natural language parser. Instead, a propositional representation of the input text is presented to the program. An example of a (simplified) problem case is presented in Table 2a along with a corresponding propositional representation in Table 2b which would be input to the simulation. In order to use ARITHPRO's existing rules to build the proposition frames, the general form of the input propositional representation had to be retained. In particular, the propositions with the head element DURATION are modelled after the quantity propositions in ARITHPRO. This proposition causes the goal MakeAbnormSchema to be placed on the goal list in much the same way that a quantity proposition creates a MakeSet goal in ARITHPRO. Subsequent rules then look into STM to find the abnormality. If the other elements of the premise are present in either short term or long term memory, the appropriate abnormality schema building rule is fired. An example of an abnormality schema rule (paraphrased in English) is given in Table 3.

The simulation proceeds by attempting to build an abnormality schema for each cue in the problem text. Only one schema is build for each abnormality and subsequent cues leading to an existing abnormality schema are appended to the indicator slot. When one or more of the indicators of an abnormality are themselves schemas, a superschema is created. The BACTERIAL-INFECTION superschema rule is shown in Table 4a while the actual superschema that it builds is illustrated in Table 4b. In this example, one of the indicators is a cue from the text, PUNCTURE-WOUNDS, while the other is the INFECTION schema. These superschemata are similar in function to the ones used by Kintsch and Greeno in that a logical relationship is established between the indicators the superschema subsumes. As with Dellarosa's program, key information such as the contents of STM (the textbase and the situation model), the goal list and the next rule to fire is constantly updated and displayed on the screen. The process continues until the end of the input text has been reached and no more schema building rules can fire. When this happens, the top-level superschema is returned as the most likely diagnosis and the simulation ends. The final result, for the case discussed in this paper, is a superschema for acute bacterial endocarditis.

The role of the consequence slots have not been completely defined or implemented yet. In the most general case, the consequences will be used to direct the focus of the schema building rules by placing various goals on the goal list. This process may be use to alert the simulation to follow up on certain cues and will probably be used extensively in a future implementation that includes backward chaining. This will enable a more complete account of the empirical phenomena found by Patel and Groen (1986) and Patel, Arocha and Groen (1986).

Table 2a.
English Text: Acute Bacterial Endocarditis

This unemployed young male was admitted to the emergency room complaining of a fever of four days duration. Functional inquiry revealed a transient loss of vision in his right eye which lasted approximately 45 s on the day before admission to the emergency ward. Funduscopic examination revealed a flame shaped hemorrhage. Examination of his limbs showed puncture wounds on his arm. Auscultation of his heart revealed a 2/6 early diastolic murmur in the aortic area. There was no splenomegaly. Urinalysis showed numerous red cells.

Table 2b.
Propositional Representation of Text

```
(((P1 (EQUAL X MALE))
 (P2 (ATT X P3))
 (P3 (DURATION UNEMPLOYED-YOUNG UNKNOWN)))
((P4 (COMPLAIN X P5))
 (P5 (DURATION FEVER 4-DAYS)))
((P6 (COMPLAIN X P7))
 (P7 (DURATION BLINDNESS 45-SEC)))
((P8 (HAVE X P9))
 (P9 (DURATION HEMORRHAGE UNKNOWN)))
((P10 (HAVE X P11))
 (P11 (DURATION PUNCTURE-WOUNDS UNKNOWN)))
((P12 (HAVE X P13))
 (P13 (DURATION EARLY-DIASTOLIC-MURMUR UNKNOWN)))
((P14 (HAVE X P15))
 (P15 (DURATION RED-BLOOD-CELLS-IN-URINE UNKNOWN)))
((P16 (HAVE X P17))
 (P17 (DURATION NORMAL-SPLEEN UNKNOWN))))
```

Table 3.
Emboli Schema rule

Rule Make-Emboli-Schema

IF: 1) MakeAbnormSchema is on the goal list, and
2) The indicator BLINDNESS is present in STM

THEN: 1) Create and add the schema EMBOLI to STM, and
2) Bind BLINDNESS to the indicator slot of the EMBOLI schema, and
3) Bind HEMORRHAGE to the consequence slot of the EMBOLI schema, and
4) Bind EMBOLI to the abnormality slot of the EMBOLI schema, and
5) Remove the MakeAbnormSchema goal from the goal list.

Table 4a.
Bacterial Infection superschema rule

```
Rule  Make-Bacterial-Infection-Schema

IF:      1)  MakeAbnormSchema is on the goal list, and
         2)  The indicator PUNCTURE-WOUNDS is present in STM, and
         3)  The INFECTION schema is present either in STM or LTM

THEN:    1)  Bring the INFECTION schema into STM from LTM if necessary, and
         2)  Create and add the superschema BACTERIAL-INFECTION to STM
         3)  Bind PUNCTURE-WOUNDS and the INFECTION schema to the indicator slot of the
             BACTERIAL-INFECTION superschema, and
         4)  Bind BACTERIAL-INFECTION to the abnormality slot of the BACTERIAL-INFECTION
             superschema, and
         5)  Remove the MakeAbnormSchema goal from the goal list.
```

Table 4b.
Bacterial Infection Superschema

```
[BACTERIAL-INFECTION
      (INDICATOR
            (INDICATOR1: (PUNCTURE-WOUNDS))
            (INDICATOR2: ((INFECTION
                                (INDICATOR (: (FEVER)))
                                (ABNORMALITY (: (INFECTION)))))))
      (ABNORMALITY (: BACTERIAL-INFECTION))]
```

It is important to note that this program is not designed to provide a direct simulation of subjects' behavior in the diagnostic explanation task examined by Patel and Groen (1986). Rather, it is designed to provide a detailed model of the diagnostic process in which the detailed rules, frames and schemata can be mapped onto propositions in diagnostic expanation protocols. This may enable a detailed examination of the hypothesis that the diagnostic explanation task reflects elements of the diagnostic process, which underlies much of our previous research. It may also enable the use more complex diagnostic problems, since it will remove the necessity of selecting clinical cases for which the logical validity of models of diagnostic reasoning can be verified by hand simulation.

It also begins to throw light on a far more general issue. There is currently a widespread belief both in Artificial Intelligence and Cognitive Psychology that problem solving is highly domain specific. Our attempt to generalize the Kintsch-Greeno model is a thrust in the opposite direction. As the program develops, it should become possible to make a far more precise differentiation between those aspects of problem solving that are domain specific and those that are common to many domains, regardless of their complexity.

ACKNOWLEDGEMENTS

This research was supported by grants from the Josiah Macy Jr. Foundation (No. B8520002) and the Province of Quebec.

REFERENCES

Dellarosa, D. (1986) A computer simulation of childrens' arithmetic word-problem solving. Behavior Research Methods, Instruments, and Computers, 18:142-154.

van Dijk, T.A. & Kintsch, W. (1983) Strategies of Discourse Comprehension. New York:Academic Press

Groen, G.J. & Patel, V.L. (in press) The relationship between comprehension and reasoning in medical expertise. In M. Chi, R. Glaser, & M. Farr (Eds.), The Nature of Expertise. Lawrence Erlbaum Associates.

Kintsch, W. & Greeno, J.G. (1985) Understanding and solving word arithmetic problems. Psychological Review, 92:109-129.

Patel, V.L., Arocha, J.F., & Groen, G.J. (1986) Strategy selection and degree of expertise in medical reasoning. In Proceedings of the Eight Annual Conference of the Cognitive Science Society. Lawrence Erlbaum Associates:NJ, 780-791.

Patel, V.L. & Groen, G.J. (1986) Knowledge based solution strategies in medical reasoning. Cognitive Science, 10:91-116.

Winston, P.H. & Horn B. (1981) LISP First Edition. Addison-Wesley.