COMPREHENDING WORD ARITHMETIC PROBLEMS:

A PSYCHOLOGICAL PROCESSING MODEL

Walter Kintsch

University of Colorado

A general theory of discourse comprehension (van Dijk & Kintsch, 1983) is used to develop a model for how people understand and solve word arithmetic problems, incorporating a problem solving model proposed by Riley, Greeno, & Heller (1982). This work is being done in collaboration with James Greeno, University of Pittsburgh.

Understanding mathematical word problems is a very special kind of understanding. Specialized strategies are involved (which have to be taught specifically in school), and the textbase that is constructed is a peculiar one, but its peculiarity lies in its specialized content - the same kind of structures are being generated as in any other situation, people reading a newspaper, a story, or a textbook. Thus, we are not proposing to build a specialized comprehension front-end for a word-arithmetic problem-solving model; instead, we are going to apply a general model of text comprehension to this special situation. Specifically, what the theory assumes is that the verbal input is decoded into a list of atomic propositions which are organized into larger units on the basis of some knowledge structure to form a coherent textbase. From this textbase, a macrostructure is constructed which represents the most essential information in the textbase. In parallel with this hierarchical text representation we also construct a situational model, which in this case is the problem representation which Riley, Greeno & Heller (1982) used as the starting point for their model, and upon which various arithmetic operations can be performed.

A set of problems can be constructed that form prototypes for all single-step addition and subtraction problems. By suitably restricting the language of these problems, only 9 propositional frames must be used in these problems, which makes the task of deriving atomic propositions from the problem sentences easy, indeed trivial. Each proposition is associated with a meaning postulate, and it is at this point where arithmetic-specific aspects enter into our analysis, because the meaning postulates used here are quite special ones. They are impoverished compared with everyday language use, and they are specialized. All we care about in these problems concerns sets of objects (always marbles here), their specification (always in terms of ownership), their numbers, and the relationships among the various sets. Thus, the only kind of information that is relevant here is that specified by the slots of the set schema. The textbase is always formed from this set schema. This is very different from other types of texts, where the textbase may be based on many diverse knowledge structures, requiring a richer interpretation. If we read "Joe gave 5 marbles to Tom" in an arithmetic problem, all we want to know is that there is a set of marbles now owned by Tom and formerly owned by Joe, which we call a transfer set and which is part of a Transfer schema, together with a startset and a resultset. In a story, on the other hand, we might be concerned with Joe's motive, or with Tom's reaction, or we would prefer dollars to marbles - all of which would be out of place here.

Thus, sentences are decoded into atomic propositions and these are assigned to the slots of a set schema. The main purpose of the model is to show exactly how this happens. The basic assumption is that the process is strategic and that the strategies involved are not the "normal" comprehension strategies, but specialized strategies for dealing with word arithmetic problems. Formally, strategies are modelled as productions. That is, we specify certain conditions in the text which, if they occur, lead to certain actions. The actions always consist in constructing a set and assigning the text propositions to its slot. We need five such schematic strategies or productions to account for the problems considered here. In each case, we find on the condition side a quantifier proposition of the form N(y) where N is a number or SOME, and y are some marbles, plus either a HAVE,GIVE, or MORE/LESS-THAN proposition which provides information on the role of the set to be created. For example, N(y) in the context of a GIVE proposition creates a transfer set of N marbles, owned by the patient of the GIVE proposition. MORE (or LESS) propositions are the condition for creating a remainderset. Simple HAVE propositions, on the other hand, provide no information about the role of the set. In this case, the role slot remains empty, until there is other contextual information that permits filling it. In the Change problems, sentence order serves that function, as well as such explicit propositions as NOW, THEN, or PAST which further specify temporal order and hence the role of sets in the transfer schema. In other problems, there may be no linguistic indicators of set role at all, and hence, the role slot is not filled in the text base. However, the role slot must always be inferred in the problem representation which is constructed in parallel with the text base. Thus, the problem representation is in part a copy of the text base, except for those schema slots which the text base does not specify and which must be inferred from the knowledge of some higher order schema. For instance, consider the combine problem: Joe has 3 marbles. Tom has 5 marbles. How many marbles do they have altogether? The first two sentences provide the conditions for creating two sets of marbles, owned by Joe and Tom, with a certain number in each, but with unknown roles. The third sentence triggers a MAKE-SUPERSET because of the HOWMANY(MARBLES) in the context of HAVE(JOE & TOM,MARBLES). Having a superset, we need subsets, and we infer that the two sets created previously are the subsets in question: their specification in terms of ownership permits us to make this inference, but it does not force us to do so logically - the inference is only justified within the pragmatic conditions of word problems. The inferred roles are specified in the problem representation, but no corresponding proposition "S1 is a subset" is inferred in the textbase. Thus, the problem representation may contain more information than was explicit in the textbase. The reverse may also be true: if we had included in our problem the irrelevant information that Tom had blue marbles, the proposition BLUE(MARBLES) would have been assigned to the specification slot of the corresponding set in the textbase, but would not affect the problem representation.

Once the problem representation is completed, the arithmetic operations themselves are performed, the different set constellations serving as the conditions for appropriate operations. Thus, for instance, a transfer-in schema with a calculation goal on the result set is the condition for a count-on operation in young children, or addition in the older.

How all this works to produce the right solution to a word problem is best illustrated by a few examples. However, because of space limitations, we can only describe here a particularly simple example, the Change 1 problem: Joe has 3 marbles. Tom gives him 5 more marbles. How many marbles does Joe have now?

The first sentence is "Joe has three marbles". It is parsed into the propositions P1-P4 as described in Kintsch (1982). P3 and P4 turn out to be the condition for a MAKE-SET operation, creating S1: the four propositions are assigned to the appropriate slots of the set schema. At the same time, a parallel set is established in the problem representation, with entries derived from the textbase (indicated by arrows). Note that at this point there are no entries in the Role slot of S1, neither at the textbase level nor at the level of the problem representation.

The second sentence is similarly organized into S2 via a MAKE-TRANSFERSET operation, while S1 is held in short-term memory. Since S2 is a transferset, requests are created for the corresponding start- and resultsets. S1 is identified as the desired startset on the basis of an explicit linguistic cue, the "then" of the second sentence. Short-term memory now contains a partially completed transfer schema consisting of S1 and S2, and a request for the missing resultset.

The third sentence provides this resultset and completes the schema, which then triggers the arithmetic operation count-on (or add), as in the Riley model.

All the other problems can be treated similarly, using the propositional schemata for the construction of the proposition lists, and the schematic strategies to organize them into TRANSFER, SUPERSET, or MORE-THAN schemata. However, the process does not always run off as smoothly as for Change 1 problems: sometimes, inferences need to be made to specify a slot of a schema for which the text provides no explicit cues (an example was mentioned above for Combine problems) and sometimes sets no longer available in the limited-capacity short-term memory buffer must be reinstated from long-term memory (or by rereading a sentence) to complete a problem.

Thus, our model leads us to distinguish three separate sources of problem difficulty. In order to do these problems, you need first of all knowledge about the right arithmetic schemata and operations - the set schema, the transfer, superset, and more-than schemata, as well as the actual counting and arithmetic operations; in this respect, our model is no different than the Riley et al. model in its implications. But you also need to be able to use these knowledge schemata in word problems, i.e., you need to have this knowledge in the form of productions adapted to the textual input. Finally, even if you have all the right knowledge, the way a problem is stated may make it easy or hard because some problem versions make only minimal demands on short-term memory (e.g., Change 1 or Combine 1) while others can only be solved if large amounts of sometimes incoherent material can be remembered.

Preliminary analyses have shown that the need to make inferences, and especially the short-term memory load (the size of the units to be carried in STM, the number of requests that must be kept track of, and number of propositions that can not immediately be attached to some set-unit) are factors which greatly contribute to problem difficulty, over and beyond the knowledge structures and schematic productions needed to solve the various problem types.

References

van Dijk, T.A. & Kintsch, W. Strategies of comprehension. New York: Academic Press, 1983.

Greeno, J.G., Riley, M.S., & Heller, J.I. Development of children's problem-solving ability in arithmetic. In H.P. Ginsburg (Ed.), The Development of Mathematical Thinking. New York: Academic Press, 1983.

Kintsch, W. Aspects of text comprehension. In J.F. LeNy & W. Kintsch (Eds) Language and comprehension. Amsterdam: North-Holland, 1982.