STRATEGIES FOR SOLVING WORD ARITHMETIC PROBLEMS:
EXTENSION OF THE KINTSCH & GREENO MODEL

Walter Kintsch

University of Colorado

Technical Report No. 133

Institute of Cognitive Science
University of Colorado
Boulder, Colorado 80309

Working Paper - Do Not Quote

November 6, 1984

STRATEGIES FOR SOLVING WORD ARITHMETIC PROBLEMS: EXTENSION OF THE KINTSCH & GREENO MODEL.

by

WALTER KINTSCH

A theory of how children solve simple word arithmetic problems has been proposed by Kintsch and Greeno (1985). This theory represents an amalgamation of previously different research areas: the rich work on problem solving in mathematics has been combined with results from studies on text comprehension. The theory assumes that understanding word problems involves constructing a propositional textbase and a parallel problem representation, as described by van Dijk & Kintsch (1983). Textbase construction is a strategic process, and word problems require the use of some special comprehension strategies, which ensure that the text will be organized around mathematical concepts, such as set, rather than around the actors' motivations and goals, as would be appropriate for a narrative. Thus, a situation model - here called the problem model - is constructed from the text which highlights the important arithmetic relations in the problem. The formal problem solving methods described by Riley, Greeno, & Heller (1983) can then operate upon this structure and produce the desired solution. The rather complex details of this process are described in the original publication. A computer simulation of these processes has been produced by Fletcher (1984).

The Kintsch & Greeno model is restricted to a problem set which can be formed by paraphrasing the 14 prototype problems given in Riley et al. These include Change problems (give-take), Combine problems (super-subsets), and Compare problems (more-or-less-than). All of these problems are degenerate word problems, in that they only contain information that is directly problem relevant. An extension of the theory is presented here which deals with all word arithmetic problems that can be solved by addition or subtraction in one step, and where each whole is divided into no more than 2 parts. The second of these limitations is a trivial one: it would be quite straightforward to admit more than two subsets, although the technical details would be cumbersome. The restriction to single-step problems, on the other hand, is substantive: a much more complex problem solving component than the one envisaged here would be required for dealing with two-step problems.

The sample problems which we are specifically concerned here include the 11 well-formed Nesher problems (from Nesher & Katriel, 1977), and 12 arbitrarily selected problems from Bereiter's Thinking Stories (Willoughby, et al., 1981). Both sets of problems are shown in the Appendix.

What is proposed here is not a new theory, but merely en extension of Kintsch & Greeno, requiring only minor modifications of their theory. The changes in the theory which are necessary for this extension have to do with the relationship between the textbase and the problem model, and with the introduction of new strategies to deal with a wider range of arithmetic problems.

In the original model, the textbase and the problem model are closely tied together: if it were not that the vanDijk & Kintsch theory requires that distinction, Kintsch & Greeno could have just as well done without it. This is no longer the case, if we consider word arithmetic problems such as those shown in the Appendix: textbase and problem model become separate, only partially overlapping structures. The textbase is to be formed through the application of general comprehension strategies as described in vanDijk & Kintsch; from it, a problem representation is constructed, using the arithmetic strategies identified in Kintsch & Greeno, plus some additional ones introduced below.

The Kintsch & Greeno model includes the following four arithmetic strategies: MAKESET, MAKE-TRANSFERSET, MAKE-DIFFERENCESET, and MAKE-SUPERSET. To these, we add MAKE-SUBSET, a strategy which works very much like the previous ones, plus three strategies of a somewhat more complex nature, the ZERO-STRATEGY, CONJUNCTION-STRATEGY, and a set of POSSESSION-STRATEGIES.

However, even with the addition of these strategies, the model is not powerful enough, for many word problems do not fulfill the conditions of these strategies. What is needed is another set of problem solving strategies, which we shall call elaboration strategies. These strategies are used to elaborate the problem model in such a way as to create the conditions for the application of the arithmetic strategy, and hence, for the solution of the problem. Such elaboration requires knowledge, of course, and appropriate knowledge, lexical as well as world knowledge, must be available.

1. ARITHMETIC STRATEGIES:

1.1. MAKESET is the basic strategy; originally, it built a set whenever it encountered a proposition of the form QUANTITY(OBJECT), assigning the OBJECT to the Object slot of the set, QUANTITY to the Quantity slot, and the other propositions in the same processing cycle to the Specification slot, except for those propositions which appear on the condition side of one or the other arithmetic strategies, which are assigned to the Role slot.

For the Nesher problems, this works fine and makes sense: like our original problems, they contain no (or hardly any) irrelevant information, it is usually clear what the objects are for which sets are to be formed (except for N4, where there is a conflict); certainly the adult reader will organize the whole problem text in terms of sets right away, on first reading.

This is clearly not so in the case of the Thinking Stories: here the arithmetic problem is embedded in a richer context, some of which is irrelevant, but some of which provides crucial information for the problem. It would appear that in these problems first a general, default textbase is formed, yielding a general understanding of the situation. Only when the question is encountered at the end of the problem is MAKESET employed. This must be done in a somewhat different way:

(a) MS are applied in the reverse order, so that the first set formed contains the question, HOWMANY(X).

(b) Once S1 has been formed, further MS are constrained to the same object.

(c) Not all propositions are assigned to the Specification slot, but only those that contain X (the contents of the object slot) as an argument, or propositions embedding already selected propositions. Thus, some propositions in the textbase have no counterpart in the problem model.

Example: TS7

Manolita tried to change her father's garden by pulling out the weeds."You changed it all right" said Mr. Mudanza. "There were 14 tulips, and now there are only 6." How many tulips did Manolita pull out by mistake?

The question constrains the object of further MS to be TULIP. Thus, S2 will have TULIP as Object, 6 as Quantity, and HAVE(GARDEN,TULIP)^NOW(P) as specification; SAY(MR.M,P) and CHANGE(MANOLITA, GARDEN) are omitted.

1.2. TRANSFERSET remains unchanged: Its condition is

Si has Spec.: GIVE(x1,x2,Q(y)) &

Sj has Spec.: HAVE(xi,Q(y))^before Si

TS assigns the role of transfer-in or transfer-out to Si, depending on the identity of xi, the role of startset to Sj, and sets up a request for a resultset.

1.3. DIFFERENCESET is slightly generalized, to deal not only with the predicate HAVE, but with any arbitrary P that is used in conjunction with more-(or less)-than. Its condition becomes

Si has Spec.: P-MORE(LESS)THAN(x1, x2, Q(y))

Its action is to assign the role of Difference to Si, and to request a LARGESET with specification P(xi) and a SMALLSET with specification P(xj); i and j are reversed for LESS.

1.4. SUPERSET is similarly generalized, its new condition becoming

Si has Spec.: P-ALLTOGETHER(xi & xj, Q(y))

with the action of assigning superset role to Si, and requesting subsets with specifications P(xi) and P(xj).

1.5. SUBSET has the condition

Si has Obj.: x and Role: OF(x,y) &

Sj has Obj.: y

and assigns subset role to Si and superset role to Sj.

2. We now turn to some new arithmetic strategies. Since we are always dealing with simple set relationships, these could be stated in a general, abstract way (indeed, subsuming the previous strategies). However, it is useful to keep various special cases apart for computational purposes, and it is probably also more realistic psychologically, in that children in grades 1-3 - who are the subjects being modelled here - do not appear to be working with a single, abstract superset rule, but with a variety of more concrete strategies.

The strategies discussed here differ from those above in that their conditions always consist of patterns of specifications across all three sets that are to be assigned roles. Thus, these strategies go beyond simple key-word strategies (if you find a GIVE, it's a "transferset"); in order to determine the respective roles of the sets that have been created, the whole pattern of specifications must be considered.

In order to describe these strategies more compactly, a number of notational conventions will be observed here: Conditions always refer to specifications, and Actions to role, unless otherwise noted. It is assumed that the objects of the three sets in question are the same. A, B, and C may stand for either propositions or arguments in the same case slot ($ indicates an unspecified case). All strategies require a disjunction check which is not specifically indicated below. That is, if A and B are the specifications of two sets which have been identified as subsets, "If A, then ~B" must be verified before this role assignment can be accepted.

2.1. ZERO-STRATEGY for propositions:

Condition:	Action:
Si: A^C	sub1
Sj: B^C	sub2
Sk: C	super

Example: N2

There were 4 big windows and 3 small windows in the hall. How many windows were there in the hall?

Si: BIG(WINDOW)^LOC(WINDOW,HALL)	sub1
Sj: SMALL(WIND)^LOC(WINDOW,HALL)	sub2
Sk: LOC(WINDOW,HALL)	super

2.2. ZERO-STRATEGY for arguments:

Condition:	Action:
S1: A	sub1
S2: B	sub2
S3: $	super

Example: N3

Mrs. Eshkoli invited 4 visitors to her party. Two other visitors were invited by her husband. How many visitors were invited to the party?

S1:INVITE(MRS.ESH,VISITOR,PARTY)	sub1
S2:INVITE(HUSBAND,VISITOR,PARTY)	sub2
S3:INVITE($ VISITOR,PARTY)	super

2.3. CONJUNCTION-STRATEGY:

(A and b may be either propositions or arguments in the same case slot)

Condition:	Action:
Si: A	sub1
Sj: B	sub2
Sk: (A&B)	super

Example: N6

Mark lost 3 coins yesterday and 2 coins today. How many coins did he lose yesterday and today?

S1:LOSE(MARK,COIN)^YESTERDAY(P)	sub1
S2:LOSE(MARK,COIN)^TODAY(P)	sub2
S3:LOSE(MARK,COIN)^YESTERDAY&TODAY(P)	super

2.4. OBJECT-CONJUNCTION-STRATEGY:

In this case, A and B are objects rather than specifications:

Condition:	Action:
Si: A	sub1
Sj: B	sub2
Sk: (A&B)	super

Example:N1

There are 2 dolls and 3 teddybears on the shelf. How many dolls and teddybears are there on the shelf?

S1: doll	sub1
S2: teddybear	sub2
S3: doll&teddybear	super

2.5. POSSESSION and LOCATION STRATEGIES:

The specification slots of the three sets contain either HAVE(AGENT,OBJECT) or LOC(OBJECT,PLACE) propositions or their negations, and are ordered temporally:

Si:+/-HAVE(LOC)^TIME1

Sj:+/-HAVE(LOC)^TIME2

Sk:+/-HAVE(LOC)^TIME3

The following patterns of assertions and negations occur, requiring actions as indicated:

	S1	S2	S3	Assign superset role to
1	H	H	H	S3
2	H	H	~H	S1
3	H	~H	H	S1
4	~H	H	H	S2
5	H	~H	~H	S1
6	~H	H	~H	S1
7	~H	~H	H	-
8	~H	~H	~H	S3

Example:

There were 5 books on the shelf yesterday and today two are no longer there. How many books are on the shelf now?

S1: LOC(BOOK,SHELF)^YESTERDAY(P)	super
S2:~LOC(BOOK,SHELF)^TODAY(P)	sub1

S3: LOC(BOOK,SHELF)^NOW(P)   sub2

Location-Strategy 3 was applied in this case.

3. ELABORATION STRATEGIES

If no condition for an arithmetic strategy is met, elaboration strategies can be used to elaborate the set structures until one of the arithmetic strategies becomes applicable.

3.1. OBJECT-ELABORATION: Are the objects of the three sets identical? If not, is there information in knowledge about superset relations that would create conditions satisfying Strategy 2.4?

Example:N1

There were 2 dolls and 3 teddybears on the shelf.

How many toys were on the shelf?

Obj: S1:doll

S2:teddybear

S3:toys

Knowledge contains the propositions ISA(DOLL,TOY) and ISA(TEDDYBEAR,TOY), hence we can substitute dolls&teddybears for toys above, and apply <2.4>.

3.2. SPECIFICATION-ELABORATION: Do any of the propositions in the specification slots have implications (check knowledge); if so, add to specification slots.

Example:TS7

Manolita tried to change her father's garden by pulling out the weeds."You changed it all right" said Mr. Mudanza. "There were 14 tulips, and now there are only 6." How many tulips did Manolita pull out from the garden by mistake?

After the MAKESET operations (see the worked example below), the specification slots of the three sets contain

S1: PULLOUT(MANO, TULIP, GARDEN)^BY-MISTAKE(P)^PAST(P)

S2: HAVE(GARDEN, TULIP)^PAST(P)

S3: HAVE GARDEN, TULIP)^PRESENT(P)

No arithmetic strategy applies; check of implications retrieves from knowledge IMPLY(PULLOUT(agent,object,source),~HAVE(source,object)). This is used to elaborate the specification of S1. To apply a POSSESSION (or LOCATION) rule, we need a time ordering: PRESENT(P) identifies S3 as T3; the PAST in S1 is derived from P17, and the PAST in S2 is derived from P10, hence S2 is earlier than S1. Thus, the specification slots of the three sets can be rewritten as (omitting now irrelevant information):

S1: ~HAVE(GARDEN, TULIP)^T2

S2: HAVE(GARDEN, TULIP)^T1

S3: HAVE(GARDEN, TULIP)^T3

and Possession Strategy 3 identifies S2 as superset, with S1 and S3 as disjoint subsets.

3.3. ADDITION STRATEGY:

If <3.1> and <3.2> do not apply, add further text propositions to the specification slot: find all propositions that have argument overlap with propositions already in the specification slot and add them to the specification slot. Then try <3.2.> again.

Example:TS5

Mrs. Nosho was telling Mark about the two huge aquariums she kept when she was a little girl. "There were 30 fish in one asnd 40 fish in the other, so you can tell how many fish I had." How many fish did Mrs. Nosho have?

P1 TELL(MRS.N, MARK, P2-8)

P2 2(AQU)

P3 HUGE(AQU)

P4 KEEP(MRS.N, AQU)

P5 WHEN(P4,P6)

P6 PAST(P7)

P7 ISA(MRS.N, GIRL)

P8 LITTLE(GIRL)

P9 40(FISH)

P10 LOC(P9,P11)

P11 ONE(AQU)

P12 30(FISH)

P13 LOC(P12,P14)

P14 OTHER(AQU)

P15 ABLE(MARK, P16)

P16 TELL(MARK, P18)

P17 HOWMANY(FISH)

P18 HAVE(MRS.N, P17)

P19 SAY(MRS,N,P9-P18)

P20 HOWMANY(FISH)

P21 HAVE(MRS.N, P20)

We first build a conventional textbase. As default strategy, vanDijk & Kintsch use the frame of the main verb in the sentence as a basis for the organization of the text base; however, in this case the main verb is "tell", which belongs to a class of verbs to be disregarded in favor of their complement, according to another general comprehension strategy described by vanDijk & Kintsch. Thus, "keep" provides the initial schema for the textbase; "keep" takes as arguments an agent and an object, both of which may be modified and quantified. The "keep" schema, and the assignment of the propositions from the first sentence to the slots of that schema are as follows (the symbol * indicates that agent and object are derived from the proposition at the head of the schema, in this case P4, KEEP(MRS.N, AQU)):

```
KEEP     4
 agt:    *
  mod:   5-6-7-8
  qu:
 obj:    *
  mod:   1-3
  qu:    2
```

KEEP has the agent MRS.N which is modified by "when she was a little girl", and the object AQU, which is modified by P1 and P3, and P2 in the quantity slot.

The second sentence starts a new processing cycle. As a general default strategy, we try to maintain the existing schema as long as that is possible, that is, as long as the input propositions can be assigned to the slots of that schema. In this case, we simply add the second sentence propositions to the "keep" schema in the modifier slots for MRS.N (who said something) and AQU (part of what was said concerned it):

```
KEEP     4
 agt:    *-------------*
```

```
 mod:   5-6-7-8         19-15-16-17-18

 qu:

obj:    *-------------*

 mod:   1-3             10-9-11

                        13-12-14

 qu:    2
```

The lines indicate here that we are still dealing with the same agent and object.

The third sentence brings us to the question, which triggers a word-problem specific strategy. Instead of continuing the default textbase, use the main verb of the question to establish a new schema. Nevertheless, there is some continuity with the previous schema, since both have a common agent, Mrs.N.; the object of the "have" schema is, however, different from that of the "keep" schema:

```
KEEP    4                               HAVE 21

 agt:   *-------------*-----------------*

  mod:  5-6-7-8         19-15-16-17-18

  qu:

 obj:   *-------------*                  *

  mod:  1-3             10-9-11          22

                        13-12-14

  qu:   2
```

A MAKESET is executed on the basis of P22, which results in the construction of the following portion of the problem model:

```
S1 ob:       FISH

   qu:22     ?

   sp:21     HAVE(MRS.N,FISH)
```

```
    rl:
```

Having once decided on FISH as the object of the question set, further MAKESET's are constrained to this object. P10, P13 and P17 are, therefore, the conditions for further MAKESET operations. However, the set constructed from P17 is matched into S1, so that only two new sets are generated, yielding the following problem model:

```
S2 ob:        FISH
   qu:10       40
   sp:9-11    LOC(FISH,AQU)^ONE(AQU)
   rl:
S3 ob:        FISH
   qu:13       30
   sp:12-14   LOC(FISH,AQU)^OTHER(AQU)
   rl:
```

There is no arithmetic strategy that could be applied to this problem model. Furthermore, there are no implications known about the propositions in the specification slots of the three sets that could be used to elaborate them. Hence, <3.3> is tried: elaborate the specifications by adding propositions from the text that share an argument with any of the propositions already selected. This brings in considerable garbage (a lot of MRS.N-propositions in S1), but also introduces into both S2 and S3 KEEP(MRS.N, AQU). We now try the arithmetic strategies again, still without success. So we use <3.2> again, searching knowledge for implications of propositions in the specification slots, and come up with

IMPLY(KEEP(agent,object),HAVE(agent, object)).

Furthermore, we find

IMPLY (HAVE(agent, obj1)^LOC(obj2, in obj1), HAVE(agent,obj2)).

This leaves the specification slots looking like (where && stands for the now irrelevant propositions)

S1: 21-&& HAVE(MRS.N,FISH)

S2: 4-9-11-&& HAVE(MRS.N,FISH)^LOC(FISH,AQU)^ONE(AQU)

S3: 4-12-14-&& HAVE(MRS.N,FISH)^LOC(FISH,AQU)^OTH(AQU)

so that strategy <2.1> can be used to assign superset role to S1, after verifying via ONE(AQU) and OTHER(AQU) that the two subsets are mutually exclusive.

4. MEASUREMENT PROBLEMS

Many word arithmetic problems are measurement problems, for which the set frame is inappropriate. However, the current theory can be readily extended to deal with these problems, although we may eventually run into difficulties with the elaboration strategies which may have to be more complex, involving some kind of spatial imagery representation, than we have so far envisaged.

First, we must assume that in deriving the propositional representation of the problem, the reader always infers the measurement dimension, whether or not it is explicitly stated in the problem. Thus, if we read that someone "used 2 kg of flower", we assume that the appropriate propositional representation is USE($,FLOWER)^WEIGHT(FLOWER,2(KG)); if, in the same problem, the question is "How much did the flower weigh?", we similarly infer that the weight is to be measured in kg: WEIGHT(FLOWER,?(KG)).

Secondly, we introduce a MEASUREMENT-FRAME, with slots analogous to the SET-FRAME:

M:measure: DIMENSION(P,QUANT)
quantity: QUANT(MEASURE-SCALE)
specification: P
role: WHOLE, PART1, PART2

The conditions for a MAKE-MEASURE would be the presence of the dimension and measurement propositions during a procesing cycle.

Thus, for the four measurement problems in the Appendix, we obtain the following propositional representations, omitting the problem irrelevant portions of the textbase, and listing the relevant propositions always in the order General Prop - P, Measurement Prop - M(P,Q), Quantity Prop - QU(SCALE), and Other:

PRBL. NO.		P	M	Q	OTHER
1	M1	GO(MR.N)	DIST(P,Q)	5Ø(KM)	STOP(MR.N)
	M2	GO(MR.N)	DIST(P,Q)	75(KM)	AFTER(P,P)
	M3	GO(MR.N)	DIST(P,Q)	?(KM)	
8	M1	DO(SH,ERRAND)	DURATION(P,Q)	1Ø(MIN)	FIRST(ERRD)
	M2	DO(SH,ERRAND)	DURATION(P,Q)	35(MIN)	SECOND(ERR)
	M3	DO(SH,ERRAND)	DURATION(P,Q)	?(MIN)	BOTH(ERRDS)
1Ø	M1	FROM(HOUSE,FENCE)	DIST(P,Q)	2Ø(M)	
	M2	FROM(FENCE,END(YARD))	DIST(P,Q)	5(M)	
	M3	FROM(HOUSE,END(YARD))	DIST(P,Q)	?(M)	

12	M1	HAVE(LO,MAILBAG)	WEIGHT(P,Q)	14(KG)	TIME(MORNING)
	M2	DELIVER(LO,M)	WEIGHT(Q)	8(KG)	TIME(BY NOON)
	M3	HAVE(LO,MAILBAG)	WEIGHT(P,Q)	?(KG)	TIME(BY NOON)

Problems 1 and 8 are directly solvable by the zero- and conjunction strategy, respectively, after appropriate modifications. Problem 1Ø requires elaboration: the problem solver must know the transitivity rule for distances, so that FROM(HOUSE,END(YARD)) can be decomposed into FROM(HOUSE,FENCE) & FROM(HOUSE,END(YARD)), which satisfies the conditions for the Conjunction Strategy <2.2>. For Problem 12 the following meaning postulates must be supplied: DELIVER(agent,object) -> ~HAVE(agent,object), and HAVE(agent,object)^WEIGH(object,Q)=M -> HAVE(agent,M); Problem 12 can then by solved by Possession Strategy 3.

5. DISCUSSION

It appears that these rather modest additions to the arithmetic strategies of Kintsch & Greeno will be sufficient to handle the word problems in the domain considered here. Perhaps, however, these extensions are not as minor as they first appear: previously, the arithmetic strategies were cued by key words, or rather, key propositions; now the condition for these strategies in every case is a pattern involving all the sets which have been generated during the comprehension phase.

The real bottleneck in this kind of work promises to be the elaboration component. What we have now is sufficient for the set problems at hand, and perhaps, sufficient for the measurement problems. But it is clear that problems could arise which would require more powerful elaboration strategies - problem solving routines - than the ones considered here. Certain problems, as we have already seen with the present measurement problems, seem to demand a non-propositional, spatial-imagery representation. The most suitable problem model is sometimes not a (purely) propositional schema, but a graph. It is not clear how propositional and non-propositional representations are to be integrated in a workable manner.

Nevertheless, what we have here might at least provide a starting point for further work.

REFERENCES

Kintsch, W. & Greeno, J. G. Understanding and solving word arithmetic problems. Psychological Review, in press.

Nesher, P. & Katriel, T.A. A semantic analysis of addition and subtraction word problems in arithmetic. Educational Studies in Mathematics, 1977, 8, 251-269.

vanDijk, T.A. & Kintsch, W. Strategies of Discourse Comprehension. New York: Academic Press, 1983.

Willoughby, S. S., Bereiter, C., Hilton, P. & Rubinstein, J. H. Real Math: Thinking Stories Book. Level 2. La Salle, Ill.: Open Court, 1981.

Appendix

THINKING STORIES PROBLEMS

1 <p36-1> Mr. Nosho was telling Mark about a trip he planned to take. "I haven't taken a trip in years and I'm going to take my time. I'll go 50 km then stop; then I'll go another 75 km." How far does Mr. Nosho plan to travel?

2 <p36-2> Mr. Sleeby knew that he often forgot things. One day he decided to keep track of how many things he forgot during the day. At the end of the day he said, "Let's see, today I forgot to go to the bank, and there are two other tings whic I forgot, but I can't remember what they are." How many things did Mr. Sleeby forget that day?

3 <p37-7> Mr. Nosho was clearing out his garage. "If I throw away all of these old magazines, I wonder how many I'll have left." How many will he have left?

4 <p37-8> One day Mr. Nosho was talking with Mr. Sleeby. "If I buy some bushes for my frontyard now and I buy 13 more in the fall, how many bushes will I have bought? Mr. Sleeby thought for a minute. "I thought I could answer that question a minute ago, but I think I have forgotten the answer." How many bushes did Mr. Nosho buy?

5 <p42-3> Mrs. Nosho was telling Mark about the two huge aquariums she kept when she was a little girl. "Thee were 30 fish in one and 40 fish in the other, so you can tell how many fish I had." How many fish did Mrs. Nosho have?

6 <p42-4> Mr. Sleeby was planting pea plants. He planted one row with 9 plants in it. He started the next row, but he stopped after he put in 4 plants. "I've forgotten how many pea plants I have planted,"

he said. How many plants did Mr. Sleeby plant?

7 <p42-5> Manolita tried to change her father's garden by pulling out the weeds. "You changed it all right" said Mr. Mudanza. "There were 14 tulips, and now there are only 6." How many tulips did Manolita pull out by mistake?

<p43-9> Sharon did her first errand of the day in 10 minutes, and her second errand in 35 minutes. Mr. Breezy told her that it would take an hour to do the errands. How much did it take her?

9 <p48-2> Mark was worn out from walking 10 dogs. "You won't have so many dogs to walk tomorrow," said his father. "Three of them are going home." How many dogs will Mark have to walk tomorrow?

10 <p48-3> Mr. Breezy asked fred to set up a fence between the house nad the end of the yard. "It should be 20m from the house. That's 5m from the end of the yard." How far is it from the house to the end of the yard?

11 <p49-6> "I understand that you have 8 sisters," said Mr. Breezy. "No, I don't." said Sharon. "But my brother does." How many sisters does Sharon have?

12 <p67-9> When Loretta started out in the morning, her mailbag weighed 14 kilograms. By noon she had delivered 8 kilograms. How much did her mailbag weigh by noon?

NESHER PROBLEMS:

1. There were 2 dolls and 3 teddy bears on the shelf.

How many toys were there on the shelf?

2. There were 4 big windows and 3 small windows in the hall.

How many windows were there in the hall?

3. Mrs. Eshkoli invited 4 visitors to her party.

Two other visitors were invited by her husband.

How many visitors were invited to the party?

4. Five boys bought 3 tickets, and the girls bought 4 tickets.

How many tickets did the boys and girls buy?

6. Mark lost 3 coins yesterday and two coins today.

How many coins did he lose on both these days?

7. Dan took 3 books from the upper shelf and 4 books from the lower shelf.

How many books did Dan take off the shelves altogether?

9. Dave found 3 stamps and bought 3 stamps.

How many stamps does Dave have?

11. Dan gave 3 candies to Joe.

Joe got 3 candies from Moses.

How many candies does Joe have?

14. Mark tore up 4 stamps from his collection and lost 2 stamps.

How many stamps are missing from his collection?

15. Mark lost 5 pounds yesterday and found 5 pounds today.

How many pounds is he still missing?

16. Thirty cars entered the parking structure and 20 cars left it.

How many cars passed through the gate?