Scientific DataLink

Report 83-11
Stanford -- KSL

An Attempt to Understand Pupils' Understanding of Basic Algebra.

Derek H. Sleeman,

Feb 1983

BLANK PAGE

# 1. INTRODUCTION

The impetus for work in Intelligent CAI has two major sources: firstly, the practical aim of producing teaching systems which are truly adaptive to the needs of the student and secondly the "theoretical" interest involved in formulating these activities as algorithms. It has been argued by Hartley and Sleeman, [Har 73], that an intelligent teaching system requires access to the following information: Knowledge of the task domain; Student model; List of teaching operations; Means-ends guidance rules which relate teaching decisions with conditions in the student model.

A number of systems have been implemented during the last decade which include some or all of these databases. In particularly, during the last 5 years a number of systems have been implemented which attempt to provide supportive learning environments intended to facilitate learning-by-doing. These systems, which include SOPHIE, [Bro 82], GUIDON, [Cla 82], WEST [Bur 82a], WUMPUS [Gol 82], and PSM-NMR [Sle 82c], have been called Coaches or Problem Solving Monitors. In this paper, we address a particular aspect of the problem of inferring a model from the pupil's behaviour on a set of tasks, [1]. We shall report, in outline, the results of a recent experiment with 24 14 year-old pupils, who were considered to be of average ability. The issue to be considered in this paper is whether the models inferred by the Leeds Modelling System, LMS, can be given a cognitive interpretation, that is whether it is possible to say something about the processes used by the pupil given the models inferred by LMS.

In common with BUGGY [Bro 78], LMS uses a generative mechanism to create hypotheses/models from primitives. Without a generative facility, the ability of a system to model complex, and errorful, behaviour is severely limited. However, the use of such a mechanism also causes difficulties as such an algorithm can readily lead to a combinatorial explosion, where given N primitives, N! models are produced. BUGGY uses a collection of primitive bugs from which to generate models; LMS uses domain rules and corresponding mal-rules (incorrect) rules, which have been observed in the analysis of earlier protocols. On the other hand, whereas BUGGY uses a series of heuristics to limit the size of its search space, a major feature of the LMS work has been the formulation of

---

1. For a more detailed discussion of this, and related issues please see the Introductory essay to Intelligent Tutoring Systems, [Sle 82b].

the search so as to focus on particular rule(s). As has been demonstrated, [Sle 81], [Sle ap1], this technique drastically reduces the number of models that must be considered at each stage, [2]. Before considering the results of the experiment, we briefly review the Production System, PS, representation which has been used for pupil models and explain the main features of the PS interpreter used to "execute" these models.

Figure 1a gives a set of Production Rules which are sufficient to solve linear algebraic equations. Figure 1b gives a set of mal-rules for this domain which have been observed in protocols analysed earlier and Figure 1c shows pairs of correct and "buggy" models executing typical tasks.

[Figure 1 about here]

In this work, a model is thus an ordered list of rules. Order is significant, as the interpreter used executes the action of the first rule in the model whose conditions are satisfied by the state, i.e., the task or the partially solved task. In this way we are able to capture precedence which is important in this subject domain. The match-execute cycle continues until no further rules fire.

---

2. Initially, we made the assumption that the domain was hierarchical and so we have referred to the stages as levels; and thus modelling proceeds by first considering level 1, then 2, etc.

## 2. TYPICAL REPORTS/MODELS INFERRED BY LMS

Figure 2 gives 3 sets of models for a competent, a "patchy" and a poor pupil. The information given for each pupil for <u>each</u> task set is to be interpreted as follows: "Total" is the total number of tasks in the task-set, "Correct" the number correct, "Mal" is the number of times a model contained one of the original set of Mal-rules (given in Figure 1b), "Parsing" is the number of errors caused by the way the pupil represents the task - a new type of error encountered in this experiment, "New Mal" are the number of errors explained by additional (manipulative) Mal-rules, "Clerical" is the number of errors considered to be caused by clerical/simple arithmetic errors and "Wild" are errors which so far have defied explanations, [3]. The next piece of information provided gives LMS's overall assessment for the pupil's performance on this particular task set, [which may be null if the pupil is pretty random, otherwise it will be a Production system model], [4]. Figure 3 gives examples of <u>typical</u> tasks for each of the task sets.

[Figures 2 and 3 about here.]

The first pupil was remarkable consistent, getting 56 out of 63 correct. However, he had difficulties when the task became more difficult as he had a tendancy <u>not</u> to type in any of the intermediary steps, but only the final result. A new mal-rule was used twice at level 13; and at level 14, MSOLVE was activated twice.

The second pupil did well on task sets 2 to 6 and 9, and made <u>systematic</u> errors on task sets 7,8 and 10. LMS recognized the errors at levels 7 and 10; the errors at level 8 have been subsequently analysed as systematic by the Investigator.

The third pupil did fairly well on task-sets 2-4 (simply invoking MSOLVE twice), but behaved pretty randomly on the 5th set.

It is likely that these pupils would be "treated" as follows. Pupil a) would be advised to be more careful and take smaller steps, pupil b) would probably merely be shown how to work the task sets in which she made systematic errors. (But indeed, her working

---

3. Originally all those not in the "correct" and "Mal" classes were classified as "wild", and so some considerable progress has been made.

---

4. In practice the analysis provides a model for every task, as well as an "averaged" model for each task-set.

needs analysing more fully as it shows an interesting type of error - which will be discussed in more detail later). The third pupil needs to be (re)taught Algebra virtually from the beginning.

In the next section we give an overview of "new" mal-rules observed in this experiment, together with protocols from which they have been inferred.

## 3. THE 1981 EXPERIMENT WITH LMS

An earlier experiment had been run with a group of 15 year old pupils and a *very* close agreement had been achieved between LMS's diagnosis and those made by a group of investigators, [Sle 82a]. However, the design of LMS was such that if the pupil did not make an error with say XTOLHS when it was introduced, then LMS assumed that XTOLHS would be used successfully at ALL subsequent levels. The 1980 experiment showed that this was NOT a valid assumption. For example, some pupils who were able to work the following types of tasks correctly:

$$M*X = N*X + P$$

where M, N and P are integers, appear to forget to *change* the sign of the X-term when the side is changed, when they had to expand brackets. For example, given the task:

$$12*X = 2*<4*X+5>$$

we have seen the answer:

$$20*X = 10$$

It was, in fact, easy to remove this assumption from LMS's code, but unfortunately the modification led to an "explosion" in the number of models to be considered, and so a reformulation of the algorithm was carried out, [Sle ap1]. This experiment was carried out with the *revised* modeller, LMS-II, but with the same data-base of rules and tasks as used in the 1980 experiment. This group of 24 pupils, average age 14 years 3 months, were judged to be of average ability at Mathematics; however the results were *dramatically* different from the earlier group's, [5]. Indeed many of their difficulties were not

---

5. These pupils had been at this High School for nearly one school year when this experiment was carried out, but it should be noted that the intake of this particular school is from 30 or so Middle schools (age-range 9-13 years).

diagnosed by LMS-II and had to be analysed by the investigator. This analysis was made very difficult because, it had been assumed that pupils would, at most make one or two minor manipulative errors, (e.g. changing side and not sign), and so LMS had been designed so the pupil could merely input his final answer, and none of his intermediary steps. In figure 4 we give a sample of the protocols observed, together with the mal-rules which the investigator suggested were appropriate for each task-set. In figure 5, we summarise the complete set of new mal-rules which the investigator considered explained the pupils' behaviour with LMS.

Note that by stating that a protocol can be explained by a mal-rule, say, of the form

$$M*X \Rightarrow M+X$$

(Figure 4a), we do not wish to imply that given a problem of the type

$$3*X + 4*X = 5$$

that the pupil would produce the response:

$$3+X + 4+X = 5$$

Indeed, we have seen several pupils write

$$X+X = 5-3-4$$

and when asked to provide intermediary steps they have said categorically that there were none as the above was done in "one step". Nevertheless we are happy to accept that both forms are "explained" by the mal-rule; the first form however requires that several additional rules fire in order to get it into the state given by the "second" pupil. (It should be noted that the mal-rules given in Figures 4d and 4e and many of those given in Figure 9 are more "comprehensive" and carry out several "housekeeping" steps). Note that the difference between "basic" and "comprehensive" mal-rules is significant when one tries to perform remedial instruction, as it is important to ensure that the "grain" of the instruction matches the pupil's.

[Figures 4 and 5 about here]

Further as the result of analysing these protocols, a number of questions were raised, including:

-What is the crucial difference between the task-sets which the pupil is and is not able to correctly solve?

- Does the pupil's perception of Algebraic tasks vary from one task type to another? (See figure 3 for a list of task types given to the pupils and [Sle ap1] for a more detailed discussion of how this set was constructed).

Unfortunately, because this analysis took a while and the school vacation interceded, it was not possible to meet with the pupils again until September (1981). Because of the time that had elapsed, the pupils were given a paper-and-pencil test which covered comparable tasks to those set by LMS. These tests were analysed in detail by the investigator, and as a result of this certain pupils were given detailed diagnostic interviews. The next sections give more details of these stages.

## 3.1 The Paper-and-Pencil Test

From a comparative review of the May and September data, see [Sle ap3] for the details, we made the following observations:

1. The performance was generally considerably better in September than in May. (Note no additional teaching in Algebra had been given, however the pupils had presumably done some self-study in preparation for their end of year examinations).

2. A considerable number of tasks were not solved on the written test - (whereas LMS insisted on the pupil giving a response to each question).

3. Some pupils who appeared to have "wild" rules in May, seemed to solve this type of task correctly in September, eg AB5.

4. Some pupils whose behaviour had been "random" or "wild" in May had now settled to use mal-rules consistently, e.g. pupil AB18.

5. One pupil at least, AB7, gave multiple values in an equation where X occurred more than once.

6. Many of the pupils made the "common precedence" error, namely given a task of the form:

   $2 + 3*X = 11$ they return $5*X = 11$.

As a result of this comparison it was decided to interview all those who appeared, on the written test, still to have major difficulties and all those who had had major difficulties which appeared to have "cleared up". (But not those with the common "precedence" errors).

### 3.2 The Interviews

These proved to be remarkably revealing and very rewarding as the pupils without exception were extremely articulate. The dialogues were taped, and the figures are a reconstruction from the tapes and the worksheets used.

After analysing the pupils' protocols obtained with LMS we conjectured that some pupils may actually "see" some of these tasks differently from the standard algebraic interpretation. That is a pupil like AB18 might actually see a task of the type:

$$M{\bullet}X + N{\bullet}X = P \quad \text{as} \quad M + X + N + X = P$$

To test this hypothesis, we started each interview by asking the pupil merely to read a list of Algebra tasks (given in [Sle ap3]). Each pupil without exception read them correctly, but some pupils, like AB18, still processed them as indicated above, (more details are given in section 3.2.4).

In all cases the investigator then presented the pupil with a series of tasks and asked him/her to work each one explaining as he went along exactly what he was doing. In some cases the investigator asked the pupil to tell him which of two alternative forms were correct and frequently asked the pupil to explain why. The tasks presented were different for each pupil, and were based on the difficulties noted in the individual's September test. The interviewer thus started each session with a list of task types to be explored, but often generated particular tasks as a result of answers given to questions.

The following is a summary of the main features noted during the interviews:

1. Some pupils searched for solutions as they were unable to compute $M{\bullet}X + N{\bullet}X$ or deal with $M{\bullet}X = N$, when $M>N$.

2. Some compute a separate value for each X given in the equation.

3. One pupil admitted that there were a number of quite distinct ways of solving an equation (even when it is demonstrated that each approach leads to different answers).

4. Some pupils have "hard", consistent, mal-rules.

5. Some pupils have the correct rules and can explain why it is not permissible to perform the illegal transformation (*frequently the one the pupil appeared to use in* May was selected).

Each of these points are discussed in the following sub-sections.

### 3.2.1 Searching for Solutions

Searching for a solution appears to be a very common way of solving equations with pupils beginning Algebra, and presumably arises because the initial equations presented could be solved using this algorithm. That is given an equation of the form:

$$3*X = 2*X +5$$

the pupil substitutes X=1, then X=2, then X=3.... (See [Sle ap3] for further details of pupil AB-11's protocol).

Indeed, in a more recent test with 100 13 year olds, it appears that about 95% of them use this approach. And we argue that this leads to the type of errors noted below for tasks where this naive algorithm is inappropriate. For instance, a significant proportion experience difficulties with tasks of the form

$$3*X = 2$$

*Many of them return the answer:*

X = -1, explaining they had subtracted 3 from both sides.

These same pupils are often unable to solve equations which contain 2 Xs and attempt to *guess a value for each X, this will be discussed in more detail in the next subsection.*

It is indeed intruiging to watch pupils changing their approach when solving tasks. For example, we have seen:

$$M*X=N$$

solved differently depending on whether the task is solvable by search. Pupils do not appear to notice this discrepency. [Clearly this point should have been raised in an interview with these pupils].

Teachers should thus be suspicious that a pupil is using the naive algorithm if he appears to be unable to solve tasks where the variable is a negative integer, large-integer or non-integer. The teacher should be concerned because the naive algorithm is only applicable to a sub-set of Algebraic equations, and hence should be deemed a highly significant weakness, and one to be remedied. It seems clear that the use of "simplistic tasks" leads to a naive algorithm which causes major conceptual difficulties on more "advanced tasks".

### 3.2.2 Multiple Values for X

In this section, we report a pupil who has a very wierd, but nevertheless very consistent algorithm for solving tasks involving 2 Xs. When pupil AB-7 was orignally working at the terminal, she was heard to mutter:

"If this X was 2, then it would work if this second X was 4".

Moreover, in both the paper-and-pencil exercise and in the interview this pupil has been remarkably consistent. (See [Sle ap3] for her protocol). That a pupil would follow such an algorithm was originally a great source of amazement to me, and I should add to the School's Maths staff. (However, as noted below, this behaviour has since been observed with a substantial proportion of 13 year olds).

On the other hand, pupil AB7 was able to explain exactly what she was doing. Given the task:

$$3 * X + 2 * X = 12$$

She gave the following explanation:

"What I do is take the 3 and I make the first X equal to 2, so I write:

$$3 * 2$$"

When asked by the interviewer why the "first" X is equal to 2, she explains that it's the next number along, and then added "she thinks this is the wrong thing to do, but that's what she does".

She then continued " and then I write down the +2 making

$$3 * 2 + 2$$

I then work this out, this is equal to 8 and so the second X is

12 - 8, that is 4".

She then completed the solution and gave the 2 values for X, and so the final state of her worksheet was:

$$3 * 2 + 2 + 4 = 12$$

$$X = 2$$

$$X = 4$$

NB she used this algorithm consistently on 9 tasks, see [Sle ap3].

Initially, I had supposed this to be a very idiosyncratic algorithm, but subsequently noted that a variant was used by a substantial number of 13 year olds. And so I have seen:

$$3*X+4*X=3$$

"solved" as:

$3*1 + 4*0 = 3$, making $X=1$ and $X=0$.

Similarly,

$$3*X + 4*X = 98$$

has been "solved" as:

$$3*22 + 4*8 = 66 + 32 = 98.$$

Note that in "complicated" cases the two sides often are not "balanced". Thus I have seen

$$3*X + 4*X = 100$$

"solved" as:

$$3*30 + 4*2 = 100$$

and when asked the pupil explained that "this one did not work out exactly".

### 3.2.3 Alternative Algorithms

Although for several task types, pupil AB17 was able to solve the tasks correctly, he was easily "distracted" and quite unable to tell the investigator why the investigator's "alternatives" were illegal. On some tasks the pupil suggested several illegal solutions, and again was really unable to distinguish between them, see Figure 6 for details.

On the other hand, this pupil did give, as an aside, a rationale for his "method", which we will discuss in detail in section 3.3c.

[Figure 6 about here]

### 3.2.4 "Hard"/Consistent Mal-Rules

Many of the pupils were using consistent mal-rules. Just over half of the 24 pupils we saw mis-handled precedence in equations of the form:

$$2 + 3*X = 9$$

Part of a protocol for one such pupil is given in Figure 7.I, [6]. Figure 7.II is part of the protocol produced by the pupil discussed in section 3.2.3, where he consistently applies a further intriguing transformation to a complete set of tasks. In order to understand this protocol fully we have suggested that a normalization step takes place between stages 1 and 2 of say protocol a. That is we are suggesting that the pupil applies the mal-rule to the original task, this results in an "unusual" form which the pupil then "normalizes" before continuing to process the rest of the task. (See [Sle ap3] for a lengthier discussion of "normalization").

[Figure 7 about here]

Pupil AB-18, Figure 7.III, is remarkably consistent with his mal-rules over a whole range of task types. Note the application of his algorithm to task c which involves 3 X-terms. (To give him justice, he realises that he had got tasks d) - g) wrong as he noticed that the equations did not balance when he substituted his answers back in). Further having worked task h), he noticed that when he moved the 4 across to the Right Hand Side, he changed the sign and so he then suggested that when he move the X

---

6. Recently we have discovered that 90% of a sample of 13 year olds had precedence difficulties with arithmetic expressions involving the "+" and "*" operators.

(associated with $2*X$) to the LHS, he should also change its sign. He then verbalised that

$$X - X \text{ is } 0,$$

and so the LHS became 0 and the RHS did not, and so he realized that this proposed solution was impossible. However, for good measure he also worked task (i) with the "revised" algorithm.

In the course of our discussion this pupil also gave the basis for his "algorithm", which is discussed in more detail in Section 3.3c.

3.2.5 "Saved Souls"

In September pupil AB5 worked correctly tasks which she had got consistently wrong in May, namely task sets 7 and 8. For task set 8 she appeared to use mal-rule:

$$M*X = N*X + P \Longrightarrow X + X = M + N + P.$$

Moreover, when presented with a falatious alternative during the September interview, she was able to spot it and to say why it was wrong. (For example, not able to add a number to an X term, not able to separate a number from an X term etc., see [Sle ap3] for more details).

In May, this pupil showed a lack of understanding of basic Algebraic notation, which appeared to be remedied by September. To see whether this was the case I also presented tasks from sets 12 and 13 of Figure 3, ie tasks of the form:

$$M + N*X + P*X = Q$$

and

$$M*X = N + P*\langle Q*X + R\rangle$$

All of which she worked correctly and was able to verbalise the stages she went through. I also presented an equation which contained an "unusual" variable, AA, and again this was worked correctly.

Similarly several other pupils, eg. AB4 showed substantial "progress", and again it was associated with the ability to explain what they were doing. In the next section we give a summary of the points inferred from these various analyses.

### 3.3 Summary of the Experiment.

There appear to be six major, and not totally unrelated, points:

#### a) Why is Level 5 a "Blackspot"?

In section 3 we listed a number of questions which occurred to us as a result of a preliminary analysis of the modelling session with LMS. These included the observation that some pupils could solve certain earlier task types, but appeared to fail on a particular set, and yet can subsequently go on to solve further (more difficult) sets. In particular, we had noticed pupils having difficulties with tasks of type:

$$2*X + 3*X = 10$$

yet be able to solve tasks of type:

$$2*X + 4 = 16 \quad \text{and} \quad 4 + 2*X = 18.$$

From talking to the pupils it became clear that they were searching for a solution, <u>because</u> they did <u>not</u> know how to compute the sum of M*X and N*X. The second and third tasks thus appear <u>easier</u> to them because it did not contain this "difficulty". Thus, the interviews very nicely resolved this issue.

#### b) Mal-Rules retained as genuine

As suggested in the Introductory section, we believe that many of the mal-rules reported in Figure 5, were "phantom" and a result of the pupil <u>having</u> to give a response to the Modeller. Indeed, we now beleive mal-rules sets 4 and 6, of figure 5, are spurious. (The modeller has since been changed so that it is now possible for the pupil to indicate that he wishes to give up on a particular task). Also there are additional mal-rules that should be added, namely those which can be generated by the Schema discussed in sub-section 3.3d.

#### c) A proposed Categorization of pupil errors for Algebra.

Given the mal-rules reported in Figures 1b and 5, and bearing in mind the various anomalies not explained in the pupil protocols we wish to propose 4 classes of errors:

1. Manipulative errors.

2. Incorrect Representation of the Task, ie "parsing" errors.

3. Executive/Clerical errors.

4. Random or "Wild" errors.

We shall deal with each of these classes in a separate sub-paragraph.

Manipulative Errors We would now claim that all the mal-rules reported in figure 1 and those numbered 7-9 in figure 5 fall within this category. [In sub-section 3.3d we propose a schema for generating manipulative mal-rules].

We have also observed "Overload" causing errors to occur with lower level rules. Earlier, we noted that each task set focussed on specific rule(s). What we have observed in this experiment is that although the pupil may sucessfully work all the tasks in the task set which focuses on SOLVE, he may well make errors with this rule when he is focussing on more complex rule(s). This is essentially what the pupil in figure 2a does: note in particular that with task-set 14 where he is concentrating on the MULT and XADDSUB rules he makes 2 errors with SOLVE. Note too that in figure 2b at level 10 this pupil makes consistent errors with the XTOLHS rule when the "focus" is the BRA2 rule. This phenomena, we again suggest, is due to overload.

Incorrect Representation of the Task/Parse Errors. We have categorized the first 6 sets of Mal-Rules in figure 5 as ones which summarize what happens when a child "mis-sees" or mis-parses, an algebraic equation. The above sections have addressed this issue at some length. (For the moment, then we merely note that parsing errors appear to have a totally different basis from "overload" errors and hence need to be treated very differently in remedial sessions).

Executive/Clerical/Unexplained errors. Analysing some of the protocols, one is happy with the explanation that some "slips" occur. For example:

$$10 \cdot X = 25 \Rightarrow X = 25/18$$

$$2 \cdot X = 6 \cdot 5 \Rightarrow X = 18$$

In the first case the pupil has probably seen the "0" as an "8". In the second he has probably made an arithmetic error. DEBUGGY [Bur 82b] considers an answer to be a "number-bond" "slip" if the answer is within 2 of the correct one. The second slip given above could be explained if we had an analogous algorithm for the evaluation of multiplicative expressions. However, the first slip, a "visual" one, could clearly not be. So we suspect that to account for the variety of "slips" encountered in this domain a more sophisticated approach, cf that advocated by Norman [Nor 81] would be necessary. However, we have not thought this worth investigating as clerical errors appear to be relatively infrequent when the pupil has settled into the system.

"Wild"/Inconsistent Errors. However, many of the mistakes which we have not so far explained will be due to the consistent use of mal-rules which we have not so far identified. [7]; others will be caused by strange processes which even the pupil may not be able to reproduce on the identical task.

d) Schema for "Generating" Mal-Rules

Parsing Mal-rules.

In Figure 7.III we gave a substantial section of pupil AB18's protocol. In the course of our discussion he explained that he was carrying out the teacher-given algorithm of: "Collecting all the Xs on the left hand side and collecting all the numbers on the right hand side", and added that he was not really sure what to do about the "extra multiply signs". Pupil AB17 gave a similar explanation for his actions.

This gives us a schema for generating mal-rules. For example given the task type:

$$M*X + N*X = P$$

This schema gives the following "action sides" for mal-rules:

$$X+X = P-M-N$$

$$X+X = P+M+N$$

where in the second case the X coefficients are treated "specially", ie the coefficients

7. As this is clearly a very demanding task there is a need to implement some computational device to assist the investigator. A preliminary system has been implemented which has already given several "explanations" not spotted by the Investigator.

of the Xs were taken across to the RHS of the equation but the signs were NOT changed. And the form given by pupil AB-17, and quoted in Figure 7.II, namely:

$$*X*X=P-M-N$$

which he went on to "normalize" (see section 3.2.4) to:

$$X*X=P-M-N$$

Its "complementary" form being:

$$X*X=P+M+N$$

Similarly, given the task type:

$$M*X = N*X + P$$

This schema creates the following forms:

$$X = N + P - M$$

$$X = N + P + M$$

$$X + X = N + P - M$$

$$X + X = N + P + M$$

$$X - X = N + P - M$$

$$X - X = N + P + M$$

For example on task h, pupil AB18 suggested the use of both the third and the fifth forms (see Figure 7.III).

Manipulative Mal-rules.

Analogously, it appears possible to generate all the mal-rules associated with a manipulative rule by systematically removing one or more of the rule's sub-steps. Also in this sub-section we briefly discuss the (apparently) related phenomena of confusion of operands.

In figure 5 we report 3 new (manipulative) mal-rules, namely mal-rules 7, 8 and 9, variants of SOLVE, SIMPLIFY and BRA2 respectively, which can be explained by such a mechanism, as can most of the "original" mal-rules given in Figure 1b. Note that this schema would ALSO generate many mal-rules which we have NOT yet observed.

A variant on SOLVE. [8]. The pupil realizes he has a task in which the SOLVE rule should be activated and he forgets to apply one of the operations, namely dividing by M. SOLVE has three principal actions: noting down N, the divide symbol and M, and so this mal-rule could be said to be omitting some of the principal steps, [9]. As Brown and Burton, [Bro 78], note the pupils have an idea about the acceptable FORM of answers and so given:

M*X = N, they do NOT produce X = /M or X = N/

A variant on SIMPLIFY [8]. Examples of the two rules given here, which have occurred reasonably frequently are:

X = 6/4 => X = 3/4

X = 6/4 => X = 6/2

Again this rule can be represented as having 3 principal steps (calculate the common factor, divide "top" by common factor, divide bottom by common factor) and that each of these mal-rules corresponds to one of the latter steps being omitted.

A variant on BRA2 [8]. Clearly the action here has many more steps and so one might expect to find a corresponding larger number of mal-rules. This is indeed true. This "new" mal-rule also conforms to the pattern noted above, as it is caused by the omission of one sub-action.

## Confusion of Operands.

We have noted errors of the following form:

5*X = 12 ==> X = 2 2/12

where clearly one operand is confused for another. Norman, [Nor 81], "explains" such slips by saying that they are a consequence of a noisy processor.

---

8. The variant on SOLVE reported in Figure 5 is:

M*X = N => X=N

Two variants on SIMPLIFY reported in Figure 5 are:

M*X = N => X = (N/F)/M

M*X = N => X = N/(M/F)

where F is a factor of M and N. [More correctly both SOLVE and SIMPLIFY rules are being applied in one step, but the error appears to be associated with the SIMPLIFY rule].

The variant on BRA2 reported in Figure 5 is:

M*<N*X+P> => M*X + M*P

---

9. This level of analysis for each rule, essentially gives us a representation for the whole of the task domain. Further it appears that this grain is sufficient to capture all the problems encountered by the pupils.

e) Longitudinal Studies

From this set of pupils alone, I think it is possible to infer that behaviour progresses as:

UNPREDICTABLE/"WILD" -> CONSISTENT USE of MAL-RULES -> CORRECT

Pupils who were really unsure which method to use would apply different "methods" to different tasks, and quite randomly. (Analogously, Greeno and his collaborators who are investigating pupils' performance whilst they are being taught Algebra, have noticed very considerable variations with many of the pupils' performance, [Gre 81]). Then there are the pupils, cf Figure 7 who are using consistent mal-rules on particular task sets. Further, there are some, like pupil AB18 of Figure 7.III, who are using a schema which guides their action in a whole variety of situations. Once, the correct algorithms are understood, it appears that pupils are able to explain them quite articulately.

The observation I wish to make here is that with experience/maturation pupils appear to move through the stages given above. I am not wishing to suggest that every pupil acquires a consistent mal-rule before he is able to do the task correctly, but am wishing to indicate a "trend". Similarly, it is not intended to suggest that no pupil regresses.

Results obtained over a period of a year and a half, with various groups aged from 13 to 15 support this claim, as does the study reported by [Kch 81].

f) The significance of the different types of Mal-rules and the Remedial Teaching Experiment.

We argued in section 3.3c that the errors pupils make, fall within a number of distinct classes. The interviews gave us clear evidence that these different types of errors (manipulative and parsing) arise because of considerably different processes. (In the case of the manipulative errors the pupil overlooks a sub-step, in the case of the parse error the pupil has a non-standard perception of the task). Practically, this distinction is of considerable importance as it enables one to give appropriate remedial instruction for the several types of error. In the case of the "manipulative" mal-rules it would appear that the pupil basically "knows" the rule, but due to Cognitive overload, or inattention is omitting one or more sub-steps. The parsing errors appear to arise from a profound misunderstanding of algebraic notation. See also the discussion in section 5.

However, it was necessary to determine whether, once a pupil's shortcomings have been "diagnosed", it is possible to carry out remedial instruction, which will lead to a long-term improvement in the pupil's solving of Algebraic equations. As a high percentage of the group mis-handled mixed expressions of the form:

$$2+3*X = 6$$

this and related points, were dealt with in a class lesson. Those pupils with very individualistic difficulties were seen individually by the investigator, who spent on average half an hour with each pupil. A post-test was administered 2 months later. In order to factor out the effects of "natural Maturation" and subsequent instruction, the post tests were also given to pupils who had not been screened earlier; the staff paired the pupils on the basis of their performance on their 1981 end-of-year examination.

The results for the post-tests are given in detail in [Sle ap3]; but briefly they show that it IS possible to give pupils in this area effective remedial instruction, [10]. Some other studies, for example [Bro 78] and [Gin 77], have suggested that it is hard to remove persistent bugs. It is suggested that algebra may be more easily remediated than mechanics, or counting, since in the latter areas intrinsic personal knowledge about the real-world may effect the pupil's perspective. Similarly, it might be that bugs in electronics are easier to remedy than bugs in aspects of mechanics where the pupil again may draw upon his own (direct) experiences, [11].

## 4. PROTOCOLS OBTAINED WITH LMS.II USING AN ENHANCED DATA-BASE

---

10. VanLehn has suggested (personal communication) that to aid the pupil's long-term retention that a mnemonic should be taught which helps the pupil distinguish between the correct rule and the mal-rule(s) which he is "known" to be using.

---

11. It is planned to analyse the remedial teaching dialogues discussed above, with a view to enhancing the Modelling System, such that it will perform Remedial Teaching, using the techniques noted in these dialogues, and using the models inferred by the Modeller.

Figure 8 gives the Protocols obtained with LMS using an enhanced data-base, when it is presented with the same type of task as the pupils had been given in their interviews. (That is, Figure 8.I corresponds to Figure 7.I and 8.II corresponds to 7.III). We notice that LMS now gives the expected diagnosis for each of these situations, ie the diagnosis which agrees with the observations made during the interviews. (NB the first diagnosis could have been achieved with the "original" data-base). In Figure 9, we give a list of the parsing mal-rules which were newly introduced into the data-base.

[Figure 8 and 9 about here]

### 4.1 Enhancement to LMS to accommodate the Parsing Mal-Rules.

In our earlier formulations we assumed that a pupil would use a rule or its associated mal-rule consistently within the same task. That is, if the task was:

$$2*X + 5 + 6 = 19$$

we assumed that the pupil would either use NTORHS or say MNTORHS consistently in moving the 5 and the 6 to the RHS. Given this framework the Parsing Mal-rules caused some difficulty, as they appear only to be used during the initial processing of the task and NOT in subsequent stages.

Suppose the task is: $3*X + 4*X = 13$

and suppose we are attempting to replicate the folowing protocol:

a) $3*X + 4*X = 13$

b) $X + 3 + X + 4 = 13$

c) $X + X = 13 - 3 - 4$

d) $2*X = 6$

e) $X = 3$

Now suppose we included the appropriate mal-rule, MPO, in a model then it would fire at both a), as desired, but also at d) giving the answer $X = 4$. [12]. However, we have also observed the following protocol:

a) $8*X = 5 - 3$

b) $8*X = 2$

---

12. Indeed if MPO fires when the X coefficient is 1, it would loop indefinitely!

c) X = -6

And so we have concluded that the parse mal-rules should only be able to fire once with each task; indeed, when present these mal-rules should be the first rule to fire. Thus the algorithm for generating models, the Complete and Non-redundant model generating algorithm [Sle ap1], has had a second phase added. The complete algorithm is now:

1. Generate all the Non-redundant models appropriate to that level's Ideal model.

2. Generate the models which include a (satisfied) Parsing mal-rule, using the following sub-algorithm:

    1. Create an enhanced model consisting of one Parse Mal-rule, which is satisfied by the task-type, the original ideal model and the following rules:

       (REARRANGE ADDSUB NTORHS SOLVE SIMPLIFY FIN)

       The latter are added to the model so that the state generated by the Parse Mal-rule could be completely processed. For example, for Task-set 3 the ideal model is:

       (ADDSUB SOLVE SIMPLIFY SGNS FIN)

       Parse mal-rule, MPO, is satisfied by this task-type, so this sub-algorithm generates the configuration:

       (MPO REARRANGE ADDSUB NTORHS SOLVE SIMPLIFY SGNS FIN)

    2. Carry out the previous sub-step for each of the parse mal-rules satisfied by the task-type.

    3. All of these configurations are then processed by the Complete and Non-Redundant model generation algorithm. (Each of this set of models will include a parse mal-rule).

The off-line phase of the Modeller is completed by appending the 2 sets of models together, and evaluating this model-set against the prespecified tasks for that "level". (Note, when the pupil is on-line his answers are then merely checked against those produced in the off-line phase just described, see [Sle ap1] for more details).

## 5. LMS and RELATED WORK in COGNITIVE MODELLING

There is a steadily growing body of data about how pupils and students solve Algebra tasks, [Pag 66], [Lew 80], [Dav 78] and [Kch 81], for a review of this work see [Sle ap3] and [Lov 80]. In this section, I discuss the contribution which this work makes to Cognitive Modelling. In subsection 5.1 I outline some of the approaches made by earlier workers and in sub-section 5.2 I attempt to interpret some of the observations made in this experiment using these frameworks.

### 5.1 An overview of some pertinent work in Cognitive Modelling

BUGGY [Bro 78] analysed the responses which pupils gave to column subtraction tasks. The system reported a diagnosis for each pupil in terms of procedures, or procedures which had some of their sub-steps replaced by incorrect variants, which they called bugs.

Young and O'Shea [You 81] point out that although BUGGY produces models which behave functionally as the pupils, these models are not very convincing as psychological models as many of the bugs appears to be very similar (by and large, they are all connected with borrowing from zero) yet this relationship is not made clear. More particularly Young and O'Shea go on to show that the "BUGGY data" can be analysed more simply in terms of certain competences being omitted from the "ideal" model.

Brown and VanLehn's stimulating article [Bro 80] on Repair theory is a further attempt to provide a psychological explanation for the same data. Here they take a correct procedure for performing the task, column subtraction, and apply a deletion operator to the proceduro. This perturbed procedure is then used to solve tasks, and when it encounters an impasse (i.e. a situation where it is about to violate a precondition, e.g. attempting to take a number from 0), it applies a repair to the perturbed procedure, and attempts to continue solving the task at the stage before the impasse occurred. This process also uses critics to throw out some bugs which are considered impossible, and it also rejects certain deletion operations, [13].

More recently, VanLehn [Leh ap] has suggested a variant of repair theory, which does

---

13. In its initial form, some impossible bugs were generated and only 21 out of an observed set of 89 bugs were generated by Repair Theory.

not delete steps from procedures (as it is argued that the blocking, or inhibition, of the deletion operator was unprincipled); and secondly this version overcomes the difficulty that certain core procedures cannot be generated easily by rule deletion. Instead, VanLehn has suggested a series of core procedures, which correspond to the various stages of instruction, cf [Sle .81], and a set of repairs which can be applied to any of these procedures. From this perspective an impasse occurs when the pupil encounters a sub-task which he has not yet learnt, or has forgotten.

Both variants of the Repair Theory explain what they have called bug migration, namely that with a particular type of task the pupil may display different bugs both during the same test-period and between different tests. Moreover, in his more recent analysis VanLehn [Leh 81] has analysed protocols in which it was possible to generate all the observed bugs by applying different repairs to a common (partially learnt) core procedure. So the explanation for a consistent bug is that the child stores the "patch" and merely uses it with the next task. The argument for inconsistency is that the patch is NOT retained and that one of the repair-set is selected randomly.

The Illinois group [Dav 78], has reported Algebra pupils over generalising from instances, using an "old" operator instead of a more recently introduced one [14], and regressing under cognitive load. Matz [Mat 82] has further analysed these pupils' performances and has suggested a number of high level schema which explain series of observed errors, these include her "extrapolation principle" which explains why a pupil who has seen the legal transformation:

$$(A * B)\ 1/n \implies A\ 1/n * B\ 1/n$$

would then write:

$$(A + B)\ 1/n \implies A\ 1/n + B\ 1/n$$

---

14. + instead of *, * instead of Exponential.

She also discusses the confusion which seems to arise between the Arithmetic- and Algebra-notations. For instance, she argues that as 3 3/4 is to be interpreted as 3 + 3/4, it is not unreasonable that the pupil should interpret an analogous algebraic expression, 3X as 3 + X, [15].

### 5.2 Pertinent Observations from this Experiment

In this section we consider possible explanations for the following observations made in the 1981 Algebra experiment.

1. Pupils appear to regress under Cognitive load. (section 5.2.1).

2. There appears to be a number of clearly identifiable *types* of errors. (Section 5.2.2).

3. Pupils use a number of Alternative "methods" to solve tasks of the same set. (section 5.2.3).

#### 5.2.1 Regression under condition of overload

This seems a reasonable explanation of the behaviour which was noted for task set 14 for the first protocol of figure 2, and again for task set 10 for the second protocol. (Similarly, such behaviour was noted by Davis and his colleagues [Dav 78]). So far LMS reports a model for each pupil's performance on each task set. And so in the case of pupil TRAB3 (figure 2a), SOLVE is contained in LMS's models for the pupil for all *except* set 14, where LMS returns two alternatives, one of which contains SOLVE and the other MSOLVE. Thus if we were to produce a "composite model" for this pupil to cover this range of tasks we might include:

if OVERLOAD then MSOLVE else SOLVE

---

15. However, although this explanation would explain *some* of our observations with pupils AB17 and AB18, they gave us an *alternative* and more comprehensive explanation for their actions, see section 3.3d.

where the conditions for OVERLOAD would have to be specified (inferred) additionally. Analogously, we have observed the following responses from a pupil:

3*X = 12 ==> X = 4

6*X = 10 ==> X = 5/3

12*X = 3 ==> X = -9

10*X = 6 ==> X = -4

Namely the pupil uses a different rule in the case where M>N (where the task is represented as M*X=N). That is, the pupil's behaviour on these tasks can be summarized as:

if M>N then X=N-M else X=N/M

This issue arises here for the first time because LMS attempts to infer models for a relatively diverse set of tasks, that is much more diverse than those covered with the subtraction tasks. (Note that BUGGY, as LMS, produces a model for each individual pupil, whereas Young and O'Shea carried out an analysis on a task-by-task basis).

### 5.2.2 Mechanisms to explain the Observed Mal-Rules

In section 3.3c we proposed a categorization of pupils' errors observed in this experiment; the Manipulative and Parsing errors being the most significant classes. In section 3.3d we proposed a schema for generating Manipulative mal-rules which involved dropping one of the sub-steps (or a variant, a confusion of operands) which is thus consistent with Young and O'Shea's modelling of subtraction.

However, the Parsing mal-rules cannot be explained by omitting a component, (Young and O'Shea). Neither it seems can they be explained by performing a repair to a core procedure, unless one is prepared to broaden one's view of a repair to include the Schema which were observed with pupils AB17 and AB18 and the "extrapolation" procedure noted by Matz. Further, VanLehn's notions of the repair being applied to a partially learnt/remembered core procedure seems convincing, as many of the children interviewed quite clearly did not know how to add M*X and N*X and hence search for a solution (section 3.2.1) or apply the extrapolation "schema" (sections 3.2.3 and 3.2.4). A further mechanism is proposed in the next section.

### 5.2.3 Bug Migration or Using Alternative Methods

As we saw in 5.1, Repair Theory gives a neat explanation for their observed phenomena of bug migration, namely that the pupil will use a related family of bugs during a single session with one particular task set.

There seems to be an alternative explanation which should also be considered. Although a task-set may have been designed to highlight one particular feature, the pupil may spot completely different feature(s) and these may dominate his solution, [16].

Repair Theory accounts for some bugs by hypothesizing that the pupil had not encountered the appropriate teaching necessary to perform the task. Suppose we make the converse assumption, that the appropriate teaching had been carried out, and further suppose that pupils do not gain competence, in this domain by being _told_ the rules but rather by _inferring_ rules for themselves by noting the transformations which are applied to tasks by the teacher and in texts. It seems reasonable that the pupil's inference procedure should be guided by his previous knowledge of the domain, in this case the number system, and that the pupil will, in general, infer several rules which are consistent with the example, and not just the "correct" rule. Indeed due to some missing knowledge the "correct" rule may not be inferred. (And so the fact that the pupil _never_ uses the "correct" method along with several "buggy" methods is _not_ evidence that he has NOT encountered the material before).

Suppose, the pupil saw the following stages in an algebraic simplification:

$$3*X = 6 \qquad ==> \qquad X = 6/3$$

Then he might infer

---

16. Earlier Sleeman and Brown, [Sle 82b], have argued: "......Perhaps more immediately, it suggests that a Coach must pay attention to the sequence of worked examples, and encountered task states, from which the pupil is apt to abstract (invent) functional invariances. This suggests that no matter how carefully an instructional designer plans a sequence of examples, he can never know all the intermediate steps and abstracted structures that a pupil will generate while solving an exercise. Indeed, the pupil may well produce illegal steps in his solution and from these invent illegal (algebraic) "principles". Implementing a system with this level of sophistication still presents a major challenge to the ITS/Cognitive Science community... "

X = RHS number/LHS number

OR

X = LARGER NUMBER/SMALLER NUMBER

The argument then is that both these rules are stored with the initial task type and that the pupil will use these randomly, (suppose the selection actually depends on the weighting factors for the several rules). The successful use of an inferred rule, will strengthen the pupil's belief in the rule. Tasks, and the corresponding "worked" solution, which show that a rule is inadequate will weaken belief in the rule, but it is argued that once a (mal)-rule is created it may never be completely eliminated; particularly if the "counter-examples" are not presented to the pupil for some period. Thus given this view point, the phenomena of bug-migration occurs because the pupil has inferred a whole range of rules and he merely "randomly" selects a rule. Given a further task, he again randomly chooses a method and hence selects the same or an alternative algorithm, depending on the relative strengths of the rules. That is if the relative weights are comparable, it is more likely that the pupil will select a different method for each task. If one weight "dominates" then it is likely that the corresponding method will be selected frequently.

Further, if only one (mal) rule is generated by the induction process then this approach predicts that the pupil will act consistently.

I suggest that many of the bugs encountered in the Subtraction domain can be accounted for by this (inference), mechanism. For instance the Subtract-Larger-from-Smaller bug, where the smaller number is subtracted from the larger independent of whether the larger number is on top or the bottom row, seems one such example, [Bro 78] and [You 81]. Brown and VanLehn [Bro 80] report that because borrowing was introduced, with one group of pupils, using only tasks with 2 columns, that these pupils inferred that whenever borrowing was involved they should borrow from the left-most column, their "Always-Borrow-Left" bug. (Thus it appears important to ensure that the example set includes some counter examples; indeed it seems as if task-sets can be damaging if they are too preprocessed and contain too little "intellectual ruffage". Michener, [Mic 78], puts a similar argument).

Additionally, Ginsburg, [Gin 77], quotes several instances of young children inferring information, for instance the names for "40", "50", "60", given the names for "4", "5" and "6" and for "30".

So given the wealth of experimental evidence this alternative explanation should be given serious consideration.

## 5.3 Summary

Firstly, we have two explanations for some of the misunderstandings we have noted with Algebraic notation. Namely, that given by Matz, and that given by pupils, AB17 and AB18, see section 5.1. Certainly Matz's explanations would explain some of our observations, but not all as in some cases the coefficients are treated "specially", and their sign is not changed when they are moved to the RHS. For example, we have observed:

$$3*X+4=12 \quad => \quad X=12+3-4$$

ie the pupil changes the sign of the 4 but not the 3.

Secondly, we have two hypotheses which explain bug-migration the one given by Repair Theory and the one put forward here, which I shall refer to as Knowledge Directed Inference of Multiple Rules. (Of course it is quite possible that each may be applicable in different situations). If for instance the investigator knows that the pupil has not been exposed to a particular topic then he is disposed to accept the (revised) Repair Theory rather than the other. In general, it should be possible to investigate this issue experimentally.

Thirdly, several "algorithms" have been presented about how to create student models and these indirectly suggest the cause of pupil's buggy behaviour. Repair Theory suggests that it can be explained by making "repairs" to incomplete core-procedures, whereas Young and O'Shea suggest that it is adequate to take a core procedure and merely delete components. The data for the Algebra Manipulative Mal-Rules can be adequately explained by either. However, Young and O'Shea's approach seems inadequate to explain the parsing mal-rules. Indeed, we have to extend the revised Repair Theory before the results reported here can be accommodated. (An analogous extension is needed to accommodate the Davis/Matz results). This paper has contributed a further explanation, namely Knowledge

Directed Inference of Rules, which appears to explain many of the facets of both the Algebra mal-parsing rules and the Subtraction data.

## 6. FOOTNOTE on METHODOLOGY

The evaluation of competing explanations illustrates an important aspect of Cognitive Science's research methodology. Namely, that it may be possible for more than one scheme to reproduce faithfully the principal features of the experimentally observed data. In which case, one attempts to discriminate by noting what the several systems predict for some other (lesser) observed data. If such data does not exist, or if this does not resolve the issue, one needs to carefully examine the constraints (psychological assumptions) which the two systems use and select the system with the preferred set of assumptions, [17].

The above comparisons of explanations (or theories) are important in that they remind us of the essentially pragmatic nature of our Science.

## 7. ACKNOWLEDGEMENTS

To Mr. M. McDermot and pupils of Abbey Grange School, Leeds, for providing fascinating sets of protocols. To Pat Langley, Jaime Carbonell, Jim Greeno, K. Lovell and Alan Bundy for numerous discussions about this work. Kurt VanLehn and Peter Jackson made valuable comments on an earlier draft of this paper. To the University of Leeds for granting me sabbatical leave. To the Sloan Foundation for providing me with some financial support. And to CMU for providing a wonderful environment in which to continue this work.

---

17. This is currently a central concern of the field, and it is unlikely that such criteria will be fully specified for some considerable time.

## 8. REFERENCES

[Bro 78] J.S. Brown & R.R. Burton (1978). Diagnostic Models for procedural bugs in basic Mathematical Skills, in Cognitive Science, 2,2. pp155-192.

[Bro 80] J.S. Brown & K. VanLehn (1980). Repair Theory: A generative theory of bugs in procedural skills. Cognitive Science, 4. pp 379-426.

[Bro 82] J.S. BROWN, R.R. BURTON, & J. deKLEER, (1982). Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II and III. In Intelligent Tutoring Systems, edited by D. Sleeman & J.S. Brown. New York: Academic Press, pp 227-282.

[Bur 82a] R.R. BURTON & J.S. BROWN, (1981). An investigation of computer coaching for informal learning activities. In Intelligent Tutoring Systems, edited by D. Sleeman & J.S. Brown. New York: Academic Press, pp 79-98.

[Bur 82b] R.R. Burton (1982). Diagnosing bugs in a simple procedural skill. In Intelligent Tutoring Systems, ed. D.Sleeman and J.S. Brown, New York: Academic Press, pp 157-183.

[Cla 82] W.J. CLANCEY (1982). Tutoring rules for guiding a case method dialogue. In Intelligent Tutoring Systems, edited by D. Sleeman & J.S. Brown. New York: Academic Press, pp 201-225.

[Dav 78] R.B. Davis, E. Jockusch & C. McKnight (1978). Cognitive Processes in Learning Algebra. Journal of Children's Mathematical Behaviour, 2, pp1-320.

[Dav 80] R.B. Davis & C. McKnight, (1980). The influence of Semantic Content on Algorithmic Behaviour. Tech report, Curriculum Laboratory, University of Illinois.

[Gin 77] H.P. Ginsburg, (1977). Children's Arithmetic: the Learning Process. New York: Van Nostrand.

[Gol 82] I.P. GOLDSTEIN, (1982). The genetic graph: a representation for the evolution of procedural knowledge. In Intelligent Tutoring Systems, edited by D. Sleeman & J.S. Brown. New York: Academic Press, pp 51-77.

[Gre 81] J. Greeno (1981). Personal Communication.

[Har 73] J.R. HARTLEY & D.H. SLEEMAN, (1973). Towards intelligent Teaching Systems. Int. J. Man-Machine Studies, 5, pp215-236.

[Kch 81] D. Kuechemann, (1981). Algebra in Children's Understanding of Mathematics: 11-16, edited by K.M. Hart, London: John Murray, pp102-119.

[Leh 81] K. VanLehn, (1981). Bugs are not enough: empirical studies of bugs, impasses and repairs in procedural skills. Xerox PARC, Cognitive and Instructional Sciences Tech. report, CIS-11.

[Leh ap] K. VanLehn, (to appear). On the Representation of Procedures in Repair Theory, in The Development of Mathematical Thinking ed H.P. Ginsburg. New York: Academic Press.

[Lew 80] C. Lewis (1980). Kinds of Knowledge in Algebra. Paper presented to the AERA Annual Conference, 1980.

[Lov 80] K. Lovell, (1980) Intelligent Teaching Systems and the teaching of Mathematics. AISB Quarterly, 38, pp 26-30.

[Mat 82] M. Matz, (1982) Towards a process model for high school algebra errors. In Intelligent Tutoring Systems, edited by D. Sleeman and J.S. Brown, Academic Press, pp 25-50.

[Mic 78] E.R. Michener (1978). Understanding Understanding Mathematics. Cognitive Science, 2, pp 361-383.

[Nev 78] D.M. Neves. (1978). A computer program that learns algebraic procedures by examining examples and working problems in a textbook. Proceedings of the 2nd National Conference of the Canadian Society for Computational Studies of Intelligence. pp191-195.

[Nor 81] D.A. Norman, (1981). Categorization of Action slips. Psychological Review, 88. pp1-15.

[Pag 66] J.M. Paige & H.A. Simon, (1966) Cognitive Processes in Solving Algebra Word Problems. In Problem Solving: Research, Method and Theory, edited by B. Kleinmuntz, Published by Wiley.

[Sle 81] D.H. Sleeman & M.J. Smith, (1981). Modelling student's Problem Solving. Artificial Intelligence, pp 171-187.

[Sle 82a] D. Sleeman, (1982) Assessing competence in basic Algebra. In Intelligent Tutoring Systems, edited by D. Sleeman and J.S. Brown, Academic press, pp 186-199.

[Sle 82b] D. Sleeman & J.S. Brown, (1982) Editorial in Intelligent Tutoring Systems, edited by D. Sleeman and J.S. Brown, Academic Press, pp 1-12.

[Sle 82c] D.H. Sleeman, & R.J. Hendley, (1982). ACE: A system which Analyses Complex Explanations. In Intelligent Tutoring Systems, edited by D. Sleeman & J.S. Brown. New York: Academic Press, pp 99-118.

[Sle ap1] D.H. Sleeman, (to appear). A Rule Directed Modelling system. In Machine Learning, R. Michalski, J. Carbonell, and T. M. Mitchell, (editors), Tioga Press.

[Sle ap2] D.H. Sleeman & K. Lovell (to appear). Some experiences of an Algebra game with pre-school children.

[Sle ap3] D.H. Sleeman, (to appear). Basic Algebra revisited: a study with 14 year olds, Journal of Mathematical Behaviour. [Also available as HPP memo 83-9].

[You 81] R. Young & T. O'Shea, (1981). Errors in Children's Subtraction, Cognitive Science, 5. pp153-177.

Figure 1

a) RULES for the ALGEBRA domain (evaluative form and slightly styalized).

| RULE NAME | CONDITION | ACTION |
|---|---|---|
| FIN2 | (SHD X = M/N) | (SHD (M N)) or (SHD evaluated) |
| SIMPLIFY | (SHD X = M/N) | (SHD X = M'/N') |
| SOLVE | (SHD M * X = N) | (SHD X = N/M) or (SHD INFINITY) |
| ADDSUB | (lhs M +!- N rhs) | (lhs [evaluated] rhs) |
| MULT | (lhs M * N rhs) | (lhs [evaluated] rhs) |
| XADDSUB | (lhs M*X +!- N*X rhs) | (lhs (M +!- N) * X rhs) |
| NTORHS | (lhs +!- M = rhs) | (lhs = rhs -!+ M) |
| REARRANGE | (lhs +!-M +!- N*X rhs) | (lhs +!- N*X +!- M rhs) |
| XTOLHS | (lhs = +!- M*X rhs) | (lhs -!+ M*X = rhs) |
| BRA1 | (lhs < N > rhs) | (lhs N rhs) |
| BRA2 | (lhs M*<N*X +!-P> rhs) | (lhs M*N*X +!- M*P rhs) |

Where M, N and P are integers and where lhs, rhs etc are general patterns (which may be null), where +!- means either + or - may occur, where SHD indicates the String Head and where < and > represent standard "algebraic brackets".

b) Some MAL-RULES for the Domain

| RULE NAME | CONDITION | ACTION |
|---|---|---|
| MSOLVE | (SHD M*X = N) | (SHD X = M/N) or (SHD INFINITY) |
| MNTORHS | (lhs +!- M = rhs) | (lhs = rhs +!- M) |
| M2NTORHS | (lhs1 +!- M lhs2 = rhs) | (lhs1 +!- lhs2 = rhs -!+ M) |
| M3NTORHS | (lhs1 +!- M lhs2 = rhs) | (lhs1 +!- lhs2 = rhs +!- M) |
| MXTOLHS | (lhs = +!- M*X rhs) | (lhs +!- M*X = rhs) |
| M1BRA2 | (lhs M *<N*X +!- P> rhs) | (lhs M*N*X +!- P rhs) |
| M2BRA2 | (lhs M*< N*X +!- P> rhs) | (lhs M*N*X +!- M +!- P rhs) |

Using the same conventions as above.

c) Pairs of correct and "buggy" models executing typical tasks.

i) shows (MULT ADDSUB SOLVE FIN2) and (ADDSUB MULT SOLVE FIN2) solving 3*X = 5 + 3 * 4.

[The first line gives the initial state and all subsequent lines give the rule which fires and the resulting state.]

| | | | |
|---|---|---|---|
| | 3*X = 5 + 3 * 4 | | 3*X = 5 + 3 * 4 |
| MULT | 3*X = 5 + 12 | ADDSUB | 3*X = 8 * 4 |
| ADDSUB | 3*X = 17 | MULT | 3*X = 32 |
| SOLVE | X = 17/3 | SOLVE | X = 32/3 |
| FIN2 | (17/3) | FIN2 | (32/3) |

ii) shows (NTORHS ADDSUB SOLVE FIN2) and (MNTORHS ADDSUB SOLVE FIN2) solving 4*X + 5 = 19.

| | | | |
|---|---|---|---|
| | 4*X + 5 = 19 | | 4*X + 5 = 19 |
| NTORHS | 4*X = 19 - 5 | MNTORHS | 4*X = 19 + 5 |
| ADDSUB | 4*X = 14 | ADDSUB | 4*X = 24 |
| SOLVE | X = 7/2 | SOLVE | X = 6 |
| FIN2 | (7/2) | FIN2 | (6) |

Figure 2

Task set 2
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 5 0 0 0 0 0
Models which explain behaviour for this Task set are
(SOLVE SIMPLIFY FIN FIN2)

Task set 3
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 4 1 0 0 0 0
Models which explain behaviour for this Task set are
(ADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 4
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 5 0 0 0 0 0
Models which explain behaviour for this Task set are
(MULT SOLVE SIMPLIFY FIN FIN2)

Task set 5
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 7 7 0 0 0 0 0
Models which explain behaviour for this Task set are
(XADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 6
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 6 6 0 0 0 0 0
Models which explain behaviour for this Task set are
(NTORHS ADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 7
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 6 6 0 0 0 0 0
Models which explain behaviour for this Task set are
(REARRANGE ADDSUB NTORHS SOLVE SIMPLIFY FIN FIN2)

Task set 8
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 6 6 0 0 0 0 0
Models which explain behaviour for this Task set are
(XADDSUB XTOLHS SOLVE SIMPLIFY FIN FIN2)

Task set 9
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 4 4 0 0 0 0 0
Models which explain behaviour for this Task set are
(MULT ADDSUB BRA1 SOLVE SIMPLIFY FIN FIN2)

Task set 10
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 4 3 1 0 0 0 0
Models which explain behaviour for this Task set are
(BRA2 XADDSUB XTOLHS SOLVE SIMPLIFY FIN FIN2)

Task set 11
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 3 2 0 0 0 1 0
Models which explain behaviour for this Task set are
(MULT ADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 12
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 3 3 0 0 0 0 0
Models which explain behaviour for this Task set are
(REARRANGE XADDSUB ADDSUB NTORHS SOLVE SIMPLIFY FIN FIN2)

Task set 13
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 3 1 0 0 2 0 0
Models which explain behaviour for this Task set are
(BRA2 REARRANGE ADDSUB XTOLHS XADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 14
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 3 1 2 0 0 0 0
Models which explain behaviour for this Task set are
(MULT XADDSUB SOLVE SIMPLIFY FIN FIN2)
(MULT XADDSUB MSOLVE SIMPLIFY FIN FIN2)

Task set 15
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 3 3 0 0 0 0 0
Models which explain behaviour for this Task set are
(MULT BRA2 XADDSUB XTOLHS SOLVE SIMPLIFY FIN FIN2)

Overall Result (Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild"):
63 56 4 0 2 1 0

<u>Figure 2a</u> <u>Summary of results for a competent pupil IRAB3</u>

Task set 2
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 4 1 0 0 0 0
Models which explain behaviour for this Task set are
(SOLVE SIMPLIFY FIN FIN2)

Task set 3
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 5 0 0 0 0 0
Models which explain behaviour for this Task set are
(ADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 4
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 5 0 0 0 0 0
Models which explain behaviour for this Task set are
(MULT SOLVE SIMPLIFY FIN FIN2)

Task set 5
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 7 6 0 1 0 0 0
Models which explain behaviour for this Task set are
(XADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 6
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 6 6 0 0 0 0 0
Models which explain behaviour for this Task set are
(NTORHS ADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 7
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 6 0 6 0 0 0 0
Models which explain behaviour for this Task set are
(ADDSUB REARRANGE NTORHS SOLVE SIMPLIFY FIN FIN2)

Task set 8
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 6 0 0 6 0 0 0
Models which explain behaviour for this Task set are NIL

Task set 9
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 4 3 0 0 0 0 1
Models which explain behaviour for this Task set are
(MULT ADDSUB BRA1 SOLVE SIMPLIFY FIN FIN2)

Task set 10
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 4 0 4 0 0 0 0
Models which explain behaviour for this Task set are
(BRA2 XADDSUB MXTOLHS SOLVE SIMPLIFY FIN FIN2)

Overall Result (Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild"):
48 30 11 0 7

Figure 2b Summary of results for a "patchy" pupil AB5

Task set 2
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 5 0 0 0 0 0
Models which explain behaviour for this Task set are
(SOLVE SIMPLIFY FIN FIN2)

Task set 3
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 4 1 0 0 0 0
Models which explain behaviour for this Task set are
(ADDSUB SOLVE SIMPLIFY FIN FIN2)

Task set 4
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 5 3 1 0 0 0 1
Models which explain behaviour for this Task set are
(MULT SOLVE SIMPLIFY FIN FIN2)

Task set 5
Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild": 7 2 0 2 0 0 3
Models which explain behaviour for this Task set are NIL

Overall Result (Total, Correct, Mal, "Parsing", New Mal, Clerical and "Wild"):
22 14 2 1 0 0 5

Figure 2c Summary of results for a poor pupil AB18

| Task Set | Rules Focussed On | Typical Task |
|---|---|---|
| 2 | SOLVE | 5 * X = 7 |
| 3 | ADDSUB | 3 * X = 5 + 3 |
| 4 | MULT | 5 * X = 2 * 2 |
| 5 | XADDSUB | 2 * X + 3 * X = 10 |
| 6 | NTORHS | 2 * X + 4 = 16 |
| 7 | REARRANGE | 4 + 2 * X = 16 |
| 8 | XTOLHS | 4 * X = 2 * X + 3 |
| 9 | BRA1 | 2 * X = 5 * <3 + 1> |
| 10 | BRA2 | 6 * X = 2 * <2 * X + 3> |
| 11 | ADDSUB/MULT | 2 * X = 2 + 4 * 6 |
| 12 | ADDSUB/XADDSUB | 2 + 3 * X + 4 * X = 16 |
| 13 | ADDSUB/BRA2 | 15 * X = 2 + 4 * <2 * X + 3> |
| 14 | MULT/XADDSUB | 2 * 4 * X + 2 * X = 12 |
| 15 | MULT/BRA2 | 14 * X = 2 * 3 * <2 * X + 3> |

Figure 3 Typical Task for each Task set and which rule(s) are being focussed on

Figure 4

Protocols from which new mal rules were inferred. (Note the teacher specified the way in which the X-coefficient should be represented. NOTE too that some of the protocols are not totally consistent: the Investigator has given the mal-rules which summarizes the pupil's behaviour on the majority of the tasks).

```
Task set 5
Task is (2 • X + 4 • X = 12) Solution is (1 • X = 3)
Task is (2 • X + 3 • X = 10) Solution is (2 • X = 10 - 2 - 3)
Task is (3 • X + 2 • X = 11) Solution is (1 • X = 6 // 2)
Task is (2 • X + 6 • X = 10) Solution is (1 • X = 1)
Task is (3 • X + 4 • X = 9) Solution is (1 • X = 1)
Task is (2 • X + 4 • X = 3) Solution is (1 • X = -3 // 2)
Task is (4 • X + 2 • X = 4) Solution is (8 • X = 4)
```

Figure 4a. Protocol apparently showing M • X => M + X (pupil AB17).

```
Task set 6
Task is (2 • X + 4 = 16) Solution is (1 • X = 8 // 3)
Task is (2 • X + 3 = 9) Solution is (1 • X = 9 // 5)
Task is (3 • X - 4 = 6) Solution is (1 • X = -6)
Task is (2 • X + 5 = 10) Solution is (1 • X = 10 // 7)
Task is (6 • X + 4 = 6) Solution is (1 • X = 3 // 5)
Task is (5 • X + 2 = 5) Solution is (1 • X = 5 // 7)
```

Figure 4b. Protocol apparently showing M • X + N => (M + N) • X (pupil AB20).

```
Task set 7
Task is (4 + 2 • X = 16) Solution is (1 • X = 8)
Task is (2 + 4 • X = 14) Solution is (1 • X = 6)
Task is (3 + 5 • X = 11) Solution is (1 • X = -4)
Task is (4 + 6 • X = 11) Solution is (1 • X = -13)
Task is (4 + 5 • X = 6) Solution is (1 • X = -14)
Task is (5 + 2 • X = 8) Solution is (1 • X = -2)
```

Figure 4c. Protocol apparently showing M + N • X => M • N + X (pupil AB3).

```
Task set 8
Task is (4 • X = 2 • X + 6) Solution is (1 • X = 6)
Task is (3 • X = 2 • X + 5) Solution is (1 • X = 5)
Task is (3 • X = -2 • X + 7) Solution is (1 • X = 4)
Task is (4 • X = 2 • X + 3) Solution is (1 • X = 9 // 2)
Task is (4 • X = -2 • X + 8) Solution is (1 • X = 5)
Task is (6 • X = 2 • X + 3) Solution is (1 • X = 11 // 2)
```

Figure 4d. Protocol apparently showing MX = NX + P => X + X = M + N + P (pupil AB1).

```
Task set 7
Task is (4 + 2 • X = 16) Solution is (1 • X = 2)
Task is (2 + 4 • X = 14) Solution is (1 • X = 4 // 2)
Task is (3 + 5 • X = 11) Solution is (1 • X = 5 // 3)
Task is (4 + 6 • X = 11) Solution is (1 • X = 6 // 4)
Task is (4 + 5 • X = 6) Solution is (1 • X = -5 // 4)
Task is (5 + 2 • X = 8) Solution is (1 • X = 8 // 2)
```

Figure 4e. Protocol apparently showing M + N • X = P => N • X = M (pupil AB7).

1a. $M \cdot X + N \cdot X \Rightarrow M \cdot X \cdot N$
$\Rightarrow M \cdot X + N$
$\Rightarrow M + X + N + X$

1b. $M \cdot X \Rightarrow M + X$

2. $M + N \cdot X = \Rightarrow M \cdot N + X =$
$\Rightarrow M + N + X =$

3. $M \cdot X + N = \Rightarrow M + X + N =$
$\Rightarrow M \cdot X \cdot N =$
$\Rightarrow (M + N) \cdot X =$

4. $M \cdot X = N \cdot P \Rightarrow X = M$

5. $M \cdot X = N \cdot X + P \Rightarrow X + X = M + N + P$

6. $M \cdot X = N + P \Rightarrow M \cdot X = N$
$\Rightarrow M \cdot X = P$

---

7. $M \cdot X = N \Rightarrow X = N$

8. $M \cdot X = N \Rightarrow X = (N/F)/M$
$\Rightarrow X = N/(M/F)$

9. $M \cdot <N \cdot X + P> \Rightarrow M \cdot X + M \cdot P$

Figure 5. Summary of major new mal-rules encountered in recent experiments. Where sets, 1 to 6, give "parsing" mal-rules and 7-9 additional manipulative mal-rules, and where F in mal-rule 8 represents a common factor.

<u>Figure 6.</u>
Protocol for a pupil who has a number of "Alternative Methods".

Pupil AB17 on task set 6.

a) The task given was : $2 \bullet X + 3 = 9$

pupil writes
1) $2X = 9 - 3$
2) $X = 3$

Interviewer writes $X = 9 - 3 + 2$
Interviewer: says could you say whether you step 1) above or what I've just written is correct.

Pupil says he really could not.

b) The task given was : $2 \bullet X + 4 = 16$

Pupil writes
1) $2X = 16 - 4$
2) $2X = 12$
3) $X = 6$

Interviewer writes $X = 16 - 4 - 2$
Interviewer: says could you say whether your step 1) above or what I've just written is correct.

Pupil says his 1) probably is.

Interviewer says : can you say why?

Pupil : I'm afraid not.

Interviewer: Now look back at the last example, there I suggested a slightly different method there. Would that be possible here?

Pupil : That's right, it would.

Interviewer : Which of these do you think is correct?

Pupil : Really not sure. I often have a lot of methods to choose between, which makes it pretty confusing. I sometimes have as many as 5 or 6.

[And so this conversation continues. After this point the pupil voluntarily offers 2 or 3 solutions to each task, as in the next task.]

c) The task given was $4 \bullet X = 2 \bullet X + 6$

Pupil writes
1) $X = 2 - 4 + 6$
$X = 4$

Then suggests the following reworking:

1) $4X = 2X + 6$
2) $4X = 8X$
Then Quits.

Interviewer : Which solution do you think is right?

Pupil : Oh, I'm not really sure.

Interviewer : If you were a betting man, which would you put your money on?

Pupil : Probably the first.

Figure 7
Three examples of very consistently used MAL-RULES.

I) Pupil AB-11 on task set 7.
a) The task given was : 4 + 2 • X = 16

Pupil writes 1) 6X = 16
2) X = 2.6666

b) The task given was : 2 + 4 • X = 14

Pupil writes 1) 6 • X = 14
2) X = 2.333

c) The task given was : 3 + 5 • X = 11

Pupil writes 1) 8 • X = 11
(and is told she can leave it in that form)
d) The task given was : 5 - 3 • X = 11

Pupil writes 1) 2 • X = 11
(and is told she can leave it in that form)

II) Pupil AB-17 on task set 5
a) The task given was : 2 • X + 4 • X = 12

Pupil writes 1) • X • X = 12 - 2 - 4
2) X •• 2 = 6
3) X = ROOT 6
b) The task given was : 2 • X + 3 • X = 10

Pupil writes 1) • X • X = 10 - 2 - 3
2) X •• 2 = 5
(and is told he can leave it in that form)
c) The task given was : 2 • X - 3 • X = 10

Pupil writes 1) • X • X = 10 - 2 + 3
2) X •• 2 = 11
(and is told he can leave it in that form)

III) Pupil AB-18 on task sets 5, 6, 7 and 8.
a) The task given was : 2 • X + 3 • X = 10

Pupil writes 1) 2 • X = 10 - 2 - 3
2) 2 • X = 5
3) X = 2.5
b) The task given was : 3 • X + 5 • X = 24

Pupil writes 1) X + X = 24 - 3 - 5
2) 2 • X = 16
3) X = 8
c) The task given was : 3 • X + 4 • X + 5 • X = 24

Pupil writes 1) X + X + X = 24 - 3 - 4 - 5
2) 3 • X = 12
3) X = 4
d) The task given was : 2 • X + 4 = 20

Pupil writes 1) X = 20 - 2 - 4
2) X = 14

e) The task given was : $3 \cdot X + 5 = 7$

Pupil writes 1) $X = 7 - 3 - 5$
2) $X = -1$

f) The task given was : $4 + 3 \cdot X = 14$

Pupil writes 1) $X = 14 - 3 - 4$
2) $X = 7$

g) The task given was : $5 + 6 \cdot X = 20$

Pupil writes 1) $X = 20 - 5 - 6$
2) $X = 9$

h) The task given was : $4 \cdot X = 2 \cdot X + 6$

Pupil writes 1) $2 \cdot X = -4 + 2 + 6$
2) $2 \cdot X = 4$
3) $X = 2$

Pupil then wrote 1) $X - X = 2 + 6 - 4$
2) $0 = 4$
and QUITS.

i) The task given was : $5 \cdot X = 3 \cdot X + 6$

Pupil writes 1) $0 = 4$
and QUITS.

Figure 8
Protocols with the Enhanced LMS.

I. For Pupil AB-11 on task-set 7.

```
Task is (4 + 3 * X = 19) Solution is (7 * X = 19)
NB Previous State was: (4 + 3 * X = 19)
Behaves as model(s):
(ADDSUB REARRANGE NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)


Task is (2 + 4 * X = 14) Solution is (6 * X = 14)
NB Previous State was: (2 + 4 * X = 14)
Behaves as model(s):
 (ADDSUB REARRANGE NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Task is (3 + 5 * X = 11) Solution is (8 * X = 11)
NB Previous State was: (3 + 5 * X = 11)
Behaves as model(s):
 (ADDSUB REARRANGE NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Task is (4 + 5 * X = 11) Solution is (9 * X = 11)
NB Previous State was: (4 + 5 * X = 11)
Behaves as model(s):
 (ADDSUB REARRANGE NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Task is (4 + 5 * X = 6) Solution is (9 * X = 6)
NB Previous State was: (4 + 5 * X = 6)
Behaves as model(s):
 (ADDSUB REARRANGE NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Task is (3 + 7 * X = 8) Solution is (10 * X = 8)
NB Previous State was: (3 + 7 * X = 8)
Behaves as model(s):
 (ADDSUB REARRANGE NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Total, correct, unanswered, Mal, Perturb and Wild: 6 0 0 6 0 0

0 RIGHT OUT OF 6
Modelling Vector (0 600 0 0 0 0 0 0 0 0 0 0 0 0)
Models which explain overall behaviour are:
 (ADDSUB REARRANGE NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Overall Result: (6 0 0 6 0 0)
```

II. Pupil AB-18 on task sets 5, 6, 7 and 8.

Task set 5

Task is (3 * X + 4 * X = 21) Solution is (2 * X = 21 - 7)
NB Previous State was: (3 * X + 4 * X = 21)
Behaves as model(s):
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)
(MP1 XADDSUB SOLVE0 M3SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB MNTORHS REARRANGE)

Task is (6 * X + 3 * X = 36) Solution is (2 * X = 36 - 6 - 3)
NB Previous State was: (6 * X + 3 * X = 36)
Behaves as model(s):
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (3 * X + 4 * X = 98) Solution is (2 * X = 98 - 7)
NB Previous State was: (3 * X + 4 * X = 98)
Behaves as model(s):
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (4 * X - 3 * X = 72) Solution is (2 * X = 72 - 4 + 3)
NB Previous State was: (4 * X - 3 * X = 72)
Behaves as model(s):
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (4 * X + 3 * X = -14) Solution is (2 * X = -14 - 7)
NB Previous State was: (4 * X + 3 * X = -14)
Behaves as model(s):
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (3 * X + 4 * X = -21) Solution is (2 * X = -21 - 7)
NB Previous State was: (3 * X + 4 * X = -21)
Behaves as model(s):
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)
(MP1 XADDSUB SOLVE0 M2SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB MNTORHS REARRANGE)

Task is (3 * X + 4 * X = 11) Solution is (2 * X = 11 - 7)
NB Previous State was: (3 * X + 4 * X = 11)
Behaves as model(s):
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (3 * X + 4 * X = 3) Solution is (2 * X = 3 - 7)
NB Previous State was: (3 * X + 4 * X = 3)
Behaves as model(s):
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Total, correct, unanswered, Mal, Perturb and Wild: 8 0 0 8 0 0

0 RIGHT OUT OF 8
Modelling Vector (0 0 0 0 0 0 0 0 700 0 0 0 0 0 50 50)
Models which explain overall behaviour are:
(MP1 XADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task set 6

```
Task is (3 * X + 4 = 19) Solution is (1 * X = 19 - 7)
NB Previous State was: (3 * X + 4 = 19)
Behaves as model(s):
 (MP0 NTORHS ADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN REARRANGE)

Task is (2 * X + 3 = 9) Solution is (1 * X = 9 - 5)
NB Previous State was: (2 * X + 3 = 9)
Behaves as model(s):
 (MP0 NTORHS ADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN REARRANGE)

Task is (3 * X - 4 = 6) Solution is (1 * X = 6 + 4 - 3)
NB Previous State was: (3 * X - 4 = 6)
Behaves as model(s):
 (MP0 NTORHS ADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN REARRANGE)

Task is (2 * X + 5 = 10) Solution is (1 * X = 10 - 7)
NB Previous State was: (2 * X + 5 = 10)
Behaves as model(s):
 (MP0 NTORHS ADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN REARRANGE)

Task is (6 * X + 4 = 6) Solution is (1 * X = 6 - 4 - 6)
NB Previous State was: (6 * X + 4 = 6)
Behaves as model(s):
 (MP0 NTORHS ADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN REARRANGE)

Task is (5 * X + 2 = 5) Solution is (1 * X = 5 - 2 - 5)
NB Previous State was: (5 * X - 2 = 5)
Behaves as model(s):
 (MP0 NTORHS ADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN REARRANGE)

Total, correct, unanswered, Mal, Perturb and Wild: 6 0 0 6 0 0

0 RIGHT OUT OF 6
Modelling Vector (0 0 0 0 0 0 0 0 600 0 0 0 0 0 0 0)
Models which explain overall behaviour are:
(MP0 NTORHS ADDSUB SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN REARRANGE)
```

Task set 7

```
Task is (4 + 3 * X = 19) Solution is (1 * X = 19 - 7)
NB Previous State was: (4 + 3 * X = 19)
Behaves as model(s):
 (MP3 REARRANGE ADDSUB NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Task is (2 + 4 * X = 14) Solution is (1 * X = 14 - 6)
NB Previous State was: (2 + 4 * X = 14)
Behaves as model(s):
 (MP3 REARRANGE ADDSUB NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Task is (3 + 5 * X = 11) Solution is (1 * X = 11 - 8)
NB Previous State was: (3 + 5 * X = 11)
Behaves as model(s):
 (MP3 REARRANGE ADDSUB NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)
```

```
Task is (4 + 5 * X = 11) Solution is (1 * X = 11 - 9)
NB Previous State was: (4 + 5 * X = 11)
Behaves as model(s):
 (MP3 REARRANGE ADDSUB NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Task is (4 + 5 * X = 6) Solution is (1 * X = 6 - 9)
NB Previous State was: (4 + 5 * X = 6)
Behaves as model(s):
 (MP3 REARRANGE ADDSUB NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Task is (3 + 7 * X = 8) Solution is (1 * X = 8 - 10)
NB Previous State was: (3 + 7 * X = 8)
Behaves as model(s):
 (MP3 REARRANGE ADDSUB NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

Total, correct, unanswered, Mal, Perturb and Wild: 6 0 0 6 0 0

0 RIGHT OUT OF 6
Modelling Vector (0 0 0 0 0 0 0 0 0 0 0 0 600 0)
Models which explain overall behaviour are
 (MP3 REARRANGE ADDSUB NTORHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN)

                                    Task set 8
Task is (4 * X = 3 * X + 6) Solution is (2 * X = 3 + 6 - 4)
NB Previous State was: (4 * X = 3 * X + 6)
Behaves as model(s):
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (3 * X = 2 * X + 5) Solution is (2 * X = 2 + 5 - 3)
NB Previous State was: (3 * X = 2 * X + 5)
Behaves as model(s):
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (6 * X = 3 * X + 33) Solution is (2 * X = 33 + 3 - 6)
NB Previous State was: (6 * X = 3 * X + 33)
Behaves as model(s):
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (5 * X = 4 * X + 19) Solution is (2 * X = 19 + 4 - 5)
NB Previous State was: (5 * X = 4 * X + 19)
Behaves as model(s):
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Task is (4 * X = 6 * X + 6) Solution is (2 * X = 12 - 4)
NB Previous State was: (4 * X = 6 * X + 6)
Behaves as model(s):
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)
 (MP41 XADDSUB XTOLHS SOLVE0 M3SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB MNTORHS REARRANGE)

Task is (3 * X = 6 * X + 12) Solution is (2 * X = 6 + 12 - 3)
NB Previous State was: (3 * X = 6 * X + 12)
Behaves as model(s):
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)
```

```
Task is (3 * X = -2 * X + 7) Solution is (2 * X = -2 + 7 - 3)
NB Previous State was: (3 * X = -2 * X + 7)
Behaves as model(s):
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)
(MP41 XADDSUB XTOLHS SOLVE0 M1SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)


Task is (6 * X = 3 * X + 2) Solution is (2 * X = 5 - 6)
NB Previous State was: (6 * X = 3 * X + 2)
Behaves as model(s):
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)


Total, correct, unanswered, Mal, Perturb and Wild: 8 0 0 8 0 0

0 RIGHT OUT OF 8
Modelling Vector (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 700 50 0 0 0 0 0 50
0 0)
Models which explain overall behaviour are:
 (MP41 XADDSUB XTOLHS SOLVE0 SOLVE SOLVE2 SIMPLIFY SGNS FIN ADDSUB NTORHS REARRANGE)

Overall Result: (28 0 0 28 0 0)
```

FIGURE 9

<u>Some MAL Parsing Rules noted in the experiment and now</u>
used by LMS.II

| RULE | CONDITION/ACTION |
|---|---|
| MP0{1b} | (MPARSE M•X rhs)<br>(SHD 1•X SM rhs) |
| MP1{1a} | (MPARSE M•X +!- N•X = rhs)<br>(SHD X + X + M +!- N = rhs) |
| MP21{3a} | (MPARSE M•X +!- N = rhs)<br>(SHD X + M +!- N = rhs) |
| MP22{3c} | (MPARSE M•X +!- N = rhs)<br>(SHD [M +!- N]• X = rhs) |
| MP23 | (MPARSE M•X +!- N = rhs)<br>(SHD X +!- N = rhs SM) |
| MP3{2} | (MPARSE M +!- N•X = rhs)<br>(SHD X + M +!- N = rhs) |
| MP41{5'} | (MPARSE M•X = N•X rhs)<br>(SHD X + X + M = N rhs) |
| MP42{5''} | (MPARSE M•X = N•X rhs)<br>(SHD X + M = N rhs) |

Where M, N, P are integers, lhs and rhs are arbitary strings (including null). MPARSE is a token placed in the task-state which allows only one mal parsing rule to fire per task, and SM stands for the <u>signed</u> expression which corresponds to the integer M. The figures in braces give a cross reference to (the mal-rules of) Figure 5.
(NB MP21 is subsumed by MP0 and so is NOT included in the actual data base).

Copyright © 1985 by KSL and
Comtex Scientific Corporation

FILMED FROM BEST AVAILABLE COPY