Acquiring Arithmetic Knowledge in a Computer-Based Learning Environment for Long Division

Kees van Putten
Department of Education, Leiden University
The Netherlands, E-mail: putten@rulfsw.leidenuniv.nl

Karel Hurts
Department of Psychology, Leiden University
The Netherlands, E-mail: hurts@rulfsw.leidenuniv.nl

The acquisition of long division knowledge and skills in primary education takes a lot of time. In The Netherlands the final achievements of about 25% of the students are clearly below the expected level [Wijnstra 1988]. Perhaps it is quite a relief for them to read in Allan Turing's biography that one of the first computer programs ever written, an algorithm for long division, was inefficient and contained an error [Hodges 1983]. And, more recently, the Pentium processor was also found to contain a failing division algorithm [Halfhill 1995].

Some educators would like to abandon the topic of long division from the school agenda and to leave division problems to the pocket calculator. However, others are in search of new didactics and tools for division arithmetic. In this renewal of mathematics education the objective of understanding while solving arithmetic problems is generally considered to be more important than the mere acquisition of standard procedures.

In a previous study the prototype of a computer-based learning environment for insightful and effective teaching of long division knowledge and skills was described, along with a pilot study conducted to obtain a first evaluation of the effectiveness of the prototype [Hurts 1995]. The prototype was mainly designed to foster goal-oriented exploration of the number space between divisor and dividend; in particular, division was to be carried out by repeated subtraction of the divisor from the dividend, with the system imposing only minimum constraints on the way the solution was achieved. Another feature of the prototype was that the computer was assigned lower level subtasks such as subtraction and multiplication, freeing the student's attention for the division subtask (choosing proper multiples of the divisor to be subtracted). The main result of this pilot study was that students learned to solve division problems quicker and in a better way using the prototype. However, questions were raised with respect to the robustness of this result (only 6 students were studied) and with respect to the ability of students to show improved performance in the absence of the learning environment.

This paper describes a more detailed evaluation of the same prototype using a larger sample of students. In addition, this paper focuses on the effect of constraining the solution behavior of the students; specifically, a group of students that was constrained to subtract multiples of the divisor that would not be too small was compared with a group that was not constrained in this way, as was the case in the pilot study. A final objective of the present study was to assess the ability of students to transfer the knowledge and skills learned while working with the learning environment to a more traditional situation, where problems had to be solved using pencil and paper only.

Theoretical Framework

According to various accounts [Petitto 1985][Rivera & Smith 1988][Van Putten & De Ronde 1990] the standard procedure for expert long division seems to be complicated for at least three reasons. First, in contrast with written multicolumn addition, subtraction, and multiplication, in long division the "columns" of the dividend have to be treated from the left to the right, enabling remaining hundreds, for example, to be regrouped with the tens. Second, usually the division task proper cannot be solved just by retrieval of number facts, especially for large numbers (i.e., divisors above 10 and/or dividends above 10*divisor). Third, the solution notation in standard long division is highly abbreviated and therefore very efficient, but it obscures the meaning of the procedure and of the intermediate results and it does not allow probes [see Fig. 1].

The alternative approach we considered was based on the didactic method called *progressive schematization* [Treffers 1983][Gravemeijer 1994]. It emphasizes the concept of division as repeated subtraction of the divisor from the dividend. Instead of following an expert way of dividing, students are allowed to start at this novice level of problem solving, which is effective even for large numbers, although rather inefficient. The task of the instructor is to make them discover more efficient solutions (e.g., instead of 1*17, directly taking 2*17, or 5*17, from 135).

```
17/13561\6?     17/13561\8      17/13561\797   17/13561\
   102             136             119           11900       700
   336             ???             166            1661
                                    153            1530        90
                                     131            131
                                     119            119       + 7
                                      12             12       797

   (a)             (b)             (c)                 (d)
```

Figure 1: Examples of long divisions in standard notation (a-b probes; c correct) and in the alternative unabbreviated form (d).

Starting from the above didactic approach we can distinguish the following types of long division knowledge and skills: (1) *Factual Knowledge* (such as multiplication tables), (2) *Conceptual Knowledge* (such as understanding the part-whole relation of divisor and dividend and the goal concept, i.e., the general definition of the relation between divisor and subdividend when the problem is solved), and (3) *Procedural Knowledge,* such as the ability to multiply, but also strategic knowledge like deciding whether a dividend is greater or smaller than 10*divisor, or falls between 5* and 10*divisor or not.

Design of the Learning Environment and Pilot Study Results

The central idea of division as a process of repeated subtraction by itself was not enough to help us design a learning environment. In fact, many lower level decisions regarding the appearance of the interface and the style of communication between student and system are not addressed at all by the literature on progressive schematisation. Therefore, our design method was based on the following additional ideas [see Hurts 1995]:

1. Learning through *guided exploration.* This idea was accomplished by the following four elements: (a) Formulating the task as a traversal of a state space by applying certain operators; (b) Goal-orientation by requiring students to explicitly confirm goal achievement, and requiring the student to never subtract a multiple of the divisor exceeding the present dividend; (c) Transparency and reversibility of operators; and (d) Delegation of lower-level subtasks to the machine.
2. Presenting the student with an environment that initially does not support her with respect to strategic knowledge (an environment with a so-called *minimal interface*) and using that environment to collect data regarding particular strategic difficulties students experience (iterative design).

The result of the first design attempt was a learning environment (the "Long Division Machine" or LDM) that consists of a separate window for constructing multiples ("applying operators" in problem space terms) and another window for presenting the result of subtraction in a long division solution path [see Fig. 2]. The student decides whether (further) to multiply the divisor or to subtract or to stop (in case the problem is solved). The student can also use a "Reset" button to cancel a multiple that is judged too large (overshooting) or too small (undershooting). On the other hand, the system implements the operators for multiplication of the divisor, subtraction from the dividend, and addition of partial quotients. Moreover, the system shows the result of multiplying the divisor with the currently constructed factor size in a *look-ahead window* (left-hand side of Figure 2), thereby allowing the student to assess the amount to be subtracted before actually subtracting it. Finally, the system also warns the student in case (s)he is about to "overshoot", that is, subtract a multiple of the divisor exceeding the present dividend.

The left-most window presents a number of buttons to manipulate the divisor, e.g., for multiplying the divisor with 1 or with 10. Multiples of 1*, 10*divisor, etcetera, can be constructed by repeatedly clicking (up to 9 times) the same button. Note that multiples that fall in between 10*divisor, 20*divisor, etcetera, can not be constructed directly.

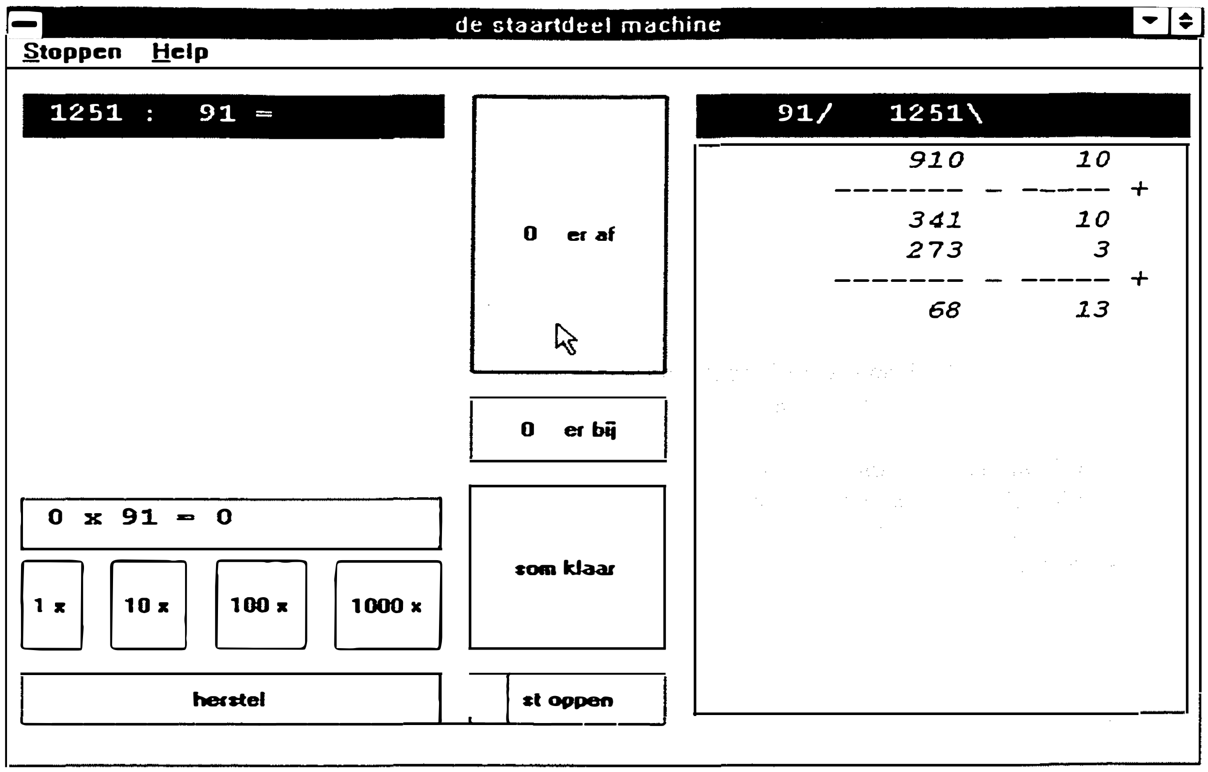

Figure 2: Snapshot of the Division Machine interface (additive button version)."Stoppen" means "Leave Program", "Er Bij" means "Add", "Er Af" means "Subtract", "Som Klaar" means "Problem Finished", and "Herstel" means "Reset".

A ***Pilot Study*** was conducted with the LDM of Figure 2 in order to obtain a first assessment of the usability and learnability of the system [Hurts 1995]. Results of 6 midschool students (age 13-14), who solved 20 problems each, showed that these students were able to learn quickly about the goal concept of long division, because they stated at the right time that the problem was solved and because they were able to recognize and correct (by using the Reset button) overshooting errors before these were detected by the system (undershooting corrections seldom occurred).

For problems of low to intermediate difficulty level, students quickly learned to achieve optimal solutions (in which all multiples of the divisor that were subtracted, represented the largest possible multiple and in which the number of solution steps was minimal), seemingly through a strategy of deliberately overshooting the largest possible multiple in combination with the use of the reset button, allowing them to correct an error if the look-ahead window showed the multiple constructed to be too large. Without this window the solutions probably would have been less optimal and overshooting might have been less frequent. This raises the question of how much the student can transfer from the LDM to a non-supported test situation, which is further addressed by the main experiment.

For difficult problems (large divisors or dividends that are large compared to the divisor) solutions tended to be less optimal (more solution steps than necessary), even though students committed (and corrected) more overshooting errors for these problems. Apparently, the strategy of deliberately overshooting and relying on the look-ahead window did not prevent more suboptimal solutions for these problems from occurring. This raises the question of whether *constraining* the student by keeping him from taking suboptimal solution steps, might be a way to help the student to learn better strategies even for difficult problems [see next section].

Problem Statement

The main experiment was designed to answer the following questions: (1) Can the findings of the pilot study regarding type of solution and problem difficulty be reproduced in a more representative study? (2) What is the effect of passively constraining the student's problem solutions? By putting constraints on the construction of multiples of the divisor, the learning environment does not "allow for more states and operators than are admissible in a problem space in which a particular skill can be developed" [Ippel 1992]. In this way the LDM evaluates the search and solution steps generated by the student, so that learning can occur [Ohlsson & Rees 1991]. (3) Is there any transfer of what is acquired in the learning environment to a paper-and-pencil test of division problems? Is this transfer due to passive guidance?

Methods

Design and procedure: A comparative study was conducted with 2 treatment conditions (working in the LDM learning environment with Constraints-ON versus Constraints-OFF) in a pre/posttest design. Working individually, subjects were given 40 divisions problems altogether in the LDM to be solved during three 20 minutes sessions. These 40 problems consisted of 5 sets of 8 division problems. In the Constraints-ON condition a new constraint was imposed on the problem solution after each 8th problem. The first set of 8 problems had no constraints. In the second set the multiple of the divisor to be subtracted should be greater than 1* if possible. In the third set another constraint was added: greater than 2* if possible. Fourth set: greater than 10* was added. Fifth set: greater than 100* was added [see Fig. 3].

Problem #	0 4 8	12 16	20 24	28 32
Constraints ON	No Constraints	Constraints >1	Constraints > 2, >1	Constraints >10, >2, >1
Constraints OFF	No Constraints	No Constraints	No Constraints	No Constraints
	Problem Set 1	Problem Set 2	Problem Set 3	Problem Set 4

Figure 3: Problem number and type of constraints for both conditions according to problem set. Set 5 is left out because of missing data [see Results].

In both conditions problems became successively more difficult for each set. Before the first and after the last session a pretest and posttest were administered in which long division problems similar to the ones given in the LDM had to be solved only using pencil and paper. Before the first session pupils were instructed in the use of the LDM according to a protocol.

Subjects: 56 primary school children, 8-9 years of age, were randomly assigned to one of the two conditions (ON: 29; OFF: 27). Boys and girls were evenly distributed over the conditions. No significant differences between the conditions on the pretest were found.

Materials: The learning environment (see section "Design" for a description; and see Figure 2) was programmed in Visual-Basic. The program stored a number of variables concerning frequency of button use, number of steps needed to reach a solution, and time needed per problem. The pretest and posttest each contained 22 problems differing only in the numbers used. Each test was scored on 3 variables: (1) time needed to complete the test; (2) division skill, a composite measure for the ability to judge whether 4 given division solutions were complete or not, and for quality and efficiency of solutions on 4 division problems (maximum score 20); (3) estimation skill, measured by 14 estimation items (each scored as correct or incorrect).

Results

Taken over all problems the average student needed 78 seconds to find a solution for a division problem. The average optimal solution consisted of nearly 3 subtraction steps. The average student solution took 5.4 steps, nearly twice as long. In only 14 percent of the problems students used the "Ready" button when in fact the problem was not finished yet. In this respect knowledge about the goal of division seems well established. This is also indicated by the high average number of self-induced resets of constructed multiples of the divisor exceeding the present dividend (1.8). However, on the average nearly once a problem students actually tried to subtract a multiple of the divisor that exceeded the present dividend (resulting in a so-called "forced reset"), thereby indicating incomplete knowledge of the goal.

To see the effect of practice on the student's behavior in the LDM we split each problem set in two equal parts of 4 problems each. Only sets 1 to 4 were used, because of missing data in set 5. Manova's with repeated measures (first-second part) indicated a number of changes: problem solution time decreased significantly within each problem set ($F[4,49]=19.10$ $p<0.01$), number of subtraction steps decreased but not significantly ($F[4,49]=2.3$ $p<0.07$), the number of self-induced resets of multiples exceeding the present dividend decreased significantly ($F[4,49]=4.7$ $p<0.01$), especially within set 1 and 2, and the number of forced resets decreased significantly too ($F[4,49]=5.23$ $p<0.01$), except in set 4 where an increase occurred. These data show, contrary to those of the pilot study, that students are able to learn to optimize their problem solutions, without necessarily committing more overshooting errors, as long as they receive enough practice. The data also show that knowledge of the goal can improve with practice, although for the most difficult problems more practice will be needed.

If we compare the sets with each other (second part of each set with first part of the next set) we can see the effect of increasing the problem difficulty on the student's solutions: the time needed to solve a problem increased, especially from set 3 to set 4 (difficult items); the number of solution steps relative to an optimal solution hardly changed, except from set 3 to set 4 (an increase); the number of self and forced resets increased between all sets, confirming our findings from the pilot study about undershoooting and overshooting.

In the condition with constraints an average number of nearly 0.4 constraints per problem was imposed when students tried to subtract a multiple of the divisor that was considered too small in relation to the present dividend. There was a significant difference between the constraints-ON and the constraints-OFF condition ($F[8,45]=2.26$ $p<0.04$), based on fewer solution steps (4.1 versus 6.8), more self resets (2.1 versus 1.5), and also fewer "empty" subtractions (0.13 versus 0.39) in the ON condition. However, the number of forced resets and time needed to solve a problem did not contribute to this difference. This indicates that a relatively small number of constraints was enough to stimulate the construction of more optimal multiples of the divisor, resulting in shorter solution paths that however were obtained at the cost of more resetting behavior.

The last question concerns whether there was any transfer of what was learned in the LDM to a paper-and-pencil test which the student had to complete without the aid of the LDM. For both conditions, constraints-ON and constraints-OFF, there was a significant pre-posttest difference (within-subjects effect: $F[3,53]=12.17$ $p<0.001$), based mainly on shorter solution times ($r=0.90$) and on somewhat better solutions ($r= -0.42$) on the posttest, but not on better insight as measured by the estimation items. However, there was no significant effect of Constraints (between-subjects: $F[3,50]=1.65$ $p<0.19$) on posttest results. Thus, only a general transfer effect could be demonstrated which was not stronger for the condition with the constraints on.

Conclusions

The empirical results showed that students were able to work with this learning environment and to learn from it as measured by the quality and efficiency of their solution paths and their likelihood to commit and correct undershooting and overshooting errors. The pilot study showed that they learned to explore and construct (near-)optimal multiples of the divisor rather spontaneously and quickly. Constraining problem solving with the purpose of avoiding certain undershooting solutions resulted in shorter solution paths and more self-corrected overshooting errors. These effects illustrate that working in a constrained learning environment can influence behavior in the direction of more optimal performance, but only at the cost of overshooting errors. Moreover, constraining did not produce a better overall learning result as measured by posttest performance. The last finding suggests that if the learning environment takes over some lower level subtasks from the student (multiplication, subtraction), and shows the student the result of multiplications in the look-ahead window, the student's learning may not transfer to non-supported situations. Therefore, the student should perhaps be encouraged to use cognitively feasible strategies that can sustain learning even in the absence of the support built into the interface. This is an issue for future research. Finally, more research is also needed to experimentally compare the use of the Long Division Machine with that of an alternative (e.g., traditional) didactic method for teaching long division.

References

[Gravemeijer 1994] Gravemeijer, K.P.E. (1994). Developing realistic mathematics education. PhD Thesis, Utrecht University.

[Halfhill 1995] Halfhill, T.R. (1995). The Truth behind the Pentium Bug. Byte, March, 163-164.

[Hodges 1983] Hodges, A. (1983). The Enigma: Allen Turing. Vintage.

[Hurts 1995] Hurts, C.M.M. (1995). Supporting cognitive learning through guided exploration: the case of long division 134-141. In D. Jonassen & G. McCalla (eds.), Proceedings of the Second International Conference on Computers in Education, Singapore. Charlottesville, VA: AACE.

[Ippel 1992] Ippel, M.J. (1992). Discovery learning of addition strategies for two-digit numbers in a computer-based microworld. Paper AERA 1992, San Francisco. Centre for the Study of Education and Instruction, Leiden University.

[Ohlsson & Rees 1991] Ohlsson, S., & Rees, E. (1991). The function of conceptual understanding in the learning of arithmetic procedures. Cognition and Instruction, 8 (2), 103-179.

[Petitto 1985] Petitto, A.L. (1985). Division of labor: procedural learning in teacher-led small groups. Cognition & Instruction, 2, 233-270.

[Rivera & Smith 1988] Rivera, D., & Smith, D.D. (1988). Using a demonstration strategy to teach midschool students with learning disabilities how to compute long division. Journal of Learning Disabilities, 21 (2), 77-81.

[Spoor 1995] Spoor, J. (1995). De Staartdeelmachine (The Long Division Machine). Doctoral Thesis, Department of Education, Leiden University.

[Treffers 1983] Treffers, A. (1983). Geïntegreerd cijferen volgens progressieve schematisering (An integrated approach to written arithmetic according to the principle of progressive schematization). Pedagogische Studiën, 60, 351-362.

[Van Putten & De Ronde 1990] Van Putten, C.M., & De Ronde, M.A. (1990). A production system explaining errors in long division of learning-disabled students. In J.M. Pieters, K. Breuer, & P.R.J. Simons, Learning environments (163-175). Berlin: Springer.

[Wijnstra 1988] Wijnstra, J.M. (1988). Balans van het rekenonderwijs in de basisschool: uitkomsten van de eerste rekenpeiling medio en einde basisonderwijs (Results of arithmetic education halfway and at the end of primary school). Arnhem: CITO.

Acknowledgements

Jos van Meegen developed a first prototype of the Long Division Machine which was used in the pilot study. Jaap Spoor further developed the LDM and, assisted by Sannah Langerak, collected and analyzed the data of the main experiment for his master's thesis.

Abstract: This paper reports on the acquisition of arithmetic knowledge in successive stages of expertise, and on the types of instruction that are needed to make that learning possible. The domain is that of integer arithmetic and the target skills and concepts of the learning system relate to long division with remainder.

We studied students' learning performance using a computer-based learning environment representing division as repeated subtraction from the dividend and facilitating the exploration and construction of multiples of the divisor to be subtracted. The learning environment also took over the subtasks of multiplication, addition and subtraction from the student. In particular, we studied the extent to which students were constrained in their freedom to construct solution paths and the effect of this factor on learning performance.

Results showed that students profited from this learning environment, but the more optimal solutions reached under constrained problem solving could not be demonstrated in a pre-post test comparison using only paper-and-pencil type problems. It appears that this performance can only be reached if the learning environment assists the student with lower-level subtasks and that more advanced strategic knowledge is required in order to reach the same level of performance without this support.