BEYOND LOCAL CONCEPTUAL CONNECTIONS: META-KNOWLEDGE ABOUT PROCEDURES

IRIT PELED, ORIT ZASLAVSKY

The never-ending discussion on the relationship between conceptual and procedural knowledge has important implications for both curricular design and instructional decisions. In this article, we aim to add another layer to this discussion by considering types of knowledge of and about procedures that are not usually investigated.

Procedures are often regarded as worthy opportunities for learning, particularly when conceptual knowledge can be attached to them. However, in cases where such connections are hard to make, the need to learn procedures is often questioned. At the same time, procedures that *do* lend themselves to connections to conceptual knowledge are not necessarily connected to sufficiently rich knowledge. In attending to these issues we claim that certain procedure-related types of knowledge have the potential of enriching conceptions of procedures. We focus on analyzing this knowledge in an effort to show that it is worth pursuing and worth considering as part of the school curriculum. Specifically, we propose examining three types of meta-knowledge of procedures:

1. *Decision making knowledge* involving conditions for procedure application;
2. *Procedural structure knowledge* involving a global view of the structure of the procedure and its similarity to other procedures;
3. *Global conceptual knowledge* that suggests a global quantitative view of the procedure, as opposed to (and a complement of) the local conceptual knowledge that is commonly attached to it.

Local connections between procedures and conceptual knowledge

Most school curricula today strive to promote learning with understanding, including the learning of procedures. Currently, teaching and learning algorithms with understanding commonly means making connections between conceptual and procedural knowledge (Hiebert & Carpenter, 1992). The connections that are considered desirable map specific steps of a procedure onto the specific conceptual knowledge on which each step is based. For example, in performing multi-digit subtraction, crossing a number such as 9 in the hundreds' column of the minuend and writing 8 instead of it, together with crossing 0 in the tens' column and writing 10, would be connected to the knowledge that a quantity of a hundred is moved or subtracted from the hundreds' column and added to the tens' column in the form of 10 tens.

Because this knowledge is specific and local, we term such connections *local connections between procedures and conceptual knowledge* (LCPC). The term raises the obvious question whether there is some other non-local procedure-related knowledge worth pursuing. This question will be answered in the second part of the article, where we discuss local knowledge and contrast it with global knowledge involving a more general perspective of a procedure.

To help set the grounds for inquiring beyond LCPC, we first deal with the difficulties that may be encountered in achieving LCPC and the question that emerges when such connections fail. This goal of achieving LCPC has two related difficulties. One has to do with epistemological barriers – that is, the connections may require special efforts regardless of how the topic is taught. The second has to do with pedagogical aspects – that is, teachers may not have the relevant knowledge and tools to make the desired connections.

It is not always possible to determine whether the source of the problem is one or the other. What originally seems to be an epistemological barrier in some countries may be successfully overcome in other countries. In discussing *epistemological obstacles* in the sense used by Brousseau (1997), Radford (1997) talks about the smoother introduction of negative numbers in China as an example of what is considered an epistemological obstacle but is, in his opinion, a socio-cultural source of difficulty. Although we (further on) show differences between countries in coping with a certain subject, we do not use Radford's (1997) socio-cultural perspective. Instead, we observe that the different pedagogical solutions involve extensive efforts and, for practical purposes, can be regarded as evidence of an epistemological difficulty.

An epistemological barrier: the example of the long division algorithm

We turn to the long division algorithm to illustrate what we mean by an epistemological barrier rooted in the algorithm. To begin with, research findings indicate that teachers' knowledge does not include conceptual knowledge that is connected to the long division procedure (Simon, 1993; Zaslavsky, 2003). The fact that this phenomenon is so widely spread indicates that the source of the problem is deeper than 'faulty' teacher education. The result, for children, is that they learn a procedure that "focuses on separate

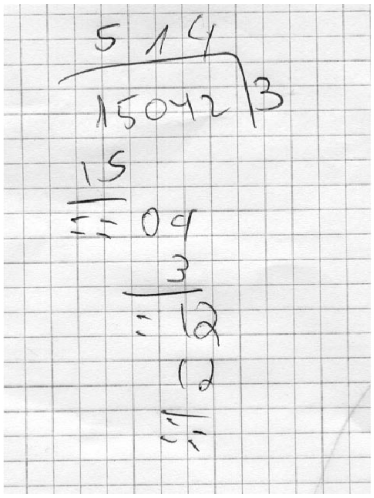

Figure 1: Ittai's initial calculation

digits with their true value implicit" (Anghileri, Beishuizen & van Putten, 2002). These findings together with our following encounter with a mathematically gifted student indicate that there is something inherent to the body of knowledge itself.

Ittai, an outstanding 8th grade student studied in one of the best schools in the country. Ittai recalled that he had learnt the long division algorithm in elementary school and that a lot of time was devoted to practice exercises. When asked to divide the number 15,042 by 3, Ittai used the long division algorithm, and came up with an erroneous result (fig. 1). He noticed that it could not be right, and was rather puzzled.

Interviewer: What is the meaning of the result you got?

Ittai: The meaning is that when I multiply 514 by 3 I get 15,042.

Interviewer: Does this make sense?

Ittai: Not at all, but I don't have, mathematically, a way to find out where I went wrong. All I know is that if this [method] works it is okay. But I don't understand its meaning.

Figure 1 indicates that Ittai does not see the connection between the steps he performed and the decimal structure. He did not develop this kind of understanding in spite of carrying out the algorithm over and over many times. Figure 2 presents Ittai's second attempt to solve the problem, including the explanation he provided.

Ittai: If we do zero by itself it has no meaning, so we need to attach it to what was before. If zero is at the end, it has a meaning. Only at the beginning we can put it down.

Apparently, Ittai had learned how to carry out the long division algorithm technically. Thus, he did not have the tools to handle the problem with which he was faced. Since he is very talented he managed to find a way that solved his current problem by reorganizing the first stage that seemed to cause the problem. However, it seems that Ittai did not perceive his first action as 150 hundreds divided by 3 to get 50 hundreds (or: of the same unit, whatever it was). If he knew that he would have written the 5 above the 5. This conclusion is supported by his wrong solution to the next problem, presented in Figure 3, where he used the same method to divide 120 by 4, not realizing that the resulting 30 is actually 30 hundreds.

Ittai's failure can be seen as indicating an epistemological problem. Most likely, an outstanding student such as him could have figured out the right knowledge connections on his own. It appears that long division is a procedure that does not lend itself easily to LCPC – that is, to attaching conceptual knowledge to its procedural steps.

Some instructional programs around the world made attempts to deal with this barrier. For example, the Dutch (van Putten, van den Brom-Snijders & Beishuizen, 2005) have solved the problem by using an alternative algorithm that fits their Realistic Mathematics Education (RME) approach. That is, instead of teaching the traditional long division algorithm, the Dutch use repeated subtraction while also promoting progressive mathematization of unorganized

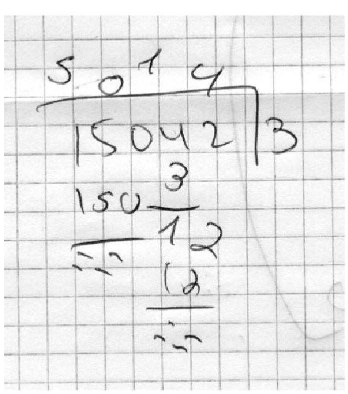

Figure 2: Ittai's second attempt of calculation

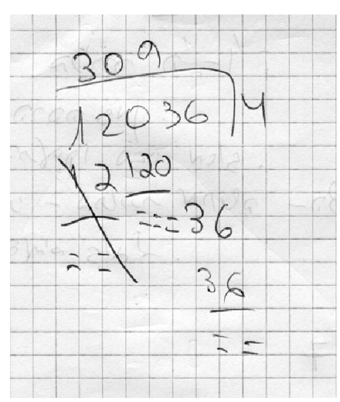

Figure 3: Ittai's application of his misleading method

and informal written calculation into a schematized (formal) procedure. This means that the original traditional algorithm is abandoned in favor of one that is strongly connected to a quantitative interpretation.

In England, however, the use of informal strategies was not a stage towards the development of an alternative procedure, but a detour before eventually teaching the traditional long division algorithm (Anghileiri *et al.,* 2002). The common feature of these two programs, the Dutch and the English, is that no connections were made between children's informal or conceptual knowledge and the traditional algorithm.

Another similarity between these two programs is manifested in their use of quotitive division – that is, the division problem result is viewed as the number of times the divisor is contained in the dividend. [1] According to Fuson (2003) this is also the common meaning offered in the United States, where, in view of the technical nature of the "usual algorithm", some educators resort to the method of taking away "copies of the divisor".

Others that have searched for meaningful instruction of the long division algorithm suggest a partitive meaning (*e.g.,* Gregg & Underwood Gregg, 2007) where the dividend is split into equal parts. This meaning enables LCPC creating good representations for small divisors, but is difficult to apply with larger numbers.

All these programs had to invest great efforts in developing connected and meaningful instruction and their solutions are only partially successful. Some of them fail to connect children's conceptual knowledge to the original algorithm and resort to an alternative algorithm. Other programs succeed in making meaningful connections only within a limited number range. More than indicating cultural differences, we view these facts as demonstrating a common difficulty and as constituting evidence of an epistemological barrier.

The nature of LCPC in alternative algorithms

In view of this barrier, the need to resort to alternative long division algorithms is not surprising. However, while alternative methods offer a certain meaning and attach conceptual knowledge to procedural steps, they are based on different mathematical principles than those on which the standard algorithm is based.

For example, in the process of using the repeated subtraction algorithm to solve a problem such as 256 ÷ 12, the total number, 256, is split into parts that are not constrained by its decimal structure but focus on finding multiples of 12. In contrast, the standard algorithm is column-based, moving from the biggest (leftmost) unit to the smallest one. A meaningful explanation for dividing 25 by 12 and writing the 2 above the 5 would involve relating to the 25 as 25-tens and getting 2-tens as a result of this sub-division. A more abstract explanation would say that 25 of any unit divided by 12 would result in getting 2 of that unit (this general perspective is used further on in discussing the iterative nature of the procedure).

As can be seen, the alternative repeated subtraction procedure for long division involves quantitative knowledge, but does not utilize the power of decimal structure that is exhibited in generalizing an operation in units (*e.g.,* 25 ÷ 12) to an operation in a complex unit (*e.g.*, 25 tens ÷ 12).

Since the choice of an algorithm might depend on the kind of LCPC that "goes with it", it would be interesting to examine the nature of knowledge used in discussing the traditional multi-digit subtraction algorithm. In contrast with long division, this algorithm is considered an easier opportunity for LCPC.

In demonstrating the use of the term 'borrowing' by teachers, Ma (1999) quotes a teacher explaining what she would say while performing the algorithm for 53 – 25. After explaining 'borrowing', and subtracting 5 from 13, the teacher gets to the final steps: "Then you move to the tens column … there are only 4 tens left. You take 20 from 40 and get 20. Put it down in the tens column" (p. 7). As can be seen, in an effort to give a quantitative explanation, the teacher talks about 40 minus 20, missing an opportunity to operate on the new units (tens) – that is, subtracting 2 tens from 4 tens.

Obviously, the use of 40 – 20 is not wrong, but it misses one of the main advantages of this algorithm that allows one to operate, within each column, on the column units (*e.g.,* tens, hundreds). An opportunity to promote thinking in complex unit terms is particularly valuable, since understanding and operating with complex units are necessary conditions for constructing mathematical concepts. For example, Fuson (1990) discusses the importance of perceiving tens and hundreds as new units in constructing place value understanding and performing multi-digit addition and subtraction. Her research suggests that programs that use manipulatives representing such conceptions, or countries (such as China) where number names help promote these conceptions have an advantage in building place value knowledge. Similarly, for multiplication, Steffe (1988) describes the sequence of shifts in unit conception in moving from addition to multiplication, and the crucial role of complex (composite) units in constructing the multiplicative structure.

Having set the grounds for learning some other desirable knowledge, such as knowledge that will promote the construction of new units, we turn to a discussion of new directions that might offer opportunities for that.

Beyond LCPC: three types of meta-procedural knowledge

Two issues have led us to this discussion. The first deals with the question whether algorithms for which it is hard to make LCPC, should not be included in the curriculum. The second involves the limitations of LCPC and the more general question of whether there is some valuable procedure-related knowledge beyond LCPC that is worth learning.

At the risk of being accused of rationalization, we will investigate the latter question in an effort to deal with both issues. The directions we choose to take go towards knowledge that is more algebraic in nature – that is, more abstract, more general, involving meta-knowledge and big ideas. Fortunately, these roads have been paved for us by other researchers.

Some arguments on this issue have emerged following a recent effort by Star (2005) to claim, first, that both conceptual and procedural knowledge might be either shallow or deep and, second, to show that procedural knowledge has merits independently of conceptual knowledge. His main focus is on procedural flexibility manifested by the way skilled problem solvers efficiently generate solution actions that best fit the specific conditions or goals of a problem. Some of Star's claims were countered by Baroody, Feil and Johnson (2007), who stated that although procedural knowledge might be deep to some extent, this extent is limited if it does not get connected to conceptual knowledge.

This last claim might be discouraging in cases where local connections are difficult to make. Yet, Baroody *et al.* also discuss the importance of developing and promoting *Big Ideas* instead of viewing mathematics as isolated procedures, and look for similar underlying structures. Using Star's suggestion that there might be deep procedural knowledge, and noting the nature of valuable knowledge analyzed by Baroody *et al.*, we will look for deep knowledge that can be used to develop a meta-perspective on procedures and to promote connections between them.

This brings us to suggest looking beyond making local connections and considering an additional type of knowledge. We propose a kind of meta-knowledge *about* a procedure that is related to it (in addition to knowing of the procedure, *i.e.,* knowing procedural steps, and to LCPC), which includes *global* aspects underlying a specific procedure or common to a number of procedures. This meta-knowledge includes: (a) knowledge involved in making decisions on the choice of a procedure in a particular situation; (b) structural knowledge of a procedure; and (c) global understanding of the conceptual basis of the procedure.

Meta-knowledge about algorithms: decision making and understanding the role of algorithms

Many different algorithms are taught at school. Some are short, such as the common-denominator method for adding fractions, and some are longer, such as multi-digit number operations. The common feature of all these algorithms is that each of them 'works' independently of the choice of numbers. This does not mean that the use of a given algorithm is similarly beneficial for all cases. A classical example of the disadvantage of blindly applying an algorithm is given by Schoenfeld (1987) in discussing the importance of metacognitive knowledge. Schoenfeld expresses his disappointment from the lack of planning exhibited by his students who ended up wasting much time on a certain procedure for finding an integral.

The evaluation of an algorithm is not trivial. In an effort to promote more appreciation for algorithms, and with a mathematician's perspective, Bass (2003) offers a list of characteristics for evaluating the usefulness of an algorithm. This list includes *accuracy, generality, efficiency, ease of accurate use,* and *transparency,* and it can be used for any algorithm as well as for comparing algorithms. Bass uses this list to discuss traditional algorithms, as he sees them, and to compare them with invented algorithms. According to his analysis, traditional algorithms are efficient but, if taught mechanically, are not transparent and consequently error-prone (*i.e.,* not accurate). His analysis reflects the difficulty involved in such an evaluation – in particular, its dependency not only on the algorithm's features but also on what he calls "an instructional milieu" (p. 327).

The suggestion to compare features of algorithms, such as those listed by Bass, is an important direction in our view. It can be extended to a more general call to strengthen the discussion of knowledge and meta-knowledge *about* algorithms in an effort to construct a more general perspective of algorithms and their role. We believe that learning *about* algorithms should be an integral part of teaching an algorithm. Such meta-knowledge would involve knowing under which conditions to use a given algorithm. For example, in the case described by Schoenfeld (1987), students would be expected to take into account the type of function in choosing a procedure for finding the function's integral, in other cases an efficient choice of a procedure might involve the nature of the given numbers.

Discussions of this nature often have merit beyond learning about the algorithm. Peled and Awawdy-Shahbari (2003) asked 7th and 8th grade students to compare fractions while trying to avoid using the common denominator algorithm. Following an investigation of different number pairs, students realized that the common denominator algorithm was valuable only under certain conditions. Specifically, being encouraged to take into account the number of parts (*e.g.,* 3 in $^3/_7$) and the fraction unit (e.g., $^1/_7$ in $^3/_7$), the common denominator algorithm was found to be efficient when the numbers suggested conflicting information. That is, when one number had more parts and the other had bigger parts (e.g., $^3/_7$ and $^5/_{12}$). Beyond finding the procedure's efficient conditions of applicability, the students also gained more understanding of rational numbers, and some were even able to transfer their knowledge to decimals. In more general terms, the construction of knowledge about the algorithm had a positive effect on related concepts.

In acquiring this meta-perspective on their actions we would like children to understand that an algorithm is an 'algebraic' idea – that is, it offers an 'action program' for a 'general case' or for a machine that has to make many calculations. For a specific case, we expect the child to take into account the specific features of numbers that are involved and to resort to the use of an algorithm following a decision making process that considers its benefits. Such a behavior would indicate that the child has developed an understanding of the role of algorithms and has become a smart user.

It should be noted that teachers can also benefit from discussions of this nature. This need is reflected in findings by Rowland, Thwaites and Huckstep (2003) showing that elementary teachers trying to demonstrate a certain algorithm, often choose a problem with numbers that would have been much easier if solved by another algorithm. For example, they choose 49×4 for illustrating the standard multi-digit multiplication.

Global structural knowledge of a procedure: connections within a procedure and between procedures

With the understanding that meaningful learning and connected knowledge are almost synonymous terms, it is simply natural to suggest that learning *about* procedures will also promote connections *between* procedures. The identification of similarities between procedures can be facilitated by getting away from the specific details of the procedure, viewing it in more abstract terms.

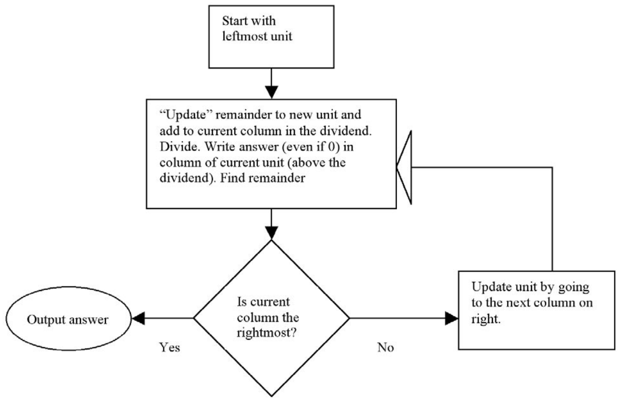

Figure 4: The iterative unit-based structure of the long division algorithm

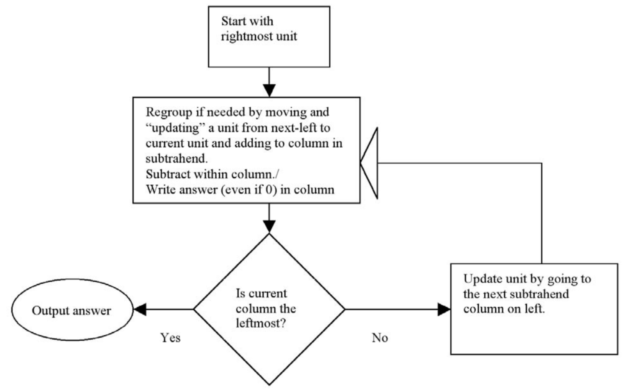

Figure 5: The iterative unit-based structure of the multi-digit subtraction algorithm

One of the rarely discussed, main features of standard basic arithmetic operations algorithms, is their iterative nature. That is, for each algorithm the procedural steps repeat themselves; however, the unit keeps changing (either from smaller to larger unit or from larger to smaller). This iterative characteristic is one that is in common to these algorithms, even though each deals with a different operation.

Looking at the long division algorithm, for example, the procedure can be described by the flowchart presented in Figure 4. As can be seen, the algorithm starts with the leftmost unit big enough for being divided, then proceeds by repeating the same sequence of actions with each move to the next unit on the right (smaller unit). The repeated sequence includes: dividing, writing the result in the column above the current unit (including the case of getting 0), then the remainder (in terms of that same unit) is calculated. If the process has reached the rightmost digit, the result can be read. If not, the process continues by moving to the next unit with the remainder properly added to the unit's digit (the remainder would have the value of 10 times the new unit, therefore the action looks as if it were a concatenation of digits). The same sequence of actions is performed again, and so on.

Although the subtraction algorithm is different, it can also be written as an iterative loop based on a unit/column update in each iteration, as presented in Figure 5. Further abstraction of the two procedures results in the same iterative flowchart. Figure 6 depicts the general abstracted iterative procedure structure that is then presented in Figure 7 as a mediator between procedures that involve different operations. Teachers and students are often unaware of this property not to mention noting that it is a unifying property over different procedures. To encourage the construction of such knowledge, we suggest using special tasks, two of which are presented in Figure 8 and Figure 9.

In each task the grey and white boxes depict digits, but only the white boxes are open for inserting digits. The task is to complete as many digits as possible in the white boxes. A problem solver trying to complete the missing digits, will note that some of them can be completed independently of the fact that the solver does not know what column is operated on – that is, what the unit attached to each digit should

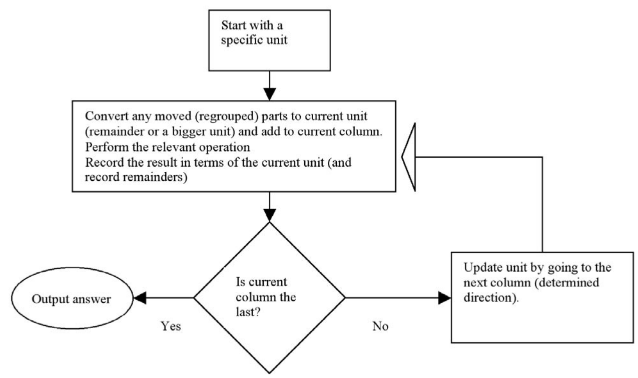

Figure 6: The abstract iterative unit-based structure of the multi-digit algorithms

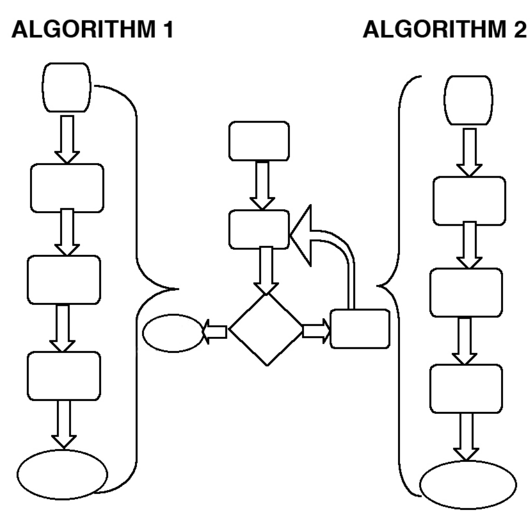

Figure 7: The iterative structure as mediating between two procedures

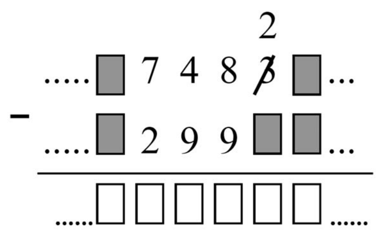

Figure 8: Task depicting multi-digit subtraction iterative structure

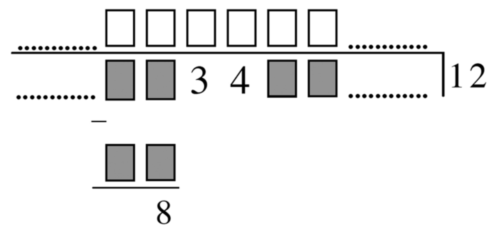

Figure 9: Task depicting long division iterative structure

be (*e.g.,* tens, hundreds) – and merely operates on the basis of the relative value of the digits. In order to perform the task, the solver has to construct the general structure of the relevant procedure. Thus, in the course of working out these activities, the iterative unit-based nature of each algorithm is likely to become apparent.

As to the goal that children will be able to abstract structure and connect procedures, at first sight it may seem unlikely to happen. However, findings by Peled and Segalis (2005) demonstrate that children can abstract and show how this process can be facilitated. Abstraction of procedural steps is done by expressing them (verbally) in general terms. For example, in multi-digit subtraction, instead of talking about going "from the tens column to the hundreds column", one talks about going "to a next larger unit". Similarly, in subtracting fractions (*e.g.,* $4\,^{1}/_{11} - 1\,^{7}/_{11}$), instead of 'fractional part' and 'whole part', number regrouping is, again, expressed in terms of "a larger unit".

The similarity between these two generalizations is expected to create connections between procedures. Peled and Segalis focus on these connections, yet the emerging general algorithm also demonstrates the similarity of procedural steps *within* each algorithm. Thus, in addition to connections *between* procedures (fig. 10), this abstraction can promote identification of the iterative structure *within* each procedure (fig. 11), strengthening the connections.

Global conceptual knowledge of a procedure

A third aspect of meta-procedural knowledge has to do with underlying principles that are general and applicable in other situations. For example, when researchers investigate teachers' knowledge about the standard subtraction procedure, they expect teachers to exhibit certain knowledge – specifically, that in the course of regrouping one does not actually change the total amount of the subtrahend (Ma, 1999). However, there is more to it that is not explicitly discussed, and we consider this another kind of meta-procedural knowledge.

Given a subtraction problem, such as 534 – 187, and using the standard algorithm, the minuend is transformed during the execution of the algorithm from 5h,3t,4ones into 5h,2t,14ones and then later into 4h,12t,14ones. In this process the student is working within a given column, often without seeing the general picture. That is, without realizing that 7 'cannot be taken away' before regrouping not because 534 is not big enough, but because this specific algorithm is based on subtracting within units.

In order to see the bigger picture, and as a complement of viewing the number as the sum of its columns, a quantity

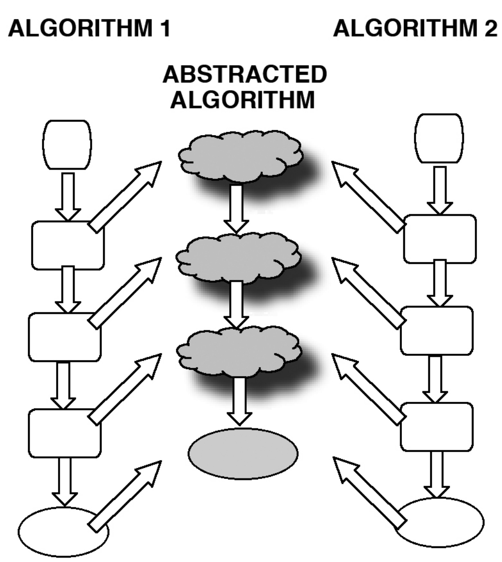

Figure 10: Connections between algorithms (Peled and Segalis, 2005)

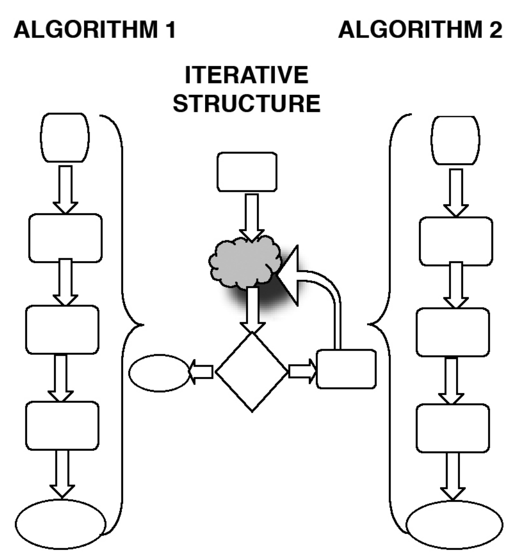

Figure 11: Emerging iterative connections within each procedure

$$A - B = (A_1+A_2+A_3) - (B_1+B_2+B_3) = (A_1-B_1) + (A_2-B_2) + (A_3-B_3) = (A^*_1-B_1) + (A^*_2-B_2) + (A^*_3-B_3)$$

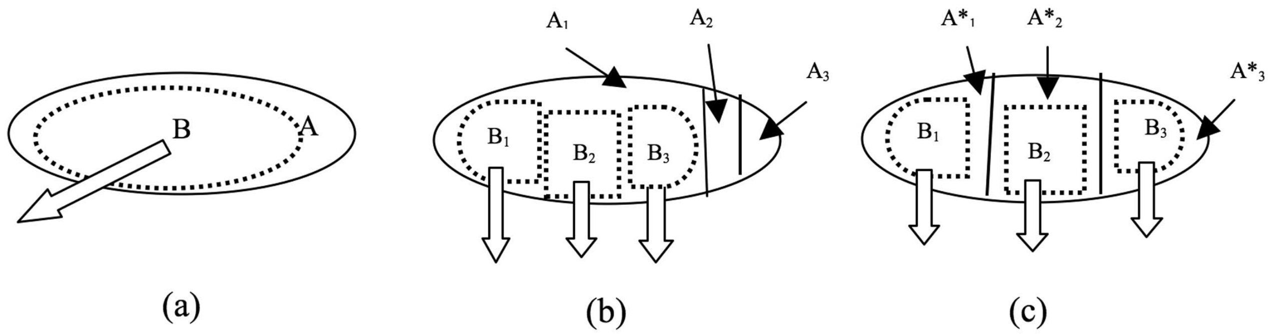

Figure 12: Meta-perspective on regrouping for subtraction

perspective can be introduced. It should be stressed that this perspective is offered here in addition to developing complex unit conception and not as an alternative to it. Using a global quantity interpretation, multi-digit subtraction can be viewed as a special case of splitting the minuend into quantities, allowing for subtraction of the subtrahend part by part. That is, subtracting B from A can be done by splitting each number into parts (as depicted in fig. 12).

In the problem 534 – 187, the initial state is based on splitting the numbers according to their decimal expression – that is,

$$534 - 187 = (500 - 100) + (30 - 80) + (4 - 7).$$

The need for regrouping emerges when within a certain unit the part that has to be taken away is 'too big', as depicted in Figure 12(b), where A_2 and A_3 are smaller than B_2 and B_3 correspondingly. The process of regrouping the minuend aims at getting the structure depicted in Figure 12(c). Specifically, when the process is described in terms of the quantities that are involved in the partial subtraction actions, we get:

$$534 - 187 = (400 - 100) + (120 - 80) + (14 - 7) = 347.$$

This global conceptual view offers an explanation and a motivation for regrouping. On the one hand it complements the iterative unit conception. On the other hand it competes with the development of the complex unit conception discussed earlier and should be handled with teacher awareness to this tension.

Such global knowledge also complements local conceptual knowledge in presenting a broader meaning for regrouping and in representing a family of subtraction algorithms. This space of algorithms offers an opportunity for investigating and inventing additional ways for solving a given problem. Specifically, given a certain computational problem, children who have acquired global conceptual knowledge together with decision making knowledge might find efficient routes within the family of algorithms. In the case of subtraction, children might search for efficient ways to split the minuend. For example, given the problem 534 – 187, different children might suggest different splits, that fit their personal preferences such as:

$$534 - 187 = (34 - 7) + (100 - 80) + (400 - 100) = 27 + 20 + 300 = 347$$

or

$$534 - 187 = (200 - 187) + (334 - 0) = 13 + 334 = 347.$$

Concluding Remarks

Taking an epistemological perspective, this paper suggests to upgrade the learning of procedures by attending to procedure-related global knowledge that has been neglected in favor of an undivided focus on local connections between procedures and conceptual knowledge.

The acquisition of global knowledge about procedures is expected to develop understanding about the role of procedures and the optimal conditions for their use. Having developed a global perspective of what a procedure means, and where it stands in relation to other procedures, the student is expected to make educated decisions with respect to the choice and the application of a procedure. The additional perspective of global conceptual knowledge is expected to support these decisions and enable flexibility in making necessary modifications.

The abstraction of procedure structure encouraged by actions of comparison of different procedures is expected to change procedure conception. Using the dual operational and structural aspects of a concept suggested by Sfard (1991), and her analysis of the stages on the route to reification, we expect these actions to facilitate growth towards conceiving a procedure both as a process and an object.

With regard to pedagogical decisions, the learning of traditional procedures that have successfully been connected to local conceptual knowledge can be enriched. As well, procedures that are difficult to connect to local conceptual knowledge might be connected to valuable global procedure-related knowledge and have more merit than originally thought.

Note

[1] A systematic analysis of the different meanings and units is given by Behr, Harel, Post & Lesh (1992).

References

Anghileri, J., Beishuizen, M. and van Putten, C. (2002) 'From informal strategies to structured procedures: Mind the gap!', *Educational Studies in Mathematics* **49**, 149–170.

Bass, H. (2003) 'Computational fluency, algorithms, and mathematical proficiency: One mathematician's perspective', *Teaching Children Mathematics* **9**, 322–327.

Baroody, A. J., Feil, Y. and Johnson, A. R. (2007) 'An alternative reconceptualization of procedural and conceptual knowledge', *Journal for Research in Mathematics Education* **38**, 115–131.

Behr, M. J., Harel, G., Post, T. and Lesh, R. (1992) 'Rational number, ratio, and proportion', in Grouws, D. A. (ed.), *Handbook of research on mathematics teaching and learning,* New York, NY, Macmillan, pp. 296–333.

Brousseau, G. (1997, translated and edited by N. Balacheff, M. Cooper, R. Sutherland and V. Warfield) *Theory of didactical situations in mathematics: didactique des mathematiques, 1970–1990,* Dordrecht, NL, Kluwer.

Fuson, C. K. (2003) 'Toward computational fluency in multidigit multiplication and division', *Teaching Children Mathematics* **9**, 300–305.

Fuson, C. K. (1990) 'Conceptual structures for multiunit numbers: implications for learning and teaching multidigit addition, subtraction, and place value', *Cognition and Instruction* **7**(4), 343–403.

Gregg, J. and Underwood Gregg, D. (2007) 'Interpreting the standard division algorithm in a candy factory context', *Teaching Children Mathematics* **14**(1), 25–31.

Hiebert, J. and Carpenter, T. P. (1992) 'Learning and teaching with understanding', in Grouws, D. A. (ed.), *Handbook of research on mathematics teaching and learning,* New York, NY, Macmillan, pp. 65–97.

Ma, L. (1999) *Knowing and teaching elementary mathematics,* Mahwah, NJ, Erlbaum.

Peled, I. and Awawdy-Shahbari, J. (2003) 'Improving decimal number conception by transfer from fractions to decimals', in Pateman, N. A., Dougherty, B. J. and Zilliox, J. T. (eds.), *Proceedings of the 27th International Conference for the Psychology of Mathematics Education,* Honolulu, HI, University of Hawai'i, 4, pp. 1–6.

Peled, I. and Segalis, B. (2005) 'It's not too late to conceptualize: constructing a generalized subtraction schema by abstracting and connecting procedures', *Mathematical Thinking and Learning* **7**, 207–230.

Radford, L. (1997) 'On psychology, historical epistemology and the teaching of mathematics: towards a socio-cultural history of mathematics', *For the Learning of Mathematics* **17**, 26–33.

Rowland, T., Thwaites, A. and Huckstep, P. (2003) 'Novices' choice of examples in the teaching of elementary mathematics', in Rogerson, A. (ed.), *Proceedings of the International Conference on the Decidable and the Undecidable in Mathematics Education,* Brno, CZ, pp. 242–245.

Schoenfeld, H. A. (1987) 'What's all the fuss about metacognition?', in Schoenfeld, A. H. (ed.), *Cognitive science and mathematics education,* Hillsdale, NJ, Erlbaum, pp. 189–215.

Sfard, A. (1991) 'On the dual nature of mathematical conceptions: reflections on processes and objects as different sides of the same coin', *Educational Studies in Mathematics* **22**, 1–36.

Simon, M. A. (1993) 'Prospective elementary teachers' knowledge of division', *Journal for Research in Mathematics Education* **24**, 233–254.

Star, J. R. (2005) 'Reconceptualizing procedural knowledge', *Journal for Research in Mathematics Education* **36**, 404–411.

Steffe, L. P. (1988) 'Children's construction of number sequences and multiplying schemes', in Hiebert, J. and Behr, M. (eds.), *Number concepts and operations in middle grades,* Hillsdale, NJ, Erlbaum, pp. 119–141.

van Putten, C. M., van den Brom-Snijders, P. A. and Beishuizen, M. (2005) 'Progressive mathematization of long division strategies in Dutch primary schools', *Journal for Research in Mathematics Education* **36**, 44–73.

Zaslavsky, O. (2003) 'On teaching and learning algorithms – the case of long division', *"Number Power" – The (Israeli) Elementary School Mathematics Journal* **6**, 38–43.

Bitstring

Consider a string of 1's and 0's. Chunk this string into adjacent pairs starting at the left and then evaluate each pair. If a pair matches replace the pair with a 0, if it doesn't match replace it with a 1. Repeat this procedure for the new string until there is only one bit left. Can you find a way to predict what the final bit will be based on the original string?

Consider the following string: 1 1 0 1 0 0 1 1 1 1 0 0 1 0 1

```
1 1 0 1 0 0 1 1 1 1 0 0 1 0 1
 0 1 1 1 0 1 0 0 0 1 0 1 1 1
  1 0 0 1 1 1 0 0 1 1 1 0 0
   1 0 1 0 0 1 0 1 0 0 1 0
    1 1 1 0 1 1 1 1 0 1 1
     0 0 1 1 0 0 0 1 1 0
      0 1 0 1 0 0 1 0 1
       1 1 1 1 0 1 1 1
        0 0 0 1 1 0 0
         0 0 1 0 1 0
          0 1 1 1 1
           1 0 0 0
            1 0 0
             1 0
              1
```

(unknown origin; selected by Peter Liljedahl)