To count or not to count, that's the question

Dewi Freitag Jasinta

UNIL | Université de Lausanne

FACULTÉ DES SCIENCES SOCIALES ET POLITIQUES

INSTITUT DE PSYCHOLOGIE

To Count or Not To Count, That's the Question

THÈSE DE DOCTORAT

présentée à la

Faculté des sciences sociales et politiques

de l'Université de Lausanne

pour l'obtention du grade de Docteure en Psychologie

par

Jasinta D. M. Dewi Freitag

Directrice de thèse

Prof Catherine Thevenot

Jury
Prof Pierre Barrouillet (Université de Genève)
Prof Pierre Lavenex (Université de Lausanne)
Prof Gordon Logan (University of Vanderbilt, États-Unis)
Dr Jérôme Prado (CNRS & Université de Lyon 1, France)
Prof Jane Zbrodoff (University of Vanderbilt, États-Unis)

Lausanne

2022

UNIL | Université de Lausanne
Faculté des sciences sociales et politiques

IMPRIMATUR

Le Décanat de la Faculté des sciences sociales et politiques de l'Université de Lausanne, au nom du Conseil et sur proposition d'un jury formé des professeurs

- Catherine THEVENOT, Directrice de thèse, Professeure à l'Université de Lausanne
- Pierre BARROUILLET, Professeur à l'Université de Genève
- Pierre LAVENEX, Professeur à l'Université de Lausanne
- Gordon LOGAN, Professeur à la Vanderbilt University, Nashville
- Jérôme PRADO, Chargé de recherche à l'Université de Lyon
- Jane ZBRODOFF, Professeure à la Vanderbilt University, Nashville

autorise, sans se prononcer sur les opinions de la candidate, l'impression de la thèse de Madame Jasinta DEWI FREITAG, intitulée :

« To count or not to count, that's the question »

Nicky LE FEUVRE
Doyenne

Lausanne, le 10 janvier 022

Abstract

Although it has been widely acknowledged that children start solving addition through counting, the evolution of this strategy in the course of learning is still a matter of debate. The proponents of retrieval models postulate that counting is replaced by the retrieval of stored associations between the problem and its answer from long-term memory. These models were noticeably supported by the instance theory of automatization, according to which algorithm-based performance is replaced by retrieval-based performance at the end of practice. On the other hand, the promoters of the automated counting procedure theory advocate for the acceleration of counting until it becomes automatic and unconscious. This thesis aims at testing the opposing models by means of 3 alphabet-arithmetic training experiments, wherein adult participants were asked to learn to solve problems such as D + 3 = G. Strong individual differences were observed in participants' performance, based on which the existence of three major groups corresponding to three different learning strategies was revealed. The implications of the results for the instance theory of automatization and for the models of arithmetic learning, as well as for the domain of education are discussed.

Bien qu'il soit largement reconnu que les enfants commencent à résoudre des additions par comptage, l'évolution de cette stratégie au cours de l'apprentissage fait encore l'objet de débats. Les partisans des modèles de récupération proposent que le comptage est remplacé par la récupération d'associations entre le problème et sa réponse stockées en mémoire à long terme. Ces modèles ont été notamment soutenus par la théorie de l'automatisation des instances, selon laquelle les performances basées sur les algorithmes sont remplacées par des performances basées sur la récupération à la fin de l'entraînement. En revanche, les promoteurs de la théorie des procédures de comptage automatisés suggèrent une accélération du comptage jusqu'à ce qu'il devienne automatique et inconscient. Cette thèse vise à tester ces modèles opposés au moyen de 3 expériences d'entraînement à l'alphabet-arithmétique, dans lequel des adultes devaient apprendre à résoudre des problèmes tels que D + 3 = G. De fortes différences individuelles ont été observées dans les performances des participants, ce qui a permis de révéler l'existence de trois grands groupes correspondant à trois stratégies d'apprentissage différentes. Les implications des résultats pour la théorie des instances d'automatisation et pour les modèles d'apprentissage de l'arithmétique, ainsi que pour le domaine de l'éducation, sont discutées.

To the memory of my father

who introduced me to the world of scientific thinking.

This one is for you...

Acknowledgements

Doing a thesis is not a walk in the park, even when we do it for the second time. It is more like a walk along the Camino de Santiago; we have started it but we never know whether we will finish it. I would therefore like to thank everybody whom I have met along this long path and have made the journey easier and more agreeable to me.

The first person is of course my thesis director, Catherine Thevenot. Thank you, Catherine, for having every confidence in me ever since you supervised my master thesis, for giving me the flexibility in how I do my work and the liberty in what to do with my work – be it creating experimental experiments or playing around with statistical analyses –, for being patient with me, and most of all for giving me the precious possibility to maintain a healthy work-and-life balance. And I really appreciate your severe but at the same time humane character. I would also like to acknowledge the members of the jury – Pierre Barrouillet, Pierre Lavenex, Gordon Logan, Jérôme Prado, and Jane Zbrodoff – for carefully reading the manuscript and for their positive comments, that made me feel that the 5 years spent doing another thesis were not a waste of time. Along the same lines, I would like to thank the referees of our papers whose criticisms allowed our work to improve.

I would like to express my gratitude to Pierre Barrouillet, Romain Mathieu, and Kim Uittenhove – then at the University of Geneva – as well as Pamela

Banta Lavenex and Pierre Lavenex for their constructive inputs during the preparation of my thesis project. My thanks are extended to the Swiss National Foundation for Scientific Research for their grant number 100014_138185 to Catherine Thevenot – that funded Experiment 1 – and to the Foundation Chuard-Schmid of the University of Lausanne for their grant to Catherine Thevenot – that funded Experiments 2 and 3. The travel grant from the Swiss Academy of Human and Social Sciences (ASSH) that allowed me to attend the 21st conference of the European Society for Cognitive Psychology (ESCoP) in Tenerife, 25–28 September 2019, and the funding from the Institute of Psychology of the University of Lausanne that allowed me to attend various scientific meetings are very much appreciated.

I would also like to acknowledge Jeanne Bagnoud, Pamela Banta Lavenex, Mathilde Bostelmann (with her constant warning against the *lerbose*), Adeline Jabès, and Simon Thuillard who kept me company during the long two months of training experiments, and without whom working in the white, sterile, windowless laboratory – from early morning till late in the evening – would have been dark, lonely, and just miserable. And then there was Paolo Ruggeri, who was always willing to help with any lab-related problems, and who was there when I had my deepest doubt. Grazie mille, Paolo, you were right, I made it! And speaking of the lab, this thesis would not have been realised without the participants, who came to the lab dutifully and faithfully every single day for 3 weeks, come rain or shine.

If could start my thesis smoothly, it was thanks to Caroline Castel and Sandrine Masson who were always ready to rescue me from any problems. Merci beaucoup, mes très chères. Next, if I could continue doing my thesis smoothly, it was thanks to my two precious colleagues – as intelligent as they are amiable – Jeanne Bagnoud with her out-of-the-ordinary sense of humour and Marie Krenger with her amazing and incredible multi talents and original ideas. Jeanne, thank you for the time we spent together going to conferences as well as to courses on random subjects (not to mention the singing course and of course the unforgettable gymstick sessions), for your concept of *recherche digestive* that has spared many an afternoon, for your initiation to cocktails, all that in addition to your advance notices about what to eat at the cafeteria and your French-as-you've-never-heard-it-before course. Marie, despite the little time we have shared together, I always enjoyed our deep, often philosophical, conversations. Thank you, the two of you. Finally, if I could defend this thesis in time, it was thanks to the Schreibcircle sessions, that we managed to run even during the period of confinement. Special thanks are due particularly to its 5 special members – *la fine équipe* – Jeanne Bagnoud, Noëllie Dunand, Liliane Efinger, Cindy Eira Nunes, and Marie Krenger, not only for the shared fun moments within and without the schreib sessions but more importantly for the shared confidences. I was really lucky to have you.

Continuous support throughout these last 5 years also came from the secretaries of the Institute of Psychology, whose patience made me feel that I was the only one working at the institute. I would extend my gratitude to the Géopolis cafeteria – where some parts of this thesis were written – for the conducive working environment (and the music!), their friendly staff, and excellent food (not to forget the free second serving!).

My family in Indonesia and my friends all over the world are thanked for accompanying me either through their messages or their presence. A particular gratitude is due to my mother for her constant understanding, trust, and pray. Finally, to my two dearest ones, Marc and Tiwi, for whom any word will fall short to express my debt and gratitude.

Contents

List of Figures

List of Tables

1

Introduction

Addition is the first arithmetic operation that children learn, and they often have encountered this notion even before they start formal schooling. It is commonly accepted that when they start learning addition, children use counting strategy (e.g., Baroody, 1987; Carpenter & Moser, 1984) and that this strategy is progressively replaced by the retrieval of the association between the numbers to add and their sum (Groen & Parkman, 1972; Siegler, 1996). This way, by the end of primary school, children would solve single-digit additions principally using memory retrieval and when they reach adulthood, counting strategies would no longer be used (e.g., Ashcraft & Battaglia, 1978; Ashcraft & Fierman, 1982; Ashcraft & Stazyk, 1981). This view has however been challenged by recent findings (e.g., Barrouillet & Thevenot, 2013; Thevenot & Barrouillet, 2020; Uittenhove et al., 2016) suggesting that adults solve very simple additions, i.e. those involving two operands ranging from 1 to 4, by counting strategy, albeit unconsciously.

Thus far, the instance theory of automatization (Logan, 1988), according to which algorithm-based performance is replaced by retrieval-based performance after repeated practice, has lent support for retrieval models of mental arithmetic. This was shown by way of the alphabet-arithmetic paradigm (e.g., Logan & Klapp, 1991), wherein a number is added to a letter augend resulting in a letter answer, e.g. B + 4 = F because F is 4 letters away from B. Such support would invalidate the concurrent counting theory that will be discussed in this thesis. Nevertheless, we identified several elements in the results of past alphabet-arithmetic studies that cast doubt on the derived conclusions. These elements will be revisited in this thesis through three training experiments, wherein adults were trained to solve alphabet-arithmetic problems during a number of sessions. The analyses presented in this manuscript will focus on the evolution of their performance, in terms of solution times and error rates, both on trained and untrained items. The results of this thesis put the retrieval models into question and highlight the individual differences in learning.

1.1 Literature Review

Strategies in Simple Additions

Studies on mental arithmetic, notably on simple additions, have found the so-called problem-size effect, i.e., solution times and error rates increase with the size of the problem (see Zbrodoff & Logan, 2005, for a review). It has been widely accepted in the literature that the slope of the function relating solution times to the problem size indicates the strategy used to solve simple additions. However, whereas non-significant slopes indicate the use of retrieval strategy, the meaning of significant slopes is still a subject of disagreement between the proponents of retrieval and procedural models. Furthermore, as will be described in this section, there is a debate in the literature as to which variable should determine the problem size.

Children and Counting

The first studies investigating strategies involved in the resolution of addition problems focused on children's performance (e.g., Groen, 1967, as cited in Jerman, 1970; Groen & Parkman, 1972; Jerman, 1970; Restle, 1970; Suppes

Table 1.1
Five Counting Models

Model	x	Process
First operand	N_1	The counter is set to N_2, and N_1 is added to N_2
Second operand	N_2	The counter is set to N_1, and N_2 is added to N_1
Minimum operand	$\min(N_1, N_2)$	The counter is set to the maximum of the two operands, and the minimum operand is added to the maximum operand
Maximum operand	$\max(N_1, N_2)$	The counter is set to the minimum of the two operands, and the maximum operand is added to the minimum operand
Sum of operands	$N_1 + N_2$	The counter is set to 0, the first operand is added to 0, and then the second operand is added to the first

& Groen, 1967, as cited in Jerman 1970; Suppes et al., 1970, as cited in Restle, 1970; Svenson, 1975; Svenson & Broquist, 1975). Considering that children most frequently use counting strategy, Suppes and Groen (1967, as cited in Jerman, 1970) proposed five counting models to explain the solution times of an addition problem $N_1 + N_2$, each of which could be expressed by the equation

$$T = \alpha + \beta\, x, \tag{1.1}$$

with T representing the solution times and x the predictor as defined by each model (see Table 1.1). Hereinafter, the discussion about problem-size effect only concerns problems wherein N_1 and N_2 are different, i.e., the non-tie problems.

Among these predictors, early works identified the model with the minimum operand, hereinafter the min model, as the best one to explain children's solution times (Groen, 1967, as cited in Jerman 1970; Groen & Parkman, 1972; Jerman, 1970; Suppes & Groen, 1967, as cited in Jerman 1970; Svenson, 1975; Svenson & Broquist, 1975). Within this model, for first graders, the slope β in Equation 1.1 was 410 ms/unit increment when non-tie additions with the sum inferior to 10 were involved (Groen & Parkman, 1972). This means that it takes about 400 ms for first graders to count 1 unit increment. Furthermore, the slope for the min model decreases with age, i.e., from 240 ms/unit increment for third graders to 110 ms/unit increment for sixth graders when all single-addition problems were included (Jerman, 1970), indicating an acceleration of counting rate. The experimental results favouring the min model were moreover in line with the collected verbal report, showing that children set their counter at the maximum operand (Svenson, 1975). Interestingly, this min model is never taught formally and children seem to find this method by themselves (Groen & Resnick, 1977; Siegler & Jenkins, 1989), probably because by doing so, they "don't have to count a very long ways" (Siegler & Jenkins, 1989, p. 93). Thus, according to the counting model, problem size effect is related to the number of counting steps.

Nevertheless, although the min model could explain 77% (Svenson, 1975) or 80% (Groen & Parkman, 1972) of the variance at the group level, it explains less than 50% of the variance at the individual level. This is because, as attested by children's verbal reports, the use of different strategies, for example retrieval, counting more than one step at a time, or using tie problems as a reference, were also reported (Mulhern, 1987; Siegler, 1987; Siegler & Robinson, 1982; Svenson, 1975; Svenson & Broquist, 1975; Svenson & Hedenborg, 1980; Svenson et al., 1976; Svenson & Sjöberg, 1983). At the level of problem, when problems were analysed separately according to the reported strategies, the min model was the best predictor only for problems reported as being solved by counting whereas none of the counting models could predict problems reported as being solved by retrieval (e.g., Svenson & Hedenborg, 1980). Similar finding was observed at the level of individual. In the study of Mulhern (1987), for children who reported of having used counting strategy exclusively, solution times were best predicted by the minimum operand. However, for the only child in the study who reported of having used retrieval exclusively, none of the counting models could explain the performance and the slope for the min model for this child was 8 ms/unit increment. Thus, Siegler and collaborators (e.g., Siegler, 1987; Siegler & Robinson, 1982) argued that averaging solution times over problems that are solved through different strategies could mislead the conclusion regarding the strategy used, either at the problem or individual level. For example, Siegler (1987) found the min operand to be the best predictor of solution times, i.e., with $R^2 = .86$, even though the min strategy was reported only in 36% of trials and retrieval was reported almost at the same proportion, i.e., 35%.

From Counting to Retrieval

Parkman and Groen (1971) were the first to study adults' performance in solving addition problems. Applying the same counting models proposed for children, they found the minimum operand ($R^2 = .73$) and the sum ($R^2 = .71$) as the best predictors of adults' solution times. Given the fact that the sum model was not a good predictor of children's performance, they concluded that adults and children do not rely on the same strategy. Furthermore, considering that the slopes, i.e., 20 and 11 ms/increment for the min and the sum models, respectively, were much lower than the rate of silent counting, i.e., 125 ms/unit increment (Landauer, 1962), Groen and Parkman (1972) concluded that counting could not be the main strategy used by adults. Instead, they suggested that the slope of 20 ms/unit increment was the artefact of averaging problems solved by different strategies, i.e., counting and memory retrieval. For each of these two strategies, Groen and Parkman rewrote Equation 1.1 as $T_{\text{counting}} = \alpha + \beta\, x$ for counting trials and $T_{\text{retrieval}} = \gamma$ for retrieval trials. Assuming that p is the proportion of problems solved by counting, the overall solution times could be written as $T = T_{\text{counting}} \,.\, p + T_{\text{retrieval}} \,.\, (1-p)$. Assuming furthermore that α is equal to γ, the latter equation can be expressed as follow

$$T = \alpha + \beta\, p\, x \tag{1.2}$$

Taking $\beta\, p$ to be the observed adults' slope, i.e., 20 ms/unit increment, and assuming that adults' real counting rate, β, resembled that of children's, i.e., 400 ms/unit increment, Groen and Parkman obtained a p of .05, or once in every 20 problems. Therefore, they concluded that adults rely on retrieval most of the time but, failing 5% of the trials, they have recourse to counting. Naturally, if we take the rate of silent counting of 125 ms/unit increment as adults' counting rate, β, Equation 1.2 yields a higher proportion of counting trials, i.e., 4 times in every 25 problems, or 16%.

Nevertheless, Ashcraft and collaborators (Ashcraft & Battaglia, 1978; Ashcraft & Fierman, 1982; Ashcraft & Stazyk, 1981, see also Fierman, 1980, as cited in Ashcraft, 1992) argued that it is not possible that adults still resort to counting strategy to solve addition. They therefore explored other variables to explain solution times, for example the difference between the two operands, the parity of each operand, the parity of the sum, and the square of the sum. This way, they found that solution times were best predicted by the square of the sum of the operands, i.e., $(N_1 + N_2)^2$. Considering that as a structural variable, the sum squared cannot explain any counting model, Ashcraft and collaborators concluded that this variable must serve instead as an index of retrieval, in a way that the larger the retrieval index the more time it takes to recall the sum.

Ashcraft and Battaglia (1978) therefore advanced a model, according to which children start learning addition by counting but, as learning progresses, the association between the addends and the sum gets stronger, in a way that counting strategy is progressively replaced by memory retrieval as children grow older. Thus, according to the so-called network-retrieval model (see Ashcraft, 1992; Ashcraft & Guillaume, 2009, for more recent versions), problem-size effect observed in adults was not the sign of counting but was the consequence of the retrieval index. They argued that smaller problems have lower retrieval index because these problems were learnt earlier and were encountered more frequently in the course of development than larger problems (Ashcraft & Christy, 1995; Hamann & Ashcraft, 1986). This model was supported by the fact that whereas the minimum operand was the best predictor of solution times for first graders, the sum squared was the best predictor for adults and children starting from the fourth grade (Ashcraft & Fierman, 1982; Fierman, 1980, as cited in Ashcraft, 1992; Kaye et al., 1986).

Nonetheless, frequency is not the only factor that could influence solution times in adults' simple-arithmetic solving. Indeed, whereas the network-retrieval model assumes that the access to retrieval was determined by frequency, the network-interference model (Campbell, 1987a, 1987b, 1995; Campbell & Graham, 1985; Campbell & Oliphant, 1992; De Visscher & Noël, 2014; Graham & Campbell, 1992) suggests that the problem-size effect is due to interference between arithmetic facts in long-term memory. The latter model was based on earlier works about the verification of additions and multiplications (e.g., Miller et al., 1984; Stazyk et al., 1982; Winkelman & Schmidt, 1974), wherein addition (or multiplication) problems associated with the product (or the sum) of the operands were rejected slower than addition (or multiplication) problems associated with other incorrect answers. These results were interpreted as due to the interference of the product of the two operands in an addition task or the interference of the sum of the two operands in a multiplication task, although multiplication products interfere in an addition task more than addition sums in a multiplication task (Miller & Paredes, 1990). The interference tends to be stronger for larger problems because these problems share the same sum or product with more problems than smaller problems, hence the problem-size effect observed in adults.

Similar to network-retrieval model, network-interference model also advocates that to solve simple additions and multiplications, adults use retrieval predominantly if not exclusively. However, to explain the problem-size effect observed in adults, another factor than a reconstructive strategy is required, which is the frequency for the former and the interference for the latter.

From Counting to Automated Counting

Despite the dominance of retrieval models in the domain of simple addition, Baroody and collaborator (e.g., Baroody, 1983, 1984, 1994, 2018; Baroody & Ginsburg, 1986) argued that simple arithmetic problems could also be solved by reconstructive methods such as reliance on rules and heuristic. This idea is in line with more-recent studies suggesting that counting procedure learnt during childhood is not replaced by memory retrieval but is still used by adults. Several chronometric studies revealed that this is particularly true for very small additions involving operands smaller than 5 (Barrouillet & Thevenot, 2013; Uittenhove et al., 2016). Barrouillet and Thevenot studied how adults solved addition problems involving operands from 1 to 4. By means of an ANOVA, they found an effect of both the first and second operands on solution times. These results echoed earlier finding of Aiken and collaborator (Aiken, 1971; Aiken & Williams, 1973) who, using a similar analysis, found an effect on both the minimum and maximum operands for all simple additions. Considering the effect of both operands, both groups of authors concluded therefore that adults must have used counting to solve addition problems.

Uittenhove et al. (2016) extended the work of Barrouillet and Thevenot (2013) to all simple-addition problems. Using a linear regression instead of an ANOVA, Uittenhove et al. showed that the sum of the operands was the best predictor of solution times. However, the plot of solution times as a function of the sum of the operands shows a complicated picture. In fact, an increase in solution times from the sum of 2 to 7 was followed by a plateau from the sum of 7 to 10, then another increase from the sum of 10 to 13, and another plateau for the sum superior to 13. This indicates that different strategies were probably used for different types of problem. Uittenhove et al. also analysed the data according to the reported strategy use. Interestingly, a significant slope of 47 ms/sum increment for problems with operands from 1 to 4 was still found for the frequent retrievers, i.e., participants who reported of using retrieval exclusively to solve this type of problems. Given this significant slope obtained for frequent retrievers, the authors concluded therefore that very-small additions could be solved by a rapid, unconscious, automated, one-by-one counting procedure. For problems involving larger operands, on the other hand, counting procedure may be too effortful, in terms of the balance between the time spent to execute the procedure and the success of outcome, such that retrieval is more favourable. This may explain the plateau between the sum of 7 and 10.

The results revealed by Uittenhove et al. (2016) echoed the finding of Svenson (1985) who showed that adults solved additions involving 1 as the minimum operand in 805 ms and additions involving 2 as the minimum operand in 841 ms. Considering that the difference of 36 ms was significant, Svenson concluded that adults must have used a counting procedure to solve addition problems involving +1 or +2 with a speed of about 40 ms/count, though they might not be conscious of having done so. Svenson proposed furthermore that larger addition problems, on the other hand, must have been solved by retrieval because neither the minimum operand nor the sum of the operands could explain solution times for these problems.

Thus, the results of the above-mentioned chronometric studies (Aiken, 1971; Aiken & Williams, 1973; Barrouillet & Thevenot, 2013; Svenson, 1985; Uittenhove et al., 2016) showed that contrary to the view of the proponents of the retrieval models, it is not unconceivable that adults still use counting procedure, though only for the smallest problems. According to this automated counting procedure theory (Thevenot & Barrouillet, 2020; Uittenhove et al., 2016), the slow, effortful, and conscious counting strategy learnt during childhood is accelerated during the course of development until it became a fast, automatic, and unconscious procedure in adulthood. More precisely, as has also been proposed by Aiken and collaborator (Aiken, 1971; Aiken & Williams, 1973), solving addition could correspond to a scanning through the mental number line from the origin to one of the operands, then, from this operand, scanning the number of steps corresponding to the other operand. The limit of operand 4 may be connected to the number of chunks that can be kept in short-term memory (e.g., Cowan, 2000) and may be related to the average limit of subitizing, i.e., the capacity to enumerate a set of items in an accurate and effortless way (Kaufman et al., 1949). In the framework of the horse-race model (Logan & Cowan, 1984; Logan et al., 1984), for additions involving operands 1 to 4, counting up to 4 steps can be executed faster than directly retrieving the answer to the problem, whereas for additions involving an operand larger than 4, retrieval would be faster than counting. That is the reason why the use of counting for additions with an operand larger than 4 cannot be automatised.

The use of counting in addition involving operands inferior to 5 may explain why the facilitation of problem solving when the operation sign was presented 150 ms before the operands was only found for small addition but not for multiplication problems or large additions (Fayol & Thevenot, 2012). Similar result was obtained in children from the age of 12 or 13 years (Díaz-Barriga Yáñez et al., 2020; Mathieu, Epinat-Duclos, Léone et al., 2018; Poletti et al., 2021). The automatic activation of counting procedure may also be related to the spatial shift of attention to the right hemifield during addition solving (Liu et al., 2017;

Masson et al., 2018; Zhu et al., 2018), that was not found during multiplication solving (Li et al., 2018; Mathieu et al., 2016). Support from neuroimaging studies come from the fact that multiplication solving activates brain regions involved in verbal retrieval, whereas simple-addition solving activates regions involved in spatial attentional processing (Mathieu, Epinat-Duclos, Sigovan et al., 2018).

Whereas the works mentioned in the previous paragraph provided indirect support for the automated counting procedure, a more-direct support was offered by the study of Pinheiro-Chagas et al. (2017). In this study, adult participants saw a number line, delimited by 0 on the left and 10 on the right, at the top of a touch-screen tablet. When they put their finger on a particular point at the bottom of the screen, an addition or a subtraction problem appeared. Without removing their finger from the screen, participants had to put their finger on the number line, at the position corresponding to the answer to the presented problem. Thus, this paradigm allows for the tracking of the finger trajectory on the screen. A linear regression showed that the minimum operand was the best predictor of solution times, with a significant slope of 21 ms/unit increment, similar to the finding of Parkman and Groen (1971). More interestingly, trajectory dynamics showed that participants first directed their finger towards the maximum operand, and then changed direction towards the correct result. More importantly, the deviation from the maximum operand was gradual and proportional to the size of the minimum operand. Thus, the results of this study revealed that additions are calculated by a displacement on the mental number line, starting at the maximum operand and adding the minimum number one by one.

Multiple Strategies

I have described earlier how protocol analysis has led to the recognition that several strategies were used by children to solve addition problems (e.g., Geary & Burlingham-Dubree, 1989; Hamann & Ashcraft, 1986; Mulhern, 1987;

Siegler, 1987; Siegler & Robinson, 1982; Svenson, 1975; Svenson & Broquist, 1975; Svenson & Hedenborg, 1980; Svenson et al., 1976; Svenson & Sjöberg, 1983). This has led Siegler and collaborators (Shrager & Siegler, 1998; Siegler, 1996; Siegler & Jenkins, 1989; Siegler & Shipley, 1995; Siegler & Shrager, 1984) to propose the strategy-choice model. In the beginning of learning, according to this model, any association between a problem and its answer is encoded, and all associations have a flat distribution. The encoding process also includes the association between the problem and incorrect answers. Then, a positive feedback on a correct association will make the distribution peaked whereas a negative feedback on an incorrect association flatter. In the course of development, more positive feedback will lead to a more and more peaked distribution. The peakedness of a distribution will then determine the probability that a problem will be solved through retrieval. In the failure of retrieval, a back-up strategy, i.e., counting, will be used instead. Thus, according to the model that is also called the overlapping-waves model, children possess several strategies to solve the same problems. The choice of which strategy to use is adaptive, because it depends on the perceived difficulty, the trade-off between the effort or time and accuracy, and the circumstance of the task, e.g., whether the task demands accuracy or speed (Siegler, 1996). However, like in the network-retrieval and network-interference models, memory retrieval is considered as the ultimate strategy that is privileged by experts while counting or any other reconstructive procedure is only a back-up strategy.

Despite the supposition that retrieval should be the dominant strategy used by adults, protocol analysis revealed that it was not the only strategy used by adults to solve simple arithmetic problems (e.g., Geary & Wiley, 1991; Hecht, 1999, 2002; LeFevre, Bisanz et al., 1996; LeFevre, Sadesky et al., 1996; LeFevre et al., 2003; Svenson, 1985). For addition problems, the use of retrieval could be 71% when all 100 combinations were tested (LeFevre, Sadesky et al., 1996) or 88% when only non-tie problems with operands from 2 to 9 were tested (Geary &

Wiley, 1991). This is lower than the proportion of 95% estimated by Groen and Parkman (1972) but corresponds more to my own estimate of 84% using Equation 1.2 and assuming the rate of silent counting. Furthermore, LeFevre, Sadesky et al. (1996) found that out of 16 participants, only 2 reported of having used retrieval exclusively, implying that multiple strategies are the norm rather than the exception. The same authors found further that among the reconstructive strategies reported by adults to solved simple additions, the choice of strategy depends on the problem characteristics. More precisely, transformation or decomposition was favoured to solve additions with a sum superior to 10, whereas counting was more frequently reported when the minimum operand was 1, 2, or 3 than when it was larger than 3.

Similar to researches involving children, studies that revealed the use of multiple strategies in adults also insist on the danger of averaging solution times over different strategies to draw conclusion about the strategy used. Indeed, LeFevre, Sadesky et al. (1996) found that for the whole sample, whereas the min model could predict the solution times with a slope of 49 ms/unit increment, it only explained 49% of the variance. However, when problems reported as being solved by counting were analysed separately, the min model explained 71% of the variance and the slope increased to 264 ms/unit increment, much larger than the rate of silent counting, i.e., 125 ms/unit increment (Landauer, 1962). Although protocol analysis has been criticised (e.g., Kirk & Ashcraft, 2001; LeFevre, Sadesky et al., 1996; Smith-Chant & LeFevre, 2003), the results for reported counting might be reliable, considering that counting is a conscious process and is therefore accessible and reportable. Furthermore, when problems reported as being solved by counting were analysed separately (e.g., LeFevre, Sadesky et al., 1996; Svenson, 1985; Uittenhove et al., 2016), a higher proportion of explained variance for the min model was obtained, implying the consistency between the used and the reported strategy, at least for counting trials.

We have seen in this section that the question of what is automatised in the course of addition learning is central to the debate between the proponents of the retrieval models and the automated counting procedure model. The transition from counting to retrieval, that would support the retrieval theories, has been modelled by the instance theory of automatization (Logan, 1988).

Instance Theory of Automatization

In cognitive psychology in general, a process is considered to be automatic if it is fast, effortless, autonomous, stereotypic, and unavailable to conscious awareness. Such a process is thought to be acquired through practice in specific task environments. If we consider the resource limitations of attention, automaticity can be viewed as the result of gradual withdrawal of attention (e.g., W. Schneider & Shiffrin, 1977; Shiffrin & Schneider, 1977, 1984). However, if we consider the memorial aspect of attention, automaticity can be considered to be equal to memory retrieval, and automatic performance can be viewed as a performance that is executed based on a single-step, direct-access retrieval from memory.

Emphasising on the memorial aspects of attention instead of the resource limitation aspects, Logan (1988) proposed the instance theory of automatization. This theory has three main assumptions. Firstly, encoding is obligatory, because it is an unavoidable consequence of attention. Although attention is sufficient for encoding, the quality and quantity of attention will determine the quality of encoding. Secondly, retrieval, whether it is successful or not, is obligatory, because it is also an unavoidable consequence of attention. The success of retrieval and the time to retrieve depend, among others, on practice on the task. In the case of slow retrieval, if a faster alternative algorithm exists, the latter will be used to provide the solution to the task. Finally, each instance of learning or each encounter with a stimulus is encoded, stored, and retrieved separately as a separate trace.

Logan (1988) proposed further three quantitative properties of the instance theory. Firstly, there is a speed-up in processing, represented by a reduction in solution time across trials, that follows a power function. The decrease in the mean of solution times, $\bar{x}_{\mathrm{T}}$, as a function of the number of trials, n, can be expressed by the equation

$$\bar{x}_{\mathrm{T}} = a + b\,n^{-c}, \tag{1.3}$$

where a is the solution times when performance reaches the asymptote, b is the difference in solution times between the initial and the asymptotic performance, and c is the rate of learning. Secondly, there is a reduction in variability, that also follows a power function. Thirdly, the power functions that govern the reduction of the mean, $\bar{x}_{\mathrm{T}}$, and the standard deviation, s_{T}, of the solution times have the same exponent c. Hence,

$$s_{\mathrm{T}} = a_1 + b_1\,n^{-c}, \tag{1.4}$$

The third property can be fulfilled if solution times follow a Weibull distribution (Logan, 1988, 1992, 1995), although other distributions such as ex-Gaussian, shifted lognormal, shifted Weibull, and Gumbel are also suitable (Colonius, 1995; Wagenmakers & Brown, 2007).

The instance theory of automatization was put forward to explain the acquisition of cognitive skills that have to be solved initially by algorithm but that can be solved by memory retrieval after repeated practice, such as the acquisition of basic arithmetic, reading, or writing skill (Logan, 1988). Indeed, in the early stages of learning, one-by-one counting is used to solve simple addition whereas reading and writing are first learnt by associating a grapheme and a phoneme. According to the instance theory of automatization, the solution process is based on a horse-race model (Logan & Cowan, 1984; Logan et al., 1984), according to which two processes, in this case algorithm and memory retrieval, are activated simultaneously at the onset of a problem-solving activity. In the beginning of a learning process, in the absence of memory trace, the algorithm will win the race. Then, throughout the learning process, with each instance of learning, one trace associating a problem and its answer or a grapheme and a phoneme is created in memory. Every time a problem is presented, all traces corresponding to this problem would participate in the race against each other as well as against the alternative algorithm. The more traces there are in memory, the higher is the chance that one trace will win the race. Contrarily, the probability for algorithm to win the race is assumed to be constant throughout practice. Therefore, at one moment during learning, algorithm will lose the race against memory retrieval. This point, where a shift from algorithm-based to memory-based performance is obtained, corresponds to what Logan (1988) called automatisation.

According to the instance theory of automatization, each memory trace is associated with one instance of learning. Hence, learning is necessarily item based, i.e., involving the association of specific responses to specific stimuli or specific problems. Automaticity is therefore specific to the stimuli or the learnt problems and hence, by definition, the learnt associations cannot be transferred to new stimuli or new problems.

Since the first publication (Logan, 1988), the instance theory of automatization (Logan, 1992, 1995, 1998; Logan & Etherton, 1994; Logan et al., 1996) has been used to illustrate the automatisation of reading by means of the lexical decision task, i.e., to judge whether or not a string of 4 letters is an English word (Grant & Logan, 1993; Logan, 1988, 1990, 1997), the automatisation of counting by means of the dot-counting task (Lassaline & Logan, 1993; Logan, 1992), and the automatisation of arithmetic fact retrieval by means of the alphabet-arithmetic task (Compton & Logan, 1991; Klapp et al., 1991; Logan, 1992, 1998;

Logan & Klapp, 1991). Because this thesis is related to the acquisition of addition skill, I will only discuss studies involving an alphabet-arithmetic task.

Alphabet-Arithmetic Tasks

Alphabet-arithmetic tasks involve adding a numerical addend to a letter augend resulting in a letter answer (e.g., A + 5 = F). The task was created for adults to mimic the way children learn addition (Logan, 1988; Logan & Klapp, 1991). This is because, firstly, when children start learning addition, they are already familiar with the sequence of numbers. In alphabet-arithmetic tasks, this is translated into familiarity with the sequence of letters of the alphabet among adults. Secondly, children start learning addition by counting. In alphabet-arithmetic tasks, due to the novelty of the task, adults also perform the task initially by counting. Lastly, both in addition and alphabet-arithmetic tasks, with enough practice, it is possible to perform the task by memory retrieval.

In Experiment 1 of their seminal paper, Logan and Klapp (1991) trained adults to perform an alphabet-arithmetic verification task, i.e., participants had to decide whether equations such as A + 4 = F was true or false. During 12 sessions, participants had to learn to verify 40 problems, consisting of 10 letters combined with addends 2, 3, 4, and 5. One half of the participants were trained on the first 10 letters of the alphabet and the other half on the second 10 letters. In each session, each problem was presented 6 times with its true answer and 6 times with false answers. The false answers were constructed either with the letter preceding the correct answer, i.e., T–1 equations, or with the letter following it, i.e., T+1 equations. After 12 sessions of training, in Session 13, participants underwent a transfer session, during which they had to verify 40 new problems associated with the untrained 10 letters. Then in Session 14, they were again tested on the learnt problems, and in Session 15, they were tested on both the learnt and new problems. To test the retention of the learnt problems, participants were again tested on these items in Session 16, which was run 30 days after Session 15.

Logan and Klapp (1991) claimed of having found support for the instance theory of automatization for three reasons. Firstly, there was an evolution in the slope of solution times as a function of addend (hereinafter: addend slope) from significant at the beginning to non-significant at the end of learning, implying a shift in the used strategy from algorithm to memory retrieval. Secondly, addend slope during the transfer session returned to a significant level, implying that there was no transfer from learnt to new problems, and hence, learning is item specific. Thirdly, both mean and standard deviation of solution times across trials could be fit with power functions with the same rate of learning, as predicted by the instance theory of automatization. Additionally, Zbrodoff (1999) provided further support for the shift from counting to retrieval with her concept of opportunistic stopping that will be described later. In what follows, by way of presenting studies in the alphabet-arithmetic literature, I will discuss the 4 points mentioned above in more detail.

Addend Slope

The first support for the instance theory of automatization is the shift in strategy from algorithm-based to memory-based strategy. In Experiment 1 of Logan and Klapp (1991), this was shown by the addend slopes that decreased from a significant 486 ms/addend in Session 1 to a non-significant 45 ms/addend in Session 12. Considering that a significant addend slope indicates the use of a counting strategy and a non-significant addend slope implies the use of memory retrieval, participants must have undergone a shift in the strategy use from counting to retrieval between the first and the last training sessions. The decrease in addend slope from 486 to 45 ms/addend found in alphabet arithmetic can be paralleled to the decrease in the slope of solution times as a function of the minimum operand observed in mental addition, i.e., from 410 ms/unit increment in children to 20 ms/unit increment in adults (Groen & Parkman, 1972).

The conclusion of the seminal work of Logan and Klapp (1991), i.e., addend slope decreased in the course of learning from a significant to a non-significant level, was replicated by several other alphabet-arithmetic training experiments (e.g., Compton & Logan, 1991; Klapp et al., 1991; Zbrodoff, 1995). However, other studies found that addend slopes at the end of learning remained significant (Campbell et al., 2016; D'Eredita & Hoyer, 2010; Rawson & Touron, 2015; Wilkins & Rawson, 2010). One possible explanation for the significant addend slope at the end of training in these latter works could be related to the fact that each problem was repeated less than 48 times. Indeed, Logan and Klapp (1991) in their Experiment 3 and Zbrodoff (1995) found that solution times reached the asymptote when a problem has been presented 48 times in the course of learning. Considering that the decrease in solution times should follow a power function (Logan, 1988), an asymptotic performance implies a reliance on retrieval (see Appendix 1.A).

Furthermore, the influence of a low number of repetitions could be coupled with the high number of problems in the study set. Adopting the terms used in classical learning literature (e.g., Wolfle, 1935), we will adopt the term "constant condition" for a group of participants who study a small number of problems that are presented a large number of times and the term "varied condition" for a group of participants who study a large number of problems that are presented a small number of times. Johnson et al. (2000), Rabinowitz and Goldberg (1995), and Wilkins and Rawson (2010) contrasted the performance of varied and constant conditions and showed that solution times decreased with practice more rapidly and reached asymptote at a lower value for participants in constant than in varied condition. Moreover, the addend slope was significant in varied condition but non-significant in constant condition (Wilkins & Rawson, 2010). It is therefore possible that significant addend slope found at the end of some practice studies stemmed from the combination of a large number of problems in the study set and a limited number of repetitions.

Another factor that can determine whether addend slopes at the end of training would be significant or not concerns the similarity between problems. For example, in the work of White et al. (2007), a group of participants had to learn 12 distinct problems, constituted of 12 different letters. Another group of participants had to learn 12 confusable problems, constituted of 6 letters, each was paired with 2 different addends. Despite the same number of problems in the study sets and the same number of repetitions in both groups, addend slope at the end of training reached about 0 ms/addend in the distinct condition but about 100 ms/addend in the confusable condition (see also Haider & Frensch, 2002).

Whereas participants' performance in terms of addend slope could be influenced by the material in the study set, i.e., the number of problems to learn, the number of repetitions, and the similarity between problems in the study set, the specificity of the task could also affect performance. This was shown by Rickard (2004) who conducted an experiment involving 2 conditions. The only difference between the conditions was that he added a verbal-report task to one condition but not to the other. Surprisingly, despite the same material used in both conditions, solution times were lower and the shift to retrieval was obtained earlier in the verbal-report condition than in the other. The shift to retrieval in this study was determined by the deviation of the solution-time data from the fitted learning curve. In parallel, Rickard found that for participants in the verbal-report condition, problems reported as being solved by retrieval had shorter solution times than problems reported as being solved by algorithm. Therefore, he concluded that the lower solution times found for participants with verbal-report task was due to the more frequent use of retrieval in this condition than in the other. More importantly, he concluded that it was the additional verbal-report task that has influenced participants to privilege retrieval strategy over counting. Similarly, Wilkins and Rawson (2011) conducted an experiment with 2 conditions that only differed in the given instruction. Participants in the accurate condition were instructed to execute the task as accurately as possible and those in the speed condition as fast as possible. The authors found that both solution times and addend slopes were significantly lower in the speed than in the accuracy condition. The two latter studies suggested that performance in alphabet-arithmetic tasks could be influenced by participants' conscious decision to privilege one strategy over another.

Thus, in this section I have shown that the shift from counting to retrieval or a non-significant addend slope at the end of an alphabet-arithmetic training experiment was not always obtained. The final addend slopes seem to depend on the number of problems and the similarity among them in the study set, the number of repetitions during training, as well as the specificity of the task.

Transfer from Learnt to New Items

The instance theory of automatization (Logan, 1988) postulates that each memory trace corresponds to a specific stimulus and therefore, that learning is item specific. In the work of Logan and Klapp (1991), this was shown by the fact that addend slope for the learnt problems in the last learning session, i.e., Session 12, was not significant, but addend slope during the transfer session, i.e., Session 13, returned to a significant level of 373 ms/addend. This implied that there was no transfer from learnt to new problems. Furthermore, when participants had to work with the learnt items again in Session 14, addend slope was not significant, i.e., 51 ms/addend, implying that the significant addend slope in Session 13 was proper to the new items.

In their Experiment 3, Logan and Klapp (1991) studied transfer differently by introducing different types of new items. During the transfer phase, the learnt items, e.g., A + 2, were compared to new items called new-old and new-new items. The former new items have the same letter augends as the learnt items but different addends, e.g., A + 3, whereas the latter new items have different letter augends but the same addends, e.g., G + 2. The authors concluded that alphabet-arithmetic learning is item specific, because not only were addend slopes significantly higher for both new-new and new-old items than for learnt items, but also addend slopes for new-new and for new-old items were similar. The non-existence of transfer in alphabet-arithmetic learning was also revealed in other works (e.g., Pyke & LeFevre, 2011; Wenger, 1999; Wilkins & Rawson, 2011).

However, similar to what we have seen with addend slopes, whether transfer takes place or not could also be determined by the number of problems in the study set and the number of presentations. Rabinowitz and Goldberg (1995) as well as Wilkins and Rawson (2010) revealed that transfer to new problems in an alphabet-arithmetic experiment was obtained in varied condition but not in constant condition. The effect of varied and constant learning conditions on the success of transfer seems to be a general phenomenon in learning because it was also observed in other cognitive domains (e.g., Vakil & Heled, 2016).

Another study that is worth mentioning here is that of Campbell et al. (2016) who found transfer in alphabet-arithmetic learning. These authors used similar stimulus construction as in Experiment 3 of Logan and Klapp (1991), namely the old-new and new-new transfer items, although they were called transfer and control items, respectively. However, instead of comparing these two types of new items to the learnt items, Campbell et al. contrasted transfer to control items, and concluded that there was a transfer, because solution times were lower for transfer items than for control items. Nevertheless, two aspects of their methodology could be called into question. Firstly, because the authors did not compare new to learnt items, it is doubtful that the reported results concern a transfer phenomenon per se. Secondly, the use of solution times in the study of transfer in alphabet-arithmetic learning was not really appropriate. This is because a difference in solution times between the two conditions may involve a difference in addend slopes, a difference in intercept, or both. In other words, the transfer and control items might have different solution times but similar addend slopes. The latter case would imply that they were both solved by the same strategy. It is therefore not clear whether transfer did take place in this particular alphabet-arithmetic study.

Power-Function Law of Practice

In the domain of cognitive learning, the power function has been accepted as the ubiquitous law of practice (e.g., J. R. Anderson & Lebiere, 1998; Newell & Rosenbloom, 1981; Taatgen, 2013). The instance theory of automatization (Logan, 1988) also predicts the power function as the basis of performance during a learning phase (see Equations 1.3 and 1.4). Ignoring the asymptote constants in these 2 equations, i.e., a and a_1, and taking the logarithm of the two sides of the equations, we obtain $\log \bar{x}_{\mathrm{T}} = \log b - c \log n$ from Equation 1.3 and $\log s_{\mathrm{T}} = \log b_1 - c \log n$ from Equation 1.4. If we plot these two equations on the same log-n axis, we will have two linear functions that are parallel to each other, both having a slope of $-c$.

Thus, Logan (1988) plotted the data from Logan and Klapp (1991)'s Experiment 1 and found that the linear line representing $\log \bar{x}_{\mathrm{T}}$ was indeed parallel to the linear line representing $\log s_{\mathrm{T}}$, confirming his theory. However, the good fits were only found for problems with addends 2, 3, and 4. Nonetheless, despite the good fits for these addends, for trials towards the end of training, there was a deviation between the fit and the data, in a way that solution times were lower for the real data than for the fitted values, indicating a faster learning than expected.

Furthermore, the fit was rather poor for +5 problems, with the deviation from the fitted values being observed after 24 presentations. In fact, a model with two learning curves, i.e., one for trials prior to the 24th and another for those after, provided a better fit than a model with one learning curve. Thus, Logan concluded that the deviation found for +5 problems must have represented the shift from one strategy to another, probably from inefficient mnemonic strategy before the 24th trial to a more efficient strategy starting from the 25th trial.

Opportunistic Stopping

Addend slopes were not the only variable that could indicate the change of strategy in an alphabet-arithmetic task. Indeed, Zbrodoff (1999) proposed another way to provide support for the shift from counting to retrieval through the concept of opportunistic stopping. According to her, if counting is used, participants would take the opportunity to stop counting when they reach the correct or the proposed answer, whichever is attained first. Therefore, false equations involving a letter answer preceding the correct answer such as D + 4 = G, i.e., a within-count equation, should be rejected faster than false equations involving a letter answer following the correct answer such as D + 4 = I, i.e., an outside-of-count equation. A lack of difference in solution times between within-count and outside-of-count equations would therefore signify the use of memory retrieval.

Zbrodoff (1999) carried out two alphabet-arithmetic experiments to test for the opportunistic stopping. Experiment 1 was conducted over 1 session and involved 18 letters paired with addends 3, 4, and 5. Each combination appeared 4 times with its true answer and 4 times with false answers. Experiment 4, on the other hand, was conducted over 13 sessions and involved 9 letters paired with addends 3, 4, and 5. In each session, each combination was presented 8 times with its true answer and 8 times with false answers. In both experiments, apart from T–1 and T+1 equations, Zbrodoff also included false equations presented with two letters before the correct answer, i.e., T–2 equations such as D + 4 = F, and two letters after it, i.e., T+2 equations such as D + 4 = J.

Opportunistic stopping was obtained in Zbrodoff (1999)'s Experiment 1 as well as in Session 1 of her Experiment 4, because on average within-count equations, i.e., T–1 and T–2 equations, were rejected faster than outside-of-count equations, i.e., T+1 and T+2 equations, with a difference of 560 and 401 ms in Experiment 1 and in Session 1 of Experiment 4, respectively. Hence, she concluded that counting was used in the beginning of learning. However, the effect was not found in Session 13 of Experiment 4, where rejection times for the two types of false equations were not different, implying the use of memory retrieval. Thus, Zbrodoff asserted that her participants in Experiment 4 underwent a shift in the strategy use from counting to retrieval during the training experiment.

Instance versus Strength Theories

I have discussed in the preceding section how the instance theory of automatization (Logan, 1988) through their alphabet-arithmetic paradigm has lent support to retrieval models of mental arithmetic. As the name implies, the instance theory of automatization is an instance theory, according to which the success of memory retrieval depends on the number of instances available in long-term memory. An instance theory can be contrasted to strength theories, according to which the success of memory retrieval depends on the strength of the association between stimuli and response, or between a problem and its answer. The network-retrieval, network-interference, and overlapping-waves models described earlier are all strength theories that are applicable to the domain of mental arithmetic. In the domain of cognitive skill in general, the adaptive control of thought – rational (ACT–R) proposed by Anderson (e.g., J. R. Anderson, 1982, 1987; J. R. Anderson et al., 2004; J. R. Anderson & Lebiere, 1998) is one of the most prominent strength theories.

At the heart of ACT–R is the distinction between procedural and declarative knowledge. Procedural knowledge is based on a series of production rules, i.e., the rules of how to retrieve and use declarative knowledge to attain a goal. The production rule itself is a series of productions, i.e., an association between condition and action which is formulated as IF (condition) THEN (action). For example, the production to solve an alphabet-arithmetic problem by counting could take the form of

IF	the goal is to answer *letter* + *number* = ?
THEN	set a subgoal to increment *letter number* times
	and then report the answer

and the production to solve it by retrieval

IF	the goal is to answer *letter* + *number* = ?
	and *letter* + *number* = *newletter*
THEN	response with *newletter*

(J. R. Anderson & Lebiere, 1998, p. 22).

Declarative knowledge, on the other hand, is represented by declarative chunks, that can be acquired either by encoding information directly from the environment or by storing the results of past productions. The former corresponds to rote learning and the latter to learning-by-doing. The shift from algorithm-based to retrieval-based performance postulated by the instance theory of automatization (Logan, 1988) is an example of the creation of declarative knowledge through past productions (J. R. Anderson & Schunn, 2000).

A production rule may call a declarative knowledge and other productions stored in long-term memory. In the first example given above, the production rule calls two productions ("set a subgoal to increment *letter number* times" and "report the answer") whereas in the second, it calls a declarative chunk ("*letter* + *number* = *newletter*") and a production ("response with *newletter*"). The selection of which productions to apply depends not only on the goal to achieve but also on the probability that this goal will be achieved successfully if a particular production is chosen, the value of the goal itself, and the cost in term of time and effort of achieving the goal.

When a declarative chunk is required in a production rule, the success and latency of its retrieval depend on the level of activation of the chunk and the strength of the production that retrieves the chunk, in a way that stronger productions and higher activation level result in higher retrieval accuracy and faster retrieval time. The strength of the production depends on how recent and how frequent this production has been used in the past. The chunk activation level, on the other hand, depends on the base-level and the associative activations. Base-level activation depends on the amount of previous learning. It decays with time and therefore it is higher for more-recently and more-frequently encountered chunks. Associative activation, on the other hand, is related to the degree of association between the chunk and the current production, and depends on the number of chunks that are connected to the current production. The more chunks there are, the lower the strength of association between each chunk and the current production, and hence the lower the associative activation. This effect is known as the fan effect (J. R. Anderson & Reder, 1999; Pirolli & Anderson, 1985).

A successful production can result in an action that is exerted to the outside world or in a change of goal, leading to another production rule. A third possible result of a successful production is the creation of a new declarative knowledge through the so-called production compilation, namely the reduction of several successive productions into a smaller number of production (J. R. Anderson et al., 2004; Taatgen & Lee, 2003). For example, to solve an alphabet-arithmetic problem, the two productions

IF the goal is to answer *letter* + *number* = ?
THEN set a subgoal to increment *letter* *number* times
and then report the answer

and

IF the goal is to increment *letter* *number* times
and *letter* + *number* = *newletter*
THEN response with *newletter*

can be compiled into a single production that can be saved as a new declarative chunk.

IF the goal is to answer *letter* + *number* = ?
and *letter* + *number* = *newletter*
THEN response with *newletter*

Indeed, ACT–R could be used to model the performance in an alphabet-arithmetic task, including the shift in strategy from counting to retrieval. Johnson et al. (1998) took the data collected by Rabinowitz and Goldberg (1995) and modelled them according to the ACT–R theory. As has been explained earlier, comparing varied and constant conditions, Rabinowitz and Goldberg found faster decrease in solution times and lower asymptote in constant than in varied condition, as well as a perfect transfer in the varied condition. In the framework of ACT–R, these two conditions differed in the frequency and recency with which each problem was presented, as well as in the number of associations that have to be learnt. More precisely, in the constant condition, the frequency of presentation was higher, and hence the same problem as the current one was encountered more recently than in the varied condition, resulting in higher base-level activation and stronger association for the constant condition. Furthermore, there were fewer problems to learn in the constant condition, resulting in a weaker fan effect and hence higher associative activation than in the varied condition. With these inputs, Johnson et al. succeeded in reproducing the data of Rabinowitz and Goldberg. Their model showed that participants in the constant group have shifted to retrieval during the practice period, such that during the transfer phase the strength of the counting rules was too low to be used efficiently, resulting in much slower performance than in the varied condition.

In the domain of mental arithmetic, ACT–R has been successful in modelling not only the problem-size effect, both in terms of solution times and error rates, but also the evolution of problem-size effect in the course of learning (Lebiere, 1999; Lebiere & Anderson, 1998). Despite the major underlying difference between instance and strength theories, both the instance theory of automatization and ACT–R postulate that the eventual goal of learning is the memorisation of the association between the problems and its answer. However, in this thesis, I will concentrate on the instance theory of automatization.

1.2 Research Problems

I have mentioned earlier 4 points that were taken as the validation for the shift in strategy from counting to retrieval, i.e., the non-significant addend slopes and the disappearance of opportunistic stopping at the end of learning, the lack of transfer from learnt to new items, and the good fit to power function of the mean and standard deviation of solution times. This shift in strategy claimed in several alphabet-arithmetic studies (e.g., Compton & Logan, 1991; Klapp et al., 1991; Logan & Klapp, 1991; Zbrodoff, 1995, 1999) has been taken as support for retrieval models of mental arithmetic (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996). Despite these claims, in what follows, I will scrutinise the first 3 points to unveil several problems that will cast doubt on the purported support for the theory. Because the fourth point will not be part of this thesis, the inspection of power function as the law of learning is presented in Appendix 1.A. In addition to the 3 points, I will also present the problem of using verification tasks to study cognitive skill acquisition that is supposed to mimic learning in a real world, and the necessity to investigate individual differences in alphabet-arithmetic learning.

Addend Slope

As has been discussed earlier, alphabet-arithmetic literature suggested that high number of repetitions, distinct stimuli, and low number of problems in the study set could lead to non-significant addend slopes at the end of training. Nevertheless, another possibility has been overlooked in the literature. In fact, several alphabet-arithmetic studies have shown a discontinuity in the increase of solution times as the function of addend (e.g., Beilock & Carr, 2001; Chen et al., 2020; Compton & Logan, 1991; Logan & Klapp, 1991; Wenger, 1999; Zbrodoff, 1995, 1999). More precisely, solution times increase from the smallest to the second-largest addend and then decrease for the largest addend. Furthermore, this observation was found systematically, irrespective of the largest addend in the study set.

The decrease in solution times for problems with the largest addend could artificially reduce the addend slopes, and this might have resulted in the erroneous conclusion that a non-significant addend slope was obtained at the end of training. More precisely, if we excluded problems with the largest addend, the resulted addend slopes might still be significant at the end of training. For example, the non-significant addend slope of 45 ms/addend at the end of Logan and Klapp (1991)'s experiment would increase to 120 ms/addend if the problems with the largest addend were excluded from their analyses. A significant addend slope at the end of training may invalidate Logan and Klapp's conclusion about the shift in strategy from counting to retrieval.

Logan and Klapp (1991) explained the decrease in solution times for problems with the largest addend, i.e., +5 in their Experiment 1, by proposing that these problems were solved by a different strategy from other problems. More precisely, problems with smaller addends were solved using an automatic encoding resulted from the repeated use of counting procedure. In other words, in the course of learning, counting B + 4 = C, D, E, F would result in the automatic encoding of (B, 4, F). Problems with the largest addend, on the other hand, were solved using a deliberate mnemonic encoding strategy. For example, as a mnemonic technique to be applied specifically for problems with the largest addend, i.e., +5, Logan (1988) associated both the letter augend and letter answer to the first and last names of someone. Thus, for example, **G** + 5 = **L** was associated with **G**ordon **L**ogan. However, this explanation is not convincing, considering that the same discontinuity in solution times was also observed in Experiment 2 of Logan and Klapp, wherein participants learnt alphabet-arithmetic facts by rote learning using a well-defined mnemonic technique that was applicable to all addends. Although the instance theory of automatization allows for the creation of memory traces through different methods, i.e., deliberate memorisation and automatic encoding (Logan & Klapp, 1991), it is not clear how different methods could lead to different number of traces.

Therefore, another explanation than the two memorisation methods is required to explain the discontinuity in solution times. We advance in this thesis that instead of automatic encoding for problems with smaller addends and mnemonic technique for problems with the largest addend as proposed by Logan and Klapp (1991), it is also possible that at the end of training, problems with smaller addends were still solved by counting procedure whereas problems with the largest addend by memory retrieval. This view corresponds in fact to the mechanism described in the horse-race model. If this is the case, then when problems with the largest addend are removed from data, the resulted addend slopes at the end of learning should provide the sign of counting, i.e., still significantly different from 0. This aspect will be addressed in Chapter 2.

Opportunistic Stopping

We have described earlier that apart from a non-significant addend slope at the end of learning, another argument has been put forward by Zbrodoff (1999) to provide support for the shift in strategy from counting to retrieval. This is the so-called opportunistic stopping, i.e., longer rejection times for outside-of-count than for within-count equations. Because opportunistic stopping was observed in the beginning but disappeared in the end of practice, Zbrodoff suggested that it was a sign of a shift from counting to retrieval. However, several elements of her results cast doubts upon her conclusions. Firstly, if opportunistic stopping reflects the use of counting, then solution times should be shorter for within-count equations than for T equation and shorter for T than for outside-of-count equations. Nevertheless, whereas T–2 equations were rejected faster than other equations in Experiment 1 and in Session 1 of Experiment 4, T–1 equations were not systematically rejected faster than T+1 or T+2 equations, nor were T equations solved slower than within-count equations. Even in Session 1 when opportunistic stopping was obtained, rejection times was faster for T–1 equations than for both T+1 and T+2 equations only for +3 problems. For +4 problems, rejection times for T–1 equations were slower than for the two other false equations whereas for +5 problems, T–1 equations were rejected faster than T+2 equations. Thus, it is possible that the observed opportunistic stopping in Session 1 did not reflect the use of counting. If this is true, then the disappearance of opportunistic stopping in Session 13 may not reflect the use of retrieval neither. In other words, the use of opportunistic stopping to study the evolution of strategy from counting to retrieval could be called into question.

Secondly, Zbrodoff (1999) found that the disappearance of opportunistic stopping occurred at the same moment as when the addend slope for T equations reached non-significant value of 60 ms/addend (i.e., Session 5). However, we have proposed in the previous subsection that problems with the largest addend might artificially reduce addend slope such that it appears to be non-significant. Thus, it is possible that the disappearance of opportunistic stopping in Session 13 was also due to problems with the largest addend. This aspect will be addressed in Chapter 3.

Ecological Validity of Verification Tasks

The seminal study of Logan and Klapp (1991) and most alphabet-arithmetic studies (Compton & Logan, 1991; D'Eredita & Hoyer, 2010; Haider & Frensch, 2002; Rawson & Touron, 2015; White et al., 2007; Wilkins & Rawson, 2010, 2011; Zbrodoff, 1995, 1999) were based on a verification task, wherein participants had to decide whether the presented equation such as B + 4 = G was true or false. The use of verification tasks in alphabet-arithmetic paradigm is particularly problematic ecologically, given the fact that the paradigm was conceived to mimic the way children learn addition, and yet, children do not learn addition by way of a verification task.

In fact, there are reasons to think that a verification task might provide different results from the more-ecological production task. Firstly, Ashcraft and collaborators (Ashcraft, 1982; Ashcraft & Battaglia, 1978; Ashcraft et al., 1984) supposed that a verification task is a production task followed by a comparison task. During this comparison stage, the found-answer is compared to the proposed answer. Given this fact, Baroody (1984) asserted that solution times in verification tasks are inevitably not representative of the time it takes to solve a problem. Secondly, Campbell (1987b) argued that when the proposed answer in a verification task corresponds to the correct answer, it could serve as a priming. This implies that, assuming that retrieval is used, a correct trial in a verification task does not guarantee a correct retrieval. Additionally, solution times in a verification task depend on the verity of the equations, i.e., they are shorter for true than for false equations (e.g., Ashcraft & Battaglia, 1978; Ashcraft & Fierman, 1982; Ashcraft et al., 1984; Ashcraft & Stazyk, 1981; Campbell, 1987a; Groen & Parkman, 1972; Hamann & Ashcraft, 1985; Parkman, 1972; Parkman & Groen, 1971; Zbrodoff, 1999).

Considering the shortcomings of a verification task, it is important to test the validity of a verification task by confirming that the results are not only specific to the task but are also replicable in a production task, wherein participants are asked to provide the answer to a problem, e.g., B + 4 = ?. This replication will be addressed in Chapter 4.

Transfer from Learnt to New Items

The seminal study of Logan and Klapp (1991) concluded that learning is item specific, because there was no transfer from learnt to new items. However, they admitted that some transfer has occurred, because addend slope during the transfer session, i.e., Session 13, was lower than addend slope in Session 2, whereas a perfect item-specific learning would predict similar addend slopes in Session 13 as in Session 1. To explain this difference in addend slope, the authors suggested that participants might have developped mnemonic techniques during the learning phase that they applied readily to the new items in Session 13. However, we cannot exclude the possibility that what the participants applied readily was a procedural strategy instead of a mnemonic technique. Another explanation for the difference in addend slopes between Sessions 1 and 13 is that participants were already familiar to the task (e.g., Pirolli & Anderson, 1985; VanLehn, 1996).

Instead of explaining the source of difference in addend slope between Sessions 1 and 13, Haider and Frensch (2002) tried to explain why there was no transfer. In other words, why addend slope in Session 13 was higher than in Session 12. They argued that the lack of transfer in Logan and Klapp (1991)'s study might be due to the fact that participants were surprised to find new materials to work with, that alphabet-arithmetic learning was indeed item specific, or both. Therefore, to eliminate the effect of surprise, they ran a 2-session transfer phase and found transfer at the second transfer session, suggesting that alphabet-arithmetic learning is not completely item specific. This observation was in fact also found by Logan and Klapp (1991) in their Experiment 1, because in their Session 15, which can be considered as the second transfer session, addend slopes for new items were not different from addend slopes for learnt items. However, the large difference, albeit non-significant, in addend slopes between learnt (69 ms/addend) and new items (189 ms/addend) in Session 15 led them to exclude the possibility of transfer.

Nevertheless, even if Logan and Klapp (1991) had concluded that transfer occurred in Session 15, the way addend slopes were compared was not neat. I have shown earlier that towards the end of an alphabet-arithmetic training, there is a decrease in solution times for problems with the largest addend, that artificially reduce the addend slope. Because this decrease was only found starting in Session 4, it is very likely that addend slopes during transfer, i.e., in Session 13, was not yet influenced by problems with the largest addend. In other words, the authors compared a biased addend slope in Session 12 to an unbiased one in Session 13.

Considering the above-mentioned criticisms, it will be interesting to investigate transfer in more than one session and to study the transfer phenomenon by first excluding problems with the largest addend in the calculation of addend slopes. This problem will be addressed in Chapter 5.

Individual Differences

We have described earlier how several studies showed that depending on the task instruction, participants could assert their conscious decision to privilege one strategy over another. However, the use of different strategies could also be observed between participants without specific experimental manipulations. This was found, for example, by Haider and Frensch (2002) who run a learning phase followed by a transfer phase. Session by session, the authors monitored closely participants' solution times and categorised 19 out of 39 participants who showed a drop in solution times by more than 1100 ms from one session to the next as shifters. Note however that according to the authors, a shift does not necessarily correspond to a transition from counting to retrieval but, instead, from any inefficient to any more-efficient strategy. Interestingly, Haider and Frensch found that participants who showed a shift during the learning phase tended to also show a shift during the transfer session. They concluded therefore that the shift resulted from participants' intentional decision.

The top-down mechanisms such as revealed in the above-mentioned studies concern inter-individual differences. However, intra-individual differences in alphabet-arithmetic, i.e., the use of multiple strategies by the same individuals, were also found, as revealed thanks to protocol analyses (e.g., Chen et al., 2020; Compton & Logan, 1991; Logan & Klapp, 1991; Rickard, 1997, 2004; Zbrodoff, 1999). A more indirect way to deduce whether the same participants have used single or multiple strategies was put forward by White et al. (2007), who proposed that whereas non-significant and highly-significant addend slopes imply the use of retrieval and counting procedures, respectively, a moderately-significant addend slope of about 100 ms/addend may indicate the use of multiple strategies.

Similar to multiple strategies in mental addition (Siegler, 1996), the choice of strategy in an alphabet-arithmetic task may be adaptive. The studies of Rickard (2004) and Wilkins and Rawson (2011) showed that the choice depends on the circumstance of the task whereas the participants in the study of White et al. (2007) might base their strategy on the perceived difficulty. Considering the reported multiple strategies in an alphabet-arithmetic learning, it will be worthwhile to analysis the data not only at the sample level but also at the individual level. For example, following the results of Haider and Frensch (2002), it will be interesting to study whether the same strategies are used during learning and transfer phase. This subject will be addressed in Chapter 5.

1.3 To Count or Not to Count, That's the Question

This thesis aims at revisiting the claimed support that the instance theory of automatization (Logan, 1988) provides for the retrieval models of mental addition (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996) by means of alphabet-arithmetic training experiments and by considering the research problems discussed above. This investigation may shed light on our understanding of the evolution of counting procedures in alphabet-arithmetic learning in particular and in addition learning in general. If, after taking all the elements discussed above into account, our results still support the instance theory of automatization, e.g., addend slopes at the end of practice are not significant even when problems with the largest addend are removed from the analysis, then our results would also support the retrieval models of mental addition. However, if signatures of counting procedure are still observed at the end of practice, e.g., addend slopes at the end of practice are still significant or there is a transfer from learnt to new items, then our results would support the automated counting procedure theory (e.g., Barrouillet & Thevenot, 2013; Thevenot & Barrouillet, 2020; Uittenhove et al., 2016), which would be contrary to the conclusion of previous alphabet-arithmetic studies.

Research Questions and Hypotheses

Based on the 5 research problems described earlier, 5 research questions will be examined in this thesis. For all of the research problems, the main hypothesis defended in this thesis is based on the basic assumption that the systematic solution-time discontinuity will also be obtained, i.e., solution times for problems with the largest addend will be lower than for problems with the second-largest addend. As explained earlier, this solution-time discontinuity could be explained by the horse-race model of the instance theory of automatization.

The first question is related to addend slopes, i.e., whether removing problems with the largest addend would result in non-significant addend slope. If the residual addend slope is non-significant, then retrieval is the dominant strategy for all problems whereas if the residual addend slope is still significant then the use of counting for problems with smaller addends at the end of an extensive training cannot be discarded. This question is the subject of Chapter 2.

The second question is related to opportunistic stopping, i.e., whether opportunistic stopping really indicates the use of counting. If it does, then at the beginning of learning, two conditions should be fulfilled. First, within-count equations are solved faster than true and outside-of-count equations, and second, true equations are solved faster than outside-of-count equations. Only when these two conditions are obtained can we conclude that opportunistic stopping is compatible with the use of counting. If such a conclusion cannot be achieved, then the non-existence of opportunistic stopping at the end of learning cannot be taken as the sign of retrieval. Instead, this disappearance could be due to a phenomenon that is related to problems with the largest addend. This question is the subject of Chapter 3.

The third question is related to the replicability of verification-task results, i.e., whether results from an alphabet-arithmetic verification task are replicable in a production task or specific to a verification task. We predict that the same results will be obtained in a verification as in a production task. This question is the subject of Chapter 4.

The fourth question is related to transfer phenomenon, i.e., whether transfer from learnt to new items would be observed when 3 transfer sessions, instead of 1, are conducted. To study the transfer phenomenon, addend slopes will be calculated without taking problems with the largest addend into account. Without these problems, if retrieval is the dominant strategy then transfer will not be obtained whereas if counting is the main strategy transfer would occur by the third transfer session. This question is the subject of Chapter 5.

Finally, the fifth question is related to individual differences. Whereas the main assumption of this thesis states that problems with the largest addends are memorised, it is possible that not all participants memorise problems with the largest addend, that some participants also memorise problems with smaller addends, or that both strategies are used interchangeably by the same participants. The fifth question, that is the subject of Chapter 5, is twofold. On the one hand, we are interested in investigating whether the strategy used during the learning phase is related to that during the transfer phase. Following the finding of Haider and Frensch (2002), we predict that participants who show a discontinuity in solution times during the learning phase also show a discontinuity during the transfer phase. On the other hand, we are interested in studying whether multiple strategies are used by participants. We predict that the performance during the transfer session could serve as a strong indication about the strategy used during the learning phase. The same as for the fourth question, the performance here is measured as the difference in addend slope between the last learning session and the 3 transfer sessions, and addend slope is calculated without including problems with the largest addend. The difference from the fourth question is that transfer performance will be studied at the individual level.

Methodology

To answer the research questions and to test the hypotheses, three alphabet-arithmetic training experiments were carried out. Participants were trained with 40 problems – either the combination of 10 letters and 4 addends or 8 letters and 5 addends – over 25 or 12 sessions. The experiments with 12 training sessions were followed by 3 transfer sessions. The materials and procedure are summarised in Table 1.2. In total, 66 adult participants, all naïve to the experimental design, were recruited for the three experiments. They were rewarded for their participation with a sum of money.

To answer the first research question, the data from the true-equation trials in Experiment 1 and the data from the true-equation trials during the learning sessions in Experiment 2 were used. The data from the false-equation trials in Experiment 1 was used to answer the second research question. The third research question was answered by taking the data from the learning phase of Experiment 3. For the fourth research question, the data from the true-equation trials in Experiment 2 was used and the performance in the last learning session was compared to the 3 transfer sessions. Finally, the data from the true-equation trials in Experiment 2 and the data from Experiment 3 will be used to answer the

Table 1.2

Summary of Materials and Procedure

Experiment	Task[a]	Material		Number of trials per problem per session[b]	Number of sessions	
		Letter augends	Number addends		Learning	Transfer
1	V	10 (A–J or K–T)	2, 3, 4, 5	6 × T 6 × F	25	
2	V	8 (A–H or I–P)	2, 3, 4, 5, 6	6 × T 6 × F	12	3
3	P	8 (A–H or I–P)	2, 3, 4, 5, 6	12	12	3

Note. The cleaned data of the 3 experiments can be found on the following pages: `https://osf.io/py6wr/` (Experiment 1), `https://osf.io/3ad92/` (Experiment 2), and `https://osf.io/xtebm/` (Experiment 3).
[a] Task: P for production and V for verification.
[b] Number of trials per problem per session: T for true equations and F for false equations.

question of individual differences. The details of each experiment are described in the chapters treating the corresponding research questions.

Data Diagnostics

Particularly for the production task that has a high risk of recording errors, a preliminary step in the data treatment was judged necessary. More precisely, to provide reliable results, we had to ensure that learning had taken place. Therefore, we excluded the data of participants whose accuracy rate was less than 85% for at least 2 sessions. Furthermore, participants whose data contained high number of recording errors, i.e., more than 20% for more than 3 sessions were also excluded.

Moreover, given the fact that the alphabet-arithmetic paradigm was conceived with the assumption that participants would start the learning process by counting procedure, manifested by a significant addend slope, we also excluded the data of participants who showed non-significant addend slopes already at the beginning of learning. This diagnostic was applied to both verification and production tasks.

Lastly, in both verification and production tasks, accuracy analyses included all trials. Solution-times analyses, on the other hand, were carried out only on correctly-solved trials and after discarding extreme values from the data. The upper cut-off value was defined as trials with solution times larger than the mean for each participant and for each session plus 3 times the corresponding standard deviation. The lower cut-off value was 300 ms for the verification task and 250 ms for the production task.

Statistical Analyses

All statistical analyses were run with R language (R Core Team, 2020). The packages afex (Singmann et al., 2019) and emmeans (Lenth, 2019) were employed to run Type-III ANOVAs. A significant level of .05 was applied in all analyses.

Appendix 1.A

Power-Function Law of Practice

Logan (1988) has shown that power function could provide a good fit to the alphabet-arithmetic data from Experiment 1 of Logan and Klapp (1991). However, the good fit was limited to problems with addends 2, 3, and 4, but not to problems with addend 5, for which a deviance from the power function was observed after 24 trials. This deviance has led to two major criticisms in the literature, one concerning the appropriateness of fitting the whole data to a single learning curve and another the adequacy of power function as the law of learning.

Single versus Separate Learning Curves

To explain the deviation from the power function for +5 problems, Logan (1988) proposed that trials later than the 24th were solved using different strategy than earlier ones. To account for this change of strategy, Rickard (1997, 1999, 2004) put forward the component power laws theory (CMPL). Apart from the fact that CMPL is a strength theory, it differs from the instance theory of automatization in two other ways (Rickard, 1997, 1999). Firstly, contrary to the horse-race model adopted by the instance theory of automatization, wherein both algorithm and retrieval take part in the race all the way to the end, CMPL assumes a mixture model. More precisely, at the onset of each trial, either algorithm or retrieval strategy was selected, but not both, and only the selected strategy would finish the race.

Secondly, in the race between instances and algorithm, instance theory of automatization assumes that algorithm has a constant finishing time while the finishing time for memory retrieval decreases with practice and follows a power function, i.e.

$$x_{\mathrm{T}} = a + b\ n^{-c}. \tag{1.A.1}$$

Contrary to this, CMPL model supposes that, irrespective of the strategy selected to finish the race, the finishing times of memory retrieval and algorithm decrease both according to a power law. Thus, Equation 1.A.1 becomes

$$x_{\mathrm{T,algorithm}} = a_{\mathrm{a}} + b_{\mathrm{a}}\ n^{-c_{\mathrm{a}}}$$

$$x_{\mathrm{T,retrieval}} = a_{\mathrm{r}} + b_{\mathrm{r}}\ n^{-c_{\mathrm{r}}}$$

(Rickard, 1999). Furthermore, the overall solution times are governed by a mixture of solution times from trials solved by algorithm and trials solved by retrieval, i.e.,

$$x_{\mathrm{T,overall}} = x_{\mathrm{T,algorithm}}\ (1 - p_{\mathrm{r}}) + x_{\mathrm{T,retrieval}}\ p_{\mathrm{r}},$$

where p_{r} is the proportion of trials on which retrieval strategy was used, that also decreases as a power function of the number of trials

$$p_{\mathrm{r}} = 1 - e^{-r(n-1)}$$

. Thus, at the beginning of practice, p_{r} approaches 0 and algorithm is the dominant strategy, whereas with a large number of trials, p_{r} approaches 1 and retrieval will be the dominant strategy. r is the rate that determines how fast the shift from algorithm to retrieval is attained, with larger r implies that the shift is attained in an earlier phase during learning.

Although CMPL was put forward to explain alphabet-arithmetic learning, the question as to whether the whole data should be fitted to one or several learning curves is found in any tasks involving a purported change of strategy from algorithm or calculation to memory retrieval. In fact, by means of the hidden Markov model, Tenison and Anderson (2016) modelled this type of learning by assuming one, two, or three learning states. One learning state would correspond to the instance theory of automatization Logan (1988), two learning states to the CMPL (i.e., algorithm-based and retrieval-based states), and 3 learning states to the ACT-R theory, i.e., algorithm-based, retrieval-based, and autonomous or stimulus-response states (e.g., J. R. Anderson et al., 2004). With this method, Tenison and Anderson found that cognitive-skill acquisition that starts with algorithm-based learning is best modelled by 3 learning curves, calling hence the instance theory of automatization into question.

Furthermore, whereas Rickard (2004) proposed that the algorithm stage can be fitted with a normal distribution having a fixed mean and the retrieval stage with a power function, Delaney et al. (1998) and Haider and Frensch (2002) argued that the two stages are best fitted with two power functions. In fact, the choice of which function would provide the best fit to a learning curve is still a matter of debate.

Beyond the Power-Function Learning Curves

Although power function has been regarded widely as the ubiquitous law of learning (e.g., Newell & Rosenbloom, 1981), its appropriateness as the function governing the learning curve is not yet unanimously accepted. This is because, on the one hand, the fitting of data to a power function often involved averaged data over participants or conditions and, on the other, a power curve could result from the averaging of individual exponential curves (e.g., R. B. Anderson & Tweney, 1997; Heathcote et al., 2000; Myung et al., 2000). An exponential function, i.e.,

$$x_{\mathrm{T}} = a + b\ e^{-cn}, \tag{1.A.2}$$

is similar to a power function (see Equation 1.A.1) in that both imply that with practice, performance improves and gets closer to the asymptote, and the rate of improvement, i.e., $\partial x_{\mathrm{T}}/\partial n$, becomes smaller. However, the two functions differ in their relative learning rate (RLR), defined as the absolute value of the proportion between the rate of improvement and the amount left to be learnt. This amount is basically the right-hand side of Equations 1.A.1 and 1.A.2 without the asymptotic term, i.e., $b\ n^{-c}$ for power function and $b\ e^{-cn}$ for exponential. Thus, for the exponential function, RLR is constant, i.e., c, and this means that the rate of improvement brought about by each additional practice trial is constantly proportional to the remaining possible improvement. For the power function, on the other hand, RLR decreases with trials, i.e., c/n. This means that the rate of improvement brought about by each additional practice trial decreases faster with respect to the remaining possible improvement and consequently, towards the end of practice there will be a depletion or exhaustion in the ability to improve (Evans et al., 2018). The decreasing RLR implied by a power function was in line with the instance theory of automatization (Logan, 1988), that predicts no more improvement towards the end of learning, because problems are now supposed to be solved by retrieval.

To show that averaging exponential functions could result in a power function, Heathcote et al. (2000) reanalysed the data included in the work of Newell and Rosenbloom (1981). They found that whereas power function provides a good fit for averaged data, exponential function is more appropriate for individual data. The authors then studied other data in the learning literature, including the alphabet-arithmetic data reported in Rickard (1997). Heathcote et al. found that, in general, exponential function provides a better fit to the various collected data than power function. Furthermore, for Rickard's data in particular, the R^2 increased from .497 for power function to .619 for exponential, i.e., an improvement of 20%, which is the largest improvement among the data collected by Heathcote and collaborators.

The better fit to exponential than to power function of Rickard (1997)'s alphabet-arithmetic data (see also Evans et al., 2018) was problematic for the instance theory of automatization (Logan, 1988). This is because the exponential-function fit for the data implies a constant RLR from the beginning to the end of training, whereas the instance theory expects that there is no more learning towards the end of training. Heathcote et al. (2000) argued that one way to reconcile the difference between the theory and the data is to assume that the race between algorithm and retrieval continues until the end of training, with a condition sine qua non that "a substantial proportion of responses continue to be algorithmic throughout practice" (Heathcote et al., 2000, p. 204)).

Nevertheless, when the data of Rickard (1997) was modelled with the hidden Markov model (Tenison & Anderson, 2016), power function provides a better fit than exponential function. Note however, that Heathcote et al. (2000) assumed a one-stage model whereas Tenison and Anderson used a three-stage model.

2

Scrutinising Patterns of Solution Times in Alphabet-Arithmetic Tasks Favours Counting over Retrieval Models

Adapted from:
Thevenot*, C., Dewi*, J. D. M., Bagnoud, J., Uittenhove, K., & Castel, C. (2020). Scrutinizing patterns of solution times in alphabet-arithmetic tasks favors counting over retrieval models. *Cognition*, *200*, 104272. https://doi.org/10.1016/j.cognition.2020.104272
(**The two authors are joint first authors with equal contributions.*)

Abstract

According to associationist models, initial sequential processing of algorithmic steps is replaced through learning by single-step access to a memory instance. In an alphabet-arithmetic task, where equations such as C + 3 = F have to be verified, the shift from algorithmic procedures to retrieval would manifest in a transition from steep slopes relating solution times to addends at the beginning of learning to a flat function at the end (e.g., Logan & Klapp, 1991). Nevertheless, we argue that computation of the slopes at the end of training is biased by a systematic drop in solution times for the largest addend in the study set. In this paper, this drop is observed even when the longest training period in alphabet-arithmetic literature is doubled (Experiment 1) and even when the size of the largest addend is increased (Experiment 2). We demonstrate that this drop is partly due to end-term effects but remains observable even when end-term problems are not considered in the analyses. As Logan and Klapp suggested, we conclude that the drop is partly due to deliberate memorisation of the problems with the largest addend. In contrast, departing from Logan and Klapp, we demonstrate that, when problems with the largest addend are excluded from the analyses, the possibility that counting is still used after learning cannot be discarded. This conclusion is reached because, after this exclusion, the slopes were still significant. To conclude, our results advocate that practicing an algorithm leads to its acceleration and not to a shift from algorithmic procedures to retrieval.

Keywords: Numerical cognition; Arithmetic; Strategies; Counting; Automatization

2.1 Introduction

One of the fundamental mechanisms of learning is the possibility for basic cognitive systems to form associations between different elements in long-term memory. In humans, these associations are not only the result of perception but can also be produced by mental computation (Thorndike, 1911). These assumptions have guided the influential instance theory of automatisation formulated by Logan (1988). According to this theory, any mental operation initially executed through an algorithm eventually leads to the construction of a trace associating the elements of the operation in memory. One additional trace is created each time the mental operation is executed. After repeated practice, the number of traces of the association is high enough for this association to be retrieved from long-term memory without any further recourse to the initial algorithm. For example, repeatedly solving a problem would eventually create an association between the question and the answer. Directly retrieving this association from long-term memory is viewed as the most efficient solving strategy because it is achievable in a single step (Logan, 1988).

This theoretical framework has found one of its most perfect fields of application in the domain of mental arithmetic and simple addition problem solving. In children and in adults, solution times increase with the size of the smaller operand involved in the problem. However, whereas the slope of the regression line is about 400 ms/increment in 6-year-old children, it is only 20 ms/increment in adults (Groen & Parkman, 1972). The steep slope of 400 ms/increment undoubtedly reflects counting procedures in children but the authors considered that 20 ms/increment was too short to reflect a plausible counting mechanism in adults. Therefore, they concluded that adults usually retrieve the answers of simple additions from long-term memory but that the retrieval process fails randomly for 5% of the trials. This occasional failure would be responsible for the 20-ms/increment slope. In perfect coherence with the associationist theory, more precisely with the instance theory of automatisation, the shift from counting to retrieval during the course of development has been interpreted as the result of repeated practice of counting procedures, which would ultimately lead to the association between the operands and the answers and to the creation of arithmetic facts in long-term memory (Geary, 1996; Logan, 1988; Logan & Klapp, 1991; Siegler & Jenkins, 1989; Siegler & Shipley, 1995; Siegler & Shrager, 1984). Retrieval of arithmetic facts is therefore considered as the dominant strategy used to solve simple addition problems in adults. Nowadays, increase in solution times is not only viewed as consequences of reversal to primitive methods in case of retrieval failures but also as the result of interference and frequency effects within a retrieval network (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996). More precisely, small problems would be solved more frequently than larger ones, which would result in higher memory access for small problems. Additionally, large problems share their sums with more problems than small ones, which would result in facilitated access to small problems suffering from less interference.

However, recent studies suggest that retrieval of arithmetic facts may not be the dominant strategy in adults and that automatised and unconscious procedural counting can also account for simple addition problem solving in expert adults. This challenging hypothesis was first formulated after it was shown that problem solving is facilitated when the arithmetic sign is presented 150 ms before the operands for simple additions but not for multiplications (Fayol & Thevenot, 2012). It was inferred from these results that abstract procedures were primed by the "+" sign and subsequently used to solve addition problems. Priming effects of the "+" sign are observed starting from the age of 12–13 years, which suggests that automatised addition procedures would emerge around this developmental point (Mathieu, Epinat-Duclos, Léone et al., 2018). Fayol and Thevenot as well as Barrouillet and Thevenot (2013) suggested that these procedures could correspond to quick scrolling of ordered numerical representations such as a number line or a verbal number sequence. This interpretation finds support in neuroimaging studies showing that multiplications are associated with brain regions involved in verbal retrieval, whereas single-digit additions are associated with brain regions involved in spatial attentional processing (Mathieu, Epinat-Duclos, Sigovan et al., 2018; Zhou et al., 2007). At a behavioural level, it has also been shown that addition problems are associated with spatial shifts of attention to the right hemifield (Liu et al., 2017; Masson et al., 2018; Zhu et al., 2018) whereas no spatial shifts of attention are observed for multiplication problems (Li et al., 2018; Mathieu et al., 2016). In fact, these different results could rehabilitate an idea that has been virtually abandoned more than 30 years ago by the research community, according to which arithmetic fluency is not necessarily achieved by arithmetic fact retrieval but can also be achieved by the automatisation of rules and heuristics (Baroody, 1983, 1984, 1994, 2018).

However, the view that one of the most recurrent mental activities, namely basic arithmetic, does not always lead to associations between elements is necessarily at odds with the conception that memory retrieval is the process underlying automaticity after extensive practice. More precisely, it is in conflict with the conclusion of one of the most influential studies in the literature supporting the instance theory of automatisation, in which Logan and Klapp (1991) trained 8 participants on an alphabet-arithmetic task. Over a 12-day period, participants learnt to add the digits 2, 3, 4, and 5 either to the letters A to J or to the letters K to T. In the 13th session, participants were tested on the untrained set of the alphabet and the authors examined transfer of practice to new items. The results showed that participants improved their performance drastically from Sessions 1 to 12. In the first session, solution times increased considerably with the magnitude of the addend. The function was clearly linear, with a slope of 486 ms/addend. By the 12th session, the function flattened substantially. The slope was reduced to 45 ms/addend and the linear trend was no longer significant (see the top panel of Figure 2.1).

The authors concluded that Session 1 was dominated by counting algorithms because solution times increased linearly as a function of the addend. On the contrary, the flattened slope in Session 12 was thought to reflect memory retrieval. The authors acknowledged that their results could be attributable to an improvement in the speed of counting algorithms rather than a shift from counting to retrieval but were quick to dismiss this possibility, based on the argument that the slope in Session 12 averaged 45 ms/addend, which was viewed as too fast to be mediated by sequential access to the alphabet (e.g., C + 3 = D, E, F). However, drawing conclusions based on those 45 ms/addend could be questioned.

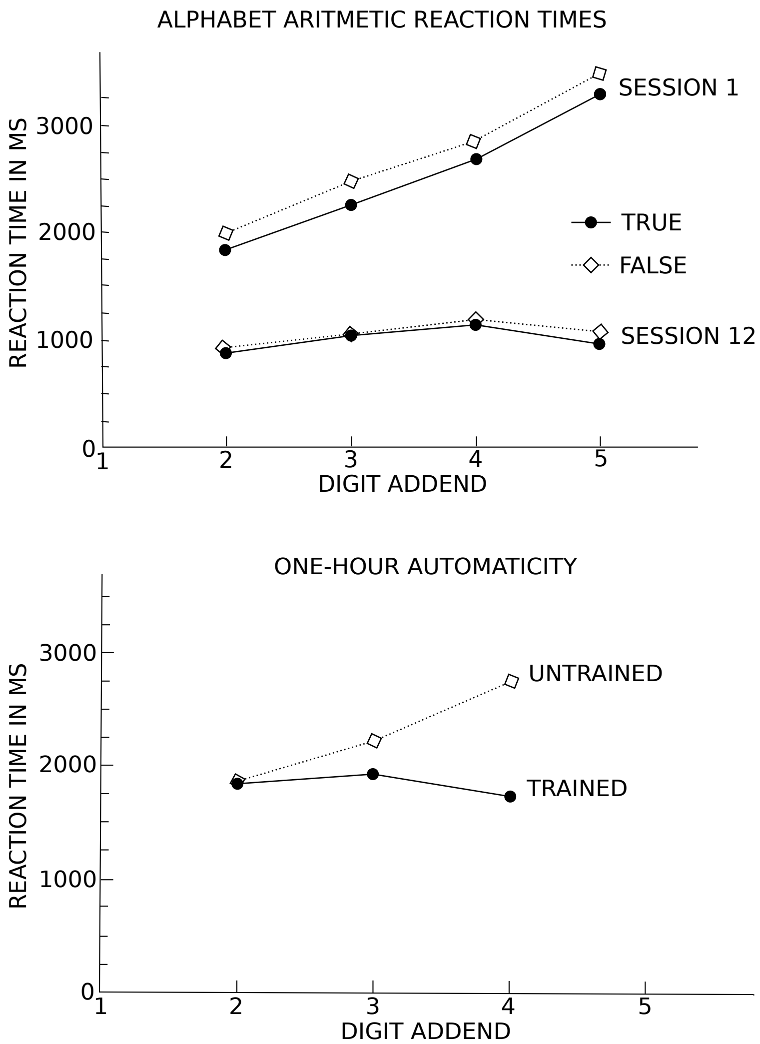
In fact, according to the authors, the slope of 45 ms/addend obtained at the end of training was the result of two different processes. Whereas the shift from procedure to retrieval was progressive across the 12 sessions for problems involving addends 2, 3, and 4, the shift was sudden for addend 5 problems between Sessions 4 and 5 (see also Logan, 1988). More precisely, after only several sessions, solution times for addend 5 problems dropped noticeably and the solving process became faster than for problems involving addend 4. The decrease in solution times for addend 5 in Session 12 is clearly visible on the top panel of Figure 2.1. The authors explained the discontinuity in solution times for +5 problems by the fact that, for these problems and from Session 4 onwards, their participants chose to use deliberate mnemonic encoding strategies. Logan and Klapp (1991) contrasted this deliberate memorisation for addend-5 problems from "a relatively "automatic" encoding of the counting episode" (p. 183) for addends 2, 3, and 4 problems. Therefore, for the authors two different mechanisms allow for the shift of strategy from counting to retrieval.

Nevertheless, the interpretation that the drop in solution times for +5 problems in Logan and Klapp (1991)'s first experiment is due to the contrast between automatic encoding strategies and deliberate memorisation is difficult to reconcile with the results of their second experiment. In this experiment, they asked their participants to study alphabet-arithmetic problems involving addends from 2 to 4 exclusively by deliberate memorisation. Nevertheless, a drop in solution times was still observed for the largest addend. Shorter solution times were indeed observed for deliberately memorised problems involving the largest addend 4 than for deliberately memorised problems with addend 3 (see the bottom panel of Figure 2.1). Thus, the drop in solution times cannot be attributable only to the fact that the problems with the largest addend are deliberately memorised. Therefore, another explanation for the discontinuity in solution times after extensive training in an alphabet-arithmetic task might be needed.

As a matter of fact, end-term effects constitute a potential candidate in order to explain that, after training, +5 problems are solved faster than +4 problems. End-term stimuli can be processed without having to pay attention to the whole problem because one of their components is systematically associated with the same response (Potts et al., 1978). For example, in a digit comparison task, the digit 9 is always the largest and the digit 1 is always the smallest of the pair of digits to be compared. Then, if the task requires the selection of the larger of two digits, 1 is never the appropriate choice but 9 is always the appropriate choice, irrespective of the other element in the pair (Moyer & Landauer, 1967).

Figure 2.1

Mean Solution Times as a Function of Addend from Experiments 1 and 2 of Logan and Klapp (1991)



Note. Top panel: Mean solution times for Sessions 1 (top lines) and 12 (bottom lines) in Logan and Klapp (1991)'s Experiment 1 for true (circles, solid lines) and false (diamonds, dotted lines) equations. Bottom panel: Mean solution times for trained (circles, solid line) and untrained (diamonds, dotted line) items in Logan and Klapp (1991)'s Experiment 2. Adapted from "Automatizing Alphabet Arithmetic: I. Is Extended Practice Necessary to Produce Automaticity?", by G. D. Logan and S. T. Klapp, 1991, *Journal of Experimental Psychology: Learning, Memory and Cognition*, *17*, p. 182 (top panel) and p. 186 (bottom panel). Copyright 1991 by the American Psychological Association.

Therefore, when a terminal marker is found in an equation or a problem, the decision process terminates without the need to examine the other elements of the equation or problem (Foos et al., 1976). Short reaction times are therefore expected for stimuli corresponding to endpoints within the study set.

More concretely, in the alphabet-arithmetic task, participants who are trained with letters A to J and with addends from 1 to 5 can decide that J + 5 = P is false because whenever P is presented, the response is always "false". There is therefore no need to examine the first part of the equation and the decision is taken very quickly. Note however, as it can be seen on the top panel of Figure 2.1, that the drop in solution times for +5 problems is also observed for problems associated with the true answer, for which partial processing is not possible before decision. Nevertheless, the answer representing the endpoint within the set of true problems also have a special status because it is presented only twice within the whole set of problems whereas the true answer for non-end-term problems is presented more than twice. J + 5 is associated with the true answer O and the only other occurrence of O is as a false answer (i.e., true answer + 1) to the problem J + 4. In contrast, the true answer for C + 4, which is G, appears as the true answer also to E + 2, D + 3, and B + 5 as well as a false answer to B + 4, C + 3, C + 5, D + 2, D + 4, E + 3, and F + 2. Therefore, the answers to end-term problems suffer from less fan effect (J. R. Anderson & Reder, 1999) than the answers to non-end-term problems, whose multiple associations produce more interference. Stated differently, the answer to end-term problems is more salient because it is shared by a limited number of problems (Campbell et al., 2016). As a consequence, end-term problems should receive more attention and be processed quicker than less salient problems.

If such end-term effects explain why a drop in solution times is associated with the largest addend in alphabet-arithmetic tasks, including end-term problems in the estimation of the slopes relating solution times to the magnitude of the addends might lead to improper conclusions. It is apparent on the top panel of Figure 2.1 that solution times for addends 2, 3, and 4 follow a clear linear function. An estimation based on this figure reveals that a slope of about 125 ms/addend is still observable after 12 practice sessions, whereas Logan and Klapp (1991) based their conclusions on a much flatter slope of 45 ms/addend when problems with addend 5 were included. Interestingly, the slope of 125 ms per letter corresponds to the overt or subvocal recitation speed reported by Landauer (1962). In this context, discarding the hypothesis that fast counting is used by participants at the end of the alphabet-arithmetic training on the basis of the 45 ms/addend slope might be premature. On a more theoretical level, this would be crucial to ensure the credibility of the automatisation of counting procedure model, according to which expertise in addition results from the acceleration of counting rather than a shift from counting to retrieval.

In order to shed light on this matter, we intended to ensure that the residual slope of about 125 ms/addend after 12 training sessions in Logan and Klapp (1991)'s experiment was not due to an insufficient number of sessions for a possible shift to retrieval. Therefore, in a first experiment, we extended Logan and Klapp's study over 25 sessions instead of 12, with a total of 300 presentations of each problem instead of 144 in the original study. According to the authors, increasing the number of problem presentations also increases the degree of automaticity at the end of training (Logan & Klapp, 1991, Experiment 3). Moreover, in order to improve the reliability of our conclusions, 19 participants instead of 8 in the original study were involved in the training program. If the drop in solution times for addend-5 problems is still observed, we will analyse the results without end-term stimuli or, in other words, without solution times related to the equations J + 5 = O and T + 5 = Y. We predict that end-term problems are responsible for the drop in solution times.

2.2 Experiment 1

Method

Participants

Nineteen students aged between 18 and 35 years were recruited at the University of Geneva. They received CHF 200 for their participation. In order to increase their motivation, participants were informed that a CHF 50 bonus would be awarded to those who obtained the best performance during the training phase. Informed consent was obtained for each of the participants.

Material and procedure

The experiment was constructed as a training study, similar to the one designed by Logan and Klapp (1991). Participants were trained on an alphabet-arithmetic verification task, wherein equations in which digits are added to letters have to be verified (e.g., A + 2 = D; FALSE). Half of the presented equations were associated with the correct letter (e.g., A + 3 = D) whereas the other half was associated with an incorrect answer (e.g., C + 4 = H). Half of the incorrect answers corresponded to the letter preceding the correct answer while the other half corresponded to the letter succeeding it. Participants were trained on equations with addends from 2 to 5, associated either with the first 10 letters of the alphabet (i.e., A to J) or with the next 10 letters (i.e., K to T). Participants were therefore trained on 40 problems and were randomly assigned to the first or second part of the alphabet.

A computerised version of the task was created using E-prime 2.0 software. Material was set up on participants' laptop for home training. The equations were presented horizontally in the center of the screen. Each trial began with a fixation point (*) presented for 500 ms, followed by the equation, which remained on the screen until the participant pressed the response key. Participants were required to press the "A" key when the presented equation was correct and the "L" key when it was incorrect. Then, the screen remained blank for 1500 ms until the onset of the next trial. Every combination of letters (A to J or K to T), addends (2 to 5), and response validity (TRUE or FALSE) was presented six times per session. Thus, every session involved 480 trials (i.e., 10 letters $\times$ 4 addends $\times$ 2 possible answers $\times$ 6 repetitions), which were divided into four blocks separated by a break. At the end of every session, the percentage of correct responses was displayed and participants had to note it down in a table. Twenty-five training sessions took place over 25 to 30 days, because participants were allowed to have a one-day break during the week.

Results

Accuracy

Overall, the percentage of correct responses was 98% ($SD = 2\%$), increasing from 96% in Session 1 to 99% in Session 25. A 2 (Letter Set: A to J vs. K to T) $\times$ 25 (Session: 1 to 25) $\times$ 4 (Addend: 2 to 5) mixed-design ANOVA was conducted on correct responses with Letter Set as the between measure. There was no effect of Letter Set and this variable did not interact with Session or Addend (Fs < 1). Therefore, we collapsed the two letter sets and conducted a 25 (Session: 1 to 25) $\times$ 4 (Addend: 2 to 5) repeated-measures ANOVA, which confirmed that the rates of correct responses increased across sessions ($F(24, 432) = 7.52$, $\eta_p^2 = .29$, $p < .001$) and revealed that they decreased as a function of addends ($F(3, 54) = 3.10$, $\eta_p^2 = .15$, $p = .03$). There was no interaction between Session and Addend ($F < 1$). Because error rates were low, the remainder of the result section will focus on solution times.

Solution times

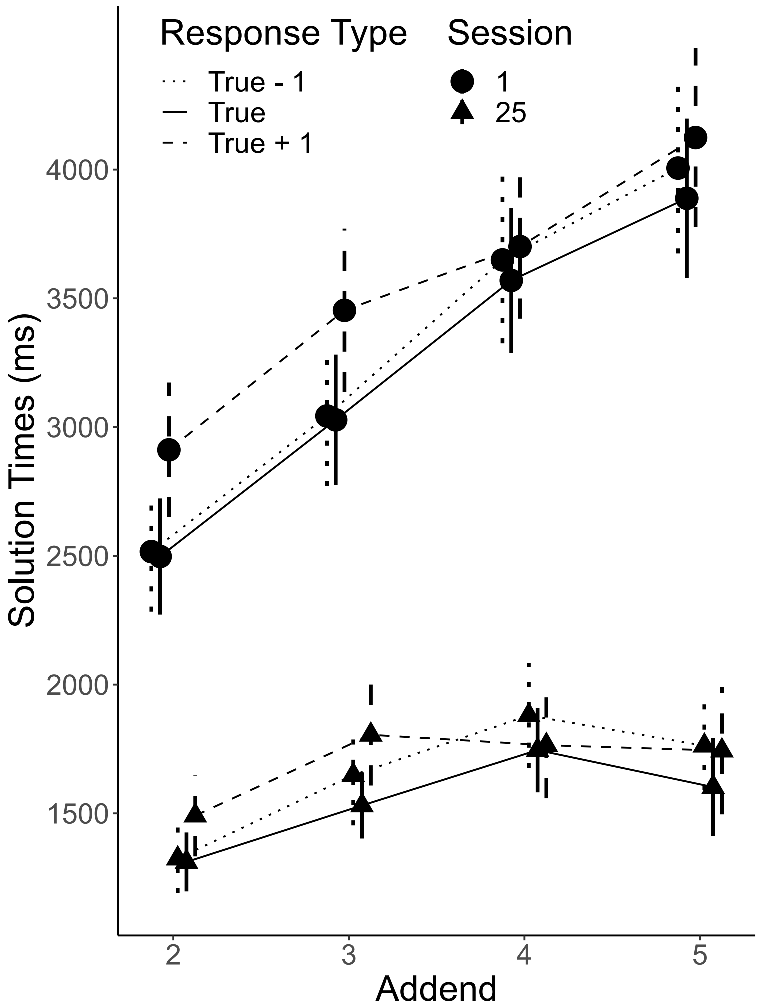
We observed that not only were solution times modulated by the type of answer presented to participants, i.e., true or false, but also by the type of false answer, i.e., +1 or −1 from the correct answer (see Figure 2.2). Therefore, in order to draw conclusions from the cleanest results, our analyses will concern only the true equations correctly solved by participants. Before conducting the analyses, 1.3% of the solution times that could not correspond realistically to processing time of the whole problem, i.e., shorter than 300 ms, or that were too large to correspond to processing time only, i.e., longer than mean + 3 *SD* per participant and per session, were identified as outliers and were discarded from the analyses.

A 2 (Letter Set: A to J vs. K to T) × 25 (Session: 1 to 25) × 4 (Addend: 2 to 5) mixed-design ANOVA with Letter Set as the between measure was conducted on solution times. There was no effect of Letter Set ($F(1, 17) = 1.28$, $p = .27$) or interactions between Letter Set and Session or Addend (Fs < 1). After collapsing the two letter sets, we conducted a 25 (Session: 1 to 25) × 4 (Addend: 2 to 5) repeated-measures ANOVA. The mean solution times across sessions was 1840 ms ($SD = 1000$ ms) and significantly decreased over sessions, from 3246 ms to 1547 ms ($F(24, 432) = 39.10$, $\eta^2_p = .68$, $p < .001$). A main effect of Addend was also found ($F(3, 54) = 26.22$, $\eta^2_p = .59$, $p < .001$), showing that solution times generally increased as a function of addends (i.e., 1461, 1821, 2074, and 2035 ms for Addend 2, 3, 4, and 5 respectively).

More important in our results was the interaction between Session and Addend ($F(72, 1296) = 8.94$, $\eta^2_p = .33$, $p < .001$, see Figure 2.3) showing that the effect of Addend decreased across sessions. Nevertheless, when the addend slope was calculated for each participant and each session, it remained significant from the beginning to the end of training (Session 1: $M = 472$ ms/addend, $SE = 41$,

Figure 2.2

Mean Solution Times as a Function of Addend and Response Type for Sessions 1 and 25 in Experiment 1



Note. Mean solution times as a function of addends for Sessions 1 (circles) and 25 (triangles) according to the nature of the presented answer, i.e., True (solid lines), True−1 (dotted lines) and True+1 (dashed lines) equations in Experiment 1. Error bars represent standard errors.

$p < .001$; Session 12: $M = 163$ ms/addend, $SE = 42$, $p = .001$; Session 25: $M = 108$ ms/addend, $SE = 45$, $p = .03$). Whereas the difference in addend slope between Session 1 and Session 12 was significant ($t(18) = 4.92$, $p < .001$), this was not the case between Sessions 12 and 25 ($t(18) = 1.85$, $p = .08$).

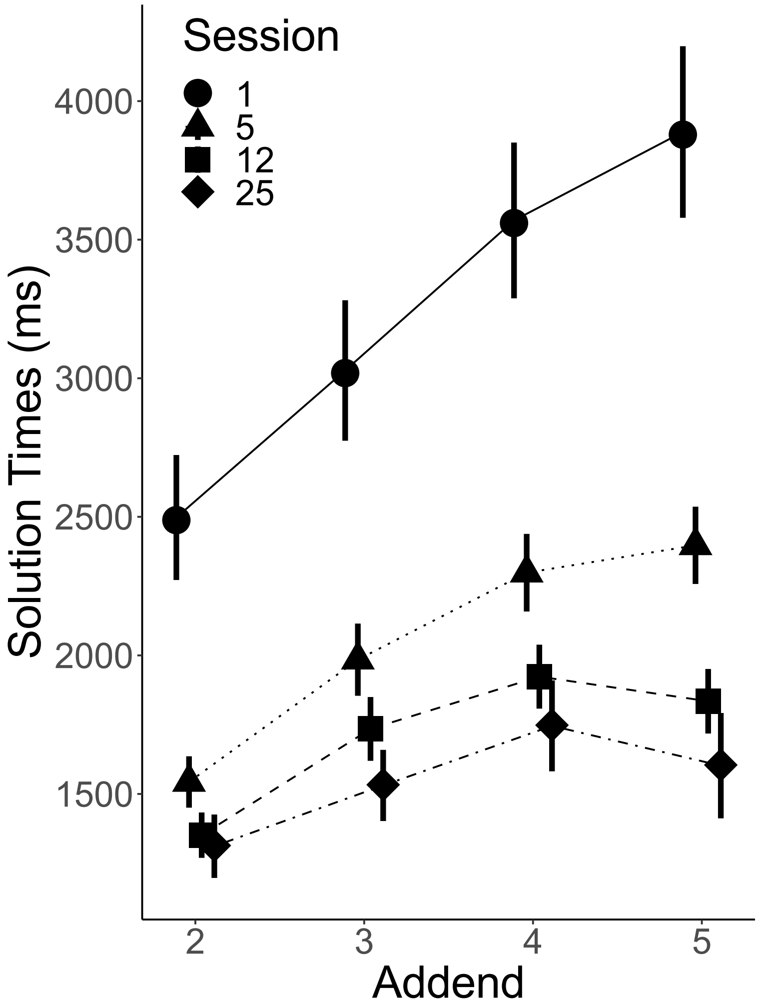
In order to determine whether solution times increased regularly with addends or, in other words, whether the effect of addend on solution times followed a linear trend, we carried out a series of contrasts with Holm correction. In Session 1, the linear ($t(18) = 11.53$, $p < .001$) and quadratic ($t(18) = -2.57$, $p = .04$) trends were significant whereas in Session 25, the linear trend became marginal ($t(18) = 2.40$, $p = .055$) but the quadratic trend remained significant ($t(18) = -2.91$, $p = .03$).

In accordance with this quadratic trend, and similar to the results of Logan and Klapp (1991), we found a discontinuity in solution times at +4 problems. Descriptively, solution times increased noticeably from addends 4 to 5 in the first session (+319 ms), but decreased in the middle session (−89 ms), and even more in the last session (−144 ms). As planned, in order to investigate the possible explanation of this decrease by end-term effects, we removed J + 5 and T + 5 problems from the data set. However, even in this case, +5 problems were solved faster than +4 problems in Session 12 (−44 ms) and Session 25 (−92 ms). When we recalculated the addend slopes without J + 5 and T + 5 problems, we found that they reached 180 ms/addend in Session 12 (against 163 ms/addend when all problems were considered) and 128 ms/addend in Session 25 (against 108 ms/addend when all problems were considered).

In order to further investigate the discontinuity in solution times at +4, we plotted the difference in solution times between +5 and +4 problems against sessions (see Figure 2.4, in which negative differences correspond to shorter solution times for +5 than for +4 problems). It can be seen that problems with addend 5 were solved faster than problems with addend 4 from Session 6 until

Figure 2.3

Mean Solution Times as a Function of Addend for Sessions 1, 5, 12, and 25 in Experiment 1



Note. Mean solution times as a function of addends for Sessions 1 (circles, solid line), 5 (triangles, dotted line), 12 (squares, dashed line), and 25 (diamonds, dot-dashed line) for true trials in Experiment 1. Error bars represent standard errors.

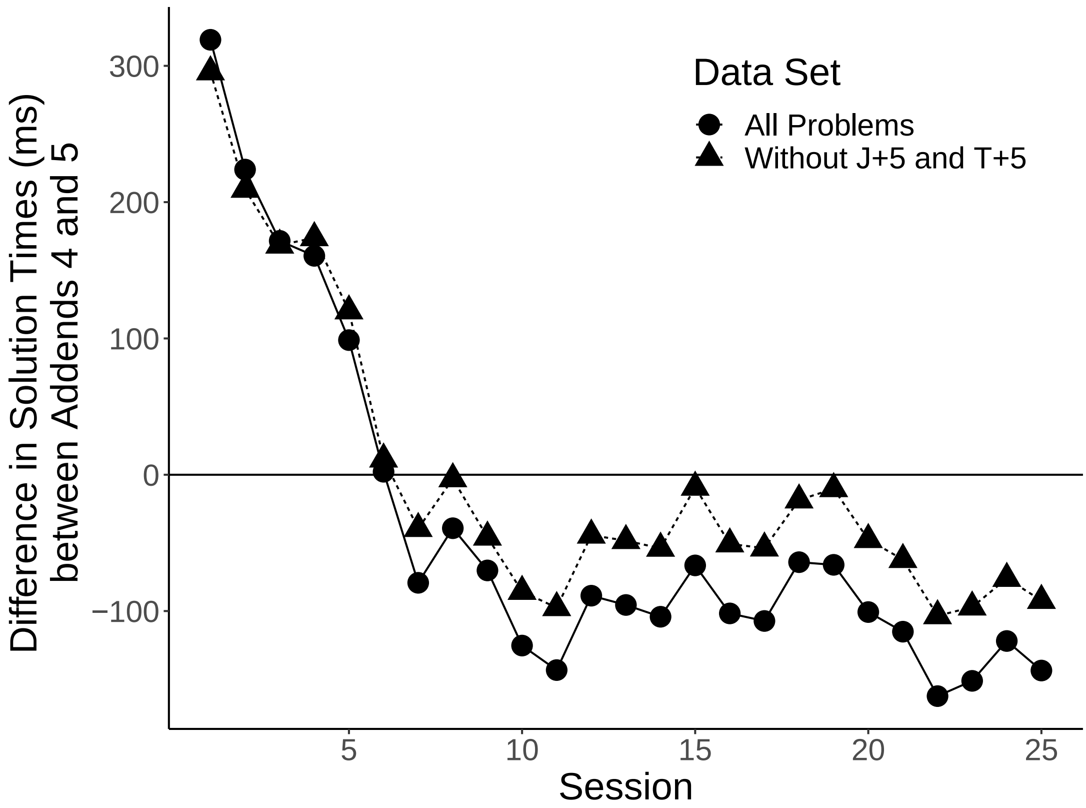
the end of the training and for every single session. This was true when we included all the problems as well as when we excluded the end-term problems. The increasing negative difference in solution times between addends 5 and 4 across sessions allows us to explain the increasing importance of the quadratic trend over the linear one.

Therefore, at least from Session 6 onwards, +5 problems were obviously not processed similarly to the other problems. This is the reason why we recalculated the addend slopes from addend 2 to addend 4 and compared them to the addend slope when all addends were considered. In Session 1, the addend slope for addends 2 to 4 corresponded to 536 ms/addend and differed significantly from the addend slope of 472 ms/addend for addends 2 to 5 ($t(18) = 2.64$, $p = .02$). In Session 12, the addend slope for addends 2 to 4 corresponded to 286 ms/addend and differed significantly from the addend slope of 163 ms/addend for addends 2 to 5 ($t(18) = 2.75$, $p = .01$). Finally, in Session 25 the addend slope for addends 2 to 4 corresponded to 217 ms/addend and differed significantly from the addend slope of 108 ms/addend for addends 2 to 5 ($t(18) = 3.37$, $p = .003$). When +5 problems were not considered, the difference in addend slope between Session 1 and Session 12 was significant ($t(18) = 4.08$, $p < .001$) but this was not the case between Sessions 12 and 25 ($t(18) = 1.82$, $p = .09$).

Interestingly, scrutinising the whole data set at an individual level revealed that, across training sessions, not all participants showed a break in solution times at addend 4. Therefore, based on participants' solution times for +4 and +5 problems, we created two groups. Six participants who never showed a break across sessions were classified as non-breakers. Six participants who showed a systematic break from one session (from as early as in Session 1 to as late as in Session 17) until the end of training were classified as breakers. Solution times for these two groups are presented in Figure 2.5. The 7 remaining participants did not show such a consistent pattern across sessions. Instead, they presented

Figure 2.4

Difference in Solution Times as a Function of Session in Experiment 1



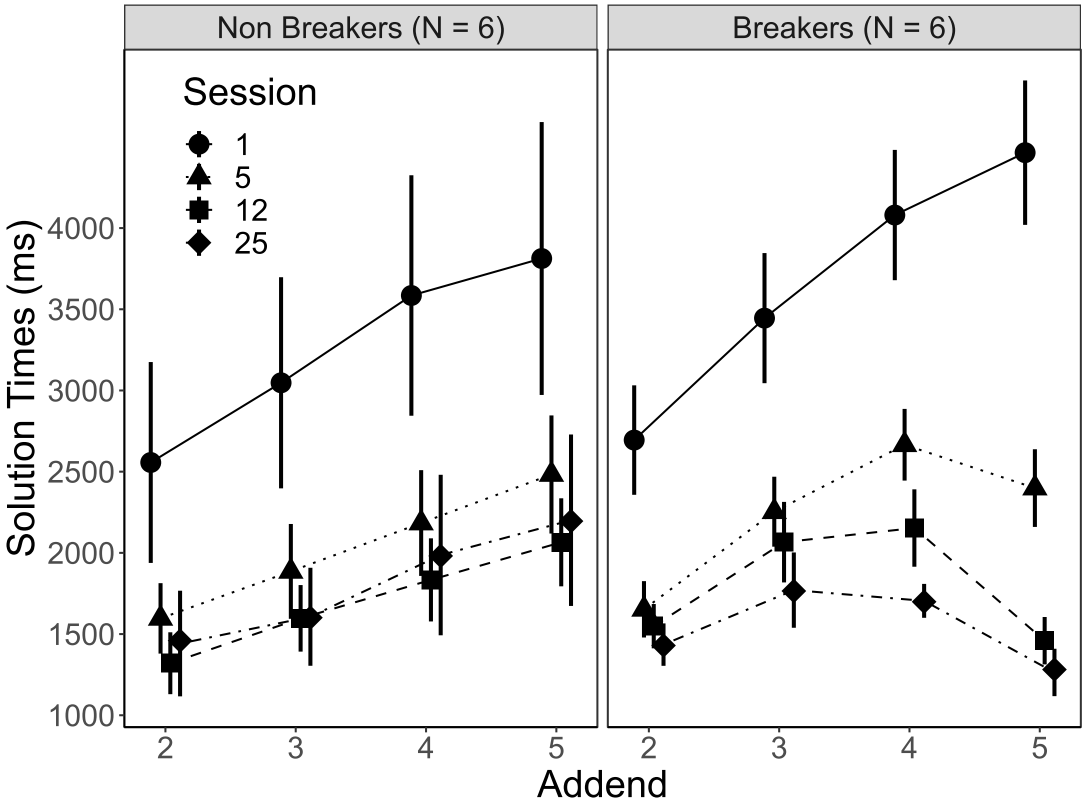
Note. Difference in solution times for true equations between problems with addends 4 and 5 across the 25 sessions in Experiment 1, when all problems are included (circles, solid line) and when the end-term problems are excluded (triangles, dotted line). Because of large variations, for the sake of clarity, error bars are not represented in the figure.

aberrant behaviours with the appearance and disappearance of the break across sessions. In order to ensure that our categorisation into breakers and non-breakers was relevant, we conducted a series of one-tailed paired-sample t-tests on solution times in Session 25 and showed that for breakers, $+4$ problems were solved faster than $+5$ problems, ($t(5) = 5.38$, $p = .001$) and that for non-breakers, $+4$ problems were solved slower than $+5$ problems ($t(5) = -3.83$, $p = .006$).

We conducted a 4 (Addend: 2 to 5) $\times$ 2 (Group: breakers vs. non-breakers) mixed-design ANOVA on solution times in Session 25, with Group

Figure 2.5

Mean Solution Times as a Function of Addend for Sessions 1, 5, 12, and 25 for Breakers and Non-Breakers in Experiment 1



Note. Mean solution times as a function of addends for Sessions 1 (circles, solid line), 5 (triangles, dotted line), 12 (squares, dashed line), and 25 (diamonds, dot-dashed line) in Experiment 1, for true trials in non-breakers (left panel) and breakers (right panel). Error bars represent standard errors.

as the between measure. There was no effect of Group ($F(1, 10) < 1$) but the interaction between Group and Addend was significant ($F(3, 30) = 7.24$, $\eta^2_{\text{p}} = .42$, $p < .001$). A series of contrasts with Holm correction revealed that for non-breakers, the effect of Addend was linear and significant ($t(10) = 3.44$, $p = .02$) while for the breaker group, the effect was not linear ($t(10) = -0.75$, $p = 94$) but quadratic ($t(10) = -5.54$, $p < .001$).

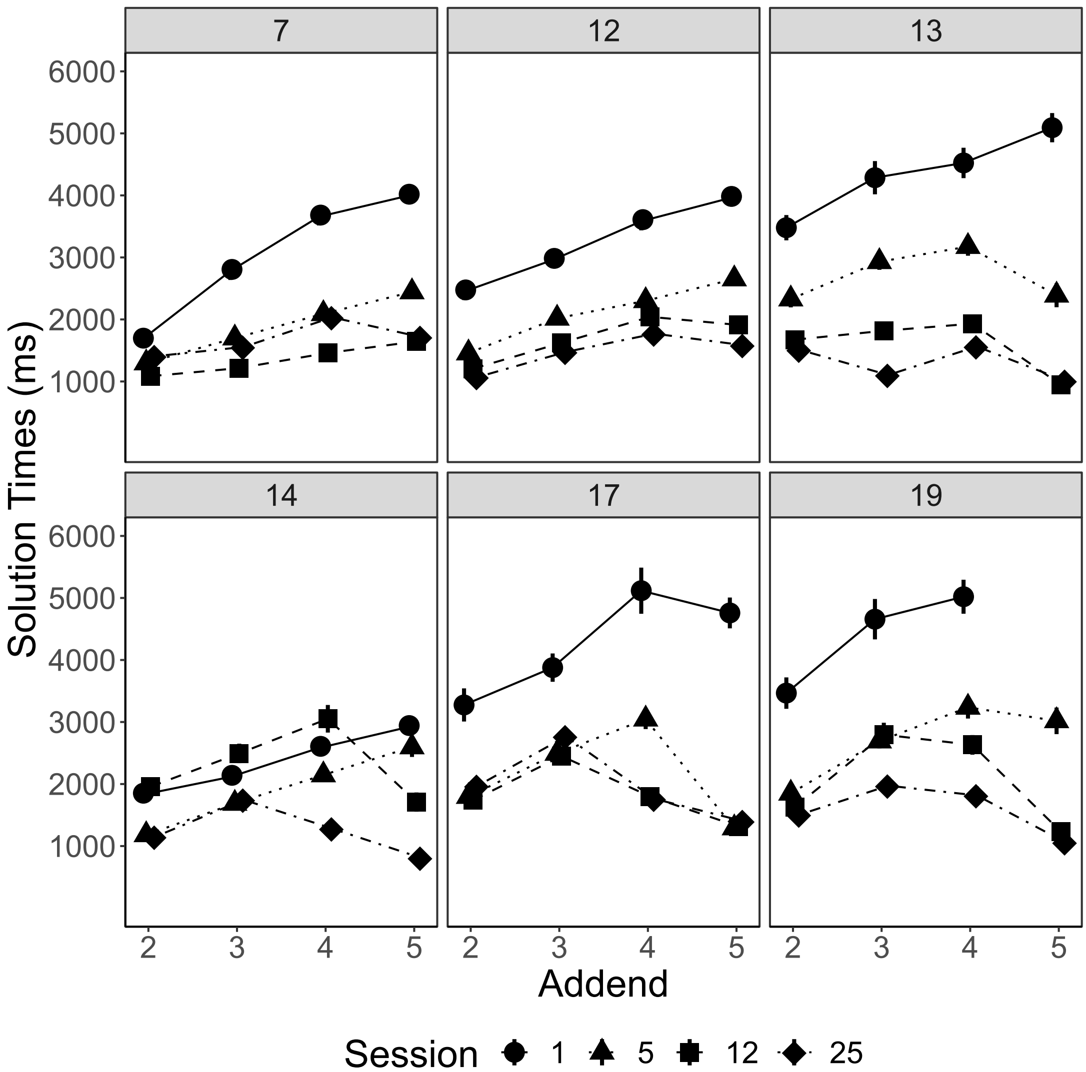
At a more descriptive level, with regards to Figure 2.5, it seems that +5 problems are not the only problems that are processed differently by the populations of breakers and non-breakers. Whereas, in non-breakers, solution times for +4 problems are consistent with the linear trend, in breakers, those problems do not conform to the linear function, because they are solved quicker (1705 ms) than +3 problems (1771 ms). In order to shed light on this matter, we examined the data in breakers at an individual level (Figure 2.6). Interestingly, except for one participant with a peculiar pattern of solution times, i.e., participant 13, two different patterns of behaviour emerged from the data. Some of the participants showed the already described discontinuity at +4 whereas some others presented the discontinuity at +3.

Discussion

This first experiment was conducted in order to address some issues about the conclusions drawn by Logan and Klapp (1991) that individuals retrieve the results of alphabet-arithmetic problems after a training experiment. To this aim, we trained 19 participants using the same material as the authors, i.e., same letters associated with same addends from 2 to 5, over 25 instead of 12 sessions in the original study. After 12 sessions, we obtained the same pattern of results as Logan and Klapp, with an increase in response times from addends 2 to 4 and a decrease between addends 4 and 5. The same pattern was also observed after 25 sessions, which ensures that these results were not due to a lack of training. This point is important considering that, from Session 1 onwards, our 19 participants were slower than the 8 participants involved in Logan and Klapp's study. The addend slope in Session 12 was also steeper in our study (163 ms) than in the original one (45 ms). It could have been suggested that after 12 sessions, our participants were not trained enough and, consequently, their behaviour could not be compared to Logan and Klapp's participants'. Nevertheless, a lack of practice cannot explain our results because, as attested by the absence of statistical

Figure 2.6

Mean Solution Times as a Function of Addend for Sessions 1, 5, 12, and 25 for Individual Breakers in Experiment 1



Note. Mean solution times as a function of addends for Sessions 1 (circles, solid line), 5 (triangles, dotted line), 12 (squares, dashed line), and 25 (diamonds, dot-dashed line) for participants classified as breakers in Experiment 1. Number above each panel indicates participant number. Error bars represent standard errors.

difference in addend slope between Sessions 12 and 25, further practice does not seem to impact the pattern observed at mid-training.

More central to our study, we questioned the conclusions in relation to the decrease in solution times observed for +5 problems in Logan and Klapp (1991)'s study. We suggested that this phenomenon could be related to end-term effects, with endpoints in a set of stimuli being particularly salient and therefore processed faster. Our results show that removing end-term problems, i.e., J + 5 and T + 5 for A to J and K to T set of letters, respectively, attenuated the discontinuity in solution times but did not make it disappear. As evident in Figure 2.4, this discontinuity appears from Session 6 onwards and remains present until the end of training, even when end-term problems were excluded. Therefore, end-term effects cannot fully explain the pattern of results observed in alphabet-arithmetic tasks and our results support the interpretation of Logan and Klapp (1991) that the discontinuity in solution times for +5 problems can be due to deliberate mnemonic encoding strategies for +5 problems.

Thus, +5 problems were not processed in the same way as the other problems during the course of the training. This is the reason why we recalculated the addend slopes by excluding those problems. Our analysis revealed steep addend slopes of 217 ms/addend and a clear linear pattern of solution times as a function of addends (Figure 2.3). This pattern of results is compatible with the idea that participants still scan the letters of the alphabet one by one at the end of training for problems involving addends from 2 to 4. Indeed, the mental recitation rate of 115 ms per letter measured by Logan and Klapp (1991) at the end of their experiment is considerably lower than the addend slope of 217 ms/addend observed in our experiment. This point will be taken up in our General Discussion.

In addition to classical analyses at the sample level, we investigated potential differences in behaviour between our participants. This approach revealed the existence of two contrasted populations. While one third of our participants presented aberrant behaviours with the appearance and disappearance of the break across sessions, one third of our participants presented a break at +4 at one point of the training until the end, whereas another third never presented this break. Within the breakers group, we went even deeper in individual observations and identified a sub-population of 3 participants who presented a break at +3. Highly interestingly, as observable in Figure 2.6, those participants presented the +3 break after they presented the +4 break in earlier sessions. If, as we concluded following Logan and Klapp (1991), a break in solution times is the sign of deliberate memorisation, our results show that some of our participants memorise the material from the largest problems to the smallest. As far as non-breakers were concerned, the addend slope for addends 2 to 5 was 265 ms/addend. Therefore, one third of participants, even after having been exposed to twice as many as the number of problem presentations in Logan and Klapp's experiment, present a behaviour that is perfectly compatible with the idea that they still count at the end of training and never deliberately memorise any category of problems.

Moreover, at an intra-individual level, we do not exclude the possibility that breakers still count for some +5 problems. This would explain why solution times for +5 problems are still higher than for +2 and +3 problems (see Figures 2.2 and 2.3). In fact, such mixed strategies could also be evoked for problems that are solved primarily by counting strategies. We do not exclude the possibility that non-breakers retrieve the results of a minority of problems. These considerations echo Siegler (1996)'s overlapping waves model, in which at a given point in development, several strategies can be used to solve the same problem.

All in all, our results do not show evidence against the fact that counting is still the dominant strategy at the end of an extensive alphabet-arithmetic training. When retrieval is probable, it concerns only some participants and is used preferentially for problems associated with the largest addends. A simple test of this interpretation is to present participants with problems associated with larger addends than in previous training studies. Therefore, in a second experiment, we asked participants to solve alphabet-arithmetic problem with addends from 2 to 6 instead of 2 to 5. If our understanding of occasional retrieval during alphabet-arithmetic training is correct, some participants should present a break in solution times at +5 problems rather than at +4 as observed in Experiment 1. Alternatively, it is possible that it is from +5 problems onwards that participants decide not to count and therefore deliberately memorise +5 and +6 problems simultaneously during the course of the experiment. In this case, a decrease in solution times should be observable for +5 problems and solution times should be similar for +6 and +5 problems. Because in this first experiment the break in solution times at addend 4 was observable from the 6th session, in the second experiment we stuck to the original number of 12 sessions used by Logan and Klapp (1991).

2.3 Experiment 2

Method

Participants

Twenty-four students (13 females) aged between 18 and 29 years were recruited by means of the student-job websites of the University of Lausanne and the Swiss Federal School of Technology in Lausanne. All participants were native French speakers and they received CHF 120 for their participation. Informed consent was obtained for each of the participant.

Material and procedure

Participants were trained on equations with addends from 2 to 6 paired with either the first 8 letters of the alphabet (A to H) or the next 8 letters (I to P). The same number of problems, i.e., 40, number of trials per session, i.e., 480, as well as number of problem presentations per session, i.e., 6 times with TRUE and 6 times with FALSE answers, as in Logan and Klapp (1991) and as in our Experiment 1 were kept. The same design as in Experiment 1 was used, except that we took an additional methodological precaution and counterbalanced the response keys across participants. Thus, half of the participants were required to press the “A” key when the presented equation was correct and the “L” key when it was incorrect, while the other half were required to press the opposite keys. Moreover, in order to better control the randomisation, the 480 trials per session in the current experiment were divided into 3 identical blocks of 160 trials, instead of 4 blocks with a different random mix of problems in Experiment 1. The presentation of problems was still fully randomised within each block.

In contrast to Experiment 1, in order to reduce potential experimental noise, participants were trained at the laboratory for 12 sessions on 12 consecutive working days. In order to maximise the efficiency of training, they were required to train at home during the week end by verifying 160 equations presented on paper as quickly as possible. This corresponded to one block of the experiment.

Results

Three participants who presented flat addend slopes from the beginning of training, i.e., Session 1 or 2, were excluded from our analyses. The results of these participants could not be informative for our research question, which concerns the evolution of counting strategy after extensive training. These participants obviously did not count at the beginning of the study and were therefore out of the scope of our purpose.

Accuracy

Overall, the percentage of correct responses was 96% (SD = 5%), from 95% in Session 1 to 96% in Session 12. A 2 (Letter Set: A to H vs. I to P) × 12 (Session: 1 to 12) × 5 (Addend: 2 to 6) mixed-design ANOVA with Letter Set as the between measure was conducted on correct responses. The main effect of Letter Set was not significant ($F(1, 19) = 1.01$, $p = .33$) and this variable did not interact with Session or Addend (Fs < 1). Therefore, we collapsed the two letter sets and conducted a 12 (Session: 1 to 12) × 5 (Addend: 2 to 6) repeated-measures ANOVA. It confirmed that the rates of correct responses increased across sessions ($F(11, 220) = 2.06$, $\eta^2_p = .09$, $p = .02$) and revealed that they decreased as a function of addends ($F(4, 80) = 9.04$, $\eta^2_p = .31$, $p < .001$). The interaction between Session and Addend did not reach the .05 significance level ($F(44, 880) = 1.33$, $p = .08$). As in the first experiment, due to high percentage of correct responses, the remainder of the result section will focus on solution times.

Solution times

As in Experiment 1, we conducted our analyses on true equations only and we removed the values shorter than 300 ms and longer than mean + 3 SD per participant and per session, which resulted in 1.3% of discarded data.

A 2 (Letter Set: A to H vs. I to P) × 12 (Sessions: 1 to 12) × 5 (Addend: 2 to 6) mixed-design ANOVA with Letter Set as the between measure was conducted on solution times. We did not find an effect of Letter Set or an interaction between this variable and Session or Addend (all Fs < 1). After collapsing the two letter sets, we conducted a 12 (Session: 1 to 12) × 5 (Addend: 2 to 6) repeated-measures ANOVA. The mean solution time across sessions was 2023 ms ($SD = 1056$ ms) and significantly decreased over sessions, from 3198 ms in Session 1 to 1631 ms in Session 12, $F(11, 220) = 93.93$, $4\eta_p^2 = .82, 4p < .001$. A main effect of Addend was also found ($F(4, 80) = 121.91$, $\eta_p^2 = .86$, $p < .001$), showing that solution times increased as a function of addends (i.e., 1436, 1802, 2085, 2367, and 2428 ms for Addend 2, 3, 4, 5, and 6 respectively).

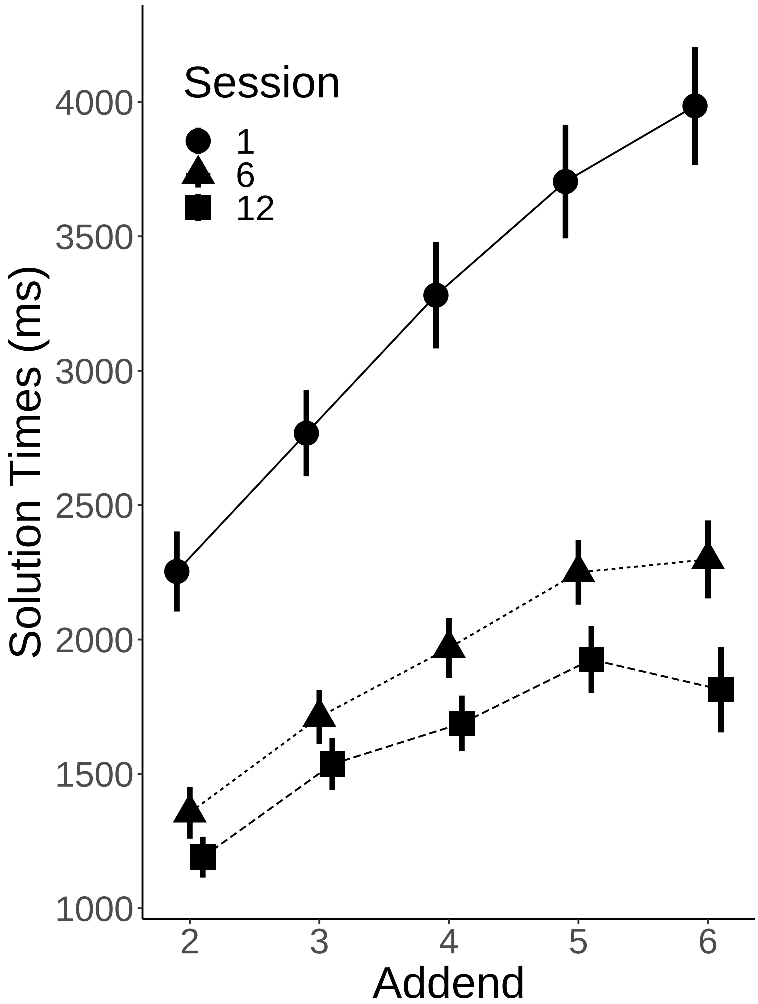
More interestingly, the interaction between Session and Addend was significant ($F(44, 880) = 20.40$, $\eta_p^2 = .50$, $p < .001$, see Figure 2.7). Furthermore, when the addend slope was calculated for each participant and each session, it remained significant from the beginning until the end of the training (Session 1: $M = 441$ ms/addend, $SE = 30$, $p < .001$; Session 6: $M = 243$ ms/addend, $SE = 20$, $p < .001$; Session 12: $M = 163$ ms/addend, $SE = 24$, $p < .001$). The difference in addend slope between Session 1 and Session 12 was significant ($t(20) = 7.99$, $p < .001$).

In order to determine whether solution times increased regularly with addends or, in other words, whether the effect of addend on solution times followed a linear trend, we carried out a series of contrasts with Holm correction. In Session 1, the linear ($t(20) = 14.62$, $p < .001$) and quadratic ($t(20) = -3.80$, $p = .003$) trends were significant. In Session 12, both trends were still significant ($t(20) = 6.80$, $p < .001$ for linear trend and $t(20) = -5.27$, $p < .001$ for quadratic trend).

Highly interestingly, in contrast to Experiment 1, we found a discontinuity in solution times at +5 instead of at +4 problems. Stated differently, whereas in Experiment 1, solution times decreased from addend-4 to addend-5 problems, solution times in this second experiment decreased from addend-5 to addend-6 problems. Figure 2.8 shows that descriptively, solution times increased noticeably from addends 5 to 6 in the first session (+281 ms) but were closer to each other in the middle session (+48 ms) and decreased in the last session (−112 ms).

Figure 2.7

Mean Solution Times as a Function of Addend for Sessions 1, 6, and 12 in Experiment 2



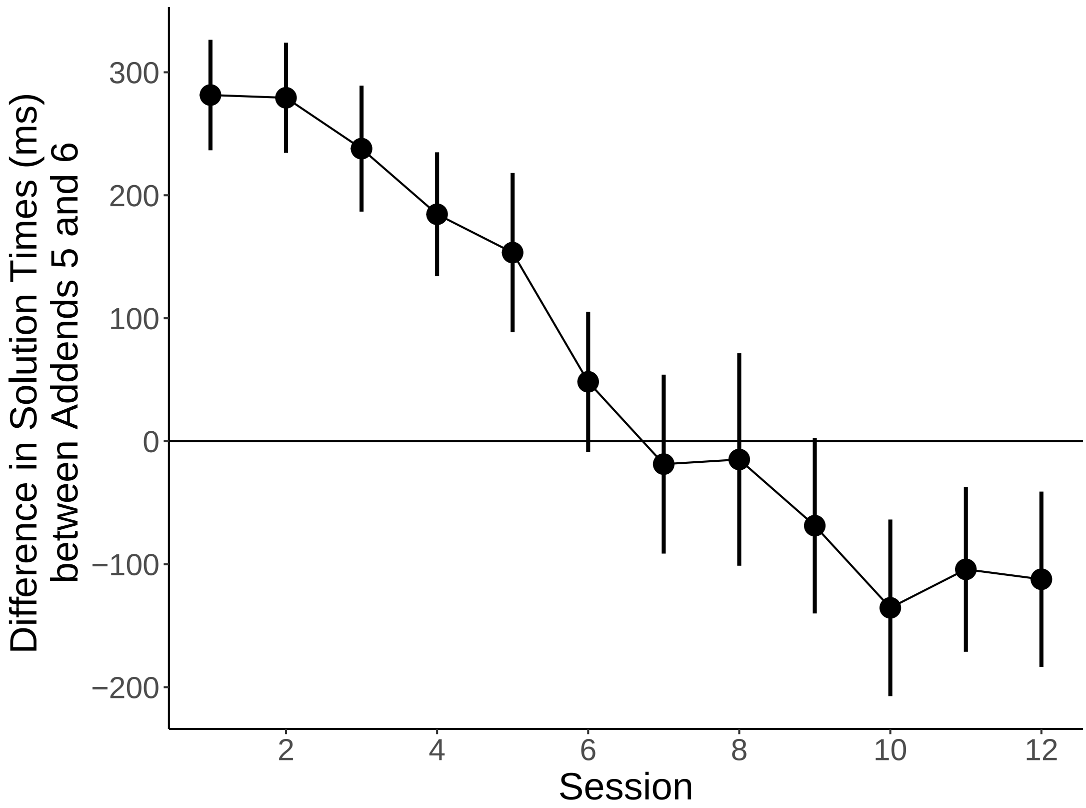
Note. Mean solution times as a function of addends for Sessions 1 (circles, solid line), 6 (triangles, dotted line), and 12 (squares, dashed line) for true trials in Experiment 2. Error bars represent standard errors.

When the addend slopes were recalculated without the largest addend, we found that they were significantly steeper than when all the addends were considered. In Session 1, the addend slope for addends 2 to 5 corresponded to 487 ms/addend and differed significantly from the addend slope of 441 ms/addend for addends 2 to 6 ($t(20) = 3.53$, $p = .002$). In Session 6, the addend slope for addends 2 to 5 corresponded to 294 ms/addend and differed significantly from the addend slope of 243 ms/addend for addends 2 to 6 ($t(20) = 4.44$, $p < .001$). In Session 12, the addend slope for addends 2 to 5 corresponded to 236 ms/addend and differed significantly from the addend slope of 163 ms/addend for addends 2 to 6 ($t(20) = 5.03$, $p < .001$). When +6 problems were not considered, the difference in addend slope between Session 1 and Session 12 was significant ($t(20) = 6.18$, $p < .001$).

Similar to Experiment 1, a closer look at the data at individual level showed that not all participants showed a discontinuity in solution times. Therefore, based on participants' solution times for +5 and +6 problems, we created two groups. Four participants who never showed a break were classified as non-breakers. Seven participants who showed a systematic break from one session (from as early as in Session 1 to as late as in Session 12) until the end were classified as breakers (Figure 2.9). The 10 remaining participants did not show such a consistent pattern across sessions. In order to ensure that our categorisation into breakers and non-breakers was relevant, we conducted a series of one-tailed paired-sample t-tests on solution times in Session 12 and showed that, for breakers, +5 problems were solved faster than +6 problems, ($t(6) = 5.96$, $p < .001$) and that, for non-breakers, +5 problems were solved slower than +6 problems ($t(3) = -1.92$, $p = .08$).

Figure 2.8

Difference in Solution Times as a Function of Session in Experiment 2



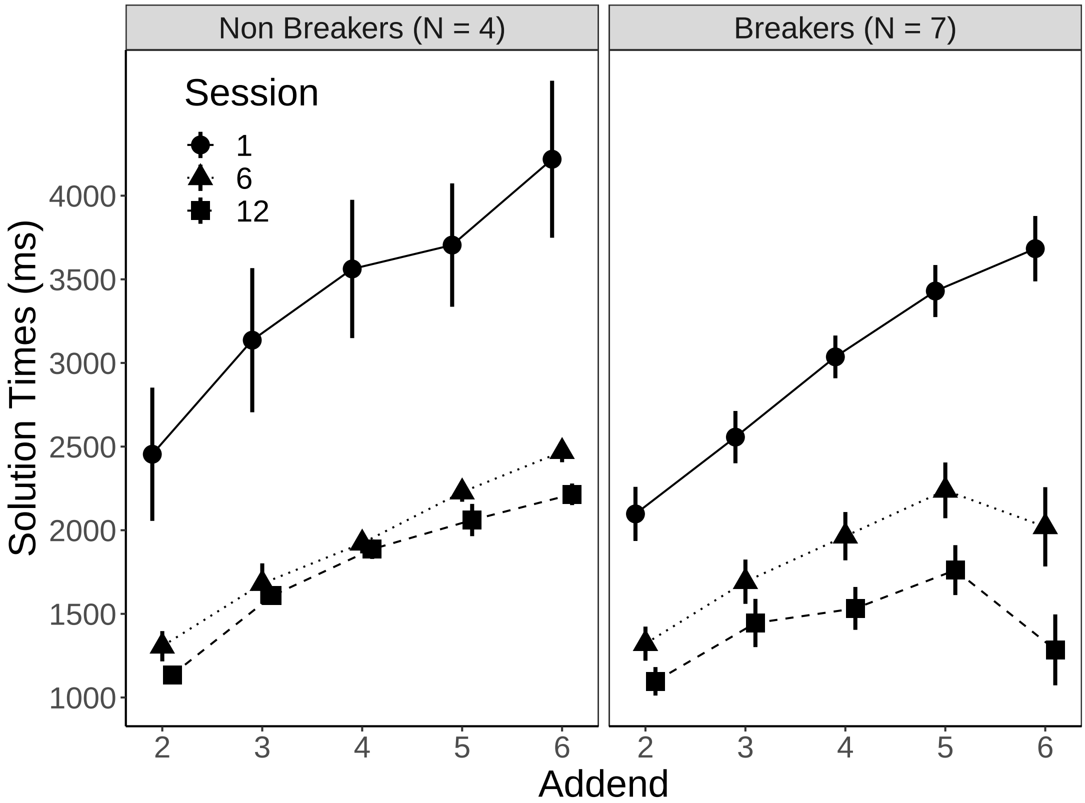
Note. Difference in solution times between problems with addends 5 and 6 across the 12 sessions for true equations in Experiment 2. Error bars represent standard errors.

Discussion

This second experiment was carried out in order to better understand the fact that, at the end of an alphabet-arithmetic task training involving addends from 2 to 5, solution times associated with +5 problems are shorter than for +4 problems. The results of our first experiment confirmed Logan and Klapp (1991)'s interpretation that this decrease in solution times is due, at least in part, to deliberate mnemonic encoding of +5 problems. Logan and Klapp's point of view was that retrieval for +2 to +4 problems would be possible after repeated practice consisting in one-unit steps counting. In contrast, for the same authors,

Figure 2.9

Mean Solution Times as a Function of Addend for Sessions 1, 6, and 12 for Breakers and Non-Breakers in Experiment 2



Note. Mean solution times as a function of addends for Sessions 1 (circles, solid line), 6 (triangles, dotted line), and 12 (squares, dashed line) in Experiment 2, for true trials for non-breakers (left panel) and breakers (right panel). Error bars represent standard errors.

some participants could consider that counting one by one for +5 problems is too demanding and too long and would decide to directly commit their answers to long-term memory shortly after the beginning of the experiment. In order to examine this interpretation, we asked the participants of this second experiment to solve problems with addends 2 to 6 and showed that the decrease in solution times was shifted from +5 to +6 problems. Therefore, if we stick to the interpretation that a break in solution times is the sign of deliberate memorisation, we can conclude that, as in the first experiment, some of the participants in this second experiment have deliberately used mnemonic strategy to learn the alphabet-arithmetic facts associated with the largest addend in the study set.

2.4 General Discussion

Automaticity in the instance theory of automatisation consists in a shift from algorithmic procedures to retrieval of instances from long-term memory (Logan, 1988). Learning would consist in a transition from time-consuming processing of each of the algorithmic steps to rapid single-step access to the instances. In an alphabet-arithmetic task, this would appear in a shift from a steep linear function relating solution times to the addends at the beginning of learning to a flat function at the end of learning. In this paper, we have questioned this conclusion because we argue that, in previous studies, the computation of the addend slope at the end of learning was biased by a drop in solution times for the largest addend.

We have shown here that the systematic drop of solution times for the largest addend was reduced but still present when end-term problems were excluded from the analyses. Then, following Logan and Klapp (1991), we concluded that the residual drop in solution times could be due to deliberate memorisation of problems associated with the largest addend. Therefore, the problems that would be memorised first by individuals would be those requiring the highest number of counting steps. Our first experiment conducted over 25 sessions instead of 12 showed that amongst the 6 out of 19 participants who memorised the problems with the largest addend, half of them also started to memorise problems associated with the second-largest operand. In the second experiment, conducted over 12 sessions using addends from 2 to 6 instead of 2 to 5, we showed that the decrease in solution times after training was, again, observed for the largest addend. This pattern of results was caused by 7 participants out of 21.

Because problems associated with the largest addend were obviously processed differently from other problems, we recalculated the addend slopes without the largest addend. Both in Experiments 1 and 2, these slopes were not at all negligible at the end of training (217 and 235 ms/addend, respectively) and their sizes were higher than the subvocal or overt alphabet recitation time per letter, i.e., 125 ms/letter in Landauer (1962) and 115 ms/letter after training in Logan and Klapp (1991). Therefore, contrary to Logan and Klapp's previous conclusions, it is impossible from these results to exclude the possibility that individuals are still counting for most of the problems after extensive practice. This leaves room to the possibility that training in an alphabet-arithmetic task simply increases the speed of counting, except for a very specific category of problems that are processed differently by a minority of participants.

The fact that repeated practice of an algorithmic procedure finally results in an increased counting speed echoes the conclusions formulated in the more-ecological context of mental arithmetic. For example, Svenson (1985) argued that simple arithmetic problems with a sum inferior to 10 and involving either 1 or 2 as addends could be solved by a counting procedure, whereas larger problems could be solved by retrieval. Interestingly, this conclusion that very-small problems are solved using counting procedures whereas the answers to larger problems can be retrieved has also been reached by Uittenhove et al. (2016). These authors revealed that adults present a linear and monotonic increase in solution times for very-small non-tie additions involving operands from 1 to 4, whereas there was no increase in solution times for problems with a sum from 7 to 10 when one of the operands exceeded 4. The conclusion of the authors was that very-small problems could be solved by very-quick and unconscious automated one-unit step procedures, as long as the operands were not too high. When operands are too large or perceived as such by individuals, the counting procedure could appear to be too demanding, long and possibly prone to errors, and retrieval would hence be favoured.

These conclusions perfectly fit the conclusions formulated in the present paper. In an alphabet-arithmetic task, practice would lead to an acceleration of procedures for problems requiring a minimum number of steps, whereas the answers to problems with numerous steps could be strategically committed to memory. This interpretation echoes the conclusion of Compton and Logan (1991) that the effects observed in alphabet-arithmetic tasks fit well with a race model (Logan & Cowan, 1984), in which memory traces race against the algorithms. When the algorithm is slow, as for large problems, the memory traces are more likely to win than when the algorithm is fast, as for small problems.

Svenson (1985) and Uittenhove et al. (2016) found that each automated step takes approximately 45 ms, which is far from the average slope of 250 ms/addend that we found after 12 sessions in the present experiments. Therefore, it is very unlikely that the counting procedures used by our participants are already automated. Our hypothesis is that automatisation would occur after a much longer training period than what could reasonably be done in a laboratory. Indeed, for mental addition, a slope of 45 ms/unit is observable after many years of practice. As a matter of fact, even at the age of 10 years, i.e., after about 4 years of practice, children do not seem to have already automatised counting procedures, i.e., 390 ms/increment when the minimum addend is considered Thevenot et al. (2016).

Of course, another possibility in the alphabet-arithmetic task would be that after many years of practice, the answers to all problems are retrieved from long-term memory. Nevertheless, in this case, the evolution pattern of retrieval that we observed is in conflict with the current models of retrieval in arithmetic. Indeed, within these models, the answers to small problems are assumed to be retrieved before the answers to large problems during the course of development (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996). Several arguments support this assumption. First, associations between operands and results are easier when the number of steps to reach the correct answer is small than when it is large (Thevenot et al., 2001). Second, the associations between operands and answers are stronger when problems are less error prone and therefore, stronger for small than larger problems (Siegler & Shrager, 1984). Third, the probability of associations between operands and answers is higher for problems that are encountered more often and earlier during development, that is higher for small than large problems (Ashcraft & Christy, 1995; Hamann & Ashcraft, 1986). Therefore, because we show here that the answers to large problems, and not to small problems, are retrieved first during practice, even if we speculate that counting procedures will eventually disappear, most of the retrieval models need to be revised.

To conclude, the results of the present paper challenge the conclusion that the pattern of solution times after extensive alphabet-arithmetic training rules out the possibility that fast counting cannot be the dominant strategy. In other words, we have dethroned one of the main arguments that procedures are taken over by memory retrieval after repeated practice, by showing that the slopes relating solution times to addends are still not negligible after 12 or even 25 sessions of training. Of course, this does not question the whole instance theory put forward by Logan (1988), according to which extensive practice eventually allows the retrieval of associations between operands and answers. First, we have seen that, in accordance with the original conclusions drawn by Logan and Klapp, the answers to some problems associated with the largest operands can be deliberately committed to memory by a minority of participants.

Second, in addition to the pattern of solution times after alphabet-arithmetic training, several other arguments were put forward by Logan and his colleagues. Zbrodoff (1999) revealed the existence of what she called the opportunistic-stopping phenomenon. This phenomenon supposes that when an incorrect equation is presented and that the proposed answer corresponds to a letter preceding the correct answer, individuals would not count up to the correct answer but only up to the proposed one. Consequently, verification times of such equations should be shorter than when the proposed answer follows the right answer. Accordingly, the author observed that at the beginning of an alphabet-arithmetic task, True−2 problems, e.g., A + 4 = C, were rejected faster than True−1 problems, e.g., A + 4 = D, whereas at the end of training, this opportunity stopping phenomenon was no longer observed. Zbrodoff naturally concluded that retrieval instead of counting became the dominant strategy. Nevertheless, this phenomenon will need to be investigated further because in her experiment, in sharp contrast with what could be predicted from the opportunity stopping phenomenon, True−1 problems were not rejected faster than True problems. In fact, only True−2 problems were processed quicker than True problems. Obviously, these inconsistent results relative to the opportunistic-stopping phenomenon will need to be studied further. This is precisely what we aim at doing in a future research, wherein opportunistic stopping effects will be examined as a function of the addends involved in the problems. Specifically, these effects need to be scrutinised in light of the results of the present paper by taking into account the influence of the largest addend.

Next, in order to strengthen our position that fast counting procedures are still used by adults after extensive practice, our future research will need to address the question of transfer of practice. Logan and Klapp (1991) showed poor transfer of knowledge to an untrained part of the alphabet and therefore concluded that learning is item specific. Stated differently, according to the authors, a general procedure should have been transferred to new problems and therefore, the lack of transfer constituted evidence that counting was no longer used at the end of training. However, we suggest that possible transfer of practice should be examined in the future without considering the problems that are responsible for the break in solution times because they probably do not rely on counting procedures any longer.

Finally, in Compton and Logan (1991), participants verbally reported that they increasingly relied on retrieval over the course of training. Moreover, participants reported more retrieval strategies for problems with larger addends than smaller addends. However, the evolution of strategies during the course of learning was not considered for each of the addend. This is unfortunate because, even if verbal reports are not always trustworthy (e.g., Kirk & Ashcraft, 2001; Thevenot et al., 2010), deliberate mnemonic strategies should be associated with an abrupt shift from reported counting to reported retrieval. Therefore, asking participants how they think they solve the problem in each session for each addend could be valuable in future experiments.

3

Automatisation through Practice: The Opportunistic-Stopping Phenomenon Called into Question

Adapted from:
Dewi, J. D. M., Bagnoud, J., & Thevenot, C. (2021). Automatisation through Practice: The Opportunistic-Stopping Phenomenon Called into Question. *Cognitive Science*, *45*(12), e13074. https://doi.org/10.1111/cogs.13074

Abstract

As a theory of skill acquisition, the instance theory of automatization posits that, after a period of training, algorithm-based performance is replaced by retrieval-based performance. This theory has been tested using alphabet-arithmetic verification tasks (e.g., is A + 4 = E?), in which the equations are necessarily solved by counting at the beginning of practice but can be solved by memory retrieval after practice. A way to infer individuals' strategies in this task was supposedly provided by the opportunistic-stopping phenomenon, according to which, if individuals use counting, they can take the opportunity to stop counting when a false equation associated with a letter preceding the true answer has to be verified (e.g., A + 4 = D). In this case, such within-count equations would be rejected faster than false equations associated with letters following the true answers (e.g., A + 4 = F, i.e., outside-of-count equations). Conversely, the absence of opportunistic stopping would be the sign of retrieval. However, through a training experiment involving 19 adults, we show that opportunistic stopping is not a phenomenon that can be observed in the context of an alphabet-arithmetic verification task. Moreover, we provide an explanation of how and why it was wrongly inferred in the past. These results and conclusions have important implications for learning theories because they demonstrate that a shift from counting to retrieval over training cannot be deduced from verification time differences between outside and within-count equations in an alphabet-arithmetic task.

Keywords: Learning; Knowledge acquisition; Training; Strategies; Counting; Retrieval

3.1 Introduction

Many activities, such as reading, writing, driving, recalling autobiographical memories, or solving simple additions can be performed by a majority of adults without considerable cognitive effort. This leads many researchers to the conclusion that individuals perform them by memory retrieval. More generally, it is widely accepted in the literature that the end product of a learning process often consists in retrieval of associations, irrespective of the way the knowledge is acquired. Some knowledge, such as the name of capitals, can be learnt directly and deliberately by the memorisation of associations (e.g., Norway–Oslo or Ecuador–Quito), while other knowledge, such as associations between operands and answers in addition problems (e.g., $4 + 3 = 7$ or $7 + 6 = 13$), can be created after repeated practice of counting procedures such as counting (e.g., $4 + 3 = 5$, 6, 7). In the latter case, a shift from procedural to retrieval strategies necessarily occurs during learning (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996).

In the domain of arithmetic, and especially in mental additions, the proponents of retrieval models infer the shift from procedures to retrieval from the evolution of solution times in the course of development and practice (e.g., Ashcraft & Battaglia, 1978; Logan & Klapp, 1991). In fact, irrespective of the developmental stage, solution times for simple addition problems increase with the size of the smaller operand involved in the problem (e.g., Zbrodoff & Logan, 2005), and this was believed to be due, among other factors, to smaller problems being practiced more often than larger ones. However, the slope of the regression line decreases drastically with age, from 400 ms/increment in 6-year-old (Groen & Parkman, 1972) to 260 ms/increment in third graders and 120 ms/increment in sixth graders (Jerman, 1970), and finally 20 ms/increment in adults (Parkman & Groen, 1971). The slope of 400 ms/increment in 6-year-olds is viewed as reflecting their counting speed because it is unlikely that they already rely on retrieval for non-tie problems. However, the reduction of the size of the slopes in older individuals can be the result of both accelerated counting speed and increased use of memory retrieval. Nevertheless, Groen and Parkman naturally considered the possibility that the slope of 20 ms/increment found in adults corresponds only to counting speed but considered it to be too fast, particularly in comparison to the overt or subvocal recitation speed, i.e., 125 ms/letter (Landauer, 1962). Therefore, they concluded that adults generally retrieve the answer of simple additions from memory but fail to do so in about 5% of the problems. In more recent models, such size effects are interpreted by better memory access and therefore, shorter retrieval times for smaller problems that are learnt earlier (Ashcraft, 1982) and more frequently (e.g., Ashcraft & Christy, 1995) during development, and would hence suffer from less interference than larger ones (e.g., Campbell, 1987a, 1995; Campbell & Graham, 1985).

In all retrieval models of mental addition, the shift from counting to memory retrieval during development has been explained by the strengthening of associations between operands and answers through repeated practice of counting procedures (e.g., Geary, 1996; Siegler & Jenkins, 1989; Siegler & Shipley, 1995; Siegler & Shrager, 1984). This shift from procedural- to retrieval-based performance in the acquisition of simple addition has been modelled by the instance theory of automatization (Logan, 1988). The theory purports that, at the beginning of learning, due to the lack of memory trace associated with the newly-encountered material, the task is accomplished by the use of algorithm-based procedures. Then, with each instance of learning, one memory trace associated with this instance is created. The probability that a task will be performed by memory retrieval increases with the number of traces in memory. Therefore, according to the instance theory of automatization, at one point during the learning process, the probability of using memory retrieval will exceed the probability of using algorithms. According to the author, this point corresponds to the so-called automatisation. From this point onwards, memory retrieval will be used predominantly and involuntarily.

The instance theory of automatization was established to explain the acquisition of several cognitive skills, including the acquisition of addition through the alphabet-arithmetic task (Compton & Logan, 1991; Logan & Klapp, 1991), where a combination of a letter augend and a numerical addend results in a letter answer, e.g., A + 5 = F. Using this task, Logan and Klapp trained adult participants to verify alphabet-arithmetic equations involving addends 2, 3, 4, and 5 over 12 days. On the first training day, the addend slope was 486 ms/addend, indicating the use of a counting strategy. In the last training session, the addend slope decreased to 45 ms/addend. The reduction in addend slope found by Logan and Klapp can be paralleled to the above-mentioned results of Groen and Parkman (1972) who obtained a slope of regression line for the smaller operand of 400 ms/increment in children and 20 ms/increment in adults. The small and non-significant addend slope of 45 ms/addend at the end of training was one of Logan and Klapp's arguments that a shift from counting to memory retrieval had taken place.

However, this argument has recently been questioned (see Chapter 2). In fact, irrespective of the addends used in the study, a systematic phenomenon has been observed in alphabet-arithmetic studies, namely that at the end of training, there is a discontinuity in the increase of solution times as a function of addend (e.g., Chen et al., 2020; Compton & Logan, 1991; Logan & Klapp, 1991; Wenger, 1999; Zbrodoff, 1995, 1999, see also Chapters 2 and 4). More precisely, in Logan and Klapp's work involving addends from 2 to 5, solution times increase from addends 2 to 4 then decrease for addend 5. This systematic finding has led us to argue that the decrease of addend slope with sessions is mainly due to the problems with the largest addend, i.e., solution times for problems involving 5 were lower than for problems involving 4 (see Chapter 2). We demonstrated this by excluding problems with the largest addend from the analyses and obtaining a significant addend slope at the end of training. However, this effect involving problems with the largest addend was only observed in a minority of participants (6 out of 19 in Experiment 1 and 7 out of 21 in Experiment 2 of Chapter 2). Therefore, we concluded that the non-significant addend slope at the end of training in Logan and colleagues' experiments must have resulted from the averaging of solution times across participants, which artificially reduced the addend slope. Furthermore, we logically concluded that memorisation of associations between operands and answers in an alphabet-arithmetic task occurs only for the largest problems and for a minority of participants, even after extensive practice. Obviously, these conclusions stand in opposition to Logan (1988)'s instance theory of automatization. However, Logan's theory is not based solely on the size of the addend slope at the end of learning.

The opportunistic-stopping phenomenon

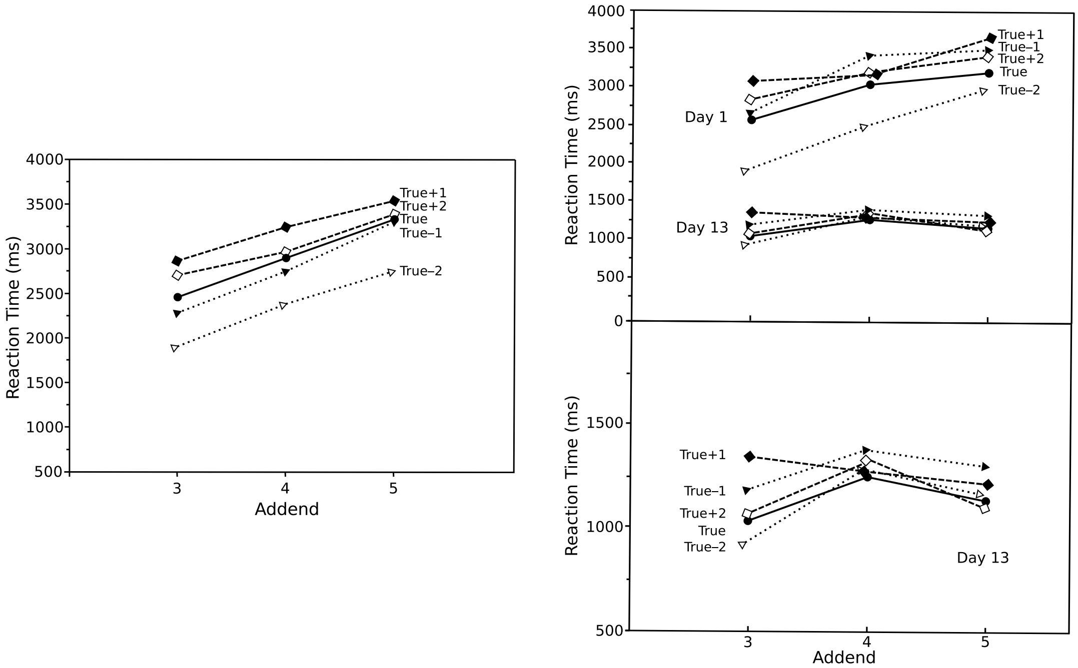
Another signature of a shift from counting to retrieval in alphabet-arithmetic verification tasks was proposed by Zbrodoff (1999). She analysed different false equations of the same problem and argued that if counting were used to verify a false equation associated with an answer preceding the true answer (i.e., within-count answer, e.g., A + 5 = E), then participants would stop counting once the proposed answer has been reached. In other words, in verifying the false equation A + 5 = E, the participant would stop counting at E and would not continue all the way to the correct answer F. In contrast, if a false equation associated with an answer following the true answer (i.e., outside-of-count answer, e.g., A + 5 = G) is presented, then participants would count up to either the correct answer or the proposed answer. As a consequence, within-count equations would be rejected faster than outside-of-count equations. Zbrodoff called this phenomenon opportunistic stopping and the present paper aims at investigating it further.

In Experiment 1 conducted over one session, Zbrodoff (1999) asked her participants to verify 54 problems, consisting of 18 letters paired with addends 3, 4, and 5. Each problem appeared 4 times with their true answer (T) and 4 times with false answers. When a false answer was presented, it could correspond to two letters following the true answer (T+2), the letter following it (T+1), the letter preceding it (T–1), or two letters preceding it (T–2). The author concluded that opportunistic stopping was observed, because, on average, within-count equations were rejected 560 ms faster than outside-of-count equations. Furthermore, T–2 equations were rejected 434 ms faster than T–1 equations. The author's results are reproduced here on the left panel of Figure 3.1.

In her Experiment 4, Zbrodoff (1999) used the same paradigm as in her Experiment 1 but with a fewer number of problems, more repetitions of the same problem, and over 13 training sessions instead of a single one. More precisely, participants had to verify 27 problems consisting of 9 letters paired with addends 3, 4, and 5. In each session, each problem appeared 8 times with their true answer and 8 times with false answers. The author observed that within-count equations were rejected 401 ms faster than outside-of-count equations on the first day. Because opportunistic stopping was obtained, the author concluded that counting strategy must have been used. Furthermore, although Zbrodoff did not report the values, it is clear from the right panel of Figure 3.1 (that replicates Zbrodoff's

Figure 3.1

Solution Times as a Function of Addend



Note. Solution times as a function of addends in Experiment 1 (left panel) and Experiment 4 (right panel) of Zbrodoff (1999). Results for Day 13 of Experiment 4 are enlarged on the bottom panel. Solid circles and solid lines represent solution times for T equations, open lozenges and dashed lines for T+2 equations, solid lozenges and dashed lines for T+1 equations, solid triangles and dotted lines for T–1 equations, and open triangles and dotted lines for T–2 equations. Adapted from "Effects of Counting in Alphabet Arithmetic: Opportunistic Stopping and Priming of Intermediate Steps", by N. J. Zbrodoff, 1999, *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*, p. 303 (left panel) and p. 311 (right panel). Copyright 1999 by the American Psychological Association.

Figure 3) that T–2 equations were rejected faster than T–1 equations, replicating the results of her first experiment. However, opportunistic stopping disappeared on Day 13 (see the right panel of Figure 3.1). Stated differently, rejection times for within-count and outside-of-count equations were similar. This disappearance was interpreted as the evidence that a shift from counting to memory retrieval had taken place.

Thus, using opportunistic stopping, Zbrodoff (1999) adduced additional evidence for the shift from counting to retrieval in tasks initially requiring procedural algorithms. Nevertheless, detailed inspections of her results, particularly of her Experiment 4, cast doubts on her conclusions. As we will explain in the following subsection, her results were not consistent with her definition of opportunistic stopping. Furthermore, an alternative explanation can account for her results.

Why opportunistic stopping does not correspond to the opportunity to stop

If opportunistic stopping reflects the use of counting strategy, then not only should solution times for T–2 equations be shorter than for any other equations but also solution times for T–1 equations should be shorter than for T and outside-of-count equations (i.e., T+1 and T+2). Following this rationale, T–2 equations were indeed solved faster than other equations in Experiment 1 and on Day 1 of Experiment 4. However, T–1 equations were not solved faster than T or outside-of-count equations. In fact, on Day 1 of her Experiment 4, when counting was supposed to be the dominant strategy, solution times for T–1 were longer than for T equations, irrespective of the problem addend. Moreover, rejection times for T–1 were longer than for T+1 equations for +4 problems and were longer than for T+2 equations for +4 and +5 problems (see the right panel of our Figure 3.1). Because these observations concerning T–1 equations are not consistent with the idea that participants stop counting once the proposed answer is reached, the validity of the opportunistic-stopping phenomenon as the signature of a counting strategy is questionable. This is highly problematic because if the existence of opportunistic stopping at the beginning of practice cannot be taken as the sign of counting, then its disappearance at the end of practice cannot be taken as the sign of retrieval.

Interestingly, Zbrodoff (1999) found that the disappearance of opportunistic stopping occurred at the same moment as when the addend slope for T equations reached the asymptotic, non-significant value of 60 ms/addend (i.e., Session 5). This correspondence supports the view that the disappearance of opportunistic stopping is indeed the sign of a shift from counting to retrieval. However, as discussed earlier, the size of this non-significant slope was artificially lowered by problems with the largest addend, i.e., +5, which were solved 110 ms faster than +4 problems. As it can be seen on the right panel of our Figure 3.1, lower solution times for +5 than for +4 problems was obtained for the 5 types of equations (i.e., T, T–1, T–2, T+1, T+2). Thus, it is possible that, similar to the non-significant addend slopes at the end of training, the disappearance of opportunistic stopping was also caused by problems with the largest addend. This interpretation is supported in Zbrodoff's study by the fact that T+2 equations for +5 problems were solved faster than any other equations (see the bottom part of the right panel of Figure 3.1). These problems correspond to what we called the end-term problems, i.e., problems that have unique combination of letter augend and letter answer in the study set, that are partially responsible for the decrease in solution times (see Chapter 2). For example, for the letter augend A, the T+2 equation A + 5 = H can be recognised quickly because it is the only equation pairing A and H. The salience of problems with a unique problem-answer combination may lead individuals to memorise and process them faster than non-unique problems. Note that the end-term problems also involve T–2 equations for +3 problems. For example, the equation A + 3 = B is the only one pairing A and B. As will be explained later, these types of equations can also be solved quickly by the so-called letter-after strategy.

How to explain (Zbrodoff, 1999)'s results

After demonstrating that the so-called opportunistic stopping does not necessarily reflect the use of counting strategy in Zbrodoff (1999)'s experiment, we propose alternative explanations for her results concerning, first, shorter rejection times for T–2 than for other problems in Day 1 and, second, the disappearance of opportunistic stopping in Day 13.

Zbrodoff (1999) found that rejection times for T–2 equations in Day 1 were shorter than for other equations. As already explained, her interpretation was that participants counted until they reached the proposed answer and then stopped. However, the use of other strategies than counting can also explain this effect. For example, participants might have used plausibility judgements, which do not require counting and can lead to short solution times (Lemaire & Fayol, 1995; Lemaire & Reder, 1999; Masse & Lemaire, 2001; Reder, 1982; Zbrodoff & Logan, 1990). It turns out that T–2 equations can easily give rise to such judgements. First, the proposed answer for T–2 problems is relatively close to the letter augend (e.g., C + 4 = E or A + 5 = D) and, without counting, it is easy to figure out that E cannot be 4 letters apart from C, or D cannot be 5 letters apart from A. Second, to solve T–2 problems, participants can use a letter-after strategy, which would be similar to the "number after N" strategy described in mental arithmetic (Baroody, 1995; Baroody et al., 2012). More precisely, to solve $N + 1$ or $1 + N$ problems, instead of counting 1 step, individuals can simply retrieve the next number after N in the counting sequence (Bagnoud, Dewi, Castel et al., 2021; Bagnoud, Dewi & Thevenot, 2021; Grabner et al., 2021). This highly-salient relation between numbers or letters of the alphabet can allow individuals to immediately realise that, for example, G + 3 = H is false because H immediately follows G in the alphabet and cannot therefore be 3 letters apart (see Table 3.A.1). Nevertheless, such plausibility judgements can also be applied to T–1 equations (e.g., G + 2 = H), and we have seen that they are not processed faster than other equations. Another rule that individuals could use to avoid counting would be to use the "skip the number after N" strategy instead of counting 2 steps (Baroody, 2018). For example, the T–2 equation A + 4 = C can be judged quickly as incorrect because C is reachable by skipping the letter after A and hence cannot be 4 letters apart from A (see Table 3.A.1). More generally, such strategies, which do not imply one-by-one counting, can be applied when the proposed false answer is close to the letter used in the problems, and this could explain why within count answers were rejected faster than outside of count equations.

Therefore, the so-called opportunistic-stopping phenomenon supposed by Zbrodoff (1999) in Day 1 does not necessarily reflect the use of counting. Hence, its disappearance in Day 13 does not necessarily imply a shift from counting to retrieval. An explanation of this disappearance can be found in a reversal, or change of sign from positive to negative, of the differences in rejection times between within-count and outside-of-count equations for +4 and +5 problems in Day 13 (see the bottom part of the right panel of Figure 3.1). More precisely, for +3 problems, both within-count equations were rejected faster than the two outside-of-count equations. For +4 problems, T–1 were rejected slower than T+1 but T–2 were rejected faster than T+2. For +5 problems, however, both within-count equations were rejected slower than the two outside-of-count equations. Although the results for +3 problems and for T–2 of the +4 problems could be explained by the letter-after or skip-letter-after strategies illustrated in Table 3.A.1, the results for +5 problems and for T–1 of +4 problems cannot. Nevertheless, we argue that averaging the differences between within-count and outside-of-count equations over addends resulted in a difference which was close to 0 and not significant, which was interpreted by Zbrodoff as the disappearance of opportunistic stopping. Although her ANOVAs included addend and equation type, Zbrodoff did not report the interaction between the two variables. We think that this interaction is in fact important because it could highlight the reversal of the differences in rejection times between within-count and outside-of-count equations, particularly for problems with the largest addend. Furthermore, we argue that it is this reversal for problems with the largest addend, and not the shift to retrieval, that was responsible for the disappearance of opportunistic stopping at the end of practice. The reason why T+2 were rejected faster than T–2 for +5 problems in Day 13 might be related to the fact that these equations contain the only combination of letter augend and letter answer in the study set. For example, the combination of letters A and H is found only once in the study set, i.e., as the T+2 equations of the letter augend A, and hence, this combination could be recognised as a false equation faster than other combinations for +5 problems. In other words, similar to solution-time discontinuity found in earlier alphabet-arithmetic studies (e.g., Chapters 2 and 4), the reversal of rejection times was also caused by problems with the largest addend.

Thus, to sum up, we argue that the obtained opportunistic stopping in Day 1 of Experiment 4 of Zbrodoff (1999) cannot be strictly associated with the use of counting but could be related to the use of other strategies based on plausibility judgements. Furthermore, we argue that the disappearance of opportunistic stopping in Day 13 is not necessarily associated with the use of retrieval but could be due to the reversal of the difference in rejection times between within-count and outside-of-count equations that is observed for problems with the largest and, to a certain extent, second-largest addends.

The present study

Therefore, in the present paper we will reinvestigate the disappearance of opportunistic stopping by hypothesising that it is due to the reversal of the sign of the difference in rejection times between within-count and outside-of-count equations. Because we do not think that Zbrodoff's results were due to opportunistic stopping, we will use the term Equation Type effect to qualify the general effect of solution times, i.e., the difference between true, within-count, and outside-of-count equations. The term of opportunistic stopping will only be used when we refer to the definition put forward by Zbrodoff (1999), i.e., shorter rejection times for within-count than for outside-of-count equations.

We predict that the disappearance of the Equation Type effect at the end of practice is due to problems with the largest addend, more precisely due to the reversal of the sign of the difference between within-count and outside-of-count rejection times. Moreover, because this reversal and solution-time discontinuity are both related to problems with the largest addend through the so-called end-term effect (Chapter 2), we predict that these two phenomena will start to occur around the same time. However, because the discontinuity in solution times due to problems with the largest addend was only observed in a minority of participants (Chapters 2 and 4), we predict that the Equation Type effect will be observed only for these participants.

In order to test our hypotheses, we used the data collected and reported in Chapter 2. The following Method section is therefore the same as described for Experiment 1 in Chapter 2, which was an alphabet-arithmetic verification training experiment ran over 25 sessions. Ten letters were paired with addends 2 to 5 and in each session, and each problem was presented 6 times with its true answer and 6 times with the false answers. In our experiment, we only used T–1 and T+1 equations. This way, although the use of the letter-after strategy is possible, i.e., for +2 problems, the possibility of using the skip-the-letter-after strategy is reduced compared to the design used by Zbrodoff because, in our design, it is applicable only for +3 problems (see Table 3.A.1). We therefore place ourselves in a situation where reliance on the opportunistic-stopping strategy is enhanced.

Whereas only the results on true equations were analysed and presented in Chapter 2, true and false equations were analysed for the present chapter. The results reported in this chapter have therefore never been reported elsewhere.

3.2 Method

Participants

Nineteen students of the University of Geneva, aged between 18 and 35 years, were recruited. A compensation of CHF 200 was offered for their participation. In order to increase their motivation, participant with the best performance during the training phase was awarded with a bonus of CHF 50.

Written informed consent was obtained for each participant. All procedures performed in this study, involving human participants, have been conducted in compliance with the Swiss Law on Research Involving Human Beings. Because only behavioural data were collected in a non-vulnerable population of adults, the approval of the local ethics committee was not required. The study was carried out in accordance with the recommendations of the Ethics Committee of the University of Geneva, following the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Material and procedure

The experiment was constructed as a training study of an alphabet-arithmetic verification task, wherein equations consisting of a digit addend and a letter augend (e.g., A + 2 = D; false) have to be verified. One half of the presented equations were associated with the correct letter (e.g., A + 3 = D) whereas the other half of the equations were associated with a false answer (e.g., C + 4 = H). Half of the false answers corresponded to the letter before the correct answer (T–1 equations) and the other half corresponded to the letter after it (T+1 equations). Participants were trained on 40 problems consisting of addends 2 to 5 associated with 10 letters. Half of the participants were trained with the first 10 letters of the alphabet (i.e., A to J) and the other half with the next 10 letters (i.e., K to T). Participants were randomly assigned to the first or second part of the alphabet. Exactly as in Zbrodoff (1999)'s experiments, participants were instructed to solve the equations as fast as they could while keeping the accuracy high. Keeping instructions similar to previous experiments was important because insisting only on speed could lead to an increased use of retrieval whereas insisting only on accuracy could result in an increased use of counting (Wilkins & Rawson, 2011).

The task was created using E-prime 2.0 software. Material was set up on participants' laptops for home training. Although it can be argued that home training could involve more distractions than a laboratory training and hence might result in more noise in the data, this is not necessarily the case. In fact, the results of Experiments 1 and 2 reported in Chapter 2 were comparable, even though the training for Experiment 1 was carried out at home and the training for Experiment 2 at the laboratory.

Each trial began with a fixation point (*) presented for 500 ms, followed by the equation. The equations were presented horizontally in Courier New font of size 18 and black colour. They were positioned in the centre of the screen and remained on the screen until a response key was pressed. Participants were asked to press the "A" key when the presented equation was correct and the "L" key when it was not. This information was given on the screen and participants had to remember the appropriate keys throughout the experiment session. Then, the screen remained blank for 1500 ms.

Every combination of letters (A to J or K to T) and addends (2 to 5) was presented 6 times with its true answer and 6 times with a false answer per session. The 6 false answers corresponded to three T–1 and three T+1 answers. Thus, every session involved 480 trials (i.e., 10 letters $\times$ 4 addends $\times$ 2 possible answers $\times$ 6 repetitions), which were divided into four blocks separated by a break. At the end of every session, the percentage of correct responses was displayed and participants had to note it down in a table. Because participants were allowed to have a one-day break during the week, the 25 training sessions took place over 25 to 30 days.

3.3 Results

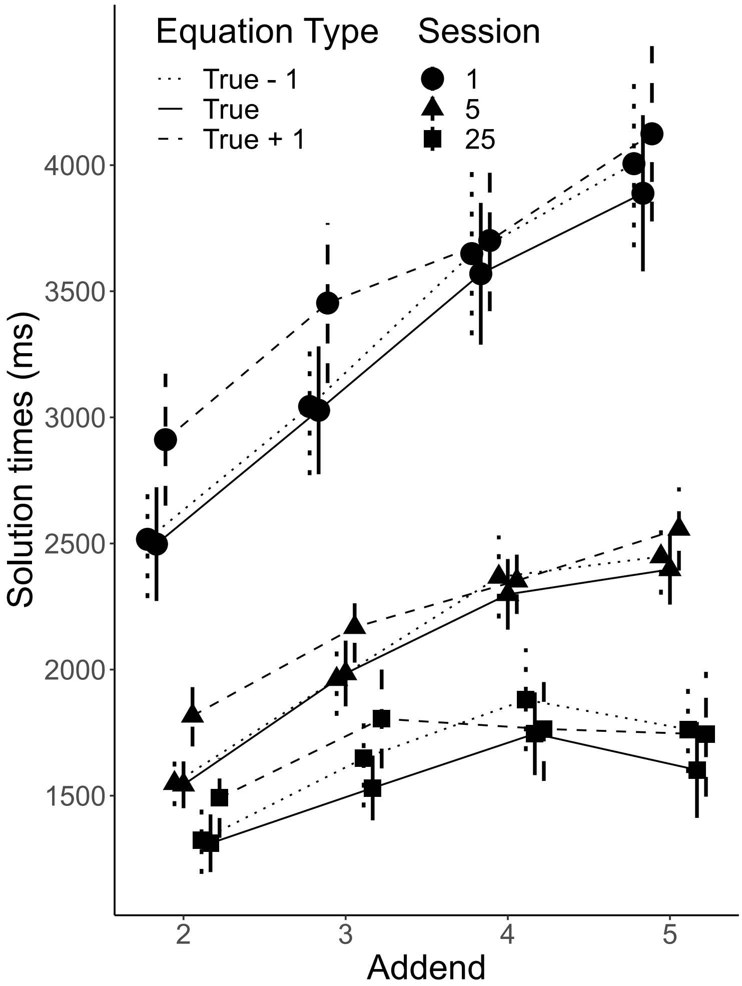
The following results were based on correct trials only, i.e., trials correctly identified as true or false, which constituted 98% of the data. We discarded further 1.6% of the correct trials due to the solution times being either too short (i.e., shorter than 300 ms) or too long (i.e., longer than the mean plus 3 standard deviations for each participant in each session). Figure 3.2 shows the solution times as a function of addend for the 3 types of equations for Sessions 1, 5 and 25. Following Zbrodoff (1999), we will present our analyses for the first and last sessions. Additionally, we also conducted analyses for Session 17. This is because this session corresponds to the session where the exact same number of T equations (i.e., 104) as at the end of Zbrodoff (1999)'s Experiment 4 has been presented to our participants.

Disappearance of opportunistic stopping

A 4 (Addend: 2, 3, 4, and 5) $\times$ 3 (Equation Type: T–1, T, and T+1) repeated measures ANOVA was carried out on solution times of correctly solved problems in Session 1. The ANOVA revealed significant main effects of Addend

Figure 3.2

Solution Times as a Function of Addend and Equation Type



Note. Solution times as a function of addend and equation type (solid lines for T, dotted lines for T–1, and dashed lines for T+1 equations) in Sessions!1 (solid circles), 6 (solid triangles), and 25 (solid squares). Error bars represent standard errors.

($F(3,54) = 98.92$, $\eta_p^2 = .85$, $p < .001$) and Equation Type ($F(2,36) = 14.77$, $\eta_p^2 = .45$, $p < .001$). After a Holm correction, a series of contrasts revealed that T equations (3246 ms) were solved faster than T+1 equations (3548 ms, $t(18) = -4.57$, $p < .001$) and T–1 equations (3304 ms) were solved faster than T+1 equations ($t(18) = -3.94$, $p = .002$). The interaction between Addend and Equation Type was significant ($F(6,108) = 2.89$, $\eta_p^2 = .14$, $p = .01$). The same pattern of results was observed for all addend (i.e., T+1 > T–1 > T) but a series of contrasts with a Holm correction showed that the differences between T and T–1 and between T–1 and T+1 did not reach significance for +4 and +5 problems (see Table 3.1).

A 4 (Addend: 2, 3, 4, and 5) × 3 (Equation Type: T–1, T, and T+1) repeated measures ANOVA was carried out on solution times of correctly solved problems in Session 17. The ANOVA revealed significant main effects of Addend ($F(3,54) = 15.38$, $\eta_p^2 = .46$, $p < .001$) and Equation Type ($F(2,36) = 8.02$, $\eta_p^2 = .31$, $p = .001$). After a Holm correction, a series of contrasts revealed that T equations (1559 ms) were solved faster than T–1 (1676 ms, $t(18) = -2.84$, $p = .02$) and T+1 equations (1719 ms, $t(18) = -3.41$, $p = .01$). The interaction between Addend and Equation Type was significant ($F(6,108) = 4.58$, $\eta_p^2 = .20$, $p < .001$). A series of contrasts with a Holm correction revealed different patterns for different addends (see Table 3.1). For both +4 and +5 problems, T–1 equations were solved descriptively slower than T+1 equation, resulting in positive difference in rejection times between T–1 and T+1 equations.

A 4 (Addend: 2, 3, 4, and 5) × 3 (Equation Type: T–1, T, and T+1) repeated measures ANOVA was carried out on solution times of correctly solved problems in Session 25. The ANOVA revealed significant main effects of Addend ($F(3,54) = 9.79$, $\eta_p^2 = .35$, $p < .001$) and Equation Type ($F(2,36) = 4.57$, $\eta_p^2 = .20$, $p = .02$). However, a series of contrasts with a Holm correction failed to reveal a difference between different equation types. The interaction between

Table 3.1

Difference in solution times between different equation types

Session	Addend	T–1 vs T		T vs T+1		T–1 vs T+1	
		Δt	p	Δt	p	Δt	p
1	2	19	.82	–414	***.001***	–395	***.009***
	3	15	.80	–425	***.002***	–410	***.002***
	4	80	.80	–132	.18	–52	.80
	5	117	.28	–236	.15	–119	.28
17	2	22	.68	–245	***.002***	–223	***<.001***
	3	109	.12	–221	***.01***	–112	.07
	4	147	.07	–21	.78	126	.48
	5	192	***.01***	–153	***.01***	39	.25
25	2	13	.79	–180	***.02***	–167	***.006***
	3	118	.25	–274	***.03***	–157	***.002***
	4	135	.13	–19	.81	116	.41
	5	161	***.007***	–142	.17	19	.82

Note. Difference in solution times (Δt) between T–1 and T, between T and T+1, and between T–1 and T+1 equations and the corresponding p-value. Positive differences indicate that solution times for the first equations were longer than for the second. For the sake of visibility, differences with $p < .05$ are written in bold italics.

Addend and Equation Type was significant ($F(6, 108) = 3.41$, $\eta_p^2 = .16$, $p = .004$) and a series of contrasts with a Holm correction revealed the same patterns as in Session 17 (see Table 3.1). Again, for both +4 and +5 problems, T–1 equations were solved descriptively slower than T+1 equation, resulting in positive difference in rejection times between T–1 and T+1 equations.

We also calculated the addend slopes for each participant and each session. The average addend slopes in Session 1 were 506, 472, and 389 ms/addend for T–1, T, and T+1 equations, respectively (ps < .001). In Session 25, they decreased to 155 ms/addend ($p < .001$) for T–1, 108 ms/addend ($p = .03$) for T, and 72 ms/addend ($p = .13$) for T+1 equations. When +5 problems were excluded, the addend slopes in Session 25 were 278 ms/addend ($p < .001$) for T–1, 217 ms/addend ($p < .001$) for T, and 137 ms/addend ($p = .007$) for T+1 equations.

Opportunistic stopping and end-term effects

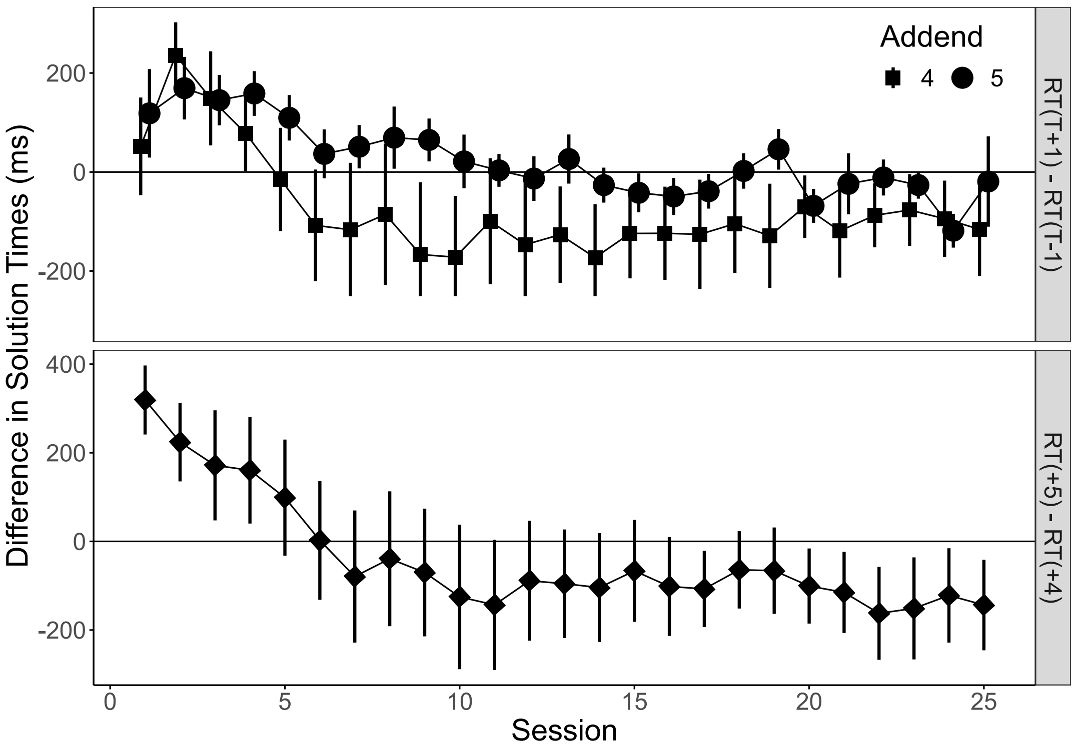
We predicted that the reversal of the difference in rejection times between within-count and outside-of-count equations for problems with the largest addend would coincide with the discontinuity in solution times. However, Figure 3.3 shows that whereas solution-time discontinuity occurred in Session 6 (i.e., negative difference between +5 and +4 problems), the reversal of the difference in rejection times for +5 problems took place in a later session, i.e., Session 11. Instead, we found that the emergence of solution-time discontinuity coincided with the reversal of the difference in rejection times for +4 problems.

Breakers and Non-breakers

As already explained, in Chapter 2 we classified the participants according to whether or not they showed a solution-time discontinuity or, in other words, whether solution times for problems with the largest addend were shorter than for problems with the second-largest addend. Following Chapter 2, 6 participants who showed a systematic solution-time discontinuity from one session (from as early as in Session 1 to as late as in Session 17) until the end of training were classified as breakers whereas 6 participants who did not show solution times discontinuity throughout the experiment were classified as non-breakers. To test whether the effect of Equation Type differs in breakers and non-breakers, a 4 (Addend: 2, 3, 4, and 5) × 3 (Equation Type: T–1, T, and T+1) × 2 (Group:

Figure 3.3

Difference in Solution Times across Sessions



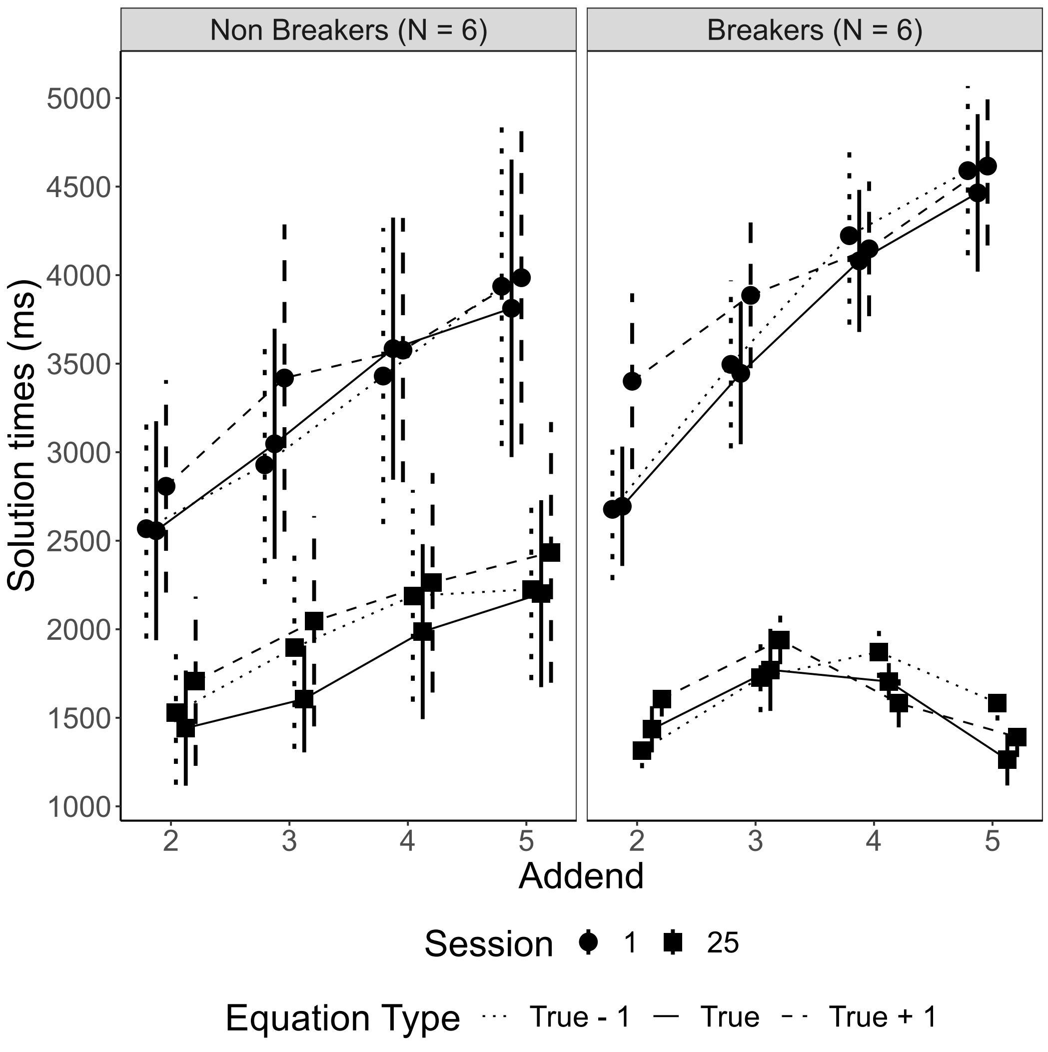
Note. Top panel: Difference in solution times between T+1 and T–1 equations for +4 (squares) and +5 (circles) problems across sessions. Positive differences imply larger solution times for T+1 than for T–1. Bottom panel: Difference in solution times between +5 and +4 problems for T equations across sessions. Positive differences imply larger solution times for +5 than for +4 problems. Error bars represent standard errors.

breaker vs non-breakers) repeated-measures, mixed-design ANOVA was carried out on solution times of correctly solved problems in Sessions 1, 17, and 25. The results for Session 1 and 25 are presented in Figure 3.4 and the results for the whole experiment in Appendix 3.B.

In Session 1, the same general results as for the whole sample were found, i.e., significant effects of Addend ($F(3, 30) = 60.56$, $\eta_p^2 = .86$, $p < .001$) and Equation Type ($F(2, 20) = 9.85$, $\eta_p^2 = .50$, $p = .001$) as well as significant interaction between Addend and Equation Type ($F(6, 60) = 2.82$, $\eta_p^2 = .22$,

Figure 3.4

Solution Times as a Function of Addend and Equation Type for Breakers and Non-breakers



Note. Solution times as a function of addend and equation type (solid lines for T, dotted lines for T–1, and dashed lines for T+1 equations) in Sessions 1 (solid circles) and 25 (solid squares) for non-breakers (left panel) and breakers (right panel). Error bars represent standard errors.

$p = .02$). There was no main effect of Group ($F(1, 10) < 1$) or interaction between Group and Addend ($F(3, 30) = 1.09$, $p = .37$), Group and Equation Type ($F(2, 20) < 1$), or Group × Addend × Equation Type ($F(6, 60) = 1.06$, $p = .40$).

In Session 17, the same general results as for the whole sample were found, i.e., significant effects of Addend ($F(3, 30) = 9.46$, $\eta^2_p = .49$, $p < .001$) and Equation Type ($F(2, 20) = 4.02$, $\eta^2_p = .29$, $p = .03$) as well as significant interaction between Addend and Equation Type ($F(6, 60) = 5.34$, $\eta^2_p = .35$, $p < .001$). The effect of Group was not significant ($F(1, 10) < 1$) but the interaction between Group and Addend was ($F(3, 30) = 7.13$, $\eta^2_p = .42$, $p < .001$). However, a series of contrasts with a Holm correction failed to reveal a group difference in any addends. The interaction between Group and Equation Type was not significant ($F(2, 20) = 1.11$, $p = .35$) but the three-way interaction was ($F(6, 60) = 4.86$, $\eta^2_p = .33$, $p < .001$). A series of contrasts with a Holm correction revealed that there was an interaction between Addend and Equation Type in breakers ($F(6, 60) = 9.92$, $\eta^2_p = .50$, $p < .001$) but not in non-breakers ($F(6, 60) < 1$). The interaction in breakers was due to T–1 rejected faster than T+1 for +2 problems (−303 ms, $t(10) = -4.72$, $p = .002$) and T+1 rejected faster than T–1 for +4 problems (−618 ms, $t(10) = -4.15$, $p = .005$).

In Session 25, we found a significant effect of Addend ($F(3, 30) = 8.40$, $\eta^2_p = .46$, $p < .001$), a marginal effect of Equation Type ($F(2, 20) = 3.19$, $\eta^2_p = .24$, $p = .06$), and a significant interaction between Addend and Equation Type ($F(6, 60) = 2.37$, $\eta^2_p = .19$, $p = .04$). The effect of Group was not significant ($F(1, 10) < 1$) but the interaction between Group and Addend was ($F(3, 30) = 8.66$, $\eta^2_p = .46$, $p < .001$). Again, a series of contrasts with a Holm correction failed to reveal a group difference in any addends. The interaction between Group and Equation Type was not significant ($F(2, 20) = 1.02$, $p = .38$) but the three-way interaction was ($F(6, 60) = 2.39$, $\eta^2_p = .19$, $p = .04$). A series of contrasts with a Holm correction revealed that there was an interaction between Addend and Equation Type in breakers ($F(6, 60) = 4.04$, $\eta_p^2 = .29$, $p = .002$) but not in non-breakers ($F(6, 60) < 1$). The interaction in breakers was due to T–1 being rejected faster than T+1 for +2 problems (−292 ms, $t(10) = -3.78$, $p = .009$) and + 3 problems (−212 ms, $t(10) = -2.77$, $p = .048$), and T being solved faster than T–1 for +5 problems (−320 ms, $t(10) = -4.47$, $p = .003$).

3.4 Discussion

This research aimed at investigating some of the mechanisms at hand in cognitive learning. In the framework of the instance theory of automatization (Logan, 1988), it has been put forward that learning can correspond to a shift of strategy from algorithm-based to memory retrieval. In the alphabet-arithmetic learning, support for this theory has been provided by the opportunistic-stopping phenomenon (Zbrodoff, 1999), according to which within-count answers (e.g., A + 4 = D) are rejected faster than outside-of count answers (e.g., A + 4 = F) at the beginning of practice but not at the end. From these effects, Zbrodoff concluded that a shift from counting to retrieval occurred during the training program. Nevertheless, we noticed that Zbrodoff's results were not always consistent with her definition of opportunistic stopping, namely that even at the beginning of practice, T–1 equations were not solved faster than T equations. Indeed, in our experiment, although we confirmed the results of Zbrodoff by obtaining both an effect of Equation Type and shorter rejection times for T–1 than T+1 equations in Session 1, opportunistic stopping was not observed for T compared to T–1 equations because solution times for these equations were similar. Therefore, although counting was used, participants did not take the opportunity to stop counting on reaching the within-count answer, or, at least, the paradigm used by us and Zbrodoff could not provide any evidence that opportunistic stopping was used.

At the end of practice, confirming the results of Zbrodoff (1999), we did not find a difference in rejection times between within-count and outside-of-count equations. We could have concluded as Zbrodoff that there was no opportunistic stopping at the end of practice but in fact, it was due to a reversal of the sign of the difference in rejection times for +4 and +5 problems. More precisely, in our last session, the difference in solution times between T–1 and T+1 equations was negative for problems with addends 2 and 3 (i.e., −167 and −157 ms, respectively) but positive for problems with addends 4 and 5 (i.e., +116 and +19 ms, respectively). Averaging positive and negative differences resulted in a non-significant difference of −47 ms at the end of training. This reversal of the difference in rejection times was already observed in our Session 17 that corresponded to Zbrodoff's Day 13 in terms of the number of repetitions for each equation. It is crucial to note that the same opposite differences for problems with addend 3 on the one hand and addends 4 and 5 on the other was also obtained by Zbrodoff in her Experiment 4 (see the bottom part of the right panel of our Figure 3.1) and that this is the reason why she erroneously concluded that opportunistic stopping was absent at the end of training. In our present work, we found that the reversal for +4 problems occurred in an earlier session than for +5 problems. This might be the reason why the reversal between T+1 and T–1 in Zbrodoff's Experiment 4 was already present in Day 1.

Thus, in the present paper, although we did not include T+2 and T–2 equations, we obtained the same results as Zbrodoff (1999) in her Experiment 4. First, the effect of equation type at the beginning of practice was not accompanied by shorter solution times for T–1 than for T equations. Second, within-count equations were not rejected faster than outside-of-count at the end of practice. However, we differed from Zbrodoff in the interpretation of the results. Whereas

Zbrodoff concluded that the similarity in rejection times was due to the shift from counting to retrieval, we think, as already explained, that this is due to the reversal of the sign of the difference between the two types of false equations for +4 and +5 problems. We advance further that this reversal was mainly due to the problems with the largest addend. This proposition is supported by the fact that this reversal was only observed in breakers, for whom problems with the largest addend were processed differently than the other problems (Chapter 2), and not in non-breakers (see Appendix 3.B for the performance during the whole training experiment).

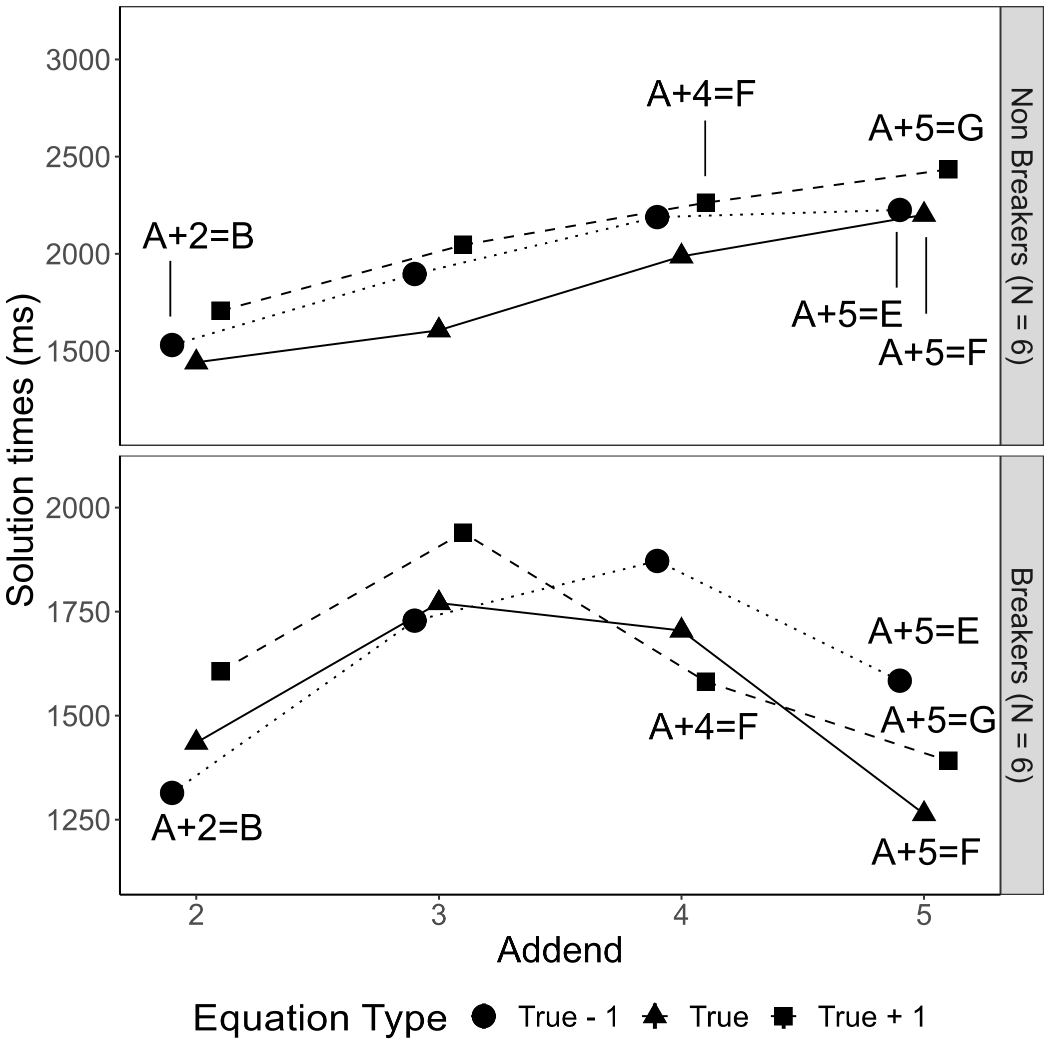
To explain the reversal, we showed in Figure 3.5 the performance of breakers and non-breakers in Session 25. In coherence with Table 3.A.1, we added an example for the letter augend A. First of all, Figure 3.5 shows that for breakers, T–1 equations for +2 and +3 problems were rejected faster than T+1 equations. As explained in Section 3.1 and illustrated in Table 3.A.1, this can be explained by plausibility judgements. However, neither letter-after nor skip-letter-after strategy could explain why, still for breakers, T–1 equations for +4 and +5 problems were rejected slower than T+1 equations. We proposed therefore the following explanation. As stated in Chapter 2, problems with the largest addend, e.g., A + 5, are special because their letter answers appear less frequently than other letter answers in the study set. However, only the breakers are sensitive to these problem particularities and commit them to memory. Indeed, the bottom panel of Figure 3.5 shows that in breakers, solution times for +5 problems were shorter than for +4 problems for the 3 equation types, which replicated the results of Zbrodoff (1999) in Day 13 (see the bottom part of the right panel of our Figure 3.1). However, T+1 equations, e.g., A + 5 = G, have an advantage in terms of rejection times over T–1 equations, e.g., A + 5 = E, because the former contains the only combination between the letter augend and the proposed answer in the whole study set. This was also found by Zbrodoff, because her T+2 equations for +5 problems in Day 13 were rejected faster than other equation types. Interestingly, the T+1 equations of the second-largest addend in our study, e.g., A + 4 = F, have an advantage over T and T–1 equations of the same addend. This is probably because having memorised A + 5 = F as true, participants could quickly reject A + 4 = F as being true.

As evoked above, the reversal of the sign of the difference between the two types of false equations for +4 and +5 problems was not found in the non-breakers (see the top panel of Figure 3.5), for whom solution times increased with addends for the 3 types of equations, i.e., T, T–1, and T+1. Therefore, it seems likely that participants in this group mainly use counting until the end of practice, supporting the interpretation and conclusion of Chapter 2. Thus, the results of the current paper reinforce those of Chapter 2, namely that the possibility that counting was still used after an extensive practice cannot be discarded, and that retrieval is only used for a minority of problems and only for problems with the largest addend. Still, we have to keep in mind that opportunistic stopping might have been used by our participants but that the paradigm that we adopted from Zbrodoff (1999) is not suitable to reveal it. A possible reason for such a failure could be that in solving a within-count equation, when individuals have reached the proposed answer and opportunistically stopped counting, they have to judge it as false. This mismatch might induce a cognitive dissonance, and therefore the time gained by opportunistic stopping for false within-count equations could end up as being longer solution times than for true equations.

Nevertheless, several reasons could explain why participants do not take the opportunity to stop counting in alphabet-arithmetic tasks. An explanation can be found in Sternberg (1966) in his high-speed memory search experiment. Sternberg asked participants to memorise a sequence of 1 to 6 items (digits or letters) that constituted a memory set. He then showed a digit or letter that was either one of the items in the memory set or another item (i.e., a target

Figure 3.5

Illustration of Reversal of the Difference in Rejection Times for Breakers and Non-breakers



Note. Solution times as a function of addend and equation type (circles and dotted line for T–1, triangles and solid line for T, and squares dashed line for T+1 equations) in Session 25 to illustrate the reversal of the sign of the difference in rejection times between within-count and outside-of-count equations for non-breakers (top panel) and breakers (bottom panel).

or distractor). The participants had to decide whether this digit or letter was among the items in the memory set. The search for a distractor is necessarily exhaustive because all items in the memory set have to be scanned before the "no" decision can be taken. In this case, response times should be a function of the size of the memory set. The search for a target, on the other hand, is not necessarily exhaustive because participants could stop scanning the memory set once the matched item is found. In this case, response times would be a function of the position of the item in the memory set. However, Sternberg found that response times for both the distractors and the targets were a function of memory-set size with similar slopes, i.e., 30 and 40 ms/item for the targets and distractors, respectively. Therefore, Sternberg concluded that individuals do not stop searching the memory set, even after they have found a matching digit, but scan the whole list before responding. The results of our experiment show that the same behaviour is adopted by individuals when they have to take a decision involving a counting sequence. Another reason as to why participants do not opportunistically stop counting in alphabet-arithmetic tasks could be that they use a counting-up strategy from the letter augend to the letter answer. This strategy would consist in counting the number of letters separating the two letters given in the equation (e.g., for the equation A + 5 = G, counting from B to G results in 6 letters) and comparing this number of counts to the addend (6 is not 5, hence the equation is not true). This strategy does not provide participants with an opportunity to stop counting.

With some exceptions, in our current chapter and in Zbrodoff (1999)'s Experiment 4, true equations were solved faster than false equations. This might be due to the fact that in both studies, true equations were presented twice as often as false ones. Another possibility is that, following the proposition of Ashcraft and Battaglia (1978) for mental arithmetic, participants first find the searched-for answer and then compare it with the proposed answer. The comparison time depends proportionally on the distance between the correct and proposed answer, such that true equations are solved faster than false equations. It is possible that participants in alphabet-arithmetic verification task adopted the same strategy.

All in all, our conclusions do not support the idea that opportunistic-stopping study could reveal a shift from counting to retrieval. Therefore, our results question the deduction of Zbrodoff (1999) and its support for the instance theory of automatization in particular (Logan, 1988). More generally, our results and those of Chapter 2 question the implication of the instance theory of automatization on retrieval models of mental arithmetic (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996). In fact, the use of counting after an extensive practice could support the automated counting procedure theory (Barrouillet & Thevenot, 2013; Thevenot & Barrouillet, 2020; Uittenhove et al., 2016), according to which small additions with operands inferior to 5 are solved by adults through very fast counting procedures instead of retrieval. It is also possible that practice helps individuals to sharpen their understanding of number relations and to discover numerical patterns on which they can base their solution process through decompositions and derived-fact strategies (e.g., Baroody, 1985). Individuals could therefore memorise only a limited number of meaningful combinations and use them as a basis for developing their reasoning and solution processes, and even for inventing new solution strategies (e.g., Baroody, 1985; Baroody & Rosu, 2006). This kind of strategies would be particularly efficient when the operands in the problems are too large for the implementation of a quick one-by-one counting procedure (Uittenhove et al., 2016).

More generally, even if addend slope in some of our participants show that counting can be the dominant strategy at the end of an intensive arithmetic training, this does not mean that counting is the unique strategy (see Chapter 5). We have described the difference between non-breakers and breakers and explain that the latter seemed to solve problems with the largest addends through memory retrieval. We have also explained how plausibility judgements can be implemented by individuals and how such judgements allow them to avoid counting (see Table 3.A). Moreover, it is possible that some participants used opportunistic stopping to solve some of the problems and that, as already evoked, either the paradigm that we used fail to reveal this strategy or that the sign of such incidental strategies is hidden in mean solution times. As also already stated, it is likewise possible that some participants memorised a limited number of combinations between letter augend, addend and letter answer, which could constitute a basis for decomposing and procedurally processing other problems (e.g., knowing that A + 2 = C could be used to solve A + 3, i.e., (A + 2) + 1, hence D). Even in participants who showed a clear linear addend slope at the end of the experiment, we cannot exclude the possibility of infrequent use of retrieval for some problems. This variety of strategies in arithmetic has been often reported and analysed in the literature (e.g. Bagnoud, Dewi, Castel et al., 2021; LeFevre, Sadesky et al., 1996; Siegler & Shrager, 1984) but we show here that they showed that retrieval might not be the dominant strategies, even after an extensive practice.

In terms of educational implications, our results are consistent with the view that the ultimate goal of primary instruction should be to foster number and operation sense and the meaningful memorisation of basic sums by rote. Specifically, education should build on children's informal counting-based addition, encourage the discovery of patterns and relations, and use these arithmetic regularities to devise reasoning strategies (Henry & Brown, 2008; National Council of Teachers of Mathematics, 2000; National Mathematics Advisory Panel, 2008). The automatisation of reasoning strategies provides an important basis for fluency with basic combinations, including transfer to unpracticed combinations(e.g., Baroody, 2006; Baroody et al., 2009; Baroody et al., 2012; Baroody et al., 2016). Such deep reasoning about numbers presupposes that numbers are accurately represented mentally and that children can easily navigate from one number to another without cognitive cost. Such assumptions provide an explanation as to why children's arithmetic skills are improved after interventions based on one-by-one counting on a number line or on fingers (e.g., Fuchs et al., 2010). In the conventional view, such training leads to better memorisation of arithmetic facts than drill does.

Finally, our results can be used to sharpen computational and mathematical modelling and simulation of learning, which are at the heart of artificial intelligence. Our conclusion that practice of procedures could develop into automatised counting and could foster reasoning-based strategies allows for a conception of more complete learning models mainly based on attention and working memory. Indeed, these cognitive factors play a central role in procedural models of arithmetic because after encoding the problem elements, individuals need to sequentially execute a series of procedural steps while keeping a goal active in working memory (e.g., the aim is the execution of 5 steps). Encoding, refreshing key elements in working memory such as the number of steps already executed, the intermediary results reached during the solving process and the goal itself, speed of working memory decay and forgetting threshold are therefore crucial parameters that need to be integrated in the models (Chouteau et al., 2021). The notion of competition or choice between counting and procedure algorithms on the one hand and direct memory retrieval on the other hand, which is reflected in our results for example by different patterns of solution time distributions in breakers and non-breakers (see also Appendix 3.B) should and will be part of our future research.

Acknowledgements

We would like to express our gratitude to one anonymous reviewer for providing us with a table that illustrates the different kinds of plausibility judgements and also for pointing out the possibility of using the counting-up strategy in solving alphabet-arithmetic equations. We would also like to thank another anonymous reviewer for her or his remark that opportunistic stopping could be accompanied by a cognitive dissonance that leads to within-count equations being solved slower than true equations.

Appendix 3.A

Plausibility Judgements

Table 3.A.1

Items Affected by Rule-Based Plausibility Judgements for Two Studies

	Type	Example: Answers to Letter Augend A Possible Addends: 3 to 5			
			A+3	A+4	A+5
Zbrodoff (1999)	T − 2		B = Letter After A	C = Skip Letter After A	D
	T − 1		C = Skip Letter After A	D	E
	T		D	E	F
	T + 1		E	F	G
	T + 2		F	G	H

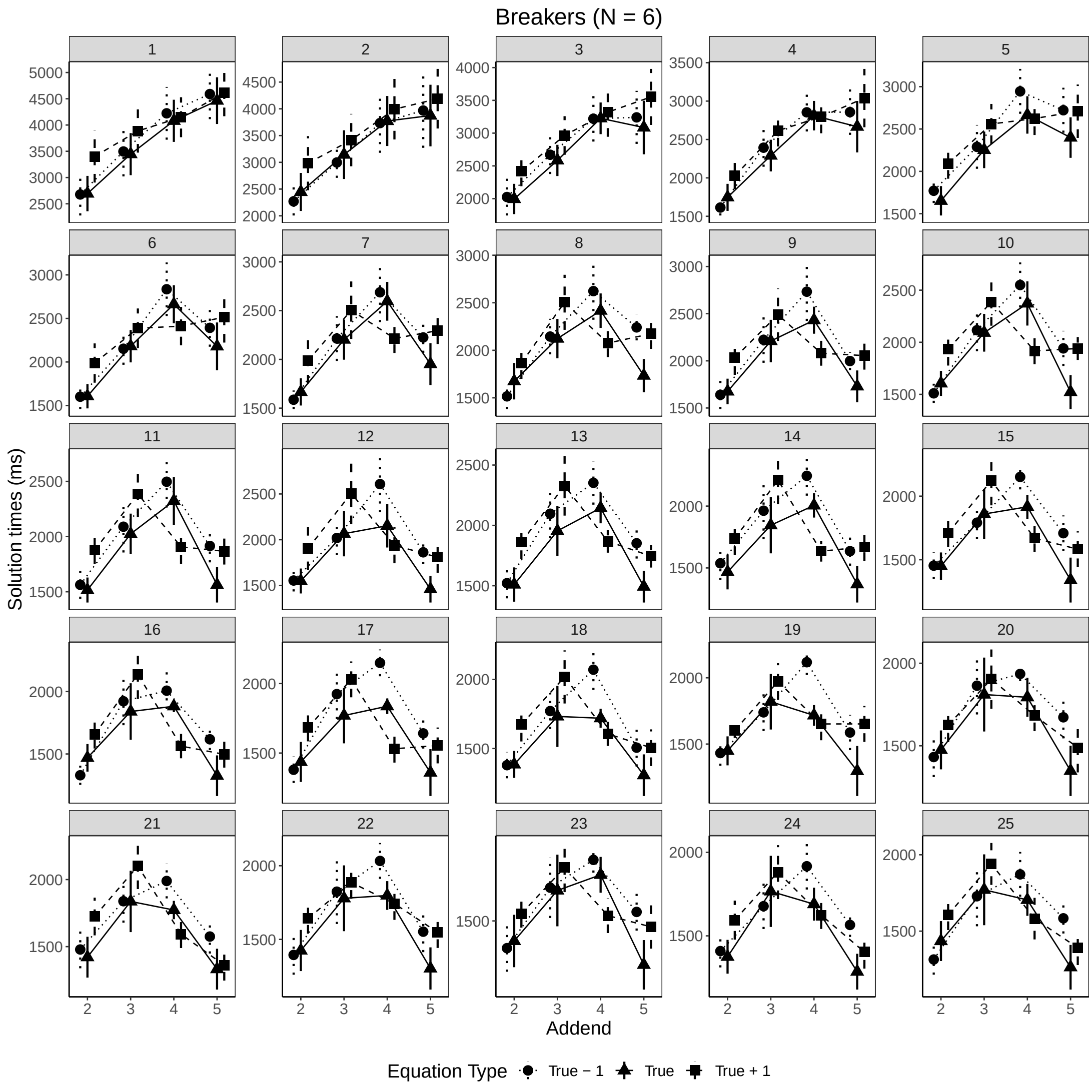
	Type	Example: Answers to Letter Augend A Possible Addends: 2 to 5			
		A+2	A+3	A+4	A+5
This work	T − 1	B = Letter After A	C = Skip Letter After A	D	E
	T	C	D	E	F
	T + 1	D	E	F	G

Appendix 3.B

Performance of Breakers and Non-breakers

Figure 3.B.1

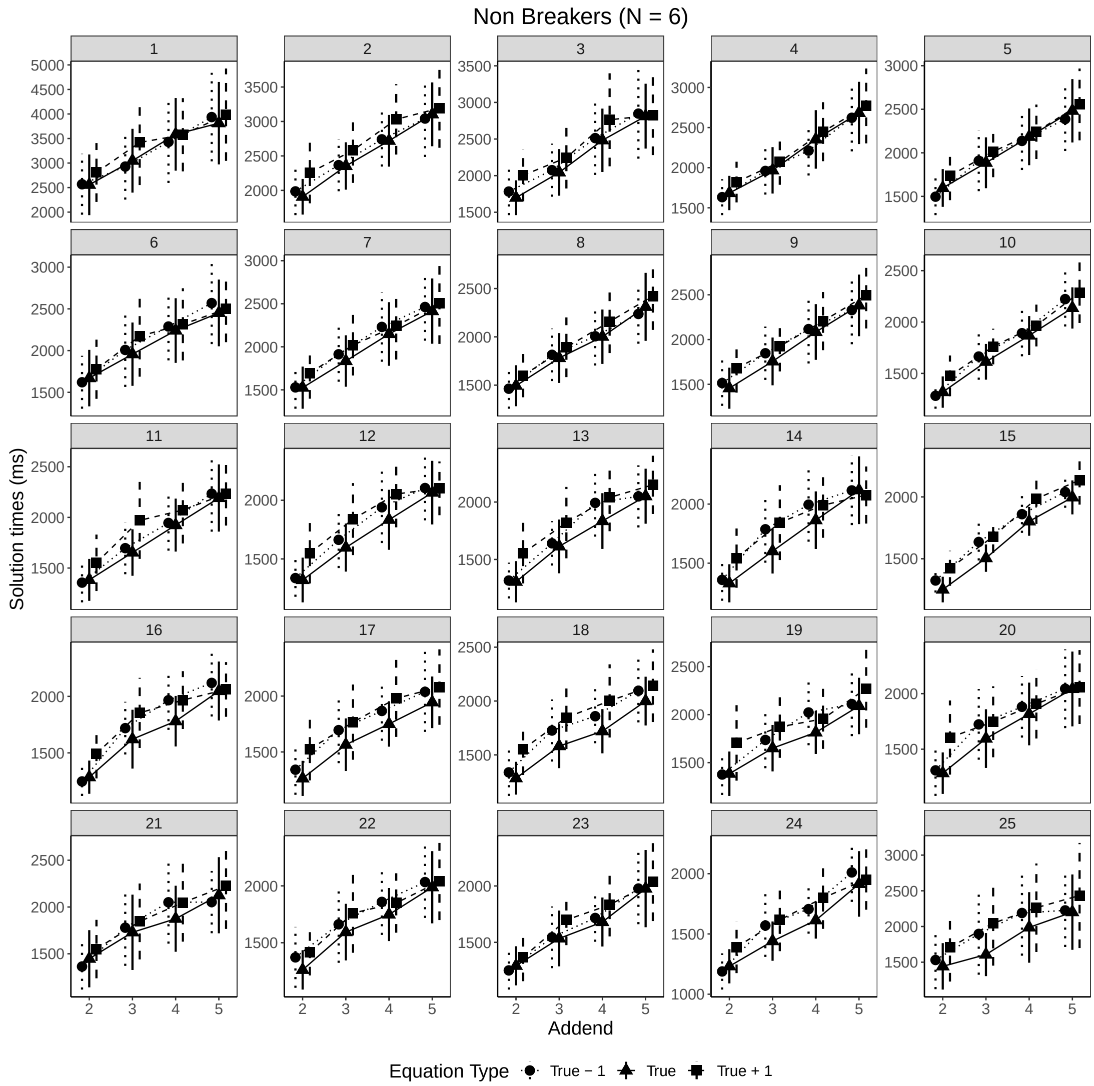
Performance of Breakers during 25 Sessions



Note. Solution times as a function of addend and equation type (circles and dotted lines for T–1, triangles and solid lines for T, and squares dashed lines for T+1 equations) for the breakers in 25 sessions. Error bars represent standard errors.

Figure 3.B.2

Performance of Non-breakers during 25 Sessions



Note. Solution times as a function of addend and equation type (circles and dotted lines for T–1, triangles and solid lines for T, and squares dashed lines for T+1 equations) for the non-breakers in 25 sessions. Error bars represent standard errors.

4

Do Production and Verification Tasks in Arithmetic Rely on the Same Cognitive Mechanisms? A Test Using Alphabet Arithmetic

Adapted from:
Dewi, J. D. M., Bagnoud, J., & Thevenot, C. (2021). Do production and verification tasks in arithmetic rely on the same cognitive mechanisms? A test using alphabet arithmetic. *Quarterly Journal of Experimental Psychology*, *74*(12), 2182–2192.
https://doi.org/10.1177/17470218211022635

Abstract

In this study, 17 adult participants were trained to solve alphabet-arithmetic problems using a production task (e.g., C + 3 = ?). The evolution of their performance across 12 practice sessions was compared to the results obtained in past studies using verification tasks (e.g., is C + 3 = F correct?). We show that, irrespective of the experimental paradigm used, there is no evidence for a shift from counting to retrieval during training. However and again regardless of the paradigm, problems with the largest addend constitute an exception to the general pattern of results obtained. Contrary to other problems, their answers seem to be deliberately memorised by participants relatively early during training. All in all, we conclude that verification and production tasks lead to similar patterns of results, which can therefore both confidently be used to discuss current theories of learning. Still, deliberate memorisation of problems with the largest addend appears earlier and more often in a production than a verification task. This last result is discussed in light of retrieval models.

Keywords: Learning; Procedural knowledge; Memory; Training; Verification tasks; Arithmetic

4.1 Introduction

Researchers agree that children start solving additions by counting procedures (e.g., Bagnoud, Dewi, Castel et al., 2021; Baroody, 1987; Carpenter & Moser, 1984; Groen & Parkman, 1972). However, the way counting strategies evolve with practice is still a matter of debate (see Baroody, 2018; Chen & Campbell, 2018; Thevenot & Barrouillet, 2020, for reviews). Two theoretical views can be contrasted. According to retrieval models, the counting strategies used during childhood are gradually replaced by memory retrieval during the course of development. In adulthood, retrieval is therefore the dominant strategy for all additions involving two single-digit numbers (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996). In opposition to this widely accepted traditional view, Baroody (e.g., 1983, 1984, 1994) argued that simple arithmetic problems could be solved by automated procedures in the form of rules and heuristics. This idea that procedures are still used by experts for very simple addition problems has been taken up recently within the automated counting procedure theory (e.g., Barrouillet & Thevenot, 2013; Fayol & Thevenot, 2012; Mathieu et al., 2016; Uittenhove et al., 2016), according to which the development of strategy in arithmetic could consist in an acceleration of counting procedures until automatisation (Thevenot et al., 2016).

Support for the shift from counting to retrieval in the course of learning has been provided by the instance theory of automatization (Logan, 1988). This theory was developed to account for the acquisition of cognitive skills that can be first learnt by algorithm-based procedures. With each instance of learning, a single memory trace associating the stimuli and the response would be created and then stored in long-term memory. Whereas the probability of using algorithm-based procedures is constant in the course of learning, the probability that memory retrieval is used depends on the number of traces in long-term memory. Therefore, with repeated practice, more and more traces will be created and, at some point, the probability of using memory will be higher than of using algorithm. This point corresponds to the shift from procedural-based to memory-based performance, or, in mental addition, from counting to retrieval.

In this framework, the shift from counting to retrieval in mental arithmetic has been studied using the alphabet-arithmetic paradigm, which was conceived to mimic the way children learn additions. In this paradigm, a number addend is added to a letter augend, resulting in a letter answer. For example, A + 5 = F because F is 5 letters away from A. In their seminal work based on a training experiment, Logan and Klapp (1991) asked adults to learn 40 alphabet-arithmetic problems, consisted of 10 letters paired with addends 2, 3, 4, and 5. Half of the participants learnt the first 10 letters of the alphabet (i.e., A to J) and the other half the second 10 letters (i.e., K to T). After the training phase, which lasted 12 days, participants had to work with the other set of letters on the 13th day. Logan and Klapp concluded that the shift of strategy has occurred, because the addend slope was significant in Session 1, implying the use of counting, but was not significant in Session 12, suggesting the use of memory retrieval (see also, e.g., Chen et al., 2020; Compton & Logan, 1991; Zbrodoff, 1995, 1999). Furthermore, the addend slope during the transfer phase on Session 13 was again significant, implying that there was no transfer and indicating the item specificity of alphabet-arithmetic learning.

However, these classical findings have recently been put into question (see Chapter 2). We argued that the non-significant addend slope at the end of Logan and Klapp (1991)'s training experiment was due uniquely to the decrease of solution times for problems with the largest addend in the study set. In Chapter 2 we showed that when these problems were excluded from the analysis, the addend slope was significant until the end of training. This decrease in solution times for problems with the largest addend was observed only in a minority of participants that we called the breakers (i.e., 6 out of 19 in Experiment 1 and 7 out of 21 in Experiment 2). For participants who did not show the discontinuity in solution times, the addend slope remained significant for all addends until the end of the training experiment. This constitutes a challenge for the instance theory of automatization (e.g., Logan, 1988) because if the slope is not null at the end of training, the possibility that its reduction and the decrease of solution times during practice are caused by an acceleration of counting procedures cannot be discarded.

Nevertheless, most alphabet-arithmetic studies (e.g., Compton & Logan, 1991; Logan & Klapp, 1991; Zbrodoff, 1995, 1999, see also Chapter 2) were based on a verification task. This can be problematic because counting or memory retrieval could be bypassed in a verification task by the use of plausibility judgements (Reder, 1982). In mental arithmetic, such judgements involve the evaluation of the equation as a whole without exact calculations (e.g., Zbrodoff & Logan, 1990). This includes the situations where the proposed answer deviates largely from the correct answer (e.g., Ashcraft & Battaglia, 1978; De Rammelaere et al., 2001; Zbrodoff & Logan, 1990), when the parity of the proposed answer differs from the parity of the expected result, e.g., $4 + 2 = 7$ can be easily judged as incorrect because the sum of two even numbers should be an even number (Krueger, 1986; Krueger & Hallford, 1984; Lemaire & Fayol, 1995; Lemaire & Reder, 1999; Masse & Lemaire, 2001), when the proposed answer to a multiplication problem involving 5 does not contain 0 or 5 (Lemaire & Reder, 1999; Masse & Lemaire, 2001), or when the equation is familiar, e.g., $3 \times 4 = 12$ can be easily judged as correct because it has been frequently practiced (e.g., Lochy et al., 2000).

Another problematic aspect concerning verification tasks is that solution times depend on whether the presented equation is true or false. Indeed, studies on both mental arithmetic (e.g., Ashcraft & Battaglia, 1978; Ashcraft & Fierman, 1982; Ashcraft & Stazyk, 1981; Campbell, 1987a; Groen & Parkman, 1972; Parkman & Groen, 1971) and alphabet-arithmetic (e.g., Compton & Logan, 1991; Logan & Klapp, 1991; Zbrodoff, 1999, see also Chapter 2) using a verification task have shown that solution times are faster for true than for false equations. Furthermore, particularly in alphabet arithmetic, solution times in verification tasks depend on whether the proposed answer precedes or succeeds the correct answer (Zbrodoff, 1999, see also Chapter 3). For mental arithmetic studies, Ashcraft and Battaglia (1978) explained the difference in solution times between true and false equations by arguing that in a verification task, the evaluation of correctness is executed only after the correct answer has been found. In fact, whereas a production task involves three stages – i.e., encoding of the problem, searching or computing the answer to the problem, and providing the answer –, a verification task involves four stages – i.e., the same three stages as in a production task plus the evaluation of the response, wherein the proposed answer in the equation is compared to the correct answer – (Ashcraft, 1982; Ashcraft et al., 1984). In short, verification is production plus comparison. Within a verification task, the evaluation stage depends on the split or distance effect, i.e., the rejection times increase with the distance between the correct answer and the proposed answer (Ashcraft & Battaglia, 1978). This is why solution times for true equations are shorter than those for false equations.

Considering that solution times are often regarded as the mirror of the processes implied in problem solving but that, as already described, solution times in verification and production tasks can differ, Baroody (1984) asserted that solution times in verification tasks are inevitably not representative of the genuine times it takes to solve a problem in an ecological situation. Furthermore, assuming that memory retrieval is used to solve the problem, Campbell (1987b) argued that memory access to the correct answer might be facilitated by the presented answer in the equation. Therefore, according to him, succeeding in a verification task does not necessarily imply that the participant has correctly retrieved the answer. The arguments put forward by Baroody and Campbell make it obvious that verification tasks are generally less ecological than production tasks.

Therefore, the choice between verification and production tasks is crucial when mental arithmetic is investigated, and particularly when the alphabet-arithmetic paradigm is used. Indeed, past conclusions based on the results obtained with this paradigm could be dependent on the over reliance on methodologies based on verification. Moreover, as already stated, this paradigm is supposed to mimic the way children learn additions because both addition and alphabet-arithmetic tasks have to be learnt initially by way of counting and scanning through a familiar sequence. Nevertheless, in real life, children do not learn additions by means of a verification task and, therefore, the results obtained in alphabet-arithmetic verification tasks might not be directly generalisable to addition learning. Thus, by adopting a production task in this chapter, we aim at verifying that the results from alphabet-arithmetic verification tasks are replicable in a more-ecological production task. Although several studies using production tasks in alphabet arithmetic have already been conducted (Campbell et al., 2016; Chen et al., 2020; Pyke et al., 2019; Pyke & LeFevre, 2011; Rabinowitz & Goldberg, 1995; Rickard, 2004), the current chapter is the only one allowing for a direct comparison between verification and production tasks in alphabet-arithmetic learning. To do so, we designed a training experiment with a production task using exactly the same stimuli as in the verification training reported in Experiment 2 of Chapter 2. This material was very similar to the one constructed by Logan and Klapp (1991) except that 8 instead of 10 consecutive letters were paired with addends from 2 to 6 instead of 2 to 5. Despite these small differences between Logan and Klapp and our material, the central variables, namely the number of problems to learn and the number of repetitions for each problem,(i.e., 40 problems and 12 repetitions per session) were kept constant across experiments. Finally, as in Chapter 2 or Logan and Klapp, the present experiment consisted of 12 learning sessions. This training program was followed by 3 transfer sessions for which the results will not be reported in the present chapter.

If the results obtained in an alphabet-arithmetic task using a verification paradigm are replicable in a more ecological production task, we should observe the discontinuity in solution times found in previous alphabet-arithmetic studies (e.g., Compton & Logan, 1991; Logan & Klapp, 1991; Wenger, 1999; Zbrodoff, 1995, 1999, see also Chapter 2). As already explained above, this discontinuity corresponds to a decrease in solution times for problems with the largest addend. Secondly, when the problems with the largest addend are excluded from the analyses, the residual addend slope should still be significant at the end of training, implying no sign of shift from counting to retrieval.

4.2 Method

Participants

Twenty-three students (6 females) aged between 18 and 32 years were recruited by means of the student-job websites of the University of Lausanne and the Swiss Federal School of Technology in Lausanne. All participants were native French speakers and they received CHF 190 for their participation.

Written informed consent was obtained for each participant. All procedures performed in this study, involving human participants, have been conducted in compliance with the Swiss Law on Research Involving Human Beings. Because only behavioural data were collected in a non-vulnerable population of adults, the approval of the Canton de Vaud ethics committee was not required. The study was carried out in accordance with the recommendations of the Ethics Committee of the University of Lausanne, following the 1964 Helsinki declaration and its later amendments or comparable ethical standards. Finally, the research protocol that we followed was approved by the Research Committee of the Faculty of Social and Political Sciences of the University of Lausanne.

Material and stimuli

Participants were trained on an alphabet-arithmetic production task (e.g., A + 2 = ?). Half of the participants were assigned to Group 1 and the other half to Group 2. During the learning phase, participants in Group 1 were trained on the first 8 letters of the alphabet (i.e., Set 1: letters A to H) and those in Group 2 on the second 8 letters (i.e., Set 2: letters I to P). Each letter was paired with addends from 2 to 6, resulting in 40 problems in each set. Each problem was presented 4 times in a block, and each session comprised 3 identical blocks of 160 trials. The 160 problems were randomised within each block. Thus, similar to Experiment 1 of Logan and Klapp (1991) and the two experiments of Chapter 2, the stimuli contained 40 problems that were presented 12 times in a session.

The experiment was programmed with the DMDX software (Forster & Forster, 2003). Each trial began with a fixation point (*) presented for 500 ms, followed by the problem, which remained on the screen until participants gave their response orally into the microphone. Then, the problem disappeared from the screen and the screen remained blank for 500 ms until the onset of the next trial.

Participants' responses were recorded in individual .WAV audio files. Solution times, which corresponded to the time elapsed between problem presentation and voice-key triggering, were recorded in a separate file. For some trials, the intensity of participants' responses did not reach the threshold at which the voice key could be triggered and the problem remained on the screen until participants repeated their response louder. In such voice-key failure cases, recorded solution times were not correct and they were therefore corrected manually using the CheckVocal software (Protopapas, 2007). The latter software also allows for the verification of the response accuracy.

Procedure

Participants were trained across 12 sessions, corresponding to 12 consecutive working days. They were tested individually in our laboratory, in separate experimental booths. The experimenter was present in the room where the experimental booths are located, but outside the booth. During the weekend, participants were required to do one session of home training consisting of 160 problems presented on paper that they had to solve as quickly and accurately as possible. The 160 problems corresponded to one experimental block.

4.3 Results

We excluded the data of 4 participants either because the accuracy was too low (less than 75% of correct responses for at least 2 sessions) or because the number of recording errors was too high (i.e., more than 20% for at least 3 sessions). The data of 2 other participants were also excluded because they showed non-significant addend slopes in Session 1. This was done because the alphabet-arithmetic task is conceived with the assumption that participants would start the learning process by a counting procedure, which implies significant addend slopes. It is therefore obvious that these 2 participants had never solved the problem through counting. Thus, the data of 17 participants were included in the analyses, i.e., 10 in Group 1 and 7 in Group 2.

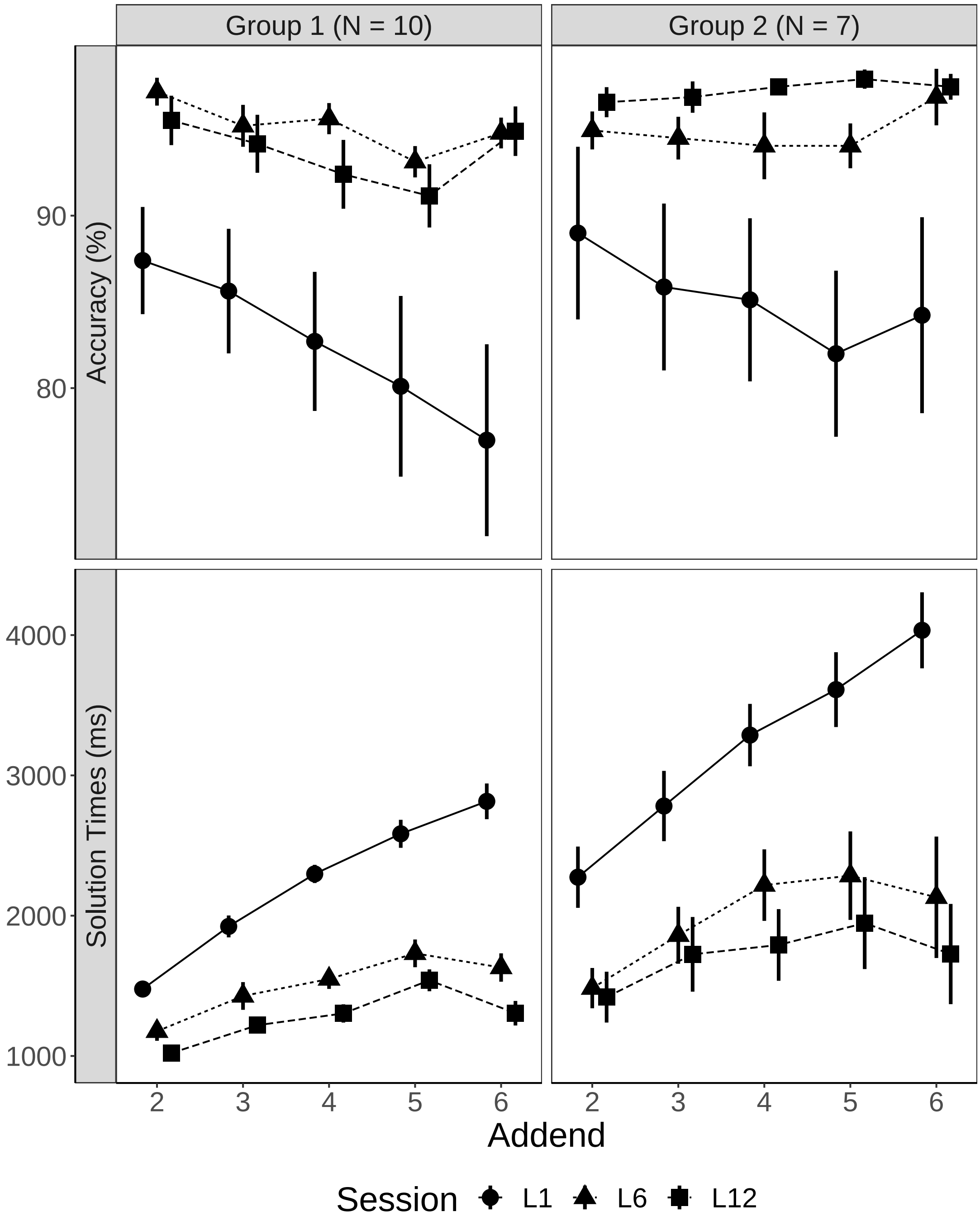
Accuracy

We first carried out a 12 (Session: 1 to 12) $\times$ 5 (Addend: 2 to 6) $\times$ 2 (Group: 1 or 2) repeated-measures, mixed-design ANOVA on accuracy with Group as a between measure (see the top panel of Figure 4.1 for accuracy in Sessions 1, 6, and 12). First of all, an effect of Group was found ($F(1, 15) = 5.71$, $\eta_p^2 = .28$, $p = .03$), with Group 1 participants having lower accuracy (92%) than Group 2 participants (96%). We also found an effect of Addend ($F(4, 60) = 9.55$, $\eta_p^2 = .39$, $p < .001$), with +2 problems being solved with the highest accuracy (95%) and +5 problems with the lowest accuracy (92%). There was also an interaction between Addend and Group ($F(4, 60) = 3.44$, $\eta_p^2 = .19$, $p = .01$). A series of contrasts with Holm correction showed that this interaction was due to +5 and +6 problems being solved with lower accuracy by Group 1 participants (89% and 92% for +5 and +6 problems, respectively) than by Group 2 participants (95%, $t(15) = 3.12$, $p = .007$ for +5 problems and 96%, $t(15) = 2.67$, $p = .02$ for +6 problems) whereas there was no difference in accuracy between the two groups for problems with addends 2, 3, and 4.

We further found an effect of Session ($F(11, 165) = 5.71$, $\eta_p^2 = .27$, $p < .001$), with accuracy increasing from 84% in Session 1 to 95% in Sessions 6 and 12. This effect did not interact with Group ($F(11, 165) < 1$) but interacted with Addend ($F(44, 660) = 1.91$, $\eta_p^2 = .11$, $p < .001$). A series of contrasts with Holm correction revealed that the interaction was due to the significant linear addend effect in Sessions 1 ($t(15) = -3.53$, $p = .01$), 2 ($t(15) = -4.12$, $p = .004$), and 3 ($t(15) = -2.89$, $p = .04$), with higher accuracy for lower addend. This addend effect disappeared from Session 4 onwards. The three variables did not interact ($F(44, 660) = 1.12$, $p = .28$).

Figure 4.1

Accuracy and Solution Times as a Function of Addends



Note. Accuracy (top panels) and solution times (bottom panels) as a function of addends for Sessions 1 (circles, solid line), 6 (triangles, dotted line), and 12 (squares, dashed line) for Group 1 (left panels) and Group 2 (right panels) participants. Error bars represent standard errors.

Solution Times

To analyse solution times, we removed invalid trials, i.e., faulty trials due to technical problems and trials solved incorrectly, which together corresponded to 6.75% of all trials. Furthermore, we removed correct trials with extreme values, which corresponded to 0.33% of the correct trials. The extreme values were defined as trials with solution times shorter than 250 ms as well as trials with solution times larger than the mean for each participant and each session plus 3 times the corresponding standard deviation.

We performed a 12 (Session: 1 to 12) × 5 (Addend: 2 to 6) × 2 (Group: 1 or 2) repeated-measures, mixed-design ANOVA on solution times with Group as the between measure (see Figure 4.1 for solution times in Sessions 1, 6, and 12). An effect of Group was found ($F(1, 15) = 6.01$, $\eta_p^2 = .29$, $p = .03$), with Group 1 participants being faster (1558 ms) than Group 2 participants (2128 ms). An effect of Addend was also found ($F(4, 60) = 39.76$, $\eta_p^2 = .73$, $p < .001$), with +2 problems being solved the fastest (1388 ms) and +5 problems the slowest (2135 ms). Addend and Group did not interact ($F(4, 60) < 1$).

We found an effect of Session ($F(11, 165) = 47.12$, $\eta_p^2 = .76$, $p < .001$), with solution times decreasing from 2708 ms in Session 1 to 1749 ms in Session 6 to 1500 ms in Session 12. This effect interacted with Group ($F(11, 165) = 2.97$, $\eta_p^2 = .17$, $p = .001$), see the bottom panel of Figure 4.1. A series of contrasts with Holm correction revealed that this interaction was due to Group 1 participants being faster than Group 2 participants in Sessions 1 ($t(15) = 4.47$, $p < .001$), 2 ($t(15) = 2.90$, $p = .01$), and 3 ($t(15) = 2.29$, $p = .04$), but not in other learning sessions.

There was also an interaction between Session and Addend ($F(44, 660) = 16.67$, $\eta_p^2 = .53$, $p < .001$). The effect of Addend was significant throughout the learning sessions, i.e., from $t(15) = 14.89$, $p < .001$ in Session 1 to $t(15) = 6.14$,

$p < .001$ in Session 12. There was no three-way interaction ($F(44, 660) = 1.06$, $p = .36$).

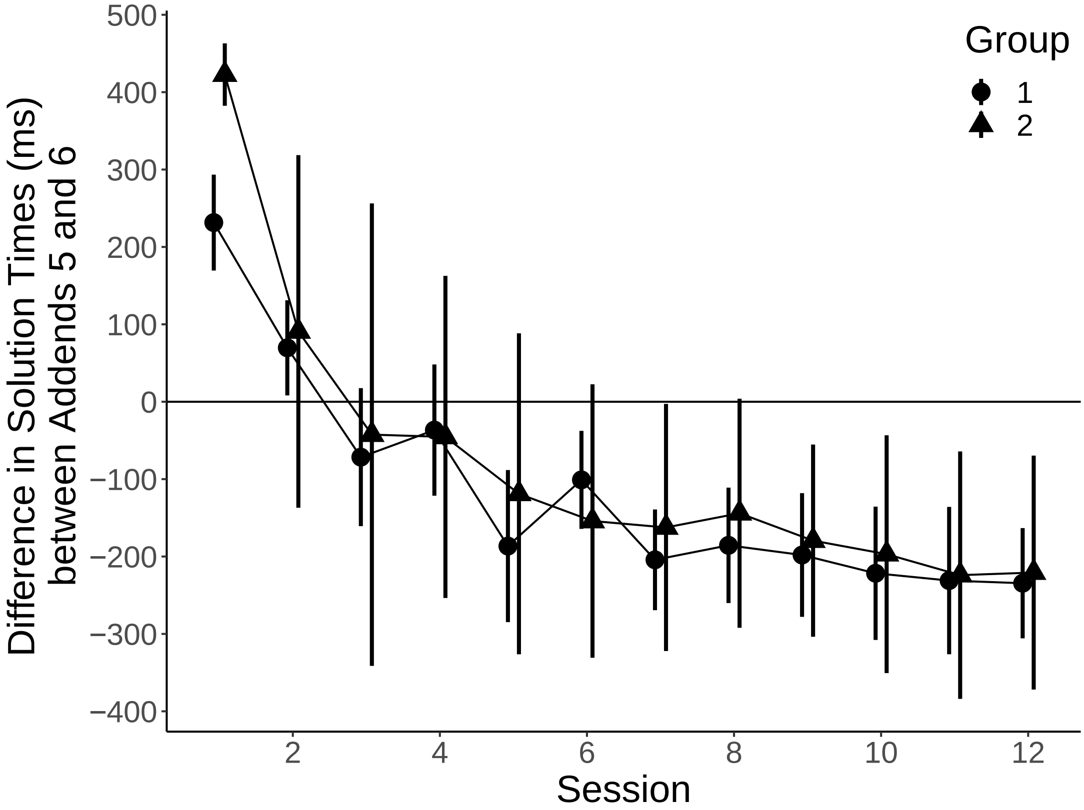
As observed when verification tasks are used, Figure 4.1 shows a discontinuity in solution times in Sessions 6 and 12. In other words, solution times for +6 problems were shorter than for +5 problems. In fact, for both Groups of participants, as can be seen in Figure 4.2, this discontinuity occurred for the first time on average in Session 3. Based on these observations at an individual level, we categorised participants according to whether or not they showed this discontinuity. Two non-breakers did not show a discontinuity in solution times at any point of the experiment. Ten breakers continuously showed a discontinuity starting from one session (i.e., as early as Session 1 and as late as Session 9) until the end of training. Finally, 5 participants did not show a consistent pattern, i.e., they showed a discontinuity in at least one session but the discontinuity disappeared in the following sessions. A χ^2-test of independence revealed that the categorization of participants into breakers, non-breakers, and inconsistent did not depend on the part of the alphabet on which they were trained on, i.e., Group 1 or Group 2, ($\chi^2(16, N = 17) = 2.92$, $p = 1$).

Addend slopes

We calculated the addend slopes for each participant and for each session. Because, as just described, the inclusion of problems with the largest addend potentially flattens the addend slopes, we also calculated the addend slopes without +6 problems. Whether +6 problems were included or not, the addend slopes were significantly different from 0 throughout the learning sessions, i.e., 385 and 411 ms/addend in Session 1, with and without +6 problems, respectively ($ps < .001$), 146 and 228 ms/addend in Session 6, with and without +6 prob-

Figure 4.2

Difference in Solution Times between +5 and +6 Problems as a Function of Sessions



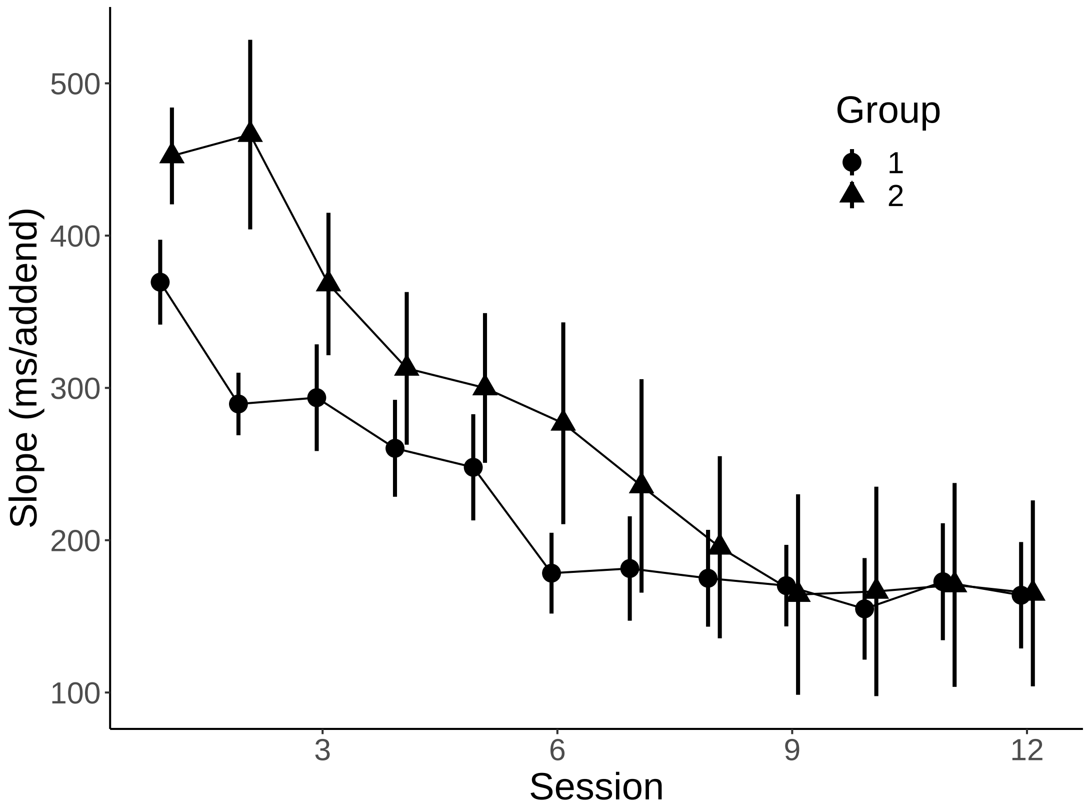
Note. Difference in solution times between problems with addends 5 and 6 across the 12 sessions. Error bars represent standard errors.

lems, respectively (ps $< .001$), and 86 ms/addend ($p = .01$) in Session 12 when +6 problems were included and 165 ms/addend when they were not ($p < .001$).

We ran a 12 (Session: 1 to 12) × 2 (Data Set: with or without +6 problems) × 2 (Group: 1 or 2) repeated-measures, mixed-design ANOVA on addend slope with Group as a between measure. We did not find an effect of Group ($F(1, 15) < 1$) but we found an interaction between Group and Session ($F(11, 165) = 2.12$, $\eta^2_p = .12$, $p = .02$). A series of contrasts with Holm correction revealed that this interaction was due to a significantly lower addend slope for Group 1 than Group 2, but only in Sessions 1 ($t(15) = 2.30$, $p = .04$) and 2 ($t(15) = 2.71$, $p = .02$), see Figure 4.3.

Figure 4.3

Addend Slopes as a Function of Sessions



Note. Addend slopes a function of sessions for Group 1 (solid circles) and Group 2 (solid triangles), without taking +6 problems into account. Error bars represent standard errors.

More importantly, we found an effect of Data Set ($F(1, 15) = 35.04$, $\eta_p^2 = .70$, $p < .001$), showing that including +6 problems (172 ms/addend) significantly flattened the addend slope compared to excluding them (247 ms/addend). The effect of Data Set did not interact with Group ($F(1, 15) < 1$) or Session ($F(11, 165) = 1.25$, $p = .26$). It was significant from Session 1 ($t(15) = 2.75$, $p = .01$) to Session 12 ($t(15) = 4.29$, $p < .001$). See Figure 4.3 for a representation of addend slopes without considering + 6 problems.

4.4 Discussion

In the present research, we investigated whether the results based on an alphabet-arithmetic verification task are replicated when a more-ecological production task is used. This question is particularly important in the current theoretical context because, as explained in Section 4.1, some assumptions of the instance theory of automatization (Logan, 1988) have been recently called into question using a verification task (see Chapter 2). It is therefore central to ensure that previous conclusions of the literature hold in a more-natural paradigm using a production task. To this aim, we replicated Logan and Klapp (1991)'s and our experiments in Chapter 2 using a production paradigm in an alphabet-arithmetic task rather than a verification paradigm, as used in the original experiments.

Exactly as in Chapter 2, we found a significant addend slope at the end of the learning phase. Therefore, contrary to Logan and Klapp (1991)'s conclusion, the possibility that the decrease of the slopes and the decrease in solution times at the end of an alphabet-arithmetic training is due to an acceleration of procedures rather than a progressive shift from counting to retrieval cannot be discarded. As also observed in previous studies, we found a decrease in solution times for problems involving the largest addend (+6 in the present experiment). Indeed, a discontinuity, or in other words, a drop in solution times was observed for these problems (see Figure 4.1), which were therefore obviously not processed as the others. The numerous counting steps required to solve problems with the largest addend probably discouraged participants to count. Deliberate memorisation of the associations between operands and answers might have therefore been preferred over counting (Logan & Klapp, 1991, see also Chapter 2). Still, similar to the finding in Chapter 2, the decrease in solution times, suggesting a deliberate memorisation of problems with the largest addend, was not found for all participants.

The observation of a discontinuity in solution times for problems with the largest addend in a production as in a verification task is important because it confirms that end-terms effects in the verification task were not fully responsible for the special processing of these problems. More precisely, in verification task where the false equations are constructed by adding a certain quantity to or subtracting from the correct answers (usually 1 or 2 to minimise the split between true and false answers), the answers proposed for the equations involving the largest addend are necessarily underrepresented. For example, in an experiment involving letters ranging from A to H, addends from 2 to 6, and a split of 1 between the correct and the proposed false answer, the letter O is presented only when H + 6 = O has to be verified. This false equation including the letter O is therefore extremely salient and can be rejected easily. In a production task, such unavoidable statistical irregularities in the material cannot impact the results.

Our result that production and verification tasks lead to the same pattern of results has important implications. It shows that the drop in solution times observed for problems with the largest addend is a generalisable phenomenon. However, in the current chapter using a production task, we found more breakers (i.e., 10 out of 17) than when a verification task was used in Chapter 2 (i.e., 7 out of 21). Furthermore, the discontinuity observed in the present chapter occurred earlier during practice than in Chapter 2's verification task, i.e., Session 3 instead of Session 7. It seems therefore that a production task more strongly elicits deliberate memorisation of the associations between the elements of the problems and their answers for problems with the largest addend. An interpretation of this result will be provided later in this section.

It is very important to note that the decrease in solution times for problems with the largest addend challenges the instance theory of automatization. As explained by Logan and Klapp (1991), both repeated counting and deliberate memorisation should lead to the creation of instances in long-term memory.

Indeed, in the framework of the instance theory, what is important for automatisation, or in other words for memory retrieval, is the number of traces made and not the way they are created. In the current chapter, as well as in Logan and Klapp (1991) or Chapter 2, the number of presentations of problems with the largest addend and the number of presentations of other problems is exactly the same. Therefore, according to the instance theory of automatization, there is no reason for problems with the largest addend to be committed faster to memory than other problems. Thus, the lower solution times observed for these problems compared to problems involving an addend immediately inferior to the largest demonstrates that they are not subject to the principles described in the instance theory of automatization. As a consequence and at the very least, the addend slopes in alphabet-arithmetic tasks have to be calculated after the exclusion of problems with the largest addend. As shown in the present chapter using a production task, in Chapter 2 using a verification task, and as estimated from Logan and Klapp (1991) depiction of data, excluding these problems results in a significant addend slope throughout the experiment, from the beginning until the end. As already explained, following Logan and Klapp's rationale that "memory retrieval should produce a slope of zero in the linear function relating reaction time to the magnitude of the digit addend" (p. 180), a decrease in the addend slope across sessions, without its disappearance, is not sufficient to infer a shift from counting to retrieval during the course of training. This invalidates Logan and Klapp's conclusion but we cannot conclude that there was no shift towards retrieval during the experiments. Nevertheless, we can conclude that there is no sign of this shift from the evolution of addend slopes in alphabet-arithmetic tasks.

We have just shown and discussed that the qualitative results we observed concerning our variables of interest are very similar in a production and a verification task. We will now examine whether the results between the two tasks are also quantitatively similar when we consider the other variables that we analysed. The following comparisons are made between the results obtained in the production task reported in this chapter and the true equations in the verification task reported in Chapter 2. As a reminder, the material used in the two experiments is strictly the same. Concerning accuracy, the percentage of errors at the beginning of learning was descriptively higher in the production task (i.e., 8% and 5% in Session 1 for the production and verification tasks, respectively) but this small difference completely disappeared at the end of learning (i.e., 4% in both verification and production tasks). Therefore, the two tasks resulted in very similar error rates. Concerning solution times, it is difficult to make direct comparisons because of the difference in the way solution times are measured in the two tasks, i.e., oral response in the production task versus keyboard pressing in the verification task. We can nevertheless compare the decrease in solution times from the first to the last sessions of the learning phase. Again, they were very similar (i.e., a decrease of 49% in the verification task and of 45% in the production task).

Finally, concerning the magnitudes of the addend slopes, they were lower in the production than in the verification task. In the first session, the addend slopes were 385 and 411 ms/addend for the production task and 441 and 487 ms/addend for the verification task when +6 problems were included and excluded, respectively. Interestingly, the addend slopes at the end of the verification task were comparable to the addend slopes in the middle of the production task. More precisely, addend slopes in Session 12 of the verification task were 163 and 236 ms/addend, when +6 problems were included and excluded, respectively, whereas addend slopes in Session 6 of the production task were 146 and 228 ms/addend, when +6 problems were included and excluded, respectively. In Session 12, addend slopes in the production task were much lower than in the verification task, i.e., 86 and 165 ms/addend, when +6 problems were included and excluded, respectively.

Concerning the set of problems including those with the largest addend, an explanation for smaller slopes at the end of training in the production task can be found in light of the results we obtained concerning breakers and non-breakers. As already noted, there were more breakers among the participants in the present chapter using a production task compared to the verification task used in Chapter 2, and the breakers in the production task showed the discontinuity in solution times earlier during the learning phase. Furthermore, in the end of the learning session, the difference in solution times between +5 and +6 problems was about 100 ms in the verification task (see Figure 2.8) and about 200 ms in the production task (see Figure 4.2). All these results show that deliberate memorisation of the problems with the largest addend is more prominent in a production than in a verification alphabet-arithmetic task.

One possible interpretation is that in a verification task, the false answers that are proposed in half of the trials interfere with the correct answers, hence more difficult associations between the different elements of the problems (e.g., Siegler & Shrager, 1984). Concerning the set of problems without the largest addend, smaller slopes in the production than in the verification task could be due to a higher acceleration of counting procedures in the production task. Alternatively, this difference could be due to more numerous shifts from counting to retrieval in the production than in the verification task. As already mentioned, it is not because such shifts are not evidenced by our results that they never occur.

Nevertheless, it is unlikely that further training in a verification task would lead to the same level of performance as in a production task, and a fortiori, would lead to a complete shift from counting to retrieval. Indeed, in Experiment 1 of Chapter 2, we ran an alphabet-arithmetic task over 25 instead of 12 sessions and showed that the addend slopes across sessions were always different from 0 (i.e., from Session 1 to Session 25). More crucially for the present point, there was no significant difference in the size of addend slopes between Sessions 12 and 25.

The asymptote was therefore reached by Session 12, showing that from this point onwards, there was no further evolution in participants' strategy choices.

To sum up, the overall pattern of results obtained in the present study using a production task replicates the results obtained in a verification task. We can therefore conclude that verification and production tasks rely on the same general cognitive mechanisms, at least when the split in the verification task between the correct and the proposed answer is small. It would be interesting to test in future studies whether manipulating the size of the split can affect alphabet-arithmetic tasks (e.g., D + 3 = P).

In fact, even if we show here that using a production or a verification task provide similar results, this does not mean that, in the previous literature, all studies using a verification task could have been conducted using a production task and vice-versa. Rather, the choice of the task depends on the purpose of the study. If the goal of the researchers is to collect precise and ecological data, then a production task is more appropriate. However, such level of precision is possible only when participants give their response orally and when, in order to correct for voice-key failures, solution times for each of the oral response are manually adjusted to correspond to the onset of the spectrogram (e.g., Poletti et al., 2021).

Despite the precision of such approach, not any questions can be answered directly using a production task. The distance (i.e., split) between the proposed and the correct answers can obviously be manipulated only in a verification task. As already evoked above, this kind of manipulation allowed researchers to discover that when the split is large, individuals do not always engage in a costly solution process leading to the exact answer but can decide that the answer is false on the basis of a plausibility judgement (e.g., Duverne & Lemaire, 2004, 2005; Hinault et al., 2016). The question of whether individuals have interiorised and can use rules such as the parity rule (e.g., the addition of two even numbers cannot result in an odd number) or the multiple-of-five rule (i.e., multiplying a number by 5 necessarily results in an answer ending by a 0 or a 5) can also be easily addressed using verification tasks (e.g., Krueger, 1986; Masse & Lemaire, 2001). Within such design, it is possible to directly observe whether a false equation is rejected quicker when the proposed answer violates the rule than when it does not. It is also possible to infer such rule use in production tasks by comparing solution times on different problems (e.g., involving a 5 or not, Miller et al., 1984) but this approach seems to be more inferential than using a verification task.

Finally, verification tasks can sometimes be more appropriate when researchers aim at recording brain activity (e.g., Avancini et al., 2014; Mathieu, Epinat-Duclos, Sigovan et al., 2018). Given that arithmetic problems can be mentally represented in a verbal format (Dehaene, 1992), interference between an oral answer and the problem-solving process can be more detrimental to recordings than interference between the solving process and a purer motor task (i.e., pressing a key for decision). Still, to overcome these complications, delayed production or delayed verification tasks can also be used (e.g., Bagnoud, Dewi & Thevenot, 2021; Didino, 2011).

5

Individual Differences in the Evolution of Counting: The Case of Alphabet-Arithmetic Tasks

Adapted from:
Dewi, J. D. M., & Thevenot, C. (In revision). Individual differences in the evolution of counting: The case of alphabet-arithmetic tasks.

Abstract

Children start learning addition by counting. In the literature, two contrasting views have been proposed to describe how counting procedure evolves. On the one hand, the retrieval models purport that in the course of learning, counting is replaced by the retrieval of the association between operands and the sum. On the other, the automated counting procedure theory claims that slow counting is accelerated until automatisation. To test these two theories, we ran two training experiments using the alphabet-arithmetic paradigm (e.g., D + 3 = G), which was conceived to mimic the way children learn addition. In both experiments, a 12-session learning phase was followed by a 3-session transfer phase, and participants had to learn problems with addends from 2 to 6. Based on participants' performance, we identified 3 groups. In the first group, counting was the dominant strategy until the end of learning. In the second, retrieval was the dominant strategy at the end of learning. In the last group, retrieval was mainly used for problems with the largest addend whereas multiple strategies were used for problems with smaller addends. The individual differences revealed in this paper may reflect individual differences in children's learning of mental addition.

Keywords: Strategies; Counting; Retrieval; Transfer; Arithmetic

5.1 Introduction

In the domain of skill acquisition, it has been widely acknowledged that individuals learn differently. Individual differences in learning have been studied in a wide variety of domains, whether it concerns school-related subjects such as reading and mathematics (e.g., Dulaney et al., 2015), scientific thinking (e.g., Koerber & Osterhaus, 2019), music (e.g., Okada & Slevc, 2018), or more-basic cognitive skills, for example statistical learning (e.g., Kidd & Arciuli, 2016). In the domain of mental arithmetic in particular, working memory (e.g., Clearman et al., 2017) and processing speed (e.g., Geary, 2011) have long been thought as important cognitive factors that influence individual differences in performance. From the point of view of specific numerical abilities, the understanding of identity and commutativity principles (e.g., Dowker, 2014), the ability to complete sequence patterns (e.g., MacKay & De Smedt, 2019), and the symbolic number-magnitude representation (e.g., Vanbinst et al., 2012) may also play a role. Moreover, individual differences in arithmetic are not only observed at the level of performance but also in individuals' use of strategy. In this paper, we are interested in investigating individual differences in the strategy used during addition learning.

That several strategies are used to solve simple arithmetic has been found in both children (e.g., Siegler & Robinson, 1982; Svenson & Broquist, 1975) and adults (e.g., Bagnoud, Dewi & Thevenot, 2021; LeFevre, Sadesky et al., 1996; Svenson, 1985; Thevenot et al., 2010). In this respect, several possibilities can be envisioned within a sample of individuals, either at inter- and intra-individual levels. Inter-individual differences include situations where different strategies are used by different individuals to solve the same problems, whereas intra-individual differences include situations where different strategies are used by the same individuals to solve either different problems or the same problems. To distinguish these two types of intra-individual differences, hereinafter we will apply the term inter-problem differences for the use of different strategies by the same individuals to solve different problems and intra-problem differences for the use of different strategies by the same individuals to solve the same problems.

In his overlapping-wave model, Siegler (1996) explains intra-individual differences in arithmetic by the fact that, at any given time in development, children have several strategies in their repertoire. These range from the ones based on procedures, such as counting to solve an addition problem or repeated addition to solve a multiplication problem, to direct retrieval. According to this model, children adaptively choose which strategy to use, depending on the perceived difficulty of the task, the trade-off between the effort or time and accuracy, and the circumstances of the task, for example whether the task demands accuracy or speed. However, Siegler argued that in the course of development, retrieval-based strategy will be the privileged one whereas procedure-based strategies serve as a backup, only to be used when retrieval fails. Protocol analyses studying adults' strategies (e.g., Geary & Wiley, 1991; Hecht, 1999, 2002; LeFevre, Sadesky et al., 1996; LeFevre et al., 2003; Svenson, 1985; Uittenhove et al., 2016) showed indeed that retrieval is self-reported as being the most used strategy, though it is not the only strategy self-reported as being used. In fact, only a minority of adults self-reported of using it exclusively to solve simple additions, e.g., 2 out of 16 participants in the study of LeFevre, Sadesky et al. (1996).

The progression in the use of strategy, i.e., from counting in novices or children to retrieval in experts or adults, which was proposed by the overlapping-wave model, is the core of retrieval models of mental arithmetic (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996). To date, supports for the retrieval models have come mostly from the instance theory of automatization (Logan, 1988) by means of the alphabet-arithmetic paradigm, wherein a number addend is added to a letter augend resulting in a letter answer (e.g., C + 3 = F). Logan and Klapp (1991) showed a decrease in the addend slope in the course of a 12-session learning, from a significant addend slope of 486 ms/addend in Session 1 to a non-significant addend slope of 45 ms/addend in Session 12. Considering that a significant addend slope indicates the use of counting and a non-significant one the use of retrieval, Logan and Klapp concluded that participants must have undergone a shift from counting to retrieval in the course of the experiment.

However, some researchers argue that retrieval is not the dominant strategy that experts use to solve simple additions. Baroody (1983, 1984, 1994, 2018), for example, suggested that adults could also solve such problems by relying on rules and heuristics. In the same line, more recently Thevenot and collaborators (Bagnoud, Dewi, Castel et al., 2021; Barrouillet & Thevenot, 2013; Fayol & Thevenot, 2012; Thevenot & Barrouillet, 2020; Uittenhove et al., 2016) put forward the automated counting procedure theory, according to which a one-by-one counting procedure is accelerated in the course of development until automatisation, at least for additions involving two operands inferior to 5. As support for this theory, recent results from an alphabet-arithmetic paradigm using a verification (Chapter 2) and a production task (Chapter 4) found significant addend slopes at the end of learning. Following Logan and Klapp (1991)'s reasoning, these significant addend slopes would imply the use of counting.

More precisely, previous alphabet-arithmetic studies have systematically shown that solution times increase as a function of addend up to the second-largest addend and then decrease for the largest addend, irrespective of the size of the largest addend in the study set (e.g., Compton & Logan, 1991; Logan & Klapp, 1991; Wenger, 1999; Zbrodoff, 1995, 1999, ; see also Chapters 2 and 4). Chapter 2 showed that when problems with the largest addend are excluded from the analysis, the addend slope was still significant, implying that the use of counting cannot be discarded. Furthermore, it was found that this decrease was observed only in a minority of participants, i.e., 6 out of 19 in Experiment 1 and 7 out of 21 in Experiment 2, implying that retrieval was used only by a minority of participants, and to solve only a minority of problems. This minority of participants were called the breakers because they showed a discontinuity in solution times as a function of addend, i.e., solution times for +5 problems were shorter than those for +4 problems, from one session (i.e., between Sessions 1 and 17 in Experiment 1 and between Sessions 1 and 12 in Experiment 2) to the last session of the learning phase. Oppositely, another group of participants were called non-breakers because they did not show a discontinuity in solution times throughout the learning sessions.

The classification of participants into breakers and non-breakers in Chapters 2 and 4could be paralleled to the categorisation of participants into shifters and non-shifters in the work of Haider and Frensch (2002). In an alphabet-arithmetic training study, the latter authors defined shifters as participants who showed an important decrease of solution times from one session to the next, i.e., more than 1000 ms. After 10 blocks of learning, Haider and Frensch ran 2 blocks of transfer where participants had to solve new items. The authors found that shifters during the learning phase were also shifters during the transfer phase, suggesting that the choice of strategy is not influenced by the study set but is the result of participants' intentional decision. Considering that we (Chapters 2 and 4) and Haider and Frensch did not use the same material construction, it is difficult to evaluate whether the shifters in Haider and Frensch's study correspond to the breakers in Chapters 2 and 4. Nevertheless, it would be interesting to investigate whether, like the shifters in Haider and Frensch's experiment, participants who show a break during the learning phase in an alphabet-arithmetic task also show a break during the transfer phase. This is the first aim of the current chapter. To achieve this aim, we will analyse the data from an alphabet-arithmetic verification task, wherein 12 learning sessions were followed by 3 transfer sessions. During the transfer sessions, participants had to solve new problems that they had not yet encountered during the learning sessions. To reach the first aim, we will examine whether, in the transfer phase, solution times for +6 problems in breakers are also shorter than for +5 problems.

As already stated, we concluded in Chapter 2 that, in the breaker group, problems with the largest addend were solved by memory retrieval whereas problems with the smaller addends by counting. In the present paper, we will not only better describe these intra-individual differences but we will also examine inter-individual differences in the strategy used within the breaker group. In Chapter 2, we already described in our Experiment 1 including addends from 2 to 5 that some participants presented a solution-time discontinuity, or break, at +3 rather than at +4 problems. Interestingly, this break at +3 problems seems to indicate a more-advanced level in the learning process, because it was always preceded by a break at +4 problems in earlier sessions. In other words, those participants who showed a break at +4 problems have memorised +5 problems whereas those who showed a break at +3 problems have memorised +5 and +4 problems. Thus, apart from intra-individual differences in the form of inter-problem differences (i.e., counting or retrieval depending on the problems), there seems also to be inter-individual differences within the breakers. Moreover, it is also possible that intra-individual differences in the breaker group go beyond inter-problem differences. More precisely, the same individuals may use both retrieval and counting for problems with the same addend, i.e., intra-problem differences. The use of multiple strategies for the same problems can be understood in the framework of the overlapping-waves model (e.g., Siegler, 1996), according to which there is a gradual transition from using more-rudimentary strategies towards using more-advanced strategies. On the other hand, the strategy choice for a given problem may be influenced, for example, by the strategy used in preceding trials. More precisely, changing strategies between two trials leads to poorer performance than using the same strategy across trials (e.g., Lemaire & Lecacheur, 2010; Luwel et al., 2009). The second aim of the present chapter is therefore to investigate such intra-problem differences, that could give indications about the evolution of counting strategy during learning.

To reach the second aim or more precisely to study the use of multiple strategies in alphabet arithmetic, we will compare performance, which is operationalised by the addend slopes, between the last learning session of the training and during the 3 transfer sessions. Transfer will be examined by first removing problems with the largest addend from the analysis. This precaution is necessary because, as already explained above, these problems are thought to be processed differently from problems with smaller addends (Logan & Klapp, 1991, see alsi Chapter 2). More importantly, we found that problems with the largest addend led to solution-time discontinuity for breakers but not for non-breakers (Chapter 2).

In the present chapter, transfer will be studied separately for breakers and non-breakers. Note that according to the instance theory of automatization (Logan, 1988), learning is item specific and therefore there is no transfer from learnt to new items. Nevertheless, because in Chapter 2 we found that addend slopes at the end of training were still significant when problems with the largest addend were excluded, transfer could still be observed for problems with smaller addends. For the non-breakers, because there is no discontinuity in solution times during the learning phase, we can confidently assume that counting is the dominant strategy. Considering that procedural knowledge can transfer to new items (e.g., Singley & Anderson, 1989; VanLehn, 1996), we predict for this group of participants that transfer will be observed already in the first transfer session. For the breakers, on the other hand, two possibilities can be considered. If, after removing problems with the largest addend, transfer is observed in the first session, then the significant addend slope observed at the end of training after excluding problems with the largest addend could be safely interpreted as the sign of counting for problems with smaller addends. In this case, intra-individual differences would only concern inter-problem differences, i.e., the use of memory retrieval for problems with the largest addend and counting for problems with smaller addends. However, if transfer is not observed in the first session, then this would indicate that the significant addend slope observed at the end of learning was not only due to counting procedures. Instead, significant addend slopes may have resulted from averaging solution times across different strategies (e.g., Siegler & Robinson, 1982). In this case, intra-individual differences concern also intra-problem differences, i.e., the use of different strategies for problems with the same addends.

5.2 Experiment 1

Method

The complete experiment comprised 12 sessions of learning, corresponding to 12 working days, followed by 3 sessions of transfer, corresponding to 3 working days. The method for the learning sessions has been described in Section 2.3 (page 71). During the transfer phase, participants who were trained with Set 1 during the learning phase had to worked with Set 2, and vice versa. Following the suggestion of Speelman and Kirsner (2001), the learning and transfer phases were not separated by a week-end. Furthermore, the 3 transfer sessions were not interrupted by a week-end. During the week-end, participants were asked to undertake a home training that consisted in verifying as quickly as possible 160 equations presented on paper. These 160 equations corresponded to one block of experiment in the laboratory.

Results

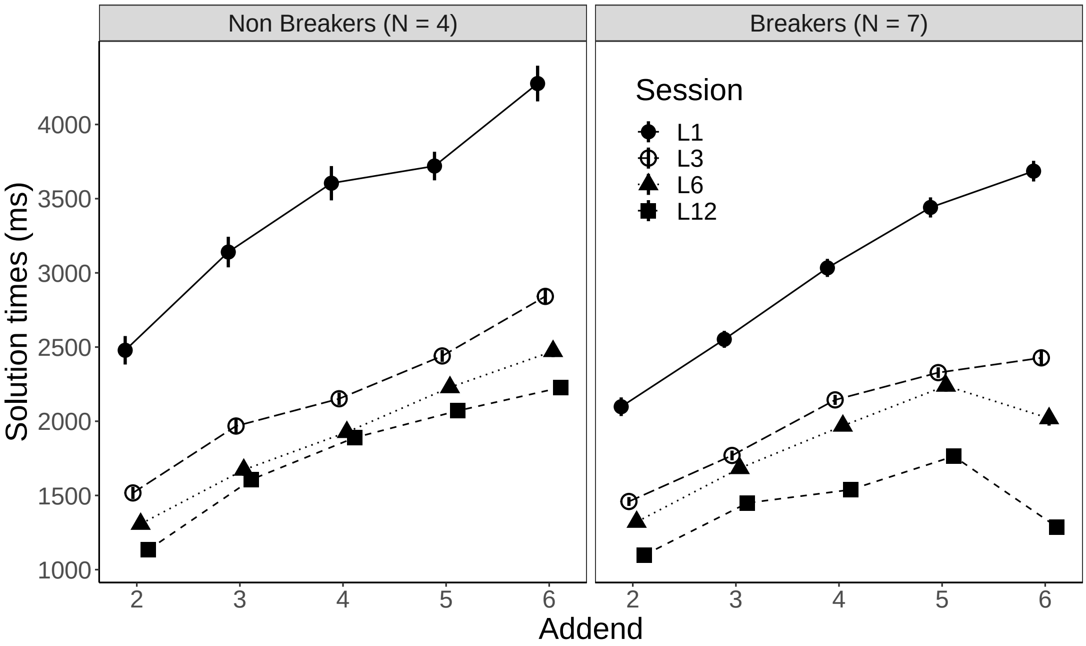
Considering that we expected participants to use counting strategy at the beginning of the experiment, we excluded the data of 3 participants who already showed non-significant addend slopes at the beginning of learning. Furthermore, for the analyses reported in this paper we analysed only true-equation trials that were solved correctly and with solution times that were not categorised as extreme values, defined as trials with solution times shorter than 300 ms or larger than the mean for each participant and each session plus 3 times the corresponding standard deviation. For the whole experiment, incorrectly-solved trials and trials with extreme solution times comprised 5.08% of the data from the remaining 21 participants.

For the sake of conciseness, we labelled the learning sessions with the letter L, i.e., from L1 to L12, and the transfer sessions with the letter T, i.e., from T1 to T3. The results for the learning phase at the sample level have been reported in Experiment 2 of Chapter 2. In this part, only the results at the group and the individual levels are reported. However, for the sake of completeness, the results for the transfer phase at the sample level are presented in Appendix 5.A.

To investigate the first aim of this paper, i.e., whether participants who show a break during the learning phase also show a break during the transfer phase, we analysed the difference in solution times between +5 and +6 problems. To investigate the second aim of this paper, i.e., intra- and inter-problem differences, we calculated addend slopes without problems with the largest addend and compared addend slopes in L12 to the 3 transfer sessions. We will first describe the results from the learning phase and then those from the transfer phase.

Figure 5.1

Solution Times as a Function of Addend in Experiment 1



Note. Solution times as a function of addend during the learning phase for Sessions L1 (solid circles, solid line), L3 (open circles, long-dashed lines), L6 (triangles, dotted line), and L12 (squares, dashed line) of Experiment 1, for non-breakers (left panel) and breakers (right panel). Error bars represent standard errors.

Learning phase

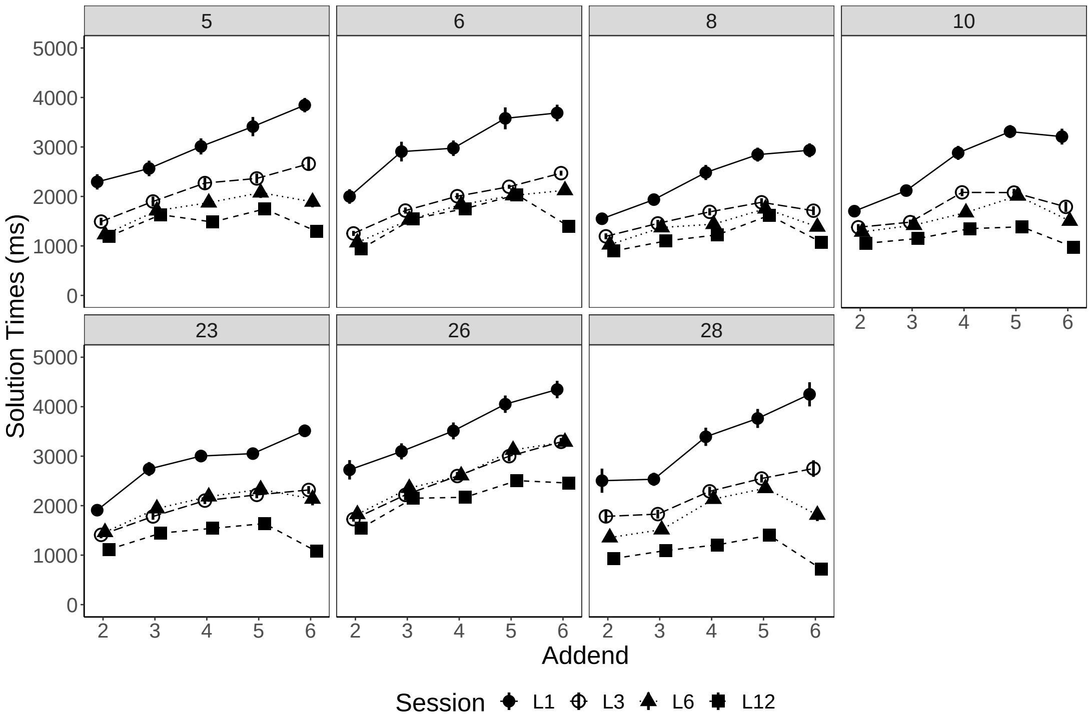
During the learning phase, 4 participants who never showed a break in solution times across learning sessions were classified as non-breakers and 7 participants who showed a break in solution times from one session until the end of learning as breakers (see Chapter 2). The remaining 10 participants did not show a consistent pattern regarding solution-time break and were categorised as "others". The performance of the breaker and non-breaker groups during the learning phase is presented in Figure 5.1.

We calculated the addend slopes for each participant and for each session, both including and excluding +6 problems. The mean addend slope for non-breakers were significantly different from 0 from L1 to L12, whether +6 problems were included (410 ms/addend, $p = .008$ in L1 and 262 ms/addend, $p = .004$ in L12) or excluded (418 ms/addend, $p = .02$ in L1 and 306 ms/addend, $p = .008$ in L12). For the breakers, addend slopes were significant in L1 (404 ms/addend, $p < .001$ with +6 problems and 447 ms/addend, $p < .001$ without +6 problems). However, whereas the addend slope in L12 was not significant when +6 problems were included (67 ms/addend, $p = .08$), it was significant when +6 problems were excluded (208 ms/addend, $p = .001$).

Although we proposed in Chapter 2 that the decrease in solution times for problems with the largest addend among the breakers was due to the memorisation of this problems, the right panel of Figure 5.1 shows that even in L12, solution times for +6 problems were still higher than for +2 problems, indicating that retrieval was not the only strategy used by the breakers to solve problems with the largest addend. To explore intra-individual differences within the breaker group, we plotted the individual performance in Figure 5.2. Five out of 7 participants, i.e., Participants 6, 8, 10, 23, and 28, presented a solution-time discontinuity at +5 problem. We called this the classical pattern, i.e., a solution-time discontinuity at problems with the second-largest addend. Three out of these 5 participants showing the classical pattern, i.e., Participants 10, 23, and 28, seemed to retrieve the answers to the problems with the largest addend, because solution times for these problems (i.e., +6) were lower than for +2 problems. Furthermore, when +6 problems were included, these 3 participants had non-significant addend slopes in L12 (6, 11, and −14 ms/addend for Participants 10, 23, and 28, respectively), giving the impression that the shift from counting to retrieval has occurred. However, when +6 problems were excluded, their addend slopes were still significant (121, 168, and 152 ms/addend for Participants 10, 23, and 28, respectively, ps $< .001$), indicating that the shift did not occur. Figure 5.2 also shows that Participant 5 presented another pattern that we called the M pattern.

Figure 5.2

Solution Times as a Function of Addend for Breakers in Experiment 1



Note. Solution times as a function of addend during the learning phase for Sessions 1 (solid circles, solid line), 3 (open circles, long-dashed lines), 6 (triangles, dotted line), and 12 (squares, dashed line) for the 7 breakers in Experiment 1. Numbers displayed above the panels represent participant numbers. Error bars represent standard errors.

In this case, solution times for +6 problems were lower than for +5 problems and solution times for +4 problems were lower than for +3 and +5 problems.

Transfer phase

To investigate whether participants classified as breakers during the learning phase also showed a discontinuity in solution times during the transfer phase, we calculated the difference in solution times between +6 and +5 problems in the 3 transfer sessions. The number of participants in each group during both learning and transfer phase is presented in Table 5.1. A χ^2-test of independ-

Table 5.1

Contingency Table of Participants in Experiment 1

		Transfer			
		Breakers	Non-Breakers	Others	Total
Learning	Breakers	5	2	0	7
	Non-Breakers	0	4	0	4
	Others	2	5	3	10
	Total	7	11	3	21

Note. The classification of participants into three groups according to their solution-time discontinuity during learning and transfer phases in Experiment 1.

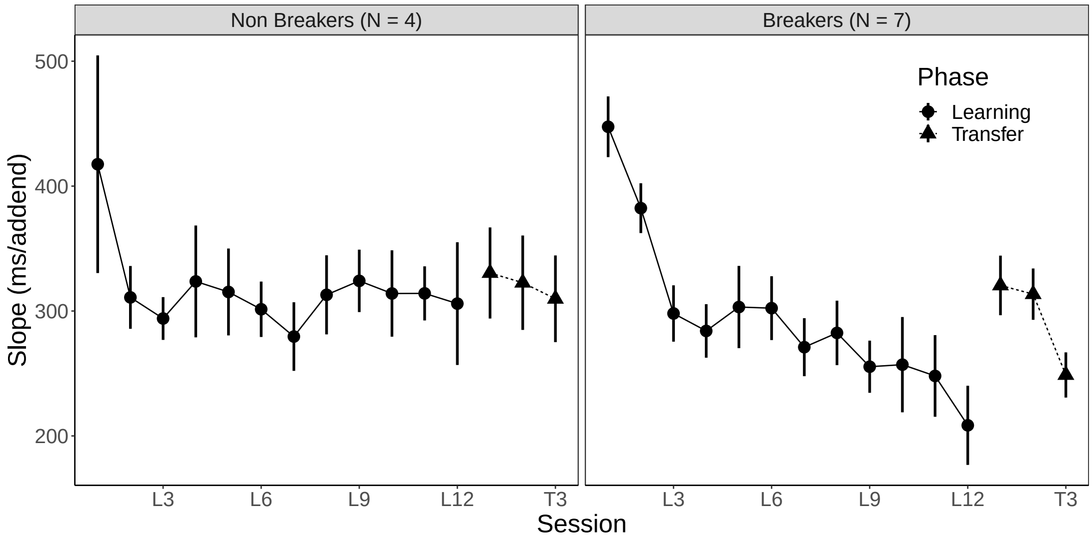
ence revealed an association between showing a break during learning and during transfer sessions ($\chi^2(4, N = 21) = 10.71$, $p = .03$), signifying that participants who showed a break during the learning phase were more likely to show a break during the transfer phase, and vice versa. Among 7 participants who showed a break during the transfer phase, 3 showed it for the first time in T1, 3 other participants in T2, and the last participant in T3.

Transfer from learnt to new items

To test our hypothesis about transfer from learnt to new items in the two groups, we ran a series of paired-sample t-tests for each learning group, comparing addend slope in L12 to those in T1, T2, and T3. As explained in Section 5.1, to compare performance between L12 to that in the transfer sessions, we calculated the participants' addend slopes without including +6 problems. Figure 5.3 shows that for non-breakers, there was no difference in addend slope

Figure 5.3

Addend Slopes as a Function of Session for Breakers and Non-breakers in Experiment 1



Note. Addend slopes of solution times as a function of addend during the learning (circles, solid line) and transfer (triangles, dotted line) phases for true equations in Experiment 1, without including +6 problems. The addend slopes are presented separately for non-breakers (left panel) and breakers (right panel). Error bars represent standard errors.

between Sessions L12 (306 ms/addend) and the 3 transfer sessions (+25, +17, and +4 ms/addend for Sessions T1, T2, and T3, respectively, *p*s > .05). For breakers, on the other hand, the difference in addend slope was significant between Sessions L12 (208 ms/addend) and T1 (+112 ms/addend, $t(6) = 4.22$, $p = .006$) and between Sessions L12 and T2 (+105 ms/addend, $t(6) = 3.50$, $p = .01$), but not between Sessions L12 and T3 (+40 ms/addend, $t(6) = 1.46$, $p = .20$).

Discussion

In this experiment, we first aimed at studying whether there was an association between showing solution-time discontinuity during the learning phase and the transfer phase of an alphabet-arithmetic task. By analysing the difference in solution times between +5 and +6 problems, we found a tendency for breakers, i.e., participants who showed a discontinuity in solution times during the learning phase, to show a break during the transfer phase as well. This echoes the results of Haider and Frensch (2002), who found that participants showing a shift to a more-efficient strategy during learning phase also showed a shift during transfer phase. Because this shift was observed in some participants and not in the others, Haider and Frensch suggested that the shift was not the result of a bottom-up mechanisms brought about by the learnt material, but resulted instead from some top-down mechanism. In other words, the participants intentionally decided to change strategy. Whereas in Haider and Frensch's study it was the decision to shift to a more-efficient strategy, in our work, it is probable that the breakers decided to memorise the answer to +6 problems. Furthermore, realising that memorising +6 problems led to shorter solution times, the breakers might have decided to use this same strategy during transfer.

The second aim of our experiment was to study individual differences in the evolution of counting strategy during learning. We tested this by comparing the performance during learning and transfer phase, operationalised by addend slopes without problems with the largest addend. The results for the non-breaker group revealed a significant addend slope throughout the learning sessions, even when problems with the largest addend were included, implying that counting procedure was the privileged strategy in this group of 4 participants. Furthermore, this strategy must have been adaptive because, immediately in the first transfer session, applying this strategy to new items led to the same performance as in the last learning session. Interestingly, as shown on the left panel of Figure 5.3, addend slopes for non-breakers were relatively stable at around 300 ms/addend starting from the second learning session until the end of the learning phase. It was also at this rate that counting was used by the non-breakers during the transfer phase. Although Logan and Klapp (1991) admitted that the use of counting at the end of practice is possible, they claimed that it concerns only a minority of trials. However, here we show that for some participants, counting remained the dominant strategy after practice. The behaviour of this group of participants confirms the conclusion of Chapters 2 and 4 that the possibility of counting after extensive practice of alphabet-arithmetic cannot be discarded.

The results for the breaker group, on the other hand, were more complicated. When problems with the largest addend were excluded from the analysis, the addend slope at the last session of learning was significant (i.e., 208 ms/addend). Although the addend slope was lower for this group than for the non-breakers (i.e., 306 ms/addend), the significant addend slopes indicate the use of counting strategy for problems with smaller addends in both groups. Nevertheless, when problems with the largest addend were excluded, we observed a sharp increase in addend slopes during the first 2 transfer sessions compared to the last learning session for the breakers. This is in contrast to the direct transfer observed in the non-breaker group. Therefore, we can deduce that the significant addend slopes when problems with the largest addend were excluded observed during training for these two groups did not originate from the same mechanisms. More precisely, it is probable that non-breakers used counting to solve problems with smaller addends whereas breakers used both counting and retrieval for these same problems. In other words, we observed intra-problem differences in breakers. Indeed, shallower addend slope for breakers than for non-breakers in the last learning session, i.e., 208 and 306 ms/addend, respectively, may support this interpretation. Intra-problem differences in breakers could be exemplified, for example, by Participant 5 who showed the M pattern might have used retrieval for +4 problems and counting for +3 and +5 problems.

Our interpretation about the use of different strategies for problems with smaller addends is not necessarily at odd with the conclusion of Chapters 2 and 4 that problems in this range of addends were solved by counting. In fact, it is possible that the significant addend slope in the previous chapters represented an averaging over different strategies. Indeed, for their alphabet-arithmetic task, White et al. (2007) proposed that while addend slopes of about 400 and 0 ms/addend could indicate the use of counting and memory retrieval strategies, respectively, an addend slope of about 100 ms/addend might reflect either an acceleration of counting procedure or the averaged addend slope resulted from the use of mixed strategies. Thus, it is possible that the addend slope of 208 ms/addend observed in breakers in the last learning session was due to an acceleration of counting in some participants and the use of mixed strategies by others. In fact, in the domain of mental addition, Siegler (1987) found that the minimum operand was the best predictor of solution times even though the frequency of reported use of counting (i.e., in 36% of trials) was similar to retrieval (i.e., in 35% of trials).

Although the use of retrieval by some participants at the end of our experiment may, in itself, support the instance theory of automatization (Logan, 1988), some top-down mechanisms, that are manifested by intra-individual differences, are not predicted by the theory postulating that the only factor determining retrieval is the number of exposures to the stimulus. Indeed, in this experiment, all true equations are presented in an equal number of times but participants seemed to have decided to memorise one equation and not the others.

Nevertheless, the results in the present experiment were collected from a verification task, which could be considered as not ecological in the context of a task that was conceived to mimic the way children learn addition, because addition is naturally learnt through production tasks. Furthermore, solution times in verification tasks are not necessarily representative of the times it takes to solve a problem (Baroody, 1984). Besides, assuming that retrieval is used, a correctly-solved trial in a verification task does not necessary represent a correct retrieval (Campbell, 1987b). Having said that, we showed in Chapter 4 that at least for the alphabet-arithmetic paradigm, results from a verification task are replicable in a production task, i.e., a decrease in solution times for problems with the largest addend and significant addend slope at the end of learning when problems with the largest addend were not included. However, our study in Chapter 4 did not include individual differences in performance. Therefore, in Experiment 2, we will investigate whether the results obtained in a verification task in Experiment 1 are also obtained in a production task. To this aim, we will make use of the data from the experiment reported in Chapter 4, that used the same material as in Experiment 1 and also consisted of 12 learning sessions followed by 3 transfer sessions.

5.3 Experiment 2

Method

Similar to Experiment 1, the complete experiment comprised 12 sessions of learning, corresponding to 12 working days, followed by 3 sessions of transfer, corresponding to 3 working days. The method for the learning sessions has been described in Section 4.2 (page 127). The material and procedure for both learning and transfer phases were the same as in Experiment 1 described earlier, except that we used a production task, i.e., participants had to give the answer to problems such as D + 5. Considering the difference in tasks, this experiment differed from Experiment 1 in two ways. Firstly, the 40 problems were presented 12 times in a session. The second difference concerned the home training, which consisted in writing as quickly as possible the answer to 160 problems presented on paper.

Results

Following the data diagnostic described above, we excluded the data of 4 participants because the level of accuracy was too low (i.e., lower than 85% for at least 2 consecutive sessions) or the level of recording errors too high (i.e., more than 20% for at least 3 consecutive sessions). Furthermore, considering that participants were expected to use counting strategy in the beginning of learning, we excluded the data of 2 participants who already showed non-significant addend slopes at the beginning of learning. Moreover, we only analysed correctly-solved problems with solution times that were not categorised as extreme values, defined as trials with solution times shorter than 250 ms. On the whole, we discarded 7.21% of the data from 17 participants.

The results for the learning phase at the sample level have been reported in Chapter 4. In this part, only the results at the group and the individual levels are reported. However, for the sake of completeness, the results for the transfer phase at the sample level are presented in Appendix 5.A. Similar to Experiment 1, we analysed the difference in solution times between +5 and +6 problems to investigate whether participants who show a break during the learning phase also show a break during the transfer phase. To investigate intra- and inter-problem differences, we calculated addend slopes without problems with the largest addend and compared such addend slopes in L12 and the 3 transfer sessions. We will first describe the results from the learning phase and then those from the transfer phase.

Learning phase

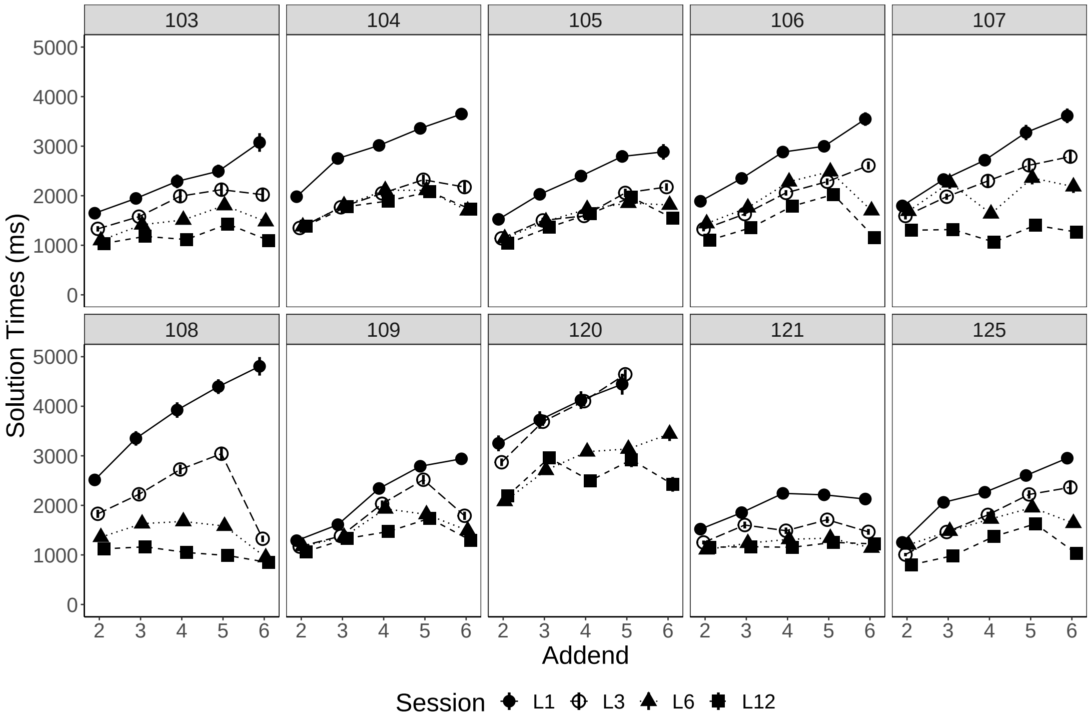
We classified the participants based on whether solution times for +5 problems were higher or lower than for +6 problems in each session (see Chapter 4). Out of 17 participants, 2 did not show a break throughout the learning sessions and were classified as non-breakers, whereas 10 showed a systematic break from one session (from as early as in Session 1 to as late as in Session 9) until the end of the learning phase and were classified as breakers. Among these 10 breakers, half of them exhibited the solution-time discontinuity for the first time in the first 3 sessions. The 5 remaining participants out of 17 showed non-consistent pattern of solution-time discontinuity and were therefore classified as "others". They showed a discontinuity in solution times during several sessions but this observation was disrupted for at least one session.

To investigate intra-individual differences among the breakers, we present their performance in Figure 5.4. Although we classified the 10 participants as one group, Figure 5.4 reveals that they showed heterogeneous behavior that was richer than in Experiment 1. Similar to Experiment 1, we identified the classical pattern in Participants 104, 105, 106, 108, and 125, and the M pattern in Participants 103, 107, 120, and 121. An exception to the two patterns was found for Participant 109 who exhibited a break at +4 in Session 6. Among those who showed the M pattern, 2 participants exhibited this M pattern early in the learning process (in L3 and L6 for Participants 121 and 117, respectively). For these 2 same participants, the M pattern evolved into a flat pattern, i.e., there was no variation in solution times as a function of addend, including the largest one. A flat pattern at the end of learning was also observed for Participant 108, who showed a classical pattern in earlier sessions.

Considering the flat pattern observed visually in several breakers, we examined the addend slopes without the largest addend for all participants. Non-significant addend slopes at the end of learning were found for Participants 107 (3 ms/addend), 112 (−16 ms/addend), 119 (−4 ms/addend), and 121 (28 ms/addend). Non-significant addend slopes when problems with the largest addend were not included could be taken as a strong indication that all problems were solved using retrieval. Additionally, addend slope for Participant 108 was

Figure 5.4

Solution Times as a Function of Addend for Breakers in Experiment 2



Note. Solution times as a function of addend during the learning phase for Sessions L1 (solid circles, solid line), L3 (open circles, long-dashed lines), L6 (triangles, dotted line), and L12 (squares, dashed line) for the 10 breakers in Experiment 2. Error bars represent standard errors. Due to the large values, solution times for +6 problems for participant 120 in Sessions L1 and L3 were not presented.

significant but negative (−52 ms/addend, p = .004). Although by convention significant slopes are taken to indicate the use of counting, this is only valid for positive slopes. For negative slopes, on the other hand, significant slopes would imply fewer counting steps for larger addend and therefore cannot indicate the use of counting. Instead, they could be considered as a sign of retrieval. Therefore, it is probable that these 5 participants, i.e., Participants 107, 108, 112, 119, and 121, showed an automatised performance as defined by the instance theory of automatization (Logan, 1988) and we classified them as retrievers. Thus, for this experiment, instead of categorising the participants into breakers, non-breakers,

Table 5.2

Contingency Table of Participants in Experiment 2

		Transfer				
		Non-Breakers	Breakers	Retrievers	Others	Total
Learning	Non-Breakers	2	0	0	0	2
	Breakers	0	4	0	3	7
	Retrievers	0	4	1	0	5
	Others	0	3	0	0	3
	Total	2	11	1	3	17

Note. The classification of participants into four groups according to their solution-time discontinuity during learning and transfer phases in Experiment 2.

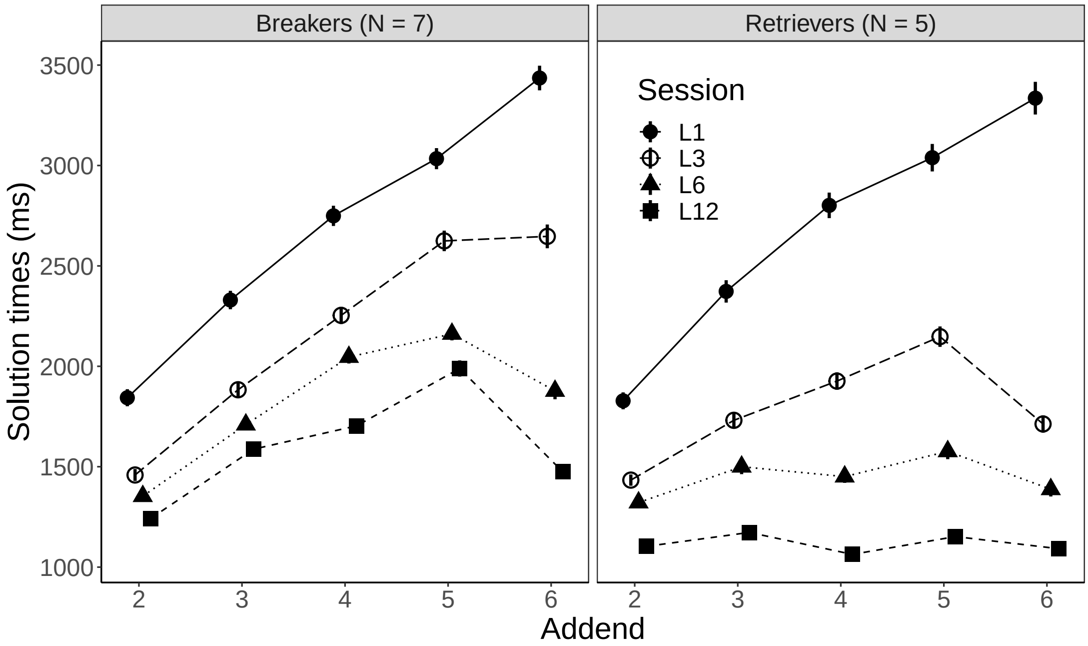
and others like in Experiment 1, we consider it more appropriate to classify them into retrievers, breakers, non-breakers, and others. The number of participants in each group is presented in Table 5.2.[1] The performance of retrievers and breakers during the learning phase is presented in Figure 5.5, that shows a tendency for the retrievers to have the classical pattern followed by the M pattern.

Having established the classification of participants, we calculated the addend slopes for each participant and for each session, with and without +6 problems. Considering that the non-breaker group only contained 2 participants, we will not discuss this group further and will concentrate our results on the retrievers and breakers. For the breaker group, the addend slopes were significant from L1 (394 and 412 ms/addend, with and without +6 problems, re-

[1] In Experiment 1, we did not find a participant who showed the sign of being a retriever. To investigate whether this phenomenon is proper to a production task, we took the data from Experiment 1 of Chapter 2 that used a verification task and calculated the addend slopes without problems with the largest addend, i.e., +5, and found 5 retrievers out of 19 participants. Their performance is presented in Figure 5.B.1 of Appendix 5.B.

Figure 5.5

Solution Times as a Function of Addend in Experiment 2



Note. Solution times as a function of addend during the learning phase for Sessions L1 (solid circles, solid line), L3 (open circles, long-dashed lines), L6 (triangles, dotted line), and L12 (squares, dashed line) of Experiment 2, for breakers (left panel) and retrievers (right panel). Error bars represent standard errors.

spectively, ps $< .001$) to L12 (87 ms/addend, $p = .002$ with +6 problems and 233 ms/addend, $p < .001$ without +6 problems). The result for L12 differed from that in the verification task, where addend slope with +6 problems was not significant (i.e., 67 ms/addend, $p = .08$). The addend slopes for retriever group, on the other hand, were significant in L1 (373 ms/addend, $p = .005$ with +6 problems and 412 ms/addend, $p = .003$ without +6 problems) but not in L12 (−4 and 5 ms/addend, with and without +6 problems, respectively). On average, the addend slope of the retrievers became non-significant starting from L3, i.e., 103 ms/addend, when +6 problems were included and from L7, i.e., 20 ms/addend, when they were excluded.

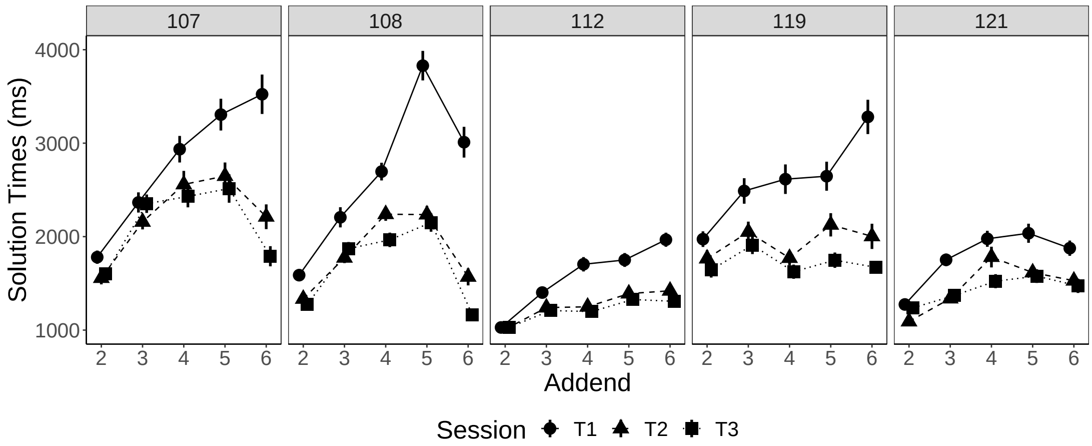
Transfer phase

Similar to the learning phase, we investigated participants' behaviour pattern with respect to whether they showed a break in solution times or not. Firstly, 2 participants never showed a break across transfer sessions and were categorised as transfer non-breakers. They were the same 2 participants who were categorised as non-breakers (see Table 5.2). Secondly, 12 participants showed a systematic break from one session until the end of transfer phase. Thirdly, the 3 remaining participants showed non-consistent pattern. They showed a break in solution times in T1 or T2 but then the break disappeared for at least one session.

Among the 12 participants who showed a solution-time break during transfer, there were the 5 participants who were categorised as retrievers during the learning phase. Their performance during the transfer phase is presented in Figure 5.6. Two observations are worth mentioning. Firstly, for Participants 107 and 108, the performance for +6 problems seemed to be based on retrieval because in T3, solution times for these problems were similar to solution times for +2 problems ($t(177) = 1.49$, $p = .14$ for Participant 107 and $t(183) = -1.68$, $p = .10$ for Participant 108). However, the performance for other problems was not based on retrieval, as indicated by the steep addend slope when +6 problems were not included (i.e., 283 and 273 ms/addend for 107 and 108, respectively, $ps < .001$). Secondly, among these 5 participants, only Participant 119 was also retriever during the transfer phase, as attested by the non-significant addend slope when +6 problems were not included, i.e., 82 and 4 ms/addend in T2 and T3, respectively. Thus, among the 12 participants who showed a break during the transfer phase, we classified 1 as transfer retriever (see Table 5.2). A χ^2-test of independence showed a significant association between categorization during learning and transfer phases ($\chi^2(9, N = 17) = 23.80$, $p = .005$), indicating that

Figure 5.6

Solution Times as a Function of Addend for Retrievers in Experiment 2



Note. Solution times as a function of addend during the transfer phase for the 5 retrievers in Experiment 2. Error bars represent standard errors.

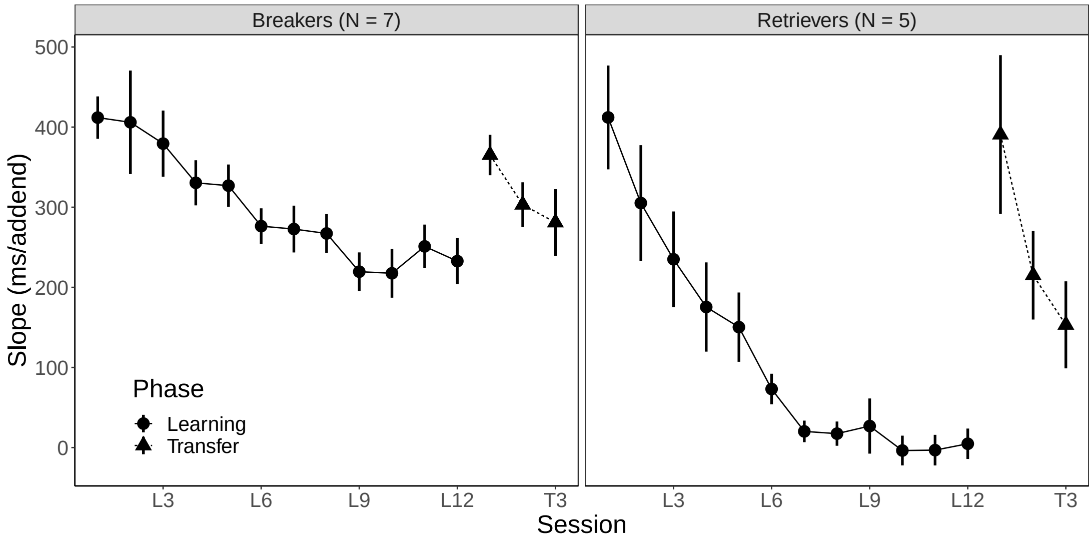
participants' classification during the learning corresponded to their classification during the transfer phase.

Transfer from learnt to new items

To investigate transfer performance, we ran a series of paired-sample t-tests on addend slopes after excluding problems with the largest addend for breaker and retriever groups separately. Figure 5.7 shows that for breakers, there was a significant difference in addend slope between L12 (233 ms/addend) and T1 (+132 ms/addend, $t(6) = 2.87$, $p = .02$) and between L12 and T2 (+70 ms/addend, $t(6) = 2.65$, $p = .04$), but not between L12 and T3 (+48 ms/addend, $t(6) = 1.32$, $p = .24$). For retrievers, similarly, the difference was significant between L12 (5 ms/addend) and T1 (+386 ms/addend, $t(4) = 3.37$, $p = .03$) and between L12 and T2 (+210 ms/addend, $t(4) = 3.10$, $p = .04$), but not between L12 and T3 (+148 ms/addend, $t(4) = 2.15$, $p = .10$). However, considering that Participant 119 was also retriever during transfer session, the observed transfer in T3 among retrievers might merely due to this

Figure 5.7

Addend Slope of Solution Times as a Function of Session in Experiment 2



Note. Addend slope of solution times as a function of addend during the learning (circles, solid lines) and transfer (triangles, dotted lines) phase for breakers (left panel) and retrievers (right panel) in Experiment 2, without including +6 problems. Error bars represent standard errors.

participant. Therefore, we redid the analysis without this participant and found that the difference in addend slope was significant between L12 and T1 (+444 ms/addend, $t(3) = 3.49$, $p = .04$), T2 (+257 ms/addend, $t(3) = 4.10$, $p = .03$), and T3 (+199 ms/addend, $t(3) = 3.31$, $p = .045$).

Discussion

This second experiment aims at investigating whether the results found in a verification task reported in Experiment 1 were also found in a production task, wherein the same material and procedure as in Experiment 1 were used. Indeed, the results of Experiment 2 replicated those of Experiment 1, because it also revealed the tendency that participants who showed a solution-time discontinuity during learning obtained it also during the transfer phase. Furthermore, concerning the breakers, we found comparable results between the two tasks, i.e., the classical and M patterns of behaviour were found within the group during the learning phase, addend slope without problems with the largest addend at the end of the learning phase was significant, and transfer was obtained in the third transfer phase. Moreover, still concerning the breakers, the magnitude of addend slope at the end of learning was similar in verification (67 and 208 ms/addend, with and without +6 problems, respectively) as in production (87 and 233 ms/addend, with and without +6 problems, respectively) tasks.

The major difference between the two experiments reported in this paper concerns the 5 participants in Experiment 2 who exhibited non-significant addend slopes at the end of learning, even when problems with the largest addend were excluded. However, this difference did not stem from the specificity of the production task, because retrievers were also found in Experiment 1 of Chapter 2 that was also based on a verification task (see Appendix 5.B). Contrary to the breakers who most probably memorised problems with the largest addend and used multiple strategies for the other problems, the retrievers most probably memorised not only problems with the largest addend but also other problems. Among the 5 retrievers, only 1 had non-significant addend slope at the end of transfer. However, Figure 5.6 shows that this participant did not use retrieval for all problems, because solution times for +3 problems were higher than for other problems.

In Chapter 4 we found that addend slopes in a production task decreased twice as fast as in a verification task, such that addend slope in the 12th session of the verification task was similar to addend slope in the 6th session of the production task, either problems with the largest addend were included or excluded. Considering the comparable performance for the breaker groups in the two tasks, the pronounced lower addend slope found in the production task could not result from the particularity of the task. Instead, it is very likely that the low added slopes in Chapter 4 were due to the 5 retrievers, which is obvious from Figure 5.7.

5.4 General Discussion

This paper aims at understanding individual differences in learning by means of two alphabet-arithmetic training experiments with different tasks, i.e. a verification task in Experiment 1 and a production task in Experiment 2. Alphabet-arithmetic paradigm was created in the framework of the instance theory of automatization (Logan, 1988; Logan & Klapp, 1991), according to which in the course of learning, algorithm-based performance will be replaced by retrieval-based performance. The theory also postulates that the probability of retrieval depends on the number of repetitions, whether the memorisation results from an automatic encoding of the counting process or from the use of a mnemonic technique.

Interestingly, in both experiments, the comparison in performance between the learning and transfer sessions revealed the consistency of the used strategy, i.e., participants who showed a solution-time discontinuity during learning also showed it during transfer. As argued by Haider and Frensch (2002), this implies that, as far as the breakers and retrievers are concerned, the choice of strategy was not determined by the studied material but was the result of participants' intentional decision.

Through the 2 experiments, we identified 3 groups of participants based on whether or not a discontinuity in solution times was observed. In other words, whether or not solution times for problems with the largest addend were lower than for problems with the second-largest addend. There is a strong indication that the categorisation of participants into these 3 group is related to different strategies adopted by different groups.

The first group contains the non-breakers. Their addend slopes remained significant at the end of training, even when problems with the largest addend were included, indicating the use of counting procedure for all addends from the beginning to the end of learning. This is confirmed during the transfer phase because transfer of procedural knowledge is observed immediately in the first session in which new items are encountered. The behaviour of participants in this group corresponds to the prediction of the automated counting procedure theory (Bagnoud, Dewi, Castel et al., 2021; Barrouillet & Thevenot, 2013; Fayol & Thevenot, 2012; Thevenot & Barrouillet, 2020; Uittenhove et al., 2016), according to which slow counting strategy is accelerated in the course of learning. However, the observed acceleration seemed to stagnate at 300 ms/addend after the second learning phase (see Figure 5.3), which is higher than the rate of silent counting, i.e., 125 ms/unit increment (Landauer, 1962), or the slope for small additions in adults, i.e., 47 ms/unit sum (Uittenhove et al., 2016). Thus, it is unlikely that our non-breakers have reached an automated and unconscious counting procedure advocated by the theory. On the other hand, it remains an open question whether more-extended practice would have led to an automated counting procedure or to the use of another strategy such as retrieval.

At the opposite end of the non-breakers are the retrievers. Their addend slopes at the end of training was not significant, even when problems with the largest addend were excluded, indicating the use of retrieval for all addends. Although in this chapter retrievers were found in our production task but not in our verification task, this difference is more due to the effect of cohort than the effect of task. This is because retrievers were also found in Experiment 1 of Chapter 2, wherein a verification task was used (see Appendix **??**). Despite the fact that the performance of participants in this group seems to provide support for the shift from counting to retrieval purported by the instance theory of automatization (Logan, 1988), it is intriguing to notice that all retrievers started by memorizing problems with the largest addend. Figure 5.B.1, that was based on the data of Experiment 1 of Chapter 2, shows that when the largest addend corresponds to +5, the classical pattern, i.e., a discontinuity in solution times at second-largest addend, was followed by the memorisation of either +4 or +3 problems. The former is manifested by a discontinuity in solution times at +3 problems, i.e., Participants 14, 17, and 19, whereas the latter by a shorter solution times for +3 than for +2 and +4 problems, i.e., Participant 13. In Experiments 1 and 2 of the current chapter, wherein the largest addend corresponds to +6, the classical pattern was followed by the M pattern, i.e., lower solution times for +4 than for +3 and +5 problems. Furthermore, although we categorised the retrievers according to their non-significant addend slopes when problems with the largest addend were excluded, not all retrievers show a completely flat function. Visual inspection of Figure 5.B.1, for example, reveals that a flat function was only observed in 1 out of 7 retrievers, i.e., Participant 21. These two observations, i.e., that different addends are memorised at different epochs during learning and that non-significant addend slopes do not always correspond to a flat function, put the instance theory of automatization into question because it means that the number of repetitions alone is not enough to guarantee retrieval. Instead, some top-down mechanism must also play a role. In other words, participants seem to have a conscious decision as to which problems to memorise.

The third group of our participants contains the breakers. They showed the classical pattern that did not develop into another pattern. When problems with the largest addend were excluded, their addend slopes were significant, also at the end of learning, indicating that the use of counting to solve problems with the smaller addends cannot be discarded. In other words, retrieval was probably used by participants in this group to solve problems with the largest addend whereas counting was used for other problems. However, the fact that transfer, that was based only on problems with smaller addends, was observed in the first session for non-breakers but not for the breakers indicates, on the one hand, that retrieval was also used by breakers to solve problems with smaller addends and, on the other, that the significant addend slopes in breakers was the result of averaging over different strategies. In future research, this explanation can be tested by means of a dual-task experiment (e.g., Beilock & Carr, 2001; Kramer et al., 1995). Compared to a single-task experiment, problems that rely on retrieval in a dual-task experiment should suffer less interference than problems that rely on counting.

Following Haider and Frensch (2002)'s suggestion, the fact that transfer was not observed in the first transfer session for breakers and retrievers might be due to the effect of surprise. However, in our Experiment 1, transfer was observed in the first session for participants in non-breaker group. This means that surprise alone cannot explain the lack of transfer in the first transfer session for breakers and retrievers. For the retrievers, because retrieval seems to be the dominant strategy during learning, it is not surprising that transfer was not obtained in the first transfer session. For the breakers, on the other hand, although there was an indication that counting was used during learning, the use of mixed strategies seems to be sufficient to prevent the transfer of procedural knowledge. It seems therefore that only when counting is used almost exclusively could we have an immediate transfer. This echoes the finding of VanLehn (1996) that the use of algorithm during learning should transfer to new items whereas the use of retrieval would not. Nevertheless, our results concerning different performance in different groups during the transfer phase reveal the subtlety of VanLehn's statement and open the way for another interpretation of what is transferred during the transfer phase. In fact, whereas it is clear that the observed transfer in non-breakers corresponds to the transfer of counting procedure, the picture is less obvious for breakers, particularly concerning problems with the largest addend. On the one hand, it is evident that the memorised answers to +6 problems in one set cannot be transferred to the other set. On the other, we have shown that there was a tendency for breakers to also memorise +6 problems during transfer. Thus, although the memorised facts are not transferrable, the used strategy is, i.e., memorising problems with the largest addend.

To conclude, our results have a major implication for the instance theory of automatization. The theory postulates that the probability of memory retrieval being used depends only on the number of encounters with the learnt material. Our results suggest that the number of encounters alone is obviously not enough to guarantee retrieval, because in our experiments all problems were presented with the same frequency. Instead, there seems to be strong inter- and intra-individual differences in how alphabet-arithmetic task is learnt and performed. Within intra-individual differences, we also observed intra-problem and inter-problem differences. In future research, it will be interesting to investigate the underlying cognitive factors that may determine the observed individual differences. In an earlier work, for instance, Brigman and Cherry (2002) found that working memory, but not processing speed, was correlated with solution times at the end of training.

Although the current paper was based on training experiment in laboratory, there are reasons to think that similar results could be found in more-ecological educational settings. More precisely, pupils might find one strategy easier to use than the others. In the classrooms, it would probably best to teach children several strategies to solve the same problems and to give them the liberty to choose the one that is more adaptable for them.

Appendix 5.A

Performance of All Participants during Transfer Sessions

This appendix reports the performance of all participants during the transfer phase. Participants who worked with Set 1 (i.e., A to H) during learning and Set 2 during transfer are classified as Group 1 whereas participants who did the other way around as Group 2. Following the labelling in the main text, the L letter referred to the learning sessions, i.e., L1 to L12, and the T letters the transfer sessions, i.e., T1 to T3. In each experiment, we first presented the performance during the 3 transfer sessions in terms of accuracy, solution times, and addend slopes. The addend slopes were calculated both including and excluding problems with the largest addend. Then, to determine whether transfer occurred or not, we excluded problems with the largest addend and calculated the addend slopes without these problems. Lastly, we compared the addend slope in the last learning session, i.e., L12 with the addend slopes in the 3 transfer sessions.

Experiment 1

Performance during the Transfer Phase

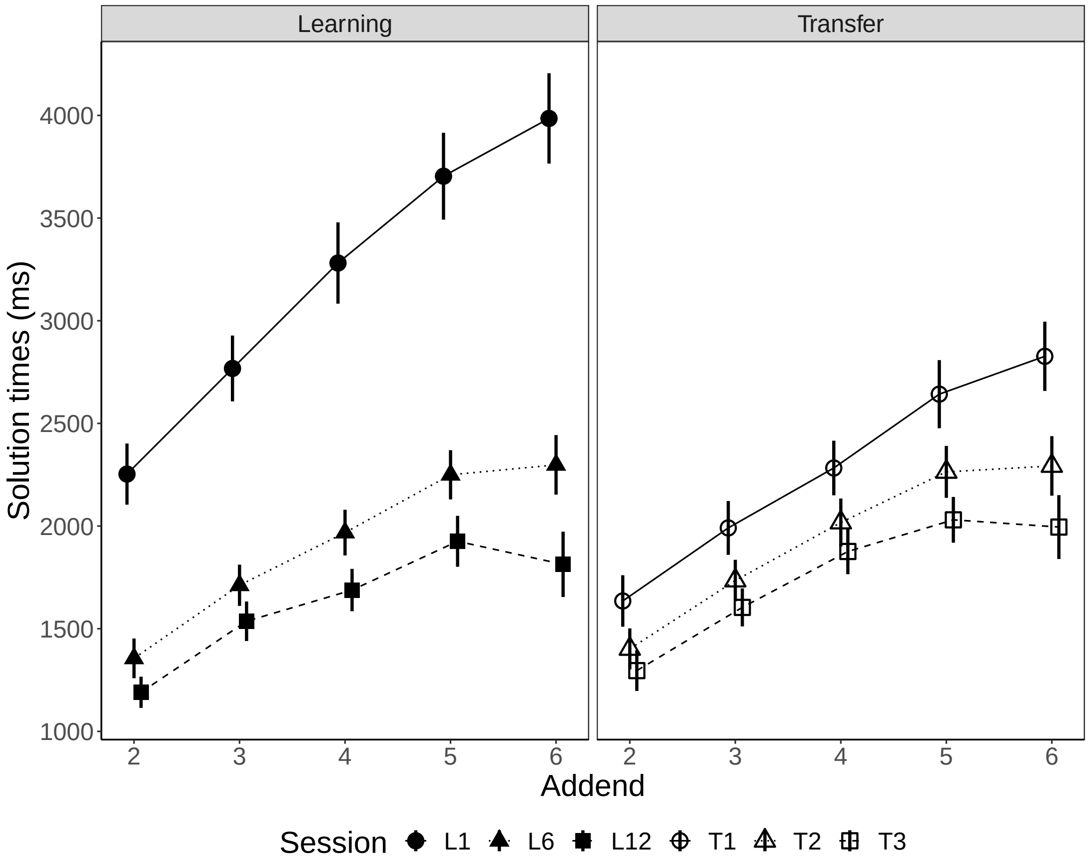
We first carried out a 3 (Session: T1, T2, and T3) × 5 (Addend: 2 to 6) × 2 (Group: 1 or 2) repeated-measures, mixed-design ANOVA on accuracy with Group as the between measure. The main effect of Group ($F(1, 19) = 1.00$, $p = .33$) was not revealed, and neither was the interaction between Group and Session ($F(2, 38) = 1.22$, $p = .31$), or the interaction between Group and Addend ($F(4, 76) < 1$). We therefore collapsed the two letter sets to perform a 3 (Session: T1, T2, and T3) × 5 (Addend: 2 to 6) repeated-measures ANOVA. The effect of Session was found ($F(2, 40) = 4.10$, $\eta_p^2 = .17$, $p = .02$), with the accuracy increased from 95% in Session T1 to 96% in Session T3. The effect of Addend was also significant ($F(4, 80) = 8.10$, $\eta_p^2 = .29$, $p < .001$), with the accuracy decreased from 97% for +2 problems to 94% for +6 problems. The two variables did not interact with each other ($F(8, 160) < 1$).

To analyse solution times, we excluded trials with solution times that were either too short (i.e., shorter than 300 ms) or too long (i.e., longer than the mean plus 3 standard deviations for each participant and each session). This has led us to discard 1.3% of the data. A 3 (Session: T1, T2, and T3) × 5 (Addend: 2 to 6) × 2 (Group: 1 or 2) repeated-measures, mixed-design ANOVA on solution times with Group as the between measure was then undertaken. Similar to the accuracy, we did not find the main effect of Group ($F(1, 19) = 1.17$, $p = .29$), the interaction between Group and Session ($F(2, 38) < 1$), or the interaction between Group and Addend ($F(4, 76) < 1$).

We then collapsed the two letter sets and performed a 3 (Session: T1, T2, and T3) × 5 (Addend: 2 to 6) repeated-measures ANOVA. The main effects of Session ($F(2, 40) = 39.10$, $\eta_p^2 = .66$, $p < .001$) was found. Solution times decreased from 2276 ms in T1 to 1942 ms in T2 ($t(20) = 7.61$, $p < .001$) and to 1760 ms in T3 ($t(20) = 3.67$, $p = .003$). The main effect of Addend ($F(4, 80) = 109.57$, $\eta_p^2 = .85$, $p < .001$) was also revealed, with solution times increased from 1444 ms for +2 problems to 2372 ms for +6 problems. Furthermore, we found an interaction between Session and Addend ($F(8, 160) = 10.45$, $\eta_p^2 = .34$, $p < .001$). A series of contrasts revealed significant linear and quadratic trends of the addend effect in all sessions. However, while the linear trend became weaker over sessions (from $t(20) = 15.69$, $p < .001$ in T1 to $t(20) = 7.58$, $p < .001$ in T3), the quadratic trend became more important (from $t(20) = -2.70$, $p = .04$ in T1 to $t(20) = -3.63$, $p = .005$ in T3). This quadratic trend was due to the break in solution times at +5 that started to emerge in Session T2 (see Figure 5.A.1).

Figure 5.A.1

Solution Times as a Function of Addend in Experiment 1



Note. Solution times as a function of addend during learning (left panel: solid circles, solid line for L1, solid triangles, dotted line for L6, and solid squares, dashed line for L12) and transfer (right panel: open circles, solid line for T1, open triangles, dotted line for T2, and open squares, dashed line for T3) sessions for true equations in Experiment 1. Error bars represent standard errors.

Because of this break, we calculated the addend slopes by including and excluding +6 problems and then ran a 3 (Session: T1, T2, and T3) × 2 (Group: 1 vs 2) repeated-measures, mixed-design ANOVA on addend with Group as the between measure for each data set separately. For data set including +6 problems, we did not find the effect of Group or the interaction between Session and Group (Fs < 1). Therefore, we collapsed the two letter sets and performed a one-way ANOVA. There was an effect of Session ($F(2, 40) = 18.50$, $\eta^2_{\text{p}} = .48$, $p < .001$). The addend slope in T1 (304 ms/addend) was higher than in T2 (232 ms/added, $t(20) = 3.63$, $p = .003$), which was higher than in T3 (183 ms/addend, $t(20) = 4.44$, $p < .001$).

When +6 problems were not included, we did not find the effect of Group nor the interaction between Session and Group neither (Fs < 1). Again, we collapsed the two letter sets and performed a one-way ANOVA. The effect of Session was significant ($F(2, 40) = 12.71$, $\eta^2_{\text{p}} = .39$, $p < .001$). Addend slope in T1 (331 ms/addend) was significantly higher than in T2 (287 ms/addend, $t(20) = 2.86$, $p = .02$), which was, in turn, significantly higher than in T3 (248 ms/addend, $t(20) = 2.99$, $p = .01$). Furthermore, a series of paired-sample t-tests revealed that addend slopes without +6 problems were significantly higher than with +6 problems ($t(20) = 2.68$, $p = .01$ in T1, $t(20) = 3.03$, $p = .007$ in T2, and $t(20) = 3.79$, $p = .001$ in T3).

Transfer from Learnt to New Items

To test whether transfer occurred, we compared addend slope without +6 problems in L12 to those in the 3 transfer sessions. We first tested whether transfer depends on the letter set by running a 4 (Session: L12, T1, T2, T3) × 2 (Group: 1 vs 2) mixed-design repeated-measures ANOVA on the addend slope with Group as the between variable. The effect of Group and the interaction between Group and Session were not significant (Fs < 1) and therefore we collapsed the two letter sets and carried out a one-way ANOVA. The results revealed a significant effect of Session ($F(3, 60) = 10.55$, $\eta_p^2 = .35$, $p < .001$).

Considering that we are interested in the difference in addend slopes between L12 and the 3 transfer sessions, we conducted a series of paired-sample t-tests with L12 as the reference level. Addend slope in T1 (331 ms/addend) was significantly higher than in Session L12 (236 ms/addend), i.e., +95 ms/addend ($t(20) = 4.30$, $p < .001$). Addend slope in T2 (287 ms/addend) was also higher than in L12 (+51 ms/addend, $t(20) = 2.60$, $p = .03$) but the difference in addend slopes between T3 (248 ms/addend) and L12 was not significant (+12 ms/addend, $t(20) = 0.60$, $p = .56$). Figure 5.A.2 shows that the addend slope in T1 corresponded to the addend slope in L3, that the addend slope in T2 corresponded to an addend slope between L6 and L7, and that the addend slope in T3 corresponded to an addend slope between L11 and L12, i.e., the two last training sessions.

Experiment 2

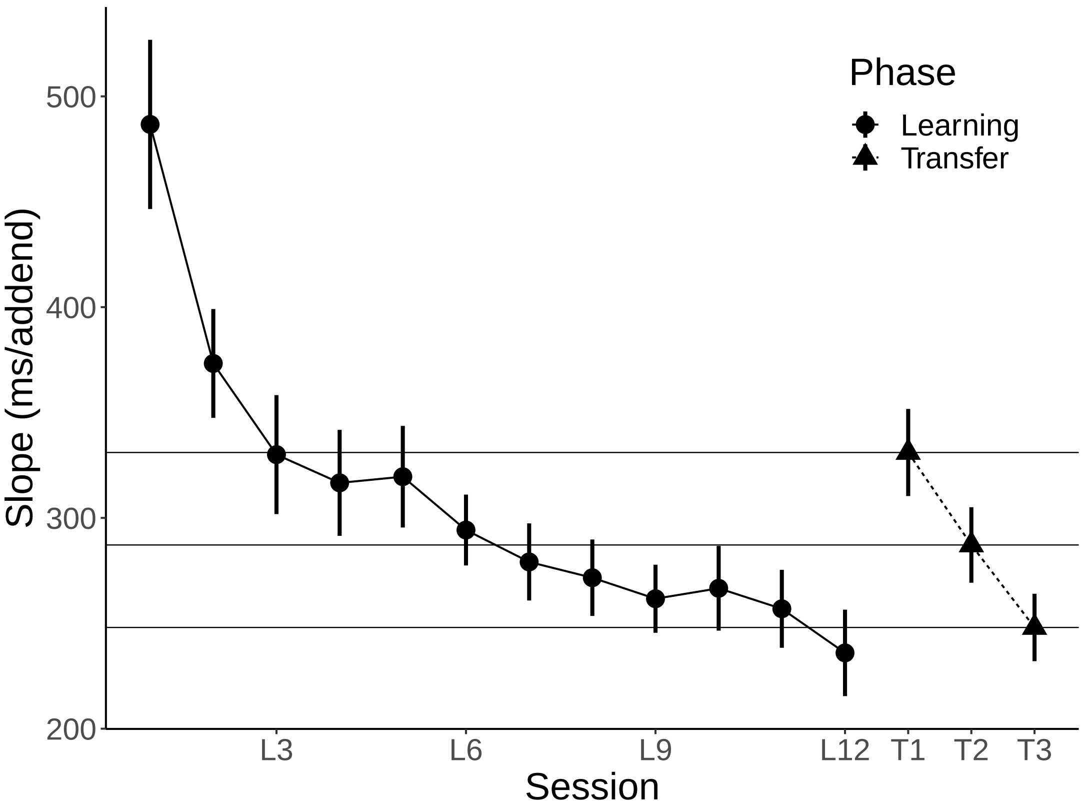
Performance during the Transfer Phase

We carried out a 3 (Session: T1, T2, and T3) × 5 (Addend: 2 to 6) × 2 (Group: 1 or 2) repeated-measures, mixed-design ANOVA on accuracy with Group as the between measure. The results revealed a main effect of Group ($F(1, 15) = 6.40$, $\eta_p^2 = .30$, $p = .02$), with Group 1 having lower accuracy (90%) than Group 2 (96%). The main effect of Addend was also significant ($F(4, 60) = 2.94$, $\eta_p^2 = .16$, $p = .03$), with +2 problems being solved with the highest accuracy (95%) and +5 and +6 problems with the lowest (92%). Neither the main effect of Session nor the interactions were significant.

To analyse solution times, we excluded trials with solution times that were either too short (i.e., shorter than 250 ms) or too long (i.e., longer than the mean plus 3 standard deviations for each participant and each session). This has led

Figure 5.A.2

Addend Slopes as a Function of Sessions in Experiment 1



Note. Addend slopes as a function of sessions during the learning (circles, solid line) and transfer (triangles, dotted line) phases for true equations in Experiment 1, without taking +6 problems into account. Error bars represent standard errors. Horizontal lines indicate the addend slopes corresponding to T1, T2, and T3 sessions.

us to discard 0.5% of the data. A 3 (Session: T1, T2, and T3) × 5 (Addend: 2 to 6) × 2 (Group: 1 or 2) repeated-measures, mixed-design ANOVA on solution times with Group as the between measure was then undertaken.

The main effect of Group was marginal ($F(1, 15) = 3.97$, $\eta_p^2 = .21$, $p = .06$), with Group 1 solved the problems faster (1819 ms) than Group 2 (2264 ms). The effect of Addend was significant ($F(4, 60) = 66.62$, $\eta_p^2 = .82$, $p < .001$), with +2 problems being solved the fastest (1469 ms) and +5 problems the slowest (2385 ms). The two variables interacted ($F(4, 60) = 3.04$, $\eta_p^2 = .17$, $p = .02$).

This interaction was due to +5 and +6 problems being solved faster by Group 1 than Group 2 (−578 ms, $t(15) = -2.30$, $p = .04$ for +5 problems and −647 ms, $t(15) = -2.23$, $p = .04$ for +6 problems).

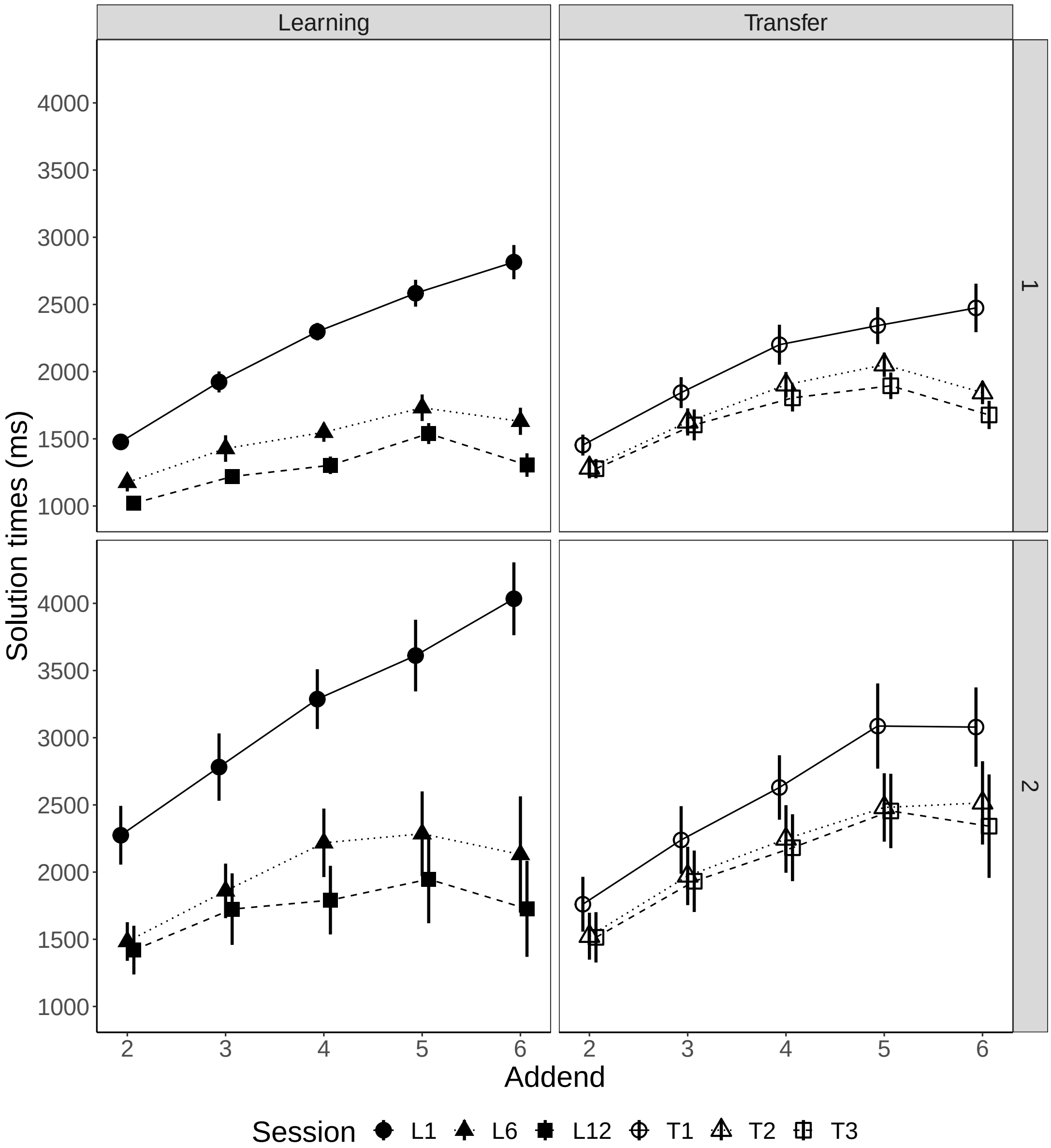
The effect of Session was also significant ($F(2, 30) = 31.57$, $\eta^2_p = .68$, $p < .001$). Solution times in T1 (2311 ms) were significantly longer than in T2 (1944 ms, $t(15) = 6.05$, $p < .001$), which in turn were significantly longer than in T3 (1868 ms, $t(15) = 2.54$, $p = .04$). Session and Addend interacted ($F(8, 120) = 10.05$, $\eta^2_p = .40$, $p < .001$). A series of contrasts revealed significant linear and quadratic trends of the addend effect in all sessions. However, while the linear trend became weaker over sessions (from $t(15) = 14.39$, $p < .001$ in T1 to $t(15) = 5.58$, $p < .001$ in T3), the quadratic trend became more important (from $t(15) = -3.88$, $p = .004$ in T1 to $t(15) = -4.72$, $p < .001$ in T3). This quadratic trend was due to the break in solution times at +5 that started to emerge in Session T2 for Group 1 and in Session 3 for Group 2 (see Figure 5.A.3). There was no interaction between the 3 variables ($F(8, 120) < 1$).

Because of this break, we calculated the addend slopes by including and excluding +6 problems and then ran a 3 (Session: T1, T2, and T3) × 2 (Group: 1 vs 2) repeated-measures, mixed-design ANOVA on addend slope with Group as the between measure for each data set separately. For data set including +6 problems, an effect of Group was found ($F(1, 15) = 6.38$, $\eta^2_p = .30$, $p = .02$), with lower addend slope for Group 1 (173 ms/addend) than for Group 2 (272 ms/addend). There was also an effect of Session ($F(2, 30) = 15.04$, $\eta^2_p = .50$, $p < .001$). The addend slope in T1 (302 ms/addend) was higher than in T2 (202 ms/added, $t(15) = 3.69$, $p = .004$), which was higher than in T3 (164 ms/addend, $t(15) = 3.03$, $p = .02$). There was no interaction between Session and Group ($F(2, 30) < 1$).

When +6 problems were not included, we also found an effect of Group ($F(1, 15) = 4.55$, $\eta^2_p = .23$, $p = .05$), with lower addend slope for Group 1 (255 ms/addend) than for Group 2 (353 ms/addend). The effect of Session was

Figure 5.A.3

Solution Times as a Function of Addend in Experiment 2



Note. Solution times as a function of addend during learning (left panels: solid circles, solid line for L1, solid triangles, dotted line for L6, and solid squares, dashed line for L12) and transfer (right panels: open circles, solid line for T1, open triangles, dotted line for T2, and open squares, dashed line for T3) sessions for Group 1 (top panels) and Group 2 (bottom panels) participants in Experiment 2. Error bars represent standard errors.

also significant ($F(2, 30) = 10.09$, $\eta_p^2 = .40$, $p < .001$). Addend slope in T1 (370 ms/addend) was significantly higher than in T2 (286 ms/addend, $t(15) = 2.89$, $p = .02$), but addend slope in T2 was only marginally higher than in T3 (256 ms/addend, $t(15) = 2.38$, $p = .06$). Furthermore, a series of paired-sample t-tests revealed that addend slopes without +6 problems were significantly higher than with +6 problems ($t(16) = 3.28$, $p = .005$ in T1, $t(16) = 5.16$, $p < .001$ in T2, and $t(16) = 4.84$, $p < .001$ in T3).

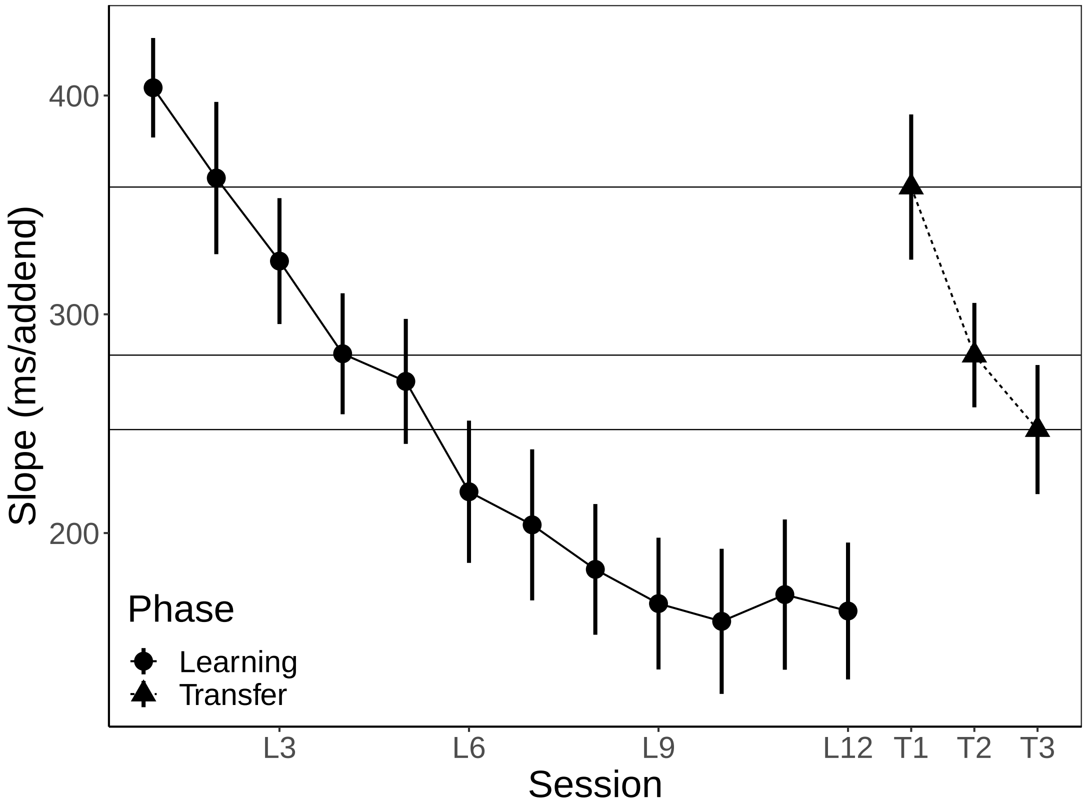
Transfer from Learnt to New Items

To test whether transfer occurred, we compared addend slopes without +6 problems between L12 and the 3 transfer sessions. A 4 (Session: L12, T1, T2, T3) × 2 (Group: 1 vs 2) mixed-design repeated-measures ANOVA on the addend slope with Group as the between variable was carried out. The effect of Group ($F(1, 15) = 2.90$, $p = .11$) and the interaction between Group and Session ($F(3, 45) = 1.70$, $p = .18$) were not significant and therefore we collapsed the two letter sets and carried out a one-way ANOVA. The results revealed a significant effect of Session ($F(3, 48) = 13.04$, $\eta_p^2 = .45$, $p < .001$).

Considering that we are interested in the difference in addend slopes between L12 and the 3 transfer sessions, we conducted a series of paired-sample t-tests with L12 as the reference level. Addend slope in Session L12 (164 ms/addend) was significantly lower than in T1 (358 ms/addend), i.e., −194 ms/addend ($t(16) = -3.92$, $p = .003$), T2 (281 ms/addend), i.e., −117 ms/addend, $t(16) = -4.23$, $p = .002$), and T3 (247 ms/addend), i.e., −83 ms/addend, $t(16) = -3.10$, $p = .007$). Figure 5.A.4 shows that the addend slope in T1 corresponded to the addend slope in L2, that addend slope in T2 corresponded to the addend in L4, and that the addend slope in T3 corresponded to an addend slope between L5 and L6.

Figure 5.A.4

Addend Slopes as a Function of Sessions in Experiment 2



Note. Addend slopes as a function of sessions during the learning (circles, solid line) and transfer (triangles, dotted line) phases in Experiment 2, without taking +6 problems into account. Error bars represent standard errors. Horizontal lines indicate the addend slopes corresponding to T1, T2, and T3 sessions.

Appendix 5.B

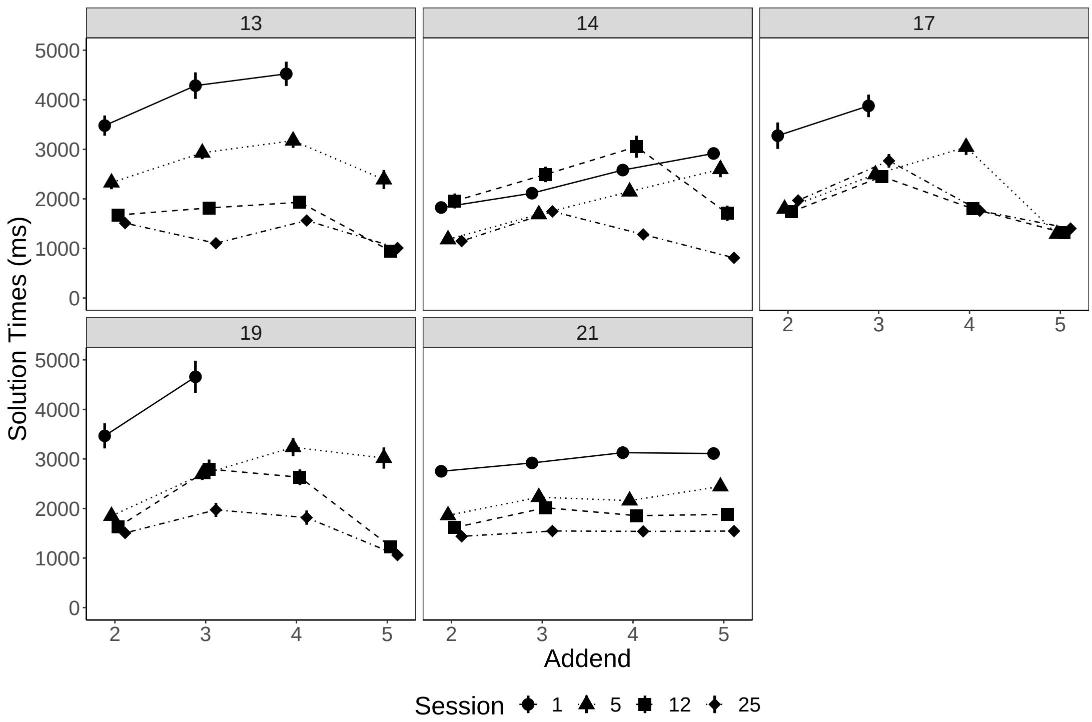
Performance of Retrievers

in Experiment 1 of Chapter 2

Retrievers are defined as participants who had non-significant addend slopes at the end of learning, even when problems with the largest addend were excluded from the addend-slope calculation.

Figure 5.B.1

Performance of Retrievers in Experiment 1 of Chapter 2



Note. Solution times as a function of addend for true equations in Sessions 1 (solid circles, solid line), 5 (triangles, dotted line), 12 (squares, dashed line), and 25 (diamonds, dash-dotted lines) for the 5 retrievers in Experiment 1 of Chapter 2. Error bars represent standard errors. Due to the large values, solution times for Sessions 1 are not always fully presented.

6

Discussion

This thesis aims at examining the theories of learning, particularly those related to addition learning. The classical retrieval models (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996) claimed that addition learning involves a progression from counting to retrieval whereas non-classical models argued that counting could evolve into not only retrieval but also the use of rules and heuristics (e.g., Baroody, 1983, 1984, 1994, 2018) or into automated counting (Barrouillet & Thevenot, 2013; Thevenot & Barrouillet, 2020; Uittenhove et al., 2016). It has been largely acknowledged that by means of the alphabet-arithmetic paradigm (Logan & Klapp, 1991), the instance theory of automatization (Logan, 1988) provides support for the retrieval models of mental addition, namely that after repeated practice, long and costly algorithm-based procedures are unavoidably replaced by a single-step retrieval strategy. However, detail examinations of the results of the past alphabet-arithmetic studies cast doubt on the claimed support and may instead corroborate the non-retrieval models, e.g., that addend slopes may not be non-significant at the end of practice, that the notion of opportunistic stopping may not be compatible with the use of counting, and that transfer to new items may take place.

Therefore, in this thesis I revisited the instance theory of automatization in general and the alphabet-arithmetic paradigm in particular. More precisely, I have identified 5 research questions that I answered by means of 3 alphabet-arithmetic training experiments. These 3 experiments are summarised in Table 1.2 and throughout this Discussion the same numbering as in this table will be used. However, before describing the answer to the research questions, I will start by describing the classical finding of alphabet-arithmetic studies, namely that a discontinuity in solution times is systematically observed towards the end of practice (e.g., Beilock & Carr, 2001; Chen et al., 2020; Compton & Logan, 1991; Logan & Klapp, 1991; Wenger, 1999; Zbrodoff, 1995, 1999). In other words, solution times for problems with the largest addend are lower than for problems with the second-largest addend. It is important to discuss this phenomenon first because it is the basis of the research questions of this thesis.

Replicating previous studies, such a discontinuity was found in our 3 experiments that included verification as well as production tasks, addends from 2 to 5 or 6, and run over 12 or 25 sessions (see Table 1.2), confirming the intrinsic nature of this phenomenon to the alphabet-arithmetic paradigm. We found that irrespective of the magnitude of the largest addend in the study set, the discontinuity was always obtained at problems with the second-largest addend, i.e., at +4 when +5 was the largest addend (see Chapter 2) and at +5 when +6 was the largest addend (see Chapters 2 and 4). This observation implies that this phenomenon is tightly related to problems with the largest addend, and not to a particular addend.

To explain why solution times for problems with largest addend were lower than for problems with the second-largest addend, Logan (1988) suggested that whereas problems with the largest addend were solved by deliberate memorisation, problems with smaller addends were solved by automatic encoding. Indeed, these two ways of memorisation were possible according to the instance theory of automatization (Logan & Klapp, 1991). Nevertheless, between these two ways of memorising, the theory did not provide an explanation as to why deliberate memorisation for +5 problems would lead to shorter solution times than memorisation by automatic encoding for +4 problems, nor as to why one type of memorisation is privileged over the other for a specific type of problems. In fact, the theory assumes that the probability of retrieval depends on the number of encounters with the problems. Because all problems were presented with the same number of repetitions, memorisation should occur at the same moment for all problems. Therefore, even if both types of memorisation are possible, we should have a relatively linear addend slope without a decrease in solution times for problems with the largest addend. The fact that solution-time discontinuity is systematically observed indicates instead that the presupposition about two different ways of memorisation does not hold. In other words, if problems with smaller problems are memorised then problems with the largest addend are not, and vice versa. Another way of interpreting this result is that automatic encoding follows the postulation of instance theory of automatization, i.e., is governed by the number of instances, whereas deliberate memorisation does not. If this is the case, then problems with the largest addend, which are purportedly solved by deliberate memorisation, should not be analysed together with problems with smaller addends. Therefore, considering the role of addend slope in determining the used strategy, we advocate that problems with the largest addend should be removed from the addend-slope calculations.

After having reported this replication, I will now detail the answer to the 5 research questions. The first research question was taken on in Chapter 2 and is related to addend slopes, i.e., whether removing problems with the largest addend would result in non-significant addend slopes. Our results showed that addend slopes at the end of practice were still significant, whether problems with the largest addend were included or excluded (see Table 6.1). More importantly, these results were also found in Experiment 1 that was run over 25 sessions, implying that the significant addend slopes at the end of training was not due to a lack of practice. In fact, the lack of difference in addend slopes between

Table 6.1

Addend Slopes with and without Problems with the Largest Addend

Experiment		1		2	3
Task		Verification		Verification	Production
Largest addend		5		6	6
		Session 12	Session 25	Session 12	Session 12
All	with	163	108	163	86
	without	286	217	236	165
Non-breakers	with	247	266	262	
	without	256	272	306	
Breakers	with			67^{ns}	87
	without			208	233
Retrievers	with	-102^{ns}	-129^{ns}	not applicable	-4^{ns}
	without	266^{ns}	40^{ns}		5^{ns}

Note. The values for breakers in Experiment 1 and non-breakers in Experiment 3 are not presented due to the very-small size ($N = 2$).
^{ns} Non-significant addend slopes.

Sessions 12 and 25 in Experiment 1 showed that participants have reached their asymptotic performance. A significant addend slope at the end of practice indicates that the possibility that alphabet-arithmetic problems were solved by a counting strategy after repeated practice cannot be discarded. This interpretation invalidates the postulation of the instance theory of automatization (Logan, 1988) that algorithm-based performance evolves necessarily into retrieval-based performance.

The second research question was taken on in Chapter 3 and is related to opportunistic stopping. More precisely, whether opportunistic stopping really indicates the use of counting. As a matter of fact, our results showed the dissociation between opportunistic stopping and the use of counting strategy. This is because, on the one hand, in the first session when counting was the dominant strategy, within-count equations were not solved faster than true equations, and on the other hand, within-count equations were solved faster than outside-of-count equations only for +2 and +3 problems. Towards the end of practice, within-count equations were solved descriptively slower than outside-of-count equations for +4 and +5 problems. We argued that this latter phenomenon, which was related to the memorisation of problems with the largest addend, i.e., 5, was responsible for the disappearance of equation-type effect found by Zbrodoff (1999) at the end of her training experiment.

The third research question is taken on in Chapter 4 and is related to the replicability of verification-task results, i.e., whether results from an alphabet-arithmetic verification task are replicable in a production task or specific to a verification task. Using the same material and procedures as in Experiment 2 using a verification task, we found a faithful reproduction in Experiment 3 using a production task, namely the existence of a solution-time discontinuity, significant addend slope at the end of practice, and similarities in accuracy and solution-time reduction across practice. However, although our verification and production tasks yielded the same results, care should be taken in generalising this conclusion to other tasks such as mental arithmetic. Indeed, it is possible that the perfect replication found in our study is limited to alphabet-arithmetic paradigm which does not include a large split effect (e.g. Ashcraft & Battaglia, 1978; De Rammelaere et al., 2001; Zbrodoff & Logan, 1990) and which, by definition, does not involve parity or multiple-of-5 rules (e.g., Krueger, 1986; Krueger & Hallford, 1984; Lemaire & Fayol, 1995; Lemaire & Reder, 1999; Masse & Lemaire, 2001), i.e., cases that are easily solved by plausibility judgement (Reder, 1982; Zbrodoff & Logan, 1990).

The results from the first 3 research questions revealed the important role of problems with the largest addend. As explained earlier, one possible explanation for the decrease in solution times for problems with the largest addend was that these problems were solved by deliberate memorisation whereas problems with smaller addends were solved by automatic encoding (Logan, 1988). However, as argued earlier, this argument is not convincing. Therefore, instead of these two ways of memorisation, we proposed another interpretation, namely that there are two different strategies used to solve different problems. More precisely, problems with the largest addend were solved by memory retrieval whereas problems with smaller addends by counting. In fact, retrieving the answer to problems with the largest addend could be faster than using counting to solve problems with the second-largest addend.

However, regardless of the interpretation, a mechanism is required to explain why problems with the largest addend were processed differently than problems with smaller addends. One possible mechanism is explained by the horse-race model (Logan & Cowan, 1984; Logan et al., 1984). Solving problems with the largest addend using one-by-one counting might take much longer time than retrieving the association between the problem and the answer from long-term memory, such that memorising the association is privileged over counting. For problems with smaller addends, on the other hand, using counting might result in shorter or comparable time than using retrieval. However, retrieval network is prone to interference (e.g., Campbell & Graham, 1985) which may lead to retrieval failure. In this case, counting could be deemed more cognitively economical (e.g., Baroody, 1983, 1984, 1994, 2018; Baroody & Ginsburg, 1986). Indeed, the horse-race model can explain the mechanism behind solution-time discontinuity. This is because in all alphabet-arithmetic studies, the decrease in solution times was always observed for problems with the largest addend in the study set, and not for problems with an addend above a certain threshold.

Another possible mechanism to explain the special processing for problems with the largest addend is the fan effect (J. R. Anderson & Reder, 1999; Pirolli & Anderson, 1985), according to which the strength of an association between a problem and its answer is higher when there are less concepts to learn or, in other words, when the probability of interference between concepts is low. Considering that our material was constructed by pairing a letter augend with consecutive addends, for a given letter augend, the answer to the problem with the largest addend is encountered only once. Take as an example the letter augend E in an experiment with +5 as the largest addend. From the letter E, the only moment that participants have to scan through the letter J is to solve +5 problem. Thus, after many repetitions, they have no doubt that J is the answer to E + 5 and therefore, in this case of low fan effect, the association between E, 5, and J can safely be committed to and retrieved from long-term memory. On the contrary, from the letter E, the letter H is scanned when participants solve +3, +4, and +5 problems. Thus, participants might be confused as to which addend the letter H is associated. In this case of high fan effect, they might decide that it would be safer to use counting for problems with smaller addends.

The description of fan effect detailed in the preceding paragraph is related to fan effect within a particular letter augend and affects only problems with the largest addend. However, fan effect could also operate in the context of study set. Keeping an experiment with +5 as the largest addend as an example, the letters C and O are the correct answer to only 1 problem in the whole study set, i.e., A + 2 and J + 5, respectively. Oppositely, the letter J is the correct answer to 4 problems in the whole study set, i.e., E + 5, F + 4, G + 3, H + 2. Thus, due to less interference, memorising J + 5 and A + 2 would be easier than memorising F + 4. In this case, we should observe similar solution times for problems with the largest as with the smallest addend. However, for problems with the smallest addend, it will be difficult to distinguish between retrieval and counting because both may involve comparable solution times. Furthermore, instead of counting 2 steps, +2 problems could be easily solved by the skip-letter-after strategy, that is similar to the "skip the number after N" strategy(Baroody, 2018).

Still another possibility to explain the decrease in solution times for problems with the largest addend is that participants try to memorise the studied facts. However, given the 40 problems that they have to memorise, they might find it more practical to memorise the 8 (i.e., in Experiments 2 and 3) or 10 (i.e., in Experiment 1) problems with the largest addend first. This is different from the horse-race model described earlier, in that there is not necessarily a competition between the two procedures to finish a race. I will return to this possibility later when I described the individual differences.

Thus far, I have shown that at the sample level, there is a decrease in solution times for problems with the largest addend, that is interpreted as the indication for the use of memory retrieval. For problems with smaller addends, on the other hand, the steady increase of solution times with addends indicates the use of counting. In other words, when a shift from counting to retrieval occurs, it is only for problems with the largest addend. Nevertheless, at the individual level, the sensibility to problems with the largest addend was not observed in all participants. Chapters 2, 4, and 5 revealed that the solution-time discontinuity in our 3 experiments was systematically obtained in two groups of participants that we call the breakers and the retrievers but was never observed in participants that we call non-breakers. In another group that we called unclassified, solution-time discontinuity was found but not systematically. The sensibility of the breakers to problems with the largest addend was also manifested in the reversal of the difference in rejection times, that was responsible for the disappearance of the opportunistic-stopping phenomenon in Chapter 3. Table 6.2 shows the number of participants in these groups in the 3 experiments.

Table 6.2

Contingency Table of Participants according to Solution-Time Discontinuity

Experiment	Task[a]	All	Non-breakers	Breakers	Retrievers	Unclassified
1	V	19	6	2	5	6
2	V	21	4	7	0	10
3	P	17	2	7	5	3

Note.
[a] Task: P for production and V for verification.

As the name indicates, non-breakers did not show a solution-time discontinuity throughout practice. The continually-increasing solution times across addends provided a strong indication that these participants solved problems with all addends using counting strategies. The breakers, on the other hand, showed a solution-time discontinuity at problems with the second-largest addend. As explained earlier, this indicates a memorisation for problems with the largest addend. Interestingly, the performance of the breakers in terms of addend slopes was comparable in verification as in production tasks that used the same materials (see Table 6.1). Considering the significant addend slopes when problems with the largest were not included, it is likely that problems with smaller addends were solved by counting. Naturally, following the study of Siegler (1987), significant addend slopes could also result from the averaging of solution times from trials using different strategies. Contrary to the breakers, the retrievers seem to use retrieval for all problems, because their addend slopes were non-significant, even when problems with the largest addend were included. In other words, solution times for this group were relatively constant for all addends, from the smallest to the largest. Thus, our results revealed individual differences in alphabet-arithmetic learning, that we will develop later in the Discussion.

The fourth research question was taken on in Chapter 5 and is related to transfer phenomenon, i.e., whether transfer from learnt to new items would be observed when 3 transfer sessions, instead of 1, are conducted. This question should be examined together with the fifth research question that was taken on in the same chapter and is related to individual differences, i.e., whether the strategy used during the learning phase is related to that during the transfer phase and whether multiple strategies are used by participants. In both verification and production tasks, we found a dependency between the strategy used during the learning and the transfer phase. Therefore, in line with the results of Haider and Frensch (2002), it is likely that a shift of strategy during learning is an intentional decision of some participants, and not the eventual outcome that could be expected from all learners.

More importantly for our transfer study, the non-breakers, who showed a strong indication of using counting as the dominant strategy during learning, showed transfer immediately in the first transfer session. This transfer of procedure to new items confirms our conclusion that counting was used by this group during practice. The performance of the breakers and retrievers, on the other hand, was similar to each other in the sense that transfer was only obtained in the third transfer session. For the breaker group, we have described earlier that addend slope at the end of learning was significant when problems with the largest addend were not considered, suggesting the use of counting for problems with smaller addends. Nevertheless, the comparison to the non-breaker group during transfer indicates that memory retrieval was also used by breakers to solve these small problems. Thus, it is possible that transfer of procedure in this group was hindered by the use of mixed strategies. The similarity in performance during the transfer sessions between breakers and retrievers indicates further that once memory retrieval is used, the application of learnt procedures to solve new items has to be relearnt from scratch.

The existence of the retriever group seems to confirm the instance theory of automatization (Logan, 1988) that learning could consist in a shift of strategy from counting to retrieval. Nevertheless, the theory does not predict the existence of solution-time discontinuity nor the progression in the memorisation of problems. More precisely, the first problems to be committed to memory seem to be those with the largest addend. From there, participants memorise either problems with second-largest addend or problems with the intermediate addend.

6.1 Implications

Among the results obtained from our alphabet-arithmetic training experiments presented in this thesis, the most important ones concern the particular status of problems with the largest addend, the use of counting after repeated practice, the fact that retrieval does not occur for all addends at the same time but progressively starting from the largest addend, and the inter- and intra-individual differences.

As described in the Introduction, several studies yielded significant addend slopes at the end of practice, probably due to a low number of repetitions (e.g., Campbell et al., 2016; D'Eredita & Hoyer, 2010; Rawson & Touron, 2015; Wilkins & Rawson, 2010), a high number of problems to learn (e.g., Rabinowitz & Goldberg, 1995; Wilkins & Rawson, 2010), or similarities between problems (e.g., White et al., 2007). In this thesis, however, the use of counting by some participants cannot be explained by one of these bottom-up manipulations. Indeed, we used the same number of problems as in Experiment 1 of Logan and Klapp (1991) and the problems were presented more than 48 times throughout the learning phase, which should lead to an asymptotic performance (Logan & Klapp, 1991; Zbrodoff, 1995). Instead, the use of counting seems to be related to participants' strategy choice, that I will describe later.

Concerning the special status of problems with the largest addend, I have described the possible explanations earlier in this Discussion. Regarding individual differences, Haider and Frensch (2002) have observed the dependency between strategy used during transfer phase and during learning phase. However, in this thesis we went further by revealing that the shift in strategy from counting to retrieval observed in some participants occurred progressively and depended on the addends. In fact, the discovery of progressive retrieval in several participants constitutes a novelty that has never been discussed in previous alphabet-arithmetic literature.

Therefore, the results of this thesis have implications not only for researches on the theories of learning but also for education. In what follows, I will discuss these implications, specifically in light of individual differences.

Instance theory of automatization

The results revealed in this thesis may call the instance theory of automatization (Logan, 1988) into question, particularly its assumption that the probability of retrieval depends principally, or even solely, on the number of traces created in the memory or, in other words, on the number of repetitions (hereinafter: retrieval-probability assumption). If we only consider the situation at the end of practice, within this assumption, the theory gives the impression of being able to account for inter-individual differences, namely the use of counting by non-breakers and memory retrieval by retrievers. Although not elaborated by the theory, it is possible that the number of traces required to allow for memory retrieval could depend on the individual. For example, for a problem to be retrieved, faster learners would require fewer number of traces than slower learners. Furthermore, the possibility that counting could still be used at the end of learning is recognised by the theory of automatization (Logan & Klapp, 1991), although its proponents were of the opinion that retrieval was the dominant strategy.

However, the retrieval-probability assumption of the instance theory of automatization cannot explain intra-individual differences observed in breakers, e.g., why problems with the largest addend were memorised before other problems. In our experiments as in Logan and Klapp (1991)'s, all problems were presented with the same frequency and therefore should be memorised at the same time. However, some problems were memorised whereas other problems were not. Furthermore, some problems were memorised before the others. Within the framework of the instance theory of automatization, this complication could probably be rectified by taking the postulate of obligatory encoding into account in the retrieval probability. According to this postulate, encoding is obligatory but the quality of encoding depends on the quantity and quality of attention. Nevertheless, how the obligatory-encoding postulate interacts with the retrieval-probability assumption was not elaborated by the instance theory of automatization. I think that actually, it is possible that the probability of retrieval depends not only on the number of traces but also on the quality of traces. More precisely, probably due to the fan effect described earlier, problems with the largest addend might receive more attention and therefore might create better-quality traces compared to problems with other addends. In turns, good-quality traces might have higher probability of being retrieved and be retrieved faster than bad-quality ones. As a consequence, whereas the alternative algorithm does not have a chance to win the race against good-quality traces, it could still win against bad-quality ones. This may explain the use of retrieval for problems with the largest addend and counting for problems with smaller addends.

Of course, as has been developed earlier (see page 192), the instance theory of automatization could account for the effect of the largest addend by means of the horse-race model. In other words, without having to rely on the assumption that the number of traces stored in memory depends solely on the number of repetitions (i.e., the retrieval-probability assumption). Nevertheless, it is unclear how the theory could explain the progressive retrieval observed among the retrievers, particularly those showing the M pattern (e.g., Participants 107 and 120 in Figure 5.4). As a reminder, these participants memorise first problems with the largest addend, then problems with intermediate addend, e.g., +3, and finally by the end of practice all problems are memorised. This kind of progression is not predicted by the instance theory of automatization, nor is it explainable by the fan effect. Instead, there seem to be some top-down mechanisms involved, for example participants' intentional decision as to which problems to memorise at a given time (see also Haider & Frensch, 2002).

In fact, compared to top-down mechanisms, bottom-up mechanisms that are directed by the stimuli only played a minor role. In her alphabet-arithmetic study, Zbrodoff (1995) manipulated two bottom-up factors, i.e., frequency and interference, in her attempt to replicate the problem-size effect in mental addition. She showed that when both frequency and interference were manipulated, addend slopes were significant, simulating the problem-size effect. On the contrary, when frequency alone or interference alone was manipulated, addend slopes were not significant. In our experiments, however, there was no effect of frequency, because all problems have the same number of repetitions. Interference, on the other hand, was present and played a role in the largest-addend effect. Nevertheless, with interference alone, our experiments showed significant addend slopes, contrary to Zbrodoff's results. More importantly, the effect of interference differed between individuals and was only found among the breakers and retrievers. Thus, even when bottom-up mechanisms were present, it is top-down mechanisms that played a more-important role in participants' performance. Indeed, this exclusion of top-down mechanisms from the instance theory of automatization was a subject of criticism in the literature (e.g., Haider & Frensch, 2002; Wilkins & Rawson, 2010).

Another shortfall of the instance theory of automatization that is relevant to be discussed here was detected by the proponents of the ACT–R and concerns the lack of possibility of erroneously retrieving similar traces (e.g., Gonzalez et al., 2003; Taatgen et al., 2006). In fact, according to the instance theory of automatization, only traces identical to the problem encountered at a given moment are included in the race against the alternative algorithm. However, it is also possible, notably in the case of high interference, that similar traces participate in the race as well. For example, in the attempt to retrieve the answer to F + 4, the traces for F + 3 or G + 4 might also be retrieved, resulting in an error. Campbell and Graham (1985) showed that most errors in multiplications are table related, indicating that similarities between problems are likely to induce retrieval errors. Taking retrieval errors into account in the instant theory of automatization may in fact be important, considering that after 12 sessions the accuracy is still at 94% or 95% (Logan & Klapp, 1991, see also Chapters 2 and 4). As a comparison, in the modelling of cognitive arithmetic using the ACT–R theory, Lebiere (1999) included not only errors during retrieval but also errors during counting that are stored as erroneous answers, as well as persistent errors that remain uncorrected for a long time.

Retrieval models of mental arithmetic

The fact that the instance theory of automatization (Logan, 1988) is put in question by the results of this thesis, does not automatically imply that the retrieval models of mental arithmetic (e.g., Ashcraft, 1982, 1992; Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018; Siegler, 1996) are also in doubt. This is because, as described in the Introduction, these models are based on strength and not instance theory. Furthermore, within the framework of the retrieval models, notably the strategy-choice or overlapping-waves model (e.g., Shrager & Siegler, 1998; Siegler, 1996; Siegler & Jenkins, 1989; Siegler & Shipley, 1995; Siegler & Shrager, 1984), intra-individual differences are possible, although the use of counting by adults or experts is considered to be the last recourse after the failure of all retrieval attempts. Nevertheless, none of the retrieval models could account for the largest-addend effect and the progressive retrieval from the largest addend found in this thesis.

As described in the Introduction, problem-size effect in mental addition could be interpreted in terms of frequency (e.g., Ashcraft, 1992; Ashcraft & Guillaume, 2009) or interference (e.g., Chen & Campbell, 2018). This interpretation was confirmed by Zbrodoff (1995)'s results on alphabet-arithmetic task reported earlier in this Discussion. As explained earlier, frequency did not play a role in our alphabet-arithmetic experiments, but interference did through the largest-addend effect. Therefore, in line with the network-interference model (e.g., Campbell, 1995; Campbell & Oliphant, 1992; Chen & Campbell, 2018), our experiments revealed that problems with less interference are solved faster than problems with more interference. However, the difference between the model and our results lies in the type of problems that have stronger or weaker interference. Whereas the network-interference model purports stronger interference for larger problems, our results revealed that problems with the largest addend are the least prone to interference. However, although the largest-addend effect per se does not exist in the learning of mental arithmetic, the salience of problems with the largest addend may parallel the salience of, for example, tie problems that are solved faster than non-tie problems (e.g., Bagnoud, Dewi, Castel et al., 2021; Blankenberger, 2001; LeFevre et al., 2004) or mutliple-of-5 problems that are solved faster than other multiplication problems (e.g., Campbell & Graham, 1985; LeFevre, Bisanz et al., 1996).

Another element of the results of this thesis that the retrieval models cannot reconcile concerns the progressive retrieval, i.e., problems with the largest addend were memorised first, followed by problems with smaller addends. Of course, the network-retrieval model purports that problems that are learnt earlier in the course of development are memorised earlier (Ashcraft & Christy, 1995; Hamann & Ashcraft, 1986). However, it is not clear whether problems that are learnt at the same time are memorised at the same time, or whether more-salient problems are memorised earlier than less-salient problems. Similarly, Campbell and Graham (1985) found that children and adults participants made less errors when solving 8×9 problems than when solving 8×4. Although both problems belong to the table of 8, it is most likely that 8×4 is learnt earlier than 8×9 in its commutative form of 4×8. In this case, therefore, lower error rates for 8×9 than for 8×4 may not correspond to 8×9 being learnt earlier than 8×4.

Among the retrieval models of mental arithmetic, only the overlapping-waves model (e.g., Siegler, 1996) could explain progressive retrieval, because problems with a peaked distribution are supposed to be memorised faster than those with a flatter distribution. As explained in the Introduction, according to the model, the peakedness of the distribution of a problem depends on the success rates in solving the problem. Hence, in this context, smaller and easier mental-arithmetic problems would have a peaked distribution whereas larger and more difficult problems a flat distribution. Nevertheless, in our 2 verification experiments, accuracy decreased with addend and hence problems with the largest addend should be memorised the latest.

Automated counting procedure theory

The results of this thesis revealed that, on the one hand, the use of counting after an extensive training cannot be discarded and, on the other, there was a group of participants who used counting exclusively until the end of practice. Therefore, these results seem to be able to corroborate the automated counting procedure theory (e.g., Thevenot & Barrouillet, 2020; Uittenhove et al., 2016). Indeed, the theory could account for individual differences, both at the level of strategy choice – i.e., counting for additions involving two operands inferior to 5 and retrieval or decomposition for larger additions – and strategy execution – i.e., stronger problem-size effect for participants with lower than with higher working-memory capacity (Barrouillet & Thevenot, 2013; Uittenhove et al., 2016). In fact, the automated counting procedure theory is the only one that postulates the use of counting for smaller addition problems and retrieval or decomposition for larger addition problems – a situation that corresponds to the performance of breakers in this thesis.

A preliminary computational model of the automated counting procedure theory was recently developed (Chouteau et al., 2021). The model took attention and working memory into account, as well as past experience in solving alphabet-arithmetic problems. However, for the moment the model ignored the possibility of retrieval. Thus, as predicted by the theory, the model revealed that counting was accelerated across sessions. A simulation over 100 sessions revealed a counting speed of 48 ms/addend, similar to the 45 ms/addend found by Uittenhove et al. (2016) in frequent retrievers. This simulated speed was still much lower than the addend slope of 217 ms/addend found in Session 25 of Experiment 1 (see Table 6.1). Therefore, even if the behavioural pattern of breakers could represent what is predicted by the automated counting procedure theory, the performance at the end of learning does not correspond yet to an automated counting but only a performance towards an automated counting.

Nevertheless, the current state of the theory needs to address several observations obtained in this thesis. First of all, although the theory could explain the general performance of the breakers, i.e., the use of memory retrieval for problems with the largest addend and counting for problems with smaller addends, the use of counting for +5 problems by breakers in Experiments 2 and 3 was not predicted by the theory. As a reminder, these experiments involved 6 as the largest addend. In fact, according to the theory, counting should be used for addition problems with two addends up to and including 4, larger problems being solved by either retrieval or decomposition. This is because the operand 4 may be connected to the number of chunks that can be kept in short-term memory (e.g., Cowan, 2000) and may be related to the average limit of subitizing (Kaufman et al., 1949). Nevertheless, both the number of chunks and subitizing range limit vary between individuals (e.g., Melcher & Piazza, 2011; Piazza et al., 2011; Tuholski et al., 2001). Therefore, to explain the use of counting for +5 problems by breakers, the theory could loosen the range of operands within which counting is used. More precisely, the upper limit is not fixed to 4 but depends on the individual's subitizing range and/or short-term memory span.

Nevertheless, if we follow this logic, then we should expect individuals with high memory span to use counting for larger problems. Yet, in the study of Barrouillet and Thevenot (2013), it was the retrievers that had the highest working-memory capacity. Furthermore, although the performance of the breakers seems to conform to the automated counting procedure theory, the mechanism that underlies the performance for problems with the largest addend in an alphabet-arithmetic task may not be comparable to the mechanism for larger mental-addition problems. More precisely, the theory purports that larger additions are solved by memory retrieval or decomposition because larger operands are beyond the limit of the number of chunks and subitizing and therefore cannot be solved by automated counting. In an alphabet-arithmetic task, however, problems with the largest addend are solved by retrieval because they have less interference than problems with smaller addends. One possible source of disagreement may be that alphabet-arithmetic tasks rely necessarily on a finite sequence whereas numerical arithmetic is based on an infinite sequence.

Another problem that is difficult to reconcile is that whereas the automated counting procedure theory could explain the performance of breakers, it could not account for the performance of non-breakers and retrievers. More precisely, at the end of practice, non-breakers used counting to solve relatively all problems and retrievers used retrieval to solve all problems. In other words, the performance of the two groups did not follow the prediction of the theory, namely that smaller problems are solved by counting whereas larger problems by retrieval. The total proportion of retrievers in our experiment, i.e., 10 out of 57 or 17% (see Table 6.2), was similar to that found in the study of Barrouillet and Thevenot (2013), i.e., 15 out of 91 or 16%. In the latter study involving additions with operands from 1 to 4, the retrievers were defined as those who have a slope inferior to 5 ms/unit increment for the first as well as for the second operands. This slope is similar to the addend slopes found for retrievers at the end of Experiment 3 (see Table 6.1). This group of retrievers in Barrouillet and Thevenot's study had higher working-memory capacity and solved additions faster than other participants. It is hence very likely that the retrievers in this thesis also have high working-memory capacity.

On the contrary, it is more difficult to parallel the performance of the non-breakers in this thesis to the performance of adults in mental additions found in the literature. The speed of counting for non-breakers seems to stagnate at about 300 ms/addend (see Chapter 5). In fact, there was no more acceleration after the second learning session. This speed is much larger than the rate of silent counting, i.e., 125 ms/unit increment (Landauer, 1962). Furthermore, due to the stagnation throughout the learning phase, it is questionable that after 100 sessions the counting speed could reach the value of 48 ms/addend obtained by the simulation of the automated counting procedure theory described previously (Chouteau et al., 2021). It is indeed possible that throughout the training sessions the non-breakers did not learn, either to memorise or to speed up their counting.

To conclude, the group whose performance is best described by the automated counting procedure theory is the breakers. Nevertheless, the theory cannot explain the progressive retrieval from the largest addend observed in this group.

All things considered, the results of this thesis put the instance theory of automatization into question. Because the theory is disputable, its support for the retrieval models of mental arithmetic is also debatable. Nevertheless, the results of this thesis fail to readily support the competing theory, i.e., the automated counting procedure theory. Instead, a theory based on mixed strategies, either at the inter-individual or intra-individual level (including inter-problem and intra-problem differences), would be more appropriate to explain the observed performance. All in all, the results of this thesis revealed the complexity of modelling a learning theory that could be suitable for all individuals. I advocate that to test a model that aims at understanding learning processes, it is important to analyse the data not only at the sample level but also at the individual level.

After having shown the implications of the results found in this thesis for research on learning, in what follows, I will describe the implications for arithmetic instruction.

Learning and instruction

Table 6.2 shows the proportion of participants in non-breaker, breaker, retriever, and unclassified groups in our 3 experiments. As a reminder, this classification was based on whether solution-time discontinuity was observed or not. In other words, whether solution times were shorter for problems with the largest addend than for problems with the second-largest addend. Considering that all participants in each of the 3 experiments received the same instruction and learnt the same study materials, the individual differences described in this thesis may indicate that our participants adopted a strategy that they considered the most appropriate for themselves. The non-breakers used a counting procedure that has a disadvantage of being slower than using retrieval. This is shown, for example, in Figure 2.5, where solution times for +5 problems in breakers were lower than solution times for +2 in non-breakers. However, as shown in Chapter 5, the advantage is that this procedural knowledge is readily transferable to new items. It is likely that these participants knew they could memorise but chose not to, probably because they found counting less effortful. For the breakers and retrievers, on the other hand, although transfer was not obtained in the first transfer session, the same level of performance that they reached in 12 learning sessions could be attained in 3 transfer sessions, indicating that their choice of strategy is suitable for their learning process. Therefore, we can conclude that participants' strategy choice is adaptive.

The question of adaptivity, or making the appropriate strategy choice, is a major challenge in mathematics learning and instruction (e.g., Baroody, 2003; Threlfall, 2009; Verschaffel et al., 2009). In this Discussion, I adopt the distinction proposed by Verschaffel et al. (2009) between flexibility, i.e., "switching (smoothly) between different strategy" (p. 337) and adaptivity, i.e., "selecting the most appropriate strategy" (p. 337). Acevedo Nistal et al. (2009) and Verschaffel et al. (2009) argued further that the notion of appropriateness in strategy adaptivity should depend not only on the problem or task characteristics but also on the person and the context. For example, the differences between non-breakers and breakers revealed in this thesis illustrate the person-based adaptivity.

In face of multiple strategies, flexibility and adaptivity are two important aspects in strategy choice (Lemaire & Siegler, 1995). At the empirical level, flexibility and adaptivity of strategy choice are often studied using the choice/no-choice paradigm (Siegler & Lemaire, 1997). During the choice condition, participants are free to use any strategy whereas during the no-choice conditions, they are forced to use one strategy in particular. Researches on children's multi-digit arithmetic strategy selection using this paradigm showed that children use the strategies available to them flexibly but not adaptively (e.g., Hickendorff, 2020; Torbeyns, De Smedt, Ghesquière et al., 2009; Torbeyns & Verschaffel, 2013, 2016; Torbeyns et al., 2005). This is because although children could use each strategy fluently and with high accuracy in either choice or no-choice conditions, in the choice condition they did not adapt their strategy according to the characteristics of the problem. Similar results were found in studies using other paradigms than the choice/no-choice, e.g., multiplication solving by estimation (Lemaire & Brun, 2016).

The lack of adaptivity was also found in high-achiever children (e.g., Star & Newton, 2009; Torbeyns et al., 2005) and adult participants (e.g., Dewi et al., 2015; Torbeyns, Ghesquiere et al., 2009). Although this may indicate that strategy choice does not correspond to expertise, Torbeyns, De Smedt, Ghesquiere et al. (2009) revealed that high-achiever students were able to find more adaptive strategies by themselves. Instead, the non-adaptivity observed in adults and some high-achiever children might result from the use of one single strategy to solve problems with different characteristics throughout the course of learning. This is typically what occurs in the so-called routine or traditional approach of learning, according to which the goal of instruction is the mastery of computational skills, often to the detriment of understanding (Baroody, 2003; Heinze et al., 2009). Nevertheless, intervention studies in secondary-school algebra revealed that the lack of problem-based adaptivity could be improved by introducing multiple strategies at early learning stage (Acevedo Nistal et al., 2014; Rittle-Johnson et al., 2009; Rittle-Johnson et al., 2012). Such an intervention is important considering that Star and Rittle-Johnson (2008) found that higher level of adaptivity led to more successful transfer to new situations.

Adaptivity, however, does not only depend on knowledge of strategies but also on conceptual knowledge. This was revealed by studies in the domains of arithmetic (e.g., Gaschler et al., 2013; Godau et al., 2014; McMullen et al., 2017; McMullen et al., 2016) and algebra (e.g., Rittle-Johnson & Star, 2009; M. Schneider et al., 2011). In the domain of arithmetic, for example, adaptivity could be tested by means of 3-term arithmetic problems, that could be solved by inversion shortcut (e.g., $7 + 3 - 3 = 7 + (3 - 3)$; Klein & Bisanz, 2000; Robinson & Dubé, 2012), associativity shortcut (e.g., $27 + 5 - 24 = (27 - 24) + 5$, Robinson & Dubé, 2012), ten-strategy shortcut ($4 + 5 + 6 = (4 + 6) + 5$; Gaschler et al., 2013; Godau et al., 2014), or make-a-round-number shortcut ($43 - 5 + 7 = (43 + 7) - 5$; McMullen et al., 2016). The spontaneous use of these shortcuts requires not only knowledge of strategies (i.e., the ability to solve basic arithmetic problems with different strategies) but also conceptual knowledge (e.g., order of operations, commutativity and associativity principles of addition and multiplication, or the inversion between addition and subtraction or between multiplication and division).

Most of the studies on strategy adaptivity found in the literature, such as described in the preceding paragraph, are related to selecting strategies based on the characteristics of the problems. There are not many studies dedicated to person-based adaptivity, which is the type of adaptivity revealed in this thesis. As argued earlier, this might be related to the fact that maths instructions are often based on the traditional approach, with the goal that children master computational skills, often to the detriment of understanding (Baroody, 2003; Heinze et al., 2009). In arithmetic learning, this is often translated into memorising practically all basic arithmetic facts (e.g., Baroody, 2003). The influence of routine approach could be illustrated by recent finding by Hopkins et al. (2020) that about 30% of third- and fourth-graders were accurate min-counters, i.e., they used min strategy with high accuracy to solve simple additions but their accuracy in retrieval was very low. With this result, the latter authors cautioned educators to be aware of the existence of this group of children and to provide better instruction for such children so that they could be efficient retrievers. However, other researchers (e.g., Brownell & Chazal, 1935; Thornton, 1978) warned against the drill for mastery of arithmetic facts without children understanding the reasoning behind these facts.

In contrast to the routine or traditional approach, the reform-based approach advocates the development of multiple strategies. Reform-based approach is in line with Baroody's schema-based view of arithmetic learning (Baroody, 2003; Baroody & Ginsburg, 1986; Baroody & Tiilikainen, 2003). In the domain of basic arithmetic, this educational approach advocates the development of not only the retrieval from long-term memory but also conceptual knowledge (i.e. relations between number combinations such as commutativity and associative principles, the use of rules such as $N \times 1 = N$, $N + 0 = N$, and $N - N = 0$, and other reconstructive processes such as counting and reasoning).

Nevertheless, although proponents of the reform-based approaches in arithmetic instruction agree that children should learn different strategies, there is a debate among educators as to how these different strategies should be taught. Verschaffel et al. (2009) contrasted the way large additions, i.e., with a sum superior to 10, are taught in Flemish Belgium with how these problems are taught in Germany. In fact, large additions, e.g., 7 + 8, could be solved efficiently by retrieval, decomposition-to-10 (i.e., 7 + 8 = (7 + 3) + 5), or tie (i.e., 7 + 8 = (7 + 7) + 1) strategies (e.g., Torbeyns et al., 2005). In Flemish Belgium, children are taught explicitly and systematically the set of problems for which a given strategy is more appropriate, i.e., retrieval strategy for tie problems, tie strategy for near-tie problems, and decomposition for other problems. Thus, the choice of strategy is adapted to the problem characteristics, i.e., problem-based adaptivity. In Germany, oppositely, children are also taught the 3 strategies, but they are free to select the strategy that suite them. In this case, the choice of strategy is adapted to the person, i.e., person-based adaptivity.

Personally, as a maths teacher, I would opt for reform-based approach and would let children free to select the strategy they are comfortable with. I think it is possible that low-achievers are weak in maths because they do not have the chance to explore different strategies. In other words, they are forced to use the strategy that is not suitable for them. In fact, according to Verschaffel et al. (2009), low-achiever children profit more from reform-based than from traditional approach. However, in order that children could master different strategies and acquire strategy flexibility, teachers should themselves be aware of the whole range of strategies and should not impose one particular strategy even though it is considered to be the most efficient one, i.e., the one that is supposed to guarantee fast solution with high accuracy. It is my personal opinion that the goal of maths instruction should be that children are at ease with maths, even if it means postponing the acquisition of the most sophisticated knowledge.

My preference for reform-based approach of instruction was based on the findings of this thesis that participants' strategy choice corresponded more to person- than problem-based adaptivity. The superiority of reform-based over traditional approach was indeed shown by, e.g., Fuchs et al. (2010) and Tournaki (2003) who found that learning grounded on drill only is not as effective as when the drill was accompanied by strategy instruction. Nevertheless, as I will describe later, because this thesis was based on artificial tasks completed in laboratory, the results may not be readily applicable to everyday-life learning settings, and the discussion about the implications of the results from this thesis to the domain of learning and instruction may be far-fetch.

6.2 Limitations

The results of this thesis revealed that the shift from retrieval to counting as purported by the instance theory of automatization (Logan, 1988) is not the default progression of learning. In fact, the number of participants who became retrievers in our Experiments 1 and 3 was slightly more than one fourth of the whole sample (see Table 6.2). However, it is possible that the length of the experiment or, more precisely, the number of repetitions, was not large enough to guarantee an automatization, whether it concerns retrieval or counting procedure. In fact, with the ACT–R theory, Lebiere and Anderson (1998) simulated the learning of arithmetic to reproduce the performance of children and adults in the study of Ashcraft (1987). They found that adults' performance can be attained after solving 250'000 simple-addition problems. As a comparison, in our longest experiments, i.e., Experiment 1 that was run over 25 sessions, the participants solved 12'000 problems, not at all comparable to 250'000.

With this high number of 250'000 problems, we could question whether the performance of retrievers found in this thesis corresponds to automated retrieval. As a comparison, the mean solution times for retrievers in the last learning session of Experiment 3 were about 1000 ms (see Figure 5.5), higher than solution times for small additions in adults, which were under 770 ms (e.g., Barrouillet & Thevenot, 2013). In the framework of the instance theory of automatization, automatisation is analogous to retrieval. In contrast, according to the ACT–R theory (e.g., Tenison & Anderson, 2016), automatisation corresponds to an autonomous performance which is the third and last phase of learning, where a stimulus automatically triggers a response without having to activate declarative memory. In the framework of this theory, the performance of retrievers in this thesis may correspond to the associative stage of learning, i.e., the second stage, where the presentation of a problem would activate retrieval from declarative memory.

On the other hand, we could also question whether an asymptotic performance corresponds necessarily to an automatisation. Logan and Klapp (1991) as well as Zbrodoff (1995) found that participants' solution times reached an asymptote after 48 trials. As described earlier, addend slope for non-breakers in Experiment 2 (see Chapter 5) reached an asymptote at around 300 ms/addend. Likewise, in our Experiment 1 (see Chapter 2), we found no difference in addend slopes between Sessions 12 and 25, indicating an asymptotic performance. However, addend slope continued to decrease from Sessions 12 to 25 and it is possible had we continued the experiment, addend slope would decrease even more. Similarly, it is possible that the mean solution times for retrievers in Session 12 of Experiment 3 would continue to decrease and reach the 750 ms obtained in mental additions.

The question of what would happen if we extended the training experiment is indeed an interesting one. As argued in the previous paragraph, there are reasons to think that solution times and addend slopes would continue to decrease. But that about the used strategy? Would all breakers become retrievers in the long run, or would they continue to use counting to solve smaller problems? Would the non-breakers accelerate their counting speed, or would they start to memorise certain problems? And what about participants in the unclassified group, would they turn into stable breakers?

We could also question what would happen if we include addends higher than 6. For example, if we use addends 2 to 8, would participants count to solve +7 problems, or would there be a limit of addend, above which participants would attempt to memorise? So far, studies that included large addends, i.e., 7 and 9, used them as discrete addends – i.e., 3, 5, or 7 in the study of Haider and Frensch (2002) and 7 or 9 in the works of Rickard (1997, 2004) – instead of as a part of continuous addends, e.g., from 2 to 7 or from 2 to 9. As described in the Introduction, this kind of material construction tend to lead to the use of retrieval.

Another limitation of this thesis concerns individual differences, on which a large part of this Discussion was based. More precisely, instead of a default progression from counting to retrieval, the evolution of counting procedure in the course of learning depends on the participants. Nevertheless, the number of participants recruited in our studies was undeniably too low to allow for meaningful statistical analyses at the group level. For example, the highest number of participants we had was 7 in the break groups of Experiments 2 and 3 (see Table 6.2). Thus, the results concerning individual differences should be considered as exploratory. Because the proportion of participants in each group was not stable across experiments, it is difficult to estimate a priori the required sample size.

Concerning the inferred strategy used by participants, it should also be noted that it is assumed that participants used either counting or retrieval. In fact, it is possible that recognition, letter-after and skip-letter-after rules, or derived fact were also implemented. Indeed, participants who showed the M pattern might have used derived fact. For example, the memorised facts for +3 problems could be used to solve +4 problems. Besides the strategies mentioned earlier, it is also possible that participants attempted first to retrieve but then counted either to verify the retrieved answer or because of retrieval failure. Indeed, one participant informally reported this use of strategy. It should also be noted that contrary to children learning addition, adults possess a larger palette of learning strategies. We have excluded 3 participants in Experiment 2 as well as 2 participants in Experiment 3 because their addend slopes at the beginning of training were already non-significant. In fact, one participant in Experiment 2 (a verification task) informally reported of having translated the letter augends into numbers, solved the problems numerically, and translated the resulted numbers back into letter answers. This use of strategy was more obvious in Experiment 3 (a production task), where some participants uttered, for example, “4” instead of “D” as the answer to B + 2. Naturally we could have collected the data about the used strategy trial by trial by means of verbal report. Although this method is not always reliable (Ericsson & Simon, 1980; Kirk & Ashcraft, 2001; Lucidi & Thevenot, 2014; Smith-Chant & LeFevre, 2003; Thevenot et al., 2010), an interview at the very end of the experiment would still be informative.

Another caveat to consider is that it is not clear how the results of this thesis is generalisable to arithmetic learning or learning in other domains. Zbrodoff (1995) made use of alphabet-arithmetic experiments to reproduce the problem-size effect in mental addition and revealed the importance of both interference and frequency in problem-size effect. However, her study did not include the largest-addend effect revealed in this thesis. Furthermore, alphabet-arithmetic paradigm does not allow us to study the progression from counting all to min strategy (e.g., Baroody, 1987) or from counting by 1 step to counting by 2 or more steps (e.g., Svenson & Hedenborg, 1980; Svenson et al., 1976), the difference between tie and non-tie problems (e.g., Groen & Parkman, 1972), or the inverse relation between addition and subtraction (e.g., Vilette, 2002).

Finally, although this thesis invalidates the instance theory of automatization, it only concerns alphabet-arithmetic tasks. As described in the Introduction, the instance theory of automatization has also been used to illustrate the automatisation of reading by means of the lexical decision task, i.e., to judge whether or not a string of 4 letters is an English word (Grant & Logan, 1993; Logan, 1988, 1990, 1997) and the automatisation of counting by means of the dot-counting task (Lassaline & Logan, 1993; Logan, 1992). On the other hand, instance theories could also be applied to learning that does not start with algorithm-based performance, for example purely associative learning (e.g., Jamieson et al., 2012) or purely procedural learning (e.g., Logan, 2018). All in all, this thesis reveals instead the complexity of modelling the progression from algorithm-based learning that includes top-down mechanisms and two eventualities, namely retrieval-based performance or accelerated algorithm-based performance in experts. It is nevertheless not clear whether strength theories such as ACT–R could take up this challenge.

6.3 Future Directions and Recommendations

Individual Differences

A large part of this Discussion was based on the results about individual differences. However, we did not have information as to the origin of these differences. In future researches, it would be interesting to investigate factors underlying individual differences. In the domain of arithmetic, cognitive capacities have been recognised as the dominant factor. For example, individuals with faster processing speed or higher working-memory capacity use retrievals with higher frequency and have higher accuracy in both retrieval and counting trials than individuals with slower processing speed or lower working-memory capacity. This result was found in addition (Imbo & Vandierendonck, 2007a), subtraction (Imbo & Vandierendonck, 2007c) as well as in multiplication and division (Imbo & Vandierendonck, 2007b), and in children like in adults (Imbo & Vandierendonck, 2008).

However, whereas working-memory capacity does not determine strategy choice, it is important for strategy execution. More precisely, disrupting central executive and phonological aspects of working memory increase the error rate in counting procedure (e.g., Hecht, 2002; Imbo & Vandierendonck, 2007c). Nevertheless, Geary and collaborators found that working-memory capacity is also important for strategy choice. More precisely, mathematically-disabled children, who tend to have lower working-memory capacity, are inclined to use less mature strategy, and are more prone to error in their use of both counting and retrieval (Geary et al., 1991; Geary et al., 2012; Geary et al., 2004; Geary et al., 1987). On the opposite side of the spectrum, gifted and intellectually precocious children tend to have a more mature mix of strategy choices (Geary & Brown, 1991; Hoard et al., 2008). Note that, as proponents of retrieval models, Geary and collaborators considered retrieval to be the mature strategy.

Nevertheless, strategy choice may also be disconnected from cognitive aspects, as shown by Siegler and collaborators (Kerkman & Siegler, 1993, 1997; Siegler, 1988). By means of cluster analysis on the performance in simple additions, these authors found 3 groups of students. The children in the first group, who used retrieval very rarely and committed high number of errors in the use of both retrieval and procedural strategies, were classed as not-so-good students. The children in the second group, who used retrieval very frequently and had high accuracy rates in the use of both retrieval and procedural strategies, were classed as good students. The children in the last group were intriguing because the frequency of retrieval use was even lower than children in the not-so-good group but the accuracy in both retrieval and procedural strategies was even higher than children in the good group. Siegler and collaborators named them the perfectionists. Indeed, it is likely that these perfectionist students opted to use procedural strategies most of the time, because these strategies guarantee high accuracy, and used retrieval only for problems for which they are sure of knowing the answer. This phenomenon, that is also found in adults (Hecht, 2006), shows that personality may likewise come into play in the strategy choice.

As far as alphabet-arithmetic paradigm is concerned, the only study involving cognitive measures was that of Brigman and Cherry (2002) who measured working memory and processing speed. They found that working memory, but not processing speed, was correlated with solution times at the end of training.

However, Brigman and Cherry did not include individual differences in strategy choice in their study. Thus, in the future it would be informative to administer a battery of cognitive tests to investigate the influence of cognitive aspects on the strategy choice and strategy evolution in alphabet-arithmetic training experiments. There are other factors that may affect strategy choice in arithmetic. For example, a large body of research was dedicated in comparing the arithmetic strategy choices between individuals from China and from Western countries (e.g., Campbell & Xue, 2001; Geary et al., 1993; Geary et al., 1996; Geary et al., 1992). This aspect is however not pertinent for studies using the alphabet-arithmetic paradigm that is artificial by design.

Automatization of retrieval

I have explained that retrievers, due to their non-significant addend slopes when problems with the largest addend were included, must have used retrievals for all addends. However, despite non-significant addend slopes, visual inspection shows that solution times were not always equal for all problems. This indicates that although retrieval was used, it was not automatised yet. For the breakers, on the other hand, although their addend slopes when problems with the largest addend were excluded were significant, I have argued that mixed strategies instead of exclusive counting might have been used for problems with smaller addends. This is because transfer was not obtained in the first transfer session. In future research, to test whether a problem has reached retrieval automatisation, a dual-task experiment could be utilised (e.g., Beilock & Carr, 2001; Kramer et al., 1995). By comparing the performance in a single-task to the performance in a dual-task for each problem, we could infer which problems have reached retrieval automatisation and which ones have not been. More precisely, compared to the single-task condition, retrieval-automatised problems in the dual-task condition should suffer less interference from the secondary task than non-retrieval-automatised problems.

Learning curve

Whereas the 2 research projects proposed earlier would require new training experiments, the following project could be carried out by making use of the data collected for this thesis. It concerns the last argument put forward by Logan and Klapp (1991) about the support for the shift in strategy from counting to retrieval, i.e., the power-function law of learning. In accordance to the third quantitative property of the instance theory of automatization (Logan, 1988), Logan and Klapp (1991) found that both the mean and the standard deviation of solution times across trials followed a power function with the same learning rate. However, as detailed in Appendix 1.A, the appropriateness of a single learning curve (e.g., Delaney et al., 1998; Haider & Frensch, 2002; Rickard, 1997, 1999, 2004; Tenison & Anderson, 2016) and a power function to model a learning curve (e.g., R. B. Anderson & Tweney, 1997; Heathcote et al., 2000; Myung et al., 2000) have been disputed. This issue becomes more important in our alphabet-arithmetic experiments, considering the revealed individual differences.

Therefore, in future research it would be interesting to scrutinise this learning-curve problem in light of these individual differences. More precisely, I would investigate the differences in learning-curve parameters that characterise, in the first place, the learning curves of retrievers, breakers, and non-breakers and, in the second place, the learning curves of different addends in these different groups. It is also worth studying whether a model with two or more learning curves would provide a better fit than a model with one learning curve (e.g., Delaney et al., 1998; Haider & Frensch, 2002; Logan, 1988; Tenison & Anderson, 2016; White et al., 2007). This is particularly important for problems that are solved by retrieval or mixed strategies towards the end of practice. To this aim, the hidden Markov model (e.g., Tenison & Anderson, 2016) would be a suitable method to use, because the performance for each participant and each problem could be analysed individually. With this method, we could model the number of learning states as well as the learning curve function to represent the speedup during the learning states.

Bibliography

Acevedo Nistal, A., Van Dooren, W., Clarebout, G., Elen, J., & Verschaffel, L. (2009). Conceptualising, investigating and stimulating representational flexibility in mathematical problem solving and learning: A critical review. *ZDM Mathematics Education*, *41*(5), 627–636. https://doi.org/10.1007/s11858-009-0189-1

Acevedo Nistal, A., Van Dooren, W., & Verschaffel, L. (2014). Improving students' representational flexibility in linear-function problems: An intervention. *Educational Psychology*, *34*(6), 763–786. https://doi.org/10.1080/01443410.2013.785064

Aiken, L. R. (1971). Mental mechanisms in performing elementary numerical operations. *Perceptual and Motor Skills*, *33*(2), 463–465. https://doi.org/10.2466/pms.1971.33.2.463

Aiken, L. R., & Williams, E. N. (1973). Response-times in adding and multiplying single-digit numbers. *Perceptual and Motor Skills*, *37*(1), 3–13. https://doi.org/10.2466/pms.1973.37.1.3

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review*, *89*(4), 369–406. https://doi.org/10.1037/0033-295X.89.4.369

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem situations. *Psychological Review*, *94*(2), 192–210. https://doi.org/10.1037/0033-295X.94.2.192

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060. https://doi.org/10.1037/0033-295x.111.4.1036

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Lawrence Erlbaum.

Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, *128*(2), 186–197. https://doi.org/10.1037/0096-3445.128.2.186

Anderson, J. R., & Schunn, C. D. (2000). Implications of the ACT-R learning theory: No magic bullets. In R. Glaser (Ed.), *Advances in instructional psychology: Educational design and cognitive science* (pp. 1–33). Lawrence Erlbaum.

Anderson, R. B., & Tweney, R. D. (1997). Artifactual power curves in forgetting. *Memory & Cognition*, *25*(5), 724–730. https://doi.org/10.3758/BF03211315

Ashcraft, M. H. (1982). The development of mental arithmetic – A chronometric approach. *Developmental Review*, *2*(3), 213–236. https://doi.org/10.1016/0273-2297(82)90012-0

Ashcraft, M. H. (1987). Children's knowledge of simple arithmetic: A developmental model and simulation. In J. Bisanz, C. J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: progress in cognitive development research* (pp. 302–338). Springer. https://doi.org/10.1007/978-1-4612-4694-7_9

Ashcraft, M. H. (1992). Cognitive arithmetic – A review of data and theory. *Cognition*, *44*(1–2), 75–106. https://doi.org/10.1016/0010-0277(92)90051-I

Ashcraft, M. H., & Battaglia, J. (1978). Cognitive arithmetic – Evidence for retrieval and decision-processes in mental addition. *Journal of Experi-*

mental Psychology: Human Learning and Memory, *4*(5), 527–538. https://doi.org/10.1037/0278-7393.4.5.527

Ashcraft, M. H., & Christy, K. S. (1995). The frequency of arithmetic facts in elementary texts – Addition and multiplication in grades 1–6. *Journal for Research in Mathematics Education*, *26*(5), 396–421. https://doi.org/10.2307/749430

Ashcraft, M. H., & Fierman, B. A. (1982). Mental addition in 3rd, 4th and 6th graders. *Journal of Experimental Child Psychology*, *33*(2), 216–234. https://doi.org/10.1016/0022-0965(82)90017-0

Ashcraft, M. H., Fierman, B. A., & Bartolotta, R. (1984). The production and verification tasks in mental addition – An empirical comparison. *Developmental Review*, *4*(2), 157–170. https://doi.org/10.1016/0273-2297(84)90005-4

Ashcraft, M. H., & Guillaume, M. M. (2009). Mathematical cognition and the problem size effect. In B. H. Ross (Ed.), *The psychology of learning and motivation* (pp. 121–151). Academic Press. https://doi.org/https://doi.org/10.1016/S0079-7421(09)51004-3

Ashcraft, M. H., & Stazyk, E. H. (1981). Mental addition – A test of three verification models. *Memory & Cognition*, *9*(2), 185–196. https://doi.org/10.3758/BF03202334

Avancini, C., Galfano, G., & Szücs, D. (2014). Dissociation between arithmetic relatedness and distance effects is modulated by task properties: An ERP study comparing explicit vs. implicit arithmetic processing. *Biological Psychology*, *103*, 305–316. https://doi.org/10.1016/j.biopsycho.2014.10.003

Bagnoud, J., Dewi, J., Castel, C., Mathieu, R., & Thevenot, C. (2021). Developmental changes in size effects for simple tie and non-tie addition problems in 6-to 12-year-old children and adults. *Journal of Experimental Child Psychology*, *201*, 104987. https://doi.org/10.1016/j.jecp.2020.104987

Bagnoud, J., Dewi, J., & Thevenot, C. (2021). Differences in event-related potential (ERP) responses to small tie, non-tie and 1-problems in addition and multiplication. *Neuropsychologia*, *153*, 107771. https://doi.org/10.1016/j.neuropsychologia.2021.107771

Baroody, A. J. (1983). The development of procedural knowledge – An alternative explanation for chronometric trends of mental arithmetic. *Developmental Review*, *3*(2), 225–230. https://doi.org/10.1016/0273-2297(83)90031-X

Baroody, A. J. (1984). A reexamination of mental arithmetic models and data – A reply to Ashcraft. *Developmental Review*, *4*(2), 148–156. https://doi.org/10.1016/0273-2297(84)90004-2

Baroody, A. J. (1985). Mastery of basic number combinations: Internalization of relationships or facts? *Journal for Research in Mathematics Education*, *16*(2), 83–98. https://doi.org/10.5951/jresematheduc.16.2.0083

Baroody, A. J. (1987). The development of counting strategies for single-digit addition. *Journal for Research in Mathematics Education*, *18*(2), 141–157. https://doi.org/10.2307/749248

Baroody, A. J. (1994). An evaluation of evidence supporting fact-retrieval models. *Learning and Individual Differences*, *6*(1), 1–36. https://doi.org/10.1016/1041-6080(94)90013-2

Baroody, A. J. (1995). The role of the number-after rule in the invention of computational shortcuts. *Cognition and Instruction*, *13*(2), 189–219. https://doi.org/10.1207/s1532690xci1302_2

Baroody, A. J. (2003). The development of adaptive expertise and flexibility: The integration of conceptual and procedural knowledge. In A. J. Baroody & A. Dowker (Eds.), *The development of arithmetic concepts and skills: Constructive adaptive expertise* (pp. 1–33). Routledge.

Baroody, A. J. (2006). Why children have difficulties mastering the basic number combinations and how to help them. *Teaching Children Mathematics*, *13*(1), 22–31. https://doi.org/10.5951/TCM.13.1.0022

Baroody, A. J. (2018). A commentary on Chen and Campbell (2017): Is there a clear case for addition fact recall? *Psychonomic Bulletin & Review*, *25*(6), 2398–2405. https://doi.org/10.3758/s13423-018-1440-y

Baroody, A. J., Bajwa, N. P., & Eiland, M. (2009). Why can't Johnny remember the basic facts? *Developmental Disabilities Research Reviews*, *15*(1), 69–79. https://doi.org/10.1002/ddrr.45

Baroody, A. J., Eiland, M. D., Purpura, D. J., & Reid, E. E. (2012). Fostering at-risk kindergarten children's number sense. *Cognition and Instruction*, *30*(4), 435–470. https://doi.org/10.1080/07370008.2012.720152

Baroody, A. J., & Ginsburg, H. P. (1986). The relationship between initial meaningful and mechanical knowledge of arithmetic. In J. Hiebert (Ed.), *Conceptual and procedural knowledge: the case of mathematics* (pp. 75–112). Lawrence Erlbaum.

Baroody, A. J., Purpura, D. J., Eiland, M. D., Reid, E. E., & Paliwal, V. (2016). Does fostering reasoning strategies for relatively difficult basic combinations promote transfer by K-3 students? *Journal of Educational Psychology*, *108*(4), 576–591. https://doi.org/10.1037/edu0000067

Baroody, A. J., & Rosu, L. (2006). Adaptive expertise with basic addition and subtraction combinations: The number sense view. In A. J. Baroody & T. Torbeyns (Eds.), *Developing adaptive expertise in elementary school arithmetic. Symposium conducted at The annual meeting of the American Educational Research Association.*

Baroody, A. J., & Tiilikainen, S. H. (2003). Two perspective on addition development. In A. J. Baroody & A. Dowker (Eds.), *The development of* *arithmetic concepts and skills: Constructive adaptive expertise* (pp. 97–148). Routledge.

Barrouillet, P., & Thevenot, C. (2013). On the problem-size effect in small additions: Can we really discard any counting-based account? *Cognition*, *128*(1), 35–44. https://doi.org/10.1016/j.cognition.2013.02.018

Beilock, S. L., & Carr, T. H. (2001). On the fragility of skilled performance: What governs choking under pressure? *Journal of Experimental Psychology: General*, *130*(4), 701–725. https://doi.org/10.1037/0096-3445.130.4.701

Blankenberger, S. (2001). The arithmetic tie effect is mainly encoding-based. *Cognition*, *82*(1), B15–B24. https://doi.org/10.1016/S0010-0277(01)00140-8

Brigman, S., & Cherry, K. E. (2002). Age and skilled performance: Contributions of working memory and processing speed. *Brain and Cognition*, *50*(2), 242–256. https://doi.org/10.1016/S0278-2626(02)00510-9

Brownell, W. A., & Chazal, C. B. (1935). The effects of premature drill in third-grade arithmetic. *The Journal of Educational Research*, *29*, 17–28. https://doi.org/10.1080/00220671.1935.10880546

Campbell, J. I. D. (1987a). Network interference and mental multiplication. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*(1), 109–123. https://doi.org/10.1037/0278-7393.13.1.109

Campbell, J. I. D. (1987b). Production, verification, and priming of multiplication facts. *Memory & Cognition*, *15*(4), 349–364. https://doi.org/10.3758/BF03197037

Campbell, J. I. D. (1995). Mechanisms of single addition and multiplication: A modified network-interference theory and simulation. *Mathematical Cognition*, *1*(2), 121–164.

Campbell, J. I. D., Chen, Y., Allen, K., & Beech, L. (2016). Transfer of training in alphabet arithmetic. *Memory & Cognition*, *44*(8), 1288–1300. https://doi.org/10.3758/s13421-016-0631-x

Campbell, J. I. D., & Graham, D. J. (1985). Mental multiplication skill – Structure, process, and acquisition. *Canadian Journal of Psychology*, *39*(2), 338–366. https://doi.org/10.1037/h0080065

Campbell, J. I. D., & Oliphant, M. (1992). Representation and retrieval of arithmetic facts: A network-interference model and simulation. In J. I. D. Campbell (Ed.), *The nature and origin of mathematical skills* (pp. 331–364). Elsevier Science.

Campbell, J. I. D., & Xue, Q. L. (2001). Cognitive arithmetic across cultures. *Journal of Experimental Psychology: General*, *130*(2), 299–315. https://doi.org/10.1037//0096-3445.130.2.299

Carpenter, T. P., & Moser, J. M. (1984). The acquisition of addition and subtraction concepts in grades one through three. *Journal for Research in Mathematics Education*, *15*(3), 179–202.

Chen, Y., & Campbell, J. I. D. (2018). "Compacted" procedures for adults' simple addition: A review and critique of the evidence. *Psychonomic Bulletin & Review*, *25*(2), 739–753. https://doi.org/10.3758/s13423-017-1328-2

Chen, Y., Orr, A., & Campbell, J. I. D. (2020). What is learned in procedural learning? The case of alphabet arithmetic. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *46*(6), 1165–1177. https://doi.org/10.1037/xlm0000775

Chouteau, S., Mazens, K., Thevenot, C., Dewi, J., & Lemaire, B. (2021). A computational model of counting along a mental line, *43*, 2010–2016.

Clearman, J., Klinger, V., & Szucs, D. (2017). Visuospatial and verbal memory in mental arithmetic. *Quarterly Journal of Experimental Psychology*, *70*(9), 1837–1855. https://doi.org/10.1080/17470218.2016.1209534

Colonius, H. (1995). The instance theory of automaticity – Why the Weibull. *Psychological Review*, *102*(4), 744–750. https://doi.org/10.1037/0033-295X.102.4.744

Compton, B. J., & Logan, G. D. (1991). The transition from algorithm to retrieval in memory-based theories of automaticity. *Memory & Cognition*, *19*(2), 151–158. https://doi.org/10.3758/BF03197111

Cowan, N. (2000). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87–185. https://doi.org/10.1017/S0140525X01003922

De Rammelaere, S., Stuyven, E., & Vandierendonck, A. (2001). Verifying simple arithmetic sums and products: Are the phonological loop and the central executive involved? *Memory & Cognition*, *29*(2), 267–273. https://doi.org/10.3758/BF03194920

De Visscher, A., & Noël, M.-P. (2014). The detrimental effect of interference in multiplication facts storing: Typical development and individual differences. *Journal of Experimental Psychology: General*, *143*(6), 2380–2400. https://doi.org/10.1037/xge0000029

Dehaene, S. (1992). Varieties of numerical abilities. *Cognition*, *44*(1–2), 1–42. https://doi.org/10.1016/0010-0277(92)90049-N

Delaney, P. F., Reder, L. M., Staszewski, J. J., & Ritter, F. E. (1998). The strategy-specific nature of improvement: The power law applies by strategy within task. *Psychological Science*, *9*(1), 1–7. https://doi.org/10.1111/1467-9280.00001

D'Eredita, M. A., & Hoyer, W. J. (2010). Transfer of instances in cognitive skill learning: Adult age differences. *Experimental Aging Research*, *36*(1), 23–39. https://doi.org/10.1080/03610730903418646

Dewi, J. D. M., Castel, C., Kerzel, D., Posada, A., & Thevenot, C. (2015). Strategies for written additions in adults. *Journal of Cognitive Psychology*, *27*(8), 979–991. https://doi.org/10.1080/20445911.2015.1074241

Díaz-Barriga Yáñez, A., Couderc, A., Longo, L., Merchie, A., Chesnokova, H., Langlois, E., Thevenot, C., & Prado, J. (2020). Learning to run the number line: The development of attentional shifts during single-digit arithmetic. *Annals of The New York Academy of Sciences*, *1477*(1, SI), 79–90. https://doi.org/10.1111/nyas.14464

Didino, D. (2011). *A study on the representation of the arithmetic facts memory: Cognitively speaking, is the commutativity a property of multiplications and additions?* [Unpublished doctoral dissertation], University of Trento.

Dowker, A. (2014). Young children's use of derived fact strategies for addition and subtraction. *Frontiers in Human Neuroscience*, *7*, 924. https://doi.org/10.3389/fnhum.2013.00924

Dulaney, A., Vasilyeva, M., & O'Dwyer, L. (2015). Individual differences in cognitive resources and elementary school mathematics achievement: Examining the roles of storage and attention. *Learning and Individual Differences*, *37*, 55–63. https://doi.org/10.1016/j.lindif.2014.11.008

Duverne, S., & Lemaire, P. (2004). Age-related differences in arithmetic problem-verification strategies. *Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, *59*(3), P135–P142. https://doi.org/10.1093/geronb/59.3.P135

Duverne, S., & Lemaire, P. (2005). Arithmetic split effects reflect strategy selection: An adult age comparative study in addition comparison and verification tasks. *Canadian Journal of Experimental Psychology*, *59*(4), 262–278. https://doi.org/10.1037/h0087479

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, *87*(3), 215–251. https://doi.org/10.1037/0033-295X.87.3.215

Evans, N. J., Brown, S. D., Mewhort, D. J. K., & Heathcote, A. (2018). Refining the law of practice. *Psychological Review*, *125*(4), 592–605. https://doi.org/10.1037/rev0000105

Fayol, M., & Thevenot, C. (2012). The use of procedural knowledge in simple addition and subtraction problems. *Cognition*, *123*(3), 392–403. https://doi.org/10.1016/j.cognition.2012.02.008

Foos, P. W., Smith, K. H., Sabol, M. A., & Mynatt, B. T. (1976). Constructive processes in simple linear-order problems. *Journal of Experimental Psychology: Human Learning and Memory*, *2*(6), 759–766.

Forster, K. I., & Forster, J. C. (2003). A windows display program with millisecond accuracy. *Behavior Research Methods Instruments & Computers*, *35*(1), 116–124. https://doi.org/10.3758/BF03195503

Fuchs, L. S., Powell, S. R., Seethaler, P. M., Fuchs, D., Hamlett, C. L., Cirino, P. T., & Fletcher, J. M. (2010). A framework for remediating number combination deficits. *Exceptional Children*, *76*(2), 135–156. https://doi.org/10.1177/001440291007600201

Gaschler, R., Vaterrodt, B., Frensch, P. A., Eichler, A., & Haider, H. (2013). Spontaneous usage of different shortcuts based on the commutativity principle. *PLoS ONE*, *8*(9), e74972. https://doi.org/10.1371/journal.pone.0074972

Geary, D. C. (1996). The problem-size effect in mental addition: Developmental and cross-national trends. *Mathematical Cognition*, *2*(1), 63–93. https://doi.org/10.1080/135467996387543

Geary, D. C. (2011). Cognitive predictors of achievement growth in mathematics: A 5-year longitudinal study. *Developmental Psychology*, *47*(6), 1539–1552. https://doi.org/10.1037/a0025510

Geary, D. C., Bow-Thomas, C. C., Fan, L., & Siegler, R. S. (1993). Even before formal instruction, Chinese children outperform American children in mental addition. *Cognitive Development*, *8*(4), 517–529. https://doi.org/10.1016/S0885-2014(05)80007-3

Geary, D. C., Bow-Thomas, C. C., Liu, F., & Siegler, R. S. (1996). Development of arithmetical competencies in Chinese and American children: Influence of age, language, and schooling. *Child Development*, *67*(5), 2022–2044. https://doi.org/10.1111/j.1467-8624.1996.tb01841.x

Geary, D. C., & Brown, S. C. (1991). Cognitive addition – Strategy choice and speed-of-processing differences in gifted, normal, and mathematically-disabled children. *Developmental Psychology*, *27*(3), 398–406. https://doi.org/10.1037/0012-1649.27.3.398

Geary, D. C., Brown, S. C., & Samaranayake, V. A. (1991). Cognitive addition – A short longitudinal-study of strategy choice and speed-of-processing differences in normal and mathematically-disabled children. *Developmental Psychology*, *27*(5), 787–797. https://doi.org/10.1037/0012-1649.27.5.787

Geary, D. C., & Burlingham-Dubree, M. (1989). External validation of the strategy choice model for addition. *Journal of Experimental Child Psychology*, *47*(2), 175–192. https://doi.org/10.1016/0022-0965(89)90028-3

Geary, D. C., Fan, L., & Bow-Thomas, C. C. (1992). Numerical cognition – Loci of ability differences comparing children from China and the United States. *Psychological Science*, *3*(3), 180–185. https://doi.org/10.1111/j.1467-9280.1992.tb00023.x

Geary, D. C., Hoard, M. K., & Bailey, D. H. (2012). Fact retrieval deficits in low achieving children and children with mathematical learning disability. *Journal of Learning Disabilities*, *45*(4), 291–307. https://doi.org/10.1177/0022219410392046

Geary, D. C., Hoard, M. K., Byrd-Craven, J., & DeSoto, M. C. (2004). Strategy choices in simple and complex addition: Contributions of working memory and counting knowledge for children with mathematical disability. *Journal* *of Experimental Child Psychology*, *88*(2), 121–151. https://doi.org/10.1016/j.jecp.2004.03.002

Geary, D. C., Widaman, K. F., Little, T. D., & Cormier, P. (1987). Cognitive addition – Comparison of learning-disabled and academically normal elementary-school-children. *Cognitive Development*, *2*(3), 249–269. https://doi.org/10.1016/S0885-2014(87)90075-X

Geary, D. C., & Wiley, J. G. (1991). Cognitive addition – Strategy choice and speed-of-processing differences in young and elderly adults. *Psychology and Aging*, *6*(3), 474–483. https://doi.org/10.1037/0882-7974.6.3.474

Godau, C., Haider, H., Hansen, S., Schubert, T., Frensch, P. A., & Gaschler, R. (2014). Spontaneously spotting and applying shortcuts in arithmetic – A primary school perspective on expertise. *Frontiers in Psychology*, *5*, 556. https://doi.org/10.3389/fpsyg.2014.00556

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*(4), 591–635. https://doi.org/10.1016/S0364-0213(03)00031-4

Grabner, R. H., Brunner, C., Lorenz, V., Vogel, S. E., & De Smedt, B. (2021). Fact retrieval or compacted counting in arithmetic – A neurophysiological investigation of two hypotheses. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. https://doi.org/10.1037/xlm0000982

Graham, D. J., & Campbell, J. I. D. (1992). Network interference and number-fact retrieval – Evidence from children's alphaplication. *Canadian Journal of Psychology*, *46*(1), 65–91. https://doi.org/10.1037/h0084310

Grant, S. C., & Logan, G. D. (1993). The loss of repetition priming and automaticity over time as a function of degree of initial learning. *Memory & Cognition*, *21*(5), 611–618. https://doi.org/10.3758/BF03197193

Groen, G. J., & Parkman, J. M. (1972). A chronometric analysis of simple addition. *Psychological Review*, *79*(4), 329–343. https://doi.org/10.1037/h0032950

Groen, G. J., & Resnick, L. B. (1977). Can preschool-children invent addition algorithms? *Journal of Educational Psychology*, *69*(6), 645–652. https://doi.org/10.1037/0022-0663.69.6.645

Haider, H., & Frensch, P. A. (2002). Why aggregated learning follows the power law of practice when individual learning does not: Comment on Rickard (1997, 1999), Delaney et al. (1998), and Palmeri (1999). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *28*(2), 392–406. https://doi.org/10.1037//0278-7393.28.2.392

Hamann, M. S., & Ashcraft, M. H. (1985). Simple and complex mental addition across development. *Journal of Experimental Child Psychology*, *40*(1), 49–72. https://doi.org/10.1016/0022-0965(85)90065-7

Hamann, M. S., & Ashcraft, M. H. (1986). Textbook presentations of the basic addition facts. *Cognition and Instruction*, *3*(3), 173–202. https://doi.org/10.1207/s1532690xci0303_2

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, *7*(2), 185–207. https://doi.org/10.3758/BF03212979

Hecht, S. A. (1999). Individual solution processes while solving addition and multiplication math facts in adults. *Memory & Cognition*, *27*(6), 1097–1107. https://doi.org/10.3758/BF03201239

Hecht, S. A. (2002). Counting on working memory in simple arithmetic when counting is used for problem solving. *Memory & Cognition*, *30*(3), 447–455. https://doi.org/10.3758/BF03194945

Hecht, S. A. (2006). Group differences in adult simple arithmetic: Good retrievers, not-so-good retrievers, and perfectionists. *Memory & Cognition*, *34*(1), 207–216. https://doi.org/10.3758/BF03193399

Heinze, A., Star, J. R., & Verschaffel, L. (2009). Flexible and adaptive use of strategies and representations in mathematics education. *ZDM Mathematics Education*, *41*(5), 535–540. https://doi.org/10.1007/s11858-009-0214-4

Henry, V. J., & Brown, R. S. (2008). First-grade basic facts: An investigation into teaching and learning of an accelerated, high-demand memorization standard. *Journal for Research in Mathematics Education*, *39*(2), 153–183. https://doi.org/10.2307/30034895

Hickendorff, M. (2020). Fourth graders' adaptive strategy use in solving multidigit subtraction problems. *Learning and Instruction*, *67*, 101311. https://doi.org/10.1016/j.learninstruc.2020.101311

Hinault, T., Tiberghien, K., & Lemaire, P. (2016). Age-related differences in plausibility-checking strategies during arithmetic problem verification tasks. *Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, *71*(4), 613–621. https://doi.org/10.1093/geronb/gbu178

Hoard, M. K., Geary, D. C., Byrd-Craven, J., & Nugent, L. (2008). Mathematical cognition in intellectually precocious first graders. *Developmental Neuropsychology*, *33*(3), 251–276. https://doi.org/10.1080/87565640801982338

Hopkins, S., Russo, J., & Siegler, R. (2020). Is counting hindering learning? An investigation into children's proficiency with simple addition and their flexibility with mental computation strategies. *Mathematical Thinking and Learning*, 1–18. https://doi.org/10.1080/10986065.2020.1842968

Imbo, I., & Vandierendonck, A. (2007a). The development of strategy use in elementary school children: Working memory and individual differences.

Journal of Experimental Child Psychology, *96*(4), 284–309. https://doi.org/10.1016/j.jecp.2006.09.001

Imbo, I., & Vandierendonck, A. (2007b). Do multiplication and division strategies rely on executive and phonological working memory resources? *Memory & Cognition*, *35*(7), 1759–1771. https://doi.org/10.3758/BF03193508

Imbo, I., & Vandierendonck, A. (2007c). The role of phonological and executive working memory resources in simple arithmetic strategies. *European Journal of Cognitive Psychology*, *19*(6), 910–933. https://doi.org/10.1080/09541440601051571

Imbo, I., & Vandierendonck, A. (2008). Effects of problem size, operation, and working-memory span on imple-arithmetic strategies: Differences between children and adults? *Psychological Research/Psychologische Forschung*, *72*(3), 331–346. https://doi.org/10.1007/s00426-007-0112-8

Jamieson, R. K., Crump, M. J. C., & Hannah, S. D. (2012). An instance theory of associative learning. *Learning & Behavior*, *40*(1), 61–82. https://doi.org/10.3758/s13420-011-0046-2

Jerman, M. (1970). A counting model for simple addition. *Educational Studies in Mathematics*, *2*(4), 438–445.

Johnson, T. R., Wang, H., & Zhang, J. (1998). Modeling speed-up and transfer of declarative and procedural knowledge. In M. A. Gernsbacher & S. J. Derry (Eds.), *Proceedings of the twentieth annual meeting of the Cognitive Science Society* (pp. 531–536). Lawrence Erlbaum.

Johnson, T. R., Wang, H. B., & Zhang, J. J. (2000). Declarative and procedural learning in alphabetic retrieval. In L. R. Gleitman & A. K. Joshi (Eds.), *Proceedings of the twenty-second annual conference of the Cognitive Science Society* (pp. 717–722). Lawrence Erlbaum.

Kaufman, E. L., Lord, M. W., Reese, T. W., & Volkmann, J. (1949). The discrimination of visual number. *American Journal of Psychology*, *62*(4), 498–525. https://doi.org/10.2307/1418556

Kaye, D. B., Post, T. A., Hall, V. C., & Dineen, J. T. (1986). Emergence of information-retrieval strategies in numerical cognition: A developmental study. *Cognition and Instruction*, *3*(2), 127–150. https://doi.org/10.1207/s1532690xci0302_3

Kerkman, D. D., & Siegler, R. S. (1993). Individual-differences and adaptive flexibility in lower-income children's strategy choices. *Learning and Individual Differences*, *5*(2), 113–136. https://doi.org/10.1016/1041-6080(93)90008-G

Kerkman, D. D., & Siegler, R. S. (1997). Measuring individual differences in children's addition strategy choices. *Learning and Individual Differences*, *9*(1), 1–18. https://doi.org/10.1016/S1041-6080(97)90017-0

Kidd, E., & Arciuli, J. (2016). Individual differences in statistical learning predict children's comprehension of syntax. *Child Development*, *87*(1), 184–193. https://doi.org/10.1111/cdev.12461

Kirk, E. P., & Ashcraft, M. H. (2001). Telling stories: The perils and promise of using verbal reports to study math strategies. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *27*(1), 157–175. https://doi.org/10.1037/0278-7393.27.1.157

Klapp, S. T., Boches, C. A., Trabert, M. L., & Logan, G. D. (1991). Automatizing alphabet arithmetic. II. Are there practice effects after automaticity is achieved? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*(2), 196–209. https://doi.org/10.1037/0278-7393.17.2.196

Klein, J. S., & Bisanz, J. (2000). Preschoolers doing arithmetic: The concepts are willing but the working memory is weak. *Canadian Journal of Ex-*

perimental Psychology/Revue Canadienne de Psychologie Experimentale, *54*(2), 105–116. https://doi.org/10.1037/h0087333

Koerber, S., & Osterhaus, C. (2019). Individual differences in early scientific thinking: Assessment, cognitive influences, and their relevance for science learning. *Journal of Cognition and Development*, *20*(4), 510–533. https://doi.org/10.1080/15248372.2019.1620232

Kramer, A. F., Larish, J. F., & Strayer, D. L. (1995). Training for attentional control in dual-task settings – A comparison of young and old adults. *Journal of Experimental Psychology: Applied*, *1*(1), 50–76. https://doi.org/10.1037/1076-898X.1.1.50

Krueger, L. E. (1986). Why $2 \times 2 = 5$ looks so wrong – On the odd-even rule in product verification. *Memory & Cognition*, *14*(2), 141–149. https://doi.org/10.3758/BF03198374

Krueger, L. E., & Hallford, E. W. (1984). Why $2+2 = 5$ looks so wrong – On the odd-even rule in sum verification. *Memory & Cognition*, *12*(2), 171–180. https://doi.org/10.3758/BF03198431

Landauer, T. K. (1962). Rate of implicit speech. *Perceptual and Motor Skills*, *15*(3), 646–646. https://doi.org/10.2466/pms.1962.15.3.646

Lassaline, M. E., & Logan, G. D. (1993). Memory-based automaticity in the discrimination of visual numerosity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *19*(3), 561–581. https://doi.org/10.1037/0278-7393.19.3.561

Lebiere, C. (1999). The dynamics of cognition: An ACT–R model of cognitive arithmetic. *Kognitionswissenschaft*, *8*(1), 5–19. https://doi.org/10.1007/BF03354932

Lebiere, C., & Anderson, J. R. (1998). Cognitive arithmetic. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought* (pp. 297–342). Lawrence Erlbaum.

LeFevre, J.-A., Bisanz, J., Daley, K. E., Buffone, L., Greenham, S. L., & Sadesky, G. S. (1996). Multiple routes to solution of single-digit multiplication problems. *Journal of Experimental Psychology: General*, *125*(3), 284–306. https://doi.org/10.1037/0096-3445.125.3.284

LeFevre, J.-A., Sadesky, G. S., & Bisanz, J. (1996). Selection of procedures in mental addition: Reassessing the problem size effect in adults. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*(1), 216–230. https://doi.org/10.1037/0278-7393.22.1.216

LeFevre, J.-A., Shanahan, T., & DeStefano, D. (2004). The tie effect in simple arithmetic: An access-based account. *Memory & Cognition*, *32*(6), 1019–1031. https://doi.org/10.3758/BF03196878

LeFevre, J.-A., Smith-Chant, B. L., Hiscock, K., Daley, K. E., & Morris, J. (2003). Young adults' strategic choices in simple arithmetic: Implications for the development of mathematical representations. In A. J. Baroody & A. Dowker (Eds.), *The development of arithmetic concepts and skills. constructing adaptive expertise* (pp. 203–228). Lawrence Erlbaum.

Lemaire, P., & Brun, F. (2016). Age-related differences in children's strategy repetition: A study in arithmetic. *Journal of Experimental Child Psychology*, *150*, 227–240. https://doi.org/10.1016/j.jecp.2016.05.014

Lemaire, P., & Fayol, M. (1995). When plausibility judgments supersede fact retrieval – The example of the odd even effect on product verification. *Memory & Cognition*, *23*(1), 34–48. https://doi.org/10.3758/BF03210555

Lemaire, P., & Lecacheur, M. (2010). Strategy switch costs in arithmetic problem solving. *Memory & Cognition*, *38*(3), 322–332. https://doi.org/10.3758/MC.38.3.322

Lemaire, P., & Reder, L. (1999). What affects strategy selection in arithmetic? The example of parity and five effects on product verification. *Memory & Cognition*, *27*(2), 364–382. https://doi.org/10.3758/BF03211420

Lemaire, P., & Siegler, R. S. (1995). Four aspects of strategic change: Contributions to children's learning of multiplication. *Journal of Experimental Psychology: General*, *124*(1), 83–97. https://doi.org/10.1037/0096-3445.124.1.83

Lenth, R. (2019). *emmeans: estimated marginal means, aka least-squares means*. https://CRAN.R-project.org/package=emmeans

Li, M., Liu, D., Li, M., Dong, W., Huang, Y., & Chen, Q. (2018). Addition and subtraction but not multiplication and division cause shifts of spatial attention. *Frontiers in Human Neuroscience*, *12*, 183. https://doi.org/10.3389/fnhum.2018.00183

Liu, D., Cai, D., Verguts, T., & Chen, Q. (2017). The time course of spatial attention shifts in elementary arithmetic. *Scientific Reports*, *7*, 921. https://doi.org/10.1038/s41598-017-01037-3

Lochy, A., Seron, X., Delazer, M., & Butterworth, B. (2000). The odd-even effect in multiplication: Parity rule or familiarity with even numbers? *Memory & Cognition*, *28*(3), 358–365. https://doi.org/10.3758/BF03198551

Logan, G. D. (1988). Toward an instance theory of automatization. *The Psychological Review*, *95*(4), 492–527. https://doi.org/10.1037/0033-295X.95.4.492

Logan, G. D. (1990). Repetition priming and automaticity – Common underlying mechanisms? *Cognitive Psychology*, *22*(1), 1–35. https://doi.org/10.1016/0010-0285(90)90002-L

Logan, G. D. (1992). Shapes of reaction-time distributions and shapes of learning-curves – A test of the instance theory of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*(5), 883–914. https://doi.org/10.1037/0278-7393.18.5.883

Logan, G. D. (1995). The Weibull istribution, the power-law, and the instance theory of automaticity. *Psychological Review*, *102*(4), 751–756. https://doi.org/10.1037/0033-295X.102.4.751

Logan, G. D. (1997). Automaticity and reading: Perspectives from the instance theory of automatization. *Reading and Writing Quarterly*, *13*(2), 123–146.

Logan, G. D. (1998). What is learned during automatization? II. Obligatory encoding of spatial location. *Journal of Experimental Psychology: Human Perception and Performance*, *24*(6), 1720–1736. https://doi.org/10.1037/0096-1523.24.6.1720

Logan, G. D. (2018). Automatic control: How experts act without thinking. *Psychological Review*, *125*(4), 453–485. https://doi.org/10.1037/rev0000100

Logan, G. D., & Cowan, W. B. (1984). On the ability to inhibit thought and action – A theory of an act of control. *The Psychological Review*, *91*(3), 295–327. https://doi.org/10.1037/0033-295X.91.3.295

Logan, G. D., Cowan, W. B., & Davis, K. A. (1984). On the ability to inhibit simple and choice reaction-time responses – A model and a method. *Journal of Experimental Psychology: Human Perception and Performance*, *10*(2), 276–291. https://doi.org/10.1037/0096-1523.10.2.276

Logan, G. D., & Etherton, J. L. (1994). What is learned during automatization – The role of attention in constructing an instance. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*(5), 1022–1050. https://doi.org/10.1037/0278-7393.20.5.1022

Logan, G. D., & Klapp, S. T. (1991). Automatizing alphabet arithmetic. I. Is extended practice necessary to produce automaticity? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*(2), 179–195. https://doi.org/10.1037/0278-7393.17.2.179

Logan, G. D., Taylor, S. E., & Etherton, J. L. (1996). Attention in the acquisition and expression of automaticity. *Journal of Experimental Psychology:*

Learning, Memory, and Cognition, *22*(3), 620–638. https://doi.org/10.1037/0278-7393.22.3.620

Lucidi, A., & Thevenot, C. (2014). Do not count on me to imagine how I act: Behavior contradicts questionnaire responses in the assessment of finger counting habits. *Behavior Research Methods*, *46*(4), 1079–1087. https://doi.org/10.3758/s13428-014-0447-1

Luwel, K., Schillemans, V., Onghena, P. & Verschaffel, L. (2009). Does switching between strategies within the same task involve a cost? *British Journal of Psychology*, *100*(4), 753–771. https://doi.org/10.1348/000712609X402801

MacKay, K. J., & De Smedt, B. (2019). Patterning counts: Individual differences in children's calculation are uniquely predicted by sequence patterning. *Journal of Experimental Child Psychology*, *177*, 152–165. https://doi.org/10.1016/j.jecp.2018.07.016

Masse, C., & Lemaire, P. (2001). Do people combine the parity- and five-rule checking strategies in product verification? *Psychological Research/Psychologische Forschung*, *65*(1), 28–33. https://doi.org/10.1007/s004260000030

Masson, N., Letesson, C., & Pesenti, M. (2018). Time course of overt attentional shifts in mental arithmetic: Evidence from gaze metrics. *The Quarterly Journal of Experimental Psychology*, *71*(4), 1009–1019. https://doi.org/10.1080/17470218.2017.1318931

Mathieu, R., Epinat-Duclos, J., Léone, J., Fayol, M., Thevenot, C., & Prado, J. (2018). Hippocampal spatial mechanisms relate to the development of arithmetic symbol processing in children. *Developmental Cognitive Neuroscience*, *30*, 324–332. https://doi.org/10.1016/j.dcn.2017.06.001

Mathieu, R., Epinat-Duclos, J., Sigovan, M., Breton, A., Cheylus, A., Fayol, M., Thevenot, C., & Prado, J. (2018). What's behind a "+" sign? Perceiving an arithmetic operator recruits brain circuits for spatial orienting. *Cerebral Cortex*, *28*(5), 1673–1684. https://doi.org/10.1093/cercor/bhx064

Mathieu, R., Gourjon, A., Couderc, A., Thevenot, C., & Prado, J. (2016). Running the number line: Rapid shifts of attention in single-digit arithmetic. *Cognition*, *146*, 229–239. https://doi.org/10.1016/j.cognition.2015.10.002

McMullen, J., Brezovszky, B., Hannula-Sormunen, M. M., Veermans, K., Rodrìguez-Aflecht, G., Pongsakdi, N., & Lehtinen, E. (2017). Adaptive number knowledge and its relation to arithmetic and pre-algebra knowledge. *Learning and Instruction*, 178–187. https://doi.org/10.1016/j.learninstruc.2017.02.001

McMullen, J., Brezovszky, B., Rodríguez-Aflecht, G., Pongsakdi, N., Hannula-Sormunen, M. M., & Lehtinen, E. (2016). Adaptive number knowledge: Exploring the foundations of adaptivity with whole-number arithmetic. *Learning and Individual Differences*, *47*, 172–181. https://doi.org/10.1016/j.lindif.2016.02.007

Melcher, D., & Piazza, M. (2011). The role of attentional priority and saliency in determining capacity limits in enumeration and visual working memory. *PLoS One*, *6*(12), e29296. https://doi.org/10.1371/journal.pone.0029296

Miller, K. F., & Paredes, D. R. (1990). Starting to add worse – Effects of learning to multiply on childrens' addition. *Cognition*, *37*(3), 213–242. https://doi.org/10.1016/0010-0277(90)90046-M

Miller, K. F., Perlmutter, M., & Keating, D. (1984). Cognitive arithmetic: Comparison of operations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*(1), 46–60. https://doi.org/10.1037/0278-7393.10.1.46

Moyer, R. S., & Landauer, T. K. (1967). Time required for judgements of numerical inequality. *Nature*, *215*(5109), 1519. https://doi.org/10.1038/2151519a0

Mulhern, G. (1987). The development of computational skill in simple mental addition. *Journal of Structural Learning*, *9*(3–4), 317–328.

Myung, I. J., Kim, C., & Pitt, M. A. (2000). Toward an explanation of the power law artifact: Insights from response surface analysis. *Memory & Cognition*, *28*(5), 832–840. https://doi.org/10.3758/BF03198418

National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics: An overview*. National Council of Teachers of Mathematics.

National Mathematics Advisory Panel. (2008). *Foundations for success: The final report of the National Mathematics Advisory Panel*. U.S. Department of Education.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 1–55). Lawrence Erlbaum.

Okada, B. M., & Slevc, L. R. (2018). Individual differences in musical training and executive functions: A latent variable approach. *Memory & Cognition*, *46*(7), 1076–1092. https://doi.org/10.3758/s13421-018-0822-8

Parkman, J. M. (1972). Temporal aspects of simple multiplication and comparison. *Journal of Experimental Psychology*, *95*(2), 437–444. https://doi.org/10.1037/h0033662

Parkman, J. M., & Groen, G. J. (1971). Temporal aspects of simple addition and comparison. *Journal of Experimental Psychology*, *89*(2), 335–342. https://doi.org/10.1037/h0031198

Piazza, M., Fumarola, A., Chinello, A., & Melcher, D. (2011). Subitizing reflects visuo-spatial object individuation capacity. *Cognition*, *121*(1), 147–153. https://doi.org/10.1016/j.cognition.2011.05.007

Pinheiro-Chagas, P., Dotan, D., Piazza, M., & Dehaene, S. (2017). Finger tracking reveals the covert stages of mental arithmetic. *Open Mind*, *1*(1), 30–41. https://doi.org/10.1162/OPMI_a_00003

Pirolli, P. L., & Anderson, J. R. (1985). The role of practice in fact retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *11*(1), 136–153. https://doi.org/10.1037/0278-7393.11.1.136

Poletti, C., Perez, J.-F., Houillon, J.-C., Prado, J., & Thevenot, C. (2021). Priming effects of arithmetic signs in 10- to 15-year-old children. *British Journal of Developmental Psychology*. https://doi.org/10.1111/bjdp.12363

Potts, G. R., Banks, B. P., Kosslyn, S. M., Moyer, R. S., Riley, C. A., & Smith, K. H. (1978). Encoding and retrieval in comparative judgments. In N. J. Castellan Jr. & F. Restle (Eds.), *Cognitive theory* (pp. 243–308). Lawrence Erlbaum.

Protopapas, A. (2007). Checkvocal: A program to facilitate checking the accuracy and response time of vocal responses from DMDX. *Behavior Research Methods*, *39*(4), 859–862. https://doi.org/10.3758/BF03192979

Pyke, A. A., Bourque, G., & LeFevre, J.-A. (2019). Expediting arithmetic automaticity: Do inefficient computation methods induce spontaneous testing effects? *Journal of Cognitive Psychology*, *31*(1), 104–115. https://doi.org/10.1080/20445911.2018.1557664

Pyke, A. A., & LeFevre, J.-A. (2011). Calculator use need not undermine direct-access ability: The roles of retrieval, calculation, and calculator use in the acquisition of arithmetic facts. *Journal of Educational Psychology*, *103*(3), 607–616. https://doi.org/10.1037/a0023291

R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Rabinowitz, M., & Goldberg, N. (1995). Evaluating the structure-process hypothesis. In F. E. Weinert & W. Schneider (Eds.), *Memory performance and competencies: Issues in growth and development* (pp. 225–242). Lawrence Erlbaum.

Rawson, K. A., & Touron, D. R. (2015). Preservation of memory-based automaticity in reading for older adults. *Psychology and Aging*, *30*(4), 809–823. https://doi.org/10.1037/a0039652

Reder, L. M. (1982). Plausibility judgments versus fact retrieval – Alternative strategies for sentence verification. *Psychological Review*, *89*(3), 250–280. https://doi.org/10.1037/0033-295X.89.3.250

Restle, F. (1970). Speed of adding and comparing numbers. *Journal of Experimental Psychology*, *83*(2), 274–278. https://doi.org/10.1037/h0028573

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, *126*(3), 288–311. https://doi.org/10.1037/0096-3445.126.3.288

Rickard, T. C. (1999). A CMPL alternative account of practice effects in numerosity judgment tasks. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(2), 532–542. https://doi.org/10.1037/0278-7393.25.2.532

Rickard, T. C. (2004). Strategy execution in cognitive skill learning: An item-level test of candidate models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(1), 65–82. https://doi.org/10.1037/0278-7393.30.1.65

Rittle-Johnson, B., Durkin, K., & Star, J. R. (2009). The importance of prior knowledge when comparing examples: influences on conceptual and procedural knowledge of equation solving. *Journal of Educational Psychology*, *101*(4), 836–852. https://doi.org/10.1037/a0016026

Rittle-Johnson, B., & Star, J. R. (2009). Compared with what? The effects of different comparisons on conceptual knowledge and procedural flexibility for equation solving. *Journal of Educational Psychology*, *101*(3), 529–544. https://doi.org/10.1037/a0014224

Rittle-Johnson, B., Star, J. R., & Durkin, K. (2012). Developing procedural flexibility: Are novices prepared to learn from comparing procedures? *British Journal of Educational Psychology*, *82*(3), 436–455. https://doi.org/10.1111/j.2044-8279.2011.02037.x

Robinson, K. M., & Dubé, A. K. (2012). Children's use of arithmetic shortcuts: The role of attitudes in strategy choice. *Child Development Research*, *2012*, 459385. https://doi.org/10.1155/2012/459385

Schneider, M., Rittle-Johnson, B., & Star, J. R. (2011). Relations among conceptual knowledge, procedural knowledge, and procedural flexibility in two samples differing in prior knowledge. *Developmental Psychology*, *47*(6), 1525–1538. https://doi.org/10.1037/a0024997

Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information-processing. I. Detection, search, and attention. *Psychological Review*, *84*(1), 1–66. https://doi.org/10.1037/0033-295X.84.1.1

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information-processing. II. Perceptual learning, automatic attending, and a general theory. *Psychological Review*, *84*(2), 127–190. https://doi.org/10.1037/0033-295X.84.2.127

Shiffrin, R. M., & Schneider, W. (1984). Automatic and controlled processing revisited. *Psychological Review*, *91*(2), 269–276. https://doi.org/10.1037/0033-295X.91.2.269

Shrager, J., & Siegler, R. S. (1998). SCADS: A model of children's strategy choices and strategy discoveries. *Psychological Science*, *9*(5), 405–410. https://doi.org/10.1111/1467-9280.00076

Siegler, R. S. (1987). The perils of averaging data over strategies – An example from children's addition. *Journal of Experimental Psychology: General*, *116*(3), 250–264. https://doi.org/10.1037/0096-3445.116.3.250

Siegler, R. S. (1988). Individual-differences in strategy choices – Good students, not-so-good students, and perfectionists. *Child Development*, *59*(4), 833–851. https://doi.org/10.1111/j.1467-8624.1988.tb03238.x

Siegler, R. S. (1996). *Emerging minds: The process of change in children's thinking*. Oxford University Press.

Siegler, R. S., & Jenkins, E. (1989). *How children discover new strategies*. Lawrence Erlbaum.

Siegler, R. S., & Lemaire, P. (1997). Older and younger adults' strategy choices in multiplication: Testing predictions of ASCM using the choice/no-choice method. *Journal of Experimental Psychology: General*, *126*(1), 71–92. https://doi.org/10.1037/0096-3445.126.1.71

Siegler, R. S., & Robinson, M. (1982). The development of numerical understandings. *Advances in Child Development and Behavior*, *16*, 241–312. https://doi.org/10.1016/S0065-2407(08)60072-5

Siegler, R. S., & Shipley, C. (1995). Variation, selection, and cognitive change. In T. J. Simon & G. S. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling* (pp. 31–76). Lawrence Erlbaum.

Siegler, R. S., & Shrager, J. (1984). Strategic choices in addition and subtraction: How do children know what to do? In C. Sophian (Ed.), *Origins of cognitive skills* (pp. 229–293). Lawrence Erlbaum.

Singley, M. K., & Anderson, J. R. (1989). *The transfer of cognitive skill*. Harvard University Press.

Singmann, H., Bolker, B., Westfall, J., Aust, F., & Ben-Shachar, M. S. (2019). *afex: Analysis of factorial experiments*. https://CRAN.R-project.org/package=afex

Smith-Chant, B. L., & LeFevre, J.-A. (2003). Doing as they are told and telling it like it is: Self-reports in mental arithmetic. *Memory & Cognition*, *31*(4), 516–528. https://doi.org/10.3758/BF03196093

Speelman, C. P., & Kirsner, K. (2001). Predicting transfer from training performance. *Acta Psychologica*, *108*(3), 247–281. https://doi.org/10.1016/S0001-6918(01)00039-7

Star, J. R., & Newton, K. J. (2009). The nature and development of experts' strategy flexibility for solving equations. *ZDM Mathematics Education*, *41*(5), 557–567. https://doi.org/10.1007/s11858-009-0185-5

Star, J. R., & Rittle-Johnson, B. (2008). Flexibility in problem solving: The case of equation solving. *Learning and Instruction*, *18*(6), 565–579. https://doi.org/10.1016/j.learninstruc.2007.09.018

Stazyk, E. H., Ashcraft, M. H., & Hamann, M. S. (1982). A network approach to mental multiplication. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *8*(4), 320–335. https://doi.org/10.1037/0278-7393.8.4.320

Sternberg, S. (1966). High-speed scanning in human memory. *Science*, *153*(3736), 652–654. https://doi.org/10.1126/science.153.3736.652

Svenson, O. (1975). Analysis of time required by children for simple additions. *Acta Psychologica*, *39*(4), 289–302. https://doi.org/10.1016/0001-6918(75)90013-X

Svenson, O. (1985). Memory retrieval of answers of simple additions as reflected in response latencies. *Acta Psychologica*, *59*(3), 285–304. https://doi.org/10.1016/0001-6918(85)90048-4

Svenson, O., & Broquist, S. (1975). Strategies for solving simple addition problems – Comparison of normal and subnormal children. *Scandinavian Journal of Psychology*, *16*(2), 143–148. https://doi.org/10.1111/j.1467-9450.1975.tb00175.x

Svenson, O., & Hedenborg, M. L. (1980). Counting processes in simple addition. *Scandinavian Journal of Educational Research*, *24*(2), 93–104. https://doi.org/10.1080/0031383800240203

Svenson, O., Hedenborg, M. L., & Lingman, L. (1976). On children's heuristics for solving simple additions. *Scandinavian Journal of Educational Research*, *20*(1), 161–173. https://doi.org/10.1080/0031383760200111

Svenson, O., & Sjöberg, K. (1983). Evolution of cognitive processes for solving simple additions during the first three school years. *Scandinavian Journal of Psychology*, *24*(1), 117–124. https://doi.org/https://doi.org/10.1111/j.1467-9450.1983.tb00483.x

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*(3), 439–471. https://doi.org/10.1037/a0033138

Taatgen, N. A., Lebiere, C., & Anderson, J. R. (2006). Modeling paradigms in ACT–R. In R. Sun (Ed.), *Cognition and multi-agent interaction: From cognitive modeling to social simulation* (pp. 29–52). Cambridge University Press.

Taatgen, N. A., & Lee, F. J. (2003). Production compilation: A simple mechanism to model complex skill acquisition. *Human Factors*, *45*(1), 61–76. https://doi.org/10.1518/hfes.45.1.61.27224

Tenison, C., & Anderson, J. R. (2016). Modeling the distinct phases of skill acquisition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *42*(5), 749–767. https://doi.org/10.1037/xlm0000204

Thevenot, C., & Barrouillet, P. (2020). Are small additions solved by direct retrieval from memory or automated counting procedures? A rejoinder to Chen and Campbell (2018). *Psychonomic Bulletin & Review*, *27*(6), 1416–1418. https://doi.org/10.3758/s13423-020-01818-4

Thevenot, C., Barrouillet, P., Castel, C., & Uittenhove, K. (2016). Ten-year-old children strategies in mental addition: A counting model account. *Cognition*, *146*, 48–57. https://doi.org/10.1016/j.cognition.2015.09.003

Thevenot, C., Barrouillet, P., & Fayol, M. (2001). Algorithmic solution of arithmetic problems and operands-answer associations in long-term memory. *Quarterly Journal of Experimental Psychology: Human Experimental Psychology*, *54*(2), 599–611. https://doi.org/10.1080/02724980042000291

Thevenot, C., Castel, C., Fanget, M., & Fayol, M. (2010). Mental subtraction in high- and lower skilled arithmetic problem solvers: Verbal report versus operand-recognition paradigms. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *36*(5), 1242–1255. https://doi.org/10.1037/a0020447

Thorndike, E. (1911). *Animal intelligence: Experimental studies*. Macmillan.

Thornton, C. A. (1978). Emphasizing thinking strategies in basic fact instruction. *Journal for Research in Mathematics Education*, *9*(3), 214–227. https://doi.org/10.5951/jresematheduc.9.3.0214

Threlfall, J. (2009). Strategies and flexibility in mental calculation. *ZDM Mathematics Education*, *41*(5), 541–555. https://doi.org/10.1007/s11858-009-0195-3

Torbeyns, J., De Smedt, B., Ghesquiere, P., & Verschaffel, L. (2009). Acquisition and use of shortcut strategies by traditionally schooled children. *Educational Studies in Mathematics*, *71*(1), 1–17. https://doi.org/10.1007/s10649-008-9155-z

Torbeyns, J., De Smedt, B., Ghesquière, P., & Verschaffel, L. (2009). Jump or compensate? Strategy flexibility in the number domain up to 100. *ZDM Mathematics Education*, *41*(5), 581–590. https://doi.org/10.1007/s11858-009-0187-3

Torbeyns, J., Ghesquiere, P., & Verschaffel, L. (2009). Efficiency and flexibility of indirect addition in the domain of multi-digit subtraction. *Learning and Instruction*, *19*(1), 1–12. https://doi.org/10.1016/j.learninstruc.2007.12.002

Torbeyns, J., & Verschaffel, L. (2013). Efficient and flexible strategy use on multi-digit sums: A choice/no-choice study. *Research in Mathematics Education*, *15*(2), 129–140. https://doi.org/10.1080/14794802.2013.797745

Torbeyns, J., & Verschaffel, L. (2016). Mental computation or standard algorithm? Children's strategy choices on multi-digit subtractions. *European Journal of Psychology of Education*, *2*, 99–116. https://doi.org/10.1007/s10212-015-0255-8

Torbeyns, J., Verschaffel, L., & Ghesquiere, P. (2005). Simple addition strategies in a first-grade class with multiple strategy instruction. *Cognition and Instruction*, *23*(1), 1–21. https://doi.org/10.1207/s1532690xci2301_1

Tournaki, N. (2003). The differential effects of teaching addition through strategy instruction versus drill and practice to students with and without learning disabilities. *Journal of Learning Disabilities*, *36*(5), 449–458. https://doi.org/10.1177/00222194030360050601

Tuholski, S. W., Engle, R. W., & Baylis, G. C. (2001). Individual differences in working memory capacity and enumeration. *Memory & Cognition*, *29*(3), 484–492. https://doi.org/10.3758/BF03196399

Uittenhove, K., Thevenot, C., & Barrouillet, P. (2016). Fast automated counting procedures in addition problem solving: When are they used and why are they mistaken for retrieval? *Cognition*, *146*, 289–303. https://doi.org/10.1016/j.cognition.2015.10.008

Vakil, E., & Heled, E. (2016). The effect of constant versus varied training on transfer in a cognitive skill learning task: The case of the Tower of Hanoi

Puzzle. *Learning and Individual Differences*, *47*, 207–214. https://doi.org/10.1016/j.lindif.2016.02.009

Vanbinst, K., Ghesquiere, P., & De Smedt, B. (2012). Numerical magnitude representations and individual differences in children's arithmetic strategy use. *Mind, Brain, and Education*, *6*(3), 129–136. https://doi.org/10.1111/j.1751-228X.2012.01148.x

VanLehn, K. (1996). Cognitive skill acquisition. *Annual Review of Psychology*, *47*, 513–539. https://doi.org/10.1146/annurev.psych.47.1.513

Verschaffel, L., Luwel, K., Torbeyns, J., & Van Dooren, W. (2009). Conceptualizing, investigating, and enhancing adaptive expertise in elementary mathematics education. *European Journal of Psychology of Education*, *24*(3), 335–359. https://doi.org/10.1007/BF03174765

Vilette, B. (2002). Do young children grasp the inverse relationship between addition and subtraction? Evidence against early arithmetic. *Cognitive Development*, *17*(3–4), 1365–1383. https://doi.org/10.1016/S0885-2014(02)00125-9

Wagenmakers, E.-J., & Brown, S. (2007). On the linear relation between the mean and the standard deviation of response time distribution. *Psychological Review*, *114*(3), 830–841. https://doi.org/10.1037/0033-295X.114.3.830

Wenger, M. J. (1999). On the whats and hows of retrieval in the acquisition of a simple skill. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(5), 1137–1160. https://doi.org/10.1037/0278-7393.25.5.1137

White, A. S., Cerella, J., & Hoyer, W. J. (2007). Strategy transitions during cognitive skill learning in younger and older adults: Effects of interitem confusability. *Memory & Cognition*, *35*(8), 2106–2117. https://doi.org/10.3758/BF03192942

Wilkins, N. J., & Rawson, K. A. (2010). Loss of cognitive skill across delays: Constraints for theories of cognitive skill acquisition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *36*(5), 1134–1149. https://doi.org/10.1037/a0019998

Wilkins, N. J., & Rawson, K. A. (2011). Controlling retrieval during practice: Implications for memory-based theories of automaticity. *Journal of Memory and Language*, *65*(2), 208–221. https://doi.org/10.1016/j.jml.2011.03.006

Winkelman, J. H., & Schmidt, J. (1974). Associative confusions in mental arithmetic. *Journal of Experimental Psychology*, *102*(4), 734–736. https://doi.org/10.1037/h0036103

Wolfle, D. L. (1935). The relative efficiency of constant and varied stimulation during learning. *Journal of Comparative Psychology*, *19*(1), 5–27. https://doi.org/10.1037/h0060123

Zbrodoff, N. J. (1995). Why is 9 + 7 harder than 2 + 3 – Strength and interference as explanations of the problem-size effect. *Memory & Cognition*, *23*(6), 689–700. https://doi.org/10.3758/BF03200922

Zbrodoff, N. J. (1999). Effects of counting in alphabet arithmetic: Opportunistic stopping and priming of intermediate steps. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(2), 299–317. https://doi.org/10.1037/0278-7393.25.2.299

Zbrodoff, N. J. & Logan, G. D. (1990). On the relation between production and verification tasks in the psychology of simple arithmetic. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *16*(1), 83–97. https://doi.org/10.1037/0278-7393.16.1.83

Zbrodoff, N. J., & Logan, G. D. (2005). What everyone finds: The problem-size effect. In J. I. Campbell (Ed.), *The handbook of mathematical cognition* (pp. 331–345). Psychology Press.

Zhou, X., Chen, C., Zang, Y., Dong, Q., Chen, C., Qiao, S., & Gong, Q. (2007). Dissociated brain organization for single-digit addition and multiplication. *Neuroimage*, *35*(2), 871–880. https://doi.org/10.1016/j.neuroimage.2006.12.017

Zhu, R., Luo, Y., You, X., & Wang, Z. (2018). Spatial bias induced by simple addition and subtraction: From eye movement evidence. *Perception*, *47*(2), 143–157. https://doi.org/10.1177/0301006617738718

Index

This thesis was realised thanks to the developers of open-source software. The statistical analyses were run under R and RStudio. The original figures were created using the ggplot2 package of R and sometimes touched using Inkscape. The manuscript was typeset using LaTeX; the defence presentation was prepared using the beamer package under LaTeX.