Journal of Experimental Psychology:
Learning, Memory, and Cognition
1990, Vol. 16, No. 1, 83-97

0278-7393/90/$00.75

On the Relation Between Production and Verification Tasks in the Psychology of Simple Arithmetic

N. Jane Zbrodoff and Gordon D. Logan
University of Illinois

Most theories of arithmetic assume that verification tasks are performed by producing an answer and comparing it with the presented answer. Verification is production plus comparison. We tested this hypothesis by imposing delays between arithmetic arguments and answers, in theory imposing delays between production and comparison. Long delays should absorb effects on production, and reaction time, from the onset of the answer, should reflect only comparison. Six experiments were conducted, three with addition and three with multiplication. Experiments 1 and 2 used experimenter-imposed delays; Experiments 3 and 4 used subject-imposed delays. In Experiments 5 and 6, subjects uttered the sum or product before exposing the answer. In Experiments 1–4, argument magnitude affected reaction time, even at the longest delay; in Experiments 5 and 6, argument magnitude effects were reduced. These results are contrary to the hypothesis that verification is production plus comparison and consistent with the idea that verification involves comparing the equation as a whole against memory.

The psychology of simple arithmetic is based primarily on two main tasks, *production*, and *verification*. In production tasks, subjects are presented with a pair of digits and are asked to produce, usually by uttering aloud, their sum, product, or difference. For example, a subject presented with $3 \times 5 =$ would be expected to utter "fifteen." By contrast, in verification tasks, subjects are presented with an equation containing a pair of digits and their putative sum, product, or difference and are asked to indicate whether the equation is true or false. For example, a subject presented with $3 \times 5 = 17$ would be expected to respond "false." This article concerns the relation between these tasks.

In both tasks, reaction times vary substantially as the magnitudes of the digits are varied, and the patterns of variation are the primary evidence for models of the underlying representations and processes. The major competitors are counting models (Groen & Parkman, 1972), table-search models (Ashcraft & Battaglia, 1978; Geary, Widaman, & Little, 1986), and associative network models (Ashcraft, 1982, 1987; Campbell, 1987a, 1987b; Siegler, 1988). In these theories, the magnitude of the digits (*arguments*) determines the amount of computation or the difficulty of retrieval, so argument magnitude is the major independent variable. The magnitude of the answer or the difference in the magnitude of the true answer and the presented answer is often treated as a secondary independent variable whose effects reflect processes that operate after computation or retrieval. These subsequent processes are a nuisance made necessary by the requirement to measure overt behavior and are not directly relevant to the representation of arithmetic knowledge.

This approach assumes that production and verification tasks are essentially the same up to the end of the computation or retrieval stage and then begin to differ because of their response requirements. Verification involves production plus comparison. That is, subjects verify by producing the sum, product, or difference of the two digits and comparing that value to the presented answer. This perspective makes a number of predictions, some of which were tested in previous experiments and some of which are tested here.

The hypothesis that verification is production plus comparison predicts that the magnitudes of the arguments will have the same effects in the two tasks. The evidence is mixed—sometimes argument magnitude effects are the same (Ashcraft, Fierman, & Bartolotta, 1984), and sometimes they are not (Campbell, 1987b). But even if the evidence is clear, it would be hard to interpret. Argument magnitude may affect production and verification in the same way for different reasons. The various models mimic each others' predictions about argument magnitude effects (see e.g., Ashcraft & Battaglia, 1978; Miller, Perlmuter, & Keating, 1984; Zbrodoff, 1979), and it is conceivable that one process may underlie production (e.g., counting) while another underlies verification (e.g., retrieval). One cannot tell from the effects of argument magnitude.

The hypothesis also predicts, following Sternberg's (1969) additive factors logic, that factors that affect the first (computation or retrieval) stage of a verification task will not interact with factors that affect the second (comparison) stage. In particular, the magnitude of the arguments, which affects computation or retrieval, should not interact with factors that affect the comparison stage. The evidence here is mixed. On the one hand, Groen and Parkman (1972; Parkman, 1972; Parkman & Groen, 1971) found no interactions between argument magnitude and the difference between true and false answers, and Geary et al. (1986) found no interaction between argument magnitude and *split* (i.e., the difference

This research was supported by Grant BNS 88-11026 from the National Science Foundation. We would like to thank Julie Delheimer for testing the subjects. We are grateful to Mark Ashcraft, Jamie Campbell, Keith Rayner, and an anonymous reviewer for helpful comments on the article.

Correspondence concerning this article should be addressed to N. Jane Zbrodoff, Department of Psychology, University of Illinois, 603 East Daniel Street, Champaign, Illinois 61820.

between "close" and "distant" wrong answers, e.g., 3 + 4 = 9 vs. 3 + 4 = 19). But on the other hand, Ashcraft and Stazyk (1981), Stazyk, Ashcraft, and Hamann (1982), and Campbell (1987b) found interactions between problem size and split (also see the present experiments).

The interactions are troublesome. The most straightforward interpretation under the additive factors logic is that argument magnitude and split affect the same stage. This suggests that one (arithmetic) stage rather than two may underlie verification, which is inconsistent with the hypothesis that verification is production plus comparison (which implies two arithmetic stages). However, the interactions can also be interpreted as a failure of the *assumption of selective influence*, which asserts that factors affect one and only one stage. Perhaps split affects production as well as comparison. Campbell (1987b) provided some evidence consistent with this interpretation. He showed that the presentation of an answer can prime production, speeding it if the answer is correct and slowing it if the answer is incorrect. A violation of selective influence need not challenge the hypothesis that verification is production plus comparison; it weakens only arguments based on additive factors arguments. Other theories that assume two underlying stages may be able to accommodate these results (cf. McClelland, 1979; Schweickert, 1983).

The most serious challenge to the hypothesis that verification is production plus comparison comes from experiments suggesting that subjects may evaluate the equation as a whole and make their decision without computing or retrieving the true answer. For example, subjects may determine whether the answer is plausible given the arguments, therefore rejecting extreme splits very quickly (Ashcraft & Stazyk, 1981; also see Restle, 1970) or rejecting false problems quickly when most are true for the opposite operation (e.g., 3 × 4 = 7; 3 + 4 = 12; Zbrodoff & Logan, 1986). Another example is Krueger's (1986; Krueger & Hallford, 1984) demonstration of quick rejection of false problems that violate parity rules: The sum of two digits is even if both addends are even or if both addends are odd but not if one is even and one is odd (so 2 + 2 = 5 can be rejected quickly); the product of two digits is odd only if both the multiplier and the multiplicand are odd (so 2 × 2 = 5 can be rejected quickly).

These effects may reflect deliberate "side-stepping" strategies by which subjects exploit their knowledge of arithmetic and their knowledge of task constraints to avoid producing and comparing. In that case, verification performance would be a mixture of production-plus-comparison and side-stepping strategies. In principle, it should be possible to isolate trials based on production and comparison because the strategies cannot work for all equations (e.g., plausibility judgments may not discriminate true problems from near misses, false problems that are not true for other operations, or false problems that do not violate parity rules). Alternatively, the effects may reflect the use of a different retrieval mechanism, one that compares the equation as a whole against memory and evaluates the goodness of match or "resonance" instead of retrieving a true answer to compare with a presented one. In principle, such a retrieval mechanism could work for all arithmetic problems (as long as true problems match memory better than false ones), so verification performance need not involve any trials in which subjects produce and compare. It may prove difficult to distinguish a mixture of production-plus-comparison and side-stepping from resonance; to do so is beyond the scope of this article, though we offer some speculations about memory retrieval in the General Discussion. For the present, our main goal is to distinguish these two alternatives on the one hand from the possibility that verification is based only on production and comparison.

The present experiments were designed to test the hypothesis that verification is production plus comparison. The idea underlying each experiment was to impose a delay between the presentation of the arguments and the answer. If verification involves production plus comparison, then the effects of factors that affect production (i.e., computation or retrieval) should diminish as the delay increases, whereas factors that affect the subsequent comparison should have the same effects regardless of delay. The idea follows the PERT logic of Schweickert (1978, 1983) and Pashler (1984): At short delays, the comparison process must wait for computation or retrieval of the true answer to finish, and that should take longer the more difficult the computation or retrieval (i.e., the larger the magnitudes of the arguments). However, at long delays, even the most difficult computation or retrieval will have had time to finish before the answer is presented, so there would be no need for the comparison process to wait. The effects of argument magnitude should diminish and ultimately disappear as delay increases. On the other hand, the effects of the difference between the true answer and the presented answer (i.e., true vs. false and split) should be the same at all delays because the comparison process is essentially the same—subjects compare the computed or retrieved answer with the presented one.

Thus, the hypothesis that verification involves production plus comparison predicts (a) an interaction between delay and argument magnitude and (b) no interaction between delay and the split (i.e., the difference between the true answer and the presented one). Failing to find an interaction between delay and argument magnitude or finding an interaction between delay and split or both would falsify the hypothesis. Six experiments were conducted to test these predictions. Three involved addition and three involved multiplication. In two experiments (one addition and one multiplication), the delay between the arguments and the presented answer was controlled by the experimenter, and in four experiments (two addition and two multiplication) the delay was controlled by the subject.

Experiments 1 and 2

In the first two experiments, subjects were presented with two single-digit arguments and an operation symbol (addition in Experiment 1, multiplication in Experiment 2), which were followed by a putative answer at a delay randomly chosen by the experimenter. The major independent variables were argument magnitude, split, and delay. Argument magnitude has been manipulated several ways in the literature (e.g., the minimum argument, the sum of the arguments, the sum of the arguments squared) to test various models of the underlying computation or memory retrieval (see e.g., Ashcraft,

1987; Groen & Parkman, 1972). We chose to manipulate it in a theoretically neutral manner, grouping problems into three different levels: those in which both arguments were 5 or smaller, those in which both arguments were 6 or larger, and those in which one argument was 5 or smaller and the other was 6 or larger. "Tie" problems, in which the arguments were identical (e.g., 2 + 2, 3 × 3, etc.) were excluded because they often do not show argument magnitude effects (Groen & Parkman, 1972). In our *problem size* manipulation, the extreme groups did not overlap by any of the conventional measures of problem size based on argument magnitude, so all theories would predict longer reaction times for the larger problem sizes.

We chose three different levels of split: 0 (correct equations), 2, and 12. The contrast between the zero split and splits of 2 and 12 is the familiar contrast between *true* and *false* responses; the former should be faster than the latter. The contrast between splits of 2 and 12 should vary the ease of discrimination; the former should be harder and thus slower than the latter. We chose splits of 2 and 12 to maintain the parity relation between arguments and answers (Krueger, 1986; Krueger & Hallford, 1984) and to vary the difference between true and presented answers over a broad range to maximize the effect.

We used five different delays between arguments and answers: 0, 250, 500, 750, and 1,000 ms. The 0-ms delay mimicked standard verification tasks in that the arguments and answer appeared simultaneously. The 1,000-ms delay was intended to provide sufficient time for subjects to compute or retrieve the true answer before the putative answer appeared: Reaction times for production tasks with adults are typically less than 1,000 ms (Campbell, 1987a, 1987b; Campbell & Graham, 1985; Zbrodoff & Logan, 1986), and reaction times for verification tasks, which by hypothesis include production time plus the time required for comparison, typically range from 900 to 1,200 ms (Ashcraft & Battaglia, 1978; Ashcraft & Stazyk, 1981; Zbrodoff & Logan, 1986). The points between the 0- and the 1,000-ms delays were intended to capture the transition from performance based entirely on verification to performance based on production followed by comparison.

Furthermore, we conducted two control experiments in which subjects saw two numbers, one corresponding to the true sum or product of the arguments used in Experiments 1 and 2 and the other corresponding to the putative sums or products presented in Experiments 1 and 2. The design of these *number comparison* experiments was the same as the arithmetic ones—we manipulated problem size, split, and delay. One control experiment (Experiment 1a) used the numbers from the addition experiment, and one (Experiment 2a) used the numbers from the multiplication experiment. The idea was to mimic the comparison between the computed or retrieved sum or product and the presented one in the arithmetic conditions by presenting the numbers that subjects would have come up with and by having subjects compare them with the putative answers used in the experiments. If subjects in the arithmetic tasks performed by computing or retrieving and then comparing, their performance at asymptote should not differ from the performance of these control subjects.

Also, the controls were intended to deal with a necessary confound between problem size and numerical magnitude: The larger the arguments, the larger the numbers to be compared, and larger numbers may take longer to compare than smaller ones. Thus, there may be a residual effect of problem size at the longest delay even if subjects had computed or retrieved the sum or product to compare with the presented answer. The number-comparison control provides a way to assess the magnitude of this effect.

These controls require a strong version of the *assumption of pure insertion*, which underlies subtractive methods for analyzing reaction time (Donders, 1868/1969): The processes in the number comparison tasks must be identical to those in the arithmetic tasks except for the computation or retrieval of a sum or product. Any other differences between the tasks will invalidate the comparison. And other differences are likely. For example, number comparison can be performed by comparing the physical characteristics of the stimuli, independent of their meaning as numbers (cf. Posner & Mitchell, 1967). A literal physical matching strategy is not possible in the arithmetic task, although subjects may mimic physical matching by generating "images" of the correct answer to compare "physically" with the presented answer (cf. Posner & Boies, 1971). However, the contrast between tasks may still be informative if the assumption of pure insertion is relaxed a little.

We present the data and interpret them as if the assumption of pure insertion were true, but we have no strong commitment to the assumption. Conclusions from the number comparison task were usually consistent with conclusions from the arithmetic tasks, and most of the points can be made without reference to number comparison. We need not rely on the data, but we present them because they are interesting nevertheless.

Method

Subjects. Each experiment and each control condition employed a separate group of 16 subjects recruited from introductory psychology classes or the general student population. There were two experiments and two control conditions for a total of 64 subjects. Introductory psychology students received class credit for participating; other subjects received $3.50.

Apparatus and stimuli. The stimuli were displayed on IBM or Amdek monochrome monitors controlled by IBM PC/XT computers programmed to measure time in milliseconds and to synchronize timing with the raster scan of the monitors. Responses were collected on the computers' keyboards; subjects pressed the "/" key or the "\" key, which were the rightmost and leftmost keys on the bottom row of the PC/XT keyboard.

The stimuli were equations representing all combinations of the digits 1 through 9 except for ties (e.g., 3 + 3, 4 × 4, etc.). The arguments were ordered according to their magnitude in such a way that the smaller argument occupied the first (leftmost) position in the equation and the larger occupied the second (center) position. Thus, only half of the possible permutations of the arguments were used (e.g., we used 1 + 9, 2 × 9, etc., but not 9 + 1, 9 × 2). In total, there were 36 combinations of arguments.

True equations included the true sum (Experiment 1) or product (Experiment 2) of the arguments; false equations included either the true sum or product plus 2 or the true sum or product plus 12. Each

of the 36 combinations of arguments occurred twice with a split of 0 (true equations), once with a split of 2, and once with a split of 12 to equate the number of *true* and *false* responses. This resulted in a set of 144 equations, which appeared at each of the five delays for a total of 720 trials.

Three levels of problem size were distinguished: (a) problems with arguments no larger than 5, (b) problems with one argument 5 or smaller and one argument 6 or larger, and (c) problems with arguments no smaller than 6. There were 10 different combinations of arguments in the first and third level and 16 in the second level. When combined with the split manipulation, there were 20 true equations in the first and third levels and 32 true equations in the second level. There were 10 false equations at each split value in the first and third level and 16 at each split value in the second level.

The five delays were 0, 250, 500, 750, and 1,000 ms, defined relative to the onset of the arguments. Thus, in the 0-ms delay, the arguments and answer appeared simultaneously, and in the 250-ms delay, the arguments appeared 250 ms before the answer. The arguments remained on the screen throughout the delay interval, and they remained on the screen when the answer was presented. After the answer appeared, the intact equation remained on the screen until the subject responded. Then the screen went blank for a 1,000-ms intertrial interval.

A total of 720 trials were required to complete the design of the arithmetic experiments. The order of splits, problem sizes, and delays was randomized separately for each subject.

Each trial began with a 500-ms warning interval in which two lines of five dashes separated by spaces (e.g., – – – – –) were presented in the center of the screen, one line above and one line below the line on which the equation was to appear. After 500 ms elapsed, the fixation display was extinguished and replaced by the arguments for that trial, arranged so that the entire equation would be centered on the screen. The equations were displayed horizontally so that the second argument appeared in the central position in the display. Each equation included the two arguments, the relevant operation symbol (+ for Experiment 1, × for Experiment 2), an equals (=) symbol, and the putative answer. The argument display included one argument, a space, the operation symbol, a space, the second argument, a space, and the equals symbol; the answer display included the whole equation (there was a space between the equals symbol and the answer). Each equation occupied seven or eight character spaces on the screen, depending on whether the answer involved one or two digits. This corresponded to 2.3 or 2.5 cm, which corresponded to 2.2° or 2.4° of visual angle when viewed at a distance of 60 cm.

The number comparison experiments were constructed in the same way as the arithmetic experiments except that only two numbers appeared on the screen. The number on the left was the true sum (Experiment 1a) or product (Experiment 2a) of the arguments and was displayed in the same fashion as the arguments. That is, it replaced the fixation display and remained on the screen throughout the delay interval. An equals symbol appeared with the first number and remained on the screen throughout the trial. The first number and the equals sign remained on the screen when the second number appeared; the intact equation remained on until the subject responded.

In the number comparison experiments, split and delay were defined in the same way as in the arithmetic experiments. Problem size was defined in terms of the arguments that generated the left-hand number. This resulted in some overlap between the first and second levels and second and third levels of problem size (e.g., 4 + 5 = 9 from the first level and 3 + 6 = 9 from the second level; 4 + 9 = 13 from the second level and 6 + 7 = 13 from the third level). However, there was no overlap between the first and third levels. Again, 720 trials were required to complete the design, and the order of splits, problem sizes, and delays was randomized separately for each subject.

Procedure. Subjects were tested individually, one to a computer. In some cases, only 1 subject was tested at a time; in other cases, 2 subjects were tested simultaneously on separate computers facing orthogonal walls in the testing room.

The instructions began by describing the events on a trial. Subjects were told that the arguments would sometimes appear before the answer and that they should respond "true" or "false" as quickly and accurately as possible after the answer appeared. Half of the subjects pressed the "/" key for true equations and the "\" key for false equations, and half did the opposite. Subjects were told to rest the index fingers of their right and left hands on the keys throughout the experiment in order to respond as quickly as possible. The program paused every 72 trials to allow the subjects to rest if they wished to do so. When they were ready to resume, they pressed the space bar (following an instruction on the screen), and the next block began.

Design and data analysis. The addition and multiplication experiments involved a 3 (problem size) × 3 (split) × 5 (delay between problem and answer) design with repeated measures on each factor. Mean reaction times and error rates were calculated for each subject for each cell of the design, and the reaction times were subjected to analyses of variance (ANOVAs). Error rates were too low to be analyzed statistically but showed no evidence of speed–accuracy trade-offs that would compromise the interpretation of the reaction times. Our analysis of split effects does not distinguish the component due to the difference between true and false equations from the component due to the difference between near (off by 2) and distant (off by 12) false equations. It was not necessary to distinguish these effects for our purposes; both reflect the comparison stage. However, we provide the means and the MS_e terms so that interested readers can separate the effects themselves.

The number comparison experiments involved the same 3 × 3 × 5 design. They were analyzed separately in one set of ANOVAs, paralleling those used for the arithmetic studies, and they were analyzed together with the appropriate arithmetic study in another set of ANOVAs, which included task (arithmetic or number comparison) as a between-subjects factor.

Results[1]

Experiments 1 and 2. Performance was highly accurate, averaging 96.5% correct in the addition task and 95.5% correct in the multiplication task, so the analyses focused on reaction time. Accuracy correlated negatively with reaction time, $r = -.711$ for addition and $r = -.812$ for multiplication, so analyses of accuracy would be redundant with reaction time analyses.

The major experimental manipulations were successful in both experiments: Reaction time increased with problem size: For problem sizes of 1, 2 and 3, the means were 649, 702, and 693 ms in addition, $F(2, 28) = 10.11$, $p < .01$, $MS_e = 15{,}170.67$, and 672, 726, and 738 ms in multiplication, $F(2, 28) = 46.09$, $p < .01$, $MS_e = 5{,}881.21$. The consistent difference between problem sizes of 1 and 3 replicates standard results. Problem size 2 was not midway between problem sizes 1 and 3 because of the assignment of digit arguments to problem size. Problem size 1 arguments were always smaller than problem size 3 arguments; problem size 2 was constructed by combining one argument from the problem size 1 range with one from the problem size 3 range. Whether

[1] Experiment 1 involves addition; Experiment 2 involves multiplication; Experiment 1a involves number comparison using addition; Experiment 2a involves number comparison using multiplication.

problem size 2 is more like 1 or 3 depends on which of the arguments dominates retrieval.

Reaction time was influenced by the split. For splits of 0 (true), 2 (false), and 12 (false), mean reaction times were 649, 753, and 674 ms in addition, $F(2, 28) = 44.44$, $p < .01$, $MS_e = 15{,}833.92$, and 667, 757, and 756 in multiplication, $F(2, 28) = 65.87$, $p < .01$, $MS_e = 9{,}686.64$. The largest difference was between *true* and *false* equations (split of 0 vs. 2 or 12), replicating standard results. The difference between splits of 2 and 12 was significant in addition, $F(1, 28) = 47.30$, $p < .01$, but not in multiplication, $F(1, 28) < 1$.

Reaction times decreased substantially as the delay between the problem and the answer increased. For delays of 0, 250, 500, 750, and 1,000 ms, the means were 920, 705, 614, 593, and 575 ms in addition, $F(4, 56) = 330.09$, $p < .01$, $MS_e = 8{,}914.03$, and 946, 736, 650, 620, and 607 ms in multiplication, $F(4, 56) = 324.34$, $p < .01$, $MS_e = 9{,}398.04$. In both cases, the data appear to be approaching an asymptote at the longest delay, so there should have been plenty of time for production to occur, if it occurred at all. The difference between the 750- and 1,000-ms delays was not significant ($p < .05$) by Fisher's least significant difference (LSD) test in addition ($LSD = 22$ ms) or in multiplication (LSD = 23 ms). Moreover, mean reaction time at the 0-ms delay was less than 1,000 ms for both addition and multiplication. In theory, that reaction time represents the time for production plus the time for comparison, so the time for production alone must have been less than 1,000 ms. These main effects set the stage for the theoretically important analyses, the interactions among problem size, split, and delay.

The interaction between problem size and delay is presented in Figure 1. The addition data appear in the top panel, and the multiplication data appear in the bottom panel. According to the hypothesis that verification is production plus comparison, there should be a strong interaction between problem size and delay so that the problem size effect should disappear at asymptotic delays. The data provide mixed support for the hypothesis. On the one hand, both sets of data show a reduction in the problem size effect as delay increased, consistent with the hypothesis, but the problem size effect was still substantial at the 1,000-ms delay, where reaction times were at asymptote, contrary to the hypothesis. In addition, reaction times for problem sizes 1, 2, and 3 were 865, 953, and 941 ms, respectively, for the 0-ms delay and 555, 584, and 586 ms, respectively, for the 1,000-ms delay. In multiplication, reaction times for problem sizes 1, 2, and 3 were 891, 970, and 977 ms, respectively, for the 0-ms delay and 581, 609, and 631 ms, respectively, for the 1,000-ms delay. The interaction between problem size and delay was significant in addition, $F(8, 112) = 2.90$, $p < .05$, $MS_e = 4{,}411.78$, and in multiplication, $F(8, 112) = 2.57$, $p < .05$, $MS_e = 3{,}069.60$, though neither effect was very strong. The simple main effect of problem size at the 1,000-ms delay was significant in addition, $F(2, 224) = 4.60$, $p < .05$, $MS_e = 3{,}143.80$, and in multiplication, $F(2, 224) = 7.41$, $p < .01$, $MS_e = 4{,}070.73$.

The interaction between split and delay is presented in Figure 2. The addition data appear in the top panel, and the multiplication data appear in the bottom panel. According to the hypothesis that verification is production plus comparison, there should be no interaction between split and delay.

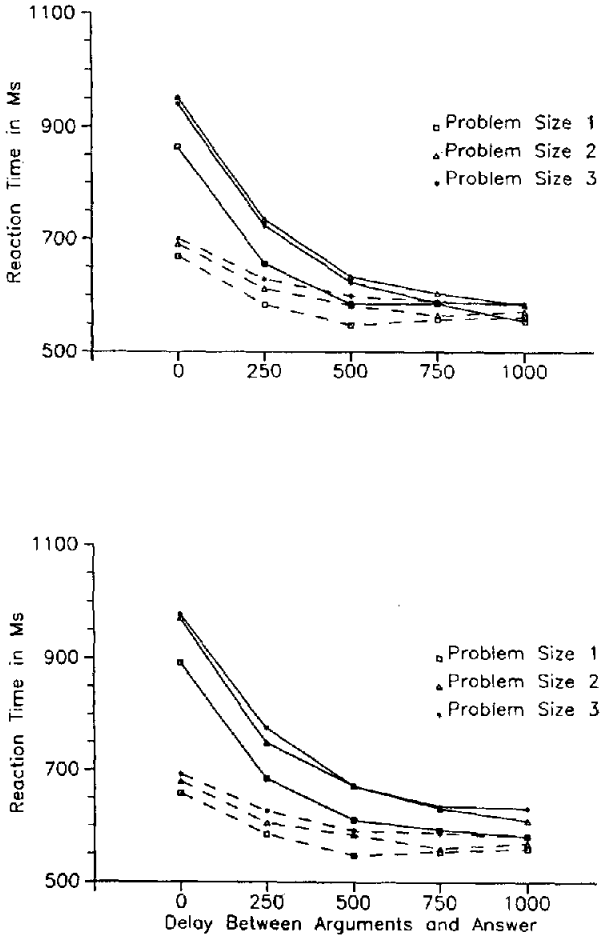

Figure 1. The effect of problem size on reaction time as a function of delay for the addition task (Experiment 1: solid lines, top panel) and the number comparison task with addition numbers (Experiment 1a: broken lines, top panel), for the multiplication task (Experiment 2: solid lines, bottom panel), and for the number comparison task with multiplication numbers (Experiment 2a: broken lines, bottom panel).

Contrary to the hypothesis, the data from both experiments show an interaction; the split effect decreased with delay. In addition, reaction times for splits of 0, 2, and 12 were 887, 1,043, and 861 ms, respectively, at the 0-ms delay and 546, 615, and 592 ms, respectively, at the 1,000-ms delay, $F(8, 112) = 12.35$, $p < .01$, $MS_e = 4{,}560.33$. In multiplication, reaction times for splits of 0, 2 and 12 were 869, 1,015, and 1,030 ms, respectively, at the 0-ms delay and 574, 641, and 640 ms, respectively, at the 1,000-ms delay, $F(8, 112) = 5.03$, $p < .01$, $MS_e = 4{,}722.98$.

The interaction between problem size and split was significant in addition, $F(2, 28) = 7.27$, $p < .01$, $MS_e = 3{,}749.04$, but not in multiplication, $F(2, 28) = 1.55$, *ns*, $MS_e = 6{,}857.33$. The interaction in addition is contrary to an additive-factors interpretation in which problem size and split affect different processing stages.

Experiments 1a and 2a. The data from the number comparison experiments are plotted (in broken lines) along with the data from the arithmetic experiments (solid lines) in

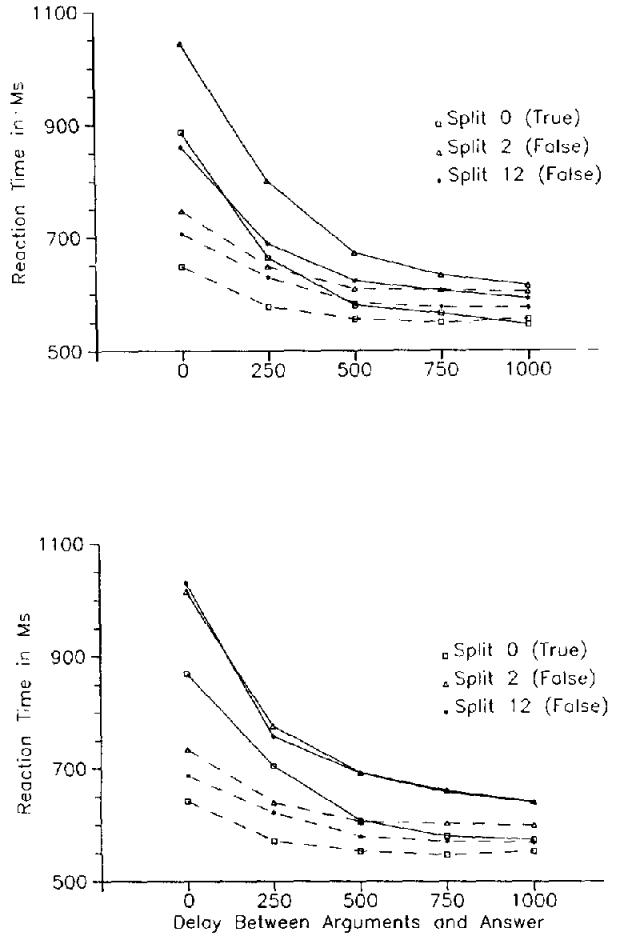

Figure 2. The effect of split on reaction time as a function of delay for the addition task (Experiment 1: solid lines, top panel) and the number comparison task with addition numbers (Experiment 1a: broken lines, top panel), for the multiplication task (Experiment 2: solid lines, bottom panel), and for the number comparison task with multiplication numbers (Experiment 2a: broken lines, bottom panel.)

Figures 1 and 2. Accuracy was high, averaging 98.5% for addition numbers and 98.7% for multiplication numbers, and was negatively correlated with reaction time, $r = -.234$ for addition numbers and $r = -.267$ for multiplication numbers. Analyses of variance on the reaction times revealed main effects of problem size [for the addition numbers, $F(2, 28) = 31.38$, $p < .01$, $MS_e = 2{,}170.53$; for the multiplication numbers, $F(2, 28) = 31.20$, $p < .01$, $MS_e = 2{,}184.40$], main effects of split [addition numbers, $F(2, 28) = 33.67$, $p < .01$, $MS_e = 7{,}044.52$; multiplication numbers, $F(2, 28) = 34.00$, $p < .01$, $MS_e = 6{,}939.12$], and main effects of delay [addition numbers, $F(4, 56) = 106.67$, $p < .01$, $MS_e = 3{,}164.03$; multiplication numbers, $F(4, 56) = 107.11$, $p < .01$, $MS_e = 3.166.40$]. The interaction between problem size and delay was not significant in either task; the interaction between split and delay was significant for addition numbers, $F(8, 112) = 2.71$, $p < .01$, $MS_e = 1{,}989.84$, and for multiplication numbers, $F(8, 112) = 2.66$, $p < .05$, $MS_e = 1{,}980.77$.

The number comparison results are interesting in contrast with the arithmetic results: According to the hypothesis that arithmetic verification is production plus comparison, arithmetic subjects ought to be comparing an internally generated number with the presented answer at the longest delay, just as number comparison subjects compare an encoded number with a presented one. Thus, reaction times should not differ between the tasks, and the effects of split and problem size should not differ between tasks. Differences at the asymptotic delay would be evidence against the hypothesis that verification is production plus comparison.

Addition reaction times converged on number comparison reaction times at the longest delay (575 vs. 573 ms, respectively), but multiplication reaction times were still longer than number comparison reaction times (607 vs. 569 ms, respectively), $F(1, 28) = 16.55$, $p < .01$, $MS_e = 6{,}282.22$. In the addition task, the split effect had the same form as the corresponding number comparison task, but it was larger, $F(2, 56) = 7.01$, $p < .01$, $MS_e = 11{,}477.24$. In multiplication, the split effect had a different form from the one observed in number comparison, $F(2, 56) = 11.48$, $p < .01$, $MS_e = 8{,}312.88$. There was no difference between splits of 2 and 12 in multiplication, whereas in number comparison (and indeed, in the arithmetic and number comparison experiments with addition), splits of 12 were responded to faster than splits of 2.

The problem size effect at the longest delay was larger in arithmetic than in number comparison for both addition and multiplication, though the difference was not significant in either task; in addition, $F(2, 224) < 1$, $MS_e = 3{,}224.22$; in multiplication, $F(2, 224) = 2.02$, $MS_e = 2{,}546.74$. One cannot accept the null hypothesis with confidence because the differences (e.g., between the largest and smallest problems) were larger in the arithmetic tasks than in number comparison. Also, the problem size effects at the 1,000-ms delay were significant in the arithmetic tasks (see the above analyses of simple main effects) but not in the number comparison tasks [for addition numbers, $F(2, 112) = 2.86$, $p < .10$, $MS_e = 2{,}036.30$; for multiplication numbers, $F(2, 112) = 2.62$, $p < .10$, $MS_e = 2{,}023.87$].

Discussion

The hypothesis that verification is production plus comparison predicts that effects of factors that affect production should be absorbed by delay as delay increases, whereas the effects of factors that affect comparison should remain constant over delay. In both experiments, problem size effects, which ought to reflect a production factor, decreased with delay as predicted, but they did not disappear as predicted at the asymptotic delay. In both experiments, split effects, which ought to reflect a comparison factor, decreased with delay, contrary to prediction.

The hypothesis that verification is production plus comparison was also tested by comparing arithmetic performance with performance on a number comparison task. The number comparison task can be interpreted as representing the comparison stage in arithmetic verification. By hypothesis, arithmetic performance at the asymptotic delay reflects only comparison, so that performance should not differ from number comparison performance at the asymptotic delay. There were many similarities between arithmetic and number compari-

son performance, but important differences were observed. Arithmetic performance approached number comparison performance in the addition task but not in multiplication. Split effects were larger in the arithmetic tasks than in number comparison, and they were qualitatively different in multiplication and number comparison. Finally, problem size effects were larger in arithmetic than in number comparison.

The bulk of the evidence seems to be against the hypothesis that verification is production plus comparison. The interaction between split and delay suggests that the presented answer influences the computation or retrieval that underlies verification, having more of an influence the earlier it appears in the process (cf. Campbell, 1987b). The failure to eliminate problem size effects suggests that the arguments continue to have an effect even when enough time has passed to have retrieved or computed their sum or product. One possibility is that the arguments and the answer act jointly as retrieval cues and that the verification decision is based on the amount of "resonance," or degree of match with memory (i.e., true equations would resonate more or match better than false equations). We will describe this idea in more detail in the General Discussion. For now, it serves as an instantiation of the view that verification performance may be based on the equation as a whole; this contrasts with the hypothesis that verification is production plus comparison, which assumes that the arguments and the answer are processed separately (in separate stages). The remaining experiments attempted to separate the processing of the arguments and the answer in order to see under what conditions the hypothesis that verification is production plus comparison may be true.

Delay effects require a different interpretation if subjects compare the equation as a whole against memory. Delay cannot absorb production time because nothing is produced. So why should reaction time decrease as delay increases? One possibility is that delay absorbs the time required to encode the digit arguments. Subjects must form an internal representation of the arguments and the operator symbol, and there is evidence that the time required to encode digit arguments contributes significantly to arithmetic verification time (Geary et al., 1986). Also, there is evidence from other paradigms that delay can absorb encoding time (Posner & Boies, 1971). The present data provide further support: The benefit from encoding can be estimated roughly as the difference between reaction times at the 0- and 1,000-ms delays. In the number comparison tasks, where only single digits are encoded, this difference was 114 ms for the addition numbers and 107 ms for the multiplication numbers. In the arithmetic tasks, where two digits and the operator must be encoded, the difference was 345 ms for addition and 339 ms for multiplication. The latter is roughly three times the former, as if about 100 ms of benefit were gained from encoding each digit or symbol. These results may not rule out alternative interpretations, but they do encourage further investigation of the hypothesis that delay absorbs encoding time.

Experiments 3 and 4

In Experiments 3 and 4, the delay between the arguments and the answer was controlled by the subject rather than the experimenter. The arguments were displayed on the screen until the subject pressed the space bar on the computer keyboard, whereupon the answer was presented. The subject then pressed one of two keys to indicate whether the equation was true or false, as in the previous experiments. We imposed a brief delay (0, 125, 250, or 500 ms) between the pressing of the space bar and the presentation of the answer in order to absorb any refractory effects of pressing the space bar and to allow subjects to become prepared for the answer. Experiment 3 used addition and Experiment 4 used multiplication. Problem size and split were manipulated in the same way as in Experiments 1 and 2.

The idea behind these experiments was to separate production and comparison in time and to measure the duration of each stage independently. The time between the onset of the arguments and the pressing of the space bar reflects the duration of the production (computation or retrieval) stage; the time between the onset of the answer and the pressing of the "\" or "/" key reflects the duration of the comparison stage. If production and comparison were in fact separated by this procedure, then variables that affect production should affect only the time to press the space bar, and variables that affect comparison should affect only the time to press an answer key. That is, problem size should affect space-bar reaction times but not answer-key reaction times, and split should affect answer-key reaction times but not space-bar reaction times.

We also ran two number-comparison control experiments in which the true sums (Experiment 3a) or products (Experiment 4a) of the arguments were compared with the putative answers to control for differences in numerical magnitude and to provide a baseline for assessing split and problem size effects in the arithmetic tasks. Subjects saw one number and an equal sign and then pressed the space bar when they were ready for the number corresponding to the answer.

Method

Subjects. Each experiment used a separate group of 16 subjects from introductory psychology or the general student body. A total of 64 subjects were tested. Introductory psychology students received course credit for participating; the others were paid $3.50.

Apparatus and stimuli. The apparatus and stimuli were the same as those used in the previous experiments. Each trial began with the 500-ms fixation display used in Experiments 1 and 2, which was extinguished and replaced by the arguments, operator symbol, and equals symbol for that trial. The arguments were displayed until the subject pressed the space bar. Then, after a delay of 0, 125, 250, or 500 ms, the answer was added to the display, and the entire equation remained on the screen until the subject responded. Again, the intertrial interval was 1,000 ms.

As in the previous experiments, factorial combination of problem size and split resulted in 144 trials, and the set of 144 was replicated at each delay between the space bar response and the onset of the answer. This resulted in a total of 576 trials. A different random order of problem size, split, and delay was constructed for each subject.

Procedure. Subjects were instructed as in the previous experiments, except that they were told that the arguments would remain on the screen until they were ready for the answer. They were told to press the space bar with the thumb of their right or left hand when they were ready for the answer and to respond to the equation as a whole by pressing the "\" or "/" key with the index fingers of their left or right hands as quickly and accurately as possible. They were

told to rest their index fingers on the response keys throughout the experiment. Half of the subjects pressed "/" for true equations and "\" for false equations, and half did the opposite.

As before, the program paused for a break every 72 trials, and subjects resumed the experiment by pressing the space bar.

Design and data analysis. The reaction time data were analyzed in several ways. Times to press the space bar were analyzed in a 3 (problem size) × 3 (split) within-subjects design, and subsequent reaction times to the complete equation were analyzed in a 3 (problem size) × 3 (split) × 4 (delay between space bar response and onset of the answer) within-subjects design. Responses to the first and second displays were compared in a 3 (problem size) × 3 (split) × 2 (first vs. second display) design, collapsing over delay for responses to the second display.

Number comparison reaction times were analyzed in a similar fashion. Number comparison reaction times were contrasted with arithmetic reaction times in a 3 (split) × 3 (problem size) within-subjects design, with tasks as a between-subjects variable. The contrasts between tasks focused on reaction times to the second display because those were the reaction times that were supposed to be equivalent under the hypothesis that verification is production plus comparison.

Results[2]

Experiments 3 and 4. Reaction times to the first and second displays are presented as a function of split and problem size in Table 1. The left panels contain data from the addition task; the right panels contain data from the multiplication task. The tables also contain accuracy data for reaction times to the second display.

Reaction times to the first display averaged 610 ms in the addition task and 639 ms in the multiplication task. First-display reaction times were affected by problem size [for addition, $F(2, 28) = 8.22$, $p < .01$, $MS_e = 7{,}566.13$; for multiplication, $F(2, 28) = 4.51$, $p < .05$, $MS_e = 25{,}840.00$] but not by split [for addition, $F(2, 28) < 1$, $MS_e = 1{,}690.68$; for multiplication, $F(2, 28) < 1$, $MS_e = 6{,}561.48$]. These effects suggest that the experiment was successful in separating production from comparison; the factor that should reflect production affected reaction time, whereas the factor that should reflect comparison did not.

Reaction times to the second display averaged 592 ms in addition and 590 ms in multiplication. These values were close to reaction times at the longest experimenter-imposed delay in Experiments 1 and 2 (575 ms for addition and 607 ms for multiplication). The similarity suggests that subjects in the present experiments were in the same state of preparation after pressing the space bar as subjects in the previous experiments were at the asymptotic delay. But did the space-bar method separate comparison from production?

Second-display reaction times were affected by split [for addition, $F(2, 28) = 17.84$, $p < .01$, $MS_e = 4{,}878.25$; for multiplication, $F(2, 28)$ 54.68, $p < .01$, $MS_e = 1{,}885.28$], as would be expected if second-display reaction times reflected only comparison. However, problem size effects were also significant [for addition, $F(2, 28) = 14.89$, $p < .01$, $MS_e = 1{,}781.11$; for multiplication, $F(2, 28) = 24.71$, $p < .01$, $MS_e = 3{,}172.18$], which is contrary to the hypothesis that second-display reaction times reflect only comparison.

In ANOVAs comparing first-display reaction times to second-display reaction times, there were significant interactions between display and split [for addition, $F(2, 28) = 9.89$, $p < .01$, $MS_e = 4{,}025.04$; for multiplication, $F(2, 28) = 12.59$, $p < .01$, $MS_e = 4{,}818.38$], indicating that split affected only second-display reaction times. The same ANOVAs showed nonsignificant interactions between display and problem size [for addition, $F(2, 28) < 1$, $MS_e = 3{,}644.33$; for multiplication, $F(2, 28) < 1$, $MS_e = 14{,}195.88$], indicating that the problem size effects were similar in magnitude for both displays, contrary to the hypothesis that second-display reaction times reflect only comparison. It appears as if subjects repeated the whole verification process when the second display appeared.

Second-display reaction times were also affected by the delay between the pressing of the space bar and the appearance of the answer. For addition, the means were 648, 600, 581, and 568 ms for the 0-, 125-, 250-, and 500-ms delays, $F(3, 42) = 36.13$, $p < .01$, $MS_e = 4{,}964.63$; for multiplication, the means were 636, 606, 585, and 586 ms, $F(3, 42) = 20.91$, $p < .01$, $MS_e = 3{,}919.10$. There were no significant interactions among delay, split, and problem size.

Experiments 3a and 4a. Number comparison reaction times are presented in Table 2. The left panel contains data from the addition numbers; the right contains data from the multiplication numbers. First-display reaction times averaged 490 ms for the addition numbers and 532 ms for the multiplication numbers. There were no significant effects of problem size or split in the first-display reaction times.

Second-display reaction times averaged 570 ms for addition numbers and 543 ms for multiplication numbers. These values are close to the values at the longest experimenter-imposed delay in Experiments 1a and 2a (573 ms for addition numbers; 569 ms for multiplication). Again, this suggests that the space-bar method can induce the same state of preparation as experimenter-imposed delays.

Second-display reaction times were affected by split [for the addition numbers, $F(2, 28) = 26.16$, $p < .01$, $MS_e = 5{,}128.09$; for the multiplication numbers, $F(2, 28) = 17.11$, $p < .01$, $MS_e = 8{,}066.20$] but not by problem size [for the addition numbers, $F(2, 28) < 1$, $MS_e = 2{,}719.39$; for the multiplication numbers, $F(2, 28) = 1.81$, $MS_e = 2{,}583.28$]. Second-display reaction times were strongly affected by the delay between the space-bar response and the onset of the answer. In addition, the means were 603, 578, 565, and 563 ms for the 0-, 125-, 250-, and 500-ms delays, $F(3, 42) = 14.73$, $p < .01$, $MS_e = 3{,}371.50$; in multiplication, the means were 576, 544, 541, and 542 ms, $F(3, 42) = 13.88$, $p < .01$, $MS_e = 2{,}952.07$. Delay did not interact with problem size in either experiment. Delay did not interact with split for addition numbers, but it did for multiplication numbers, $F(6, 84) = 3.03$, $p < .01$, $MS_e = 1{,}373.99$. The interaction reflected a reduction in the magnitude of the split effect as delay increased, though the relative ordering of the different conditions remained the same (i.e., split 0 < split 12 < split 2).

[2] Experiment 3 involves addition; Experiment 4 involves multiplication; Experiment 3a involves number comparison using addition; Experiment 4a involves number comparison using multiplication.

Table 1
Reaction Times (RT, in ms) to First and Second Displays and Accuracy (Acc.) for Second Displays in the Addition and Multiplication Tasks From Experiments 3 and 4

	Problem size in addition							Problem size in multiplication						
	1		2		3			1		2		3		
Split	RT	Acc.	RT	Acc.	RT	Acc.	*M*	RT	Acc.	RT	Acc.	RT	Acc.	*M*
							First display							
0	569		620		638		609	596		663		660		640
2	550		635		638		608	570		630		672		624
12	586		626		626		613	576		679		685		647
M	569		625		635			585		659		672		
							Second display							
0	533	97	589	95	589	95	570	503	98	563	97	588	94	551
2	623	95	662	94	660	93	648	588	99	633	96	677	92	633
12	560	97	584	98	593	98	579	597	98	630	96	648	96	625
M	562		606		608			548		598		625		

Number comparison performance was compared with arithmetic performance in ANOVAs on second-display reaction times. Arithmetic was slower than number comparison for both addition and multiplication, but the effect was significant only in multiplication, $F(1, 28) = 4.26$, $p < .05$, $MS_e = 46{,}574.13$; in addition, $F(1, 28) < 1$, $MS_e = 62{,}457.20$. Problem size effects were larger in magnitude in arithmetic than in number comparison, producing significant interactions in addition, $F(2, 56) = 12.15$, $p < .01$, $MS_e = 1{,}245.93$, and in multiplication, $F(2, 56) = 8.98$, $p < .01$, $MS_e = 1{,}762.49$. Split effects were larger in arithmetic than number comparison in addition, $F(2, 56) = 3.62$, $p < .05$, $MS_e = 3{,}201.21$, and were different in pattern in multiplication, $F(2, 56) = 6.14$, $p < .01$, $MS_e = 1{,}717.54$, as was observed in Experiments 2 and 2a.

Discussion

Experiments 3 and 4 tested the hypothesis that verification is production plus comparison by attempting to separate production and comparison in time. The procedure was partly successful in that reaction times to the first display were affected by problem size but not split, as they would be if they reflected only a production-like process. However, reaction times to the second display were affected by both problem size and split. If they reflected only comparison, they should have been affected by split alone. It is as if subjects pressed the space bar and then compared the equation as a whole against memory, contrary to the hypothesis that verification is production plus comparison.

The contrast with number comparison leads to the same conclusion. If the procedure had separated production from comparison, second-display reaction times should be the same for arithmetic and number comparison. But they weren't. Second-display reaction times were slower in arithmetic than in number comparison, and problem size and split effects were larger. Something more was going on than just the comparison of numbers.

The conclusions with subject-imposed delays between arguments and answer corroborate conclusions with experimen-

Table 2
Reaction Times (RT, in ms) to First and Second Displays and Accuracy (Acc.) for Second Displays in the Number Comparison Task With Addition and Multiplication Numbers From Experiment 3a and 4a

	Problem size in addition							Problem size in multiplication						
	1		2		3			1		2		3		
Split	RT	Acc.	RT	Acc.	RT	Acc.	*M*	RT	Acc.	RT	Acc.	RT	Acc.	*M*
							First display							
0	488		494		485		489	534		530		532		532
2	482		487		477		482	524		533		528		528
12	487		508		510		502	548		540		522		537
M	486		496		489			535		533		529		
							Second display							
0	539	97	543	96	565	95	549	513	97	523	96	529	96	522
2	601	96	605	96	598	96	601	556	97	572	96	598	96	575
12	598	97	573	97	573	97	581	547	97	553	96	564	97	555
M	569		566		575			532		543		555		

ter-imposed delays. This is remarkable because subject-imposed delays require a voluntary response before the answer appears, whereas experimenter-imposed delays do not. One might expect more active preparation when a voluntary response is required to signal the end of it, but apparently, there was little difference. In both cases, subjects seemed prepared to compare the equation as a whole.

Subject-imposed delays pose the same problem as experimenter-imposed delays: What were subjects doing during the delays if not producing an answer? Moreover, if answers were not produced, why should problem size have affected first-display reaction times? Again, we suggest that subjects encoded the arguments and the operation symbol. The problem-size effect may be open to many interpretations. One consistent with the encoding hypothesis is that encoding time for digits in equations is affected by associative connections between the digits and the equation, just as encoding time for letters in words depends on associative connections between letters and words (e.g., McClelland & Rumelhart, 1981). Digits in small-problem-size equations should be more strongly associated than digits in large-problem-size equations (Ashcraft, 1987; Campbell, 1987b; Siegler, 1988), and their encoding should benefit more from the stronger associations. Encoding should be faster for small problems sizes—hence, the problem-size effect in first-display reaction times. Whether this problem-size effect would be the same as the one produced in response to the equation as a whole is an open question, the answer depending on detailed assumptions about the underlying process. We offer some speculations in the General Discussion.

Experiments 5 and 6

Experiments 1–4 failed to provide any support for the hypothesis that verification is production plus comparison. Instead, they support views in which the entire equation is compared against memory. One surprising result was that subjects seemed to prefer evaluating the equation as a whole, showing evidence of having done so even when substantial delays were interposed—by the experimenter or by themselves—between the arguments and the answer. This preference probably reflects a strategic choice by the subjects because they all could produce sums and products if they wished to. What is interesting is the strength of their preference for this strategy. Experiments 5 and 6 were designed to test the limits of this preference by requiring subjects to produce the correct sum or product and masking the arguments before the answer appeared. This procedure guaranteed that subjects had the correct sum or product in mind before the answer appeared, and it prevented subjects from seeing the equation as a whole. Under these circumstances, subjects should abandon the strategy of comparing the whole equation, choosing instead to compare the produced answer with the presented one.

The experiments used the subject-imposed delay procedure of Experiments 3 and 4. Subjects were required to say the sum (Experiment 5) or product (Experiment 6) of the arguments out loud before pressing the space bar. Also, when the answer appeared, masks (#s) appeared in the positions that the arguments occupied, so the arguments and the answer never appeared simultaneously.

The requirement to complete the utterance of the answer before pressing the space bar resulted in long and excessively variable space-bar latencies, which made interpretation very difficult. Consequently, our predictions addressed reaction times to the answers. If subjects compared the answer against the sum or product they produced before pressing the space bar, there should be no effect of problem size in the answer reaction times. However, if they still preferred to compare the equation as a whole against memory, then problem size should affect the answer reaction times, and the magnitude of the effect should be about the same as in Experiments 3 and 4.

Method

Subjects. Each experiment used a separate group of 16 subjects from introductory psychology classes or the general student body, for a total of 32 subjects. Introductory psychology students received course credit for participating; the others were paid $3.50.

Apparatus and stimuli. These were the same as in Experiments 3 and 4, except that a string of five number signs (i.e., #####) appeared in place of the arguments and operation symbol in the answer display. The string was constructed in such a way that the first number sign covered the first argument, the third number sign covered the operation symbol, and the fifth number sign covered the second argument.

Procedure. The procedure was the same as in Experiment 3 and 4, except that subjects were told to utter the correct sum or product of the arguments out loud before pressing the space bar. To prevent confusion or interference from answers uttered by other subjects, only 1 subject was tested at a time, unlike the previous experiments. There was no formal check to ensure that subjects did completely utter the answer before pressing the space bar. We observed each subject for the first few trials to be sure they followed instructions and that they all spoke before pressing the space bar at that time. In most cases, we could hear the utterance and the click of the space bar from the office adjoining the testing room, and they seemed to be synchronized appropriately.

Design and data analysis. Reaction times were analyzed as in Experiments 3 and 4. First-display reaction times were analyzed in a 3 (problem size) × 3 (split) within-subjects design, though these data were not easily interpretable, given the requirement to utter the response out loud before pressing the space bar. Second-display reaction times were analyzed in a 3 (problem size) × 3 (split) × 4 (delay between space-bar response and onset of second display) design. They were also compared with second-display arithmetic reaction times from Experiments 3 and 4 and with second-display number comparison reaction times from Experiments 3a and 4a, in 3 (problem size) × 3 (split) within-subjects designs, with experiments as a between-subjects factor.

Results

Reaction times to the first and second displays are presented as a function of split and problem size in Table 3. The left panel contains data from the addition task; the right panel contains data from the multiplication task. The table also contains accuracy data for responses to the second display.

Reaction times to the first display are difficult to interpret because subjects were required to utter the sum or product before pressing the space bar, and subjects may have varied widely in the criteria they used to decide when to press the space bar. Some may have waited until the utterance was complete; others may have pressed the space bar in synchrony

Table 3
Reaction Times (RT, in ms) to First and Second Displays and Accuracy (Acc.) for Second Displays in the Addition and Multiplication Tasks From Experiments 5 and 6

	Problem size in addition							Problem size in multiplication						
	1		2		3			1		2		3		
Split	RT	Acc.	RT	Acc.	RT	Acc.	*M*	RT	Acc.	RT	Acc.	RT	Acc.	*M*
							First display							
0	816		949		960		911	1,121		1,349		1,495		1,322
2	824		968		931		908	1,111		1,352		1,505		1,323
12	819		928		931		893	1,138		1,353		1,480		1,324
M	819		949		946			1,123		1,351		1,494		
							Second display							
0	527	97	551	96	562	95	547	586	97	602	96	611	95	621
2	621	95	654	95	624	96	633	640	97	670	96	679	97	640
12	596	98	589	98	585	98	590	638	96	648	97	661	98	650
M	567		586		583			613		631		641		

with their utterance. The first-display reaction times were substantially longer here than they were in Experiments 3 and 4, averaging 905 ms in addition and 1,323 ms in multiplication.[3] Analyses of variance on the first-display reaction times yielded significant main effects of problem size [in addition, $F(2, 28) = 8.26$, $p < .01$, $MS_e = 30{,}103.02$; in multiplication, $F(2, 28) = 27.74$, $p < .01$, $MS_e = 60{,}334.54$] but no effects of split [in addition, $F(2, 28) < 1$, $MS_e = 3{,}857.59$; in multiplication, $F(2, 28) < 1$, $MS_e = 2{,}629.85$].

Reaction times to the second display averaged 579 ms in addition and 628 ms in multiplication, which were reasonably close to the values observed in Experiments 3 and 4 (592 and 590 ms, respectively). Second-display reaction times were affected by split [in addition, $F(2, 28) = 38.43$, $p < .01$, $MS_e = 9{,}327.82$; in multiplication, $F(2, 28) = 36.69$, $p < .01$, $MS_e = 5{,}827.13$], reflecting the underlying comparison process. Problem-size effects were nearly significant in addition, $F(2, 28) = 3.23$, $p < .06$, $MS_e = 4{,}257.23$, and highly significant in multiplication, $F(2, 28) = 9.24$, $p < .01$, $MS_e = 4{,}389.74$, which could suggest that the procedure did not separate comparison from production.

The problem size effects were intermediate between those observed in previous number comparison experiments (Experiments 3a and 4a) and those observed in previous arithmetic experiments (Experiments 3 and 4), though they were closer in magnitude to the number comparison effects. Analyses of variance comparing the present experiments with number comparison showed significant interactions between problem size and experiments [for addition numbers (Experiment 3a), $F(2, 56) = 3.48$, $p < .05$, $MS_e = 846.99$; for multiplication numbers (Experiment 4a), $F(2, 56) = 6.95$, $p < .01$, $MS_e = 875.39$]. Analyses comparing the present experiments with previous arithmetic tasks also showed significant interactions between problem size and experiments [for addition (Experiment 3), $F(2, 56) = 4.64$, $p < .05$, $MS_e = 1{,}416.10$; for multiplication (Experiment 4), $F(2, 56) = 6.56$, $p < .01$, $MS_e = 1{,}978.67$]. Thus, the procedure appears to have been somewhat successful in separating comparison from production.

Second-display reaction times were affected by the delay between the space-bar response and the onset of the answer. In addition, the means were 638, 592, 572, and 557 ms for the 0-, 125-, 250-, and 500-ms delays, $F(3, 42) = 80.70$, $p < .01$, $MS_e = 2{,}211.70$; in multiplication, the means were 675, 648, 621, and 605 ms, $F(3, 42) = 27.24$, $p < .01$, $MS_e = 5{,}066.15$]. Delay did not interact with split or problem size in the addition task, but in the multiplication task it interacted with problem size, $F(6, 84) = 2.51$, $p < .05$, $MS_e = 3{,}463.14$, and with split and problem size jointly, $F(12, 168) = 2.22$, $p < .05$, $MS_e = 2{,}531.69$. In the former interaction, small problems were always faster than large problems, but intermediate problems were intermediate in speed at short delays but slower than the other conditions at long delays. The latter interaction affords no simple description, but it does not compromise the main conclusions. The interaction between problem size and split was significant in the addition task, $F(4, 56) = 4.87$, $p < .01$, $MS_e = 2{,}974.73$, but not in the multiplication task. No other effects were significant.

Discussion

Experiments 5 and 6 tested the limits of the strategic preference to evaluate the equation as a whole, observed in

[3] The 418–ms difference between experiments in first-display reaction times is difficult to explain. The experiments were run at different times—the multiplication experiment in the spring and the addition experiment in the fall—and it is possible that subtle variations in the instructions were responsible for the differences. Subjects were instructed to say the answer out loud before pressing the space bar, and subjects' criteria for deciding when to press (relative to their utterance) may have varied between experiments. There was nothing in the task demands to force a tight coupling between producing the answer and pressing the space bar, so there was plenty of room for strategies to operate. Note, however, that these arguments do not apply to second-display reaction times. There, subjects were instructed to respond as quickly and accurately as possible, and this task demand forced prompt responding.

the previous experiments. The arguments and answer never appeared simultaneously, as they had in previous experiments; the arguments were masked when the answer appeared. And subjects had to utter the sum or product out loud before pressing the space bar, guaranteeing that they had produced an answer. The question was whether they would compare that answer with the presented one or compare the equation as a whole against memory. The evidence suggests they mostly compared answers: The effects of problem size on second-display reaction times were small, not much larger than effects in number-comparison (cf. Experiments 3a and 4a) and substantially smaller than effects in previous arithmetic experiments (cf. Experiments 3 and 4). However, second-display reaction times were longer than number-comparison reaction times, and the second-display problem-size effect was significantly larger than the number-comparison problem-size effect. Perhaps the procedure did not remove all of the differences between tasks.

General Discussion

The experiments were designed to test the hypothesis that verification is production plus comparison. That hypothesis predicts a reduction in problem-size effects when a delay is imposed between the arguments and the answer. The delay should separate production from comparison and absorb reaction-time effects of factors like problem size that affect only production. The experiments provided very little support for the hypothesis. In Experiments 1 and 2, which used experimenter-imposed delays, problem-size effects diminished with delay but did not disappear as predicted. In Experiments 3 and 4, which used subject-imposed delays, large problem-size effects were observed, contrary to prediction. Only in Experiments 5 and 6, in which the arguments were masked and subjects uttered the sum or product out loud before the answer was presented, were problem-size effects clearly diminished by delay. Thus, it appears that verification is not production plus comparison, except in very unusual circumstances. In verification, subjects seem to prefer to evaluate the equation as a whole. The data cannot tell us whether this evaluation occurs on every trial (e.g., if subjects matched the equation as a whole against memory) or as an occasional strategy (e.g., mixing production-plus-comparison with side-stepping strategies), but they rule out the possibility that verification is based only on production plus comparison. This conclusion has important implications for current theories of arithmetic performance and interesting parallels in other aspects of cognition.

Theories of Arithmetic Performance

Counting models. Counting models must predict that verification is production plus comparison (e.g., Groen & Parkman, 1972). They assume that the underlying knowledge representation is the number line (i.e., 0, 1, 2, 3, . . .) and that the "retrieval" process is a counting algorithm. Sums and products[4] are the only information available in counting models, and the only way to "retrieve" them is by counting. Thus, the equation cannot be compared as a whole in verification tasks, nor can the answer be checked for plausibility. The sum or product must first be retrieved and then compared with the presented answer.

In principle, counting could account for our results if we assumed that subjects repeat the whole verification process when the answer appears. That is, they count out the sum or product and compare it with the presented answer. But counting is voluntary and laborious; this assumption seems implausible. Why bother counting again when the answer is already available?

Table-search models. Table-search models, such as Ashcraft and Battaglia's (1978) and Geary et al.'s (1986), must also predict that verification is production plus comparison. They assume that the underlying knowledge representation is a table of arithmetic facts in which the rows and columns represent the digit arguments and the cell entries represent the sums, products, and so on. The retrieval process is spreading activation, which begins at 0,0 and spreads along the rows and columns until the sum or product is activated. As in counting models, sums and products are the only information available, and they can be retrieved in only one way. The models lack the flexibility to retrieve information differently in production and verification.

In principle, table-search models could account for our results if subjects repeated the retrieval process when the answer appears. That seems unlikely because they would have already retrieved a sum or product before the answer appeared, so there would be no point in retrieving a second one.

Associative network models. Associative network models, such as Ashcraft's (1982, 1987) and Campbell's (1987a, 1987b), also assume that verification is production plus comparison. That assumption is explicit in Ashcraft's simulation, and it is at least implicit in Campbell's verbal descriptions. If that assumption is the core of their theories, then our results falsify their models. As with counting and table-search models, these could be salvaged by assuming that subjects produced again when the answer appeared, but that seems implausible. Why produce again when a computed answer is already available?

We suspect that the assumption that verification is production plus comparison is not central to associative network models. Associative network models make three kinds of assumptions: Assumptions about *representation*, assumptions about *microprocesses* that operate on representations, and assumptions about *macroprocesses* that operate on the results of microprocesses. The models assume that arithmetic knowledge is presented in an associative network, linking digits with their sums and products. The microprocess that retrieves

[4] Existing counting theories apply to addition and not multiplication. In principle, there could be counting models of multiplication: Subjects could count in units of the multiplier the number of "counts" specified by the multiplicand (or vice versa). Thus, one could "count multiply" 5 times 3 by counting three steps in units of five (i.e., 5, 10, 15; but see Ashcraft et al., 1984, Experiment 2).

information from the network is *activation*. Activation flows along associative links, activating quiescent nodes. The macroprocess "reads" the pattern of activation in the network and generates the appropriate response. In Ashcraft's and Campbell's models, the macroprocess retrieves a sum or product, selecting the number represented by the most highly activated node in the network. Another macroprocess compares the retrieved answer with the presented one. We believe our results challenge the macroprocess assumptions of these models, yet are quite consistent with the representation and microprocess assumptions.

A Dual Macroprocess Model of Production and Verification

Our results can be interpreted in the framework of associative network models if we assume that production and verification involve different macroprocesses operating on the same representations and microprocesses. Activation flowing through an associative network provides several sources of information that may be exploited to perform arithmetic tasks. Production may involve a macroprocess that chooses the most highly activated node (e.g., Ashcraft, 1987), but verification depends on a macroprocess sensitive to other dimensions of activation.

We suggest that verification involves comparing the amount of activation or "resonance" produced by the equation as a whole against some criterion, deciding "true" if it exceeds the criterion, and "false" if it doesn't. In order for this macroprocess to work, true equations must produce more activation than false ones. That is the case in Ashcraft's simulation, and it is likely in Campbell's model. A true equation activates a set of nodes that are strongly and directly linked to each other, and activation flows among them, reinforcing the activation in each node. But a false equation activates a set of nodes that are not strongly associated or directly connected. Activation flows out through the network, but there is little reinforcement, and thus, less overall activation.

We could model reaction time effects in several ways. One straightforward alternative is to assume that reaction time is inversely proportional to distance from the criterion. This model predicts fast reaction times for high and low levels of activation and slow reaction times for intermediate levels. As activation increases, reaction times to false equations should get slower, and reaction times for true equations should get faster. The model can predict problem-size effects if activation is stronger for smaller problems. Ashcraft (1987) lists several reasons why it should be. The model may also be able to predict some of the effects associated with side-stepping strategies: The model suggests that the split effect may occur not because extreme splits are easy to reject but rather because small splits are hard to reject. The answers in small-split problems may be more strongly associated with the arguments than answers in large-split problems because small-split answers are more likely to occur as errors than are large-split answers (i.e., when we err, we are more likely to be off by a little than a lot). Frequent errors may become associated with the arguments strongly enough to ring true when presented together in an equation and, therefore, take longer to reject (cf. Campbell & Graham, 1985; Siegler, 1988). Similarly, errors may obey parity rules more often than not, so quick rejection of false problems that violate parity rules may also reflect differences in associative strength. These effects may not reflect a deliberate side-stepping strategy after all.

The macroprocess could account for our results in two ways. First, subjects could deliberately instigate the macroprocess when the answer appears, causing a strong problem-size effect even when a substantial delay occurs between the arguments and the answer. Unlike the previous approaches, one could not argue that subjects already produced the answer and therefore the process is redundant; no answer is ever produced. The arguments by themselves may prime the network, but they should not activate it enough to reach a decision. The answer plus the arguments should activate the network sufficiently.

Second, problem-size effects may be produced when the answer occurs because of the flow of activation through the network, quite independent of the person's intentions. The answer will activate the digits associated with it (e.g., in multiplication, 12 will activate 6, 2, 4, and 3), and activation will propagate back through the network from the digits to the answer, in a loop. The time taken for activation to propagate back from the digits to the answer should be at least proportional to the time taken to propagate from the arguments to the answer in the first place, producing the same sort of problem-size effect.

There are likely to be many problems with the model of verification we have sketched. The model must be analyzed in more detail before we can be sure that it must predict what we said it predicts. More analysis and perhaps simulation will be required to see whether the different effects can be predicted simultaneously when the same assumption about representation, microprocess, and macroprocess for the different effects are used. We might discover better ways to instantiate the underlying ideas (e.g., using activation levels to drive a random-walk decision process; see Ratcliff & McKoon, 1988). But we see a lot of promise in the model sketched so far.

The most important contribution of the model is to provide a concrete alternative to the idea that verification is production plus comparison. There is no production in our model, only a comparison of the equation as a whole against memory. The model underscores the idea that production and verification depend on different processes and that idea has important implications for future studies of arithmetic. It implies that no single task reflects arithmetic knowledge directly. Production reflects some aspects of arithmetic knowledge, and verification reflects others. It implies that different tasks may give apparently different answers to the same question. The different macroprocesses may respond differently to the same manipulation, producing some unsettling failures to replicate effects across tasks. More generally, it implies that we cannot learn about arithmetic in general by studying only verification or only production. We must study both tasks and understand the relation between them. Understanding requires a theory of the competence or capacity that underlies

a variety of arithmetic tasks. The tasks themselves are a surface manifestation of the underlying essence we are after. We must learn how the various tasks tap the essence. We must model the underlying knowledge representation and the processes that operate on it at micro- and macrolevels.

Resonance and Retrieval in Other Domains

Our two macroprocess model of arithmetic has precedents in several cognitive domains. Our approach to production and verification parallels the contrast between recall and recognition in studies of episodic memory (e.g., Tulving, 1983),[5] between fact retrieval and semantic verification in studies of semantic memory (e.g., Collins & Loftus, 1975), and between naming and lexical decision in studies of lexical memory (e.g., Neely & Keefe, in press). Production, like recall, fact retrieval, and naming, requires a single response to be retrieved from memory, whereas verification, like recognition, semantic verification, and lexical decision, can be done by evaluating the global response of the memory system (e.g., does it ring true?). Production, recall, fact retrieval, and naming involve a one-out-of-many choice, whereas verification, recognition, semantic verification, and lexical decision involve a binary choice. One-out-of-many choices may be more difficult than binary choices; thus subjects may prefer verification to production, recognition to recall, and so on.

The similarity of task requirements suggests that similar theoretical approaches may succeed in the various domains. The idea that a common representation and microprocess are accessed by different macroprocesses may extend far beyond arithmetic. This idea is well developed in formal theories of memory, such as the SAM (search of associative memory) model by Shiffrin and his colleagues (Gillund & Shiffrin, 1984; Raaijmakers & Shiffrin, 1981), the TODAM (theory of distributed associative memory) model by Murdock (1982, 1983), and the matrix model by Humphreys, Bain, and Pike (1989; Pike, 1984). In these models, the basic memory system consists of matrices or vectors that represent associations between the items and various cues. Recall and recognition tasks access the basic memory system in different ways. In both tasks, cues presented to the memory system evoke items and contexts that were associated with them. Recall involves selecting the most strongly associated item out of all the evoked alternatives; recognition involves evaluating the global response of the memory system, summing over all evoked items. The various authors formalize these assumptions and show that the same representation and microprocess, coupled with different macroprocesses, provides an impressive account of many qualitative and quantitative phenomena in recall and recognition.

It seems straightforward to adapt one of these models to arithmetic or to adapt their desirable features to existing arithmetic models, such as Ashcraft's (1987) or Siegler's (1988). Most of the memory models deal only with accuracy, whereas most arithmetic effects appear in reaction time, but it should not be difficult to develop versions of the models that predict reaction time. In principle, the resulting theory would be broad in scope and precise in its predictions, accounting for the effects of the structural variables that have preoccupied the field so far, as well as the contrasts and similarities between production and verification revealed in our research.

---

[5] The literature on recall and recognition contains an analog of the verification-is-production-plus-comparison hypothesis: The generate-and-recognize hypothesis claims that people recall by generating plausible alternatives, attempting to recognize them, and reporting only the ones that are recognized.

References

Ashcraft, M. H. (1982). The development of mental arithmetic: A chronometric approach. *Developmental Review, 2*, 213–236.

Ashcraft, M. H. (1987). Children's knowledge of simple arithmetic: A developmental model and simulation. In C. J. Brainerd, R. Kail, & J. Bisanz (Eds.), *Formal models in developmental psychology* (pp. 302–338). New York: Springer-Verlag.

Ashcraft, M. H., & Battaglia, J. (1978). Cognitive arithmetic: Evidence for retrieval and decision processes in mental addition. *Journal of Experimental Psychology: Human Learning and Memory, 4*, 527–538.

Ashcraft, M. H., Fierman, B. A., & Bartolotta, R. (1984). The production and verification tasks in mental addition: An empirical comparison. *Developmental Review, 4*, 157–170.

Ashcraft, M. H., & Stazyk, E. H. (1981). Mental addition: A test of three verification models. *Memory & Cognition, 9*, 185–196.

Campbell, J. I. D. (1987a). Network interference and mental multiplication. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*, 109–123.

Campbell, J. I. D. (1987b). Production, verification, and priming of multiplication facts. *Memory & Cognition, 15*, 349–364.

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

Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review, 82*, 407–428.

Donders, F. C. (1969). On the speed of mental processes. In W. G. Koster (Ed.), *Attention and performance II* (pp. 412–431). Amsterdam: North-Holland. (Original work published 1868)

Geary, D. C., Widaman, K. F., & Little, T. D. (1986). Cognitive addition and multiplication: Evidence for a single memory network. *Memory & Cognition, 14*, 478–487.

Gillund, G., & Shiffrin, R. M. (1984). A retrieval model for both recognition and recall. *Psychological Review, 91*, 1–67.

Groen, G. J., & Parkman, J. M. (1972). A chronometric analysis of simple addition. *Psychological Review, 79*, 329–343.

Humphreys, M. S., Bain, J. D., & Pike, R. (1989). Different ways to cue a coherent memory system: A theory for episodic, semantic, and procedural tasks. *Psychological Review, 96*, 208–233.

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

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*, 171–180.

McClelland, J. L. (1979). On the time relations of mental processes: An examination of systems of processes in cascade. *Psychological Review, 86*, 287–330.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review, 88*, 375–407.

Miller, K., Perlmutter, M., & Keating, D. (1984). Cognitive arithmetic: Comparison of operations. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 10*, 46–60.

Murdock, B. B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review, 89*, 609–626.

Murdock, B. B. (1983). A distributed-memory model for serial order information. *Psychological Review, 90*, 316–338.

Neely, J. H., & Keefe, D. E. (in press). Semantic context effects on visual word recognition: A hybrid prospective/retrospective processing theory. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 23). New York: Academic Press.

Parkman, J. M. (1972). Temporal aspects of simple multiplication and comparison. *Journal of Experimental Psychology, 95*, 437–444.

Parkman, J. M., & Groen, G. J. (1971). Temporal aspects of simple addition and comparison. *Journal of Experimental Psychology, 89*, 335–342.

Pashler, H. (1984). Processing stages in overlapping tasks: Evidence for a central bottleneck. *Journal of Experimental Psychology: Human Perception and Performance, 10*, 358–377.

Pike, R. (1984). A comparison of convolution and matrix distributed memory systems. *Psychological Review, 91*, 281–294.

Posner, M. I., & Boies, S. J. (1971). Components of attention. *Psychological Review, 78*, 391–408.

Posner, M. I., & Mitchell, R. F. (1967). Chronometric analysis of classification. *Psychological Review, 74*, 392–409.

Raaijmakers, J. G. W., & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review, 88*, 93–134.

Ratcliff, R., & McKoon, G. (1988). A retrieval theory of priming in memory. *Psychological Review, 95*, 385–408.

Restle, F. (1970). Speed of adding and comparing numbers. *Journal of Experimental Psychology, 85*, 274–278.

Schweickert, R. (1978). A critical path generalization of the additive factor method: Analysis of a Stroop task. *Journal of Mathematical Psychology, 18*, 105–139.

Schweickert, R. (1983). Synthesizing partial orders given comparability information: Partitive sets and slack in critical path networks. *Journal of Mathematical Psychology, 27*, 261–276.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General, 117*, 258–275.

Stazyk, E. H., Ashcraft, M. H., & Hamann, M. S. (1982). A network approach to simple multiplication. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 8*, 320–335.

Sternberg, S. (1969). The discovery of processing stages: Extensions of Donders' method. In W. G. Koster (Ed.), *Attention and performance II* (pp. 276–315). Amsterdam: North-Holland.

Tulving, E. (1983). *Elements of episodic memory*. New York: Oxford University Press.

Zbrodoff, N. J. (1979). *Development of counting and remembering as strategies for performing simple arithmetic in elementary school children*. Unpublished master's thesis, University of Toronto, Ontario.

Zbrodoff, N. J., & Logan, G. D. (1986). On the autonomy of mental processes: A case study of arithmetic. *Journal of Experimental Psychology: General, 115*, 118–130.

Received February 23, 1989
Revision received May 15, 1989
Accepted May 16, 1989 ■

Publication Practices and Scientific Conduct

The recent disclosures of fraud in the conduct of research, reporting of research, or both in a number of scientific disciplines have prompted a widespread program of self-examination of publication practices and ethics.

The editor joins with APA in reminding authors of the principles of good publication practices and scientific conduct. Prospective authors are directed to the *Publication Manual of the American Psychological Association* (3rd ed.) and to the "Instructions to Authors" printed in this issue. The requirements of data availability, replicability, authorship credit, ethical treatment of subjects, and primary publication of data are important—they are meant to ensure responsible science and appropriate use of scarce and valuable resources.