HUMAN COMPUTER INTERACTION, 1990, Volume 5, pp. 49-93

Specific Versus General Procedures in Instructions

Richard Catrambone
Georgia Institute of Technology

ABSTRACT

A good deal of research in cognitive psychology has demonstrated that, although learners can solve problems that are just like the ones they have been trained on, they often have great difficulty solving new types of problems. People also have difficulty trying to understand instructions or training materials that try to teach a procedure at a level that is general enough to apply to many different kinds of cases. These two findings lead to a quandary for people designing instructions for procedural tasks such as operating computer software: Instructions should be written with a good deal of specificity so that new users can understand and use them right away, but at the same time the user will have great difficulty generalizing what they have learned to novel cases. Experiment 1 seems to echo this quandary. Computer novices, in this study, were able to follow specific instructions for using a word processor more easily than general instructions. However, they had great difficulty generalizing the specific instructions to novel tasks. Experiment 2 demonstrates that when specific instructions are rewritten to help users form a more general procedure, novices can easily do new tasks and still maintain their initial quality of performance. A production rule formalism is used to represent the knowledge users obtain from instructions and to explore the conditions under which these productions can be generalized. Experiment 2 suggests that this knowledge can be used to improve the generalizability of instructions.

This article is based on Catrambone (1988).
Author's present address: Richard Catrambone, School of Psychology, Georgia Institute of Technology, Altanta, GA 30332.

CONTENTS

1. INTRODUCTION

Poor instructions are ubiquitous. Yet, relatively little research has appeared in the psychological literature on the design of instructions. There are many books and manuals that give advice on how to write instructions, but the writers of these guides have often had to rely on their own experience and intuitions when developing their suggestions rather than being able to point to a body of research or a useful psychological model as support. Although personal experience, or the aggregate experience of a group of writers for a company, is valuable when designing documents, it may not provide enough information on the conditions under which a principle should be used, why it works, or how to apply it.

Documentation design might benefit from the development of analytical models of cognition that are especially suited to represent learners' knowledge of procedures combined with systematic research on how people follow instructions. The models could generate performance predictions as a function of the instructions subjects receive. Although there are many questions that could be asked about the design of instructions, recent research (this article included) in cognitive psychology has begun to define an issue that is timely—that is, the effects of general versus specific instructions on learners' ability to initially use the instructions and to later adapt them to novel cases.

It is clear that specificity-generality is a continuum and that these terms are relative. Consider the levels of specificity that could be used to describe functions on a word processor. Suppose the word processor allows a user to delete, copy, and move text by first highlighting that text. The user specifies the text to be highlighted by putting the cursor at the beginning of that text, pressing a function key, and then typing the final character of that text (which causes the computer to search for that character and highlight all the text between the cursor's starting point and the point at which it encountered the first occurrence of the target character).

Although the concepts of highlighting, deleting, and moving text and doing string searches are familiar to experienced users, they are often bewildering to new users. Instead of forcing a naive user to face all these concepts at once, instructions on how to use the word processor could take a more modest approach by, for example, giving all the steps for deleting particular entities, such as words and then sentences. The procedure for deleting a word could be described quite concretely: Put the cursor under the first letter of the word, press the Delete key, press the space bar (this causes the computer to search for a space, thereby highlighting the word because the first space would be immediately after the word), and then press the Enter key.

Conversely, new users could be taught the general deletion procedure. They could be given the steps for moving the cursor to the beginning of the text, pressing the Delete key, typing the final character, and then pressing the Enter key. The next-to-last step, however, could be difficult for a new user to understand because it might seem odd to have to type a character while trying to delete something. The user could also be confused about what constitutes a character. It is a letter, number, punctuation mark, all of the above? The user might also expect the character to somehow appear on the screen. The user might wonder if he or she should really type the last character of the to-be-deleted text or maybe just the last letter of the first word, even if that is not the end of the text to be deleted. The point is that naive users often possess beliefs about how devices such as computers might operate, and these beliefs could be very much at odds with reality (Mack, Lewis, & Carroll, 1983). Thus, it could be dangerous to provide the new user with too general an explanation because it allows the user's misconceptions to play a larger role in the interpretation of the instructions.

Finally, an even more general explanation of the deletion procedure would be to describe all the "block operations" (e.g., delete, copy, move) for the word processor in terms of moving the cursor to the starting point, selecting the desired block operation, highlighting the target text, and then possibly moving the cursor again. An explanation at this level would be quite general and perhaps allow the users to grasp the "big picture" quite quickly, but it could also just as easily confuse them.

The point to be made here is that there is a continuum of instructions, from specific to general, and, at least at present, the only functional way of differentiating instructions written at different levels of specificity is by counting the number of cases to which they apply.

Based on research presented in this article, it is argued that there is a tradeoff between specific and general instructions. For the novice, specific instructions might be easier to comprehend and to use, at first, than general instructions because the specific instructions require very little inferencing on the user's part; they can be followed as written. General instructions, by definition, do not give the details for each specific case; the user must infer the details. The inferencing required may be more or less difficult depending on how broad the domain is, but it is still assumed to be more difficult than following specific instructions.

Although specific instructions might have an advantage in initial use, general instructions, once learned, are presumed to have an advantage when the user attempts tasks that cannot be accomplished using exactly the same steps as those on which the user was trained. For a person trained with specific instructions, the novel task would require constructing a new procedure tailored for the new case or deriving the more general procedure that could be used for all cases. This generalizing task might be quite difficult, depending on the nature of the specific instructions the learner originally studied. Although the person trained on general instructions will primarily try to determine low-level details in order to deal with a novel case, such as which menu item to pick in a word processor, the person trained on specific instructions might not even possess the high-level concepts in which to categorize the new problem (e.g., the user might not know which menu to go to or that they even need to use a menu). The person trained with specific instructions might possess only a series of low-level details, such as the specific keystrokes to do a specific task, that cannot easily be generalized or mapped onto the new problem; or conversely, the user may think the details map onto the new problem when in fact they do not (Mack et al., 1983).

The problem-solving literature highlights the difficulties people have in making generalizations from specific procedures they have learned. For example, Reed, Dempster, and Ettinger (1985) found that college students studying algebra word problems could not solve new problems that required them to adapt old solution procedures. This and other research suggest that learners get tied to the details of a procedure and are unable to infer a more general procedure. This conclusion has implications for instructional design inasmuch as specific instructions could also lead learners to focus on details of specific procedures and fail to infer the more general procedure. It seems that if novices in a domain are going to be taught procedures with specific instructions, then they will also require additional information that allows them to make generalizations from those procedures (Catrambone & Holyoak, in press; Kieras & Bovair, 1984), or perhaps they need to learn specific procedures that are designed to support the necessary generalizations.

1.1. An Approach to Analyzing Instructions

It is proposed here that a production rule analysis might serve as a tool for representing what learners acquire from instructions and that, combined with empirical studies, it can be used to predict the likelihood that learners will be able to form a more general procedure. This general procedure then allows learners to successfully complete tasks that did not use exactly the same steps as those tasks on which they were trained. The predicted likelihood of forming the more general procedure is constrained by a number of experiments that have indicated the situations under which generalizations are formed (e.g., see Bovair, Kieras, & Polson, 1990; Foltz, Davies, Polson, & Kieras, 1988; Singley & Anderson, 1988).

A production rule is an IF-THEN statement which states that if a particular condition exists in working memory, then a particular action will take place. This action could be a change to the contents of working memory or some physical act such as pressing a key on a keyboard. When the condition for a particular production is matched, then that production "fires." This means that the action part of the production is carried out. For example, part of a person's knowledge of a particular word processor might be represented by the following production:

```
 ( (IF   (Goal is to underline a word)
         (Note cursor is at the first letter of that word))
  (THEN  (Press underline key)
         (Delete Goal to underline a word)) ).
```

If a person is editing a manuscript and one of his or her current goals is to underline a particular word, then he or she must move the cursor to that word. This would be modeled by other productions that would move the cursor and deposit a note in working memory that the cursor has been moved to the first letter of the target word. Then, this production would fire because the goal to underline the word still exists in working memory and the note about the cursor position would also be present.

It is worth [redacted]g that a production rule analysis is a type of task analysis.
noting

One potentially useful feature of a production rule analysis is that it makes goals and the use and reuse of methods explicit. In addition, the IF-THEN nature of productions could allow them to serve as building blocks for creating instructions rather than only analyzing them after the instructions have been created. This point is discussed later. Another advantage of a production rule analysis of instructions is that it can pinpoint places where a change in the instructions should theoretically produce important changes in behavior. It is the precision of such predictions that has made a production rule approach to modeling skill acquisition attractive to some researchers. As with other forms of task analysis, different researchers may come up with different production rule analyses of the same task. However, the use of production rule templates and the development of learning constraints may help to standardize this approach.

Although it is argued in this article that a production system approach is useful for analyzing instructions and representing knowledge, it is not claimed that a production system approach captures all the complex cognitive activity that occurs when someone is learning from instructions. Rather, a production system is a simplification of the very complex human system. Nevertheless, prior research (discussed later) has shown the production system approach to be a useful tool for predicting performance as learners acquire skills.

1.2. Development of Learning Constraints

Bovair et al. (1990) found that the time it took subjects in their study to learn various procedures on a word processor was a function of the number of productions that subjects were presumed to have learned in order to carry out the procedures. In addition, once a subject learned a particular procedure, such as deleting text, the time it took to learn a new procedure, such as moving text, could be predicted as a function of the number of new productions (i.e., those that were not already learned from the deletion procedure) contained in the new procedure. That is, old productions contained in a new procedure did not contribute to the learning time of the new procedure.

Another important finding from Bovair et al. (1990) is that generalization of productions can take place under certain circumstances. Two specific productions will generalize to form a new production if the two specific productions differ only in the "predicate" term of the goal clause and wherever else that predicate appears in the productions. The formation of this generalized production seems to require less time or cognitive effort than acquiring a new production. For example, consider the following two productions (taken from Bovair et al., 1990):

```
    (IF ((GOAL Delete String)            (IF ((GOAL Copy String)
         (STEP Verify Delete))                (STEP Verify Copy))
  THEN ((VerifyTask Delete)            THEN ((VerifyTask Copy)
        (Delete STEP Verify Delete)          (Delete STEP Verify Copy)
        (Add STEP Press Accept)) )           (Add STEP Press Accept)) )
```

These productions are identical except for the predicate term in the goal clause, delete versus copy, and wherever else in the productions this term occurs. As a result, the following generalized production can form:

```
   (IF   ((GOAL ?X String)
          (STEP Verify ?X))
  THEN   ((VerifyTask ?X)
          (Delete STEP Verify ?X)
          (Add STEP Press Accept)) )
```

Thus, if an incoming production rule (i.e., a production that is assumed to be contained in the instructions a person is currently reading) is identical to an already possessed production except for the predicate term, then a generalization of the two productions is formed fairly easily. This generalized production can be used in a wider variety of procedures than the specific productions from which it was generalized. Thus, less problem solving will be required in the future.

Related work by Foltz et al. (1988) also bears on the issue of generalizing productions. They taught subjects to do a particular menu task on a word processor. Later, subjects were taught another menu task that was identical to the first except that the menu item name was different. The productions that represented these two tasks were identical except for the predicate term. Based on the work of Bovair et al. (1990), it was assumed that the time to learn the second task would be reduced compared to the first task because generalization would take place. Furthermore, given the semantic similarity of the two menu items (e.g., DISCARD and DELETE), Foltz et al. assumed that the time to form the generalization would be further reduced. However, subjects took longer to learn the second task than prior research would have predicted (e.g., Kieras & Bovair, 1986), suggesting that generalization may not be quite as effortless as had been assumed.

Although the previously mentioned work has suggested that forming a generalized production requires two productions from which to generalize, other models do not. For example, Anderson (1987) suggested that generalizations from single examples could be accommodated by models that contain a number of problem-solving productions. These productions would somehow be able to work with the semantic rather than just the syntactic components of procedures. The empirical results to be presented in the current study argue against generalization of a semantic nature; however, a large number of domains should be examined before any strong claims are made on this issue.

Another interesting finding of Foltz et al. (1988) was that once subjects learned a particular procedure for achieving some goal, they required almost no learning time to learn a new procedure for that goal if the new procedure consisted of a subset of the productions from the old procedure. This suggests that learners can delete parts of a procedure to form a new procedure (for achieving the same goal) without much difficulty. This result is related to one found by Singley and Anderson (1988), who studied people learning text editing and found that if people learned a method that was called by a particular high-level goal (e.g., specifying a string for a search operation), this same method could be used in the service of a different goal with a minimal learning cost (e.g., specifying a string for a replacement operation).

Based on results just discussed, it is assumed here that a more general production can be formed based on two more specific instructions if the specific instructions are identical except for their arguments. This assumption could potentially be relaxed if differences in the productions (or, in fact, the terms in a single production) could be mitigated by other types of reasoning such as reasoning by analogy (Anderson, 1987). These results suggest two things: (a) Forming a generalized production may be easier than learning a new production, but it is not clear by how much, and (b) when a procedure is learned, part or all of it can be used as part of another procedure with minimal cognitive cost.[1]

The experiments presented in this article were designed to shed new light on how productions acquired from instructions can be generalized and how this knowledge can be used to improve the quality of instructions.

2. EXPERIMENT 1: DELETION AND FORMATTING INSTRUCTIONS

The first experiment compared the effects of general versus specific instructions on subjects' ability to learn the instructions and to apply them to novel cases. Because the focus of the research was primarily to examine the conditions under which specific instructions could be successfully generalized, it was important to find a word-processing task for which specific instructions

---

[1] It may be the case that with a great deal of practice a method becomes "opaque" and can no longer be broken down into subparts that can be used in other procedures. However, Singley and Anderson (1988) found that components of procedures that are fairly well practiced do transfer to new procedures. Catrambone and Holyoak (in press) found that a practiced method can generalize, however, even when practice only entails four training problems.

would be difficult to generalize and one in which they would be easy to generalize.

Deletion and formatting procedures were chosen for the instructional manipulation because they represent tasks that are hypothesized to benefit differentially from specific instructions, at least for the word processor (DisplayWrite 1) used in the experiments presented here. Deletion is done by (a) putting the cursor at the beginning of the text to be deleted, (b) pressing a function key, (c) selecting deletion from a menu, (d) highlighting the text to be deleted, and (e) pressing the Enter key. When the deletion function is selected, the computer displays the prompt "Delete what?" and waits for the target character to be specified. When the user specifies the target, the computer then highlights (using reverse video) the text up to and including that character. The target character can be pressed several times if it occurs more than once in the text.

Because there are many superficially different types of text that could be deleted—such as words, sentences, paragraphs, and unwanted letters in the middle of a word—it might be more beneficial, with respect to the learner's ability to delete any arbitrary text, to explain the range-selection aspect of deletion in terms of searching for a target rather than expecting people to infer the general range-selection strategy by learning how to delete certain specific cases such as words, sentences, and paragraphs. This would be especially true if the user were taught to highlight a word for deletion by pressing the space bar in response to "Delete what?" or to highlight a sentence by pressing a period. These steps are unlikely to generalize to the notion of pressing the end character of the to-be-deleted text because a novice might assume that pressing the space bar is an arbitrary command for highlighting a word and that pressing a period will highlight any sentence, regardless of whether it ends in a period (cf. Mack et al., 1983). If pressing the space bar and period are encoded as semiarbitrary parts of procedures for highlighting words, sentences, and paragraphs, then there would not seem to be much of a basis for forming the generalization that what the user is really doing is typing the last character of the to-be-deleted text.

Although deletion might suffer from specific instructions in the long run, formatting seems to be the type of task in DisplayWrite 1 that can usefully be characterized by specific cases. That is, the specific cases are sufficiently similar to support generalization. Changing a format value requires (a) moving the cursor to the beginning of the document, (b) displaying the main format menu by pressing a function key, (c) selecting the appropriate item from the main menu, (d) selecting the appropriate item from the submenu that appears, (e) erasing the current value of the format parameter that appears, (f) typing the new parameter value, (g) pressing the Enter key, and (h) removing the main format menu item from the screen.

There are two reasons why format tasks might be best described through specific instructions. First, the different format procedures are almost identical, except for the obvious differences in menu choices, and thus, it is likely that the learner could quickly form generalizations from them (cf. Bovair et al., 1990). A second reason is that it might be difficult initially to understand a general description of the commonalities in setting various format features because it would involve keeping a good deal of information in working memory in order to deal with various branchings in the general procedure. To change a format setting, the user must traverse two menus and then specify the new format value. Because main and submenu selections are made in the same way, the menu-item selection procedure should be described once at a general level in order to adhere to the spirit of general instructions. However, selecting something on a main menu sends the user to the submenu; selecting something on a submenu sends the user to a screen that allows the user to change the format value. This branching must be described if the menu selection process is written at a general level. As a result, it becomes impossible to provide the instructions in a linear form without repeating a lot of the information. If the information is not repeated (as was the case in this experiment), then the resulting nonlinear explanation produces a large cognitive overhead for the learner in order to keep track of whether a menu selection will lead to another menu or to the format specification screen. This can make comprehension and execution of the procedure difficult. Of course, there are ways of dealing with this problem, such as the use of diagrams and flow charts (e.g., Desaulniers, Gillan, Rudisill, & Burns, 1988), but they were not the focus of this research.

Thus, it is predicted to be more beneficial to explain several specific format procedures because they can be expressed linearly and because subjects are predicted to be able to generalize the common elements rather easily.

2.1. Method

Subjects. Subjects were 48 female undergraduates at the University of Michigan and 6 females and 2 males recruited from the Ann Arbor community. Subjects had little or no computer experience (as indicated on a questionnaire) and none had used a word processor. They were paid $10 for the 1 hr experiment.

Materials. The two word-processing functions that were examined were deletion and formatting. The factor that was manipulated in constructing the instructions was whether the instructions described general procedures or procedures tailored for specific cases. The instructions were created to adhere to the spirit of general and specific instructions but were not subjected to a production rule analysis until they were complete.

The general deletion instructions (see Appendix A) showed how to delete any arbitrary text, whereas the specific deletion instructions showed how to delete a word, a sentence, and a paragraph. These instructions were broken up into screens of information in order to present them on a CRT. The presentation method is described later. There are two features of the deletion instructions worth noting. The first is that the specific instructions are approximately three times longer (i.e., required three times as many screens) than the general ones. This is because the specific instructions provide all the steps for deleting words, sentences, and paragraphs even though most of these steps are common to all three text types.[2]

The second feature is that the general deletion instructions and each of the specific instructions are almost identical. The only real difference is that the general instructions tell users to type the last character of the text they want to delete in response to the "Delete What?" prompt, and they explain why the character might have to be typed more than once (see Screen 7 of Appendix A). The word and sentence instructions simply tell the user to press the space bar or type a period in response to the "Delete What?" prompt. The paragraph deletion instructions tell the user to "type the punctuation mark at the end of the paragraph, typing it over and over again until the paragraph is highlighted." Thus, the general instructions contain some rationale for multiple presses of the target, whereas the specific paragraph deletion instructions do not.[3]

The general format instructions (see Appendix B) discuss the relationship between the main menu and submenus that must be traversed and, after a menu item has been selected, either another menu or a "request for information" would appear. The specific formatting instructions show how to change the line spacing, top line distance (from the top of the page), and bottom line distance of a document. The instructions for changing the line spacing are included in Appendix B. Although some of the screens are identical for the specific and general instructions, most are different because the general instructions do not explain the procedure linearly, as the specific instructions do. As with the specific deletion instructions, the specific format instructions repeat the common screens for each of the procedures even though these procedures are almost identical except for the menu selections.

An important issue concerning the instructions was their overall quality, irrespective of the experimental manipulation (i.e., Were the instructions clear?). The deletion instructions were critiqued by a professional documentation design company (S. Rosenbaum, personal communication, March

---

[2] The differences in the three procedures could have been provided just at the point of divergence, but it was decided to document the three procedures separately in order to examine the effects on reading times. Thus, the differences in the length of the instructions is somewhat artificial and, more than likely, influence reading times beyond any effect of the specific-general manipulation.

[3] This rationale is not a necessary component of general instructions and was, therefore, deleted from the second experiment.

1988) and were judged to be clear. Thus, there is independent, albeit subjective, evidence assuring the quality of the deletion instructions. Although the formatting instructions were not professionally critiqued (due to scheduling constraints), they were examined by a number of people. In addition, all subjects were able to complete the training tasks, suggesting that both the general and specific deletion and formatting instructions achieved some level of usability.

Procedure. Subjects participated in the experiment one at a time and were randomly assigned to either the specific instruction (SI) group or to the general instructions (GI) group and to either deletion or formatting.

Equipment Orientation. Subjects were first shown several features of the IBM computer and the word-processing program. The computer screen displayed an "empty" document. Subjects were shown what the cursor was and were told how text automatically wraps around the screen. The Shift key and the Arrow keys on the keyboard were pointed out, and the subject was shown how to move the cursor around the screen with the Arrow keys. Subjects were then asked to type a short paragraph to allow them to get comfortable with the keyboard.

Next, subjects were shown how to use a Burroughs workstation to read the instructions on how to do various tasks. The instructions always included procedures for retrieving and exiting documents and inserting text into a document. These instructions were the same for all subjects. In addition, the instructions also included procedures for either deleting text or formatting a document.

Instructions consisted of a series of screens of information. The first screen for a particular topic always contained the title of the topic, and the last screen of a topic always contained the word *END*. The instructions could be viewed only one screen at a time. The contents of a screen became visible when the subject held down the space bar. When the space bar was not held down, an outline of the instructions appeared on the Burroughs display. The outline consisted of rows of x's where each row corresponded to a screen. Each row that represented the first screen of information for a particular topic (such as retrieving a document) consisted of the title of the topic rather than x's. This allowed the subjects to keep visual track of where they were in the instructions. In addition, one row in the outline was always at a higher intensity than the others. This row corresponded to the screen that would appear if the subject pressed the space bar.

Subjects could go forward or backward through the screens by pressing the Next-Page key or the Previous-Page key. They could go to the beginning or end of the instructions by pressing the Up-Arrow key or the Down-Arrow key. Subjects' movements through the instructions and the amount of time they held down the space bar were saved in the computer's memory. The amount of time the space bar was held down serves as a measure of subjects' reading time for a particular screen.

Deletion: Training and Test Phases

After learning how to use the Burroughs workstation to read instructions, subjects were shown the first document on which they were to work. The document was marked up to show the various changes that were to be made. The name of the document was printed in the upper left hand corner because the name was needed in order to retrieve the document.

Prior to doing a task for the first time (e.g., retrieving a document) in the training phase, each subject read the instructions for that task all the way through. This was done to make sure subjects saw all the steps for the procedure at least once and would, therefore, be less tempted to guess about how to do a step later on. Once a particular set of instructions had been read once, subjects did not have to read them again unless they wanted to. Subjects could consult the instructions at any time after having read them once.

For subjects doing deletion tasks, each of the first three documents consisted of the following tasks in order: (a) retrieve the document, (b) delete a word, (c) insert a phrase, (d) delete a sentence, (e) insert a phrase, (f) delete a paragraph, (g) insert a phrase, (h) delete a word, (i) insert a phrase, (j) delete a sentence, and (k) exit the document.

The phrase insertions were always seven words long. The word deletions never involved a word whose last letter also occurred earlier in the word. This constraint was added so that GI subjects, who would delete words by typing the last letter of the word in response to the "Delete what?" prompt, would not gain experience typing a target character (i.e., the character that is typed in response to the "Delete what" prompt) multiple times for text other than paragraphs. The need for multiple specifications of a target, other than in paragraphs, is one of the deletion features that SI subjects had to infer for tasks in the test phase. Similarly, all sentences ended with a period, and no sentence contained any internal periods. The insertion and document retrieval and exiting tasks were included to make the tasks somewhat realistic and to make it less likely that SI subjects would form a generalization by virtue of being able to concentrate solely on deletion. The first three documents constituted the training phase.

The next four documents constituted the test phase; they consisted of various deletion tasks (but no insertion tasks) that were different from the word, sentence, and deletion tasks from the training phase. The deletion tasks in the test phase were constructed by combining four facts about deletion that people would need to know in order to delete text in general. The first fact was that a deletion task could begin anywhere, not just at the beginning of a word, sentence, or paragraph. The second fact was that a deletion task did not have

Figure 1. **Factors manipulated in deletion tasks in test phase.**

Test Doc. No.	Unit That Deletion Is Based On	Deletion Starts at Beginning of Unit?		Deletion Goes to End of Unit?		Deletion Requires Multiple Specification of Target?[a]		Use of Letter as Target?	
		Y	N	Y	N	Y	N	Y	N
1	Word	x		x		x[b]		x[b]	
	Sentence	x		x		x			x
	Paragraph	x		x			x		x
2	Word	x			x		x	x	
	Sentence	x			x		x	x	
	Paragraph	x			x		x		x
3	Word		x		x	x		x	
	Sentence		x		x	x		x	
	Paragraph		x		x		x		x
4	Word		x		x	x		x	
	Sentence		x		x	x		x	
	Paragraph		x		x	x		x	

[a] Multiple specifications involved pressing the target character two or three times.

[b] SI subjects deleting a word whose final letter also occurs internally in the word would be expected to press the space bar (rather than the final letter) to highlight the word and, thus, would only hit the space bar once (no multiple specification).

to include the end of some obvious unit. For example, one could delete the first few words of a sentence without deleting the rest of the sentence. The third fact was that the target character could be specified multiple times. The fourth fact was that any character (e.g., letter, number, punctuation mark) could be a target character.

Each test phase document contained five deletion tasks, two involving words, two involving sentences, and one involving a paragraph. The just-mentioned facts about deletion were applied to these units, although not factorially. The test tasks were presented in one of two orders. However, this order manipulation had no obvious effect and is not discussed further. Figure 1 summarizes the construction of the deletion tasks in the test phase.

Formatting: Training and Test Phases

For subjects doing format tasks, the three training documents each consisted of the following tasks: (a) retrieve the document, (b) change the line spacing, (c) insert a phrase, (d) change the top line distance, (e) insert a phrase, (f) change the bottom line distance, and (g) exit the document.

Subjects were shown the first marked-up document and were explicitly told that the line spacing, top line, and bottom line tasks were formatting tasks.

Figure 2. **Factors manipulated in format tasks in test phase.**

Test Doc. No.	Format That Task Is Based On	Accept Value (A) or Change Value (C)?		Old Format (O) or New Format (N)?	
		A	C	O	N
1	Line spacing	x		x	
	Top line	x		x	
	Bottom line	x		x	
2	Margins		x		x
	Pitch		x		x
3	Line spacing	x		x	
	Top line	x		x	
	Bottom line	x			x
	Margins	x			x
	Pitch	x			x
4	Line spacing	x		x	
	Top line	x		x	
	Bottom line	x			x
	Margins	x			x
	Pitch	x			x

The next four documents constituted the test phase and consisted of formatting tasks that were different from those in the training phase. These tasks involved accepting old format features (e.g., line spacing, top line, bottom line) if they equalled some value or changing/accepting new format features (e.g., margins, pitch).

The formatting tasks in the test phase were constructed by combining two facts about formatting that people would need to know in order to examine and modify format parameters. The first fact was that when a format value is displayed (after reaching the appropriate submenu), pressing the Enter key causes that value to be used. It does not matter whether the value was the old value or whether the user had just typed in a new one; whatever value is there when the Enter key is pressed will be the new value. Thus, if the current value is the one the user wants, there is no need to erase it and retype. The second fact was that all formatting tasks involve going from a main menu to a submenu to a prompt for specifying/accepting the format value. The procedure is the same; only the menu choices and parameter values vary. Figure 2 shows how the formatting tasks in the test phase were constructed.

The test tasks were presented in one of two orders. Again, the order manipulation had no effect and is not discussed further.

Experimenter Intervention. All subjects were able to complete the training tasks without intervention by the experimenter. Although the GI subjects did have some trouble following the instructions, they were able to overcome their difficulties on their own. In the test phase, however, if a subject could not do a task, the experimenter provided a hint after approximately 1 min.

In retrospect, there were two standard test phase scenarios that were followed, one for deletion and one for formatting. When SI deletion subjects reached the first test deletion task that required them to delete arbitrary text from a word, they almost invariably first tried to delete the text by pressing the space bar in response to the "Delete what?" prompt. The experimenter intervened at this point and told them to highlight only the arbitrary letters. Then, SI deletion subjects typically tried to use the Arrow keys to highlight the letters. Although this approach would work, it would be very inefficient for many tasks, so the experimenter stopped them again and asked them to find a different method. Only 1 of the 14 SI deletion subjects figured out at this point that the appropriate action was to type last letter of the arbitrary text. The other 13 subjects often examined the instructions or stared at the screen. Finally, most said they simply could not do it. At this point, the experimenter asked them the following questions: (a) "What did you type in order to highlight a word?" (b) "What did you type in order to highlight a sentence?" and (c) "What is similar about these two things?" (i.e., the space and the period). Typically, SI subjects said that the things came at the end (or after) the thing they wanted to delete. Most SI subjects then had an "aha!" experience that gave them the idea to type the last letter of the arbitrary text. There were some small variations in this scenario. GI subjects never required this assistance.

It is important to note that the hints to SI subjects were designed to be a fairly minimal intervention that would help SI subjects focus on the correct aspect of the deletion task. Subjects were computer novices and ethical considerations dictated that they not be required to struggle indefinitely with poor hints in order to form the necessary generalizations. In any case, the important point here is that only one SI subject could do the novel deletion task without any hints.

For formatting, both SI and GI subjects were able to accept format values in the test phase quite successfully. However, about one third of both SI and GI subjects required a hint for the initial margins task. The problem was not doing the format task itself but rather finding "margins" on the format submenu. Most subjects simply never noticed margins on the submenu when doing the training format tasks. Thus, when they reached the first margins task, most subjects did not know where to find margins. However, because subjects had been told that all tasks written in the left-hand column of the documents were format tasks, most subjects simply went through the various menus until they saw margins. Subjects who did not do this after about 1 min were reminded that margins was a format task. This reminder prompted them to search through the menus.

2.2. Predictions

For deletion subjects during the training phase, it was predicted that SI subjects would complete the first two or three deletion tasks more quickly than GI subjects. This prediction is based on the assumption that SI subjects should be able to follow the directions exactly (except for paragraph deletions), whereas GI subjects would have to fill in the slots for the general instructions. Although this slot filing may or may not be difficult, it seemed reasonable that it would take longer than simply following directions. After the first few deletion tasks, however, it was hypothesized that both groups would have acquired deletion procedures that would work for the rest of the training phase; thus, both groups should perform about the same. In the test phase, SI subjects were predicted to have more difficulty forming the generalization for typing the end character of the to-be-deleted text compared to forming the other generalizations such as multiple specifications of the target for nonparagraphs, arbitrary beginning of tasks, arbitrary ending of tasks. Due to the difficulty in forming this first generalization, it was predicted that SI subjects would have trouble doing the first arbtirary deletion task.

For format tasks it was predicted that SI subjects would perform the first two or three training tasks more quickly than GI subjects because specific instructions can be followed exactly, whereas general instructions require slot filling and suffer from nonlinearity. Again, after the first few formatting tasks, it was assumed that both groups would perform about the same. In the test phase, the two groups were predicted to perform about the same, because it was assumed that SI subjects would have the ability to form the necessary generalizations from the specific instructions and that both SI and GI subjects would be on the same footing in determining which menu contained margins and pitch.

2.3. Analysis of Deletion Production Rules

Some of the subroutines assumed to be formed by SI deletion subjects needed to be generalized somewhat in order to be useful for tasks in the test phase. As mentioned earlier, there are four facts about deletion that must be learned in order to do the tasks in the test phase. The first was that the task could start anywhere within a unit. The productions for finding the beginning of a word, sentence, and paragraph are almost identical; the place where they differ is in the "argument" part of the goal clause and wherever else the argument occurs in the productions. An argument is usually the last term in a clause; for example, Word is the argument in the clause (GOAL Delete Word), and Delete is the predicate. See Figure 3, Sections b and c, for word and sentence productions. Prior research using production system models has

Figure 3. **Production rules for finding the beginning of deletion tasks.**

a. Subroutine for Finding Beginning of to-be-Deleted Text (GI)

```
(StartUpFindLoc          IF   ((GOAL FIND ?X)
                              (NOT (NOTE Finding ?X))
                       THEN   ((Add NOTE Finding ?X)
                              (Add STEP Get ?X)))
(FindLoc.P1              IF   ((GOAL Find ?X)
                              (NOTE Finding ?X)
                              (STEP Get ?X))
                       THEN   ((LookMSS (Task at ?HP ?VP))
                              (Add NOTE HP ?HP)
                              (Add NOTE VP ?VP)
                              (Add NOTE Found ?X)
                              (Delete GOAL Find ?X)
                              (Delete STEP Get ?X)
                              (Delete NOTE Finding ?X)))
```

b. Subroutine for Finding Beginning of Word (SI)

```
(StartUpBegWord          IF   ((GOAL Find BegOfWord)
                              (NOT (NOTE Finding BegOfWord)))
                       THEN   ((Add NOTE Finding BegOfWord)
                              (Add STEP Get BegOfWord)))
(BegWord.P1              IF   ((GOAL Find BegOFWord)
                              (NOTE Finding BegOfWord)
                              (STEP Get BegOfWord))
                       THEN   ((LookMSS (Task at ?HP ?VP))
                              (Add NOTE HP ?HP
                              (Add NOTE VP ?VP)
                              (Add NOTE Found BegOfWord)
                              (Delete GOAL Find BegOfWord)
                              (Delete STEP Get BegOfWord)
                              (Delete NOTE Finding BegOfWord)))
```

c. Subroutine for Finding Beginning of Sentence (SI)

```
(StartUpBegSentence      IF   ((GOAL Find BegOfSentence)
                              (NOT (NOTE Finding BegOfSentence)))
                       THEN   ((Add NOTE Finding BegOfSentence)
                              (Add STEP Get BegOfSentence)))
(BegSentence.P1          IF   ((GOAL Find BegOfSentence)
                              (NOTE Finding BegOfSentence)
                              (STEP Get BegOfSentence))
                       THEN   ((LookMSS (Task at ?HP ?VP))
                              (Add NOTE HP ?HP)
                              (Add NOTE VP ?VP)
                              Add NOTE Found BegOfSentence)
                              (Delete GOAL Find BegOfSentence)
                              (Delete STEP Get BegOfSentence)
                              (Delete NOTE Finding BegOfSentence)))
```

not examined the issue of argument generalization of productions; thus, the only prediction that might be made is by analogy. Because predicate generalizations appear to be reasonably easy (Bovair et al., 1990), argument generalizations should be reasonably easy. The ease in making the generalization, however, will certainly vary as a function of factors outside the production system framework such as the semantic similarity of the terms, the perceptual salience of the task, and the experience of the user with the terms. Because the procedure for finding the location of the beginning of the to-be-deleted text has a salient visual component, it seemed likely that the generalization would occur with little difficulty for SI subjects. This generalization would look like the subroutine possessed by GI subjects (see Figure 3, Section a).

The second fact that SI subjects needed to infer was that any target could be specified multiple times. The use of multiple specifications (of a punctuation mark) was represented in the procedure for highlighting a paragraph but was not part of the procedure for words or sentences. One of the modeling constraints based on the work of Singley and Anderson (1988) and Foltz et al. (1988) was that if productions were learned for a procedure, they did not have to be relearned for another procedure. However, the relevant productions possessed by SI subjects were for paragraph highlighting; thus, again there is the issue of argument generalization here. In addition, there was not another multiple-press procedure in the word or sentence deletion procedures that could be compared to the paragraph procedure in order to form the generalization. Thus, it was predicted that this second deletion fact would be more difficult to generalize than the first fact of starting the deletion task at any arbitrary point.

SI subjects also had to infer that a deletion task does not have to go to the end of the unit. The paragraph highlighting productions do deal with highlighting by stages, yet this is not the same thing as deleting part of a paragraph. Thus, as in the previously mentioned case, it was predicted that this generalization would be relatively more difficult than the generalization of starting the task at any arbitrary point. However, there is a strong perceptual component involved here, seeing how much text has been highlighted, that may help SI subjects form the generalization.

Finally, SI subjects had to infer that any character could be used as a target in response to the "Delete what?" prompt. The subroutine for highlighting a paragraph is tantalizingly close to the subroutine for highlighting arbitrary text that GI subjects possess (see Figure 4, Section a, for general text highlight productions). One crucial difference is that the paragraph subroutine involves finding the final punctuation mark rather than the final character (called "endchar" in the productions in Figure 4, Section a). A second difference is that the paragraph productions contain the goal clause to highlight a paragraph, whereas the text productions contain the goal clause to highlight

Figure 4. **Production rules for highlighting text.**

a. Subroutine for Highlighting Text (GI)

```
(GnStartUpHilEntity    IF   ((GOAL Hilight ?X)
                             (NOT (NOTE Hilighting ?X))
                     THEN   ((Add NOTE Hilighting ?X)
                             (Add STEP Find EndChar)))
(GnHilEntity.P1        IF   (GOAL Hilight ?X)
                             (NOTE Highlighting ?X)
                             (STEP Find EndChar))
                     THEN   ((LookMSS (EndChar ?EndChar))
                             (Add NOTE EndChar ?EndChar)
                             (Delete STEP Find EndChar)
                             (Add STEP Press EndChar)))
(GnHilEntity.P2        IF   ((GOAL Hilight ?X)
                             (STEP Press EndChar)
                             (NOTE EndChar ?EndChar))
                     THEN   ((DoKeystroke ?EndChar)
                             (Delete STEP Press EndChar)
                             (Add STEP Test EndOfHilighting)))
(GnHilEntity.P3        IF   ((GOAL Hilight ?X)
                             (STEP Test EndOfHilighting))
                     THEN   ((LookScreen (EndOfHilighting ?Correct))
                             (Add NOTE EndOfHilighting ?Correct)
                             (Delete STEP Test EndOfHilighting)))
(GnHilEntity.P4        IF   ((GOAL Hilight ?X)
                             (NOTE Hilighting ?X)
                             (NOTE Endchar ?Endchar)
                             (NOTE EndOfHilighting Correct))
                     THEN   ((Delete GOAL Hilight ?X)
                             (Delete NOTE Hilighting ?X)
                             (Delete NOTE EndOfHilighting Correct)
                             (Add NOTE Hilighted ?X)
                             (Delete NOTE EndChar ?EndChar)))
(GnHilEntity.P5        IF   ((GOAL Hilight ?X)
                             (NOTE EndOfHilighting Short))
                     THEN   ((Add STEP Press EndChar)
                             (Delete NOTE EndOfHilighting Short)))
```

b. Subroutine for Highlighting a Word (SI)

```
(SpStartUpHilWord      IF   ((GOAL Hilight Word)
                             (NOT (NOTE Hilighting Word)))
                     THEN   ((Add NOTE Hilighting Word)
                             (Add STEP Press SpaceBar)))
```

(Continued)

Figure 4. *(Continued)*

```
(SpHilWord.P1        IF     ((GOAL Hilight Word)
                             (NOTE Hilighting Word)
                             (STEP Press SpaceBar))
                     THEN   ((DoKeystroke SpaceBar)
                             (Delete STEP Press SpaceBar)
                             (Delete NOTE Hilighting Word)
                             (Delete GOAL Hilight Word)
                             (Add NOTE Hilighted Word)))
```

c. Subroutine for Highlighting a Sentence (SI)

```
(SpStartUpHilSen     IF     ((GOAL Hilight Sentence)
                             (NOT (NOTE Hilighting Sentence)))
                     THEN   ((Add NOTE Hilighting Sentence)
                             ((Add STEP Press Period)))
(SpHiliteSen.P1      IF     ((GOAL Hilight Sentence)
                             (STEP Press Period))
                     THEN   ((DoKeystroke Period))
                             (Delete STEP Press Period)
                             (Delete GOAL Hilight Sentence)
                             (Delete NOTE Hilighting Sentence)
                             (Add NOTE Hilighted Sentence)))
```

text. Although these differences seem small, no basis exist for the SI subjects to bridge them based on the generalization constraints discussed earlier. That is, a parallel set of productions that could be generalized with the paragraph deletion productions does not exist.

Because the word and sentence highlighting productions (see Figure 4, Sections b and c) in this study were so different from the paragraph highlighting productions, it was predicted that SI subjects would have the most difficulty forming the endchar generalization. In other words, because neither the word nor the sentence highlighting productions line up with the paragraph highlighting productions, it was expected that SI subjects would not be able to form a generalization for typing an end character (i.e., they would not be able to form general productions for this aspect of deletion).

2.4. Analysis of Format Production Rules

For format tasks, the production rule analysis predicts that SI subjects should be able to form generalizations fairly easily. The subroutine for selecting format menu items for SI subjects essentially consisted of three small sets of three productions each: one set each for line spacing, top line, and bottom line. These sets of productions were identical except for the argument in the goal clause and an argument in one of the other clauses.

One fact that all format subjects had to learn in the test phase was that a current parameter value could be accepted by pressing the Enter key when the value was displayed; it did not have to be erased and retyped. The subroutine for specifying a new parameter value was assumed to be the same for GI and SI subjects. This subroutine could be adapted to work for accepting old values by essentially deleting one of the productions and slightly altering another. Foltz et al. (1988) showed that people can quickly learn a new procedure if it is a shorter version of one they already know. In addition, Singley and Anderson (1988) demonstrated that an old procedure can be used in the service of a new goal. The current situation combines these two conditions by having a subset of an old procedure being used in the service of a new goal. Although the necessary changes to the productions involve some modifications that are not predicted by the model, the major hypothesis here was that the two groups would do equally well in making the modifications.[4]

2.5. Results and Discussion

Deletion

The two major deletion results are straightforward: (a) SI subjects were able to do the initial tasks more quickly than GI subjects, and (b) GI subjects were able to do the first novel task in the test phase (deleting arbitrary text from a word) quite easily, whereas SI subjects were essentially incapable of doing the task unless they received hints. The time to do each task was measured from when the Function key was pressed until the appropriate text was highlighted. Figure 5 presents a summary of the results.

One difficulty with the task times in Figure 5 is that they include the time subjects spent reading the instructions once they began the actual deletion task. Recall that subjects could refer back to the instructions whenever they wished. After the first three or four deletion tasks, subjects typically stopped referring to the instructions. However, because it was not originally anticipated that subjects might switch so often between reading and executing the instructions during the first few tasks, no provision was made for calibrating the clocks in the two computers used to collect the reading times and performance times. Thus, it would be very difficult to remove the reading time from the performance times. However, it is at least possible to get a pure measure of how long subjects initially spent reading the deletion instructions (i.e., before they began the first deletion task). GI subjects spent an average of 83 s reading the instructions, with 24 of those seconds being spent on the screen that explained the target search (Screen 7; see Appendix A). SI subjects initially spent 64 s reading the word deletion instructions (they did not read the sentence or paragraph deletion instructions until they reached the first sentence and paragraph deletion tasks). The difference in the initial time

---

[4] A fuller treatment of the production rule analysis can be found in Catrambone (1988).

Figure 5. **Time to perform deletion tasks in Experiment 1 (in seconds).**

	SI Subjects ($n = 14$)	GI Subjects ($n = 14$)
Training phase		
First (delete word)	35.8	55.2
Second (delete sentence)	20.5	49.5
Third (delete paragraph)	26.4	30.4
Fourth (delete word)	26.4	13.8
Fifth (delete sentence)	11.8	9.6
Remainder of deletion tasks in training phase (average)	9.4	8.8
Test phase		
First word with arbitrary letters in beginning or middle	139.3	10.7
First Sentence with arbitrary letters in beginning or middle	16.5	9.6
Second word with arbitrary letters in beginning or middle	9.3	11.9
Second sentence with arbitrary letters in beginning or middle	12.3	8.4
Remainder of test phase tasks (average)	9.5	9.9

spent reading by GI and SI subjects is significant, $t(26) = 3.62$, $p < .01$ (two-tailed). This suggests that the target search concept was difficult for GI subjects to comprehend and accounts for the difference in reading times. However, these times must be interpreted cautiously because subjects knew they could refer back to the instructions whenever they wished and, thus, may not have read them carefully the first time.

A stepwise multiple-regression analysis was carried out on the dependent variable of time to do each of the deletion tasks in the training and test phases. There are 980 data points, 35 (one per task) for each of the 28 subjects who did the deletion tasks.

Figure 6 presents the final equation resulting from the stepwise regression. At each step, an "F-to-enter" statistic was calculated for each predictor to determine which one would enter the equation next. The predictor with the largest F-to-enter value was entered into the equation at step n. If a predictor's F-to-enter value was below 4.0, it was never entered into the equation. All F-to-enter values greater than 4.0 correspond to about a .05 level of significance for that predictor's regression coefficient. Thus, a predictor that entered the equation, such as "position," would be a significant effect in an

Figure 6. **Regression analysis of performance on deletion tasks in Experiment 1. Predictors are listed in order of entry into the regression equation.**

Variable	Final Coefficient	Standard Coefficient	F to Remove	R^2 Total
y-intercept	11.18			
Instructions × First Arbitrary Text	124.22	.67	976.68	.468
First deletion task	37.08	.28	86.6	.535
Position	−.63	−.29	56.29	.549
Subject mean	.46	.11	27.05	.561
End of unit	7.42	.17	22.48	.570
Instructions × First Deletion Task	−18.39	−.10	11.42	.575
Single/multiple target typing	2.97	.07	8.03	.578

analysis of variance (ANOVA). The F-to-enter statistic is equal to the "F-to-remove" statistic for that predictor immediately after the predictor has been entered into the equation. The F-to-remove value gets adjusted after each new predictor enters the equation. If an entered predictor's F-to-remove value fell below 4.0 after a particular step, it was removed from the equation. The F-to-remove values shown in Figure 6 are the final values after the last predictor entered the equation.

Subject means were included as a potential predictor variable in order to handle the repeated measures design. The predictors that ended up in the final equation are as follows:

1. *Instructions × First Arbitrary Text:* An interaction dummy variable that tests whether there was any variance in the time to delete the first arbitrary text that was due to the instructions. This variable indicates that SI subjects took significantly longer to do the first arbitrary deletion task than GI subjects.
2. *First deletion task:* A dummy variable that codes whether the deletion task was the first deletion task subjects attempted in the training phase. It indicates that subjects were significantly slower on the initial deletion tasks than on later tasks.
3. *Position:* This codes for effect of serial order. It shows that subjects became faster over deletion tasks. An examination of the results suggests that GI subjects might have sped up more than SI subjects once both groups possessed the deletion procedure. However, the interaction of position and instruction type was not significant.
4. *Subject mean:* This represents a subject's average time to do training deletion tasks.
5. *End of unit:* This dummy variable codes whether the deletion task went

to the end of the unit. It indicates that subjects slowed down when deleting only part of a word, sentence, or paragraph. This variable is unfortunately confounded with the fact that the first time that a deletion task did not go to the end of a unit was also the first time subjects had to delete arbitrary text in a word. However, the fact that this variable entered the equation even after Instructions × First Arbitrary Text entered the equation suggests that this variable still had an effect.

6. *Instructions × First Deletion Task:* This is an interaction dummy variable that tests whether there was any extra variance in the time to do the first deletion task that was due to the instructions. It indicates that GI subjects took significantly longer to do the first deletion task than SI subjects.

7. *Single/multiple target typing:* This dummy variable codes whether the deletion task required specifying the target more than once. It shows that tasks requiring multiple specifications took longer than those only requiring a single specification of the target. This effect is probably partly due to the time subjects required to check the screen, to realize not all the text was highlighted, and then to respecify the target.

Potential predictor variables that did not end up in the final equation were: type of instructions, time to type practice paragraph, average time to do the document retrieval and exiting tasks, dummy variables for type of text (word, sentence, paragraph) and whether or not the deletion task started at the beginning or in the middle of the to-be-deleted text, and interaction variables (with instructions) for those dummy variables.

Ultimately, about 58% of the variance in individual subject's time on each deletion task was accounted for by the final equation.

It is also worthwhile to consider again some of the predictors that did not enter the equation. It did not matter whether a task started at the beginning or in the middle of an entity. This suggests that SI subjects were able to form an argument generalization about beginning deletion tasks at the first character of the to-be-deleted text. In addition, the fact that the predictors *end of unit* and *single/multiple target typing* did not interact with instructions suggests that these factors were no more problematic for SI subjects than for GI subjects. This is taken as preliminary support for the speculation that argument generalizations in a production rule model are about as easy as predicate generalizations.

These results suggest some changes to the constraints for doing a production rule analysis of training materials. One is that if two productions are identical, except for the argument in the goal clause and wherever else that term occurs in the productions, then a generalized production can be formed fairly easily. However, argument generalization—and perhaps predicate generalization—may depend on variables such as the knowledge of the learner about the terms being generalized and the perceptual salience of the outcome of the procedures. For example, SI subjects could see where the cursor was when they moved it to the beginning of the text they wanted to delete or when the highlighting reached the target. This may have helped them generalize the subroutines for finding the beginning of the deletion task for words, sentences, and paragraphs.

A second change in the constraints is that, under some circumstances, part of a procedure can be imported to another procedure even if the arguments in the productions are different. To use the terminology from past work, part of an old procedure (Foltz et al., 1988) can be used in the service of a new goal (Singley & Anderson, 1988). For instance, SI subjects appeared to transfer the highlight-checking productions for paragraphs to the procedures for highlighting words and sentences (it is also possible that they generalized this component to any text). This was probably facilitated because the condition for the checking subprocedure, the presence of unhighlighted text, was very salient. This transferring of a subprocedure explains why SI subjects could do multiple specifications of a target and stop deleting before the end of a unit as well as GI subjects.

The constraint changes are suggested with caution because once an SI subject formed the critical generalization about typing any target character, it is possible that the rest of the generalizations occurred simultaneously. That is, all necessary generalization may have occurred at once.

Formatting

The time to do each formatting task was determined as follows. In the training phase, time was measured from when the Function key was pressed until the new format value was typed. The measure of time to do a task in the test phase differed depending on whether or not the task was the first one for a particular document. If the task was the first one, then time was measured from when the Function key was pressed until the format value was typed in or accepted. If the task was not the first one for a document, then time was measured from when the prior task was finished until the format value was typed in or accepted for the current task. This different measure was necessary for test phase tasks because subjects could leave up the format menu during all the formatting tasks for a particular document because there were no intervening insertion tasks.

The format results indicate that SI subjects were able to get started on the initial tasks more quickly than GI subjects (see Figure 7). Both groups of subjects, however, slowed down considerably when they reached the first task involving margins. This may be due to the fact that subjects in both groups tended not to notice margins on the line-format submenu. Once they found margins on the menu, however, they were able to change or accept them quite easily. The same argument holds for pitch. However, subjects were faster on pitch because they had already struggled with margins and knew enough to

Figure 7. **Time to perform format tasks in Experiment 1 (in seconds).**

	SI Subject (n = 14)	GI Subjects (n = 14)
Training phase		
First (change line spacing)	45.7	183.7
Second (change top line)	42.7	82.8
Third (change bottom line)	40.9	32.8
Remainder of format tasks in training phase (average)	25.4	14.2
Test phase		
First task involving margins	98.7	113.7
First task involving pitch	36.5	16.9
Remainder of test phase tasks (average)	9.1	7.4

scan the menus carefully in order to find pitch. The results also suggest that after the first few training tasks, GI subjects were faster than SI subjects.

As with deletion, it would be very difficult to separate the reading time from the performance time because the computers' clocks were not calibrated. However, again, it is possible to compare how long subjects initially spent reading the format instructions (i.e., prior to beginning the first format task). GI subjects spent an average of 129 s reading the format instructions, whereas SI subjects spent 79 s reading the line spacing instructions. The difference between these times is significant, $t(26) = 6.6$, $p < .001$ (two-tailed).

Figure 8 presents the final equation resulting from the stepwise regression. There are 560 data points, 20 (one per task) for each of the 28 subjects who did the formatting tasks. The predictors that ended up in the final equation are as follows:

1. *First format task:* This dummy variable codes whether the format task was the first one that subjects did. It indicates that the initial task took significantly longer than later tasks.
2. *First margin task:* This dummy variable codes whether the format task was the first one that involved margins. It indicates that subjects took especially long on this task. This was due to subjects' difficulty in finding margins on the submenu.
3. *Instructions × First Format Task:* This interaction dummy variable was used to assess whether there was any extra variance in the time needed to do the first format task as a result of the instructions given to subjects.

***Figure 8.* Regression analysis of performance on format tasks in Experiment 1. Predictors are listed in order of entry into the regression equation.**

Variable	Final Coefficient	Standard Coefficient	*F* to Remove	R^2 Total
y-intercept	6.68			
First format task	155.04	.68	314.6	.181
First margin task	82.12	.36	186.62	.317
Instructions × First Format Task	– 139.80	– .44	135.64	.413
Position	– 1.09	– .20	50.31	.449
Subject mean	0.43	.17	34.46	.467
Instructions	26.63	.11	14.87	.474
Instructions × First Pitch Task	21.34	.07	6.31	.478

It indicates that GI subjects took significantly longer to do the first format task in the training phase than SI subjects.

4. *Position:* This refers to the serial position of a task. It shows that subjects sped up over trials.
5. *Subject mean:* This represents subjects' average time to do training format tasks.
6. *Instructions:* This dummy variable codes whether the instructions were specific or general. It indicates that GI subjects were faster overall than SI subjects.
7. *Instructions × First Pitch Task:* This interaction dummy variable was used to assess whether there was any extra variance in the time needed to do the first pitch task as a result of the instructions given to subjects. It indicates that GI subjects were faster than SI subjects on this task.

The following potential predictor variables did not end up in the final equation: time to type practice paragraph, average time to do the document retrieval and exiting tasks, a dummy variable for the first task that involved accepting the current format value, and interaction variables (with instructions) for several dummy variables.

Ultimately, about 48% of the variance was accounted for in the final equation.

It is clear that the first format task and the first task dealing with margins were difficult for both groups. The fact that "accept/change" did not enter into the regression as a predictor suggests that subjects were able to infer the new procedure—accepting a format value rather than changing it—probably because it was a subset of an already learned one (Foltz et al., 1988), despite the fact that it required an argument generalization.

It was not predicted that GI subjects would be faster than SI subjects overall. However, this result echoes the finding from the deletion training phase data, although the trend there was not significant. It was also not predicted that SI subjects would be slower on the first pitch task compared to GI subjects. These last two results suggest that there may be an advantage in learning general instructions from the start rather than inferring them. Perhaps the extra productions possessed by SI subjects (the original specific instructions plus the generalized ones) slowed performance by reducing activation to individual productions (Anderson, 1983; Singley & Anderson, 1988). This reduced activation may have slowed down the execution and later proceduralization of the general procedure.

Overall, the results from the deletion and formatting tasks indicate that GI subjects took longer to get started than SI subjects. For deletion tasks, however, this penalty was offset by GI subjects' ability to handle novel tasks, whereas SI subjects were unable to form an important generalization. Conversely, formatting does not require any difficult generalizations and, thus, SI subjects fared as well as GI subjects on novel tasks.

Although the results suggest that there might be a long-term benefit to general instructions, there is a suggestion that if specific instructions can be written to allow the necessary generalizations to be made easily, then they might be more beneficial than general ones because initial learning time with specific instructions is shorter. Thus, there are two interesting issues to consider: (a) how to write specific instructions to aid the formation of general productions and (b) whether general instructions typically produce better long-term performance than specific ones. The first issue is the focus of the second experiment, although the second issue is also briefly discussed.

3. EXPERIMENT 2: MODIFYING SPECIFIC INSTRUCTIONS TO SUPPORT GENERALIZATION

Experiment 1 suggests that it might be possible to characterize the difficulty of forming generalizations from specific instructions through a production rule analysis. The analysis used in Experiment 1 indicates that it would be difficult for SI deletion subjects to form the appropriate generalization for highlighting arbitrary text for deletion. The most general procedure they possessed (i.e., deleting paragraphs) was sufficiently different from the word and sentence deletion procedures that a crucial generalization could not be formed. The word deletion procedure included a step for pressing the space bar in order to highlight a word, whereas the sentence deletion procedure included a step for pressing the Period key. These steps were not explained in terms of finding the target character and pressing it as many times as was necessary.

Conversely, perhaps the general instructions included unnecessary information. GI subjects learned that the cursor advanced to the first occurrence of the target character each time it was pressed and, therefore, concluded that the target might have to be pressed several times in order to highlight the appropriate text. GI subjects typically spent a relatively large amount of time reading this piece of information, probably because the idea of a "string search" was foreign to them. A simpler way of explaining the general procedure might be simply to tell the learners to press the end character of the text they want to delete over and over again until it is highlighted. This is a general procedure that does not explain cursor advancement or why the target needs to be pressed more than once. However, it contains the essential information and is much more similar to the wording used in the instructions for highlighting a paragraph that the SI subjects received. The only difference being that the general instructions say to type the end character, whereas the paragraph instructions say to type the final punctuation mark.

With respect to the issues discussed earlier, there were two primary goals of Experiment 2. The first was to examine whether the specific instructions could be altered so that SI subjects would be able to form a generalization about typing the end character in order to highlight arbitrary text. The alteration was based on the production rule analysis. The SI deletion subjects in Experiment 1 were assumed not to form a general highlighting procedure because there was no procedure with which the paragraph procedure could be generalized. However, if the instructions of the word highlighting procedure were changed from pressing the space bar to typing the last letter of the word, then SI subjects might be able to form the appropriate generalization concerning pressing the target character in response to the "Delete what?" prompt. Both the word and paragraph deletion procedures would then contain productions for typing either the final letter or final punctuation mark of the entity to be deleted. These productions could then be argument generalized into typing the final character of any entity to be deleted. In addition, the results from Experiment 1 suggest that this particular generalization was the major hurdle for SI subjects to overcome. Thus, with the changes in the instructions for deleting a word, SI subjects in Experiment 2 would be expected to be able to complete the initial arbitrary text deletion task as successfully as GI subjects. Only deletion instructions were used in Experiment 2 because the specific format instructions were already known to produce generalizations fairly easily.

The second goal of Experiment 2 was to see if the time subjects initially spend reading the general deletion instructions could be reduced. This was attempted by changing the wording of the general highlighting procedure to make it identical with the paragraph highlighting procedure except that the general procedure would refer to an end character, whereas the paragraph procedure would refer to a punctuation mark. It was assumed that this change would allow the GI subjects to comprehend the instructions more quickly than before because the instructions did not attempt to explain a potentially unnecessary concept. However, this is still a general description and could take GI subjects a while to comprehend because they would have to decide what "end character" means and make sense of the notion of typing the target character several times. Initially, this concept might not seem to make much sense to GI subjects because the first deletion task would consist of a word whose final letter occurs only once.[5]

A secondary goal of the experiment was to examine whether general instructions would produce a long-term benefit compared to specific instructions. This benefit was found for formatting in the first experiment but only weakly for deletion.

3.1. Method

Subjects. Subjects were 15 female undergraduates attending the University of Michigan who had very limited or no computer experience (based on self-reports).

Materials. The instructions were almost identical to those used in Experiment 1 except for two changes. First, an observation of the production rules used to represent the word deletion instructions indicated that those productions could be made almost identical to those for deleting a paragraph if the word deletion instructions required the learner to type the last letter of the word in response to the "Delete what?" prompt rather than to press the space bar (see Figure 9). Thus, Screen 7 of the specific deletion instructions was altered to read: "Type the last letter of the word you want to delete" rather than saying to press the space bar. The only difference in the word and paragraph productions is that the former refer to a word and last letter rather than paragraph and punctuation mark. Because these productions are identical except for the arguments (and the associated concepts, last letter and punctuation mark) and the final clause of the third production (which, for paragraphs, adds the step to check whether the entire paragraph had been highlighted), it was assumed, based on the constraints, that general productions were more likely to be formed by SI subjects in Experiment 2 than they were in Experiment 1. Thus, it was expected that SI subjects would be able to delete novel types of text.

The second change in the instructions was that Screen 7 of the general deletion instructions was altered to read: "Type the last character at the end of the text you want to delete, typing it over and over again until the text is highlighted." Thus, the explanatory material for deletion was removed.

---

[5] In fact, after the experiment, one GI subject commented that she initially thought that the instructions implied that the computer would not always "catch it" when the end character was pressed, and so it needed to be pressed multiple times in order to give the computer multiple chances.

Figure 9. **Partial list of production rules for the new word highlighting procedure and the paragraph highlighting procedure.**

a. Word

```
(SpStartUpHilWord  IF ((GOAL Hilight Word)
                       (NOT (NOTE Hilighting Word)))
                 THEN ((Add NOTE Hilighting Word)
                       (Add STEP Find LastLetter)))
(SpHilWord.P1      IF ((GOAL Hilight Word)
                       (NOTE Highlighting Word)
                       (STEP Find LastLetter))
                 THEN (LookMSS (LastLetter ?LastLetter))
                       (Add NOTE LastLetter ?LastLetter)
                       (Delete STEP Find LastLetter)
                       (Add STEP Press LastLetter)))
(SpHilWord.P2      IF ((GOAL Hilight Word)
                       (STEP Press LastLetter)
                       (NOTE LastLetter ?LastLetter))
                 THEN ((DoKeystroke ?LastLetter)
                       (Delete STEP Press LastLetter)))
```

b. Paragraph

```
(SpStartUpHilPara  IF ((GOAL Hilight Paragraph)
                       (NOT (NOTE Hilighting Paragraph)))
                 THEN ((Add NOTE Hilighting Paragraph)
                       (Add STEP Find PunctuationMark)))
(SpHilPara.P1      IF ((GOAL Hilight Paragraph)
                       (NOTE Highlighting Paragraph)
                       (STEP Find PunctuationMark))
                 THEN ((LookMSS (PunctuationMark ?PunctuationMark))
                       (Add NOTE PunctuationMark ?PunctuationMark)
                       (Delete STEP Find PunctuationMark)
                       (Add STEP Press PunctuationMark)))
SpHilPara.P2       IF ((GOAL Hilight Paragraph)
                       (STEP Press PunctuationMark)
                       (NOTE PunctuationMark ?PunctuationMark))
                 THEN ((DoKeystroke ?PunctuationMark)
                       (Delete STEP Press PunctuationMark)
                       (Add STEP Test EndOfHilighting)))
```

Procedure. The procedure was very similar to that used in Experiment 1. The major change involved when subjects were allowed to read the instructions. As in Experiment 1, subjects had to read the instructions for a particular topic all the way through before they could attempt a task for the first time. However, in order to get a purer measure of performance time, subjects were not allowed to refer to just part of the instructions at any arbitrary time once they began a task. Rather, if a subject made a mistake that she did not recognize, or from which she could not easily recover, then she had to go back and reread all the instructions on that topic. Meanwhile, the experimenter undid whatever work the subject had done on that task, thereby forcing the subject to redo the task from the beginning. This approach has the added benefit of making the reading time prior to the first task a more sensitive measure of how easily the instructions were comprehended by the subjects. It was assumed that these subjects would do the initial reading more carefully than subjects in Experiment 1 because they were forced to perform the tasks from memory.[6] The time to do a task was measured from the first attempt at the task until it was successfully completed. The clocks on the two computers (one for reading the instructions and one for doing the tasks) were calibrated so that, if desired, the time spent reading could be easily separated from the time spent executing the procedure.

Subjects edited the same deletion documents as in Experiment 1. Group 1 ($n = 5$) was the GI group. There were two SI groups that differed only in the order in which they did the test phase tasks. It was noted earlier that argument generalization might depend to some extent on the salience of the highlighting on the computer screen. Thus, if during the training phase an SI subject only deleted words whose final letter did not occur elsewhere in the word, then the subject might not easily form the generalization that any target letter (besides the final letter of a word) can be specified in response to the "Delete what?" prompt. Thus, when deleting a word such as *mysterious,* she might be surprised to discover that only part of the word becomes highlighted when she types *s*. She might have expected the entire word to become highlighted because *s* was the last letter. The data from Experiment 1, however, indicate that SI subjects were able to generalize the use of multiple specifications to all types of text; therefore, this hypothetical subject, presumably, would have little difficulty in deciding to press *s* again and also would have seen that only part of a word can be highlighted. Thus, this subject would be likely to delete successfully the first three letters of *swkvarious* partly due to the instructions but also partly due to her experience with *mysterious*.

Conversely, consider an SI subject whose first nonstandard deletion task is to delete the first three letters of *swkvarious*. If the subject has formed the necessary generalization, as the production rule analysis predicts, then she should quickly press *k* in response to the "Delete what?" prompt. However, if

---

[6] Kieras and his colleagues (Kieras & Bovair, 1986; Polson & Kieras, 1985) typically required their subjects to read some instructions and then attempt to execute them. If a subject made an error, he or she had to reread the instructions for that procedure. This process continued until the subject could perform the task flawlessly for a certain number of trials. Although subjects in Experiment 2 were only made to do each task correctly once, this paradigm does provide more rigor than Experiment 1, but at the cost of some naturalness.

Figure 10. **Time to perform deletion tasks in Experiment 2 (in seconds).**

	SI Subjects (n = 10)	GI Subjects (n = 5)
Training phase		
First (delete word)	60.5	116.7
Second (delete sentence)	9.6	23.3
Third (delete paragraph)	14.8	14.5
Fourth (delete word)	12.7	6.4
Fifth (delete sentence)	7.7	6.8
Remainder of deletion tasks in training phase (average)	6.7	8.6
Test phase		
First word with arbitrary letters in beginning or middle	7.7	6.9
First sentence with arbitrary letters in beginning or middle	15.6	7.0
Second word with arbitrary letters in beginning or middle	6.1	6.0
Second sentence with arbitrary letters in beginning or middle	7.8	10.8
Remainder of test phase tasks (average)	12.1	11.0

she has not formed the generalization, she might show a long delay in getting the appropriate text deleted because she might not realize that a nonfinal letter can be pressed.

Based on these considerations, SI subjects received the test phase documents in one of two orders. Group 2's (n = 5) first test phase document contained, among other things, word deletion tasks in which the final letter of the word also occurred internally (e.g., *mysterious*). Group 3 (n = 5) received the document beginning with the swkvarious deletion task as their first test phase document.

In sum, Groups 1 and 2 received the test documents in the order 1, 2, 3, 4, and Group 3 received the test documents in the order 2, 1, 3, 4 (see Figure 1).

3.2. Results and Discussion

As in the first experiment, SI subjects were able to do the initial deletion tasks more quickly than GI subjects. However, unlike the first experiment, SI subjects were able to do the first novel deletion task (deleting arbitrary text from the beginning of a word) about as fast as GI subjects. Figure 10 presents a summary of the results.

Figure 11. **Regression analysis of performance on deletion tasks in Experiment 2. Predictors are listed in order of entry into the regression equation.**

Variable	Final Coefficient	Standard Coefficient	F to Remove	R^2 Total
y-intercept	3.71			
First deletion task	106.81	.941	280.65	.371
Instructions × First Deletion Task	−54.19	−.392	48.72	.427
Subject mean	0.37	.106	10.33	.436
Instructions × Beginning/Middle	4.01	.083	6.32	.442
Paragraph	3.61	.076	5.45	.448

There were no significant differences between the two groups of SI subjects. Group 3 was about as fast as Group 2 on the swkvarious task (8.9 s vs. 6.6 s, respectively). This suggests that the argument generalization occurred (i.e., **last letter** and **punctuation mark** generalized to **end character**). Thus, these two groups are collapsed together in Figure 10. However, it should be noted that, because of the small number of subjects, the lack of a difference between the groups should be interpreted with caution.

GI subjects spent an average of 114 s initially reading the instructions, whereas SI subjects spent 74 s initially reading the word deletion instructions. This difference is marginally significant, $t(13) = 2.09$, $p < .07$ (two-tailed) and suggests that the general instructions still may have been more difficult to comprehend than the specific instructions for deleting a word.

A multiple stepwise-regression analysis was carried out on the dependent variable of time to do each of the deletion tasks in the training and test phases. The set of potential predictor variables was the same as in Experiment 1. The predictors that ended up in the final equation are presented in Figure 11. There are 525 data points, 35 (one per task) for each of the 15 subjects who did the deletion tasks.

The interaction variable of Instructions × First Arbitrary Text did not enter the equation. This suggests that, unlike Experiment 1, SI subjects did not take significantly longer than GI subjects to do the first arbitrary deletion task in the test phase. Once again though, the instructional manipulation accounts for a good deal of variance for the first deletion task in the training phase. The SI subjects got started more quickly than Gi subjects, as shown by the change in R^2 associated with the interaction variable of Instructions × First Deletion Task. However, it is also worth noting that there was one outlier in the GI group who took 249.8 s to do the first deletion task.

Removing this person drops the GI average from 116.7 s to 83.4 s. Nevertheless, this is still more than 20 s slower than the SI average.

There were three other predictors that did not appear in the final equation in Experiment 2 but did appear in Experiment 1's. The first was serial position. This suggests that subjects in Experiment 2 did not speed up as much as those in Experiment 1. The second and third were end of unit and single/multiple target typing, respectively. This suggests that, unlike Experiment 1, it did not affect subjects' performance very much whether the deletion task went to the end of a unit or whether it required multiple specifications of the target.

Two predictor variables entered this equation that did not enter it in Experiment 1:

1. *Instructions* × *Beginning/Middle:* This interaction dummy variable codes for whether there was any extra variance due to instructions for deletion tasks that started at the beginning versus the middle of a unit. It indicates that SI subjects tended to slow down more than GI subjects when a task started in the middle of a unit.
2. *Paragraph:* This dummy variable codes for whether the deletion task involved a paragraph or something else. This suggests that the extra time needed to press the target and then check to see if the paragraph was highlighted was not trivial. In addition, in the test phase, one of the paragraphs ended with a question mark and another with a quotation mark. Subjects took longer to do these deletions because they often forgot to press the Shift key in order to specify those target characters or, as some subjects reported, they did not think the question mark or the quotation mark was part of the paragraph. For some reason this did not occur in Experiment 1.

Ultimately, about 45% of the variance in individual subject's time on each deletion task was accounted for by the final equation.

A comparison of the deletion results in Experiment 1 to those in Experiment 2 shows some interesting differences besides the obvious one of SI subjects in Experiment 2 being able to do the first arbitrary text deletion task with little difficulty. Both SI and GI subjects in Experiment 2 tended to spend more time on the initial reading of the deletion instructions compared to subjects in Experiment 1 (Experiment 1: SI = 64 s, GI = 83 s; Experiment 2: SI = 74 s, GI = 114 s). A series of t tests showed no significant difference between SI subjects in the two experiments, $t(22) = 1.46$, $p < .2$, but there was a marginal effect for GI subjects, $t(17) = 2.07$, $p < .07$ (two-tailed). It was expected that GI subjects in Experiment 2 would reduce their reading time compared to GI subjects in Experiment 1 because the revised instructions contained less explanatory information. However, it is likely that this trend toward longer reading times in Experiment 2 is due to the fact that subjects in Experiment 2 knew they were going to have to do the tasks from memory and, thus, read the instructions more carefully the first time through.

Another interesting difference between the experiments is that both SI and GI subjects took longer to do the first deletion task in Experiment 2 (GI: Experiment 1 = 55.2 s, Experiment 2 = 116.7, $t(17) = 10.12, p < .001$; SI: Experiment 1 = 35.8 s, Experiment 2 = 60.5, $t(22) = 4.42, p < .001$). There are several likely reasons for this. First, as mentioned before, subjects in Experiment 2 had to work from memory. If they made a mistake, they had to reread the entire set of instructions for that task and redo the task from the beginning. In Experiment 2, GI subjects spent an average of 49 s rereading the instructions, and SI subjects spent 41 s. Due to the clock calibration problem in Experiment 1, it would be very difficult to determine the rereading times for subjects in the first experiment.

A factor that affected the performance times of SI subjects across the two experiments was that SI subjects in Experiment 2 were told to press the last letter of the word they wanted to delete in response to the "Delete what?" prompt rather than simply to press the space bar. The latter instruction is quite simple, whereas the former requires a bit of problem solving and, thus, may take longer to process and execute. It is probably not the case that SI subjects had difficulty determining what the last letter was, but rather they may have had doubts about whether that was really what they were supposed to do.

It is also worth noting that the difference in word deletion instructions for SI subjects in the two experiments resulted in different word deletion tasks for those subjects. That is, SI subjects in Experiment 1 deleted words and the space following them, whereas SI subjects in Experiment 2 deleted just the words. This difference in tasks may have produced learning and generalization differences by making the end character differentially salient. The degree to which instructional differences and task perception differences influenced generalization cannot be disentangled in the current study but should be examined in future work.

A factor that affected the comparison of GI subjects' initial performance across experiments was that GI subjects in Experiment 2 did not receive an explanation for why they were told to type the final character of the text they wanted to delete "over and over again until the text is highlighted." Because their first task was to delete a word (whose final character only occurred once), there would be no obvious reason why the final letter should or should not be pressed multiple times. GI subjects in Experiment 1 were given an explanation for why the final character might need (or not need) to be pressed more than once; this may have helped them do the first word deletion more quickly (cf. Smith & Goodman, 1984). This factor probably influenced the initial reading time of GI subjects. Conversely, SI subjects in both experiments were not exposed to the idea of multiple target specification until they had to delete a paragraph. Because they deleted their first paragraph after having just deleted a sentence by typing a period, and because they could see that the paragraph was made up of several sentences, it presumably made sense to them that the final punctuation mark of the paragraph should be pressed several times. They were ready for this idea.

Finally, the fact that instructions interacted with whether or not a deletion task began at the beginning or middle of a unit in Experiment 2, but not in Experiment 1, suggests that this result might not be reliable. A similar conclusion could be drawn from the inconsistent effects of multiple specifications and whether or not the text was a paragraph.

Regarding a separate issue, it was proposed, based on the results from Experiment 1, that SI subjects might be slower than GI subjects in the long run because SI subjects would have more productions. However, this did not occur. Thus, it remains unclear whether there is a long-term execution and/or proceduralization cost to learning from specific instructions, at least in the situations described here.

In sum, Experiment 2 suggests that specific instructions can be used to promote generalization while at the same time allowing the learner to get started on the basic tasks fairly quickly.

4. GENERAL DISCUSSION

The two experiments presented here were designed to test the conditions under which general or specific instructions are more effective for teaching novices how to do procedures in a particular domain. It was hypothesized that these two classes of instruction might be effective in different situations. Specific instructions were hypothesized to be more effective in domains in which the procedures were easily generalizable. General instructions were hypothesized to be more effective in cases in which the generalizations that needed to be formed could not be made easily by studying instructions for specific cases.

The results from Experiment 1 supported the previously mentioned hypotheses. The specific deletion instructions did not help subjects from a crucial generalization. As a result, all but 1 of these subjects needed help to do a novel task that required the generalization, whereas subjects who received general instructions were able to do the task easily. However, the SI subjects were able to do the initial training tasks more quickly.

Conversely, formatting seemed to be the type of procedure that would be best described by specific instructions. Again, the results supported the predictions. GI subjects spent a long time reading and executing the instructions, whereas SI subjects got started much more quickly. When going on to new tasks, SI subjects fared as well as GI subjects.

Experiment 2 indicates that specific instructions can be rewritten to allow people to form needed generalizations while still enabling them to accomplish initial tasks fairly easily. The rewriting of the instructions was guided by a production rule analysis based largely on the work of Bovair et al. (1990), Foltz et al. (1988), and by the results of Experiment 1. It is worth noting that the production rule analysis correctly predicted that a relatively small change in the instructions would change SI subjects' performance. This suggests that a production rule analysis may be a particularly useful form of task analysis because the prediction was probably unique to this type of analysis.

Clearly, the trade off between the time to comprehend instructions versus success on novel tasks needs to be investigated more thoroughly. A production rule analysis might well be an important part of this future research in terms of designing and analyzing the instructional materials. This approach could also be augmented by other theoretical tools for analyzing and improving the comprehensibility of instructions and teaching materials (e.g., Desaulniers et al., 1988; Frase, Macdonald, & Keenan, 1985; Kintsch, 1986).

4.1. A Production Rule Approach to Designing Instructions?

Although the production rule form of task analysis may still be unwieldy for practical document design, this analysis may be of use right now in cases in which instructions seem to be giving test subjects trouble. That is, a person designing instructions and testing them out on learners may be able to use a form of production rule analysis to help determine how certain parts of the instructions might be changed. Perhaps over time a computerized tool could be developed that would allow procedures to be specified easily with productions. These productions could then be examined for completeness and generalizability. Once the productions are debugged they could then be converted into instructions. Assuming that the translation from productions to instructions can be done straightforwardly, the resulting instructions should be quite good. The practicality of this approach, in terms of time and effort, would have to be examined. Nevertheless, this approach seems more precise than other types of design and documentation checking, such as the use of guidelines or standards.

The experiments presented here refute, to some extent, the pessimistic conclusion from most problem-solving studies that learners cannot solve novel problems. It has been demonstrated over and over again that novices tend not to notice the deep structure in examples (e.g., Black, Kay, & Soloway, 1986; Larkin, McDermott, Simon, & Simon, 1980; Reed et al., 1985; Seifert, McKoon, Abelson, & Ratcliff, 1986). However, the SI subjects in Experiment 2 were able to solve novel problems, presumably because the training materials helped them to generalize their procedures. A similar result was found by Catrambone and Holyoak (in press) for students learning how to solve probability problems. Their subjects solved novel problems more successfully if the training examples were constructed in such a way that enabled learners to generalize old methods.

4.2. Combining Specific and General Instructions and Examples

It is worth investigating the results of combining features of general and specific instructions and mental model information. Although specific instructions may help the novice get started more quickly, additional general instructions or mental model information could help them when they get to novel situations, even if the specific instructions are already written to support generalization (Schmalhofer, 1987; Smith & Goodman, 1984).

An issue not explored here is the use of examples in instructions. Examples may aid people in following instructions by clarifying ambiguities arising from a description of a procedure and by providing information that may be awkward to describe otherwise. For instance, it was clear that GI deletion subjects did not think of pressing the space bar in response to the "Delete what?" prompt because a space does not intuitively seem like a character. However, if GI subjects had seen an example that demonstrated searching for a space, they might have realized that, as far as the computer is concerned, a space is a character. Examples are also useful in reinforcing instructions. For instance, GI format subjects in Experiment 1 would have benefitted from seeing an example of a request for information when reading about how to make a format change. At first, many of the GI subjects seemed confused about what it was. These hypothesized effects of examples are consistent with the work of various researchers (e.g., Felker, Redish, & Peterson, 1985; LeFevre & Dixon, 1986). The effects of examples could be incorporated into the production rule model, but it would be important to use simple examples at first in order to make the modeling reasonably straightforward.

Another interesting issue to examine systematically is the possibility (suggested by Experiment 1) that the more productions there are for procedures in a particular domain, the slower the individual procedures are executed. This suggests that it is better to provide the learner with the general procedures directly rather than having him or her form them from specific instructions. This approach could cost the learner though, due to the training time differences in learning from general versus specific instructions. Perhaps the best way to write instructions would be to provide general procedures coupled with examples.

It is important to replicate and extend this work to other domains and to apply and refine the production rule analysis, including the constraints. Fairly clear claims have been made about when generalizations and importations of procedures should occur, and these claims need to be tested in many other domains under various training conditions. In addition to answering questions about how people follow instructions and learn procedures, the approach taken here might aid considerably in the development of more effective training and teaching materials.

Acknowledgments. I thank the members of my dissertation committee: Gary Olson, Ed Smith, Judy Olson, and Leslie Olsen. Thanks are also given to David Kieras, Susan Bovair, Pat Baggett, Dan Fisk, and Jack Carroll for their comments and help. Finally, I thank the members of the Doctoral Consortium at the 1987 Conference on Human Factors in Computing Systems.

Support. This research was supported by a Rackham Dissertation Grant (Serial No. E-102519) from the University of Michigan.

REFERENCES

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review, 94,* 192–210.

Black, J. B., Kay, D. S., & Soloway, E. (1986). Goal and plan knowledge representations: From stories to text editors and programs. In J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction.* Cambridge, MA: Bradford/MIT Press.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human–Computer Interaction, 5,* 1–48.

Catrambone, R. (1988). *Specific versus general procedures in instructions.* Doctoral dissertation, University of Michigan, Ann Arbor.

Catrambone, R., & Holyoak, K. J. (in press). Learning subgoals and methods for solving probability problems. *Memory & Cognition.*

Desaulniers, D. R., Gillan, D. J., Rudisill, M., & Burns, M. J. (1988, May). *The computer-based display of procedures.* Paper presented at CHI Human Factors in Computing Systems Conference, Washington, DC.

Felker, D. B., Redish, J. C., & Peterson, J. (1985). Training authors of informative documents. In T. M. Duffy & R. Waller (Eds.), *Designing usable texts* (pp 43–61). New York: Academic.

Foltz, P. W., Davies, S. E., Polson, P. G., & Kieras, D. E. (1988). Transfer between menu systems. *Proceedings of the CHI '88 Conference on Human Factors in Computing Systems,* 107–112. New York: ACM.

Frase, L. T., Macdonald, N. H., & Keenan, S. A. (1985). Intuitions, algorithms, and a science of text design. In T. M. Duffy & R. Waller (Eds.), *Designing usable texts* (pp. 97–112). New York: Academic.

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science, 8,* 191–219.

Kieras, D. E., & Bovair, S. (1986). The acquisition of procedures from text: A production-system analysis of transfer of training. *Journal of Memory and Language, 25,* 507–524.

Kintsch, W. (1986). Learning from text. *Cognition and Instruction, 3,* 87–108.

Larkin, J., McDermott, J., Simon, D. P., & Simon, H. (1980). Models of competence in solving physics problems. *Cognitive Science, 4,* 317–345.

LeFevre, J., & Dixon, P. (1986). Do written instructions need examples? *Cognition and Instruction, 3,* 1–30.

Mack, R., Lewis, C. H., & Carroll, J. (1983). Learning to use word processors: Problems and prospects. *ACM Transactions on Office Information Systems, 3,* 254–271.

Polson, P. G., & Kieras, D. E. (1985). A quantitative model of the learning and performance of text editing knowledge. *Proceedings of the CHI '85 Conference on Human Factors in Computing Systems,* 207–212. San Francisco: ACM.

Reed, S. K., Dempster, A., & Ettinger, M. (1985). Usefulness of analogous solutions for solving algebra word problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 11,* 106–125.

Schmalhofer, F. (1987). Mental model and procedural elements approaches as guidelines for designing word processing information. *Proceedings of INTERACT '87,* 269–274. New York: Elsevier.

Seifert, C. M., McKoon, G., Abelson, R. P., & Ratcliff, R. (1986). Memory connections between thematically similar episodes. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 12,* 220–231.

Singley, M. K., & Anderson, J. R. (1988). A keystroke analysis of learning and transfer in text editing. *Human-Computer Interaction, 3,* 223–274.

Smith, E., & Goodman, L. (1984). Understanding written instructions: The role of an explanatory schema. *Cognition and Instruction, 1,* 359–396.

HCI Editorial Record. First manuscript received February 27, 1989. Revision received June 14, 1989. Accepted by Clayton Lewis. —*Editor*

APPENDIX A: GENERAL DELETION INSTRUCTIONS (EXPERIMENT 1)

Screen 1: DELETING TEXT

Use the arrow keys to move the cursor so that it is under the first character of the text you want to delete.

Screen 2: Press the F4 key.

Screen 3: Notice that the following "menu" appears at the bottom of the computer screen:

Move Copy Delete Recall Save

Also notice that Move is highlighted.

Screen 4: Press the right-arrow key twice to highlight:

Delete

Screen 5: Press the ENTER key.

Screen 6: Notice that at the bottom of the screen the computer displays the question:
Delete what?

Screen 7: Move the cursor to the end of the text you want to delete.
To do this, type the last character of the text you want to delete. The cursor will automatically advance to the first occurrence of this character, highlighting all the text it passes through. If this character occurs more than once in the text you want to delete, then type the character again and again until the cursor has finally reached the last occurrence of the character, at the end of the text you want to delete.

Screen 8: If you highlighted the wrong text and you want to start over, press the ESC key. After you press the ESC key, you will have to move the cursor back to the beginning of the text you wish to delete and then press the F4 key in order to restart the deletion procedure.

Screen 9: If you highlighted the correct text, then press the ENTER key.

Screen 10: Notice that the text that was highlighted disappears. END

APPENDIX B: FORMAT INSTRUCTIONS

B1. General Format Instructions (Experiment 1)

Screen 1: CHANGING THE FORMAT OF A DOCUMENT
Suppose you have a document on the screen and you want to change some formatting feature. Use the arrow keys to move the cursor so that it is under the first letter of the first word of the document.

Screen 2: Press the F7 key.

Screen 3: Notice that the following "main menu" appears at the bottom of the computer screen:
Line_format Page_format Tabs
Also notice that Line_format is highlighted.

Screen 4: Any time you wish to make a format change, you have to first choose an item from the main menu that corresponds to the type of change you wish to make.

Screen 5: To choose an item from a menu, you do the following two things:

1) Highlight the item you want. Use the right and left arrow key to do this (unless the item is already highlighted).
2) When you have highlighted the item you want, then press the ENTER key in order to actually choose the item.

Screen 6: If you wish to change the line spacing, choose Line_format from the main menu. If you wish to change the distance of the top or bottom line of the text from the top of the paper, choose Page_format from the main menu.

Screen 7: Once you have chosen a menu item, one of two things will happen. Either a new "sub-menu" will appear at the bottom of the screen or a "request" for information will appear at the bottom of the screen.

Screen 8: If a sub-menu appears at the bottom of the screen, then choose the menu item you want using the arrow keys (if necessary) and then press the ENTER key.

Screen 9: If a "request" appears at the bottom of the screen, press the ESC key to erase the current information that is to the right of the request and then type in the new information.

Screen 10: After you type in the new information, press the ENTER key.

Screen 11: Notice that now the main menu reappears at the bottom of the screen.

Screen 12: Press the ESC key once in order to make this menu disappear.

Screen 13: You are now finished making the format change:
Note though that the document on the screen might still look the same after you make a format change; the format change would be seen if you printed out the document on paper. One general piece of advice is that if you are making a format change and want to cancel the last step you did, press the ESC key. END

B2. Specific Format Instructions for Changing the Line Spacing of a Document (Experiment 1)

Screen 1: CHANGING THE LINE SPACING OF A DOCUMENT
Suppose you have a document on the screen and you want to change the line spacing. Use the arrow keys to move the cursor so that it is under the first letter of the first word of the document.

Screen 2: Press the F7 key.

Screen 3: Notice that the following "menu" appears at the bottom of the computer screen:
Line_format Page_format Tabs
Also notice that Line_format is highlighted.

Screen 4: Press the ENTER key.

Screen 5: Notice that the following menu appears at the bottom of the computer screen:
Spacing Margins Pitch Adjust Justify
Also notice that Spacing is highlighted.

Screen 6: Press the ENTER key.

Screen 7: Notice that at the bottom of the screen the computer displays the request:
Enter line space (1, 2, or 3):
Also notice that this request is followed by a number that indicates the current line spacing.

Screen 8: Press the ESC key to erase the number that indicates the current line spacing.

Screen 9: Type a number to indicate the new line spacing.

Screen 10: Press the ENTER key.

Screen 11: Notice that the following menu appears at the bottom of the computer screen:
Line_format Page_format Tabs
Also notice that Line_format is highlighted.

Screen 12: Press the ESC key.

Screen 13: Notice that the menu has disappeared.

Screen 14: You are now finished changing the line spacing.
Note though that the document on the screen might still look the same after you change the line spacing; the change in line spacing would be seen if you printed out the document on paper. One general piece of advice is that if you are making a line spacing change and want to cancel the last step you did, press the ESC key. END