

Learning Artificial Grammars With Competitive Chunking

Emile Servan-Schreiber and John R. Anderson
Carnegie Mellon University

When exposed to a regular stimulus field, for instance, that generated by an artificial grammar, subjects unintentionally learn to respond efficiently to the underlying structure (Miller, 1958; Reber 1967). We explored the hypothesis that the learning process is chunking and that grammatical knowledge is implicitly encoded in a hierarchical network of chunks. We trained subjects on exemplar sentences while inducing them to form specific chunks. Their knowledge was then assessed through judgments of grammaticality. We found that subjects were less sensitive to violations that preserved their chunks than to violations that did not. We derived the theory of competitive chunking (CC) and found that it successfully reproduces, via computer simulations, both Miller's experimental results and our own. In CC, chunks are hierarchical structures strengthened with use by a bottom-up perception process. Strength-mediated competitions determine which chunks are created and which are used by the perception process.

The world is regular, and people are efficient regularity detectors. Sometimes people are intentionally looking for structural regularities. Other times, however, people learn to respond to structured stimuli even though they do not suspect an underlying structure. This latter phenomenon, which we think is best captured by the phrase *unintentional learning*, has been most consistently studied by using artificial grammars to generate regular stimulus fields. Such a grammar is shown in graphic form in Figure 1. In particular, two studies by Miller (1958) and Reber (1967) demonstrated the basic phenomenon.

Miller reported that subjects can memorize lists of letter strings generated by an artificial grammar faster than lists of randomly generated strings. While his subjects were kept intentionally ignorant of the generating principles underlying the two types of lists, they responded efficiently to the greater interstring similarity of grammatical strings.

Reber elaborated on Miller's experiment by following the memorization task with a discrimination task (1967, Experiment II). The combination of these two tasks is what we henceforth refer to as the *Reber task*. The general design is as follows: Subjects are first asked to memorize some letter strings that, unknown to them, are generated by an artificial grammar. After they have reached some learning criterion, the existence of the grammar is revealed, and subjects are asked to discriminate grammatical from nongrammatical strings on the basis of their experience with the memorized grammatical strings. Reber reported that subjects are able to do so efficiently, even though their ability to verbalize their knowledge of the underlying grammar is weak.

This research was supported in part by Grant 87-51890 from the National Science Foundation and in part by Contract N00014-86-K-0678 from the Office of Naval Research.

Axel Cleeremans, Don Dulany, Leigh Nystrom, Arthur Reber, Kurt VanLehn, and an anonymous reviewer all contributed helpful comments. SOAR (Newell, in press) provided the original inspiration.

Correspondence concerning this article should be addressed to Emile Servan-Schreiber, Department of Psychology, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.

As Reber pointed out, two questions must be answered: (a) What is the form of the knowledge acquired during memorization that allows the subjects to efficiently discriminate grammatical from nongrammatical strings and (b) how was that knowledge acquired?

Miller (1958) proposed that subjects who memorize grammatical lists "group and recode" them (p. 49). Evidently this idea had its roots in an earlier article in which Miller introduced the idea of chunks (Miller, 1956). However, in both these articles Miller seemed to argue that the formation of chunks is an explicit and intentional recoding process. Reber reasoned that if this were the learning mechanism, subjects' knowledge of the grammar would have to be mostly verbalizable. Because his subjects were unable to verbalize their knowledge, he rejected Miller's hypothesis of recoding and proposed the existence of an unconscious covariation detection process that yields unverbalizable, abstract grammatical knowledge. Such a learning process he termed *implicit*, to contrast it with an explicit, intentional search for rules.

Our own position, which we propose and defend here, is that the learning mechanism is some sort of chunking and that the resulting knowledge on which grammatical judgments are based is a hierarchical network of chunks that, by virtue of having been created from grammatical strings, implicitly encodes grammatical constraints.

Our position differs from Miller's mainly because the concept of chunking has evolved since he introduced it 30 years ago. Whereas Miller (1956) proposed chunking mainly as a conscious recoding strategy, it is now usually understood to be a general learning mechanism not necessarily bound to the conscious and intentional realms of cognition (e.g., Newell, in press). In the absence of a good definition, we like to think of chunking as a natural, maybe automatic, tendency to process stimuli by parts.

There is plenty of evidence that chunking is the natural learning process when the task is to memorize complex non-sense verbal material (e.g., sequences of letters). For example, the 26 letters of the alphabet seem to be encoded into seven chunks: *abcd, efg, hijk, lmnop, qrst, uvw, and xyz* (Klahr, Chase, & Lovelace, 1983). Gestalt principles of proximity

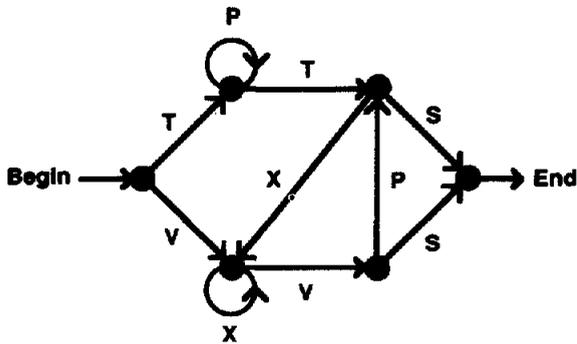


Figure 1. State diagram representation of an artificial finite state grammar. (This grammar was introduced in Reber, 1967, and was used to generate stimuli in our Experiment 1.)

induce subjects to form specific chunks (e.g., Bower & Winzenz, 1969; Johnson, 1970). But even if no organization exists a priori in the stimuli, subjects impose their own (e.g., Chase & Ericsson, 1981; Johnson, 1970; Tulving, 1962). A subject's chunking behavior is often revealed at recall by "transition-error probabilities" (e.g., Bower & Springston, 1970; Johnson, 1970) or "subjective organization" (Tulving, 1962).

Dulany, Carlston, and Dewey's (1984, 1985) analysis of the Reber task also points to the crucial role that chunking seems to play in the unintentional learning phenomenon. They proposed that the basic unit of subjects' grammatical knowledge is a small group of letters—what they called a "feature." In their analysis, subjects based their grammatical judgments on a dynamically evolving collection of explicit rules of the form: *The presence of this feature implies that the string is (or is not) grammatical.* Although we agree with Dulany et al. that features (we call them chunks) are the crucial units of grammatical knowledge, we disagree that such rules are needed to account for subjects' discrimination behavior.

Hence, there is reason to believe that subjects faced with the task of memorizing a meaningless and long enough string of letters will break it into chunks. For example, if a string to be memorized is TTXVPXVS, a subject may first create the chunks (TTX), (VP), and (XVS). Then further chunking may proceed until a single chunk encodes the whole string, at which point the string is assumed to be memorized. So, for example, the two following chunks are created in succession: ((TTX) (VP)) and (((TTX) (VP)) (XVS)), which encodes the whole string. In the process of memorizing the string, the subject has created five chunks. These chunks are organized in a hierarchy at the bottom of which are *elementary* chunks, which are the letters themselves. At the next level up are the *word* chunks, which are made of those elementary chunks. At the top of the hierarchy are the *sentence* chunks, which encode a full stimulus. In between the word and sentence levels are any number of hierarchically organized levels of *phrase* chunks. Figure 2 represents a portion of what such a hierarchical network of chunks would look like if the above string, among others, had been chunked as specified above.

In a situation where many strings have to be memorized and they are intrinsically similar to each other (because of an underlying grammar), the chunks may reveal those similar-

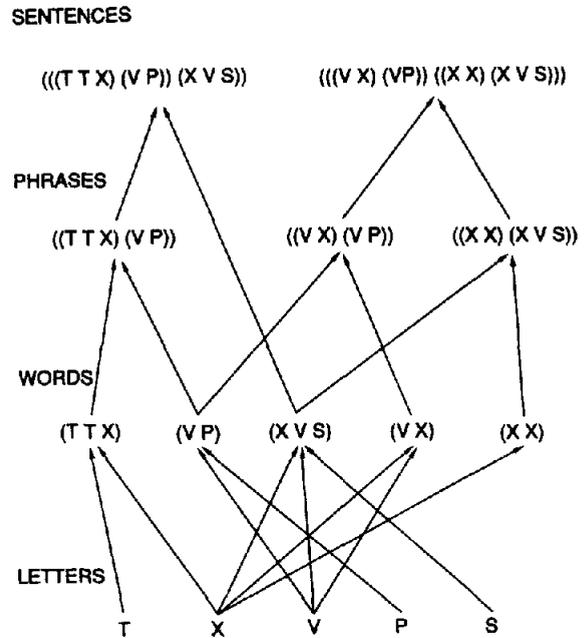


Figure 2. A hierarchical network of chunks that encodes two grammatical sentences from the grammar in Figure 1: TTXVPXVS and VXVPXXVS. (The representations of the two sentences overlap where they share chunks [i.e., (VP) and (XVS)]. The network is a multilevel structure. The bottom-most level is that of elementary chunks, in this case letters. The next level up is that of words. The next levels up are those of phrases, of which the highest level is that of full sentences.)

ties to some degree, thereby both facilitating and constraining further learning. For example, if the string TTXVPXVS had been chunked as specified above, and the string VXVPXXVS had to be memorized next, some of the chunks created while memorizing the former could be used to chunk the latter. Instead of perceiving the new string as just a sequence of nine letters (nine chunks), a subject may perceive it as a sequence of six chunks: V X (VP) X X (XVS). Not only does the new string immediately appear less complex, but this representation also constrains further chunking. The next chunks to be created would be (VX) and (XX)—which yields a four-chunk representation. The final representation of that string, once memorized, may then be (((VX) (VP)) ((XX) (XVS))).

The smaller the number of chunks that are needed to describe a string, the more familiar that string appears. The crucial variable is not the total number of chunks in the hierarchical representation of a string, but rather the number of chunks at the top level of that hierarchical representation. So, for instance, in the situation where the string TTXVPXVS is already memorized as above, its hierarchical representation has a single chunk at the top level: (((TTX) (VP)) (XVS)). Hence that string is maximally familiar.¹ On the other hand, the new string VXVPXXVS may be represented as V X

¹ Indeed, the original definition of a chunk in Miller (1956) is that of a "familiar" unit of knowledge (p. 93).

(VP) X X (XVS) with six chunks at the top level. It is less familiar than a memorized string but more familiar than it would have been if no word chunk had transferred to its representation (thus yielding nine chunks at the top level, one per letter). We call that crucial number of chunks *nchunks*. We tentatively define *familiarity* as

$$e^{1-nchunks} \quad (1)$$

Hence, the familiarity of a stimulus ranges from 1 (maximally familiar) to an asymptotic 0 (maximally unfamiliar).²

In summary, our hypothesis was that during the memorization task, subjects build a hierarchical network of chunks in which, due to the redundancy inherent in a grammar-generated set of strings, the representations of the strings overlap where they share chunks. The familiarity of a string increases with the degree to which a compact representation of that string can be built from the existing chunks.

We hypothesized further that the probability that a string is judged grammatical increases with how familiar it appears, given the network of chunks acquired during the memorization task. Hence, the more existing chunks are preserved in a string, the more chance it has of being judged grammatical.

In order to test this hypothesis, we had to control which chunks subjects created during the memorization task. Then we could test their grammatical knowledge by including in the test basically two sorts of nongrammatical strings: strings that violated the grammar but preserved subjects' chunks and strings that violated both the grammar and subjects' chunks. Our hypothesis directly predicted that the latter would appear less familiar and therefore would be rejected more often.

We tested this hypothesis with two analogues to Reber's (1967) experiment. The basic design of both our experiments was the same. The only significant departure from the original Reber task design was our addition of experimental conditions in which subjects were strongly induced to form specific chunks instead of being left to chunk the string by themselves. The first experiment tested our hypothesis at the level of word chunks, and the second tested it at the level of phrase chunks. In each experiment the nongrammatical strings included in the test either preserved the chunks (words in Experiment 1, words and phrases in Experiment 2) or did not preserve them (in Experiment 2, those strings preserved only the words).

Experiment 1

Method

Stimulus Materials

The artificial grammar we used to generate the stimulus set is represented in Figure 1. It is the one used by Reber (1967). There are exactly 34 grammatical strings of length six to eight in this language. We chose 20 of those to be memorized by subjects. For clarity, we will henceforth reserve the word *sentence* to refer to grammatical strings as opposed to nongrammatical strings. It is also important to note that whenever a string is referred to as grammatical or not, it is relative only to the grammar we used to generate the set of training strings (*Experimenter's grammar*, in Figure 1), not relative to any of the numerous other grammars which could also have generated that set.

Subjects and Conditions

The subjects were 37 Carnegie Mellon University undergraduates, participating as a requirement in an introductory psychology course and receiving \$5.

There were four conditions with 10 subjects in one and 9 in each of the other three. Three of the subject groups (well-structured-1, well-structured-2, and badly structured) saw sentences in which groups of letters were separated by spaces. Johnson (1970) demonstrated that the Gestalt principle of spatial proximity is very effective in biasing subjects toward forming specific chunks. The remaining group of subjects (unstructured) saw the same sentences but with no a priori organization into words, just sequences of letters. These subjects were in the same situation as Miller's (1958) and Reber's (1967) subjects.

The point of having three different conditions in which the sentences were structured was to investigate the effects that different structurings would have on subject performance in both the memorization and judgment tasks. In the two well-structured conditions, the chunks were designed to make the similarities among the sentences more apparent than in the unstructured condition. In contrast, in the badly structured condition, the chunks were designed to make these similarities more difficult to notice. In all four conditions, the 20 sentences presented for memorization were the same but for the change in format.

String format in the well-structured-1 condition. Reber (1967) noted that experimenter's grammar (see Figure 1) can be expressed³ as the union of five sentence types: (1) T (P)* TS, (2) T (P)* TX ((X)* | VPX)* VS, (3) T (P)* TX ((X)* | VPX)* VPS, (4) V ((X)* | VPX)* VS, and (5) V ((X)* | VPX)* VPS. This way of structuring the grammar yields a set of initial, middle, and final words on which the alternative representation of experimenter's grammar in Figure 3a is based: *Initial words* are T and V; *middle words* are TX, VPX, (P)*, and (X)*; *final words* are TS, VS, and VPS—a total of nine words.

String format in the well-structured-2 condition. Another way, among many, of expressing experimenter's grammar in terms of a set of words is given in Figure 3b (this grammar is equivalent to experimenter's grammar for all strings of more than four letters). It is based on a total of 11 words: *Initial words* are T(P)*T, TT, TT(X)*, VVP, V, and V(X)*; *middle words* are VP and (X)*; *final words* are S, (X)*VS, and VPS. This particular structuring of experimenter's grammar has the following two properties: (a) No word is of length larger than three (assuming that the length of a run is one), and (b) the average length of a sentence, in words, is close to three when the length in letters is limited to eight (as is the case in the training sentences). Our choice of the number three is based on its mention as "the largest size chunk everyone is willing to use" (Johnson, 1970, p. 211).

String format in the badly structured condition. In contrast with the two well-structured formats, the badly structured format was very unsystematic. It was designed to have as many words as possible, within the constraint that no sentence (in the memorization set) should have more than three words. By minimizing the transfer of words among the sentences, this structuring made the similarity of the sentences nonapparent. The sentences were based on 27 words

² This definition has not been experimentally derived. However, our intuition is that familiarity must be a rapidly decreasing function of *nchunks*.

³ In the notation used, "(...)*" indicates 0 or more apparitions of the contents of the parentheses, and "(...)+" indicates 1 or more apparitions of the contents of the parentheses. A "|" indicates a choice.

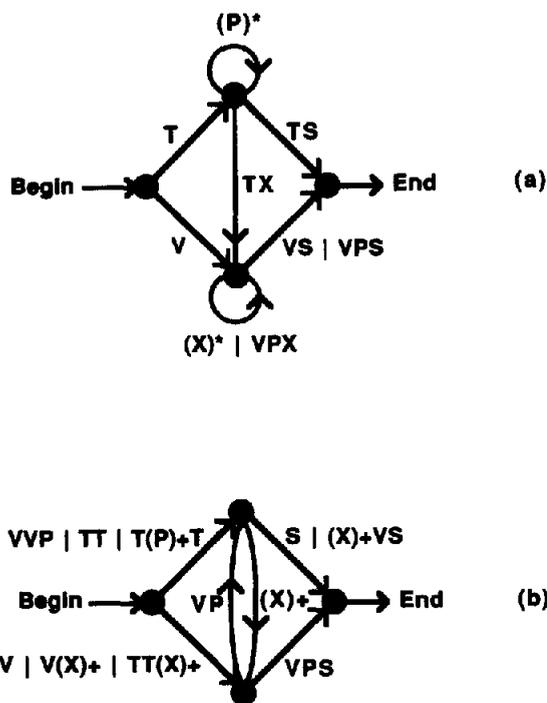


Figure 3. Two alternative, word-based, representations of experimenter's (E's) grammar (see Figure 1): (a) was used to structure the sentences of E's grammar in the well-structured-1 condition, and (b) was used to structure the sentences of E's grammar in the well-structured-2 condition.

(versus 9 and 11 in the well-structured-1 and well-structured-2 conditions, respectively): Initial words were $T(P)^+$, $V(X)^+$, VV , V , T , TTX , TT , VVP , and TPT ; middle words were $(X)^+VP$, $(X)^+V$, $P(X)^+$, $T(X)^+$, PTX , VP , XX , $VPXV$, $XVPX$, $PPPP$, and VPX ; final words were $(X)^+VS$, PTS , S , VPS , VS , PS , and TS .

As an example of how the different structurings made the strings look more or less similar to each other, Table 1 shows how the two strings $TPPTXXVS$ and $TPPPTXVS$ were differently formatted in each of the four conditions.

The memorization task (training). The 20 sentences chosen for the memorization task were distributed in five sets of four. Subjects were asked to memorize each set, numbered 1 through 5, in that order. However, once they had memorized a set, they were told that they did not have to try to remember its sentences any more, but just to concentrate on memorizing the sentences in the next set. The sentences in a set were presented individually for 5 s on a computer screen and in a fixed order. After viewing the four sentences once (one trial), subjects were asked to reproduce the set by writing on a blank card. They had to achieve two correct reproductions in a row before moving to the next set, without delay. The sets (included in full in the Appendix) were identical in the four conditions, except for

the changes in sentence format. The sets were designed so that four of the five different sentence types of the well-structured-1 format were represented in each set. Subjects were not informed of the true nature of the sentences, which were referred to simply as "strings," and the task was presented as a rote memory experiment.

The discrimination task (testing). Immediately after completing the memorization task, subjects were told that the 20 sentences they had memorized were all examples of "good strings"; they would now be asked to judge whether many other strings were "good" or "bad," on the basis of their experience with the training strings. They were not asked to give any reasons for their answers, but were given the following general clues about what could make a string "bad": (a) Something may be missing, (b) something may be extra, and (c) the order of letters in the string may be wrong. As the strings presented in this task were all in unstructured format, subjects in the three structured conditions were warned of the format change. All the subjects were tested on the same strings. There were 228 strings in this task: 82, or 36%, were grammatical (G), and 146, or 64%, were nongrammatical (NG). Subjects were not told about these proportions. Each string was presented individually on a computer screen for as long as it took a subject to make a judgment. No feedback on the correctness of the judgments (with respect to experimenter's grammar) was given.

Grammatical strings. There were two types of grammatical strings in the test. The *old-grammatical* strings were the 20 sentences that the subjects had just memorized. The *new-grammatical* strings were 21 newly generated sentences. Each was presented twice during the test. The length of the sentences varied from four letters to eight letters.

Nongrammatical strings. The 146 nongrammatical strings, 6 to 10 letters long, were all different (no repetitions). The words of the well-structured-1 format had a special role in the design of these strings. Specifically, the nongrammatical strings either preserved or did not preserve these words; a nongrammatical string could either be parsed by using only these words, or some part of it could not be parsed by using only these words. There were five types of nongrammatical strings, two of which preserved the well-structured-1 words (*all-words strings*) and three of which did not (*nonwords strings*).

All-words strings were nongrammatical because they violated the word-order constraints of the grammar. The violation occurred either at the end of a string or elsewhere. To generate such a string, we first generated a grammatical string, structured it in terms of well-structured-1 words, and then deleted a word, added a word, or replaced one word with another. If, as a result, the last word of the string was not one of the three valid final words of the well-structured-1 format, then the string was of the *all-words/bad-final* type. Otherwise, it was of the *all-words/good-final* type.

Nonwords strings were of one of three types. The *nonwords/random* type strings were either completely randomly generated from the five letters of the grammar or had a randomly generated middle part framed by valid initial and final words. In the two other types—*nonwords/bad-final* and *nonwords/good-final*—the strings were generated as follows: First, a grammatical string was generated and structured in terms of well-structured-1 words. Then, one of these words was made a nonword by simply replacing one of its letters with

Table 1
Examples of How the Same Strings Were Structured Differently in the Four Conditions

Condition			
Unstructured	Well-structured-1	Well-structured-2	Badly structured
TPPTXXVS	T PP TX X VS	TPPT XXVS	TPPTX XVS
TPPPTXVS	T PPP TX VS	TPPPT XVS	TPPP TX VS

another. If the nonword was the last word of the string, then the string was of type nonwords/bad-final. Otherwise, it was of type nonwords/good-final.

It seemed important to explicitly distinguish violations occurring at the end of a string from violations occurring elsewhere because other researchers have reported that, at least with a similar grammar, subjects are especially sensitive to violations occurring at the end of a string (e.g., Reber, 1967; Reber & Lewis, 1977). There were 88 strings of type all-words/good-final, 30 strings of type all-words/bad-final, 8 strings of type nonwords/good-final, 8 strings of type nonwords/bad-final, and 12 strings of type nonwords/random.⁴

Order of presentation of the test strings. The order of presentation of the test strings was randomized once with two constraints: (a) that each grammatical string appear once in the first half and a second time in the second half of the presentation sequence, and (b) that an equal number of strings of each nongrammatical type appear in each half of the presentation sequence. The order of test strings was then the same for every subject. Subjects were told that the length of a string is not an indication of its "goodness," although they had seen strings only of lengths six to eight in the memorization task. They were also warned about the presence of strings they had already seen (hence "good") and of the repetition of some strings, but not that only the "good" strings would appear twice.

Specific Predictions

The memorization task. If, as we hypothesized, subjects in the unstructured condition chunk the sentences themselves, then they should not be at a significant disadvantage compared with the subjects in the two well-structured conditions. Hence, their ease of memorizing the 20 exemplar sentences should be comparable to that of subjects in these well-structured conditions. On the other hand, subjects in the badly structured condition should have more difficulty because their sentences are structured so as to minimize the transfer of chunks among sentences.

The discrimination task. Our general prediction, that subjects would reject more the strings that did not preserve their chunks than the strings which did preserve them, was testable only within the well-structured-1 condition because the nongrammatical strings were especially designed either to preserve or not preserve word chunks acquired by these subjects. The specific prediction was that subjects in the well-structured-1 condition would reject the strings of type nonwords/bad-final more than the strings of type all-words/bad-final and would reject the strings of type nonwords/good-final more than the strings of type all-words/good-final.

Results

The Memorization Task (Training)

We had intended, before any subject was tested, to look for indirect evidence of chunking in the written protocols of subjects in the unstructured condition. However, this proved unnecessary; these subjects had a strong tendency to *overtly* chunk the training sentences, that is, to reproduce a sentence as separate groups of letters (e.g., reproducing TTXXXVS as TT XXX VS) or to write the end of a sentence before writing its beginning. More precisely, we computed how many of the 20 training sentences were overtly chunked at least once (on a correct reproduction) by each subject in this condition. We found that, on average, these subjects overtly chunked 14.5 of the 20 training sentences, that is, 72.5%. This does not

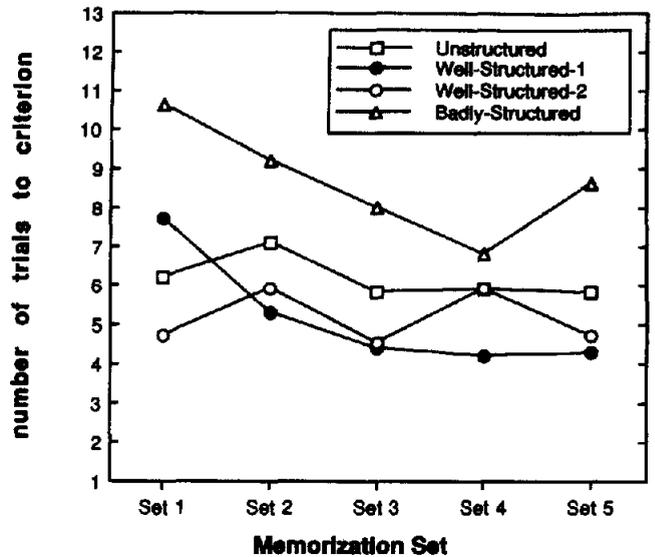


Figure 4. Mean number of presentation trials needed to reach criterion as a function of training set number (Experiment 1).

mean that these subjects did not chunk the remaining 27.5% of the training sentences but just that they did not do so overtly. Such direct evidence of chunking is, for our present purpose, the crucial result we extracted from the protocols.

There were systematic regularities in how many presentations of each set were necessary for subjects to reach criterion. These results are plotted in Figure 4. An analysis of variance (ANOVA) including the four conditions and the five successive sets revealed a main effect of condition, $F(3, 33) = 8.54$, $MS_e = 14.23$, $p < .001$, and a main effect of set number, $F(4, 132) = 5.52$, $MS_e = 3.81$, $p < .001$. Furthermore, sentence format interacted significantly with the ease of learning of individual sets, $F(12, 132) = 2.13$, $MS_e = 3.81$, $p < .025$. To check for the effect of presenting the strings in well-structured versus unstructured format, we did a two conditions by five sets ANOVA in which the conditions were unstructured and well-structured (i.e., the well-structured-1 and well-structured-2 subjects were grouped together). We found no significant main effect of condition, $F(1, 26) = 1.91$, $MS_e = 15.62$, $p = .18$, or of set number, $F(4, 104) = 2.02$, $MS_e = 4.02$, $p = 0.97$, and no interaction, $F(4, 104) = .46$, $MS_e = 4.02$, $p = .76$. Hence, we concluded that the main effect of condition found when the four conditions were included in the analysis was essentially due to the poorer performance of the subjects in the badly structured condition. As predicted, the manipula-

⁴ This striking imbalance in the numbers of strings per types results from the original exploratory nature of this experiment. When it was designed, we had no really precise hypothesis about subjects' behavior in the Reber task except that it should reflect some chunking activity. Twenty-two different types of nongrammatical strings, generated from 22 possible operations involving initial, middle, and final words, alone or in combination, were each represented by 6 to 10 strings in the test. Only later, when our theory became precise enough, did it also become clear that the fundamental tests of it rested in the comparisons of the all-words versus nonwords types of strings.

Table 2
Mean Percentages of Correct Classifications (PCCs) and Standard Deviations (SDs)

Measure of correct classifications	Condition							
	W-S-1		W-S-2		U		BS	
	PCC	SD	PCC	SD	PCC	SD	PCC	SD
% of grammatical strings accepted	78.7	5.8	83.4	10.8	75.7	19.3	74.6	16.3
% of nongrammatical strings rejected	63.5	9.5	69.9	16.4	65.2	16.6	50.3	7.4
Weighted average ^a	68.8	5.7	74.8	9.0	68.9	7.1	59.1	7.7

Note. The names of the conditions are abbreviated as follows: W-S-1 = well-structured-1; W-S-2 = well-structured-2; U = unstructured; BS = badly structured.

^a Includes weighted average of correct classifications over all strings (grammatical or nongrammatical).

tion of inducing subjects to form specific chunks in the two well-structured conditions, instead of letting them decide which chunks to form (unstructured condition), did not yield any significant advantage in the memorization task.

The Discrimination Task (Testing)

Table 2 contains the mean percentage of strings correctly classified by the subjects in each condition, and the same data broken down into correct acceptance of grammatical strings and correct rejection of nongrammatical strings. A four conditions by two types of strings ANOVA on these data revealed a main effect of condition, $F(3, 33) = 6.42$, $MS_e = 100.11$, $p < .005$, which was caused essentially by the poorer performance of the subjects in the badly structured condition, and a main effect of the grammatical status of strings, $F(1, 33) = 17.05$, $MS_e = 270.90$, $p < .001$, which reflected the fact that in all conditions the grammatical strings were more accepted than the nongrammatical strings were rejected. No interaction was revealed, $F(3, 33) = .59$, $MS_e = 270.90$, $p = .63$.

More fine-grained results are included in Table 3. It shows the mean percentage of strings rejected in each of the seven types of strings by the subjects in each condition. Within the well-structured-1 condition, both our predictions were veri-

fied. These subjects rejected the nonwords/good-final strings significantly more than the all-words/good-final strings, $t(8) = 3.6$, $p < .01$. They also rejected the nonwords/bad-final strings significantly more than the all-words/bad-final strings, $t(8) = 2.5$, $p < .05$. In none of the three other conditions did these same comparisons reveal significant differences.

Another interesting result is that the subjects in the well-structured-1 condition rejected the all-words/good-final strings more than the new-grammatical strings, $t(8) = 4.7$, $p < .005$. Because the new-grammatical strings were more likely than the all-words/good-final strings to preserve the phrase chunks of these subjects (in addition to their word chunks), that difference may indicate that the subjects were also sensitive to the preservation of their phrase chunks. An alternative explanation for the difference could have been that each new-grammatical string appeared twice, whereas the all-words/good-final strings appeared only once each. To rule out this explanation we compared the percentages of new-grammatical strings that these subjects rejected on their first presentation, 29.6%, and on their second presentation, 28.6%. Apparently the effect of repetition on judgment of new-grammatical strings was negligible.

Discussion

It could have been argued that our main experimental manipulation of presenting the training sentences already structured, instead of unstructured as in the Miller (1958) and Reber (1967) experiments, significantly altered the nature of the memorization task. If that were true, then we could not extend our analysis of the subjects in the well-structured conditions to the subjects in the unstructured condition. However, our results not only confirmed that the subjects in the unstructured condition chunked the sentences themselves—that is, imposed their own structure—but also that their discrimination performance matched that of the subjects in the well-structured conditions. Hence, we are justified in extending our analysis of the subjects in the well-structured-1 condition to the subjects in the well-structured-2 and unstructured conditions. We must, however, consider the possibility that the subjects in the badly structured condition may have used more complex memorization strategies than simple chunking, for the poor structure of their training sentences made this task more difficult for them than for the other subjects. Whenever an effect of condition was found, in either

Table 3
Mean Percentages (MPs) and Standard Deviations (SDs) of
Strings Rejected in Each of the Seven String Types

String type	Condition							
	W-S-1		W-S-2		U		BS	
	MP	SD	MP	SD	MP	SD	MP	SD
Old-grammatical	13.9	6.0	15.8	12.4	18.3	15.2	19.7	17.7
New-grammatical	29.1	7.7	17.1	12.3	30.4	24.4	30.9	18.4
All-words/good-final	50.1	10.8	63.9	19.4	56.9	19.0	42.2	7.8
Nonwords/good-final	73.6	22.9	63.8	32.0	54.2	25.0	40.3	13.7
All-words/bad-final	78.5	21.3	76.3	22.8	78.9	21.1	65.2	15.3
Nonwords/bad-final	97.2	5.5	82.5	18.8	84.7	16.3	59.7	21.4
Nonwords/random	95.4	8.4	92.5	8.3	87.0	11.1	73.1	19.5

Note. The names of the conditions are abbreviated as follows: W-S-1 = well-structured-1; W-S-2 = well-structured-2; U = unstructured; BS = badly structured.

the memorization or discrimination task, it was always due to the poorer performance of these subjects.

Morgan, Meier, and Newport (1987), in a strikingly similar experiment, found that subjects trained with well-structured stimuli performed much better than subjects trained with unstructured stimuli. Our results do not reproduce this effect. This difference may be entirely due to the fact that their subjects were instructed to look for the rules of the language, whereas ours were kept ignorant even of the existence of an underlying language. Reber, Kassin, Lewis, and Cantor (1980) have shown that explicit instructions to look for structure significantly alter the outcome of the learning process. In our view, it must alter the learning process itself. Hence, our study and Morgan et al.'s may not be very relevant to each other.

Our hypothesis was that subjects chunk the exemplar sentences and then base their judgments of grammaticality on the degree to which compact representations of strings can be built with their chunks. We found accordingly that subjects were more prone to reject strings which did not preserve their word chunks than strings which did: The subjects in the well-structured-1 condition rejected the nonwords/good-final strings more than the all-words/good-final strings and rejected the nonwords/bad-final strings more than the all-words/bad-final strings. Furthermore, there was some evidence that the degree to which the phrase chunks were preserved also affected judgments of grammaticality: The subjects in the well-structured-1 condition rejected the all-words/good-final strings more than the new-grammatical strings, which are more likely to preserve phrase chunks.

We designed our second experiment to investigate further the effect of preserving phrase chunks. The basic design was the same as that of our first experiment. Exemplar sentences were memorized in different formats, and a discrimination task followed. The difference was that instead of inducing different groups of subjects to form different word chunks, we induced them to form the same word chunks but different phrase chunks. Whereas all the test strings preserved the word chunks of all subjects, they either preserved or did not preserve subjects' phrase chunks. We predicted that subjects would tend to reject the strings that preserved only their word chunks more than the strings that preserved both their word chunks and their phrase chunks.

Experiment 2

Method

Stimulus Materials

For this experiment we created a grammar of a different kind, one more immediately defined in terms of word sequences than in terms of letter sequences. The grammar we designed is depicted in Figure 5. Each of the nine symbols in the figure could be instantiated by two different words. The 18 possible words were meaningless pairs of consonants (e.g., FK). There were three sorts of words, as there are three sorts of symbols in the figure: initial words (I1, I2, I3), middle words (M1, M2, M3), and final words (F1, F2, F3). Every one of the 64 possible sentences was three words long. Each was made of an initial word, followed by a middle word, followed by a final word.

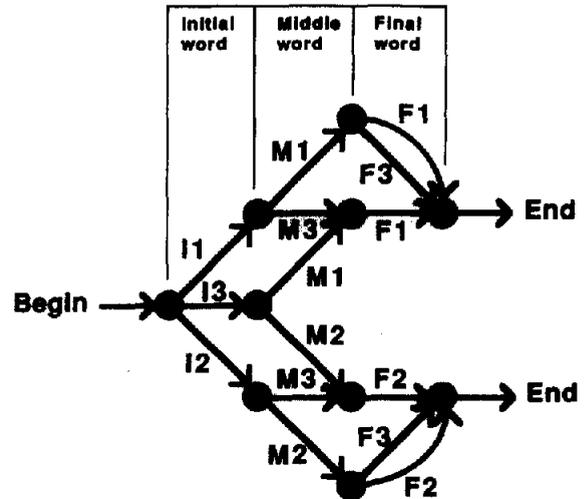


Figure 5. The grammar used to generate sentences in Experiment 2. (Each of the nine symbols could be instantiated by two two-letter words [for instance, I1 could be either FK or MV]. This grammar could generate 64 different three-word sentences. Each sentence was made of an initial word, a middle word, and a final word.)

The possible sentences could be classified into four types. Sentences of Type 1 instantiated [I1 M1 F1], sentences of Type 2 instantiated [I2 M2 F2], sentences of Type 1/3 instantiated either [I1 M1 F3] or [I1 M3 F1] or [I3 M1 F1], and sentences of Type 2/3 instantiated either [I2 M2 F3] or [I2 M3 F2] or [I3 M2 F2]. Sentence Types 1/3 and 2/3 could be viewed as one-word distortions of the prototype sentence Types 1 and 2, respectively.

Subjects and Conditions

There were two experimental conditions—*initial-middle* and *middle-final*—and a control condition. As in Experiment 1, they differed in the way sentences were presented during the memorization task. Subjects were 38 Carnegie Mellon University undergraduates participating for credit in an introductory psychology course and receiving \$3. There were 14 subjects in the initial-middle, 12 in the middle-final, and 12 in the control condition.

The memorization task (training). Sentence presentation was manipulated in the two experimental conditions to induce specific phrase-chunkings of sentences. In Condition Initial-Middle, we wanted subjects to chunk sentences as ((initial middle) final) structures, whereas in Condition Middle-Final, we wanted subjects to chunk sentences as (initial (middle final)) structures. As in the first experiment, we used spacing between words to induce the first-order (word) chunking. The new problem was to find some mode of presentation that would get subjects to chunk the desired phrases. Our solution was to first present only the desired phrase, and then the whole sentence. For instance, to get initial-middle subjects to chunk the initial and middle words of FK TM KS together, we first presented FK TM //.⁵ Only after they had memorized that did we present the whole sentence. Hence, they had no choice as to which phrases to chunk. In that case, the sentence would end up being memorized as ((FK TM) KS). In contrast, middle-final subjects first had to memorize \ TM KS before they could memorize the whole

⁵ Subjects were told that “//” was simply a place holder for a future pair of letters.

sentence. Hence, their representation of the sentence would be (FK (TM KS)). Finally, control subjects were left to decide for themselves how to chunk the sentences (like the unstructured condition subjects in Experiment 1). They always saw complete sentences on the memorization trials.

The 48 sentences of Types 1/3 and 2/3 were presented in the memorization task (the 16 sentences of Type 1 and 2 were reserved for the discrimination task). They were distributed in 16 sets of 3 sentences each. The distribution of sentences among sets was different in each condition, but there were always 9 different words per set, so that every subject saw all 18 words every two sets. As in Experiment 1, the sentences of a set were presented individually for 5 s on a computer screen. After having seen the 3 sentences of a set, subjects were asked to type them back on the computer's keyboard. The protocols were recorded directly by the experiment's program. Whereas control subjects saw complete sentences on every trial, experimental subjects saw only two words per sentence until they could correctly reproduce that. Only then did the full-sentence trials begin for a specific set. One correct reproduction of three full sentences was required before moving to the next set. As in Experiment 1, subjects were not informed of the true (grammatical) nature of the sentences, which were referred to simply as "strings," and the task was presented as a rote memory experiment.

The discrimination task (testing). Immediately after having mastered the 16th training set, subjects were told that they had just seen 48 "good" strings and that they would now have to try to discriminate between new "good" strings and "bad" strings. They were told that the good strings would be those in which the "the three words seemed to fit well together." This was the first time that "words" were mentioned, but no reference was made to a grammar or a language.

Grammatical sentences. The grammatical sentences in the test were the 16 sentences of Types 1 and 2. Note that although the specific three-word combinations which those sentences represent had not been seen during the training, every pairwise combination of words in such sentences had been seen twice during training (in two different sentences). Thus, the three words in a Type 1 or 2 sentence would presumably seem to "fit well together." Because none of these strings had been seen during training, they were collectively referred to as *new-grammatical*. Each was repeated once.

Nongrammatical strings. These strings were designed to contain nongrammatical word pairs. There were three types of such strings: *replace-initial*, *replace-middle*, and *replace-final*. To build one of these strings, we took a Type 1 or Type 2 sentence (*new-grammatical*) and simply replaced one of the words with another of the same sort but which had never occurred with the other two in the memorized sentences. So, for instance, to build a *replace-initial* string we took a Type 1 string and replaced the initial I1 word with an initial I2 word. The resulting string instantiated [I2 M1 F1], where the initial word had never occurred with any of the other two words in the memorized sentences. *Replace-middle* and *replace-final* strings were built similarly by replacing middle or final words. All the 48 possible *replace-initial*, *replace-middle*, and *replace-final* strings were included in the test (16 in each type). Thus, the subjects were to judge 80 strings (16 × 2 grammatical and 48 nongrammatical).

Specific Predictions

Given this specific experimental design and given our general prediction that subjects will tend to reject the strings that do not preserve their chunks more than the strings that do, we predicted that initial-middle subjects would tend to reject *replace-initial* and *replace-middle* strings more than *replace-final* strings (which preserved their initial-middle phrase chunks). In contrast, we predicted that middle-final subjects would tend to reject *replace-middle* and *replace-final*

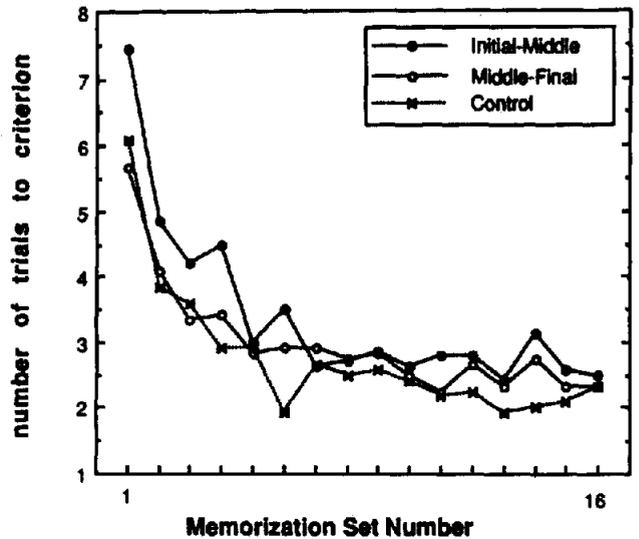


Figure 6. Mean number of presentation trials needed to reach criterion as a function of training set number (Experiment 2).

strings more than *replace-initial* strings (which preserved their middle-final phrase chunks).

Results

The Memorization Task (Training)

The learning curves of the three subject groups are plotted in Figure 6. The only significant effect was that of trial number, $F(15, 525) = 29.75$, $MS_e = 1.44$, $p < .0001$. Apparently, forcing the experimental subjects to process the sentences by parts made them neither significantly slower nor faster than the control subjects to memorize the sentences.

The Discrimination Task (Testing)

Table 4 contains the mean percentages of strings rejected in each of the four types of strings by the subjects in each condition. A three conditions by four types of strings ANOVA revealed a main effect of string type, $F(3, 105) = 12.87$, $MS_e = 4.82$, $p < .0001$, because of the lower rejection level, across conditions, of the *new-grammatical* strings compared with

Table 4
Mean Percentages (MPs) and Standard Deviations (SDs) of Strings Rejected in Each of the Four String Types

String type	Condition					
	Initial-middle		Middle-final		Control	
	MP	SD	MP	SD	MP	SD
New-grammatical	37.7	12.6	36.5	11.9	37.8	10.0
Replace-initial	67.9	19.0	38.6	18.4	48.5	18.3
Replace-middle	65.6	22.7	52.6	19.1	50.0	16.0
Replace-final	50.9	21.6	53.2	19.7	39.1	11.3

nongrammatical strings. There was also a main effect of condition, $F(2, 35) = 3.62$, $MS_e = 15.97$, $p < .05$, due to the initial-middle subjects' overall higher rejection of nongrammatical strings. Finally, these factors interacted significantly, $F(6, 105) = 4.13$, $MS_e = 4.82$, $p < .001$, suggesting that subjects in different conditions reacted differently to different types of nongrammatical strings, as expected.

Our two predictions were verified in the data. First, the initial-middle subjects rejected the replace-initial and replace-middle strings significantly more than the replace-final strings, $t(13) = 2.9$, $p < .025$, and $t(13) = 2.5$, $p < .05$, respectively. Second, the middle-final subjects rejected the replace-middle and replace-final strings significantly more than the replace-initial strings, $t(11) = 2.7$, $p < .025$, and $t(11) = 3.1$, $p < .025$, respectively.

It appeared also that the control subjects behaved more like the initial-middle subjects than like the middle-final subjects. They rejected the replace-middle strings significantly more than the replace-final strings, $t(11) = 2.7$, $p < .025$, and they had a tendency to reject the replace-initial strings more than the replace-final strings, $t(11) = 2.0$, $p = .068$.

Discussion

The results of Experiment 1 showed that subjects were more willing to classify as "good" the strings that preserved their word chunks than the strings that did not. The results of Experiment 2 demonstrated further that among the strings that preserved subjects' word chunks, those that also preserved subjects' phrase chunks were preferred.

Interestingly, the behavior of the subjects who were left to impose their own phrase structure on the sentences was very similar to the behavior of the subjects who were induced to form initial-word-middle-word phrases. This may be due to a natural bias, given that usual reading is left to right, toward forming such phrases.

If such a bias was real, then it may also explain why the initial-middle subjects were better discriminators than the middle-final or control subjects; the training in the initial-middle condition may have reinforced the natural tendency to form initial-word-middle-word phrases, whereas the training in the middle-final condition may have fought against it. Hence, the initial-middle subjects may have been made additionally sensitive to nonpreservation of their phrase chunks.

General Discussion

Together, the results of our two experiments provided strong support for our hypothesis that the main learning process in the Reber task was some sort of chunking and that grammatical discrimination was based on the degree to which compact representations of strings could be built from the collection of learned chunks. This encouraged us to formulate a precise theory of the processes involved, both in learning and discrimination. We call it *competitive chunking* (CC). In the rest of this article we first describe the theory and then report two simulations of the experimental results of Miller (1958) and our own results from Experiment 1.

The Theory of Competitive Chunking

Chunks are traces in long-term memory, hierarchical structures whose elements are themselves chunks. Indeed, everything in long-term memory must be a familiar unit of knowledge, hence a chunk. In CC, the processor knows two things about every chunk: (a) what its immediate subchunks are and (b) a composite score reflecting how often and recently it has been used in the past. This composite score is called a chunk's *strength*. It determines, as we will see later, how likely it is that the system will use that chunk when an opportunity to do so arises. The strength construct in CC is identical to that in ACT* (Anderson, 1983) for declarative memory traces: A newly created chunk has a strength of one unit. Strength is increased by one unit every time the chunk is used or recreated. However, strength decays with time. At any point in time, the strength of a chunk is the sum of its successive individually decaying strengthenings:

$$\text{Strength} = \sum_i T_i - d, \quad (2)$$

where T_i is the time elapsed since the i th strengthening, and d is the *decay parameter* ($0 < d < 1$). Once a chunk is created, it exists forever in long-term memory, and there is no bound on how much strength it can accumulate. Strength decay implements one type of forgetting, as weaker chunks are less likely to be used by the system. Anderson (1983) showed how this strength construct explains, among other things, the power law of forgetting.

Chunks are used to perceive environmental stimuli. Indeed, the system can perceive only that part of its environment that exhibits its chunks. The perception process is assumed to be bottom-up⁶ from the simplest, *elementary*, chunks (i.e., those that the system never had to learn). For example, let us make the approximation that letters are elementary chunks. Let us also consider a system that has the set of chunks below:⁷

A, B, . . . , Z
(CARNEGIE), (MELLON), (UNIVERSITY),
((CARNEGIE (MELLON)),
(((CARNEGIE (MELLON)) (UNIVERSITY))).

If we confront the system with the following stimulus:

carnegiemellonuniversity,

it would first perceive every letter, yielding the elementary percept

CARNEGIE MELLON UNIVERSITY;

⁶ We chose to ignore top-down processing only because we did not feel that it was critical to our theorizing in the Reber task, and we wanted to make a minimum of assumptions. However, we recognize that the case for top-down processing in perception is strong. It is not hard to consider how such processing may occur in CC. We may formalize that capability in more sophisticated versions of the theory.

⁷ It is convenient in this example and those that follow to consider uppercase letters as letter chunks and to reserve lowercase letters for stimuli.

then it would be in a position to use its more complex chunks, yielding the percept

(CARNEGIE)(MELLON)(UNIVERSITY);

then it could elaborate on this percept with still more complex chunks, yielding

((CARNEGIE)(MELLON))(UNIVERSITY)

and finally

((((CARNEGIE)(MELLON))(UNIVERSITY))).

The perception process is a recursive cycle. The elementary percept is formed by matching elementary chunks to the stimulus. Then the next percept is formed by elaborating on the elementary percept with more complex chunks. Then the next percepts are formed by elaborating on the current percept with increasingly complex chunks until no more chunks are available to elaborate on the current percept.

The number of chunks at the top level of a given percept gets smaller with every cycle of the perception process. In the example it is 24 in the elementary percept, 3 in the following percept, then 2, and 1 in the final percept. This number, in the final percept, is what we called earlier *nchunks*. It is the crucial variable in CC. The value of *nchunks* is a measure of how compact the representation of a stimulus is, and that translates into how familiar it is perceived to be (recall Equation 1).

The example doesn't illustrate that point, but just because a chunk matches the current percept doesn't imply that it will be available to elaborate on it. The strength construct mediates how readily available each chunk is to the perception process. Let us call the average strength of a chunk's subchunks its *support*. Then the probability that a matching chunk is retrieved and becomes available to elaborate on the current percept is a function of its support:

$$\frac{1 - e^{-c \cdot \text{support}}}{1 + e^{-c \cdot \text{support}}}, \quad (3)$$

where *c*, for reasons discussed later, is called the *competition parameter* ($c > 0$) and determines the steepness of the probability curve.⁸ The larger *c* is, the easier it is to retrieve chunks at all levels of support. Figure 7 plots this function for different values of *c*.

An exception: Because elementary chunks do not have any subchunks, both their supports and strengths are assumed to be constant and very large (in our simulations we set both at 10). Hence, independently of the value of *c*, they are always readily available, and able to efficiently support their immediate superchunks.

Note that a chunk's strength does not affect its own probability of being retrieved but directly affects its superchunks' probabilities of being retrieved. In that way, when a chunk's strength decays, its superchunks are being forgotten. Conversely, when a chunk's strength increases, its superchunks are being learned.

In our example above, if (CARNEGIE) and (MELLON) did not have enough strength, then their superchunk ((CARNEGIE)(MELLON)) may not have been

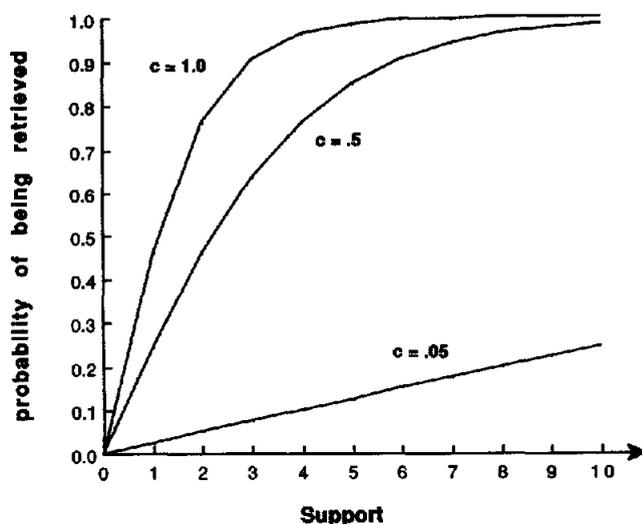


Figure 7. A matching chunk's probability of being retrieved (for perception) as a function of its support (defined as the average strength of its immediate subchunks). (This function is also used to determine a potential new chunk's probability of being proposed [for creation] as a function of its support.)

retrieved. (In that case, the final percept would have yielded *nchunks* = 3.)

Consider a system which, in addition to the letter chunks, has the five complex chunks: (CARNEGIE), (MELLON), (BANK), ((CARNEGIE)(MELLON)), and ((MELLON)(BANK)). Suppose that it is confronted with the following stimulus:

carnegiemellonbank.

It will first build the elementary percept

CARNEGIE MELLON BANK,

then the more complex percept

(CARNEGIE)(MELLON)(BANK).

At this point, both of its most complex chunks match this percept. Each chunk then is or is not retrieved, according to Equation 3. There are therefore four possible situations. In the first situation, neither chunk has enough support to be retrieved. In the second and third situations one of the chunks is retrieved, but not the other. In the fourth situation, both chunks are retrieved. If no chunk is retrieved, then the final percept is the one above. If only one chunk is retrieved, then it is used to elaborate on the percept above to yield the final percept. If both chunks are retrieved, then there is a conflict because the situation is such that only one of them can elaborate on the percept. In this situation, the two chunks are

⁸ The reader may recognize that this function is a sigmoid that is inflected at 0 and that has values bound between an asymptotic -1 and an asymptotic +1. However, we are considering only its positive part because support is always positive.

said to be competitors.⁹ How should the winner be determined? Because a chunk's strength is a composite measure of how often and recently it has been used in the past, it is natural that the winner should be the stronger chunk. This chunk is strengthened (by one unit), its losing competitors are not.

Note that even though the probability that a chunk is retrieved depends exclusively on the strength of its subchunks (i.e., its support), it is its own strength, relative to that of its competitors, that determines if it is used by the perception process (and strengthened consequently). A chunk can be effectively forgotten because of interference from a stronger competitor. Therefore, both a chunk's support and strength are critical to its being used by the perception process. Put another way, a chunk's strength is a critical parameter for both itself and its superchunks. When a chunk is strengthened, both it and its superchunks are being learned. Conversely, when a chunk's strength is left to decay, both it and its superchunks are being forgotten.

We have examined one type of learning in CC: the strengthening of existing chunks. Now we examine how CC creates new chunks. We have defined elementary chunks as those that the system never had to create. All other chunks must be created. The creation process is a direct extension of the perception process. Its input is the final percept. Its output is a single new chunk whose immediate subchunks are chunks in the final percept. For example, if the final percept is made of four chunks,

Chunk1 Chunk2 Chunk3 Chunk4,

then at least three potential new chunks may be proposed: (Chunk1 Chunk2), (Chunk2 Chunk3), and (Chunk3 Chunk4). The probability that a potential new chunk is actually proposed is computed by Equation 3. Therefore, the process of proposing potential new chunks for creation is assumed to be equivalent to the process of retrieving existing chunks for perceiving. Just as there are competitions in the process of perceiving, there are competitions in the process of creating new chunks, for at most one chunk may be created at one time. The winning chunk in chunk-creation competitions is that with the largest support. It is given a strength of one unit. In case it already exists, it is simply strengthened. CC does not create multiple copies of a chunk.

Let us note a few implications of CC's processes. (a) The elementary chunks never compete with each other because they do not overlap each other. (b) A chunk whose immediate subchunks are elementary is always readily available to the perception system because its support is large, for it is equal to the strength of an elementary chunk. (c) Just as there are situations in which no matching chunk has enough support to be retrieved for perception, there are situations in which no potential new chunk has enough support to be proposed for creation. CC avoids these situations as much as it can in two ways: first, by letting the strongest chunks, able to provide the most support, win perception competitions, and second, by letting the most supported chunks win chunk-creation competitions. (d) However, this does not guarantee that the perception process always builds as compact a percept as it could. The process is only doing hill climbing. (e) To accu-

multate strength, chunks must be retrieved and win perception competitions. As the winner of a creation competition is the chunk with the most support given a particular percept, it is also the chunk most likely to be retrieved when the same or a similar percept is built at a later time. Hence, it is the chunk with the most potential for strength accumulation. Not only does the creation process extend the perception process, it also feeds back the chunks that are the most likely to be useful to the perception process, which in turn makes them likely to be able to support further chunk creation, and so forth.

Below, we discuss two simulations of experimental data with the dual goal of illustrating CC's important properties and providing evidence for it. Miller-CC, a simulation of the data of Miller (1958), illustrates the learning process, the building of a complex network of chunks. Reber-CC, a simulation of the data in our Experiment 1, illustrates how such a network, when created from grammatical stimuli, can then be used to perform the grammatical discrimination task.

In each simulation we made additional assumptions in order to extract from the outcome of the perception process (i.e., the value of *nchunks*) data in the same form as those collected from human subjects. In Miller-CC we had to transform the value of *nchunks* into an act of recall. In Reber-CC we had to transform the value of *nchunks* into an act of rejection.

A common simplifying assumption to both simulations was that made about time. We increased time by one unit after each perception episode (i.e., the presentation of one string).

When the stimuli were unstructured strings of letters, we assumed that the letters were elementary chunks, and we put the following constraints on the generation of potential new chunks. (a) Consistent with the Gestalt principle of proximity, the subchunks of a chunk must be adjacent. (b) The *word* chunks, whose subchunks are elementary (letter) chunks, can have at most three subchunks, except if they encode runs. This constraint reflects the already mentioned finding that the preferred word-chunk size of subjects is three. For runs, though, additional letters do not add to the difficulty of encoding, so we relaxed that constraint. (c) The *phrase* chunks, whose subchunks are either words or phrases, can have at most two subchunks. This was to enforce an intuition that length constraints must become more severe as the complexity of chunks increases. (Two is the minimum number of subchunks that still permits chunking.)

When the stimuli were structured strings (as in the three structured conditions of Experiment 1), constraint (b) was relaxed to accept the words given in the a priori segmentation of a string, whatever their length. Constraints (a) and (c) were unchanged.

Miller-CC: A Simulation of Miller's (1958) Experimental Results

Miller (1958) examines the effect of a stimulus field's redundancy, or lack thereof, on recall. Subjects were asked to

⁹ When a chunk is retrieved, it can also be considered to be entering a competition (even if there are no other competitors). Hence the name *competition parameter* for *c*.

memorize and free-recall a list of strings of consonants. There were two kinds of lists. Language lists (L) were made of strings generated by the same simple finite state grammar. Random lists (R) were made of randomly generated strings (using the same letters as the grammar). A list was presented one string at a time (a few seconds each), and afterwards a subject was asked to free-recall all the strings he or she could. The dependent variable was the number of strings correctly recalled after each of 10 successive presentations of a list.

There were two L-lists, L1 and L2, among which 18 strings generated from the grammar were evenly distributed (9 strings per list). There were also two R-lists, R1 and R2, with 9 randomly generated strings in each. Subjects went through 10 trials on one of these lists.¹⁰ Subjects in Condition L studied L1 or L2, while subjects in Condition R studied R1 or R2. Table 5 contains the four lists used by Miller, and Figure 8a plots the recall performance of Miller's subjects. Evidently, a redundant stimulus field (L1 or L2) facilitated recall compared with a nonredundant stimulus field (R1 or R2).

In the simulation it was necessary to transform the outcome of the perception process—the value of *nchunks*—into an act of recall. The assumption we made was that a string would be recalled if and only if it appeared maximally familiar, that is, *nchunks* = 1. The lists we used for the simulation are those in Table 5, and 20 simulated subjects were tested in each condition (10 simulated subjects per list). It was necessary to have more than one simulated subject in each condition because, due to the probabilistic nature of the retrieval and proposal processes, no two simulated subjects learned identical sets of chunks or behaved identically on each trial.

Many different values were tried for the two parameters *c* and *d*. Miller-CC always had an easier time learning the redundant L-lists than the nonredundant R-lists. Even when the decay parameter was set extremely close to 0 (no decay), and the competition parameter was set to a large value so that every matching chunk was sure to be retrieved, the simple fact that fewer words were needed to encode the L-strings than to encode the R-strings, because of the redundancy in the L-lists, let the simulation learn the L-lists faster than the R-lists. However, a more interesting aspect of the data was more difficult to reproduce: the increase in the apparent

Table 5
Lists of Strings Used in Miller (1958) and the Miller-CC Simulation

Language strings		Random strings	
L1	L2	R1	R2
SSXG	NNSG	GNSX	NXGS
NNXSG	NNSXG	NSGXN	GNXSG
SXSXG	SXXSG	XGSSN	SXNGG
SSXNSG	NNXSXG	SXNNGN	GGSNXG
SXXXSG	NNXXSG	XGSXXS	NSGNGX
NNSXNSG	NNXXSXG	GSXXGNS	NGSXXNS
SXSXNSG	NNXXSXG	NSXXGSG	NGXXGGN
SXXXSXG	SSXNSXG	SGXGGNN	SXGXGNS
SXXXXSG	SSXNXSG	XXGNSGG	XGSNGXG

Note. From "Free Recall of Redundant Strings of Letters" by G. A. Miller, 1958, *Journal of Experimental Psychology*, 56, p. 487. Copyright 1958 by the American Psychological Association. Reprinted by permission.

advantage of the L-lists on successive trials. We found that this aspect of the data was reproduced qualitatively well when $c = d = .5$ (see Figure 8b).

The redundancy in the L-lists is at the word level. Small groups of letters in the L-strings are shared by many L-strings. For instance, the 18 redundant strings in the L-lists of Table 5 all end with one of three triplets: XSG, SXG, or NSG. In contrast, only one final triplet is shared by 2 of the 18 nonredundant strings in the R-lists. Because they were in general more redundant, the words chunked in the L-strings accumulated more strength more rapidly than the words chunked in the R-strings. This, in turn, made the phrases chunked in the L-strings easier to create and more reliably useful than the phrases chunked in the R-strings. Hence, the former were able to start to accumulate strength sooner and to accumulate it more reliably than the latter. In the same manner, stronger phrases chunked in the L-strings made it possible to learn entire L-strings sooner and to use them more reliably. The strength and support constructs, and the way that they mediate perception and creation processes, made it possible to transfer the word-level advantage of redundancy in the L-lists to the phrase level and finally to the sentence level, even though the phrases and entire strings in the L-lists are no more redundant than the phrases and entire strings in the R-lists.

This simulation illustrates how the strengthening and decay processes in CC interact with the natural structure of a stimulus field to facilitate learning or to make it harder. A critical level of redundancy in the stimulus field is needed for the chunks to accumulate strength in spite of continuous decay. Below this level, nonelementary chunks would never be able to accumulate enough strength to support additional learning. (The R-lists were not redundant but enough redundancy was provided, across trials, by their repetition to allow some learning). Above this level, additional redundancy speeds

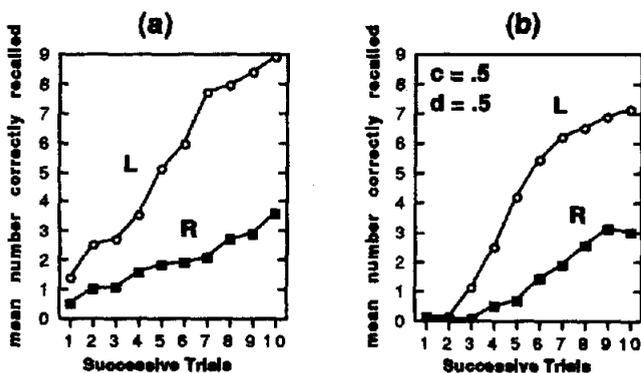


Figure 8. Mean number of strings correctly recalled by (a) human and (b) simulated subjects on successive trials. (Human data reproduced with permission from Miller, 1958.)

¹⁰ The actual experiment had subjects study a second list after the first. Here we are mostly interested in the data obtained with the first list, so no more mention will be made of the second list.

learning. The critical level depends, of course, on the specific value of the decay parameter.

Reber-CC: A Simulation of the Results of Our Experiment 1

Despite superficial differences, the Miller task with language lists and the training phase of the classic Reber task, as instantiated in our Experiment 1, are the same tasks. Hence, learning in Reber-CC occurs exactly as it does in Miller-CC. Here also we had to use multiple simulated subjects in each condition (10 each) because each created its own personalized set of chunks during the memorization task. This echoes the observation that subjects in the Reber task seem to learn personalized grammars (Dulany et al., 1984). Whereas our human subjects had to attain a criterion of two consecutive reproductions of a set before moving to the next, the simulated subjects had to "recall" (i.e., $nchunks = 1$) the four strings of a set on the same trial once. During the discrimination task the creation of new chunks was turned off, but the strengthening and decay processes continued to operate.

To get the simulated subjects to produce discrimination data in the same form as the human subjects, we had to transform the value of $nchunks$ into a rejection act. The assumption we made was that the probability that a string would be rejected increased with the value of $nchunks$. The higher the value of $nchunks$ was, the less familiar a string was perceived to be and the higher was the probability that it would be rejected. The probability function was the simple sigmoid:

$$\frac{1}{1 + e^{n - nchunks}} \quad (4)$$

where the inflection point n was computed, on each individual discrimination trial, as the average value of $nchunks$ on a few of the previous trials (we chose, arbitrarily, to consider the previous 20 trials). Hence, the same value of $nchunks$ yielded different probabilities of rejecting a test string, depending on the average value of $nchunks$ for the previous few test strings. This means that the simulated subjects based their judgments not on how absolutely familiar a string appeared to them but on how familiar it appeared to them relative to the average familiarity of the previous few strings.

It makes sense that the judgments be made in context rather than independently of each other. Given the nature of the task, a subject was led to believe that he or she should neither accept all strings nor reject all strings. If the test strings were such that they all appeared equally familiar (i.e., same value of $nchunks$: n), then we would expect that the subject would arbitrarily reject roughly half of them. The context-dependent rejection probability function that we propose would produce such behavior, whereas a context independent function could not.

We want to stress that n was dynamically adjusted by the simulated subjects themselves in response to their success in creating more or less compact representations of the test strings. It is not, like c and d , a parameter that we could control and vary to fit the data. Figure 9 plots the rejection-probability function for different values of n .

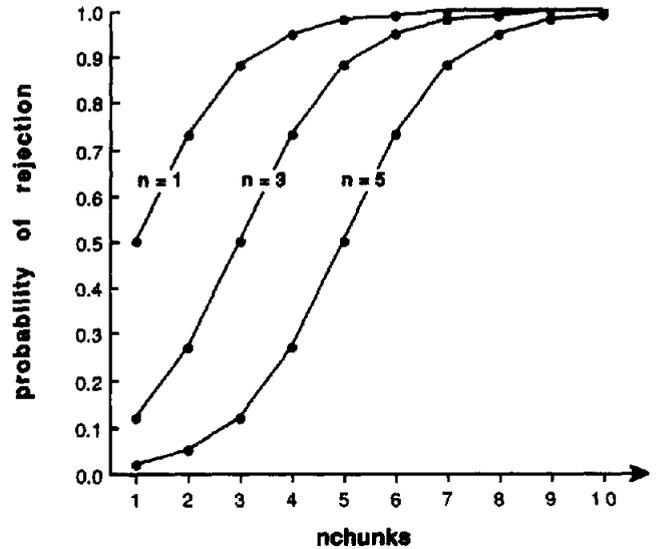


Figure 9. Probability that a string will be rejected as a function of $nchunks$. (The sigmoid is inflected at point $[n, .5]$ where n is computed on every trial as the average value of $nchunks$ in the previous few trials.)

Another assumption we made was that any string of letters implicitly includes extremity makers that signal the beginning and ending of that string. In the simulation we made these markers explicit in the representation of a string. Hence, the representation of a string like T T X V P S was actually "begin T T X V P S end." The extremity markers were treated like single letters as elementary chunks (except that they did not contribute to the lengths of word chunks). This assumption was important because it helped explain why subjects are apparently more sensible to grammatical violations occurring at the extremities of a string than to those occurring in the middle of a string, as first observed by Reber (1967) and as is evident in the data of our Experiment 1 (compare all-words/good-final with all-words/bad-final, and nonwords/good-final with nonwords/bad-final, in Table 3 or in Figure 10).¹¹ Those strings that have extreme groups of letters unmatched by extreme chunks cannot integrate the extremity symbols in their chunked representations and hence yield increased $nchunks$, which increases the probability that they are rejected.

The values of c and d that we used for the Reber-CC simulation were the same ones that yielded a good qualitative match to the data in the Miller-CC simulation: $c = d = .5$. Figure 10 plots, in each condition, the mean percentages of strings rejected within each string type by our subjects and by

¹¹ The set of test strings we used in Experiment 1 did not allow us to test systematically for subjects' sensibility to violations occurring at the beginning of a string, although it allowed us to test systematically for enhanced sensibility to violations occurring at the end of a string. Given the grammar used in that experiment and the format of choice for structuring and generating the nongrammatical strings (see Figure 3a), the two possible initial words were unfortunately reduced to a single letter each. This contrasted with the richness of the final words.

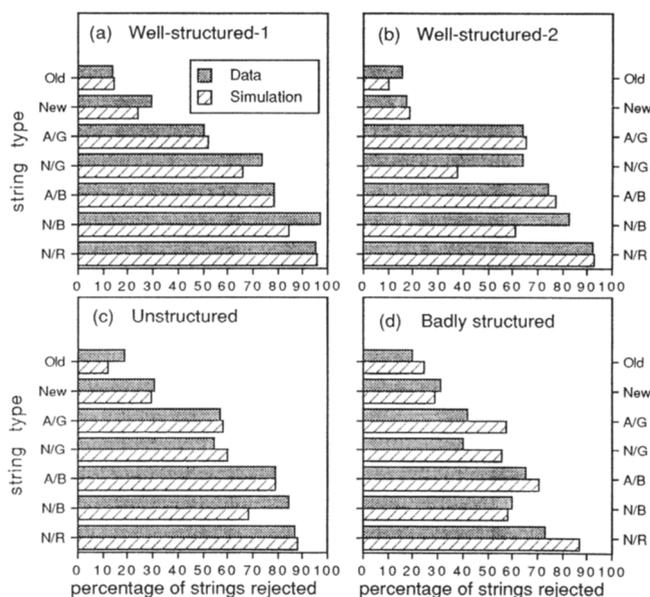


Figure 10. Human and simulated data in the discrimination task of Experiment 1: mean percentage of strings rejected within each string type (a) in the well-structured-1 condition, (b) in the well-structured-2 condition, (c) in the unstructured condition, and (d) in the badly structured condition. (The names of the string types are abbreviated as follows: Old = old-grammatical; New = new-grammatical; A/G = all-words/good-final; N/G = nonwords/good-final; A/B = all-words/bad-final; N/B = Nonwords/bad-final; N/R = nonwords/random.)

Reber-CC. The coefficient of correlation between the human and simulated data is .935.

This simulation demonstrates that chunking the exemplar sentences into hierarchical codes yields a data base of chunks that is complex enough to be used to discriminate sentences from nongrammatical strings. Judgments of grammaticality were based on an overall familiarity measure, given a set of exemplars. This is quite different from considering a string's similarity to a single exemplar, as nonanalytic models do (Jacoby & Brooks, 1984). Reber-CC could easily generalize to new-grammatical sentences because those were bound to exhibit some of the "grammatical" chunks (words, phrases) created from the exemplars. The more chunks transferred and the higher the level of the transferring chunks, the more familiar the new sentence was perceived to be. In contrast, nongrammatical strings were less likely to contain "grammatical" chunks; hence, they were more likely to be rejected. This account of the discrimination process is very much in accord with its characterization by Reber and Lewis (1977) as a dual process in which both specific information (chunks) and global apprehension (familiarity) play a role.

It is interesting to try to reconcile our account with Dulany et al.'s (1984, 1985) apparent success at describing subjects' grammatical knowledge in terms of rules. These investigators tried to assess subjects' knowledge by asking them to explicitly mark the part of a test string that made it grammatical or nongrammatical. Subjects were supposed to underline a piece of a string if they accepted it or cross out a piece of a string if

they rejected it. Dulany et al. considered each mark as a manifestation of a conscious rule of the form: *The presence of this group of letters implies that this string is (or is not) grammatical.* They found that the rules reported in this manner by their subjects predicted the subjects' discrimination behavior, although each was of limited scope and imperfect validity. They took this as evidence that such rules (a) existed and (b) were not forced justifications of judgments based on more abstract representations. Our theory, of course, denies both points. Reber-CC was able to simulate the discrimination behavior of our subjects without recourse to any explicit rules of the form proposed by Dulany et al. Our theory is at an advantage, however, because it accounts not only for the discrimination behavior but also for the learning by chunking behavior in the memorization task. In contrast, Dulany et al. offer no precise account of either the learning process or the representation of the memorized exemplars.¹² Thus the question becomes: On what basis would Reber-CC mark strings that it accepts or rejects so that these marks could be interpreted by Dulany et al. as explicit rules that predicted subjects' discrimination behavior? We are not sure. We note that Reber-CC was never guaranteed to always reject or always accept the same string. The probability that a string was accepted really depended on how familiar it appeared (i.e., how compact a percept was built) relative to how familiar the previous few strings appeared. Perhaps, if Reber-CC decided to accept a string, it would justify that by underlining the strongest chunk. And if it decided to reject that same string, it would justify that by crossing out a few adjacent letters in two adjacent chunks that are not integrated into a larger chunk.

CC assumes that the strength of chunks automatically decays with time. After a long enough delay, and no practice or retraining, the strengths of all word and phrase chunks would effectively be brought down to 0. Given the finding by Allen and Reber (1980) that even after 2 years, and without retraining, subjects are still able to discriminate grammatical from nongrammatical strings, we wondered whether such a drastic loss of information would render Reber-CC incapable of discrimination. It doesn't. Recall that CC assumes the strength and support of elementary chunks (i.e., letters in Reber-CC) to be large and constant (because they are task independent). Therefore, even after a long delay, word chunks, with the support of elementary chunks, are still readily available to the perception process. Eventually, as they gain strength again from their use by the perception process, they can support the retrieval and use of phrase chunks, which themselves can eventually support the retrieval and use of higher level chunks. In that way, most of the grammatical knowledge can quickly be reactivated. We checked the correctness of this explanation with a simulation of the delayed

¹² We note that Druhan and Mathews (1989) have taken Dulany et al.'s analysis seriously enough to make such rules the basis of their THYIOS simulation model, with some success. However, although they successfully simulate the tuning of such rules, they fail to offer an account of how these rules come to be formed. We believe that such rules could be formed only if some sort of chunking goes on during training.

Table 6
Mean Percentages of Grammatical and Nongrammatical Strings Rejected by Reber-CC Immediately After Training (With Unstructured Stimuli) and After a Long Delay

String type	Immediate test	Delayed test
Grammatical	20.7	26.6
Nongrammatical	66.0	61.7

discrimination task. Ten simulated subjects went through a training phase using unstructured stimuli, an immediate discrimination test, and a delayed discrimination test. The delay was set sufficiently large so that all chunks (but letters) had their strengths brought down to less than .01. The parameters *c* and *d* were set at .5 as in the other simulations, and the material was the same as in the Reber-CC simulation. The results are in Table 6. Clearly, the simulated subjects were still able discriminators after a long delay.

We hope to establish CC as the most viable candidate theory of the unintentional learning phenomenon, at least in the Reber task. We have already shown that it is a detailed and integrated framework in which to consider, at least, issues of learning, representation, and grammatical discrimination. It is also interesting to try to consider, within CC, the less mechanistic issues that previous investigators have tended to focus on (see Reber, 1989, for a review).

Foremost is the issue of consciousness. To what extent is the learning process operating outside of consciousness? And to what extent is the resulting knowledge available for conscious inspection? The first question is not very interesting because all processes of mind are unconscious; we can be aware only of the results of processes.¹³ Perhaps what is usually meant by this question is to what extent the learning process is primitive and automatic as opposed to controlled. We think of learning in CC—the combination of chunk-creation and strength-revision processes—as a process that is extremely primitive¹⁴ and going on all the time. We assume that whenever attention is paid to complex stimuli, chunks are created and their strengths revised automatically. Perhaps chunk creation occurs at a relatively constant rate, as is proposed in SOAR. Reber and his colleagues have used two types of neutral exemplar processing instructions: memorization and observation. The observation instructions have typically resulted in lower discrimination performance than have the memorization instructions.¹⁵ Although we think that chunk creation occurs under both types of instructions, it seems natural that instructions to memorize the exemplars should yield a richer set of chunks because they require that every exemplar be fully encoded at least once.

The consensus answer to the second question is that the knowledge produced by the learning process is at least partly unverbalizable, or unconscious (but see Dulany et al., 1984 and 1985, for a dissent). CC can easily accommodate that. Knowledge is encoded not only in chunks but also in their supports and relative strengths. Although one may be aware of a chunk to the degree that it is retrievable, its strength and support, which determine if it can be retrieved and how efficiently it can compete against other chunks, may well be

unconscious information. Druhan and Mathews (1989) make a similar point with their model of the grammatical knowledge as a set of competing conscious rules, each with an associated unconscious strength.

Note that CC is only a theory of immediate perception and automatic learning. Intentional, controlled, behavior is not within its scope. But most real-life learning situations are confronted with a blend of automaticity and control. Hence CC is not necessarily silent about such situations. For instance, Reber et al. (1980) have looked at the effect of mixing neutral instructions to observe exemplars with an explicit tutorial about the underlying grammar. The tutorial was given either before, during, or after the exposure to exemplar sentences. Reber et al. found that performance on a subsequent discrimination test was better the earlier in the training the tutorial was given. Their explanation is that the later in the training the tutorial is given, the more personalized knowledge the subjects have and the more difficult it is to reconcile the two. This explanation makes sense within CC. If a large set of chunks already existed when the tutorial was given, the subjective grammatical knowledge that it embodies would be likely to deviate substantially from the actual experimenter's grammar. But an early tutorial may help bias CC to create a set of chunks that is more economical, more in tune with the underlying grammar.

One aspect of other investigators' theorizing is difficult to reconcile with CC. It is the characterization of grammatical knowledge as abstract, not tied to the specific set of letters used to instantiate the grammar during training. In contrast, the chunks CC creates are very much tied to that letter set and would be unusable to perceive strings that instantiate the same grammar but with a different set of letters. Both Reber (1969) and Mathews, Buss, Stanley, Blanchard-Fields, Cho, and Druhan (1989) have looked at transfer from a grammar with one letter set to the same grammar with a different letter set and observed that it was non-null. These findings support the view of grammatical knowledge as abstract. However, Mathews et al. also found, for subjects with neutral instructions ("memory" conditions), significant performance deterioration whenever the letter set was changed. Learning was also greatly facilitated by constancy in the letter set. These findings support the view that grammatical knowledge is tied to a specific letter set. At the moment, CC simply cannot explain the findings in favor of abstraction.

Conclusion

We investigated the unintentional learning phenomenon demonstrated by Miller (1958) and Reber (1967). Previous

¹³ If that was not the case, then introspection would not have been so totally discredited as a method for building psychological theories.

¹⁴ Terrace (1987) reports evidence that even pigeons exhibit chunking behavior.

¹⁵ There is a single anomalous case (Reber & Allen, 1978) in which instructions to simply observe the exemplars produced discrimination performance significantly higher than instructions to memorize them (see Reber, 1989, for a comparative review). However, the subject population was also atypical, consisting of specially trained and unusually motivated students.

investigators already agreed that the basic unit of grammatical knowledge in the Reber task is a small group of letters. Reber and Lewis (1977) called it a *bigram or trigram covariation pattern*, Dulany et al. (1984) called it a *feature*, and, loosely following Miller (1958), we called it a (word) *chunk*. The results of our two experiments demonstrated the important role that some sort of chunking process played in this task. We built a precise theory of such a process called *competitive chunking* (CC).

To summarize, CC is a theory of immediate perception and automatic learning. It has processes for chunk creation, chunk retrieval, and percept elaboration, which are all mediated by chunk strength. Two parameters determine how easy it is for chunks to retain their strength (*d*, the decay parameter) and how easy it is to retrieve chunks (*c*, the competition parameter).

We found that, given a particular pair of values for its competition and decay parameters, CC was able to reproduce the important aspects of Miller's (1958) data on the effect of redundancy on free recall—namely, that the advantage of a redundant stimulus field increases with successive free-recall trials. Study strings were encoded into hierarchical chunks. The more redundant the list of strings was, the fewer chunks needed to be created and the stronger they were. Because chunk strength positively mediates chunk creation, chunk retrieval, and percept elaboration, the more redundant list of strings was more easily encoded.

Using that same pair of values for its two parameters, and an additional process assumption for making grammatical judgments based on the output of its perception process, CC was also able to capture 87% of the variance in the discrimination data of our Experiment 1—a Reber task analogue. Training exemplars were encoded into hierarchical chunks, resulting in a network of “grammatical” chunks. Those chunks were then used to perceive the test strings, resulting in more or less compact percepts. The more compact a percept, the more familiar a string was perceived to be and the more likely it was to be considered grammatical. Because grammatical strings were more likely to exhibit “grammatical” chunks than nongrammatical strings, discrimination was possible.

References

- Allen, R., & Reber, A. S. (1980). Very long term memory for tacit knowledge. *Cognition*, 8, 175–185.
- Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Bower, G. H., & Springston, F. (1970). Pauses as recoding points in letter series. *Journal of Experimental Psychology*, 83, 421–430.
- Bower, G. H., & Winzenc, D. (1969). Group structure, coding and memory for digit series. *Journal of Experimental Psychology, Monograph Supplement*, 80, 1–17.
- Chase, W. G., & Ericsson, K. A. (1981). Skilled memory. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 141–189). Hillsdale, NJ: Erlbaum.
- Dulany, D. E., Carlston, R. A., & Dewey, G. I. (1984). A case of syntactical learning and judgment: How conscious and how abstract? *Journal of Experimental Psychology: General*, 113, 541–555.
- Dulany, D. E., Carlston, R. A., & Dewey, G. I. (1985). On consciousness in syntactical learning and judgment: A reply to Reber, Allen, and Regan. *Journal of Experimental Psychology: General*, 114, 25–32.
- Druhan, B. B., & Mathews, R. C. (1989). THYOS: A classifier system model of implicit knowledge of artificial grammars. *Proceedings of the 11th annual conference of the Cognitive Science Society*, 66–73.
- Johnson, N. F. (1970). Chunking and organization in the process of recall. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 4, pp. 171–247). New York: Academic Press.
- Jacoby, L. L., & Brooks, L. R. (1984). Nonanalytic cognition: Memory, perception, and concept learning. In G. Bower (Ed.), *The psychology of learning and motivation* (Vol. 18, pp. 1–47). New York: Academic Press.
- Klahr, D., Chase, W. G., & Lovelace, E. A. (1983). Structure and process in alphabetic retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 9, 462–477.
- Mathews, R. C., Buss, R. R., Stanley, W. B., Blanchard-Fields, F., Cho, J. R., & Druhan, B. (1989). Role of implicit and explicit processes in learning from examples: A synergistic effect. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15, 1083–1100.
- Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81–97.
- Miller, G. A. (1958). Free recall of redundant strings of letters. *Journal of Experimental Psychology*, 56, 485–491.
- Morgan, J. L., Meier, R. P., & Newport, E. L. (1987). Structural packaging in the input to language learning: Contributions of prosodic and morphological marking of phrases to the acquisition of language. *Cognitive Psychology*, 19, 498–550.
- Newell, A. (in press). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- Reber, A. S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior*, 5, 855–863.
- Reber, A. S. (1969). Transfer of syntactic structure in synthetic languages. *Journal of Experimental Psychology*, 81, 115–119.
- Reber, A. S. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General*, 118, 219–235.
- Reber, A. S., & Allen, R. (1978). Analogic and abstraction strategies in synthetic grammar learning: A functionalist interpretation. *Cognition*, 6, 189–211.
- Reber, A. S., Kassin, S. M., Lewis, S., & Cantor, G. (1980). On the relationship between implicit and explicit modes in the learning of a complex rule structure. *Journal of Experimental Psychology: Human Learning and Memory*, 6, 492–502.
- Reber, A. S., & Lewis, S. (1977). Toward a theory of implicit learning: The analysis of the form and structure of a body of tacit knowledge. *Cognition*, 5, 333–361.
- Terrace, H. S. (1987). Chunking by a pigeon in a serial learning task. *Nature*, 325, 149–151.
- Tulving, E. (1962). Subjective organization in the free recall of “unrelated” words. *Psychological Review*, 69, 344–354.

(Appendix follows on the next page)

Appendix

The Five Successive Training Sets Used in Experiment 1, in Each of the Four Conditions

Condition			
Unstructured	Well-structured-1	Well-structured-2	Badly structured
Set 1			
TPPPTS VXXXXVPS TPPTXVPS VVPXVS	T PPP TS V XXXX VPS T PP TX VPS V VPX VS	TPPPT S VXXXX VPS TPPT X VPS VVP XV S	TPP PTS VXX XXVP S TP PTX VPS VV PX VS
Set 2			
VXXVPS TTXXXVS TTXVPS VVPXXVS	V XX VPS T TX XX VS T TX VPS V VPX X VS	VXX VPS TT XXXVS TTX VPS VVP XXVS	V XXV PS T TXX XVS TTX VP S VVP XX VS
Set 3			
TPPTXXVS VXVPXVPS TPPPPTS VXXVPXVS	T PP TX X VS V X VPX VPS T PPPP TS V XX VPX VS	TPPT XXVS VX VP X VPS TPPPPT S VXX VP XVS	TP PTX XVS VX VPXV PS T PPPP TS VXX VPX VS
Set 4			
TTXVPXVS VVPXXVPS VXXXXVS TTXXVPS	T TX VPX VS V VPX X VPS V XXXX VS T TX X VPS	TTX VP XVS VVP XX VPS V XXXXVS TTXX VPS	TT XVPX VS VV PXX VPS VXXX XVS TTX VS PS
Set 5			
TPTXXVPS VVPXXXVS TPPPTXVS VVPXVPS	T P TX X VPS V VPX XX VS T PPP TX VS V VPX VPS	TPT XX VPS VVP XXXVS TPPPT XVS VVP X VPS	TPT XX VPS V VPX XXVS TPPP TX VS VVP XVP S

Received March 3, 1989
Revision received August 7, 1989
Accepted December 11, 1989 ■