CHAPTER TWELVE

Chunking, Familiarity, and Serial Order in Counting

Wayne A. Wickelgren
Columbia University

At a Little League game awhile back, my 9-year-old son Peter told me, "Dad, this is a six Jason game!" By this he meant that there were six kids named Jason in the game—two on his team and four on the other. *How* one might recognize this is the topic of the present chapter. *Why* one might recognize it is another matter.

This chapter analyzes sequential counting and is not concerned with the discrimination of different numbers of simultaneously presented stimuli via subitizing mechanisms. Throughout this chapter, *counting* means *sequential counting.*

The chapter focuses on sequential counting for successive events where there is a substantial time interval between repetitions. When we sequentially count a set of simultaneous stimuli, we may use many of the same mechanisms required for sequential counting of successive events, but I have not considered this question in any detail. Thus, the emphasis here is quite different from that of Gelman and Gallistel (1978), for example, who concentrated on sequential counting of simultaneous stimuli.

This chapter considers sequential counting to require three basic competences: chunking, recognition of repetition, and the capacity to represent serial order. The serial order capacity corresponds to Gelman and Gallistel's stable-order principle. Neither repetition recognition nor chunking is present at all in Gelman and Gallistel's list of five basic principles that "govern and define counting," though the one-one principle has some similarity in function to repetition recognition and, there is a

small functional overlap between chunking and Gelman and Gallistel's cardinality and abstraction principles.

It surprised me how much Gelman and Gallistel's model of counting relies on the simultaneous availability of the things to be counted and does not easily apply to counting successive events with long time intervals between repetitions. This is particularly noteworthy because Gelman and Gallistel argued against the claim that subitizing plays much of a role in the perception of even small numerosities of objects, and, like me, they are primarily concerned with sequential counting.

Gelman and Gallistel were also interested in children's understanding of number concepts, not just the ability to count, which I do not address at all. By contrast, I wish to make some progress toward more mechanistic, neural net, models of counting. Thus, although we share a focus on cognitive sequential counting by humans, there is very little overlap between this chapter and Gelman and Gallistel (1978).

The two major goals for this chapter are (a) to analyze the semantics of sequential counting by human beings in the sense of understanding what kinds of repeated events people count and (b) to analyze three component mechanisms that may play important roles in counting—chunking, repetition recognition, and serial ordering. The discussion of these mechanisms focuses primarily on a semantic or functional analysis of what these mechanisms accomplish and secondarily on making some progress toward a neural model of these mechanisms.

REPETITION, FAMILIARITY, AND NOVELTY

Semantics of Counting: The Role of Repetition Recognition

Counting Jasons is unusual in a baseball game, but counting balls, strikes, runs, and so forth is typical. How do you know when you have struck out? Umpires often use a mechanical or electronic counter to aid them, but batters usually know the count without the aid of auxiliary devices. Coding the number of strikes is paradigmatic of the human capacity for counting events.

Counting is not identical to coding duration, though there are times when we count the seconds. We can count events that occur with irregular time intervals between them, and experimental studies of counting often design counting tasks so that time duration is not too highly correlated with number of repetitions. Theoretical mechanisms for count-

ing are likely to differ in some respects from mechanisms for encoding duration.

Counting is not limited to objects. We can count events of any type, including actions such as swinging a bat.

Counting is not limited to events that are identical or even very similar in their sensory qualities. The Jasons in Peter's game were all young boys, but they did not otherwise look much alike. The similarity was in their names, which I grant you is a physical sensory similarity. However, Peter did not recognize the similarity from hearing or seeing their names. He recognized the six Jason game by associative memory for the names from their disparate visual appearances. Finally, a strike in baseball is a disjunctive class of events that have no common physical component property that distinguishes them from balls, foul balls, hits, and so forth. There are called strikes that are over the plate and between the knees and the letters. There are swinging strikes where the batter misses the ball completely, foul ball strikes, and tipped balls that are caught by the catcher. In all of these latter cases it is irrelevant to classification as a strike whether the ball was in the strike zone.

However, although sensory, motor, and physical similarity are not necessary for counting, conceptual similarity of the events being counted is basic to counting and the recognition of repetition is a fundamental component of counting. What is repeated may be sensory or motor events that are nearly identical in some cases or merely possessing a single common sensory, motor, or cognitive attribute in other cases. We can count repetitions of bell rings, the number of objects found in a box, or the number of different outcomes in five throws of a die, but in all cases, there is an idea that classifies what is and is not to be counted. It is the ability to recognize that the representative of that idea has been activated repeatedly that lies at the base of our ability to count events.

I would like to dispense with a tricky problem by fiat. Although it can be argued that no two experiences are ever identical or identically encoded, it is clear that we often encode two events as equivalent in certain respects. I assume that in such cases there is some aspect of our representation of each event that is identical, though, to be sure, there are almost always other aspects of our encoding that are different on the two occasions. In my model, to recognize the repetition, the repeated part must at some point be the focus of attention, that is, be the complete thought active at that point in time. Perhaps the human mind is able to recognize and count repeated partial thoughts, but the model presented here only counts repeated complete thoughts.

Repetition, Recognition, and Recognition Memory

I remember that as a child I read *Treasure Island* seven times. My memory for that is probably no different from any other memory in how it is stored. But my ability to update the count each time I read it points out over how long a period of time one can count events and how extended and temporally complex the events may be. Like Dorothy following the yellow brick road to the Emerald City, I began at the beginning and, keeping my place via bookmarks and memory for what I had recently read, I continued to read, with frequent interruptions, to the end of the book, whereupon I activated the concept of "finishing *Treasure Island.*"

The first time I did this, I may well have simply encoded that I had read *Treasure Island,* not that I had read it once, though this is implicit. The second time I finished reading it and activated the "finishing *Treasure Island*" node, I somehow recognized that this had happened before. I recognized a repetition.

The recognition of repetition in this case is probably identical to what happens in recognition memory tests. We have a feeling of familiarity that is greater, on the average, for events that have been encoded by our minds in the past than for novel or unattended events. Thus, it would appear that explaining the ability to recognize repetition is identical to explaining recognition memory, though it is quite possible that this ability is mediated by more than a single mechanism and different mechanisms predominate in different cases, such as at short versus long time intervals between repetitions.

Familiarity/Novelty Detection Mechanisms

What are some possible mechanisms for recognizing repetitions in biological minds?

Immediate Repetition. One plausible algorithm for detecting immediate repetition in neural nets is to compute the sum of the absolute differences between the activation levels of each neuron at adjacent time periods as a fraction of the total activation level. Second-order immediate-change (velocity, difference, or derivative) detectors are quite common in real nervous systems—for example, motion, brightness changes, pitch transitions. Often some neurons respond to increases and others to decreases. The sum of the absolute differences would just be the sum of activity in both positive and negative change detectors. The total activation level (for the first-order neurons) must also be computed. Computing total activation in various subsets of neurons is relatively easy for

neural nets. The primary problem in computing total activation is handling the wide range of totals, and real neural nets have obviously solved such range problems, as witnessed by our ability to encode brightness over an enormous range of light intensities.

A neural mechanism that encoded this sum of absolute changes over total activation is an immediate novelty detector, where immediate means over a time period on the order of a millisecond or few milliseconds. This mechanism computes novelty first, with repetition being signaled by very low novelty.

Clearly, this is not the sort of repetition-detection mechanism we need for cognitive counting, because the likely size of a time step for immediate repetition in the nervous system is on the order of a millisecond or a few milliseconds (basically neural communication delay times—axonal, synaptic, dendritic, and spike-generating delay times), and we can count events with seconds to years separating repetitions.

Long-term Repetition

Successor Thoughts. Perhaps we distinguish familiar versus novel thoughts on the basis of the different properties of the thoughts that follow the given thought. Although it may not always be true that recognition of familiarity implies some ability to recall associated ideas, it is often the case that when we recognize a familiar event, other ideas pop into our minds by association. Perhaps it is the immediacy and/or strength of activation of successor thoughts to a given thought that identifies that given thought as familiar versus novel.

For two reasons, I do not regard a mechanism based on the properties of successor thoughts to be as likely a basis for the identification of familiarity as a mechanism based on the properties of the given thought itself. First, such a mechanism is almost necessarily slower than one based on properties of the given thought. Second, because unfamiliar thoughts have familiar components and familiar components have associations, successor thoughts may well not provide much basis for distinguishing whether or not the prior thought was familiar or unfamiliar. In any case, I have no clear conception of how the properties of successor thoughts might distinguish familiar versus novel prior thoughts.

Chunking. It is likely that the mechanism for familiarity recognition is based on different properties of familiar versus novel thoughts during the activation of the given thought itself. The first time a particular thought is activated, chunking adds a new idea representative to the set of ideas and strengthens up associations from the constituents to the chunk and down associations from the chunk to its constituents. Accord-

ing to the basic hypothesis, a thought is judged familiar to the extent that it is already strongly associated to a single chunk idea. There are many possible specific mechanisms for this, three of which are discussed briefly here.

First, a thought may be judged to be novel if it triggers the chunking process, and familiar if it does not. The component of the chunking process that selects a new idea representative seems like an all or none event, whereas familiarity appears to come in various degrees. However, the strengthening of up and down associations could be a graded process that occurs to the greatest extent for novel or forgotten thoughts. Thus, it may be that some difference in the learning processes connected with chunking is the basis for familiarity recognition. However, because some property of the thought itself must trigger the chunking process, it seems more reasonable to make familiarity dependent on the earlier trigger property than on the later learning consequence.

One such trigger property might be complexity of coding or some other property that depends on whether or not there already is a single (chunk) idea to encode the current thought. That is, when an attention span of ideas is activated, retrieval may just automatically proceed to the highest, most economical level of coding, with chunks inhibiting their constituents after reaching full activation. If the final thought contains but a single chunk, then the thought is judged familiar. If the thought contains more than one idea, it is novel.

Familiar thoughts would have fewer active neurons than unfamiliar thoughts, and hence a control neuron or set of neurons that measures total activation would be a novelty detector. Novel thoughts would have more active neurons than familiar thoughts. Perhaps the recognition of novelty is based on activation of one or more neurons that innately encode the total activation of all neurons in some module or set of modules of the mind. Conscious recognition of novelty is limited to those modules of the mind of which we are conscious. These are presumed to be certain higher-order cognitive modules, and it is only in cognitive modules that I think chunking takes place (Wickelgren, 1979b).

However, it is not clear to me that thought activation automatically proceeds to the highest level in a single step. I think it is more likely that this requires several steps, with a set of constituents being activated in one step, and their chunk idea, if any, activated in the next step. It may be that the familiarity of an activated set of ideas is determinable from the activation of the set, before the chunk idea, if any, is activated. One way to determine the familiarity of a thought is as follows:

First, I assume that the human mind has multiple types of associations and can selectively enable and disable each type of idea and each type of association during different phases of thinking. Second, I assume that

the mind has the ability to inhibit the currently active thought. Third, I assume that the mind has the ability to vary the maximum number of active ideas (neurons) in any particular phase of thinking.

To judge the familiarity of the active thought at time *t,* do the following for the transition to the next time step: (a) enable only the up associations, (b) set the threshold for activation of ideas sufficiently high so that only an idea with strong enabled input associations from all of the ideas in the prior thought could be strongly activated, and (c) temporarily inhibit the currently active thought. Then on or immediately after time $t + 1$, a control idea that judges the total activation at time $t + 1$ is a familiarity detector. The thought whose familiarity was to be judged was inhibited and only up associations were enabled from time t to $t + 1$, so only an idea with strong up associations from all of the ideas in the prior thought could be strongly activated at time $t + 1$. To the extent that any chunk idea is activated strongly, the familiarity idea, which judges total activation or maximum activation, will be strongly activated too.

This familiarity mechanism differs from the successor thought mechanism in limiting possible successors to ideas with strong up associations to the ideas in the current thought, whereas the successor thought mechanism permitted successor ideas with any of a variety of relations to the ideas in the prior thought, implicitly enabling all types of associations, including sequential associations.

Additional discriminative power is gained by setting activation thresholds high enough so that only a single idea receiving strong inputs from all of the ideas in the prior thought can be strongly activated.

Note that this familiarity mechanism uses the same sort of control idea measuring total activation as was used by the novelty mechanism discussed previously. First, this familiarity mechanism ought to give graded values of familiarity for repeated thoughts with different strengths of up associations to the chunk idea due to differences in forgetting or degree of learning. Second, the familiarity mechanism uses information gained from the very next time period after inhibition of the thought, which in a distributed neural net with attractor dynamics is generally many time steps prior to achieving a stable next thought. Thus, the familiarity mechanism acts fairly quickly. Third, this familiarity mechanism can be used to decide whether or not the mind should attempt to retrieve an existing chunk idea for this thought or select a new one. In my current neural modeling, this means enabling different types of links.

Short-term Repetition

Identification of repetition via a familiarity mechanism based on chunking seems highly appropriate for long-term memory counting, such as counting the number of times I read *Treasure Island.*

However, whether or not the current thought is already represented by a single chunk seems less appropriate for what could be called *short-term repetition,* namely, recognizing repetition, not over one's lifetime, but over some short period on the order of a few seconds. Such short-term repetition is still very long compared to what I referred to as *immediate repetition,* which was based on the cycle time of the nervous system—on the order of milliseconds. So the question is, do we need a third mechanism for repetition recognition over an interval of seconds?

It does not seem practical to develop a short-term repetition recognition mechanism out of the immediate repetition mechanism, because that mechanism uses second-order change neurons that monitor the activity of the first-order neurons and measures something approximating the derivative of the activation of the first-order neurons. In simpler language, such second-order neurons encode how much the first-order neurons changed activation from time step t_1 to time step t_2. To compute such differences between two nonadjacent activation states requires storing a sequence of activation states. This seems impractical over more than one or two intervening states and is highly implausible over a period as long as several seconds. Thus, we may confidently rule out extension of the immediate repetition mechanism to handle short-term repetition.

Consider an example of short-term repetition to see whether it can be handled by the long-term repetition mechanism. Imagine you are walking along the street and a nearby bell tower begins to ring out the hour of the day via a series of bongs. You count the bongs. How? You have heard these bongs many times before. You do not start the count where you left off last time, say at 347, and count 347, 348, 349, 350. No, you count 1, 2, 3, 4. Aha, four o'clock, you say to yourself.

There is difference in what is counted in short-term versus long-term recognition of repetition. Indeed, if you have been in this neighborhood on only a few occasions, you might also remember on how many prior occasions you have heard this bell chime before, perhaps on two prior occasions. You note that you have now heard this bell ring out the hour on three different occasions in your lifetime.

Nevertheless, the long-term recognition mechanism based on chunking can handle short-term recognition. Analysis of this problem deepens our appreciation of the cognitive complexities of repetition recognition and counting. The bell ring is familiar, so we could just recognize that and count lifetime frequency of our hearing this bell ring. But the context in which we hear the bell ring is different from past contexts (that is, the other ideas active in our mind are different), and we can form a chunk to represent the idea of the bell ring in this context the first time we hear the bell in the present context and thereafter count repetitions in just this context. The first time we hear the bell ring in the present

context, we may just identify the bell ring as a familiar chunk in our long-term memory or we may follow this by encoding a new chunk of the bell ring in the present context. If we want to know what time it is, we must do the latter and count bell rings in this context alone, not over our whole lifetime.

SERIAL ORDER

One basic competence that underlies counting is knowing the order of the numbers, for example, being able to recite the sequence, "1, 2, 3, 4, 5, and so forth." Gelman and Gallistel (1978) referred to a slightly more general version of this competence by the term *stable order.* One might refer to this competence as rote counting, to distinguish it from other counting competences, such as repetition counting or object counting.

One might assume that representing the order of small positive numbers is more fundamental than, or at least different from, representing the order of items in other sequences, such as letters in written words or the alphabet or phonemes in spoken words. However, it seems more parsimonious to make the working assumption that the representation of ordered sets is the same in all of these cases.

I divide models of serial order into two large categories: nonhierarchical models that do not assume chunking to play any role in the coding of serial order and hierarchical chunking models that assume that long sequences are broken into subsequences. Subsequences are chunks, but these chunks have ordered sets of constituents.

Nonchunking Models

I do not regard nonchunking models of serial order as even remotely tenable, but brief consideration of them uncovers relevant types of associations and sets the problem in historical perspective.

Item-to-Item Association. In an associative memory, the simplest model for coding the order of the small positive numbers is to assume that there is a strong association from the 1 idea to the 2 idea, a strong association from the 2 idea to the 3 idea, and so forth.

However, a phenomenon often observed in children suggests this assumption is, at best, only a partial account of the associations involved in rote counting. The phenomenon purports that even children who are very accurate at counting from 1 to 10 often have considerable difficulty telling you the successor of some number between 2 and 9, for example,

giving the answer to the question, "What comes after 4?" The prior number is not the only cue to the next number in the counting sequence.

Position Coding and Grouping. Essentially the same conclusion can be derived from a large number of studies concerned with determining the nature of the effective stimulus in serial list learning, namely, the prior item is not the only cue (Wickelgren, 1977, pp. 235–236; Young, 1968). People also use serial position as a cue.

Whether or not we use serial position as a cue for the next counting number is an interesting question to contemplate. At first, it may seem logically circular, like using an idea as a cue to itself. However, it is likely that serial position ideas are different from number ideas. Studies of serial position coding in short-term memory suggest that subjects often use only three different serial position ideas—beginning, middle, and end. However, they use these three position ideas in a cross-classification scheme at two different levels—beginning, middle, and end groups, and beginning, middle, and end positions-within-a-group (Wickelgren, 1964, 1967). In any case, it is very unlikely that we ordinarily use as many serial position ideas to code the order of items in a sequence as we have distinct counting number ideas.

Remote Associations. In addition to the prior item and serial position cues, we may also make some use of earlier, more remote, items as cues to the next item in a sequence. There is no definitive evidence that I know of to support this, but it is introspectively compelling to assign some cue value to remote prior items beyond their role in cuing serial position. So, it may be that *q r s t u* is a better cue to the next letter *v* in the alphabet than *u* alone, because *q, r, s,* and *t* have remote associations to *v.*

Remote associations raise serious theoretical problems and have no definitive empirical support. There are more parsimonious explanations of the facilitatory effect of remote prior context including (a) more effective cuing of serial position, (b) inhibiting recall of the remote prior items as the successor item, and (c) cuing chunks representing sequences of items.

Thus, in naming the next letter of the alphabet, *q r s t u* is better than *u* alone as a cue to *v* for any or all of the following reasons: (a) It more effectively cues that the desired letter is at a late-middle position in the second half of the alphabet (small effect). (b) It rules out *q, r, s,* and *t* as possible successors to *u.* (c) For all of us who know the alphabet song, it cues the chunks for *qrs* and *tuv* much better than does the single-letter cue *u,* with the chunk for *qrs* being associated to its successor chunk *tuv.* There are other chunking explanations as well.

Chunking Models

Although chunking in human learning and memory is not well understood, it is clear that humans chunk sets of ideas to obtain a more abstract representation of the entire chunk that is about as easy and simple to think with as the constituent ideas (Miller, 1956; Wickelgren, 1979a, 1979b). Chunking undoubtedly occurs for both ordered and unordered sets, but in the case of ordered sets (sequences), some additional property must be specified to encode the order information.

Note that a chunk for a sequence is a plan representative for the execution of that sequence. One aspect of planning in this model is to have a single abstract idea for a plan represent a sequence of constituent ideas.

As I see it, any viable model for serial order in long-term memory must assume that down associations from a chunk idea prime the chunk's unordered set of constituents. When these down associations are strong, this restricts activation to the constituents of the chunk out of the vast set of all possible ideas. In models of memory that assume chunking of sequences, the lion's share of the memory for a sequence is carried by the associations that pick out the unordered set of constituents, but the small remaining share that orders these constituents is essential, and there are a number of possibilities.

Gradient of Down Associations. It is possible that there is a gradient of down associative strength such that the strongest down association from a sequence chunk is to its first constituent, then its second, third, and so forth. Once a constituent node has been activated for some time, I assume that it gets inhibited, so the next most strongly activated constituent can become the most activated. Thus, when a sequence node is activated, it first activates its first constituent, then its second, third, and so forth. Grossberg (1978) employed a variant of this gradient model for sequence generation.

I have never liked this alternative, because I think it would be difficult to establish properly ordered and discriminable strength levels in learning that would correctly order the constituents in sequential retrieval. Many factors such as the discriminability, duration, and number of repetitions of constituents presented as stimuli ought to affect the strength of down associations, in addition to delay (remoteness).

It is interesting to consider what sort of learning process would establish this gradient of down associative strength. Superficially, it might seem that a simple contiguity conditioning process would do, because after a chunk is activated in production, the initial constituent of the chunk is the first to be activated, followed by the second, and so forth. However, it is logically circular to explain the ordering in production by a gradient

in the strength of down associations and explain the gradient in the strength of down associations by the temporal ordering of the constituents in production. Some other learning process had to cause the gradient in down associative strength before the constituents of the sequence were first produced in the proper order.

The obvious candidate is the thought or perception that initially produced the sequence. But here the contiguity is probably reversed. The sequence of constituents is surely activated before the chunk representative is selected and activated, unless chunk representatives are selected in advance of their constituents and thus independently of them (assuming the usual forward direction of causality). If the first activation of a chunk follows activation of its constituents, if constituents of a sequence are activated in sequential order, and if down associations are strengthened in proportion to this contiguity, then it would seem that the strength gradient would be the reverse of what is required.

However, this is by no means necessary. For one thing, behavioral classical conditioning usually has an optimum interstimulus interval on the order of .5 s, not at 0. Perhaps all sequences that can be chunked are mentally activated within this half-second window yielding the desired gradient. A better argument offers that behavioral classical conditioning does not necessarily provide us with a direct window to observe the dynamics of associative change. We should not assume that all learning processes have the same temporal dependencies. Indeed, if up associations (uplinks) and down associations (downlinks) are different types of associations with different semantics and serving different functions, it would be quite reasonable for them to have different learning dynamics. Down associations might have a reversed strength gradient from up associations.

There is a more serious problem for the gradient model. In perception, constituents are not necessarily activated in the order of presentation. Longer, more intense, and, in general, more discriminable constituents may frequently be activated earlier, even if they are presented later. In a speed accuracy tradeoff study of speech recognition, Remington (1977) found that the medial vowel in a consonant-vowel-consonant (CVC) syllable is frequently recognized before the initial consonant. If perceptual recognition dynamics measures the dynamics of activation, this evidence suggests that neither recognizing the sequential order of stimuli, nor learning this order is dependent on the temporal order of activation. Although it is perfectly reasonable to permit the strength gradient model of order coding to use any functional relation between degree of learning and temporal contiguity of activation that achieves the functionally desired strength gradient, it is not obvious that the gradient model can survive order coding not being rigidly linked in some manner to the temporal order of constituent activation.

The Remington (1977) study also suggested that sequentially presented stimuli are often recognized in parallel. Although constituents with earlier stimulus onsets may increase in level of excitation earlier, the growth of excitation of all constituents of a short sequence may be heavily overlapping and the constituents that reach the threshold for activation first are often not those whose stimuli were presented first. After a constituent is activated, its activation may be maintained while other constituents are activated and each constituent may help to activate adjacent constituents by lateral sequential associative links prior to recruiting the chunk representative. Such dynamics of perception are difficult to square with the gradient model.

There is also another negative indicator for the model. In memory for sequences, it is almost always the middle items of the sequence that have the lowest probability of recall in response to a name for the sequence, suggesting, but not implying, that the down associations from the chunk idea to its middle constituents are weaker than those to either its initial or terminal constituents. The gradient model requires the association to the terminal constituent to be the weakest.

Jordan (1986) had an interesting discussion of the severe problems that the gradient model has with sequences that involve repeated elements, for example, *abcccd*. In such a sequence, the down association from the chunk to the *c* in the third, fourth, and fifth positions could easily be stronger than the association from the chunk to the *b* in the second position, causing *c* to be output before *b* erroneously. There are ways to overcome this problem. One way is to assume that the link strength to item i is greater than the sum of the link strengths to all items $j > i$. This requires a very large dynamic range of discriminable strengths for sequences of any appreciable length, but it might be viable for sequences of fewer than three or four elements. Another way is to recode sequences with repeated items into sequences of subsequences, where each sequence and subsequence has no repetition. This seems reasonable. Indeed, this latter approach may be needed to solve the problem of being able to activate a repeated item more than once, because an item is inhibited after it has been active for some time.

Despite many problems, a gradient of down associations from a sequence chunk is a possibility, especially if all long sequences are coded hierarchically such that no chunk has more than three or four constituents and no repeated constituents. The most attractive aspect of this alternative is the fact that only vertical associations are needed, up associations from constituents to the chunk and down associations from the chunk to the constituents. No lateral sequential associations are needed. All of the remaining models for serial order, with the possible exception of Jordan (1986), assume lateral associations somewhere, though not

necessarily between the representatives of the elements of the sequence themselves.

Contingent Association. As Lashley (1951) pointed out, items like letters and numbers appear in many different sequences, so lateral associations from one such letter or number to another could hardly be relied on to carry the order information for each and every different sequence in which those items appeared. If the strongest associate to 1 is 2, that is fine for recalling the order of the cardinal numbers, but not so fine for recalling the successor digit of the first 1 in the value of pi, 3.14159.

One way to use lateral associations among constituents to mediate serial order is to make the strength of these lateral associations contingent on which chunk idea is active. Assume that an idea can have associations, not just to other ideas, but also to the associations between other pairs of ideas. Neurally, this is analogous to a link to a link, a synapse on a synapse.

In particular, assume that a sequence chunk idea sends down associations to the lateral sequential associations among its constituents. That is, chunks enable constituent lateral associations as well as constituent ideas. When the cardinal number chunk node is active, the sequential association from 1 to 2 is enabled by a down association from the cardinal number chunk, but when the pi chunk node is active, it is the sequential association from 1 to 4 that is enabled.

Rumelhart and McClelland (1986) implement such contingent associations by what they called *sigma-pi units.* The excitation of a normal unit (neuron) in most neural nets is the simple sum of the inputs received from all input links. The excitation of a sigma-pi unit in the present case would be the sum of the products of the input received from each lateral input link multiplied by the input from the most strongly activated down associative link from a chunk idea to that lateral link. Hence the name "sigma-pi"—the sigma is the summation operation applied to a bunch of terms that are obtained by multiplying two or more factors (the pi operation).

Sigma-pi units or associations to associations (links-to-links) are complex to implement in a neural net in both retrieval and learning, which is one strike against any contingent association model. A second strike against this particular contingent association model of sequence ordering is clear from the 3.14159 example, namely, the model does not discriminate different successors to repeated items at different positions of the same sequence, for example, to the 1 in 3.14159. Contingent associations can do anything that simple associations can do, so there is no question that an adequate contingent association model can be devised, if a simple association model can. The issue is whether we need to as-

sume such more powerful and complex forms of associative memory. At this point, I think we do not.

Position Coding. Let the sequence *abc* be coded as *a1, b2, c3.* Let the sequence *cab* be coded as *c1, a2, b3.* When the constituents of a chunk are ordered, each constituent is itself a chunk with two constituents—a qualitative idea and a serial order idea. In retrieval of *abc,* the *abc* chunk excites its *a1, b2,* and *c3* constituents. Convergent excitation comes from a special serial order system that excites the "1" idea which, in turn, excites the "2" idea, and so forth. During the time the 1 idea is active, the 1 idea excites all of the chunks with a 1 constituent via up associations. This causes *a1* to be most strongly activated during the first time period of reciting *a1, b2, c3.* Via a down association from *a1* to *a, a* is the first item output in the sequence. Then via a strong innate or learned sequential association from 1 to 2, activation shifts to 2, which primes all of the chunks with 2 constituents, and so leads to the activation of *b2* and then *b,* and so forth.

Lashley's (1951) pioneering ideas concerning serial order are closer to position coding than to any of the other models discussed here, and MacKay (1987) developed this sort of serial order mechanism in considerable detail. MacKay's serial order mechanism is elaborated to handle serial order at each level of a multilevel hierarchy of constituent nodes. MacKay illustrated his model of sequence ordering by speech from the syntactic level of organization of phrases and words down to the phonetic levels of segments of words and features of segments. This is a necessary extension of any model of serial order, but it goes beyond the scope of this chapter.

Here I am concerned only with ordering the set of constituents of a single chunk. In position-dependent coding, a chunk first primes its unordered set of constituents, just as in every other model of serial order that assumes a role for chunking. In the position-dependent coding model the constituents of a chunk are somehow tagged for serial position, and a general serial order mechanism then activates the most strongly primed x1, followed by the most strongly primed x2, and so forth. The tags need not be labeled 1, 2, 3, and so forth, they could be labeled syntactic categories, such as "article," "number adjective," "size adjective," "color adjective," "noun," or "initial consonant group," "vowel group," as discussed in MacKay (1987). However, with respect to the control of serial order in production of a sequence, they might just as well be labeled 1, 2, 3. The labels become important when you relate these ordering ideas to other tasks. For example, if you want to cue a subject to produce one particular constituent of a sequence, then you need to use a label that communicates which constituent you want.

It is important to note that the position coding model of serial order has not eliminated sequential associations. To be sure, there are no sequential associations among the items of each and every learned sequence. But there are sequential associations among the generic position ideas. Consider the set of position ideas 1, 2, 3 used to order the sequence *a, b, c* by the coding *a1, b2, c3*. Neither the set of ideas *a b c,* nor the set *a1, b2, c3* needs to have sequential associations, but the set 1, 2, 3 does have to have sequential associations in order to impose its order on other sequences. There are two extant models of how this order is accomplished:

In the most obvious model, the command to produce a sequence activates the 1 idea. The 1 idea has its strongest sequential association to the 2 idea, the 2 idea has its strongest sequential association to the 3 idea, and so forth.

The less obvious model was suggested by Estes (1972) and was also favored by MacKay (1987). In this model the sequential associations are inhibitory, not excitatory. The 1 idea inhibits all the later position ideas. The 2 idea inhibits the 3, 4, and later ideas, and, in general, position idea i inhibits position idea j, if and only if $i < j$. This means that when the command to produce a sequence activates the set of position ideas, position 1 will win out initially. Then when the first item in the sequence has been produced, it must be inhibited, as must the position idea that activated it. Some active inhibition of a state to permit transition to a next state is undoubtedly a necessary component of any model of human thinking. In this case, once position 1 has been inhibited, position 2 will dominate all of the other serial position ideas and be maximally activated, and so forth.

Restricting the sequential links to one or more special sets of position ideas raises the issue of whether these sequential links among position ideas need to be learned. Clearly, they do not. The main advantage of the position model is that the sequential links can be innate, whether excitatory as in the first model or inhibitory as in the second model. In models that employ sequential associations among the items of each sequence to order the items in the sequence, it is necessary that these sequential associations be learned, that is, associations with modifiable strength.

Of course, while the lateral associations among the position ideas can be innate in the aforementioned position model, the association to each item must be learned, which completely negates any savings in number of learnable associations to encode a sequence compared to a model using lateral associations between adjacent items in the sequence.

However, there are other versions of the position model than the one I already described that make even the association from the position idea

to each item an innate, rather than a learnable, association. The trick is to assume that each position idea is innately associated to a large set of representatives that can be used to represent every idea that ever appears in that position in a human's experience. This appears to be MacKay's (1987) model. When the sequences are syntactically structured sentences or phrases of words, it seems all right to assume that all noun ideas are innately associated to the noun position idea, all color adjectives are innately associated to the color adjective position idea, and so forth, though even here some might object to having separate representations for a word in each of the syntactic classes in which it can appear. However, this problem of multiple representations gets uglier and more ad hoc when, in order to learn arbitrary sequences of 5 words (the immediate memory span for words), we need to assume that every word has a separate representation for position 1, position 2, position 3, position 4, and position 5. And, of course, humans are capable of learning much longer random sequences of words.

But when humans learn long sequences, they typically employ grouping and other mnemonic devices that piggyback the new sequence on already learned sequences. I once studied grouping in short-term memory and concluded that in this context humans use just three position concepts—'beginning, middle, end'—but they can employ these in a cross-classification scheme consisting of beginning, middle, and end group with beginning, middle, and end position within a group (e.g., Wickelgren, 1964, 1967). It is important to note that the substitution errors tended to be from either the same group or the same position within a group, which indicates that the coding was more like a cross-classification scheme than a simple hierarchy.

Without a more detailed specification of the position model, it is not possible to conclude just how many different representations there would need to be for each idea to be able to distinctively encode its position in all of the different positions of all of the different kinds of sequences in which it could appear. At this point I see no reason to believe that the number of positional replications of each idea would be excessive.

My main reservation about position coding is rooted in the fact that I do not feel intuitively that humans use position information as much as prior item information in some of the sequences to which MacKay (1987) applied the position model.

Consider the issue of how we code the order of the phonemes in a word such as "strand," as discussed by MacKay (1987, p. 51). Assume that the subject has been shown the word strand and must now answer a yes–no question concerning a single constituent's relative position in the word. One could do an experiment to determine which of the following types of questions yields faster and more accurate retrieval (ideally

by determining the speed–accuracy tradeoff functions for each): (a) Is the final nasal of the vowel group *n*? (b) Is the sound that follows /tra/ *n*?

There are lots of problems to designing this experiment so as to treat the issue of whether order is cued by position or prior items. For instance, I chose three prior phonemes, because that is how many prior phonemes are needed to cue a single context-sensitive allophone in the model to be described next, but it would be interesting to know how subjects would perform with a greater or lesser number of prior phonemes. Although I chose the same words MacKay (1987) used to describe his position coding model, I feel sure that MacKay would deny the validity of the first question as a test of his model on the very reasonable grounds that the position ideas of 'final nasal' and 'vowel group' are not going to be cued by the phrases "final nasal" and "vowel group," except possibly after some linguistic training, and not necessarily even then. There are lots of other problems. Yet I feel that pronouncing a few prior phonemes is a natural cue for the next phoneme in a word, and no position cue is equally natural.

In visual iconic memory, however, there is pretty strong evidence favoring position coding over prior item coding for serial order (Wickelgren & Whitman, 1970), and intuitively I feel that something like MacKay's syntactic categories plays an important role in ordering ideas at the level of concepts and words.

What is a plausible position coding model for the order of the small positive integer names that children learn when they learn rote counting or the order of the letter names in the alphabet that children probably learn in much the same way? The most plausible model is probably some multilevel grouping scheme, but I doubt that the groups are innate and invariant across all children. For numbers, no particular grouping seems terribly compelling. For the letters, *a b c* seems like a group, until we recall that in the alphabet song, the first group is *abcd.*

In principle, it is quite attractive to imagine that there is an innate representation of the basic number ideas from 1 to 7 or is it 1 to 3 or is it 1 to 9 or is it 1 2 3 many or . . . ? The evidence for three basic position cues provides some support for the 1 2 3 model, but counting gets to be pretty complex if you cannot have more than three different position ideas at each level of grouping. Here again, as in most other levels of coding below semantic memory, it seems intuitively wrong to think that humans are encoding order by means of some set of position ideas or banks of position-specific representations, except in the case of visual iconic memory, which is a very short-lasting nonassociative memory.

Context-Sensitive Coding. Although a single prior item often provides inadequate information to determine its successor in an unordered set, a sequence of two prior items is adequate in almost all cases. A plausible model to provide this information is provided by context-sensitive coding, which codes an idea context-sensitively with a different representation depending on prior and succeeding context (Wickelgren, 1969, 1972, 1979a). Although one might use more context, it appears to be sufficient to code a sequence of items by overlapping triples. Consider the sequence of five phonemes, /s t r u k/, composing the word "struck." In overlapping triple code, this is ${}_{\#}s_{t}$, ${}_{s}t_{r}$, ${}_{t}r_{u}$, ${}_{r}u_{k}$, ${}_{u}k_{\#}$, where # means juncture, the boundary between two words. It is important to note that semantically, each phoneme triple means the particular allophone of the central phoneme that occurs in this left and right phonemic context. It does not mean a chunk of three phonemes. For this reason, I prefer to call these phoneme triples, *context-sensitive allophones,* or *c-s allophones,* to put the emphasis on the central element.

If accent is considered a distinctive feature of vowels, then to my knowledge, there are no ambiguities in ordering the unordered set of c-s allophones for any word in English. In any word, there is only one initial c-s allophone, that is an allophone of the type ${}_{\#}x_{y}$, and only one terminal allophone, that is of the type ${}_{y}x_{\#}$. Note that the context-sensitive coding model assumes a separate set of phoneme representatives for initial and terminal positions, as a simple position model might, but there must be more such representatives in the context-sensitive coding model, because each initial allophone is also conditioned by the second phoneme and each terminal allophone is conditioned by the next-to-last phoneme. The same internal allophone may appear in many different positions in different words, but it is likely that c-s coding requires considerably more phoneme representatives than any position coding model, which is a strike against it on grounds of representational efficiency, though the number of neurons required for c-s coding of phonemes is so tiny in comparison to the number of neurons in the nervous system that the force of this argument is much attenuated.

I believe it is useful vis-à-vis serial order to make a sharp distinction between creative behavior, such as the activation of sentences in thinking, and noncreative behavior, such as the activation of the phonetic constituents of words (Wickelgren, 1969). Lashley (1951) and MacKay (1987) do not. I think that c-s coding is simpler than position coding in learning and retrieval, and that it evolved earlier. As MacKay (1987) illustrated, position coding is a powerful device that can order words in novel utterances. Parrots can mimic words (noncreative serially ordered behavior), but they cannot create novel sentences. Perhaps it is precisely because

they lack position coding capacity that birds' linguistic capacities fall short of ours.

I do not favor the version of position coding in which a set of units is innately dedicated to each syntactic category (position). Rather, I favor the version in which each concept or word has a syntactic category idea as a constituent. Then syntactic programs, coded via vertical and lateral associations among the syntactic categories, control the activation of concepts via up associations from syntactic category ideas to the concepts.

I believe these syntactic programs themselves rely on context-sensitive coding of syntactic categories, that is, that we use overlapping triples of syntactic categories such as $_{\#}\text{article}_{\text{size-adjective}}$, $_{\text{article}}\text{size-adjective}_{\text{noun}}$, $_{\text{size-adjective}}\text{noun}_{\#}$ to control the order of output of the words in a phrase such as "the large ball." I have a hunch that the evolutionary advance that permitted syntactically structured thinking was primarily in using order information stored in one system to control the order of activation in another system. Doubtless, MacKay and I will continue to agree to disagree.

Counting is not creative behavior, and, hence, I assume that our knowledge of the order of the small positive integers is context-sensitively coded in terms of overlapping triples of integers. Thus, the chunk 'count' has down associations to $_{0}1_{2}$, $_{1}2_{3}$, and so forth. The chunk for the ratio of the circumference of a circle to its diameter, pi, has down associations to $_{\#}3.$, $_{3}1$, $.1_{4}$, $_{1}4_{1}$, and so forth. In context-sensitive coding the strong sequential associations are assumed to be between number triples, for example, $_{2}3_{4}$ has a strong sequential association to $_{3}4_{5}$. Although the digit sequences in some possible numbers would require richer context-sensitive coding than overlapping triples, such digit sequences are rare.

Jordan's Plan-State Model. Jordan (1986) proposed a neural net to code sequences that used three layers of neurons (input, hidden, and output), two classes of input neurons (plan neurons and state neurons), recurrent links among the state neurons, and backward (efference copy) links from the output neurons to the state neurons, in addition to the usual forward links from input neurons to hidden neurons and from hidden neurons to output neurons. Plan neurons encode the higher-level idea of the entire sequence, that is, an entire word, the alphabet, and so forth. The output neurons represent the lower-level constituents of the plan, that is, letters. State neurons represent serial order ideas, such as positions. Hidden neurons represent compounds of plans and states designed to permit nonlinear functions linking plans and states to constituents. Nonlinear functions are necessary to overcome the sort of associative interference, as pointed out by Lashley (1951).

A task Jordan gave to his network was to learn each of the six possible

sequences of three letters, *A, B, C,* that have no repeats: *ABC, ACB, BAC, BCA, CAB, CBA.*

One simple way to make such a network generate these sequences is to have one plan neuron for each of the six different sequences, one state neuron for each of the three positions in the sequence (first, second, third), one output neuron for each of the three letters, and one hidden neuron for each of the nine compound ideas: *A* first, *B* first, *C* first, *A* second, *B* second, *C* second, *A* third, *B* third, *C* third. It is necessary to assume an initial state for the state neurons, so we assume it to be that in which the first state neuron is active and no other state neurons. We assume that if the plan is to generate sequence *ABC,* then the *ABC* plan neuron is active and no others. Let the *ABC* plan neuron have a + 1 link to the three hidden neurons: *A* first, *B* second, *C* third, and zero or negative links to the other six hidden neurons. Let the first state neuron have a + 1 link to the three hidden neurons: *A* first, *B* first, *C* first, and zero or negative links to the other six hidden neurons. If the threshold for activating hidden neurons is 1.5, then only the *A* first hidden neuron will be activated to this input. Of course, we let the *A* first neuron have a + 1 link to the *A* output neuron and zero or negative links to the other two output neurons, assume a threshold for all output neurons of .5, and then only the *A* neuron will be active as the first output of the *ABC* plan. Now if we assume that state one has a + 1 link only to state 2, then state 2 will be active next, which, along with the maintained activation of the *ABC* plan neuron, activates only the *B* second hidden neuron, which activates the *B* output neuron, and so forth.

This simple model uses specific node coding, whereas Jordan uses distributed coding for inputs and outputs, and it makes no use of the backward links from outputs to state neurons, which Jordan admits are unnecessary to account for the basic ability to generate sequences. However, because distributed coding is simply a generalization of specific node coding, the previous simple, but high-interference, example makes it clear that Jordan's neural network has sufficient capability to overcome Lashley's serial order interference problem.

Jordan assumed the net learns the required link strengths by supervised learning, namely, the error (back) propagation learning algorithm of Rumelhart, Hinton, and Williams (1986) modified to handle recurrent nets. Personally, I think error propagation is too complex to be likely for real neural nets, but who can be sure of such matters? Anyhow, it enriches understanding to have contrasting theoretical alternatives, such as supervised versus unsupervised learning.

The specific example I described makes Jordan's model sound like the position coding model, but this is misleading. To be sure, Jordan's model can incorporate position coding ideas into the state and hidden neurons,

but it can also incorporate prior context in a somewhat different way from context-sensitive coding via the backward efference copy links from output neurons to state neurons. The cue for the initial item of a sequence is the plan idea alone. The cue for the second item is the plan idea plus the initial item. The cue for the third item is the plan idea and the first two items of the sequence, and so on. This is called *efference copy* because some trace of the items remains active after they are output, until activation of some 'end' or 'juncture' idea terminates all activation associated with this sequence plan.

Jordan also used distributed coding instead of specific node coding. This can economize on the number of neurons required to code a set of ideas such as a set of sequences, positions, constituents, and so forth, but economizing on neurons in this way often leads to serious interference problems when learning many associations with the same net. I am quite enthusiastic about distributed coding, but I think that to avoid serious associative interference problems, it is necessary to implement chunking in distributed coding models. Hidden units in three-layer nets can perform many of the functions of chunking. Whether or not this is the optimal way to implement these functions remains to be seen.

COUNTING LONG-TERM REPETITIONS

A possible semantic model for counting long-term repetitions is as follows: When idea A is activated a second time, the familiarity idea is activated. If there are any associated memories for the number of prior activations of idea A, these are activated. Assume that the highest such number idea inhibits the others. Imagine that it is recalled that A has been activated on n prior occasions ($n > 1$), then what is activated is the digit triple ${}_{n-1}n_{n+1}$. Assume that simultaneous activation of the familiarity idea and the ${}_{n-1}n_{n+1}$ idea have previously been learned as the necessary and sufficient conditions for activation of ${}_{n}n+1_{n+2}$, and so the count is incremented by one. Then A is chunked with the new count idea ${}_{n}n+1_{n+2}$, so that the next time A is activated, it will activate the new chunk and its higher count ${}_{n}n+1_{n+2}$.

When $n = 1$, the foregoing model works also, but I doubt that people always store a memory that an idea has been activated once, the first time it is activated. So one just adds to the model the proviso that if A is active, the familiarity idea is active, and no number idea is active, then it is ${}_{1}2_{3}$ that is activated and chunked with A. The next time A is activated ${}_{1}2_{3}$ will be activated along with the familiarity idea, which causes ${}_{2}3_{4}$ to be activated and chunked with A, and so forth.

This is no more than a brief sketch of a model for counting long-

term repetitions. Many problems and surprises would occur in any attempt to implement this in a functioning neural net model or a more mathematical semantic model, but some model more or less along these lines could probably be developed.

It occurs to me now that we frequently know that we have experienced an idea on many prior occasions without recalling any specific count. We do not confuse such cases with cases where an item was experienced on one prior occasion and no explicit count of one occurrence was stored. Of course, there are ways to distinguish these cases, but until such matters are handled, the model's development is incomplete.

Nevertheless, it seems likely to me that any model of sequential counting of long-term repetitions in humans will involve submechanisms for chunking, familiarity recognition, and serial order.

REFERENCES

Estes, W. K. (1972). An associative basis for coding and organization in memory. In A. W. Melton & E. Martin (Eds.), *Coding processes in human memory* (pp. 161–190). Washington, DC: Winston.

Gelman, R., & Gallistel, C. R. (1978). *The child's understanding of number.* Cambridge, MA: Harvard University Press.

Grossberg, S. (1978). A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. *Progress in Theoretical Biology, 5,* 232–302.

Jordan, M. I. (1986). Serial order: A parallel distributed processing approach (Tech. Rep. No. ICS-8604). La Jolla, CA: University of California, Institute for Cognitive Science.

Lashley, K. S. (1951). The problem of serial order in behavior. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior: The Hixon symposium* (pp. 122–130). New York: Wiley.

MacKay, D. G. (1987). *The organization of perception and action.* New York: Springer-Verlag.

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.

Remington, R. (1977). Processing of phonemes in speech: A speed-accuracy study. *Journal of the Acoustical Society of America, 62,* 1279–1290.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 318–362). Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing* (Vol. 1). Cambridge, MA: MIT Press.

Wickelgren, W. A. (1964). Size of rehearsal group and short-term memory. *Journal of Experimental Psychology, 68,* 413–419.

Wickelgren, W. A. (1967). Rehearsal grouping and hierarchical organization of serial position cues in short-term memory. *Quarterly Journal of Experimental Psychology, 19,* 97–102.

Wickelgren, W. A. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behavior. *Psychological Review, 76,* 1–15.

Wickelgren, W. A. (1972). Context-sensitive coding and serial vs. parallel processing in speech. In J. H. Gilbert (Ed.), *Speech and cortical functioning* (pp. 237–262). New York: Academic Press.

Wickelgren, W. A. (1977). *Learning and memory.* Englewood Cliffs, NJ: Prentice-Hall.
Wickelgren, W. A. (1979a). *Cognitive psychology.* Englewood Cliffs, NJ: Prentice-Hall.
Wickelgren, W. A. (1979b). Chunking and consolidation: A theoretical synthesis of semantic networks, configuring in conditioning, S-R versus cognitive learning, normal forgetting, the amnesic syndrome, and the hippocampal arousal system. *Psychological Review, 86,* 44–60.
Wickelgren, W. A., & Whitman, P. T. (1970). Visual very-short-term memory is nonassociative. *Journal of Experimental Psychology, 84,* 277–281.
Young, R. K. (1968). Serial learning. In T. R. Dixon & D. L. Horton (Eds.), *Verbal behavior and general behavior theory* (pp. 122–148). Englewood Cliffs, NJ: Prentice-Hall.