*Nonlinear Dynamics, Psychology, and Life Sciences, Vol. 4, No. 2, 2000*

# The Complexity of Artificial Grammars

**Erik M. Bollt**[1] **and Michael A. Jones**[2,3]

*In experimental psychology, artificial grammars, generated by directed graphs, are used to test the ability of subjects to implicitly learn the structure of complex rules. We introduce the necessary notation and mathematics to view an artificial grammar as the sequence space of a dynamical system. The complexity of the artificial grammar is equated with the topological entropy of the dynamical system and is computed by finding the largest eigenvalue of an associated transition matrix. We develop the necessary mathematics and include relevant examples (one from the implicit learning literature) to show that topological entropy is easy to compute, well defined, and intuitive and, thereby, provides a quantitative measure of complexity that can be used to compare data across different implicit learning experiments.*

**KEY WORDS:** artificial grammars; implicit learning; symbolic dynamics; complexity; topological entropy.

## INTRODUCTION

A popular paradigm for testing rule-based learning in the psychology literature uses finite state grammars or ''artificial grammars'' (e.g., Reber, 1989). The sequences of symbols (letters) of an artificial grammar are generated by a grammatical rule, summarized by the allowed transitions of a directed graph. A typical experiment involves showing subjects words, or strings of letters, or even simply sequences of letters in turn, of a list of grammatically generated samples, in various formats and introductory

[1]United States Naval Academy, Mathematics Department, Annapolis, MD 21402-5002.
[2]Montclair State University, Department of Mathematical Sciences, Upper Montclair, NJ 07043.
[3]Correspondence should be directed to Michael A. Jones, Department of Mathematical Sciences, Montclair State Univesity, Upper Montclair, NJ 07043; e-mail: jonesma@pegasus.montclair.edu.



1090-0578/00/0400-0153$18.00/0 

instructions (e.g., subjects may or may not be told that the sequences are rule based). A main thesis of this paper is that an important and natural comparison between experiments of different researchers' studies should involve the inherent complexity of the grammatical rules by which letter sequences are generated.

It seems reasonable that a small directed graph (an artificial grammar with few letters and few nodes) with many grammatical restrictions (few arrows) should be easier to learn than a grammar with, say, a thousand letters and thousands of transition rules. A grammar with thousands of letters in which any letter may follow any other letter can really be said to have no rules at all: it would actually be grammatically simple and statistically uniform. However, a grammar with just a dozen letters, with many restricted transitions is more difficult to learn. For example, see Fig. 4, the grammar used in the experiments performed by Reber and Allen (1978). A mathematical measure of ''grammatical complexity'' would be useful to the artificial grammar learning community, not only to standardize and to evaluate results from different studies but to aid in the design and evaluation of experiments in a single study.

In this paper, we introduce the necessary mathematics of symbolic dynamics to view artificial grammars as dynamical systems under the Bernoulli shift map and directly apply topological entropy as a mathematical measure of the grammatical complexity. There is a classical and deeply rooted link between chaos theory, symbolic dynamics, and information theory which examines the growth rate of distinct states of the dynamical system (Adler, Konheim, & McAndrew 1965). In Shannon's information theory, a sequence of events conveys information only if the events are not fully predictable; therefore, the topological entropy may be considered as a quantitative measurement of the information generating capacity of the chaotic dynamical system (Blahut, 1988; Shannon, 1948). We define the complexity of the artificial grammar to be the growth rate of the set of possible sequences generated by the directed graph. Artificial grammars, and thereby the results of implicit learning experiments under different artificial grammars, can be compared by their complexity.

We introduce the necessary preliminaries from experimental psychology and dynamical systems, including symbolic dynamics and transition matrices. Directed graphs can be represented by transition matrices, but only a matrix of large enough dimension separates the useful information; this idea is explained by a new lifting technique, which lifts the matrix to a Markov representation.

Further, we define topological entropy as a measure of the complexity of artificial grammars. Not only is this measure well-defined and easy to

**Table 1.** Ten Observed Sequences

| Sequence | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | . . . |
|---|---|---|---|---|---|---|---|---|---|---|---|
| First | *d* | *c* | *a* | *d* | *c* | *a* | *b* | *d* | *c* | *a* | . . . |
| Second | *c* | *a* | *d* | *a* | *b* | *d* | *a* | *d* | *a* | *b* | . . . |
| Third | *c* | *a* | *d* | *c* | *a* | *d* | *c* | *a* | *b* | *c* | . . . |
| Fourth | *a* | *d* | *c* | *a* | *d* | *a* | *b* | *c* | *a* | *b* | . . . |
| Fifth | *d* | *d* | *a* | *b* | *c* | *a* | *b* | *d* | *c* | *a* | . . . |
| Sixth | *b* | *d* | *a* | *d* | *c* | *a* | *d* | *c* | *a* | *d* | . . . |
| Seventh | *b* | *d* | *c* | *a* | *b* | *d* | *a* | *b* | *c* | *a* | . . . |
| Eight | *d* | *a* | *d* | *c* | *a* | *d* | *a* | *d* | *c* | *a* | . . . |
| Ninth | *c* | *a* | *b* | *c* | *a* | *b* | *d* | *a* | *b* | *c* | . . . |
| Tenth | *c* | *a* | *d* | *a* | *b* | *c* | *a* | *b* | *c* | *a* | . . . |

calculate, but natural. Finally, we compute the topological entropy of the artificial grammars from the examples used to motivate the notation, as well as from Reber and Allen (1978). As part of our concluding remarks, we discuss the previous attempts to grapple with the complexity of artificial grammars in the experimental psychology literature. Further, we suggest natural areas for research which would incorporate the quantitative measure of the complexity of artificial grammars.

## NOTATION AND SYMBOLIC DYNAMICS

To introduce the notation and terminology, we consider the following example. Suppose that the sequences of symbols in Table 1 are generated by the same rule. The sequences are read across in the table. For example, the first sequence is "*dcadcabdca . . .*"

Is it possible to determine if the three sequences in Table 2 were generated by the same rule as the sequences from Table 1?

Rather than explicitly articulate the properties of the rule, the ability to determine the rule is measured, implicitly, by how well we recognize which candidate sequences were generated by the same rule. It so happens that the first and third candidate sequences were generated by the same rule as the first ten observed sequences. Without fully knowing the rule

**Table 2.** Three Candidate Sequences

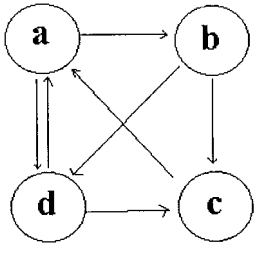
| Candidates | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | . . . |
|---|---|---|---|---|---|---|---|---|---|---|---|
| First | *b* | *c* | *a* | *d* | *a* | *b* | *d* | *c* | *a* | *d* | . . . |
| Second | *d* | *b* | *c* | *d* | *c* | *b* | *a* | *b* | *c* | *b* | . . . |
| Third | *c* | *a* | *d* | *c* | *a* | *b* | *d* | *c* | *a* | *b* | . . . |

which generates the sequences, we may find the second candidate sequence suspicious and notice that in the original sequences, the letter *c* was always followed by the letter *a.* However, in the second candidate sequence, the letter *c* is followed, at different times, by *b* and *d.*

This is an example of an implicit learning experiment. Actual experiments vary as to the conditions of which the subjects receive the information. However, the rule is typically given by a directed graph, which generates an artificial grammar. In the following development, we will view the directed graph as a dynamical system and its artificial grammar by the subshift, or set of all permissible sequences of letters generated by the directed graph. For ease of introduction, we will examine the directed graph (Fig. 1) which generated the sequence data for our mock experiment and introduce the terminology through this example.

The nodes of the directed graph are represented by a finite set of letters called an alphabet, denoted by $A$. In our example, $A = \{a, b, c, d\}$. Define the symbol space on $A$ or, equivalently, the full shift on $A$, as the set of *all* possible infinite letter words, denoted as $\Sigma_A = \{\sigma_0.\sigma_1\sigma_2 \ldots | \sigma_i \in A\}$. In this setting, a point in symbol space is a possible sequence where any element of $A$ may follow any other element of $A$ and the subscript indicates the ordering of the letters. The decimal place after the first letter, $\sigma_0$, is used as a frame of reference.

The directed graph defines an artificial grammar by generating sequences of letters by moving along the arcs of the graph. An artificial grammar reduces the possible sequences of letters that can occur. This




$$M = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

**Fig. 1.** A directed graph on the alphabet $A = \{a, b, c, d\}$ and its associated transition matrix $M$.

restriction on possible sequences defines a subspace of $\Sigma_A$. All arcs exiting from a node are equally probable; therefore, in Fig. 1, a *b* is followed by a *c,* or a *d,* both with probability $\frac{1}{2}$. Also, in this example, all nodes are equally probable to be the first node of the sequence. The sample data in Table 1 and the first and third sequences in Table 2 were created by generating random numbers to follow paths through the directed graph.[4]

**Definition 1.** An *artificial grammar* is the set of all letter sequences generated by a directed graph.

This definition allows for both finite and infinite length words. The following related concept from dynamical systems is equivalent to reading an infinite sequence of letters from left to right. Since the topological entropy measure of complexity is in terms of an asymptotic growth rate, we restrict our attention to infinite sequences. Certain implicit learning experiments use sequences of only finite length. In a later section, we discuss how these artificial grammars can be reasonably considered as an infinite number of concatinated finite length words.

**Definition 2.** The *Bernoulli shift* on $\Sigma_A$ is a map $s$: $\Sigma_A \rightarrow \Sigma_A$ defined by

$$s(\sigma) = s(\sigma_0.\sigma_1\sigma_2\sigma_3 \ldots) = \sigma_1.\sigma_2\sigma_3. \ldots \quad (1)$$

The Bernoulli shift maps an infinite sequence $\sigma = \sigma_0.\sigma_1\sigma_2\sigma_3 \ldots$ to another infinite sequence $\sigma' = \sigma_1.\sigma_2\sigma_3. \ldots$. Alternatively, we can view the Bernoulli shift as following an arc of the directed graph from a node $\sigma_0$ to the node $\sigma_1$. An infinite word, or point in $\Sigma_A$, describes an infinite itinerary of nodes visited by a particular path through the graph.

**Definition 3.** A subshift $\Sigma$ is a subspace of $\Sigma_A$ which is invariant under the Bernoulli shift, i.e., if $\sigma \in \Sigma$ then $s(\sigma) \in \Sigma$.

Hence, the Bernoulli shift map is a dynamical system on symbol space which maps sequences to sequences, by the operation of shifting the decimal to the right and eliminating the leftmost symbol, as described by Eq. 1; this is known as symbolic dynamics. Given our restriction to sequences of infinite length, or equivalently, finite length words that are concatinated into infinite length sequences, the following proposition is immediate.

**Proposition 1.** *An artificial grammar is a subshift.*

Notice that in our mock experiment, the sequence '*abcabcabc* . . .' is in the subshift $\Sigma$. The sequence '*cbacbacba* . . .' is not in the subshift since there is not an arc from *c* to *b* in the directed graph of Figure 1; however, this sequence is part of the full shift, $\Sigma_A$, which allows transitions to all elements of $A$ from any element of $A$.

[4]It was statistically unlikely, specifically with probability $(\frac{3}{4})^{12} \approx 3.2\%$, that the letter *a* did not start one of the 12 sequences generated by the directed graph in Fig. 1.

Since every artificial grammar can be represented by a subshift where the subshift is the set of all possible sequences generated by the directed graph, then the topological entropy measure of complexity for the subshift also measures the complexity of the artificial grammar. We will show how to compute the topological entropy of an artificial grammar directly from an associated transition matrix.

To define the associated transition matrix for the artificial grammar of our mock experiment, we need to enumerate the elements, or states, of $A$. Let $a$ be called state 1 or $a = 1$ and $b = 2$, $c = 3$, and $d = 4$. Define a transition matrix, $M$, to have entries $m_{i,j}$ defined to be 0 or 1: $m_{i,j} = 0$ if state $i$ cannot immediately follow state $j$ under the directed graph, and $m_{i,j} = 1$ if state $i$ can immediately follow state $j$ under the directed graph. For the directed graph in Fig. 1, identifying the entries of a transition matrix under this rule yields the matrix $M$, also in Fig. 1. This matrix defines the artificial grammar that is generated by the directed graph. We can find a transition matrix for any artificial grammar generated by a directed graph (with a finite number of nodes and arcs). To make this relationship precise, we need the lifting technique discussed in the next section.

## THE LIFT

Consider the directed graph in Fig. 2, which merely adds an additional node and two additional arcs to the directed graph from Fig. 1. Realize that the alphabet is still $A = \{a, b, c, d\}$. If we construct the $4 \times 4$ transition matrix for this directed graph (keeping the same enumeration of the states as in Section 2), then the first column would contain all 1's except for the first row. The implications are that $a$ can be followed by $b$, $c$, or $d$. A quick inspection of Fig. 2 agrees with this statement. However, notice that the pair $ba$ can be followed by $c$, but not by $b$ or $d$. So, $a$ cannot always be followed by $b$, $c$, or $d$, since the next node depends not only on the current

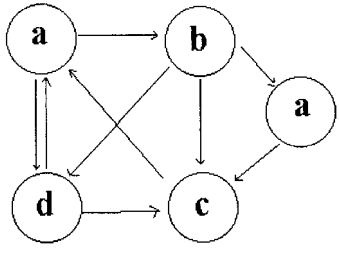



**Fig. 2.** A directed graph on the alphabet $A = \{a, b, c, d\}$ without the *memoryless property.*

node, but its predecessor, as well. By focusing on blocks of letters (the psychology literature refers to these as "chunks," see Higham, 1997 for a partial review on this literature), we see that the transition matrix does *not* represent the directed graph.

Artificial grammars which rely on $k$ previous letters also define a subshift. However, the subshift of such a multiple-step artificial grammar on $m$ letters cannot be described by an $m \times m$ transition matrix. To be able to compute the complexity of the artificial grammar, it is necessary to find a transition matrix which does describe the artificial grammar. To adequately describe this grammar, we develop a *lift* of the symbol space $\Sigma_A$ to a larger ("higher dimensional") symbol space in which the $k$ states of information generate an $m^k \times m^k$ transition matrix. This idea is developed by the following definitions.

**Definition 4.** The *memoryless property* holds for transitions between a collection of states if only the current state is sufficient to decide if a following transition is permitted.

The directed graph in Fig. 2 does *not* satisfy this property. Knowledge that the current node is $a$ does not imply which letters can follow $a$, since it is not clear which $a$ node is specified. If every letter of the alphabet occurs at a single node of the directed graph, then the directed graph has the memoryless property. If we define a "state" in terms of two letters, and so there are 16 possible such two-letter states, then all permitted transitions are well defined by simply examining only the current two-bit "state." For our example, the states are the elements in $A \times A$. Given the current letter and its predecessor is enough information to determine the node in the directed graph. In general, to achieve the memoryless property, we are forced to choose states with enough previous letters to accomodate a multi-step rule. Defining the appropriate state definitions to achieve the memoryless property motivates the definition of a Markov representation.

**Definition 5.** A collection of states is a *Markov representation* if transitions between the states have the memoryless property.

Any finite rule artificial grammar with a finite alphabet has a Markov representation, meaning that there is some finite $k$-letter length so that the set of all $k$-letter blocks (considered as $m^k$ states) is a Markov representation. Definition 5 does not indicate that there is a single Markov representation. In practice, it is best to choose the minimal $k$ which achieves a Markov representation, since the number of nodes, $m^k$, grows exponentially in $k$. However, for each $k$ greater than or equal to this minimum value, there is a unique Markov representation. After a formal explanation of this process, we will return to the directed graph in Fig. 2.

Artificial grammars which incorporate $k$ previous states are repre-

sented by a Bernoulli shift on a higher dimensional symbol space. Again, the subshift defines all sequences of letters. To define a $k$-state memory dependent transition rule of the $m$ symbols in $A$ requires a transition rule on the $m^k$ symbols of $A^k$, where $A^k = \underbrace{A \times A \times A \cdots A \times A}_{k \text{ times}}$. Denote each $k$-tuple of symbols from $A$ as a symbol in $A^k$.

The Bernoulli shift applied to a point $\sigma \in \Sigma_A$, where $k$ states are recorded, may be written as

$$s(\sigma) = s(\underbrace{\sigma_0\sigma_1\sigma_2 \cdots \sigma_{k-1}}_{k \text{ symbols}}.\sigma_{k\cdots}) = \underbrace{\sigma_1\sigma_2\sigma_3 \cdots \sigma_{k-1}\sigma_k}_{k \text{ symbols}}.\sigma_{k+1\cdots}$$

Writing the decimal immediately after the $k^{th}$ symbol emphasizes the most recent node, as $k$ states are necessary to distinguish which node in the directed graph.

To generate the transition matrix, we have developed a technique to lift the symbol space to a higher dimension. The lift defines an associated Markov representation for a grammar which might otherwise be ambiguous. Consider the following geometric analogy for the lift. The figure eight in a plane may be a projection of a curve in three space. While the three–dimensional curve may not intersect itself, the planar projection does. Only when viewed in the higher dimension ($3D$), can this curve without an intersection be uniquely distinguished from a curve with an intersection.

Define the symbol space with labels from $A^k$ as $\Sigma_{A^k} = \{\sigma'_0.\sigma'_1\sigma'_2 \ldots | \sigma'_i \in A^k\}$ or all infinite sequences of symbols from $A^k$.

**Definition 6.** The $k$-step lift is a map $l$: $\Sigma_A \to \Sigma_{A^k}$ defined by collecting groups of $k$ symbols from $A$ in the following "overlapping" manner where $l(\sigma) = \sigma'$. Let

$$\sigma = \sigma_0.\sigma_1\sigma_2 \ldots \sigma_{k-1}\sigma_k\sigma_{k+1} \ldots \in \Sigma_A.$$

The image of $\sigma$ under $l$ is $\sigma' = \sigma'_0 \cdot \sigma'_1\sigma'_2 \ldots \in \Sigma_{A^k}$ where $\sigma'_0 = \sigma_0\sigma_1\sigma_2 \ldots \sigma_{k-1}$, $\sigma'_1 = \sigma_1\sigma_2\sigma_3 \ldots \sigma_k$, $\sigma'_2 = \sigma_2\sigma_3\sigma_4 \ldots \sigma_{k+1}$, etc.

The ambiguity raised by the two different transitions from the two nodes labeled $a$ in the directed graph in Fig. 2 can be lifted to a Markov representation by, in this case, considering $k = 2$ steps at a time. The nodes of the higher dimensional directed graph are in $A \times A$ and can be enumerated by the lexicographic ordering, e.g., $aa = 1$, $ab = 2$, $ac = 3$, $ad = 4$, $ba = 5$, . . . , and $dd = 16$. This uniquely defines nodes and their allowed transitions. So, in this case, considering all the possible transitions between the $m^k = 4^2 = 16$ states in the 2-bit Markov representation uniquely defines the grammar. The transitions between states can be found in Table 3. In other examples, a lift to larger $k$-bit groupings may be necessary, such

**Table 3.** States for a Markov Representation of the Directed Graph in Fig. 2. The Enumerated States Are Used to Create the Transition Matrix in Fig. 3

| | |
|---|---|
| *aa* = 1 | *ca* = 9 → *ab, ad* |
| *ab* = 2 → *ba, bc, bd* | *cb* = 10 |
| *ac* = 3 → ca | *cc* = 11 |
| *ad* = 4 → *da, dc* | *cd* = 12 |
| *ba* = 5 → *ac* | *da* = 13 → *ab, ad* |
| *bb* = 6 | *db* = 14 |
| *bc* = 7 → *ca* | *dc* = 15 → *ca* |
| *bd* = 8 → *da, dc* | *dd* = 16 |

as the artificial grammar used by Reber and Allen (1978) which is examined in Section 5.

While it is necessary to lift a $k$-state artificial grammar to form an $m^k \times m^k$ transition matrix, there is no danger in lifting to an even higher dimensional matrix. There is a unique $m^j \times m^j$ transition matrix $M$ for $j \geq k$. Over-lifting the subshift is analogous to representing a $2D$ plane in a higher dimensional space. The plane is still two dimensional, even though it is embedded in a higher dimensional space.

The shift map on $\Sigma_{A^k}$ cannot correspond to all letter sequences on the $m^k$ states of $A^k$ due to the use of labels in $A^k$ to include the history of the preceding letters. History cannot be chosen independently at each term, since the first $k-1$ terms are determined from the previous state. Therefore, transitions between each of the $m^k$ labels in $A^k$ can have at most only $m$ outcomes each. For our example from Fig. 2, the 2-bit state *ba* can only have an arc between other 2-bit states that have *a* as its left entry, namely *aa, ab, ac,* or *ad.* However, an examination of the directed graph indicates that *ba* is only followed by *ac.* The associated transition matrix appears in Fig. 3. This implies an important restriction to the size of a lifted full shift.

**Proposition 2.** *The full shift $\Sigma_A$ lifts to a subshift of $\Sigma_{A^k}$.*

*Proof.* Applying the shift map to points $\sigma \in \Sigma_A$ causes the $k^{th}$ previous bit to be forgotten, while only one new bit from the set $A$ is generated. The other $k-1$ bits are remembered, and therefore each of the $m^k$ nodes in $A^k$ can transition to only one of $m$ possible new nodes.

**Theorem 3.** *Given a k-state artificial grammar on A, described by a subshift of $\Sigma_A$, the lift $l$: $\Sigma_A \rightarrow \Sigma_{A^k}$ yields an $m^k \times m^k$ transition matrix M which uniquely defines the artificial grammar and generates the proper subshift $\Sigma_M \subset \Sigma_{A^k}$.*

*Proof.* All possible transitions between symbols of $A^k$ can be represented by an $m^k \times m^k$ transition matrix. Such a matrix generates the subshift of all paths through the directed graph.

$$M = \begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}$$

**Fig. 3.** The transition matrix for the artificial grammar shown in Fig. 2. To achieve the memoryless property, a lift from the $m = 4$ letters of the alphabet, to the $m^2 = 16$ 2-bit letters was required.

## COMPLEXITY AND TOPOLOGICAL ENTROPY

Cardinality is not a sufficient measure to distinguish between two obviously varied grammars, as typically, subshifts will be uncountably infinite. Topological entropy measures the asymptotic growth rate of the subshift. This is in contrast to a classification scheme which may classify complexity in terms of rules. In fact, as the number of rules increases (fewer arcs), the associated subshift, or set of possible sequences, tends to get smaller.

Originally introduced by Adler, *et al.,* (1965) in the context of information theory, topological entropy has become a familiar tool in the theory of dynamical systems as a measure of the complexity of chaos. Calculating topological entropy is particularly straightforward for the dynamics of a subshift (when the transition matrix is finite).

To define the topological entropy of a subshift $\Sigma_M \subset \Sigma_{A^k}$, where $M$ is the $m^k \times m^k$ transition matrix associated with a Markov representation, some additional definitions and concepts must be given. Define a word of length $n$ as a sequential combination of $n$ symbols from $A^k$; $(\underbrace{x_0, x_1, x_2, \ldots, x_{n-1}}_{n \text{ bits}})$ where $x_j \in A^k$. Thus, a point $\sigma \in \Sigma_{A^k}$ can be thought of as a word of infinite length, with a decimal added as the place holder, indicating the current position. The topological entropy $h$ of a subshift $\Sigma_M$ is the logarithm of the growth rate of the number of words of length $n$

found in the subshift, as $n$ goes to infinity. The following definitions appear in Robinson (1995).

**Definition 7.** The word count of a subshift $\Sigma_M$ is the number of subsequences of length $n$ which are contained in the subshift and is denoted as

$$w_n(\Sigma_M) = \#\{(x_0, \ldots, x_{n-1}) \mid x_i = \sigma_i \text{ for some } \sigma = \sigma_0.\sigma_1\sigma_2 \ldots \in \Sigma_M\}.$$

**Definition 8.** The topological entropy of the subshift $\Sigma_M$ is the scalar quantity $h(\Sigma_M)$, where

$$h(\Sigma_M) = \lim_{n\to\infty} \frac{\ln(w_n(\Sigma_M))}{n}. \tag{2}$$

For most artificial grammars, as $n$ increases, the number of possible words grows exponentially. Topological entropy measures this exponent. Recall that $A$ is the alphabet or set of symbols and $\Sigma_A$ is the full shift on $A$. Further, $M$ refers to a transition matrix and $\Sigma_M$ is the subshift defined by $M$.

**Theorem 4.** *The range of the entropy function for a subshift $\Sigma_M \subseteq \Sigma_A$ is given by the inequality*

$$0 \le h(\Sigma_M) \le h(\Sigma_A) = \ln m.$$

*Proof.* For a subshift $\Sigma_M$, $h(\Sigma_M)$ is bounded below by 0 since $w_n(\Sigma_M)$ is positive. Since $\Sigma_M \subseteq \Sigma_A$, it follows that $w_n(\Sigma_M) \le w_n(\Sigma_A)$ and $h(\Sigma_M) \le h(\Sigma_A)$.

For $\Sigma_A$, the possible words of length $n$ are all permutations of $m$ elements $n$ at a time, or $m^n$. Eq. 2 implies that

$$h(\Sigma_A) = \lim_{n\to\infty} \frac{\ln(w_n(\Sigma_A))}{n} = \lim_{n\to\infty} \frac{\ln m^n}{n} = \ln m.$$

The full shift on the $m^k$ symbols of $A^k$ is $h(\Sigma_{A^k}) = k \ln m$. If it were true that the full shift $\Sigma_A$ lifts to the full shift $\Sigma_{A^k}$, then the entropy would not be a well-defined measure of complexity. However, as stated in Proposition 2, the full shift $\Sigma_A$ lifts to a subshift, as required to preserve the entropy between different Markov representations of artificial grammars. The following theorem is a direct consequence of the lifting technique and how matrices of different sizes can represent the same sequence space.

**Theorem 5.** *If $M$ and $N$ are transition matrices of Markov representations of the same artificial grammar, then their topological entropies are equal, i.e., $h(\Sigma_M) = h(\Sigma_N)$.*

The next theorem gives a useful technique to calculate the entropy of an arbitrary $k$-step artificial grammar. The topological entropy of the subshift is equated with the spectral radius of the associated transition matrix of a Markov representation. Recall that the spectral radius of a finite matrix is the maximum of the modulus of the, possibly complex, eigenvalues of the

matrix. However, since the transition matrices generated from artificial grammars contain nonnegative real entries, Perron-Fröbenius Theory states that the spectral radius is largest nonnegative eigenvalue of the matrix (e.g., Gantmacher, 1974). Let $\rho(M)$ denote spectral radius of the matrix $M$. The proof of Theorem 6 appears in Robinson (1995).

**Theorem 6.** *Given a subshift $\Sigma_M$ generated by the transition matrix M,*

$$h(\Sigma_M) = \ln \rho(M). \tag{3}$$

## AN EXAMPLE FROM THE EXPERIMENTAL PSYCHOLOGY LITERATURE

It is more difficult to discover the rules of the grammar if the sequence space is large. Simply put, there are just more possibilities. In this section, we compute and compare the topological entropies for the artificial grammars generated by the directed graphs in Figs. 1 and 2. We also compute the topological entropy for a directed graph in Reber and Allen (1978) which requires a lift. Since Reber and Allen only allow for finite sequences of letters, we explain how infinite sequences can be generated by a simple concatination of the finite words.

Since we determined transition matrices for Markov representations of the artificial grammars generated by the directed graphs in Figs. 1 and 2, computing the topological entropies is equivalent to finding the largest positive eigenvalues of their matrices. The topological entropy for the artificial grammar from Fig. 1 is $\ln 1.7455 \approx 0.5570$, while the topological entropy from the augmented artificial grammar from Fig. 2 is $\ln 1.8084 \approx 0.5924$. Intrepreting these numbers: the original artificial grammar is less complex than the augmented artificial grammar. Although inspection of the two directed graphs yields a similar conclusion, the topological entropy is a quantifiable measure of the difference in complexity. Since the artificial grammar from Fig. 2 has a larger topological entropy, it contains more sequences of letters; therefore, it should be harder to learn this artificial grammar.

**Example 1.** The necessity for a lift in an artificial grammar in Reber and Allen (1978).

Consider the artificial grammar generated by the directed graph in Fig. 4. This is essentially the directed graph used in Reber and Allen (1978). In their experiment, the ''spaces'' did not appear on the directed graph. They allow only finite sequences which begin on the left and exit the directed graph on the right. Since we require infinite sequences, we concatinate finite words by separating finite sequences of letters generated by the directed

graph with blank spaces. Specifically, when a sequence of symbols exits the directed graph, we assume that a space is entered in the sequence. Further, the next symbol is an *M* or a *V,* as the directed graph is re-entered on the left. As before, all arcs are equally probable. In Table 3, we denote a "space" by the letter "S."

Further, notice that in Fig. 4, the letters are on the arcs, as opposed to the nodes. This problem is easily remedied; any directed graph with symbols on the arcs can be represented by a directed graph with symbols at the nodes. To consider an artificial grammar as the subshift of a dynamical system, we believe it is easier to develop the necessary mathematics with the symbols at the nodes.

Notice that the letters *M, R,* and *X* can follow an *R* under the directed graph. However, these letters cannot always follow an *R.* For example, no sequences of the form ". . . *XRX* . . ." can occur. In fact, lifting the directed graph to a directed graph on 25 ($5 \times 5$) symbols is also insufficient. This directed graph needs to be lifted to a directed graph with 125 ($5 \times 5 \times 5$) nodes. Although this seems like an enormous prospect, there are shortcuts which reduce the transition matrix to a more manageable size.

These shortcuts eliminate nodes which will never occur since these nodes generate rows and columns of all 0's in the transition matrix. We can therefore eliminate the column and row from the matrix. For example, the node *XRX* can be eliminated. This reduces the matrix to a $124 \times 124$ matrix. We have kept the essential states and transitions in Table 4. We

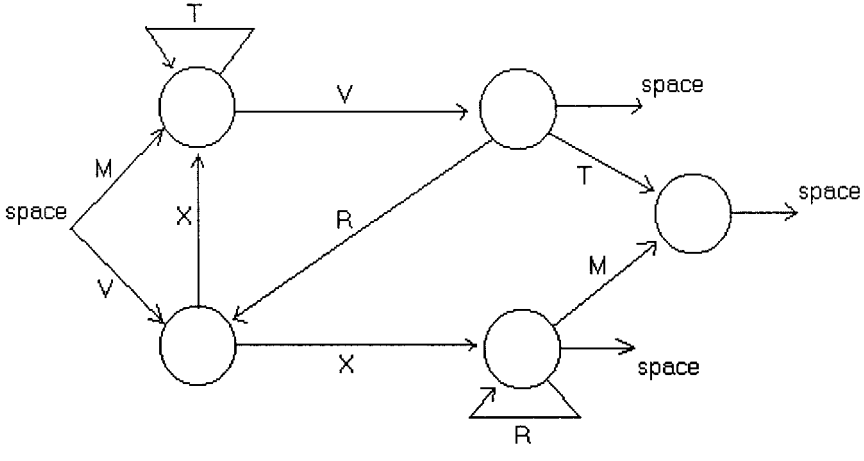



**Fig. 4.** The directed graph found in Reber and Allen 1978. See Table 4 for all of the allowed 3-bit transitions.

**Table 4.** The Artificial Grammar Found in Reber and Allen (1978) and Represented in Fig. 4

| | | | |
|---|---|---|---|
| $SMT \to MTV$ | $RMS \to MSM$ | $VSM \to SMT$ | $XTV \to TVR$ |
| $\to MTT$ | $\to MSV$ | $\to SMV$ | $\to TVS$ |
| $SMV \to MVS$ | $RRM \to RMS$ | $VSV \to SVX$ | $\to TVT$ |
| $\to MVT$ | $RRR \to RRM$ | $VRX \to RXR$ | $XTT \to TTT$ |
| $\to MVR$ | $\to RRR$ | $\to RXM$ | $\to TTV$ |
| $SVX \to VXT$ | $RXR \to XRM$ | $\to RXV$ | $XVR \to VRX$ |
| $\to VXV$ | $\to XRR$ | $\to RXT$ | $XVS \to VSM$ |
| $\to VXR$ | $\to XRS$ | $\to RXS$ | $\to VSV$ |
| $\to VXM$ | $RXM \to XMS$ | $VTS \to TSM$ | $XVT \to VTS$ |
| $MSM \to SMT$ | $RXV \to XVR$ | $\to TSV$ | $XRS \to RSM$ |
| $\to SMV$ | $\to XVS$ | $VXT \to XTT$ | $\to RSV$ |
| $\to VXR$ | $\to XVT$ | $\to XTV$ | $RSS \to RSV$ |
| $\to VXM$ | $RXT \to XTT$ | $VXV \to XVR$ | $\to RSM$ |
| $MSV \to SVX$ | $\to XTV$ | $\to XVT$ | $RXS \to XSV$ |
| $MTV \to TVR$ | $TSM \to SMT$ | $\to XVS$ | $\to XSM$ |
| $\to TVT$ | $\to SMV$ | $VXR \to XRR$ | $RSM \to SMT$ |
| $\to TVS$ | $TSV \to SVX$ | $\to XRM$ | $\to SMV$ |
| $MTT \to TTV$ | $TTV \to TVR$ | $\to XRS$ | $RSV \to SVX$ |
| $\to TTT$ | $\to TVS$ | $VXM \to XMS$ | $XSV \to SVX$ |
| $\to TVS$ | $\to TVT$ | $XMS \to MSM$ | $XSM \to SMT$ |
| $MVR \to VRX$ | $TVR \to VRX$ | $\to MSV$ | $\to SMX$ |
| $MVS \to VSM$ | $TVS \to VSM$ | $XRR \to RRM$ | |
| $\to VSV$ | $\to VSV$ | $\to RRR$ | |
| $MVT \to VTS$ | $TVT \to VTS$ | $XRM \to RMS$ | |

have created the associated $47 \times 47$ matrix and found its largest eigenvalue. The topological entropy of the artificial grammar is $\ln 2.08 \approx 0.7324$.

However, to emphasize how the change of an arc in a directed graph can affect the topological entropy, we also have modified the directed graph in Fig. 4 and computed the topological entropy. By removing the arc which exits the directed graph after an $R$ or an $X$, the transition matrix drops to a $40 \times 40$ matrix with topological entropy $\ln 2 \approx 0.6931$.

## CONCLUDING REMARKS

As the study of implicit learning through artificial grammars continues to mature, there will be more of a necessity for comparing results across experiments and artificial grammars. Topological entropy provides a quantifiable means to measure the complexity of an artificial grammar. The relationship between the probability of successful learning of artificial grammars and the artificial grammars' topological entropy is a natural area for study. Below we give an example of how topological entropy could be used to clarify when memorizing and learning are successful. Further, we relate

topological entropy to a simpler measure of complexity that can be used in certain artificial grammar experiments.

There have been many approaches to the study the implicit learning of the rules of an artificial grammar. Among them, one has the subjects focus on memorizing valid sequences of letters. Other experiments do not allow the subjects enough time to memorize valid sequences, by showing many strings in short amount of time. Mathews, *et al.* (1989) discuss these two approaches. We think that more rules (which reduces the topological entropy) would be easier to memorize, while less rules (which increases the topological entropy) would be easier to learn. Which of these two cases is more complex? More rules corresponds to less arrows in the directed graph, while less rules corresponds to more arrows in the directed graph. The complexity of the subshift or the rule can be measured. Of course, when any letter can follow any other letter, the artificial grammar is the full shift and has the largest possible topological entropy. Although a complex sequence space, the rule is not complex since every sequence can be generated by the rule! However, too few arrows can restrict the subshift to a finite number of sequences. In this case, the rule can be perceived as complex, but the sequence space as simple.

Reed and Johnson (1998) discuss the relationship between the informational complexity and the representational abstractness of artificial grammars and which are more successfully learned implicitly and explicitly. Reed and Johnson (1994, 1998) incorporate a simple measure of complexity used by Cohen, Ivry and Keele (1990), where the complexity is measured by the number of preceding symbols necessary to determine the next symbol. This makes the sequence deterministic, not probabilistic. However, even extending this idea, to basing the complexity solely on the number of symbols needed to determine which symbol is possible in the next stage, is equivalent, in our language, to suggesting that the complexity should be measured by the minimum lift size needed for the transition matrix to be lifted to a Markov representation of the grammar. Among other limitations, this does not consider the size of the alphabet. In comparison, we believe that topological entropy provides a finer, more robust measurement of complexity.

## REFERENCES

Adler, R., Konheim, A., & McAndrew, M. (1965). Topological entropy. *Transactions of the American Mathematical Society, 114,* 309-319.

Blahut, R. E. (1988). *Principles and practice of information theory.* Reading, MA: Addison-Wesley.

Cohen, A., Ivry, R. I., & Keele, S. W. (1990). Attention and structure in sequence learning. *Journal of Experimental Psychology: Learning, Memmory, and Cognition, 16,* 17-30.

Gantmacher, F. R. (1974). The theory of matrices, Volume II. New York, NY: Chelsea Publishing, Co.

Higham, P. A. (1997). Dissociations of Grammaticality and Specific Similarity Effects in Artificial Grammar Learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23,* 1029-1045.

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.

Reber, A. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General, 118,* 219-235.

Reber, A. & Allen, R. (1978). Analogic and abstraction strategies in synthetic grammar learning: A functionalist interpretation. *Cognition, 6,* 193-221.

Reed, J. M. & Johnson, P. J. (1994). Assessing implicit learning with indirect tests: Determining what is learned about sequence structure. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20,* 585-594.

Reed, J. M. & Johnson, P. J. (1998). Implicit Learning: Methodological Issues and Evidence of Unique Characteristics. In Stadler, M. A. & Frensch, P. A. (Eds.), *Handbook of Implicit Learning* Thousand Oaks, CA: Sage Publications, Inc. 261-294.

Robinson, C. (1995). *Dynamical systems: Stability, symbolic dynamics, and chaos.* Ann Arbor, MI: CRC Press.

Shannon, C. E. (1948). The mathematical theory of Communications. *Bell Systems Technical Journal, 27,* 379-423.