The Use of Formal Language Theory in Studies of Artificial Language Learning: a proposal for distinguishing the differences between human and nonhuman animal learners

James Rogers[1]
Earlham College
Richmond, IN USA
jrogers@cs.earlham.edu

Marc D. Hauser[2]
Depts. of Psychology, Organismic & Evolutionary Biology
and Biological Anthropology
Harvard University

1 Introduction

Most of the articles in this collection reference, directly or indirectly, the conjecture from the abstract of Hauser et al. (2002):

> We hypothesize that FLN only includes recursion and is the only uniquely human component of the faculty of language. (Hauser et al. 2002:1569)

While, this conjecture has been productive in sparking research, it was incidental to the primary focus of the original article. Rather, quoting from the conclusion, Hauser et al. presented three central points:

> First,. . . [to] move beyond unproductive debate to [a] more collaborative, empirically focused and comparative research program aimed at uncovering both shared (homologous or analogous) and unique components of the faculty of language. Second, although we have argued that most if not all of FLB is shared with other species, whereas FLN

> may be unique to humans, this represents a tentative, testable hypothesis in need of further empirical investigation. Finally, we believe that a comparative approach is most likely to lead to new insights about both shared and derived features, thereby generating new hypotheses concerning the evolutionary forces that led to the design of the faculty of language. (Hauser et al. 2002:1578)

Our goal here is to contribute to that research program by exploring criteria for experimental design of comparative studies across different populations, including different species, age groups, and subjects with neurological deficits, targeting capabilities relevant to the language faculty.

The comparative method provides a way of analyzing evolutionary phenomena in the absence of genetic or fossil evidence, using empirical data concerning contrasts and parallels between traits in living species to draw inferences about their extinct ancestors. In particular, we are looking to identify shared and unique components of the faculty of language, to identify which of the shared traits are actually homologous rather than independently evolved under similar constraints, to distinguish whether current discontinuities in traits between species are the result of gradual divergence or reflect discontinuities in human evolution and to distinguish whether human traits evolved by gradual extension of common preexisting communication systems or were exapted away from previous non-language adaptive functions. Empirical tests of such issues require studies of differential capabilities across species, in both communication and non-communication domains, including both spontaneous and trained behaviors.

To set the stage for what follows, consider a study, cited in Hauser et al. (2002), reported in Fitch and Hauser (2004). This study employed familiarization/discrimination experiments to test the ability of cotton-top tamarin monkeys to spontaneously generalize patterns of CV syllables spoken by a female (A) and a male (B) voice. The experiments were designed to contrast the ability of the subjects to recognize sequences of the form $(AB)^n$, in which syllables of the two classes alternate, with the ability to recognize sequences of the form A^nB^n, in which there are equal numbers of syllables from each class, all those from one preceding all those from the other. The form $(AB)^n$ was chosen as an example of the class of *Finite State* stringsets, stringsets in which there is an *a priori* bound on the amount of information that must be inferred in distinguishing strings that fit the pattern from those that do not. The form A^nB^n was chosen as an exemplar of the class of *Context Free* stringsets, stringsets that can be generated by *Context Free Grammars* (CFGs) and for which, in principle, the amount of information required to distinguish strings that fit the pattern is proportional to the length of the string. Human languages are generally believed to be at least as complicated as Context Free stringsets.

The ease with which the tamarins mastered the Finite State pattern, in contrast to their inability to master the Context Free pattern, suggests that the ability to generalize non-Finite-State patterns has evolved in humans since the divergence between their ancestors and the ancestors of tamarins. This result, again, has been fruitful in spawning further research, both in terms of comparative evidence and in terms of refining the potential underlying mechanism (Gentner et al. 2006; Perruchet and Rey 2005; Zimmerer and Varley 2007). But one of the things that is clear in reviewing the research thus far is that the problem of designing such experiments and, in particular, of interpreting their results, is extremely challenging. We turn next to a proposal for how one might beneficially continue this line of research, and in particular, set up a range of patterns or stringsets that enable more systematic explorations and discoveries of the underlying psychological mechanisms.

2 Formal design of recognition experiments

While very different cognitive processes are involved, training experiments and familiarization/discrimination experiments have essentially the same formal structure. We will concentrate on the latter. In these experiments, subjects are familiarized with the intended stringset by exposure to some sample of the strings in the set. They are then tested with some sample of strings including both those in and those not in the intended set. The task of the subject, then, is to infer the pattern of a relatively large, possibly infinite, stringset from a small sample. As they are exposed to only positive examples, any stringset that includes this sample is consistent. Clearly the subjects cannot extract patterns that are more complicated than they are able to distinguish. Our expectation, as well, is that the subjects will not consistently fail to extract patterns of a given level of complexity if they are capable of recognizing them. Thus, the stringset they arrive at is an indicator of the capacity of the cognitive machinery they can bring to bear on the task.

If a subject consistently, over a variety of strings, finds strings within the intended set to be “unsurprising” and those not in it to be “surprising” (where these terms refer to the relative novelty of the stimuli and the extent to which they trigger greater attention or more robust responses) then there is reason to conclude that they have generalized correctly. From such results, we are licensed to conclude that the target species has cognitive faculties sufficient to recognize stringsets of at least this level of complexity. If a subject consistently, over a variety of strings, finds some string in the intended set to be surprising or some string not in the intended set to be unsurprising, then one can conclude that they have not extracted the intended pattern. If subjects of a particular species do this consistently, over a variety of stringsets within a given complexity class, then we may plausibly conclude that the species does not have the cognitive faculties required to recognize stringsets of that level of complexity.

The primary issue for the experimental design is determining which set the subject has generalized to. There will always be simpler sets that are consistent with the familiarization set (the set of all strings over the relevant alphabet, for example) and in general these will include both supersets and subsets of the intended set. The strings in the set A^nB^n, for example, have a variety of different features which could be generalized by the subject. For example, all of the As precede all of the Bs, the strings are all of even length and the number of As is equal to the number of Bs. In any finite subset, there will be additional features, upper and lower bounds on the number of As, for example.

[Insert Figure 1 here]

Figure 1 illustrates the situation when the subject has been exposed to just the string $AAABBB$. The set marked A^mB^n includes all strings in which the As precede the Bs, an extremely simple stringset (in fact, as described below, a strictly 2-local stringset). The set marked $A^iB^j\ 2|(i+j)$ is the subset of those that are of even length,[3] a Finite State stringset. The set marked $A^iB^i\ i \leq 3$ is the finite subset of A^nB^n in which n is no greater than three. The set marked $|w|_A = |w|_B$ is the set of all strings in which As and Bs are equinumerous,[4] a Context Free stringset.

In order to distinguish subjects that have generalized to A^nB^n, and therefore must be able to recognize at least some Context Free stringsets, from those that can recognize only stringsets of strictly lower complexity we need to expose them to strings that are in the set-theoretic symmetric difference between the simpler sets and A^nB^n. One can detect that the subject has generalized to a simpler set that is a subset of the intended set if there are strings in the intended set but not in the simpler set that are surprising to them. One can detect that the subject has generalized to a simpler set that is a superset of the intended set if there are strings in the simpler set but not in the intended set that are not surprising.

Pairing $AAABBB$ with $AABBBB$, for example, can reveal whether the subject has generalized to A^nB^n or $A^iB^j\ 2|(i+j)$ and provides evidence of being able to recognize stringsets beyond the Finite State. Pairing it with $AABBB$, on the other hand, distinguishes subjects that have generalized to the strictly local stringset A^mB^n, but fails to distinguish Finite State from non-Finite-State; $AABBB$ will be novel to both those that have generalized to A^nB^n and those that have generalized to $A^iB^j\ 2|(i+j)$.

Pairing $AAABBB$ with $AAAABBBB$ can reveal whether the subject has erroneously generalized to $A^iB^i\ i \leq 3$. Although one can never rule out the possibility that the subject has generalized to a finite subset of the intended set that happens to include both the strings in the familiarization set and those in the discrimination set that are in the intended set, one can be reasonably sure that they have not generalized to a finite set by including strings in the discrimination set that are longer than any in the familiarization set and therefore unlikely to be in any finite generalization. Thus, failure to perceive $AAAABBBB$ as novel suggests that the subject has not generalized to any finite subset of A^nB^n.

Finally, $AABBBA$ or $ABABAB$ will appear novel even to subjects that can recognize only strictly local stringsets, although they will not appear novel to a subject that has overgeneralized to $|w|_A = |w|_B$. Thus these strings fail to provide any evidence at all about the boundary between Context Free and simpler stringsets. The punch line here is straightforward: it is critical to the validity of these experiments that we have a clear idea not only of the intended stringset but also of the stringsets that it could, potentially, be mistaken for.

Even with careful choice of stimuli there are still a variety of pitfalls in design and interpretation of these experiments that must be avoided. Examples of utterances of English that satisfy the patterns of Fitch and Hauser (2004) include (1a), of the form $(AB)^n$, and (1b), of the form A^nB^n.[5]

(1)
a. $\{($**ding dong**$)^n\}$
b. $\{$**people**^{n} **left**$^n\}$
c. $\{$**people who were left** $($**by people who were left**$)^n$ **left**$\}$
d. $\{$**people who were left** $($**by people who were left**$)^{2n}$ **left**$\}$

The fact that (1b) is well-formed is usually not clear at first encounter for n greater than two or three. The paraphrase in (1c) is much easier to parse, but it is, in fact, Finite State. This points out one of the difficulties of probing the boundaries of recognition capabilities in this way. Humans seem to be able to process, with no conscious effort, many types of utterances which, as stringsets, are not Finite State. But there are many well formed utterances of the same sort that are utterly opaque without careful conscious analysis. The fact that, in humans, the same stimulus may be processed with potentially distinct faculties with differing degrees of success demonstrates the difficulty of isolating a particular faculty experimentally. This is one of the reasons for testing both spontaneous and learned behavior.

A more fundamental issue is raised by the fact that (1d) represents a class of utterances that, while Finite State, is unlikely to be accurately identified by most English speakers without consciously searching for the key to the pattern (i.e., the number of prepositional phrases is required to be even). Thus, while (1a) and (1d) are both Finite State there seems to be a very dramatic difference in the degree of difficulty of recognizing them. The Finite State stringsets do not, in fact, present a uniform level of difficulty of recognition. There is a rich hierarchy of classes of stringsets within the Finite State which corresponds to a range of gradations in cognitive capabilities. Viewed as an instrument for probing the boundaries of these capabilities, the pair of stringsets $(AB)^n$ and A^nB^n lack resolution. Thus it is not only important to understand which sets the subject may erroneously infer, it is also important to understand what ranges of complexity classes may be relevant to the faculties that are being explored.

Finally, an issue of interpretation arises when the Fitch and Hauser (2004) result is taken as suggesting that one of the capabilities distinguishing humans from tamarins is a faculty for handling recursion. Fitch and Hauser do *not* claim that their results suggest that tamarins lack a capacity for handling recursion, *per se*, only that they cannot handle phrase structure, i.e., Context Free, patterns. In fact recursion is not actually necessary for recognizing the pattern A^nB^n. Algorithmically, one can recognize this using a single counter or by inferring a single binary relation over the input. Context Free grammars are not the only algorithmic mechanisms capable of generating or recognizing Context Free stringsets. Results establishing the definitional equivalence of distinct mechanisms are ubiquitous in Formal Language Theory (FLT), suggesting that there is no bound on the varieties of algorithmic mechanisms that are equivalent in the sense of defining the same classes of stringsets. Consequently, one cannot make inferences about cognitive mechanisms based on the details of any particular generative mechanism. Rather one is licensed only to make inferences based on the common characteristics of the entire class of equivalent mechanisms. The question is, how are those common characteristics to be determined?

3 Dual characterizations of complexity classes

Most of FLT has developed from a foundation of abstract algorithmic processes: grammars, which generate strings, and automata, which recognize them (Hopcroft and Ullman 1979). The characteristics of these processes allow one to reason about the structure of the stringsets they define, establishing pumping lemmas,[6] for example, or Nerode-style characterizations.[7] An understanding of this structure is critical to the design of experiments of the sort we are interested in: How are the classes of stringsets related to each other? Which stringsets distinguish the classes?

In parallel with these algorithmic characterizations, are *descriptive* characterizations, characterizations of classes of stringsets based directly on the properties of the strings they contain (Medvedev 1964; Büchi 1960; Elgot 1961; Thatcher and Wright 1968; McNaughton and Papert 1971; Thomas 1982; Straubing 1994; Libkin 2004). When one specifies a stringset with properties such as "there are equal numbers of syllables from each class of syllables" one is specifying the stringset descriptively. Descriptive characterizations of classes of stringsets focus on the nature of the information required to distinguish strings that meet or do not meet such patterns—the kinds of relationships between the components of the strings that must be detected if the patterns are to be recognized.[8] The main strength of these characterizations, from our point of view, is that they do not presuppose any particular algorithmic mechanism. Any mechanism that can recognize or generate stringsets that fall properly within a descriptive class must necessarily be sensitive to the sort of information about strings that determines the class. Hence they provide a foundation for reasoning about the common characteristics of entire classes of formally equivalent mechanisms, be they abstract algorithmic mechanisms or concrete cognitive mechanisms realized in organisms.

A second strength is their generality. Any description of strings that can be expressed within the means of a descriptive class defines a stringset within that class. By varying these means systematically we can cover an extremely broad range of seemingly disparate ways of specifying patterns with a relatively small set of descriptive classes.

The fulcrum of the methodology we employ here is the deep fact that these two approaches to distinguishing classes of stringsets correspond. Descriptive classes can be characterized in terms of grammars and automata and *vice versa*. This allows us to use the descriptive characterizations as the basis for reasoning about cognitive mechanisms (avoiding the fallacy of basing such reasoning on the corresponding algorithmic mechanisms) while using the algorithmic characterizations to guide the experimental design.

4 The sub-regular hierarchy

The range of complexity classes that falls between stringsets such as (1a) and (1d), the *sub-regular hierarchy*, has been largely overlooked by linguists in part because the Chomsky hierarchy starts with the Finite State and in part because, as human languages are widely assumed to be at least Context Free, there has been little motivation to explore sub-Finite-State classes in this context. But our interest is in recognition capabilities across many populations, including different species, age groups and neurological populations, targeting potential precursors to human faculties. Consequently, we are fundamentally interested in classes of stringsets that are simpler than human languages as these may well form some of the critical evolutionary and ontogenetic building blocks. More importantly, the classes in the sub-regular hierarchy correspond to a clear hierarchy of cognitive mechanisms and, since the classes in the hierarchy are defined purely in terms of types of relationships between positions in strings, that hierarchy of cognitive mechanisms will be relevant to any faculty that provides a syntax-like function, i.e., that processes stimuli solely as sequences of events, independent of meaning.

The remainder of this section presents a brief overview of the formal aspects of the sub-regular hierarchy. Due to space limitations we skip most of the formal detail, presenting just the relevant characteristics of the classes of the hierarchy. We close the section with a list of criteria for the design and interpretation of acoustic pattern recognition experiments based on these classes. A more through exposition can be found in Rogers and Pullum (2007).

4.1 Strictly Local Stringsets

The simplest definitions we will explore are those that specify strings solely in terms of the sequences of symbols that are permitted to occur adjacently in them, that is, in terms of the n-grams making up the string. Since our n-grams are not associated with probabilities we will refer to them with the standard FLT terminology: k-factors.

A k-factor is just a length k sequence of symbols. A *Strictly k-Local Definition* is a set of k-factors drawn from some finite alphabet of symbols augmented with beginning of string ($\rtimes$) and end of string ($\ltimes$) markers. A string w *satisfies* a strictly k-local definition if and only if (*iff*) the set of k-factors of the augmented string $\rtimes w \ltimes$ is a subset of those included in the definition. A stringset is in the class SL_k iff it can be defined with a strictly k-local definition, it is *Strictly Local* (SL) (McNaughton and Papert 1971) iff it is SL_k for some k.

The set of strings of the form $(AB)^n$ is an example of an SL_2 stringset, being definable by the (minimal) set of 2-factors:

(2) $$\mathcal{D}_{(AB)^n} \stackrel{\text{def}}{=} \{\rtimes A, AB, BA, B\ltimes\},$$

which asserts that the string must begin with an A, end with a B, that every A is followed by a B and that every B other than the last is followed by an A. The set of strings licensed by this definition, denoted $L(\mathcal{D}_{(AB)^n})$, is $\{(AB)^i \mid i > 0\}$.

Membership in a Strictly Local stringset depends only on the k-factors in isolation: a string satisfies an SL_k definition iff each of the k-factors in the string is independently licensed by that definition. From a cognitive perspective, all that is required to recognize a strictly k-local stringset is attention to each block of k symbols which occurs in the string. If the string is presented sequentially in time, this amounts to remembering just the last k consecutive events that have been encountered.

The key to reasoning about the structure of SL stringsets in general is a theorem which characterizes them in terms of a property known as *suffix substitution closure*: a stringset is SL iff there is some k for which it is closed under the substitution of suffixes that begin with the same $(k-1)$-factor.

This allows us to identify non-SL stringsets and to construct minimal pairs of stringsets which can be diagnostic of the ability to generalize SL patterns. One simple non-SL stringset is the set of strings of As and Bs in which there is at least one B, a stringset we call Some-B:

$$\text{Some-}B \stackrel{\text{def}}{=} \{w \in \{A,B\}^* \mid |w|_B \geq 1\}. \tag{3}$$

To see that this is not SL_k for any k, note that, for any k, strings of the form $A \ldots A \cdot \underbrace{A \ldots A}_{k-1} \cdot BA \ldots A$ and those of the form $A \ldots AB \cdot \underbrace{A \ldots A}_{k-1} \cdot A \ldots A$ are all in Some-B, but the result of substituting the suffix of a string of the second form, starting at the marked sequence of $k-1$ As, for the suffix of a string of the first form is $A \ldots A \cdot \underbrace{A \ldots A}_{k-1} \cdot A \ldots A$ which is not in Some-B. So, Some-B does not exhibit suffix substitution closure and can not be specified with an SL-definition.

4.2 Locally Testable Stringsets

In order to distinguish stringsets like Some-B it is necessary to differentiate between strings on the basis of the whole set of k-factors that they contain, not just on the basis of the individual k-factors in isolation. Descriptions, at this level, are k-*expressions*, formulae in a propositional language in which the atomic formulae are k-factors which are taken to be true of a string w iff they occur in the augmented string $\rtimes w \ltimes$. More complicated k-expressions are built up from k-factors using the usual logical connectives, e.g., for conjunction ($\wedge$), disjunction ($\vee$) and negation ($\neg$). Stringsets are *Locally Testable* (LT) (McNaughton and Papert 1971) iff they are definable by a k-expression, for some k. As an example, Some-B is defined by the following 2-expression, which is true of strings that either start with B or include AB:

$$\varphi_{\text{Some-}B} \stackrel{\text{def}}{=} \rtimes B \vee AB \tag{4}$$

The relation between strings and the k-expressions which holds iff the string satisfies the k-expression is denoted $\models$; the set of strings licensed by a k-expression φ, then, is $L(\varphi) \stackrel{\text{def}}{=} \{w \mid w \models \varphi\}$. For the k-expression $\varphi_{\text{Some-}B}$, $L(\varphi_{\text{Some-}B}) = \text{Some-}B$. As this witnesses, k-expressions have more descriptive power than SL_k definitions.[9]

Because membership in an LT_k stringset depends on the whole set of k-factors that occur in a string k-factors can be required to occur as well as prohibited from occurring and they can be required to occur in arbitrary combinations—to occur together, to occur only if some other combination of k-factors does not occur, etc.

On the other hand, membership in an LT_k stringset depends *only* on the set of k-factors which occur in the string. Any mechanism which can distinguish strings on this basis is capable of recognizing (some) LT_k stringset. Conversely, any mechanism that can distinguish members of a stringset that is LT_k (but not SL) must be able to distinguish strings on this basis. Cognitively, this corresponds to being sensitive to the set of all k-factors that occur anywhere in the input. If the strings are presented sequentially, it amounts to being able to remember which k-factors have and which have not been encountered in the stimulus.

An example of a stringset that is not LT is the set of strings over $\{A, B\}$ in which *exactly* one B occurs:

(5) $$\text{One-}B \stackrel{\text{def}}{=} \{w \in \{A, B\}^* \mid |w|_B = 1\}$$

To see that this is not LT note that, for any k, A^kBA^k is in One-B while $A^kBA^kBA^k$ is not. Since these have exactly the same set of k-factors, however, there is no k-expression that can distinguish them.

4.3 FO(+1) Definable Stringsets

The next step in extending the complexity of the stringsets we can distinguish is to add the power to discriminate between strings on the basis of the specific positions in which blocks of symbols occur rather than simply on the basis of blocks of symbols occurring somewhere in the string. Descriptions at this level are first-order logical sentences (formulae with no free variables) over a restricted string-oriented signature. Atomic formulae assert relationships between variables $(x, y, \ldots)$ ranging over positions in the string: $x \triangleleft y$ (meaning that y is the next position following x), $x \approx y$ (x and y are the same position) and $A(x), B(x), \ldots$ (A occurs in position x, etc.). Larger formulae are built from these using the logical connectives and existential ($\exists$) and universal ($\forall$) quantification. A string w satisfies an existential (sub)formula $(\exists x)[\varphi(x)]$ (i.e., $w \models (\exists x)[\varphi(x)]$) iff some assignment of a position for x makes it true. It satisfies a universal (sub)formula iff all such assignments make it true. The class of stringsets definable in this way is denoted FO($+1$).

As an example, One-B is FO($+1$) definable:

(6) $$\text{One-}B = \{w \in \{A, B\}^* \mid w \models (\exists x)[B(x) \wedge (\forall y)[B(y) \rightarrow x \approx y]\,]\},$$

which asserts that there is some position x in which a B occurs and that all positions y in which B occurs are that same position. As One-B witnesses, FO($+1$) definitions are more expressive than k-expressions.[10]

It turns out that a stringset is FO($+1$) definable iff it is *Locally Threshold Testable* (LTT) (Thomas 1982). Such stringsets distinguish strings only on the multiplicity of the k-factors which occur in them and only relative to a threshold t above which multiplicities of the k-factors cannot be distinguished.This characterization is the key to identifying stringsets that are not FO($+1$) definable.

An example of a non-LTT, hence non-FO($+1$)-definable, stringset is the set of strings over $\{A, B, C\}$ in which some B occurs before any C:

(7)

$$B\text{-before-}C \stackrel{\text{def}}{=} \{w \in \{A, B, C\}^* \mid \text{ at least one B precedes any C}\} \notin \text{LTT}.$$

To see this, note that, for any k, $A^kBA^kCA^k$ and $A^kCA^kBA^k$ have exactly the same number of occurrences of every k-factor and are therefore indistinguishable in the LTT sense for any threshold t.

4.4 FO($<$) Definable Stringsets

We can, again, increase the complexity of the stringsets we can distinguish by, beyond just discriminating individual occurrences of k-factors, differentiating between strings based on the order in which those k-factors occur. FO($<$) descriptions extend FO($+1$) descriptions to include a *precedence* relation $\triangleleft^*$ which corresponds to $<$ on the domain of the structure. An example of a FO($<$) definable stringset that is not FO($+1$) definable is B-before-C:

(8) $$B\text{-before-}C = \{w \mid w \models (\exists x)[\ C(x) \rightarrow (\exists y)[B(y) \wedge y \triangleleft^* x]\ ]\ \}.$$

This formula asserts that if there is a position x in which C occurs, then there is a position y in which B occurs which precedes x in the string.

Strikingly, it turns out that extending the signature with the precedence relation in this way is exactly equivalent to adding a concatenation operator (•) to the language of k-expressions. A stringset is *Locally Testable with Order* (LTO) (McNaughton and Papert 1971) iff it is definable with a k-expression augmented in this way. Note that we can define B-before-C with

(9) $$(\neg(\rtimes C \vee AC \vee BC) \bullet (\rtimes B))\ \vee\ \neg(\rtimes C \vee AC \vee BC),$$

which is satisfied either by strings which consist of a substring in which no C occurs followed by one which starts with B or by those in which, simply, no C occurs at all.

If the strings are presented sequentially, this corresponds to being able to apply a fixed set of sequences of LTT-style threshold counting recognition strategies. Effectively, it corresponds to being able to count occurrences of events up to some threshold, coupled with the capacity to reset the counters some fixed number of times.

The most useful abstract characterization of the structure of the FO($<$) definable stringsets follows from an automata-theoretic characterization due to McNaughton and Papert (1971).[11] In essence, a stringset is FO($<$) definable iff there is some constant $n > 0$ for which, whenever a string in the set includes a block of symbols which is iterated at least n times, then it also includes all strings which are identical except that that block is iterated an arbitrary number of times greater than n.

An example of a non-FO($<$) definable stringset is the set of strings over $\{A, B\}$ in which the number of Bs is even:

(10) $$\text{Even-}B \stackrel{\text{def}}{=} \{w \in \{A, B\}^* \mid |w|_B = 2i,\ 0 \leq i\} \notin \text{LTT}$$

This set includes strings in which B is repeated at least n times, whatever value the constant n might have. But adding one more B to that block of Bs produces a string with an odd number of Bs.

4.5 MSO Definable Stringsets

The next step in increasing the power of our descriptions is to introduce abstraction—to assign the occurrences of symbols in the strings to abstract categories and to discriminate between the strings on the basis of the sequence of categories rather than the sequence of symbols themselves. Categories of this sort can be captured in logical languages by allowing quantification over not just individual positions in the strings but over sets of positions, i.e, by adding *Monadic Second-Order* variables along with their quantifiers. An example of an MSO definition of a stringset is:

(11) $$\begin{array}{ll}(\exists X_0)[& (\forall x)[\neg(\exists y)[y \triangleleft x] \rightarrow X_0(x)] \wedge \\ & (\forall x, y)[x \triangleleft y \rightarrow (X_0(x) \leftrightarrow \neg(X_0(y))] \wedge \\ & (\forall x)[\neg(\exists y)[x \triangleleft y] \rightarrow \neg(X_0(x))] \quad]\end{array}$$

This defines the set of all strings that are of even length: it asserts that there is a subset of positions in the string (X_0) which includes the first position in the string, alternates between inclusion and exclusion of adjacent positions and does not include the last position.[12] This formula can be modified to ignore everything except positions labeled B, which gives a definition of Even-B. So MSO is a proper superclass of FO($<$).

The assignments of positions in the strings to MSO variables is equivalent, in a strong sense, to the association of those positions with states in the runs of Finite State automata. And, in fact, the MSO-definable stringsets are exactly the Finite State stringsets (Medvedev 1964; Büchi 1960; Elgot 1961).

The abstract character of the MSO definable stringsets is a consequence of their characterization by Finite State automata. Two strings w and v are *Nerode Equivalent* with respect to a stringset L over an alphabet Σ (denoted $w \equiv_L v$) iff for all strings u over Σ, $wu \in L \Leftrightarrow vu \in L$. A stringset L is Finite State iff $\equiv_L$ partitions the set of all strings over Σ into finitely many equivalence classes. These equivalence classes correspond to the categories (to the distinguished subsets) of the MSO definitions.

Since the MSO definable stringsets extend the FO definable stringsets, the ability to discriminate between strings based on categorizing the events that comprise them in this way implies the ability to discriminate between them based on the multiplicity of their k-factors, counting up to a fixed threshold. As the definability of Even-B suggests, it extends this to the ability to distinguish strings based on the residues (remainders) of those multiplicities relative to some modulus, in essence to count modulo those thresholds, resetting the counters unboundedly often. It is, in fact, equivalent to using a finite set of counters that count modulo some threshold.

It does not, on the other hand, provide the ability to count arbitrarily high; the stringset $\{A^n B^n \mid n > 0\}$, for example, is not MSO definable because the Nerode equivalence for the set is:

(12)

$$w \equiv_{A^n B^n} v \Leftrightarrow w, v \notin \{A^i B^j \mid i, j \geq 0\} \text{ or } |w|_A - |w|_B = |v|_A - |v|_B \,.$$

for which there are infinitely many equivalence classes.

Cognitively, the Finite State stringsets characterize an extremely broad range of mechanisms: whenever there is a fixed finite bound on the amount of information a mechanism can retain while processing a string the mechanism will be limited to recognizing at most Finite State—equivalently, MSO definable—stringsets. Note, though, that the fact that an organism can recognize non-Finite-State stringsets does not imply that the physical mechanisms they employ must have access to a store of unbounded size. The mechanisms may implement an algorithm which, in principle, requires unbounded storage which fails on sufficiently long or sufficiently complicated inputs. Or would if it ever encountered such.

4.6 Beyond Finite State

The ability to recognize stringsets that are not Finite State implies discriminating between strings on the basis of information about the string the size of which depends on the length of the string—being able to count to values that are proportional to the length of the string, for example. The weakest class of the Chomsky hierarchy which properly extends the class of Finite State stringsets is the class of *context free* stringsets, those that can be generated by context free grammars. But there are many ways of utilizing amounts of information that depend on the length of the string that do not provide the ability to recognize context free stringsets in general. The single counter that suffices to recognize A^nB^n can also recognize sets of well-nested brackets (the single bracket Dyck stringsets, D_1), but it does not suffice to recognize D_2, the two bracket Dyck stringsets, for example. There are, in fact, multiple hierarchies which partition the context free stringsets just as the sub-regular hierarchy partitions the Finite State stringsets. Experiments that are intended to differentiate abilities beyond the Finite State will need to be based on formal analyses of this territory similar to the analysis of the sub-regular hierarchy we have provided here.

5 Recognition experiments

[Table 1 goes here]

Tables 1 and 2 summarize the conclusions that can be drawn about the cognitive capabilities of the subjects given the results of recognition experiments based on the sub-regular hierarchy.

To test the recognition capabilities of a species relative to these classes one could use experiments based on patterns such as those given in the *Example* column of Table 1. Species with the ability to recognize patterns of that sort, having been familiarized to strings of the form given in the *In* column, should generally find novel strings of that form unsurprising while finding strings of the form given in the *Out* column surprising. If, on the other hand, subjects consistently fail to find strings of the form given in the *Out* column surprising after being familiarized to strings of the form given in the *In* column, this would be evidence that the target species is unable to distinguish patterns of that level of complexity.

[Insert Table 2 here]

Table 2 summarizes the conclusions that can be drawn from such results. *Recognize* outcomes represent lower bounds on the species' capacity. They indicate that the subject has at least the cognitive capabilities given in the *Cognitive significance* column. *Fail* outcomes represent upper bounds, evidence that the subject is limited in the way indicated in the *Cognitive significance* column.

6 Conclusion

Acoustic pattern recognition or artificial language learning experiments provide a powerful approach to understanding the ontogenetic and evolutionary building blocks of the human language faculty, distinguishing what is shared with other species and what is uniquely human. To design experiments of this sort with highly interpretable results, one must be clear about a bewildering array of formal considerations. These include the cognitive capacities that the experiment is designed to probe, the abstract structure of the classes of stringsets that those capabilities characterize, the structure of the stringsets the intended stringset may be confused with and what such confusion signifies about the capabilities of the subjects.

The subregular hierarchy is a range of complexity classes that correspond to a range of cognitive capabilities that are, by virtue of the descriptive characteristics of the classes, necessarily relevant to any faculty that processes aural stimuli solely as sequences of events. As such it provides a clear framework for resolving these issues. We hope that our exposition of this hierarchy will facilitate continued experimentation of this type.

Notes

[1] This work was undertaken while the first author was in residence at the Radcliffe Institute for Advanced Study

[2] The authors would like to express their gratitude to Geoff Pullum for his extensive collaboration on this work and to Barbara Scholz and the referees for their helpful suggestions.

[3] The expression $x|y$ denotes the property of y being an integral multiple of x.

[4]The expression $|w|$ denotes the length of the string w—the number of symbols it contains. The expression $|w|_A$ denotes the number of As it contains.

[5]These two examples are due to Geoffrey K. Pullum.

[6]Pumping lemmas establish certain types of closure properties of stringsets: if a stringset includes strings with length exceeding an arbitrary but fixed bound then it also includes strings in which certain substrings are repeated arbitrarily many times. Shieber (1985) and Huybregts (1984) employed results of this sort to show that certain dialects of Swiss-German are not representable as Context Free stringsets.

[7]These characterize the structure of stringsets in terms of the way in which certain relations on strings, which depend on the stringset, partition the set of all strings into equivalence classes.

[8]We provide concrete examples of what we mean by "nature of the information" as we survey a range of descriptive classes in the next section.

[9]It is easy to translate an SL_k description into a k-expression defining the same stringset, so LT_k is a strict superclass of SL_k.

[10]Again, k-expressions can be easily translated into FO($+1$) formulae; so FO($+1$) is a strict superclass of LT.

[11]A Finite State automaton recognizes an FO($<$) stringset iff its syntactic monoid is aperiodic.

[12]The reader should note the similarity between this definition and the SL_2 definition of $(AB)^n$ (Equation 2).

References

Büchi, J. R.
1960 Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **6**:66–92.

Elgot, C. C.
1961 Decision problems of finite automata and related arithmetics. *Transactions of the American Mathematical Society*, **98**:21–51.

Fitch, W. T. and M. D. Hauser
2004 Computational constraints on syntactic processing in nonhuman primates. *Science*, **303**:377–380.

Gentner, T. Q., K. M. Fenn, D. Margoliash, and H. C. Nusbaum
2006 Recursive syntactic pattern learning by songbirds. *Nature*, **440**:1204–1207.

Hauser, M. D., N. Chomsky, and W. T. Fitch
2002 The faculty of language: What is it, who has it and how did it evolve. *Science*, **298**:1569–1579.

Hopcroft, J. E. and J. D. Ullman
1979 *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, Mass.

Huybregts, R.
1984 The weak inadequacy of context-free phrase structure grammars. In G. J. de Haan, M. Trommelen, and W. Zonneveld, eds., *Van Periferie Naar Kern*, pp. 81–99. Foris Publications, Dordrecht.

Libkin, L.
2004 *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. Springer, Berlin/New York.

McNaughton, R. and S. Papert
1971 *Counter-Free Automata*. MIT Press, Cambridge, MA.

Medvedev, Y. T.
1964 On the class of events representable in a finite automaton. In E. F. Moore, ed., *Sequential Machines—Selected Papers*, pp. 215–227. Addison-Wesley, Reading, Mass. Originally in Russian in Avtomaty (1956), pp. 385–401.

Perruchet, P. and A. Rey
2005 Does the mastery of center-embedded linguistic structures distinguish humans from nonhuman primates. *Psychonomic Bulletin and Review*, **12**:307–373.

Rogers, J. and G. K. Pullum
2007 Aural pattern recognition experiments and the subregular hierarchy. In *Mathematics of Language 10*. UCLA.

Shieber, S.
1985 Evidence against the context-freeness of human language. *Linguistics and Philosophy*, **8**:333–343.

Straubing, H.
1994 *Finite Automata, Formal Logic and Circuit Complexity*. Birkhäuser, Boston.

Thatcher, J. W. and J. B. Wright
1968 Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, **2**(1):57–81.

Thomas, W.
1982 Classifying regular events in symbolic logic. *Journal of Computer and Systems Sciences*, **25**:360–376.

Zimmerer, V. and R. Varley
2007 Recursive syntax in patients with severe agrammatism. Recursion in Human Languages, Illinois State University, Normal, Illinois.

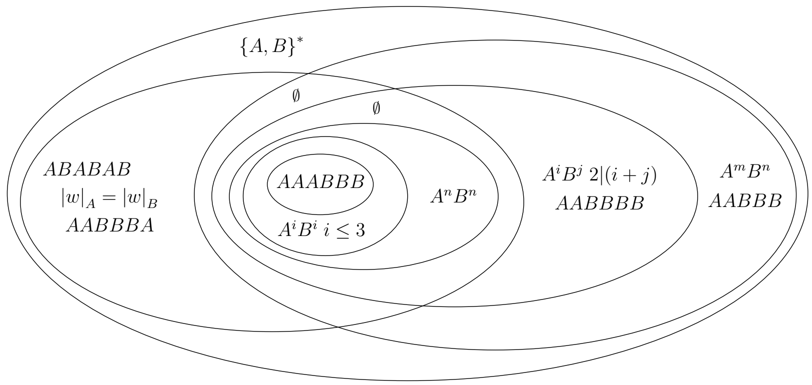

Figure 1: Testing recognition of A^nB^n.

Class	Example	In	Out
SL	$(AB)^n$	$(AB)^{i+j+1}$	$(AB)^iBB(AB)^j$
LT	Some-B	A^iBA^j	A^{i+j+1}
FO(+1)	One-B	A^iBA^{j+k+1}	$A^iBA^jBA^k$
FO(<)	B-before-C	$A^iBA^jCA^k$	$A^iCA^jBA^k$
MSO	Even-B	B^{2i}	B^{2i+1}
CF	A^nB^n	A^nB^n	$A^{n+1}B^{n-1}$

Table 1: Distinguishing classes of the sub-regular hierarchy experimentally.

Class	Outcomes	Cognitive significance
SL	Recognize $(AB)^n$	Sensitive to a fixed length block of the immediately prior events
	Fail Some-B	*Only* sensitive to the immediately prior events
LT	Recognize Some-B	Sensitive to which fixed length blocks of events occur in the input, effectively being able to recall sequences of events that occur at arbitrary points
	Fail One-B	Insensitive to multiplicity or order of blocks
FO($+1$)	Recognize One-B	Sensitive to the multiplicity of events that occur in the input, at least up to some fixed threshold
	Fail B-before-C	Insensitive to order of blocks
FO($<$)	Recognize B-before-C	Sensitive to the multiplicity, up to some fixed threshold, of events that occur in the input and to the order in which a fixed number of events occur, in effect counting to a threshold and resetting the counters up to a fixed number of times
	Fail Even-B	Insensitive to order of events beyond some fixed number
MSO	Recognize Even-B	Capable of classifying the events in the input into a finite set of abstract categories and sensitive to the sequence of those categories. Subsumes *any* recognition mechanism in which the amount of information retained is limited by a fixed finite bound.
	Fail A^nB^n	fixed finite bound, independent of the input, on amount of information retained

Table 2: Cognitive significance of recognition results.