Aural Pattern Recognition Experiments and The Subregular Hierarchy

1 Introduction

The last several years have seen a number of intriguing experiments aimed at differentiating the aural pattern recognition capabilities of various species [HCF02, FH04, GFMN06, PR05]. The idea is that these experiments may provide indirect evidence about those cognitive faculties of our common ancestors and the foundation for the evolution of human language.

The fulcrum connecting the empirical data and conclusions about the characteristics of the cognitive mechanisms the experiments are intended to explore is formal language theory (FLT). Characterizations of the structure of language classes (such a Nerode-style characterizations and pumping lemmas) motivate the choice of stimulus patterns the experiments employ and characterizations of those classes in terms abstract computational mechanisms (such as grammars and automata) provide clues about the cognitive mechanisms in play.

Nearly all of the FLT employed in this work is based in the Chomsky Hierarchy, but there are a number of reasons for questioning whether this is really the right place to be looking. To begin with, viewed as an instrument in the context of these experiments, the CH seems to lack resolution. What is typically taken as emblematic of CFLs is far too hard for humans (processing $\{\mathbf{people}^n\ \mathbf{left}^n \mid n \geq 1\}$ rapidly outstrips any human being's aural pattern recognition abilities), while what is taken as representative of regular languages is far too easy (being able to recognize $\{(\mathbf{ding\ dong})^n \mid n \geq 1\}$ should hardly count as evidence of ability to handle arbitrary regular languages).

Beyond that, traditional grammar- and automata-theoretic classes seem to presuppose specific types of structure or specific classes of recognition mechanisms, raising questions about whether these are necessarily relevant to the cognitive mechanisms under study.

There is, in fact, a rich hierarchy of classes of stringsets, ranging from the Strictly-Local through the Star-Free stringsets, that is properly included in the class of Regular stringsets and which is reiterated within the CFLs when they are taken as sets of structures rather than just sets of strings. In addition to providing finer resolving power in the range of capabilities that seem to be relevant to these studies, this subregular hierarchy includes some of the earliest and most well-developed examples of *descriptive* characterizations of language-theoretic complexity classes [MP71, Tho82, GR90, BS05].

Descriptive (model-theoretic) characterizations focus on the nature of the information about the properties of a string (or structure) that is needed in order to distinguish those which exhibit a pattern from those which do not. Consequently, they are independent of specific mechanisms for specifying or recognizing the patterns.

The descriptive characterizations, however, do not stand alone. These classes are also characterized by a range of abstract mechanisms including grammars, automata and, in some cases, artificial neural networks. These characterizations in more traditional FLT terms provide a means of reasoning about the structural properties both of the individual stringsets that satisfy a pattern and about the class of stringsets that can be described within the means of the class.

Note that the grammar- and automata-theoretic characterizations do not determine the actual form of a cognitive mechanism for recognizing patterns within the class. The fact that there are many equivalent characterizations of each class implies that no conclusions can be drawn about the specific details of the mechanism. What one can conclude, on the other hand, is that whatever the actual mechanism is it must be sensitive to the kind of information that characterizes the descriptive class and that the class of patterns that it can recognize will exhibit those properties that are characteristic of the class.

Thus classes that can be characterized both automata- or grammar-theoretically and descriptively, while not presupposing any concrete details of the mechanism, serve both ends of these aural pattern recognition studies. By providing characterizations of the structure of the stringsets that satisfy particular classes of patterns, they provide a means of designing experimental protocols that can resolve contrasts between the cognitive capabilities of the subjects. By providing highly abstract characterizations of the mechanisms which can process particular classes of patterns they provide a means of drawing conclusions about the differences in the capabilities of cognitive mechanisms that can account for observed differences in recognition abilities.

In this paper we revisit the subregular hierarchy paying particular attention three aspects: the broad generality of the descriptive characterizations of the classes, the consequences their structural characteristics have for the design of pattern recognition experiments and the nature of the conclusions about the cognitive mechanisms involved that these experiments can support. Most of the FLT results are either already well established or could probably be best attributed to folklore. What we have to offer are two methodological contributions. The first is general. We believe that our allocation of roles between the descriptive and mechanism-oriented characterizations, in which conclusions about potential cognitive mechanisms are based only on the more abstract descriptive characterizations and the mechanism-oriented characterizations are reserved for providing structural information about the definable sets, provides an approach that is applicable to a wide range of pattern-based experiments. The second is specific to the current experiments in aural pattern recognition. We believe that experiments directed at distinguishing capabilities with respect to the subregular hierarchy will provide a great deal of evidence about the distinctions between the cognitive mechanisms of humans and those of other species.

In the remainder of this extended abstract we summarize these points.

2 The Subregular Hierarchy

The subregular hierarchy is actually a hierarchy of infinite hierarchies some of which can be further refined into sub-hierarchies. We concentrate on the the main classes, each of which is the closure of an infinite hierarchy of subclasses.

2.1 Strictly Local Stringsets

The base of the hierarchy is the class of *Strictly Local* (SL) stringsets, those that can be distinguished simply on the basis of which symbols occur adjacently. A Strictly k-Local description is a set of k-factors (or k-grams), length k sequences of symbols drawn from the alphabet augmented with start and end symbols. The stringset such a description defines is the set of strings which include only k-factors from the set. The set $\{(\textbf{ding dong})^n \mid n \geq 1\}$ is SL_2.

Computationally, such Strictly k-Local stringsets are those that can be recognized by *scanners*, automata that simply scan a k-symbol window across the input failing if, at any point, the window contains a factor that is not in the permitted set. A grammar-theoretic characterization can be obtained by taking the k-factors to be productions permitting the extension of a string with the k^{th} symbol of the factor if the string currently ends with the first $k-1$ symbols.

The Strictly k-Local stringsets form a proper hierarchy in k, with the class of Strictly Local stringsets, in general, being the union of the Strictly k-Local stringsets for all finite k. From a cognitive perspective, this is the class of stringsets that can recognized while only remembering the previously encountered block of symbols for blocks of some finitely bounded size or, more generally, that can be distinguished solely on the basis of whether particular finitely bounded blocks of symbols do not occur.

The characteristic property of SL stringsets is that if there are two strings in the set in which the same sequence of $k-1$ symbols occurs, then the result of substituting the suffix starting at that sequence in one of them for the suffix starting at that sequence in the other must also be in the stringset. Note that this is a *characterization*: a stringset can be recognized while remembering only the preceding $k-1$ symbols if and only if it is closed under this type of substitution of suffixes.

One of the consequences of this characterization is that SL specifications cannot require some particular symbol to occur in every string unless the k-factors that may precede it are distinct from those that may follow it. For example, the set of strings of 'A's and 'B's in which at least one 'B' occurs is not SL (for any k). We will refer to this set as "Some-B".

An animal that used a strategy to recognize strings that was based only on remembering the most recently encountered fixed-length sequence of symbols could potentially distinguish strings of the form $(AB)^n$ from those, for instance, of the same form in which one or more of the 'B's was replaced with an 'A', but would not be able to distinguish strings of the form A^nBA^m from strings of only 'A's.

The failure of an animal to recognize a particular SL stringset, however, does not imply that they do not employ this strategy. Their ability to apply the strategy may be limited in ways that exclude that set. Similarly, while success in recognizing a SL stringset establishes that the subject employs a strategy that is at least as powerful as scanning k-factors, it does not imply that they are capable of recognizing every SL stringset. So we can establish lower bounds experimentally (individuals of this species can recognize at least some SL stringsets) but experiments of this sort will provide firm evidence only of possible upper bounds (these individuals fail to recognize at least some SL stringsets). The strength of the evidence depends on the preponderance of negative results and the breadth of variation in the experimental setting.

2.2 Locally Testable Stringsets

While SL descriptions cannot require some factor to occur, they can forbid it. So Some-B *is* in co-SL. Since the class of Strictly Local stringsets is closed under intersection, if we close SL_k under complement we get closure under all Boolean operations. Descriptions of stringsets in this class can be taken to be formulae in a propositional language in which the propositional variables are k-factors which are taken to be true for a string iff they occur in that string. Stringsets definable in this way are called *Locally k-Testable* (LT_k). Again, these classes form a proper hierarchy in k with the class LT being the closure of the hierarchy under union. These formulae are just expressions of which k-factors occur and do not occur in a string and which employ a truth-functionally complete set of logical connectives. Any pattern that can be described in these terms defines a Locally Testable stringset.

Automata for recognizing LT_k stringsets can be obtained by extending scanners to keep track of which k-factors occur, feeding the result into an arbitrary Boolean network. Since there are but finitely many k-factors over a fixed alphabet this is a finite-state recognition strategy.

The characteristic of Locally k-Testable sets is that strings that include exactly the same set of k-factors cannot be distinguished—they either both must be included in the set or they must both be excluded. One of the consequences of this is that, while Some-B is Locally Testable, the set of strings of 'A's and 'B's in which *exactly* one 'B' occurs (we'll call this "One-B") is not Locally Testable (for any k): the strings A^kBA^k and $A^kBA^kBA^k$ include exactly the same sets of k-factors, but the first is in One-B while the second is not.

So, an animal that used a strategy based on keeping track of exactly which subset of the set of all k-factors over the relevant alphabet occurs in a string (or which was otherwise sensitive to only this information) could potentially distinguish strings of 'A's and 'B's in which some 'B' occurs from those in which no 'B' occurs, but would not be able to distinguish strings in which exactly one 'B' occurs from those in which more than one occurs.

2.3 Locally Threshold Testable

A stronger strategy would be to keep track not just of which k-factors occur but *how many* times each occurs. If one limits this to counting up to a finite threshold (not distinguishing different numbers of occurrences if they both exceed the threshold) then this will still be finite-state. This would allow recognition of "n-B" for any value of n less than the threshold. Stringsets that can be distinguished in this way are called *Locally Threshold Testable* (LTT) and there are proper hierarchies in both k and t, the size of the threshold.

Strikingly, this turns out to be exactly the class of stringsets one can define using First-Order logic to reason about positions in strings using a monadic predicate for each symbol (picking out the set of positions in which it occurs) and a binary predicate relating positions to their immediate successor. LTT definitions are just finite sets of FO formulae over this signature. Again, any pattern that can be described in this way defines an LTT stringset.

The characteristic of LTT stringsets is analogous to that of LT stringsets. We say that two strings are (k,t)-equivalent whenever, for each k-factor, they include either the same number of occurrences of that k-factor or they both include at least t occurrences. A stringset is LTT if and only if there is some k and t for which it does not distinguish (k,t)-equivalent strings: every pair of equivalent strings are either both included in the set or both excluded.

One could probe the limits of LTT by exploring a range of values of k and t, but it is doubtful that a sufficiently large range could be covered in a practical set of experiments. A better approach would be to

explore the ability of animals to recognize the set we will call "B-before-C". This is the set of strings over 'A', 'B' and 'C' in which at least one occurrence of 'B' occurs before any occurrence of 'C'. To see that this is not LTT for any k or t, it suffices to note that the strings $A^kBA^kCA^k$ and $A^kCA^kBA^k$ have exactly the same number of occurrences of every k-factor and therefore are (k,t)-equivalent for all t.

An animal that used a strategy of counting k-factors up to some threshold (or which was otherwise sensitive to only this information), then, could potentially recognize One-B, but would fail to recognize B-before-C.

2.4 Star-Free Stringsets

The next step is to extend the FO signature with a predicate for the transitive closure of the successor relation ("precedes" or "less-than"). Stringsets FO-definable over this signature are called *Star-Free* (SF).[1] Any pattern that can be described by a finite set of FO formulae over this signature defines a Star-Free language.

In abstract language-theoretic terms, SF turns out to be the closure of LT under concatenation and Boolean operations. This class is called *Locally Testable with Order* (LTO). From a cognitive perspective it corresponds to using a fixed sequence of threshold-counting strategies—equivalently allowing the threshold counters to be reset up to a fixed, finitely bounded number of times—and keeping track of which succeed and which fail. Automata-theoretically, SF is the class of stringsets recognized by *Counter-Free* automata, automata for which the syntactic monoid is aperiodic. The key consequence of this is that the automata cannot do modular counting. Membership in SF stringsets cannot depend on the number of times some factor occurs modulo some fixed value. (In other words, the threshold counters cannot be reset arbitrarily many times.) One simple stringset that requires modular counting is the set of strings of 'A's and 'B's in which the number of 'B's is even ("Even-B").

So, while an animal using a sequence of LT strategies could potentially recognize B-before-C they would be unable to recognize Even-B.

2.5 Regular Stringsets and Beyond

Finally, if we move to a Monadic Second-Order language, allowing quantification over subsets of positions rather than just individual positions (quantification over finite subsets, wMSO, actually suffices), using either signature (since transitive closure is MSO definable), we obtain the Regular stringsets. This characterization of Regular stringsets dates back to the late '50's, due to Büchi [Büc60], Elgot [Elg61] and Medvedev [Med64]. Cognitively, these are the Finite-State stringsets: any mechanism for which there is a finite bound on the amount of information that is retained while processing a string recognizes, at most, a regular stringset.

What CFLs introduce is a need, in principle, to retain an amount of information that is proportional to the length of the string, hence not finitely bounded. This can be captured in a descriptive setting by reasoning about additional structure beyond the linear ordering of symbols in the string. The grammar-theoretic characterization that gives the class its name generates strings, in effect, by tracing out one sort of additional structure. It can be captured in automata-theoretic terms, of course, by augmenting finite-state automata with a stack, or some other store with a size that is not finitely bounded. It should noted, however, that evidence that a subject employs a strategy that is not finite-state does not imply that the mechanism they are using actually has access to a store of unbounded size. It may be the case that the mechanism implements a strategy which, in general, requires unbounded storage but which fails on strings beyond a particular length or degree of complexity or, rather, which would fail if it ever encountered such strings.

Conclusion

Formal language theory provides useful tools for exploring pattern recognition capabilities in general and, in particular, for designing and interpreting experiments for distinguishing differential capabilities of this sort

[1] These are also the stringsets definable by *star-free* expressions, regular expressions which may employ complement but not Kleene-closure.

across species. Paradoxically, the machine-oriented aspects—grammar- and automata-theory—have little to say about the cognitive mechanisms involved. Descriptive characterizations, in contrast, because they focus on the kind of information that distinguishes structures in a class, can provide clues about the cognitive mechanisms under study that are independent of a priori assumptions about the details of the physical implementation of those mechanisms. What the grammar- and automata-theoretic characterizations do provide is highly specific information about the nature of stringsets that can or cannot be recognized by these means, providing clear criteria on which to choose sets of stimuli to test the boundaries of the class. Together, these provide a sound foundation for studying classes of stringsets that might be relevant to the experimental study of prerequisites to language.

The subregular hierarchy encompasses a range of extremely general ways of defining classes of patterns which are well understood from both a descriptive and machine-oriented perspective. The prevailing focus on the boundary between the finite-state and the context-free seems to have been taken over from the work of Chomsky in 1956. But he was concerned with the capacities of human beings, who we already knew spoke complex languages. With tamarins and starlings we are fairly sure that no developed language capability is in place. What we seek is a glimpse of any cognitive capacity for syntactic pattern recognition they might have that could serve as an evolutionary building block for linguistic capacities. As yet we do not even know whether an animal could recognize a pattern that is LT but not SL. The range of complexity classes that make up the subregular hierarchy seems to span a great deal of the territory that is likely to be relevant to distinguishing human aural pattern recognition capabilities from those of other species. This seems to be a particularly promising range of classes on which to focus.

References

[BS05] Michael Benedikt and Luc Segoufin. Regular tree languages definable in FO. In Volker Diekert and Bruno Durand, editors, *22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS 2005)*, volume 3404 of *Lecture Notes in Computer Science*, pages 327–339, 2005.

[Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[Elg61] Calvin C. Elgot. Decision problems of finite automata and related arithmetics. *Transactions of the American Mathematical Society*, 98:21–51, 1961.

[FH04] W. Tecumseh Fitch and Marc D. Hauser. Computational constraints on syntactic processing in nonhuman primates. *Science*, 303:377–380, 2004.

[GFMN06] Timothy Q. Gentner, Kimberly M. Fenn, Daniel Margoliash, and Howard C. Nusbaum. Recursive syntactic pattern learning by songbirds. *Nature*, 440:1204–1207, 2006.

[GR90] Pedro García and José Ruiz. Inference of k-testable languages in the strict sense and applications to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:920–925, 1990.

[HCF02] Marc D. Hauser, Noam Chomsky, and W. Tecumseh Fitch. The faculty of language: What is it, who has it, and how did it evolve. *Science*, 298:1569–1579, 2002.

[Med64] Yu. T. Medvedev. On the class of events representable in a finite automaton. In Edward F. Moore, editor, *Sequential Machines—Selected Papers*, pages 215–227. Addison-Wesley, 1964. Originally in Russian in Avtomaty (1956), pp. 385–401.

[MP71] Robert McNaughton and Seymour Papert. *Counter-Free Automata*. MIT Press, Cambridge, MA, 1971.

[PR05] Pierre Perruchet and Arnuad Rey. Does the mastery of center-embedded linguistic structures distinguish humans from nonhuman primates? *Psychonomic Bulletin and Review*, 12:307–313, 2005.

[Tho82] Wolfgang Thomas. Classifying regular events in symbolic logic. *Journal of Computer and Systems Sciences*, 25:360–376, 1982.