Carib.J.Sci.Tech

A REVIEW OF APPLICATIONS OF THEORY OF COMPUTATION AND AUTOMATA TO MUSIC

Authors & Affiliation:

Agbaje M Olugbenga and Y A Adekunle,

Babcock University, Nigera.

ISSN 0799-3757

http://caribjscitech.com/

1.0 INTRODUCTION

Theory of Computation and Automata is a theoretical branch of computer science. It established its roots during 20th Century when mathematicians began developing theoretically and literally machines which mimic certain features of man, completing calculations more quickly and reliably. The word automaton is closely related to the word "automation", meaning automatic processes carrying out the production of specific processes. Automata theory deals with the logic of computation with respect to simple machines, referred to as automata. Through automata, computer scientists are able to understand how machines compute functions and solve problems and more importantly, what it means for a function to be defined as computable or for a question to be described as decidable (Stanford(2004),Cristopher(2013))

Automatons are abstract models of machines that perform computations on an input by moving through a series of states or configurations. At each state of the computation, a transition function determines the next configuration on the basis of a finite portion of the present configuration. As a result, once the computation reaches an accepting configuration, it accepts that input. The most general and powerful automata is the Turing machine(Stanford,2004)..

The major objective of automata theory is to develop methods by which computer scientists can describe and analyze the dynamic behavior of discrete systems, in which signals are sampled periodically. The behavior of these discrete systems is determined by the way that the system is constructed from storage and combinational elements. Characteristics of such machines include:

- **Inputs**: assumed to be sequences of symbols selected from a finite set *I* of input signals. Namely, set *I* is the set $\{x_1, x_{,2}, x_3... x_k\}$ where *k* is the number of inputs.
- **Outputs**: sequences of symbols selected from a finite set Z. Namely, set *Z* is the set $\{y_1, y_2, y_3 ... y_m\}$ where *m* is the number of outputs.

  States: finite set *Q*, whose definition depends on the type of automaton.

Theory of computation course is core in computer science ACM curriculum. It has not gained much existence in other branches viz. Information systems, Software engineering, Information Technology in ACM Curriculum. Automata Theory is important because it is used for modelling interesting kinds of hardware and software. Hardware designers have understood its importance but it's an irony that software developers are unaffected with the immense power of this subject (Mukta et al, 2009).

1.1 Benefits of Computation Theory

David (2010) highlights speed and correctness as one of the benefits of computation.

1. Speed: when programmers work on a large programming project, writing a build script, which quickly handles all the tedious steps that must be taken to create a program from its source code.
2. Correctness: The build script does not make the mistakes that every programmer eventually makes.

3. Computational devices are all over the places and several products have computational devices in them for example TV, vending machines, and mobile phones.

4. Though models and forms of computation may differ the essential underlying idea is the same for example a vending machine may dispense soft drinks, train tickets and so on.

5.Theory of computation helps to develop an understanding of what computation is, what are its limit, what can be done and cannot be done and how much resources is needed.

1.2 Why is computation beneficial to music?

A music piece consists of notes. A note is a single sound, represented by pitch, i.e., how high or low the sound is, and duration, i.e., how long the sound is held. Other properties of a note include its volume (loudness) and timbre (the influence of the instrument that the note is played on the sound).Jan(2011). A long standing debate has been on for generations among philosophers, psycholinguists, semiologists and musicologists as to whether music is a language or not. What is clear beyond this battle of words is that there exists striking similarities between the two, running throughout all their respective levels of description. The obvious similarity is that both are conveyed by acoustic signals,i.e. variations of air pressure propagating through space and unfolding in time Colin(2005).

Linguistics and automata theory were at one time tightly knit (Kevin et al, 2007). Musical notes can be represented as strings therefore theory of automata can be applied to music for both music identification and music style recognition.

Mehryar(2009) describe the use of factor automata of automata in music. They presented the results of experiments applying factor automata to a music identification task with more than 15,000 songs. When the number of strings is large, such as thousands or even millions, the collection of strings can be compactly stored as an automaton, which also enables efficient implementations of search and other algorithms. In fact, in many contexts such as speech recognition or information extraction tasks, the entire set of strings is often directly given as an automaton. An efficient and compact data structure for representing a full index of a set of strings is a suffix automaton, a minimal deterministic automaton representing the set of all suffixes of a set of strings. A string is an ordered sequence of symbols drawn from a finite vocabulary.

Suffix automata and factor automata are efficient data structures for representing the full index of a set of strings. They are minimal deterministic automata representing the set of all
suffixes or substrings of a set of strings(Mehryar et al,2009).

2.0 A review of Automata Theory

Automata theory is the basis for the theory of formal languages. A proper treatment of formal language theory begins with some basic definitions:

A symbol is simply a character, an abstraction that is meaningless by itself.

An alphabet is a finite set of symbols.

A word is a finite string of symbols from a given alphabet.

A language is a set of words formed from a given alphabet.

The set of words that form a language is usually infinite, although it may be finite or empty as well. Formal languages are treated like mathematical sets, so they can undergo standard set theory operations such as union and intersection. Additionally, operating on languages always produces a language. As sets, they are defined and classified using techniques of automata theory.(Stanford,2004 ;Pedro,2006).

Formal languages are normally defined in one of three ways, all of which can be described by automata theory:

i.regular expressions

ii. standard automata

iii. a formal grammar system

There are four major families of automaton:

- Finite-state machine
- Pushdown automata
- Linear-bounded automata
- Turing machine

The families of automata above can be interpreted in a hierarchal form, where the finite-state machine is the simplest automata and the Turing machine is the most complex. The focus of this project is on the finite-state machine.

2.1 Automata Overview

The transducer representation of our music collection, and also devices of interest are weighted automata and transducers.

Factors of a finite automaton

We denote by ∑ a finite alphabet. The length of a string x ϵ ∑* over that alphabet is denoted by /x/. A factor, or substring, of a string x ϵ∑* is a sequence of symbols appearing consecutively in x. Thus, y is a factor of x iff there exist u; v ϵ∑ such that x = uyv. A suffix of a string x ϵ∑* is a factor that appears at the end of x. Put otherwise, y is a suffix of x iff there exists u ϵ∑ such that x = uy. applications such as music identification the strings considered may be long, e.g., sequences of music sounds, but with relatively short common suffixes. This motivates the following definition.

Suffix automata

An efficient and compact data structure for representing a full index of a set of strings is a suffix automaton, a minimal deterministic automaton representing the set of all suffixes of a set of strings. Since a substring is a prefix of a suffix, a suffix automaton can be used to determine if a string x is a substring in time linear in its length O./x/, which is optimal.
Additionally, as with suffix trees, suffix automata have other interesting properties in string-matching problems which make their use and construction attractive. (Mehryar, 2009).

Factor automata

Another similar data structure for representing a full index of a set of strings is a factor automaton, a minimal deterministic automaton representing the set of all factors or substrings of a set of strings. Factor automata offer the same optimal linear-time search property as suffix automata, and are never larger. The construction and the size of a factor automaton have been specifically analyzed in the case of a single string.

Weighted automata

A weighted automaton is specified by an alphabet, a finite set of states, a finite set of transitions, an initial state, a set of final states, and a final weight function. Each transition associates pairs of states with a symbol and a weight. A weighted finite-state transducer is specified by input and output alphabets, a finite set of states, a finite set of transitions, an initial state, a set of final states, and a final weight function. Each transition associates pairs of states with an input symbol, an output symbol, and a weight. (Mehryar, 2008).

Applications of Weighted String Automata

Probabilistic finite-state machines are used today in a variety of areas in pattern recognition, or in fields to which pattern recognition is linked: computational linguistics, machine learning, time series analysis, circuit testing, computational biology, speech recognition and machine translation are some of them (Kevin 2011).

Finite State Machines (FSM)

The first people to consider the concept of a finite-state machine included a team of biologists, psychologists, mathematicians, engineers and some of the first computer scientists. They all had a common interest to model the human thought process, whether in the brain or in a computer. Probabilistic finite-state machines such as probabilistic finite-state automata (PFA) ,hidden Markov models (HMMs) , stochastic regular grammars , Markov chains, *n*-grams, probabilistic suffix trees, deterministic stochastic or probabilistic automata (DPFA) , weighted automata are some names of syntactic objects which during the past years have attempted to model and generate distributions over sets of possible infinite cardinality of strings, sequences, words, phrases but also terms and trees.

Finite-state acceptor (FSA)

A finite-state acceptor (FSA) is a network of states and labelled transitions, with exactly one start state and one final state. A string is an ordered sequence of symbols drawn from a finite vocabulary. An FSA accepts string w1;w2; : : : ;wn if there is a path from the start state to the final state along transitions labelled w1; w2; : : : ;wn. The empty symbol is also a valid transition label, denoting that no new symbol is read from the string. An acceptor with more that one final state can be transformed to have only one final state by adding transitions on empty symbols from the old final states to a single new final state(Jan,2011).

Finite state transducer (FST)

A finite-state transducer (FST) is similar to an FSA, but each of its transitions has an input label and an output label. An FST therefore transforms an accepted input string into an output string.

Weighted finite-state acceptor (WFSA)

A weighted finite-state acceptor (WFSA) assigns a weight to each string that it accepts. Every transition is assigned a weight, and the weight of the string is the product of the transition probabilities along the path by which the string is accepted. Similarly, a weighted finite-state transducer (WFST) is an FST with probabilities assigned to the transitions. A weighted finite-state acceptor can be represented as a WFST with the same input and output symbols on every transition.

Stochastic finite automata to describe music

Probabilistic finite-state automata , hidden Markov models , n-grams , probabilistic suffix trees , deterministic stochastic or probabilistic automata , weighted automata are variants of finite state machines used to model and generate distributions over sets of strings of possibly infinite cardinality, sequences, words, phrases but also terms and trees. These models have been successful in a wide amount of fields ranging from computational linguistics to pattern recognition, and including language modelling in speech recognition , bioinformatics or machine translation.

Let $\sum$ be a finite alphabet and $\sum^*$ the set of all strings that can be built from $\sum$ including the empty string denoted by λ. A language is a subset of $\sum^*$. By convention, symbols in $\sum$ will be denoted by letters from the beginning of the alphabet (a, b, c, ...)

and strings in ∑* will be denoted by end of alphabet letters (..., x, y, z). A stochastic language D is a probability distribution over ∑*.

We denote by PrD(x) the probability of a string x϶∑* under the distribution D; it must verify ∑x϶∑* PrD(x) = 1.

If the distribution is modelled by some syntactic machine A, the probability of x according to the probability distribution defined by A is denoted PrA(x). The distribution modeled by a machine A will be denoted DA and simplified to D in a non ambiguous context.

If L is a language over ∑ and D is a distribution over ∑*,

PrD(L) =∑Px϶ L PrD(x)

3.0 Applications of automata to music

1. Large-scale music identification

Automatic identification of music has been the subject of several recent studies both in research and industry. Given a test recording of a few seconds, music identification systems attempt to find the matching reference recording in a large database of songs. This technology has numerous applications, including end-user content based search, radio broadcast monitoring by recording labels, and copyrighted material detection by audio and video content providers such as Google YouTube (Mehryar, 2008).

Music Identification With weighted Finite-State Transducers(WFST)

In this music identification approach Mehryar(2008) represents each song is by a distinct sequence of music sounds, called music phonemes in our work. The system learns the set of music phonemes automatically from training data using an unsupervised algorithm.

The unique sequence of music phonemes characterizing each song is also learned. The music identification problem is then reduced to a mapping of music phoneme sequences to songs. As in a speech recognition system, this mapping can be represented compactly with a finite-state transducer. Specifically, a test audio snippet can be decoded into a music phoneme sequence using the Viterbi beam search algorithm. The transducer associates a weight to each pairing of a phoneme sequence with a song, and the search process approximates the most likely path through the transducer given the acoustic evidence (Mehryar, 2008).

2. Music style recognition

Musical style classification could allow a user-centered, more sensible indexing of songs than the usual statically pre-defined categories such as “pop”, “rock”, “folk”, etc... The concept of musical style or genre is ill-defined because of the subjectivity involved, and it can be based on many different aspects of the music: sound or instrument (“piano, distorted guitar”), period (“Baroque, 60s”), dance type (“Salsa”)... We refer here to musical style as it relates to the tonal (choice of notes), compositional, structural aspect of a piece. This ignores the acoustic level, the sound quality (or timbre) characterizing each instrument, even though this could be very relevant to other classification strategies.

Learning Stochastic Finite Automata for Musical Style Recognition

Music can be encoded in a simple way through the pitch and the length of the notes of a melody. A melody will therefore be a string and from a set of melodies or strings it will be possible to infer a grammatical representation of the language corresponding to the musical style. The languages can be represented by stochastic deterministic finite automata (SDFA).

Stochastic deterministic finite automata have been introduced and are used in a variety of settings. They are used to model musical styles. The same automaton can be used to classify new melodies but also to generate them (Colin, 2006).

Symbolic encoding of music allows a melody to simply translate into a string of letters each representing a note. The structural regularities of a given musical style being embedded in such sequences over many tunes, it is not farfetched to consider a set of tunes or strings as being a language and the melodic regularities as corresponding to those of regular languages. Therefore, automata appear as being good candidates for their extraction.

The advantage of symbolic encoding is twofold.

1. First, it reduces drastically the amount of data to be processed, thereby permitting the use of techniques that would not be able to handle millions of data.
2. Second, it seems a necessary step to access the semantic level, which is important if we aim at a deeper understanding of music and care about the explanation capabilities of the models.

3. Music style classification.

For music style classification one automaton is built for each style to be recognized, a style being defined by a set of strings. Then states of this automaton are iteratively merged when considered similar or when some entropy measure indicates that the intended merge is valid; this process results in a much smaller SDFA which generalizes the learning sample.

To classify a new melody into one style, parse the corresponding string through each of the SDFA. It is then assigned to the style corresponding to the Sdfa yielding the highest parsing probability. Smoothing techniques are developed in order to cope with parsing failures by an SDFA Colin(2006).

The melody code alternates pitch and duration symbols that constitute two separate alphabets. Therefore in the original automaton generated, transitions are always between two states of different types. However after the state merging step, this rule does not hold anymore, since differently typed states have been merged. Nevertheless, it is possible to type the automaton, that is, assign a type to each state in order to allow merges only between same-type states.

Other Applications of theory of computation and automata.

1. Language Translation

This model was implemented in a WFST (weighted finite-state acceptors and transducers) framework. Translations from this model are much more accurate, and by using a WFST toolkit are able to build a cascade of transducers and execute translations using generic finite-state procedures.

Speech Recognition

The noisy-channel framework to the problem of speech recognition,i.e. recovering the sequence of spoken words that generated a given acoustic speech signal.

Lexical Processing

In most natural language applications, it is necessary to cut an information stream into word units. This is especially hard in languages without white- space, such as Chinese. [51] show how to automatically break Chinese into words by constructing a series of WFSTs. Word-internal units must also be processed.

Tagging

A wide variety of natural language problems can be cast as tagging problems, in which each word of input is assigned a tag from some finite set. The classic example is part-of-speech tagging, which seeks to disambiguate the syntactic category of each word in a sentence.

Summarization

Text summarization is the shrinking of a document or set of documents into a short summary that contains a useful subset of the information. One application of summarization, headline generation, drops unnecessary words from an input text and performs limited transformation of the remaining words to form an appropriate news headline.

Optical Character Recognition

The automatic conversion of hard-copy printed material to electronic form is useful for preserving documents created before the digital age, as well as for digitizing writing that is still generated in a non-digital manner, e.g. converting hand-written notes. Scanner technology has progressed considerably in recent years thanks to probabilistic recognition techniques, which are represent able in the noisy channel framework.

Many other branches of science also involve unbelievable levels of complexity, impossibly large degrees of variation, and apparently random processes, so it makes sense that automata theory can contribute to a better scientific understanding of these areas as well. The modern-day pioneer of cellular automata applications is Stephen Wolfram, who argues that the entire universe might eventually be describable as a machine with finite sets of states and rules and a single initial condition. He relates automata theory to a wide variety of scientific pursuits, including:

- Fluid Flow
- Snowflake and crystal formation
- Chaos theory
- Cosmology
- Financial analysis(Stanford,2004)

4.0 CONCLUSION AND FUTURE WORKS

A long standing debate has been animating generations of philosophers, psycholinguists, semiologists and musicologists as to whether music is a language or not. What is clear beyond this battle of words is that there exists striking similarities between the two, running throughout all their respective levels of description. The obvious similarity is that both are conveyed by acoustic signals,i.e. variations of air pressure propagating through space and unfolding in time. The amplitude of these variations can be coded numerically after quantization of the temporal dimension, allowing numerical recordings. In music, notes are grouped (played sequentially or simultaneously in the case of chords) to instantiate classes (e.g. the sequence of notes C-E-G-C instantiates the C Major scale because it contains its more important notes, a scale being an ordered subset of the 12 notes used in western music). Further, there exists rules of well-formedness and typical sequences in the structure of musical passages, which vary according to musical styles, just like a verb can be in the middle or at the end of a sentence depending on the language.The above considerations suggest that tools used for language modeling such as formal language theory and grammatical inference could be of great help for the analysis and understanding of music(Colin,2006).

5.0 REFERENCES

1. Learning Stochastic Finite Automata for Musical Style Recognition Colin de la Higuera, Fr´ed´eric Piat and Fr´ed´eric Tantini(2005).
2. Mukta Goyal, Shelly Sachdeva(2009):International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009,Enhancing Theory of Computation TeachingThrough Integration with other Courses
3. Applications of the theory of computation to nanoscale self-assembly By David Samuel Doty(2010), Phd thesis. Iowa State University
4. General suffix automaton construction algorithm and space bounds Mehryar Mohria,b, Pedro Morenob, Eugene Weinstein,Theoretical Computer Sc,Elsivier(2009).
5. C. Allauzen, M. Mohri, M. Saraclar, General indexation of weighted automata _ Application to spoken utterance retrieval, in: Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval (HLT/NAACL), Boston, Massachusetts, 2004, pp. 33_40.

6. E. Weinstein, P. Moreno, Music identification with weighted finite-state transducers, in: Proceedings of the International Conference on Acoustics,
7. applications of automata theory ,Stanford university, sophomore college,2004
8. A study of Grammatical Inference Algorithms in Automatic Music Composition and Musical Style Recognition. Pedro P. Cruz-Alcázar Enrique Vidal-Ruiz,1997.
9. Algorithm on strings (2006), Maxime Crochemore Christophe Hancart Thierry Lecroq
10. Factor Automata of Automata and Applications ,Mehryar Mohri1,2, Pedro Moreno2, Eugene Weinstein, New York University
11. IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. X, NO. Y, JANUARY 2008 Efficient and Robust Music Identification with Weighted Finite-State Transducers ,Mehryar Mohri, Pedro Moreno, and Eugene Weinstein
12. Jan Buys(2011): Generative Models of Music for Style Imitation and Composer Recognition
13. Applications of Weighted Automata in Natural Language Processing Kevin Knight1 and Jonathan May,2007.
14. IEEE TRANSACTION PAMI (2004) Probabilistic Finite-State Machines – Part I E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta and R. C. Carrasco
15. Cristopher Moore(2013): Lecture notes on automata, languages, and grammars, January 11, 2013