Automata Theory

home | basics | the firing squad problem | applications | references

linguistics
biology
other applications

Applications of Automata Theory

Linguistics

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.
- Finally, 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.

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

- regular expressions
- standard automata
- a formal grammar system

Regular Expressions Example

alphabet A1 = {a, b}
alphabet A2 = {1, 2}
language L1 = the set of all words over A1 = {a, aab, ...}
language L2 = the set of all words over A2 = {2, 11221, ...}
language L3 = L1 ∪ L2

language L4 = {an | n is even} = {aa, aaaa, ...}

language L5 = {anbn | n is natural} = {ab, aabb, ...}

Languages can also be defined by any kind of **automaton**, like a Turing Machine. In general, any automata or machine M operating on an alphabet A can produce a perfectly valid language L. The system could be represented by a bounded Turing Machine tape, for example, with each cell representing a word. After the instructions halt, any word with value **1** (or **ON**) is accepted and becomes part of the generated language. From this idea, one can defne the complexity of a language, which can be classified as **P or NP**, **exponential**, or **probabilistic**, for example.

Noam Chomsky extended the automata theory idea of complexity hierarchy to a **formal language hierarchy**, which led to the concept of formal grammar. A formal grammar system is a kind of automata specifically defined for linguistic purposes. The parameters of formal grammar are generally defined as:

- a set of non-terminal symbols N
- a set of terminal symbols Σ
- a set of production rules P
- a start symbol S

Grammar Example

start symbol = S

non-terminals = {S}
terminals = {a, b}
production rules: S → aSb, S → ba

S → aSb → abab
S â†’ aSb → aaSbb → aababb

L = {abab, aababb, ...}

As in purely mathematical automata, grammar automata can produce a **wide variety** of **complex languages** from only a few symbols and a few production rules. Chomsky's hierarchy defines **four nested classes** of languages, where the more precise aclasses have stricter limitations on their grammatical production rules.

The formality of automata theory can be applied to the analysis and manipulation of **actual human language** as well as the development of **human-computer interaction** (HCI) and **artificial intelligence** (AI).

[top]

Biology

To the casual observer, biology is an impossibly complex science. Traditionally, the intricacy and variation found in life science has been attributed to the notion of **natural selection**. Species become "intentionally" complex because it increases their chance for survival. For example, a camoflauge-patterned toad will have a far lower risk of being eaten by a python than a frog colored entirely in orange. This idea makes sense, but automata theory offers a simpler and more logical explanation, one that relies not on random, optimizing mutations but on a **simple set of rules**.

Basic automata theory shows that **simplicity can naturally generate complexity**. Apparent randomness in a system results only from **inherent complexities** in the behavior of automata, and seemingly endless variations in outcome are only the products of **different initial states**. A simple mathematical example of this notion is found in **irrational numbers**. The square root of nine is just 3, but the square root of ten has no definable characteristics. One could compute the decimal digits for the lifetime of the universe and never find any kind of recurring patter or orderly progression; instead, the sequence of numbers seemse utterly random. Similar results are found in simple **two-dimensional cellular automaton**. These structures form gaskets and fractals that sometimes appear orderly and geometric, but can resemble random noise without adding any states or instructions to the set of production rules.

The most classic merging of automata theory and biology is John Conway's **Game of Life**. "Life" is probably the most frequently written program in elementary computer science. The basic structure of Life is a two-dimensional cellular automaton that is given a start state of any number of filled cells. Each time step, or **generation**, switches cells on or off depending on the state of the cells that surround it. The rules are defined as follows:

- All eight of the cells surrounding the current one are checked to see if they are on or not.
- Any cells that are on are counted, and this count is then used to determine what will happen to the current cell:
  1. **Death**: if the count is less than 2 or greater than 3, the current cell is switched off.
  2. **Survival**: if (a) the count is exactly 2, or (b) the count is exactly 3 and the current cell is on, the current cell is left unchanged.
  3. **Birth**: if the current cell is off and the count is exactly 3, the current cell is switched on.

Like any manifestation of automata theory, the Game of LIfe can be defined using extremely simple and concise rules, but can produce incredibly complex and intricate patterns.

In addition to the species-level complexity illustrated by the Game of Life, complexity within an **individual organism** can also be explained using automata theory. An organism might be complex in its full form, but examining constituent parts reveals consistency, symmetry, and patterns. Simple organisms, like maple leaves and star fish, even suggest mathematical structure in their full form. Using ideas of automata theory as a basis for generating the wide variety of life forms we see today, it becomes easier to think that sets of mathematical rules might be responsible for the complexity we notice every day.

Inter-species observations also support the notion of automata theory instead of the specific and random optimization in natural selection. For example, there are striking similarities in patterns between very different orgranisms:

- Mollusks and pine cones grow by the Fibonacci sequence, reproducible by math.
- Leopards and snakes can have nearly identical pigmentation patterns, reproducible by two-dimensional automata.

With these ideas in mind, it is difficult not to imagine that any biolgical attribute can be simulated with abstract machines and reduced to a **more manageable** level of simplicity.

[top]

Other Applications

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**

[top]

