Automata Models of Self-Reproduction

Arthur W. Burks*

The late John Von Neumann once pointed out that in the past science has dealt mainly with problems of energy, power, force and motion. He predicted that in the future science would be much more concerned with problems of control, programming, information processing, communication, organization, and systems. Toward this end, he wished to develop a theory of automata which would cover two kinds of automata: man-made automata, such as digital and analog computers; and natural systems, such as cells, nervous systems, and brains. My talk today is about Von Neumann's theory of automata. I will attempt to relate computers to natural systems, particularly with respect to the notion of self-reproduction.

The self-reproduction of a natural system, such as a living cell, is familiar to all of you. It sounds strange to talk about the self-reproduction of a computer. Let me show you how this gap is not so large as it seems.

Think now of a digital computer which operates synchronously and which is composed entirely of switches (e.g., "and", "or", and "not") and of delay lines (which delay pulses for one time unit). This is a very poor way to build a computer, but for certain abstract purposes it is desirable to view a computer this way. Such a computer processes information if fed with cards or tape, and produces cards or tape as output. Let us now add to this computer some other kinds of primitive elements: first, an element that will sense (e.g., see or feel); second, an element which acts (e.g., an artificial hand); third, an element which will cut or separate other elements; fourth, an element which will join or solder things together; fifth, a bar or girder to provide rigid support to assemblies of these other elements. I will call a computer composed of these elements a kinematic automaton, because it is capable of motion, sensation and action, as well as of information processing. Thus a kinematic automaton has, besides the usual powers of computation, the ability to sense, act, move, take things apart, and construct things.

There are two main kinds of abstract automata: finite automata and Turing machines. A finite automaton is a device with a finite number of parts and a finite number of states. A Turing machine is a finite automaton plus a tape unit with an indefinitely expandable tape.

I will show you first how to design a universal kinematic Turing machine and then how to design a universal kinematic constructing machine.

Please use yor imagination. Imagine an infinite lake or ocean with infinitely many copies of each kind of part floating on its surface. There are switches, delays, sensing organs, acting organs, cutting organs, joining organs, and girders, all swimming around in random fashion, after the manner of the molecules of a gas. Hence an indefinite supply of parts is available to any kinematic automaton placed on the surface of the lake.

* Professor of Philosophy and of Communication Sciences, University of Michigan and Indian Institute of Technology at Kanpur.

Lecture delivered to the Information Processing Society of Japan, Tokyo, June 30, 1965.

John Von Neumann Collected Works, Volume V (Design of Computers, Theory of Automata, and Numerical Analysis). Edited by A.H. Taub. New York: Macmillan, 1963. See especially "The General and Logical Theory of Automata," pp. 288-328.

John Von Neumann, Theory of Self-Reproducing Automata. Edited by Arthur W. Burks. To be published by the University of Illinois Press. See especially the Fifth Lecture of Part I.

Now imagine that there is floating on the surface of this lake a kinematic automaton M_u* T composed of a finite machine M_u and a tape T. The matrix of tape T is a zig-zag arrangement of girders; at each intersection a protruding girder may be added (to represent a "one") or not (in which case "zero" is represented). The finite automaton M_u has the power of the finite part of a universal Turing machine: that is, it can operate the tape T and can interpret an arbitrary computer program stored on that tape. In addition, M_u has sensing, acting, joining, and cutting organs so arranged that it can pick up girders from the surface of the lake and can use these girders to change the information on the tape and to expand the tape indefinitely. Hence the complex M_u*T is a universal kinematic Turing machine.

A universal kinematic constructing machine is an extension of a universal kinematic Turing machine. For any finite machine M composed of switches, delays, sensing organs, acting organs, cutting organs, joining organs, and girders, let there be stored on the tape T a description $\mathcal{D}(M)$ of M. We will denote the finite part of the universal constructing machine by "M_c". The constructor M_c has three kinds of powers. First, it has all the powers of the universal computer M_u. Second, M_c can fead and interpret any descrption $\mathcal{D}(M)$, it can sense and pick up from the surface of the lake the parts needed for machine M, and it can construct M according to the plan $\mathcal{D}(M)$. Third, M_c can copy $\mathcal{D}(M)$ onto a new tape and attach this tape to M.

The construction of an arbitrary machine M proceeds in this way. We begin with the complex M_c*$\mathcal{D}(M)$, which consists of a tape containing the description $\mathcal{D}(M)$ attached to the constructor M_c. Construction takes place in two steps. First, M_c reads $\mathcal{D}(M)$, picks up the parts needed for M, and interconnects these parts according to the plan $\mathcal{D}(M)$. Second, M_c makes a copy of the tape $\mathcal{D}(M)$ and attaches it to M. This construction process may be symbolized.

$$M_c*\mathcal{D}(M) \rightarrow M*\mathcal{D}(M)$$

Thus the universal constructor, starting with an infinite supply of parts, can construct any arbitrary finite machine M.

One may wander why we had the constructor M_c copy the description $\mathcal{D}(M)$ and attach it to the machine M. It would be more general to have M_c copy an arbitrary tape content and attach it to M. But the former procedure leads more directly to self-reproduction than the latter procedure. Note that the constructed machine $M*\mathcal{D}(M)$ contains a description of itself. To get self-reproduction from the general scheme

$$M_c*\mathcal{D}(M) \rightarrow M*\mathcal{D}(M)$$

we merely substitute "M_c" for "M", thereby obtaining

$$M_c*\mathcal{D}(M_c) \rightarrow M_c*\mathcal{D}(M_c)$$

This is the logical scheme of self-reproduction. The constructor M_c first reads the description $\mathcal{D}(M_c)$, picks up the parts needed for M_c, and interconnects them according to the plan $\mathcal{D}(M_c)$. M_c next makes a copy of the tape $\mathcal{D}(M_c)$ and attaches it to M_c. The constructor M_c does all this without knowing that it is making a copy of itself and copying a description of itself. Nevertheless the whole process begins with one copy of $M_c*\mathcal{D}(M_c)$ and ends with two copies of this complex. This is self-reproduction.

This concludes the presentation of Von Neumann's kinematic model of self-reproduction. While this model is highly artificial, it bears some resemblance to natural self-reproduction. Consider the self-reproduction of a living cell. The tape T with description $\mathcal{D}(M)$ is analogous to the genetic strands of deoxyribonucleic acid (DNA) of the cell nucleus. The operation of copying the tape $\mathcal{D}(M)$ is analogous to the replication of DNA which takes place under the control of the enzyme DNA polymerase. Finally, the construction of M on the basis of the information $\mathcal{D}(M)$ is analogous to the construction of a new cell. Molecular biologists are now engaged in tracing out the logic of this construction process. The process involves a system

of messenger ribonucleic acid (RNA), enzymes, and ribosomes. This system assembles amino acids according to the code of DNA to make proteins (including enzymes) which direct the construction of the new cell.

Von Neumann proved later that the kinematic process of self-reproduction can be simulated in an infinite iterative system which consists of an infinite number of copies of the same finite automaton. We can imagine each finite automaton occupying a square of an infinite "chess board". Each finite automaton feeds information to its four neighbors and receives information from these four neighbors. Von Neumann showed how to simulate self-reproduction in an infinite iterative system in which each finite automaton has 29 states. He did this by designing a universal constructing automaton $\boldsymbol{M}_c$ and coding its description $\mathcal{D}(\boldsymbol{M}_c)$ in such a way that, when a finite area of the infinite iterative system is put in the state $\boldsymbol{M}_c*\mathcal{D}(\boldsymbol{M}_c)$, information will flow through the system and organize another finite area of the system into the same state $\boldsymbol{M}_c*\mathcal{D}(\boldsymbol{M}_c)$. Recently, Dr. Edgar Codd showed that self-reproduction can occur in an infinite iterative system in which each finite automaton has only eight states.

Von Neumann, Theory of Self-Reproducting Automata, op. cit., Part II.

Edgar F. Codd, Propagation, Computation, and Construction in Two-dimensional Cellular Spaces. Ph.D. Dissertation, University of Michigan, Ann Arbor, Michigan, 1965.

I will conclude with some general comments about automata self-reproduction. The self-reproducing automaton $\boldsymbol{M}_c*\mathcal{D}(\boldsymbol{M}_c)$ contains, as a proper part of itself, its own description $\mathcal{D}(\boldsymbol{M}_c)$. Indeed, the same "information" appears in $\boldsymbol{M}_c*\mathcal{D}(\boldsymbol{M}_c)$ twice: first, in the form of the logical structure of the constructor $\boldsymbol{M}_c$, and second, in the form of the description $\mathcal{D}(\boldsymbol{M}_c)$ of this constructor. Now suppose we measure the complexity of a finite area of an infinite iterative system by the logarithm of the number of states this area is capable of. For example, in Von Neumann system an area 10 cells wide and 10 cells long would have a complexity of $\log 29^{100}$.

In any given infinite iterative system there is an optimally designed self-reproducing automaton of some minimal complexity $\boldsymbol{C}$. Any automaton with a complexity of less than $\boldsymbol{C}$ cannot reproduce itself. Hence automaton self-reproduction is an informational process requiring a certain minimal complexity.