3.2 Von Neumann's Universal Constructor

The self-replication of digital logic circuits has thus far engendered a relatively modest amount of research, probably, as we mentioned, because of the technological problems associated with the implementation of such a feature. The existing approaches to the self-replication of computing systems are essentially derived from the work of John von Neumann [10], who pioneered this field of research. Unfortunately, the state of the art in the fifties restricted von Neumann's investigations to a purely theoretical level, and the work of his successors mirrored this constraint. In this section, we will analyze von Neumann's research on the subject of self-replicating computing machines, and in particular his *universal constructor*, a self-replicating cellular automaton [104].

3.2.1 Von Neumann's Self-Replicating Machines

Von Neumann, confronted with the lack of reliability of computing systems[6], turned to nature to find inspiration in the design of fault-tolerant computing machines. Natural systems are among the most reliable complex systems known to man, and their reliability is a consequence not of any particular robustness of the individual cells (or organisms), but rather of their extreme redundancy. The basic natural mechanism which provides such reliability is self-reproduction[7], both at the cellular level (where the survival of a single organism is concerned) and at the organism level (where the survival of the species is concerned).

Thus von Neumann, drawing inspiration from natural systems, attempted to develop an approach to the realization of self-replicating computing machines (which he called *artificial automata*, as opposed to *natural automata*, that is, biological organisms). In order to achieve his goal, he imagined a series of five distinct models for self-reproduction [104, p. 91-99]: the *kinematic* model, the *cellular* model, the *excitation-threshold-fatigue* model, the *continuous* model, and the *probabilistic* model.

- The kinematic model, introduced by von Neumann on the occasion of a series of five lectures given at the University of Illinois in December 1949, is the most general. It involves structural elements such as sensors, muscle-like components, joining and cutting tools, along with logic (switch) and memory elements. Concerning, as it does, physical as well as electronic components, its goal was to

---

[6] Let us remember that the computers von Neumann was familiar with were based on vacuum-tube technology, and that vacuum tubes were much more prone to failure than modern transistors. Moreover, since the writing and the execution of complex programs on such systems represented many hours (if not many days) of work, the failure of a system had important consequences in wasted time and effort.

[7] You will note that we use the terms *self-replication* and *self-reproduction* interchangeably. In reality, the two terms are not really synonyms: self-reproduction is more properly applied to the reproduction of organisms, while self-replication concerns the cellular level. As we will see, the more correct term to use in this circumstance would probably be self-replication, but since von Neumann favored self-reproduction, we will ignore the distinction.

define the bases of self-replication, but was not designed to be implemented.

- In order to find an approach to self-replication more amenable to a rigorous mathematical treatment, von Neumann, following the suggestion of the mathematician S. Ulam, developed a cellular model. This model, based on the use of cellular automata as a framework for study, was probably the closest to an actual realization. Even if it was never completed, it was further refined by von Neumann's successors and was the basis for all further research on self-replication.
- The excitation-threshold-fatigue model was based on the cellular model, but each cell of the cellular automaton was replaced by a neuron-like element. Von Neumann never defined the details of the neuron, but through a careful analysis of his work, we can deduce that it would have borne a fairly close relationship to today's artificial neural networks, with the addition of some features which would have both increased the resemblance to biological neurons and introduced the possibility of self-replication.
- For the continuous model, von Neumann planned to use differential equations to describe the process of self-reproduction. Again, we are not aware of the details of this model, but we can assume that von Neumann planned to define systems of differential equations to describe the excitation, threshold and fatigue properties of a neuron. At the implementation level, this would probably correspond to a transition from purely digital to analog circuits.
- The probabilistic model is the least well-defined of all the models. We know that von Neumann intended to introduce a kind of automaton where the transitions between states were probabilistic rather than deterministic. Such an approach would allow the introduction of mechanisms such as mutation and thus of the phenomenon of evolution in artificial automata. Once again, we cannot be sure of how von Neumann would have realized such systems, but we can assume they would have exploited some of the same tools used today by genetic algorithms.

Of all these models, the only one von Neumann developed in some detail was the cellular model. Since it was the basis for the work of his successors, it deserves to be examined more closely.

3.2.2 Von Neumann's Cellular Model

In von Neumann's work, self-reproduction is always presented as a special case of *universal construction*, that is, the capability of building any machine given its description (Fig. 3-3). This approach was maintained in the design of his cellular automaton, which it therefore much more than a self-replicating machine. The complexity of its purpose is reflected in the complexity of its structure, based on three separate components:

- A memory tape, containing the description of the machine to be built, in the form of a uni-dimensional string of elements. In the special case of self-reproduction, the memory contains a description of the universal constructor itself[8](Fig. 3-4).
- The constructor itself, a very complex machine capable of reading the memory tape and interpreting its contents.
- A constructing arm, directed by the constructor, used to build the offspring (i.e., the machine described in the memory tape). The arm moves across the space and sets the state of the elements of the offspring to the appropriate value.

The implementation as a cellular automaton is no less complex. Each element has 29 possible states, and thus, since the next state of an element depends on its current state and that of its four cardinal neighbors, 29^5=20,511,149 transition rules are required to exhaustively define its behavior. If we consider that the size of von Neumann's constructor is of the order of 100,000 elements, we can easily understand why a hardware realization of such a machine is not really feasible.

In fact, as part of the Embryonics project, we did realize a hardware implementation of a set of elements of von Neumann's automaton [12, 89]. By carefully designing the hardware structure of each element, we were able to considerably reduce the amount of memory required to host the transition rules. Nevertheless, our system remains a demonstration unit, as it consists of a few elements only, barely enough to illustrate the behavior of a tiny subset of the entire machine.

Before we continue, we should mention that von Neumann went one step further in the design of his universal constructor. If we consider the universal constructor from a biological viewpoint, we can associate the memory tape with the genome, and thus the entire constructor with a single cell (which would imply a parallel between the automaton's elements and molecules).

However, the constructor, as we have described it so far, has no functionality outside of self-reproduction. Von Neumann recognized that a self-replicating machine would require some sort of functionality to be interesting from an engineering point of view, and postulated the presence of a *universal computer* (in practice, a universal Turing machine, an automaton capable of performing any computation) alongside the universal constructor (Fig. 3-5).

Von Neumann's constructor can thus be regarded as a *unicellular* organism, containing a genome stored in the form of a memory tape, read and interpreted by the universal constructor (the mother cell) both to determine its operation and to direct the construction of a complete copy of itself (the daughter cell).

---

8. The memory of von Neumann's automaton bears a strong resemblance to the biological genome. This resemblance is even more remarkable when considering that the structure of the genome was not discovered until after the death of von Neumann.

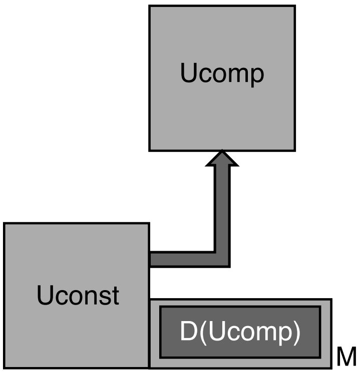

Figure 3-3: Von Neumann’s universal constructor *Uconst* can build a specimen of any machine (e.g., a universal Turing machine *Ucomp*) given its description *D(Ucomp)*.

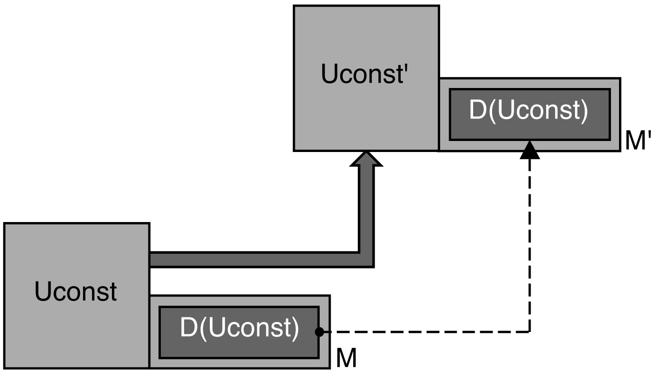

Figure 3-4: Given its own description *D(Uconst)*, von Neumann’s universal constructor is capable of self-replication.

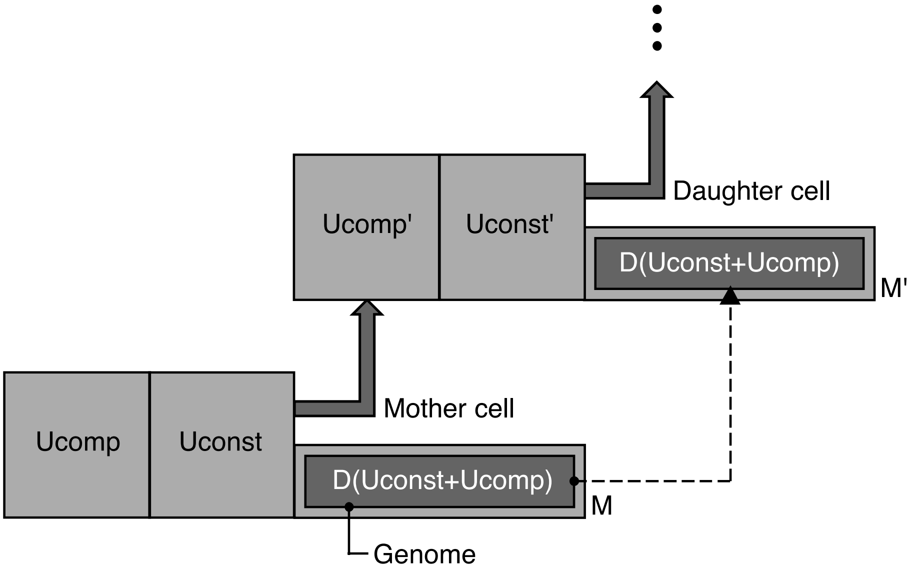

Figure 3-5: By extension, Von Neumann’s universal constructor can include a universal computer and still be capable of self-replication.

3.2.3 Von Neumann's Successors

The extreme size of von Neumann's universal constructor has so far prevented any kind of physical implementation (apart from the small demonstration unit we mentioned). But further, even the simulation of a cellular automaton of such complexity was far beyond the capability of early computer systems. Today, such a simulation is starting to be conceivable. Umberto Pesavento, a young Italian high school student, developed a simulation of von Neumann's entire universal constructor [78]. The computing power available did not allow him to simulate either the entire self-replication process (the length of the memory tape needed to describe the automaton would have required too large an array) or the Turing machine necessary to implement the universal computer, but he was able to demonstrate the full functionality of the constructor. Considering the rapid advances in computing power of modern computer systems, we can assume that a complete simulation could be envisaged within a few years.

The impossibility of achieving a physical realization did not however deter some researchers from trying to continue and improve von Neumann's work [11, 54, 71]. Arthur Burks, for example, in addition to editing von Neumann's work on self-replication [17, 104], also made several corrections and advances in the implementation of the cellular model. Codd [20], by altering the states and the transition rules, managed to simplify the constructor by a considerable degree. However, without in any way lessening these contributions, we can say that no major theoretical advance in the research on self-reproducing automata occurred until C. Langton, in 1984, opened a second stage in this field of research.

3.3 Langton's Loop

Von Neumann's Universal Constructor was so complex because it tried to implement self-reproduction as a particular case of construction universality, i.e. the capability of constructing any other automaton, given its description. C. Langton approached the problem somewhat differently, by attempting to define the simplest cellular automaton capable exclusively of self-reproduction.

As a consequence of this approach, his automaton, commonly known as *Langton's Loop* [53], is orders of magnitude simpler than von Neumann's. In fact, it is loosely based on one of the simplest organs[9] in Codd's automaton: the periodic emitter (itself derived from von Neumann's periodic pulser), a relatively simple structure capable of generating a repeating string of a given sequence of pulses.

Langton's loop (Fig. 3-6) is named after the dynamic storage of data inside a square sheath (red in the figure). The data is stored as a sequence of instructions for directing the constructing arm, coded in the form of a set of three states. The data turns counterclockwise in permanence within the sheath, creating a loop.

---

[9] An organ in this context can be seen as a self-supporting structure capable of a single sub-task.