

# Chapter 25

## The Emergence of Pathological Constructors when Implementing the Von Neumann Architecture for Self-reproduction in Tierra

Declan Baugh and Barry Mc Mullin

**Abstract** John von Neumann’s architecture for *genetic reproduction* provides an explanation in principle for how arbitrarily complex machines can construct other (“offspring”) machines of equal or even greater complexity. We designed a von Neumann style self-reproducing ancestor within the framework of the Tierra platform, which implements a (mutable) genotype-phenotype mapping during reproduction. However, we have consistently observed a particular phenomenon where what we call *pathological constructors* quickly emerge, which ultimately lead to catastrophic ecosystem collapse. Pathological constructors are creatures which rapidly construct multiple short malfunctioning offspring within their lifetime. Pathological constructors are a hindrance to an ecosystem because their offspring, although sterile, still occupy both memory space and CPU time. When several pathological constructors coincide in time, their production rate can be so high that their non-functional offspring displace the entire population of functional self-reproducing creatures, resulting in ecosystem collapse. We investigate the origin of pathological constructors, and consider how a more mutational robust architecture which is less susceptible to the emergence of these creatures can be created.

**Keywords** Von Neumann · Genetic reproduction · Tierra · Artificial life · Genotype-phenotype mapping · Evolutionary growth of complexity · Pathological constructors

### 25.1 Introduction

As early as 1948, John von Neumann had formulated his theory of the evolutionary growth of machine complexity [1, 2]. This theory provides a proof-of-principle demonstration that machines can directly, or indirectly, give rise to machines arbi-

---

D. Baugh (✉) · B. Mc Mullin  
The Rince Institute, Dublin City University, Dublin, Ireland  
e-mail: [declan.baugh2@mail.dcu.ie](mailto:declan.baugh2@mail.dcu.ie)

B. Mc Mullin  
e-mail: [barry.mcmullin@dcu.ie](mailto:barry.mcmullin@dcu.ie)

trarily more complex than themselves. This machine architecture is comprised of two specific parts, the phenotype and the genotype.

The phenotype is the functional, active section of the machine, and the genotype is the passive section, dedicated to information storage. For genetic reproduction, under some arbitrary genotype-phenotype mapping the genotype must contain an encoded description of the phenotype. Conversely, the phenotype must include the functionality to both decode the genotype and construct an offspring phenotype.

Previous work with evolutionary systems where the agents are responsible for their own self reproduction has been based exclusively on machine architectures which reproduce via *template-reproduction*, where there is no division of labour between genotype and phenotype. In this case, self reproduction is performed by self inspection, and no explicit mutable genotype-phenotype mapping is implemented.

Within the platform of Tierra, we designed an ancestor that reproduces via genetic reproduction. More importantly, this design implemented a mutable genotype-phenotype mapping as described by the von Neumann architecture, where the arbitrary mapping between genotype and phenotype is subject to heritable mutations. We aim to explore if alternative, viable mutational pathways are introduced while implementing this architecture. However, during implementation within the Tierra platform, several unanticipated phenomena emerged, which are examined and documented here.

## 25.2 Implementation of the von Neumann Architecture for Machine Self-reproduction Within Tierra

Classically, the reproductive mechanism of Tierran creatures rely on self-copying,<sup>1</sup> which involves the creatures activating a reproduction mechanism which incrementally copies the contents of each memory location of the parent creature to an available space in memory which will become the offspring. There is no distinction between phenotype and genotype for a self-copying creature as the entire creature acts as both the template for replication, and the implementation of the reproduction cycle and all other functionality.

In order to achieve von Neumann style reproduction we must first devise a method of mapping inert numbers within the genotype to active instructions to be executed within the phenotype. There exists an infinite number of possible mappings which we could implement, however, we chose to implement ours via the inclusion of a *look-up table*. The look-up table provides a method of translating the inert numbers within the memory locations of a parent genotype to functional instructions which can be executed as part of the offspring phenotype. For this particular genotype-phenotype mapping we chose a 1:1 mapping where a single number within the genotype is translated to a single instruction within the phenotype, therefore, the look-up table consists of 32 memory locations, each containing a value

---

<sup>1</sup>Analogous to RNA template replication.

which corresponds to an instruction within the Tierran instruction set. The look-up table of the seed ancestor will therefore represent a random permutation of the all possible instructions.

Prior to reproduction, this seed creature must first allocate space for an offspring. During constructing of an offspring phenotype, the parents  $Ax$  register incrementally steps through each memory location within its genotype, and the number stored at each address is inspected. A second register,  $Bx$  which initially points to the start of the look-up table, is displaced by the number which was inspected by  $Ax$ . The number within the updated  $Bx$  memory location (which lies within the look-up table), is now written to the offspring phenotype, where it will subsequently function as an instruction. This activity facilitates the mapping of numbers which are stored within the parent genotype, to instructions incorporated in the offspring phenotype. Furthermore, random perturbations within the look-up table facilitate the alteration of the genotype-phenotype mapping. This may have the effect of introducing new mutational pathways for the creature, which was not possible under the previously unaltered look-up table.

Upon construction of the offspring phenotype, the parent's genotype is incrementally copied to the offspring space and the connection between parent and offspring is severed. At this point, the parent loses write access to the offspring's memory block, and a new CPU is created and allocated to the offspring. While copying the genotype, should a random perturbation occur which affects the encoded description of the look-up table (or otherwise modify the decoding process), then the creature's offspring will incorporate a mutated genotype-phenotype mapping. This is the particular phenomenon which we initially set out to investigate.

### 25.3 The Emergence of Pathological Constructors from Genetic Reproducers

When all random perturbations are disabled our seed creature reproduces effectively and populates the memory to form a stable ecosystem of identical creatures. However, when all random perturbations are switched on we immediately see a large emergence of pathological constructors which saturate available CPU time and memory space. Under a series of simulations where each source of random perturbation was individually disabled, the disabling of the segment deletions showed an apparent prevention against the emergence of pathological constructors. When a large segment deletion occurs while copying the genotype from parent to offspring, the resultant creature will typically consist of a functional phenotype, assigned to a partial genotype. This creature continues to rapidly produce offspring, (due to the short genotype), but these offspring are non-functional as they consist of a corrupt phenotype, assigned to a corrupt genotype. When several such pathological constructors coincide in time, their production rate can be so high that their non-functional offspring displace the entire population of functional self reproducing creatures, i.e., ecosystem collapse.

For von Neumann style genetic reproducers, all random perturbations which corrupt the genotype will result in a constructor which will create at least one functional or non functional offspring. A genotype which experiences a segment deletion will result in a pathological constructor which can construct many non-functional offspring before it is killed by the reaper.

This analysis concludes that the mechanism which results in ecosystem collapse due to pathological constructors appears to depend critically on both the one generation delay from when a random perturbation occurs in a genotype and when it is expressed in the phenotype, and the inclusion of segment deletions. The combination of these factors results in a high level of ease in which segment deletions can lead to corrupt genotypes, while still leaving a functioning phenotype.

By contrast, in order for a pathological constructor to emerge from a self-copier, relatively much more specific random perturbations must occur upon very specific locations which will alter, but not corrupt the reproductive functionality. This suggests that the probability of emerging pathological constructors within a population of genetic-reproducers is much higher than that of a population of self-copiers.

## 25.4 Conclusions and Future Work

The highlighted intricate properties of the von Neumann self reproducing automata, implemented in Tierra suggest that this may not be mutationally robust architecture to support genetic reproduction. A combination of the effects of the segment deletions and the generation delay in expressing random perturbations contribute to the abundant emergence of pathological constructors, hence increasing the ecosystem's susceptibility to catastrophic collapse.

It is worth noting that in the typical reproduction cycle of complex (multi-cellular) biological organisms, most of the "decoding" of the genotype takes place as development of the offspring, i.e., it is under the direction of the (embryonic) offspring phenotype rather than the parental phenotype [3]. If we incorporate this concept within the von Neumann architecture, where the offspring phenotype is decoded from the offspring genotype (as opposed to the parent genotype which is the case with von Neumann's architecture), then this design may not exhibit the one generation delay from when a random perturbation occurs in a genotype, and when it is expressed in the phenotype. A corrupt genotype will immediately be assigned a corrupt phenotype, and hence will not reproduce. It seems likely that such an architecture, implemented in Tierra, would be more evolutionary stable and much less vulnerable to emergence of pathological constructors.

**Acknowledgements** This work has been supported by the European Complexity Network (Complexity-NET) through the Irish Research Council for Science and Technology (IRCSET) under the collaborative project EvoSym.

## Appendix

Source code to reproduce results in this paper can be accessed at: <http://alife.rince.ie/evosym/sab-2012-db.zip>.

## References

1. von Neumann J (1948) The general and logical theory of automata. In: *Cerebral Mechanisms in Behaviour*, pp 1–32
2. McMullin B (2000) John von Neumann and the evolutionary growth of complexity: looking backward, looking forward. *Artif Life* 6(4):347–361
3. Buckley W (2008) Computational ontogeny. *Biol Theory* 3(1):3–6