Evolution of Genotype-Phenotype mapping in a von Neumann Self-reproduction within the Platform of Tierra

Declan Baugh and Barry Mc Mullin

The Rince Institute, Dublin City University, Ireland
declan.baugh2@mail.dcu.ie, barry.mcmullin@dcu.ie

Abstract. John von Neumann first presented his theory of machine self-reproduction in the late 40's [3], in which he described a machine capable of performing the logical steps necessary to accommodate self-reproduction and provided an explanation in principle for how arbitrarily complex machines can construct other ("offspring") machine of equal or even greater complexity. This project implements the von Neumann architecture for self-reproduction within the pre-existing evolutionary platform of Tierra, which implements a (mutable) genotype-phenotype mapping during reproduction. Initially, the memory image of the automaton's genotype and phenotype are physically identical, and each symbol in memory may be interpreted as either as passive numerical data (g-symbol), or a functional instruction (p-symbol) depending on how the symbol is interpreted. If redundancy is introduced to a mutable genotype-phenotype mapping, the mapping system becomes non-invertible, rendering it impossible to compute an automaton's exact genotypic memory image by analysis of the phenotype alone. However, this non-invertible mapping may allow for a more robust genotype, increasing its robustness to fatal mutations and therefore increasing its ability to preserve its phenotypic form under perturbations.

Keywords: von Neumann, Genetic Reproduction, Tierra, Artificial Life, Genotype-phenotype Mapping, Evolutionary Growth of Complexity

1 Introduction

Von Neumann's architecture for machine self-reproduction, presented in his *Theory of Self-reproducing Automata* [4,1], provides a proof-of-principle demonstration that machines can directly, or indirectly, give rise to machines arbitrarily more complex than themselves. The described machine, M, was decomposed into two primary components, a functional component P, and a passive component G, such that $M = (P + G)$ [2].

G represents a one-dimensional string of symbols which has no active/functional capability, but can be interpreted as information, similar to the *tape* of a Turing machine. The information within G is used to describe an arbitrary machine X under some function, $\phi()$, such that $G = \phi(X)$. P is further divided into four fundamental subcomponents, a *general constructive automaton* A, a *general copying automaton* B, a *control unit* C, and *ancillary machinery*, D. G will be referred to as the *genotype* while P as the *phenotype.*[1].

The general constructive automaton A, has the ability to read the symbols within G, and interpret them as an encoded description of an arbitrary machine X by applying an inverse function, $\phi^{-1}()$, or $\psi()$, to G, and construct the described machine X. We denote this by saying, $\psi(G) = \psi(\phi(X)) = \phi^{-1}(\phi(X)) = X$. In other words, when supplied with a genotype, the general constructive automaton applies the decoding function $\psi()$ to G, in order to construct the arbitrary machine X.

The general copying automaton B reads and duplicates the machine description $\phi(X)$. A control unit C is required to govern the automaton $(A + B)$, directing its operation and insuring that the offspring creature is "activated" once its construction is complete. The final component, the ancillary machinery D, refers to all conceivable functionality that the machine may possess which does not interfere or hinder the reproductive operation of $(A + B + C)$.

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 when introducing redundancy to the genotype-phenotype mapping.

2 Implementation within The Tierra Platform

Tierra is an artificial life platform where populations of assembler language self-reproducing automatons (creatures) compete with one another within a one-dimensional circular core memory for both CPU time and memory space [5].

Typically, self-reproduction within Tierra is accomplished via *self-copying*, where a creature must inspect its entire memory image in order to construct an identical offspring. In order to implement the von Neumann architecture within the platform of Tierra, the seed automaton must enforce a division of labour between the storage of genetic information and the catalytic functionality, hence recognising the roles of genotype and phenotype.

In order to introduce a distinction between genotype and phenotype, an arbitrary genotype-phenotype mapping must be implemented. The evolutionary

[1] Although von Neumann never used these terms, we now associate the components in question with the genotype and phenotype in organic biology.

trajectory of such an automaton will be in part, determined by the nature of this arbitrarily elected mapping. We can only claim that any phenomenon observed will be characteristic to the specific implemented mapping system. For the purpose of this project, a bijective, mono-alphabetic substitution cipher was chosen. This method was loosely based on the genetic code, in which an mRNA, consisting of a one-dimensional string of symbols (nucleotides), is transcribed into a different specific string of symbols (amino acids). Such a mapping system which allows perturbations to the genotype which may alter the description of the general constructor, specifically altering the genotype-phenotype mapping function $\psi()$, may give rise to new evolutionary trajectories which operate an altered genotype-phenotype mapping.

This type of a mutable genotype-phenotype mapping was facilitated via the inclusion of a lookup table within the general constructive automaton. The lookup table consists of a one-dimensional string representing the full list of p-symbols[2] available within the phenotype space. During construction of an offspring phenotype, the parent's genotype is incrementally examined and the g-symbol at each memory location is read and interpreted as a numerical value. The p-symbol situated at the corresponding location in the lookup table is read, and is written to the offspring phenotype where it will function as an active instruction. In an attempt to replicate conditions early in the phase change from simple RNA replicators to a system of mRNA and amino acids, an identity mapping from genotype to phenotype is implemented, where the content of the one-dimensional memory image of the genotype is physically identical to that of the phenotype. This situation is analogous to that of the RNA world, where in order to replicate, an RNA molecule can act as both a functional catalyst, or a string of symbols to be interpreted, depending on whether it is acting as the catalyst or template.

The designed ancestor requires a minimum of 28 phenotypic instructions in order to self-reproduce. In an attempt to mirror the redundancy of the genetic code, a symbol set of 64 separate 6-bit binary digits was implemented. These may be interpreted at 64 different numerical values within the genotype, or 64 different phenotypic instructions, 28 of which are employed within the phenotype and contribute towards self-reproduction.

3 Experimental Procedure

The Tierra soup was inoculated with the described von Neumann style ancestor and the system was run for 100 billion CPU cycles, which is approximately 250 thousand generations. Every strain of creature to emerge throughout the run was

[2] The symbols within the phenotype memory image which are interpreted as functional instructions will be referred to as p-symbols. Similarly, symbols within the genotype which are interpreted as passive data will be refereed to as g-symbols

captured and the number of employed p-symbols[3] and non-employed p-symbols within the lookup table for each creature was counted.

If a specific p-symbol exists within the lookup table, then there must exist a specific g-symbol which maps onto it. If a p-symbol is absent from the lookup table then it is lost from the genotype-phenotype mapping. With our current mapping system, it is impossible for a p-symbol which is absent in a parents lookup table to be included in its offspring's phenotype (with the exception of random perturbations introducing p-symbols to an offspring).

The population of employed vs. non-employed p-symbols in the lookup table of each creature was then plotted against the time of emergence of that individual.

4 Results - Reduction of Phenotype Space

Initial simulations showed evolutionary behaviour such as informational parasitism, similar to that documented in Ray's initial experiments. However this evolutionary behaviour has already been studied and documented [5]. We wish to focus on the specific evolutionary lineages which arise as a direct result of a change in the genotype-phenotype mapping. A change in the genotype-phenotype mapping will be most easily recognised by a change in the lookup table.

Initially, non-fatal inheritable silent perturbations[4] of the genotype will occur in the description the lookup table. This will alter the genotype-phenotype mapping and allow previously silent g-symbols[5] to be mapped onto employed p-symbols. This allows single employed p-symbols to be mapped onto by multiple g-symbols.

The initial ancestor has 36 silent g-symbols, which are mapped onto 36 different non-employed p-symbols. As neither the silent g-symbols nor the non-employed p-symbols functionally contribute to the reproduction of offspring, the silent mutations which affect these symbols are random and arbitrary. However, it was found that there was a strong bias towards the mapping of silent g-symbols onto employed p-symbols. During an evolutionary run, we see a sharp decrease in the number of non-employed p-symbols within the lookup tables of newly emerging creatures. Eventually, all 36 non-employed p-symbols are eliminated from the descendants of the initial ancestor, and the 64 positions in their lookup tables consist entirely of employed p-symbols. This result can be seen in Figure 1.

[3] An employed p-symbol refers to those which have a functional role in the process of reproduction. Non-employed p-symbols are included to introduce redundancy, and do not actively contribute towards the reproduction process.

[4] A silent perturbation is one which alters the genotypic sequence, but does not affect the structure of the phenotype.

[5] By silent g-symbols, we refer to g-symbols which initially mapped onto non-employed p-symbols.

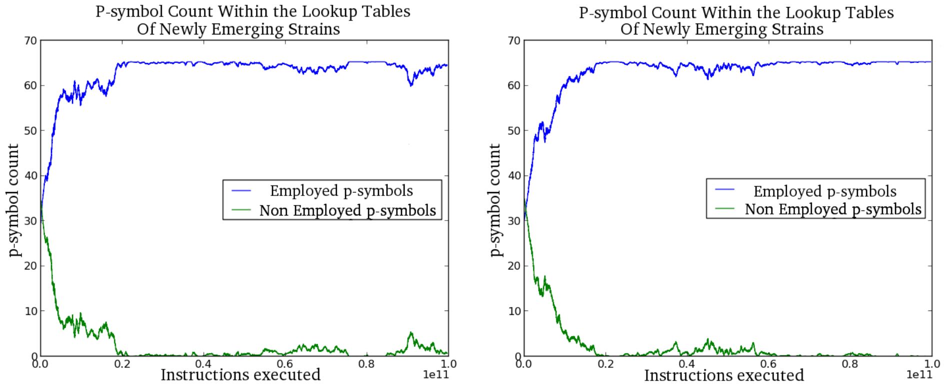

Fig. 1. Two evolutionary simulations displaying the number of employed vs. non-employed p-symbols present in the lookup table of strains of newly emerging lineages.

5 Conclusion and Further Work

The evolution of the genotype-phenotype mapping will initially be driven predominantly by the underlying physical dynamics of the coding system. The nature of the substitution cypher mapping mechanism employed means that certain perturbations of the lookup table are not directly reversible. Because of this non-reversibility, if successive mutations occur they will accumulate, somewhat similar to Muller's ratchet in evolutionary genetics[6], resulting in a biased drift in the genotype-phenotype mapping, eventually eliminating all non-employed p-symbols from the phenotype by ensuring that they are not mapped onto by any elements of the genotype space.

The mapping system is now non-injective and non-surjective, as there exists symbols within the phenotype space (employed p-symbols), which are mapped onto by more than one symbol in the genotype space. There are also symbols within the phenotype space (non-employed p-symbols) not mapped onto by any element of the genotype space. Furthermore, this renders the mapping invertible, as it is now impossible to determine the exact genotypic sequence via inspection of the phenotype alone as $\psi() \neq \phi^{-1}()$ as now $G = \psi'(P)$, where $\psi'()$ represents the altered genotype-phenotype mapping function.

Mutational robustness and Darwinian selection. Darwinian selection may then sharpen the genotype-phenotype mapping, and create a more mutationally robust genotype. The allocation in which silent g-symbols are mapped upon employed p-symbols may be subject to Darwinian selection. Following a perturbation to a g-symbol, a phenotype may still preserve form if both g-symbols

[6] In evolutionary genetics, Muller's ratchet (named after Hermann Joseph Muller, by analogy with a ratchet mechanism) is the process by which the genomes of an asexual population accumulate deleterious mutations in an irreversible manner.

transcribe to the same p-symbol. A p-symbol which is very common within the phenotype has a high probability of being perturbed within the genotype. If a large percentage of the silent g-symbols are mapped upon the most frequent employed p-symbols, then the phenotype will have an increased probability of holding form following an inheritable perturbation to the genotype. However, this work is pending and requires further experimentation.

The Tierra source code for these experiments along with the analysing software can be found at: `http://alife.rince.ie/evosym/alife_2013_dbbm.zip`.

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. Computing facilities were facilitated by The Irish Centre for High-End Computing (ICHEC).

References

1. Baugh, D., McMullin, B.: The emergence of pathological constructors when implementing the von neumann architecture for self-reproduction in tierra. In: From Animals to Animats 12. Springer (2012)
2. McMullin, B.: Architectures for self-reproduction: Abstractions, realisations and a research program. In: Artificial Life 13. pp. 83–90 (2012)
3. von Neumann, J.: The general and logical theory of automata. In: Cerebral Mechanisms in Behaviour, pp. 1–32 (1948)
4. von Neumann, J.: Theory of self-reproducing automata (1966), edited and completed by A.W. Burks.
5. Ray, T.: An approach to the synthesis of life. Artificial life II 10, 371408 (1991)