Computational Ontogeny

William R. Buckley
California Evolution Institute, San Francisco, California, U.S.A.

Keywords: Automata, Cellular, Construction, Efficient, Learning, Machine, Replication.

Abstract: Of interest to the theory of machines that construct is ontogeny, by which process of development the constructor is transformed from immature to mature form. Whereas we have already shown that self-replicating machines generally are able to bootstrap themselves through the construction of sub-machines (such as organs that rewind a tape, or replicate a tape, or initiate the behavior of a construct), in this paper we present in abstract a constructor that bootstraps its ability to construct, through the construction of sub-constructors. This is to say, we present a constructor that learns how to construct, and does so by constructing; our constructor is in truth a proto-constructor. Here, learning occurs by the addition of new machine configuration; each learned lesson is correlated with specific additions to machine configuration.

1 INTRODUCTION

The theory of machines that construct began with the seminal work of von Neumann (Jeffress, 1951; von Neumann, 1966) via his self-replicating machines. Such constructing machines are viewed as having *universal* competence over construction where the yield is passive. Passivity[1] correlates formally with the absence of signal to be found coursing within and through configuration, though the practical correlate is in-animation which is quite contrary to typical expectations of automata such as clocks. Indeed, von Neumann ignored many lessons of Nature in developing his self-replicator model, eliciting observations such as those of (Shalizi, 2012) who gives cautiously reserved praise.[2] Maynard Smith offers his own cautious praise while pointing to a lack of a machine embryology in artificial life models (Smith, 1986); see (Buckley, 2008a) for a rudimentary model that addresses some of Maynard Smith's concerns.

These constructing machines have been further examined in the work of McMullin, who mused over the relationship between construction and evolution, and how this relationship was the proper question being addressed by von Neumann, as opposed to machine self-replication per se (McMullin, 2000). The key point of McMullin's argument is that the kind of reduction in artifact complexity that is the result of manufacturing processes could perhaps be countered by an understanding of how systems of constructors might be organised to yield ever more complex constructs, with the expectation being that some of these constructs would themselves in fact be constructors. That is, McMullin addressed notions of knowledgeable, qualitative leverage over construction processes as a means of gaining quantitative leverage in the production of artifacts of ever increasing complexity.

Missing from McMullin's model is an example. Presently, we give one such example.

The traditional view of machine self-replication holds that the mother machine constructs all of the daughter machine, and that during construction of the daughter machine, the daughter machine is passive; the daughter machine initiates behavior subsequent to its construction. Such a self-replicating machine is necessarily composed of many subordinate machines, with the constructor proper being of special importance. The portion of the self-replicating machine that is properly the constructor machine is itself subject to decomposition, and it happens that proper subsets of resulting subordinate constructor machines are sufficient as to machine construction even if they are not sufficient as to self-replication. This is to say, constructors proper can observe a developmental process. In this paper, we present a self-replicating machine that observes properly the ontogeny of its con-

[1] Especially for cellular automata.

[2] The relevant quote is: "CA were not invented, however, to be realistic models of Nature. They started with John von Neumann, who wanted to study self-reproduction, and decided that the first thing to do was ignore everything biologists had learned about the way actually existing organisms reproduce themselves. This is known as hubris, and is *especially galling* when it works." (Our emphasis.)

Buckley W..
Computational Ontogeny.
DOI: 10.5220/0004204101160121
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (ECTA-2012), pages 116-121
ISBN: 978-989-8565-33-4

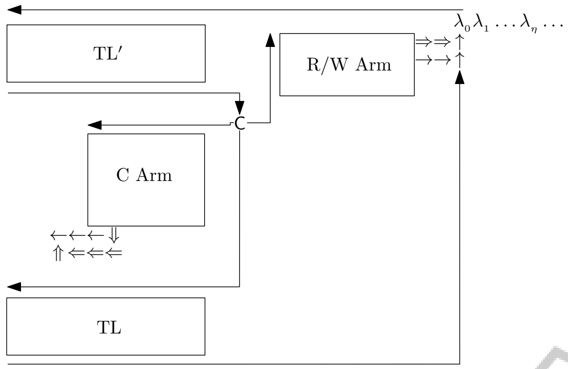

Figure 1: High-level architectural depiction of a von Neumann self-replicator. Each of the four organs shown in the figure is composed entirely of stages, arranged with input at figure top, and output at figure bottom, except for the R/W Arm, where stages are arranged from left to right. The arrows show major signal paths within the configuration, and the capital C symbol represents a confluent state, and so a point of signal duplication. Additional stages are added to the left edge of the TL, TL′ and C Arm organs; the construction arm has no access to the R/W Arm organ. No sense of scale should be inferred from this drawing.

structor, and this in addition to the otherwise general ontogeny of the self-replicating machine as a whole (Buckley, 2008a). In accomplishing this ontogeny, our machine acquires new configuration, thereby increasing the set of configurations that the machine can construct; our machine learns to construct, and does so by constructing and incorporating into its own configuration that newly constructed configuration which represents those lessons. Further, the set of instructions acceptable from tape correspondingly increases.

Our example self-replicating machine is implemented in von Neumann 29-state cellular automata (von Neumann, 1966), and has physical characteristics common to such machines. In particular our machine reads its tape twice, has distinct construction and tape replication phases of its behavior and suffers the consequences of destructive reading of its tape. For a thorough and yet succinct review of the characters of state types and their interactions within von Neumann cellular automata, see the opening three paragraphs of (Buckley and Mukherjee, 2005). For discussion of types of signal, see (Burks, 1970) or (Buckley, 2008b).

2 ABSTRACT SELF-REPLICATOR DESIGN

We should like to make as concrete as possible the design of this configuration, so as to strengthen reader understanding of the details of our model of con-

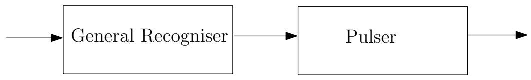

Figure 2: Abstract structure of the *stage*. A stage is composed of a general recogniser followed by a (large) pulser. Stages generally emit a series of one to five signals for each accepted signal.

structor ontogeny. To assist with this goal, we give a self-replicating configuration that has a rather regular structure, this serving to simplify reader understanding of configuration behavior. This configuration is composed of exactly four different types of sub-configuration: multiple copies of one kind of signal processor (emits a set of signals in response to acceptance of a specific, *recognised* signal; this organ is called a *stage*), a single read/write arm (allows reading and writing of the tape), a single construction arm (allows for the alteration of state class of target cells, and for the articulation of space traversal for construction signal), and simple signal paths (constructed largely of cells set to one of the four ordinary transmission states, with the occasional confluent cell for signal duplication). The sequential acceptance of signals by stages yields a further abstraction, in that the configuration behaves according to a microprogram. This microprogram is expressed as the signaling that is generated by the various stages comprising the configuration. A block diagram of this self-replicator is given in figure 1.

A stage is a combination of a signal pulser and a signal recogniser; see (Thatcher, 1970) for a description of each of these two organs. An understanding of the detailed operation of stages is not critical to understanding of our thesis regarding machine ontogeny; such understanding is amply served by knowledge of the relationship between signal acceptance and signal generation, that generation follows acceptance. Instead, the point critical to our thesis is the set of states used to build these two organs, a point we explore later in this paper. We see the stage abstractly diagrammed in figure 2. Stages serve the purpose of translation, bringing about a sequence of operations in response to a specific signal, and in so doing yield the conversion of instruction found on tape into purposeful construction and articulation signal.

The various pairings of signal recogniser and pulser, the stages of the self-replicator, are organised into four groupings, these correlating to all portions of the configuration save the construction and read/write arms, proper, and the lowly signal paths. Two of the groupings yield signals that direct articulation of the two arms; the *C Arm* organ and the *R/W Arm* organ. A few of the signals accepted by the stages of the C Arm organ serve not articulation but construction.

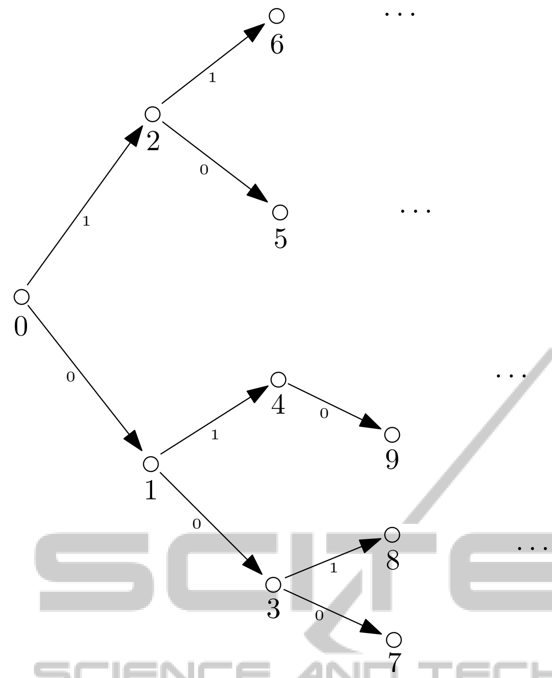

Figure 3: Tree structure of the TL and TL′ languages. Edges of the graph are labeled with the corresponding value of input read from tape. For each node ν, the successor nodes are numbered $2\nu+1$ (for inputs of value zero) and $2\nu+2$ (for inputs of value 1).

The other two groupings serve the tape read and instruction translation processes, proper; they define a language by which the tape is read, and instructions discerned. It is these latter two groupings of stages that are of prime concern here in this paper, and define the *TL* and *TL′* organs. For our self-replicator, the reading of the tape occurs by a sequential alternating pattern of TL generation, followed by TL′ recognition. The TL organ generates signal that is then used to read the tape, and so by consequence of von Neumann destructive reading yields generation of TL′ signal, that is then recognised by the TL′ organ. Upon such signal recognition it is trivially possible to discriminate between the reading of a one bit and the reading of a zero bit, and as we have already said that any one stage accepts only one signal, it is clear that for each TL′ signal, there is one and only one stage that accepts the signal; no two TL′ stages point to any one TL stage, and no two TL stages point to any one TL′ stage. This implies that the stages of TL are arranged in a binary tree structure; the first three levels of the tree are shown in figure 3. Traversal of the tree always begins at the root, with signal TL_0. Traversal of the tree terminates at level six, where an instruction from tape is accepted. Terminal nodes in the tree are numbered 31 through 62, inclusive. Figure 4 shows the organisation of stages in the TL and TL′ organs,

TL'_0	R_f W_0 H_f IC_1
TL_0	R_f H_f IC_2
TL'_1	R_f W_0 H_f IC_3
TL_1	R_f H_f IC_4
TL'_2	R_f W_0 H_f IC_5
TL_2	R_f H_f IC_6
TL'_3	R_f W_0 H_f IC_7
TL_3	R_f H_f IC_8
⋮	
TL_{43}	R_f W_0 H_f IC_0 OD
⋮	

Figure 4: Organisation of stages in the TL′ organ. Implied is that signal IC_0 triggers generation of TL_0 by the TL organ. Notice that only for TL′ signal is there call to generate instruction that brings the writing of a zero (W_0) upon the tape; this accounts for the destructive read of zero.

and the microprogram for a few stages of the TL′ organ.

We should mention that the TL′ organ has exactly twice as many stages as has the TL organ. Also, the signal emitted by a TL′ stage that acts as a trigger to bring subsequent emittance of TL signal is itself not a member of the TL/TL′ set of signal. Instead, this trigger signal is of an internal code (IC) which is quite different from TL and TL′ signals.

We may see how the sequence of TL → TL′ → TL → TL′ ... yields reading of an instruction from tape and translation of that instruction into either construction arm articulation signal or cell state construction signal. We see that the instruction <011001> is read with the sequence [TL_0 : TL_1 : TL_4 : TL_{10} : TL_{21} : TL_{43}]. This is to say that in reading the instruction <011001>, the TL organ is directed to generate the foregoing sequence of signals. The origin of this direction is the TL′ organ. For each TL′ signal recognised, a corresponding set of signals is issued. These issued signals direct the extension and retraction of the read/write arm and the return signal path, any necessary repair to tape (owing to destructive read), any signal corresponding to construction arm articulation and construction, and the next TL signal to issue.

The mechanism of destructive read is simple enough. In our implementation (which differs slightly from that of von Neumann), a special signal <100011> is used to actually read the bit as represented upon the tape. If the tape at the location of reading holds a representation of the value one, then the signal is returned unchanged. If however the tape at the location of reading holds a representation of the value zero, then a cell is constructed that represents the value of one, and the signal is changed, returning as <1>. It is this alteration of read signal that is responsible for the conversion of TL signal to TL′ signal. So, for the previously given TL sequence, there will be generated in the read process the TL′ sequence [TL'_0 : TL_1 : TL_4 : TL'_{10} : TL'_{21} : TL_{43}]. We see that in reading bits valued at one, the corresponding TL signal is unaltered, and that for zero valued bits, the TL signal is altered to TL′ signal; the TL′ organ accepts both TL and TL′ signal.

This is the location in the text at which the key point of our thesis comes into view. It is clearly the case that only those instructions on tape that are represented within the read sequence of the set of stages comprising the TL′ organ will be accepted by the configuration. Further, if it so happens that the set of states of which any stage is composed is itself a proper subset of the states available for construction, then the constructor need be able only to construct that proper subset of states in order to construct a stage, and hence for the machine to observe ontogeny. Von Neumann configurations are composed generally of nine passive states, yet it happens that for the TL, TL′ and C Arm organs of our self-replicator, the stages and their interconnections are composed of only six of these states. It is therefore sufficient for the observation of ontogeny that our self-replicator be initially able to construct only these six states.

Of course, the reason for engaging in ontogeny is that the configuration needs be able to construct all nine passive von Neumann states in order to engage in the act of self-replication. The limit on usage of just six cell states applies only to those stages used in the construction of the TL, TL′ and C Arm organs. For example, the stages of R/W Arm organ employ a different subset of six cell states. Thus, our current demonstration of constructor ontogeny, as opposed to the more general argument for self-replicator ontogeny given earlier (Buckley, 2008a).

While this has implications for the tree suggested in figure 3, that not all of the stages represented in the figure need be constructed at the start of machine behavior, the issue is more broad, for it is also true that the C Arm organ need only include stages sufficient to construction of these six states employed in construction of TL, TL′ and C Arm organ stages, and therefore it too can exhibit ontogeny. Thus, we show directly and distinctly the ontogeny both of the control mechanism over construction, and of the mechanism of construction itself. While we do not take effort in this paper to show the result, it happens that within the given model one may even observe ontogenic development of the R/W Arm organ, at the cost of using exactly one perfect signal crossing organ; see (Buckley, 2008b) for a discussion of signal crossing in von Neumann cellular automata.

000000	RX	010001	LUR
000001	RR	010010	DX
000010	RUX	010011	DR
000011	RUR	010100	DRX
000100	RDX	010101	DRR
000101	RDR	010110	DLX
000110	UX	010111	DLR
000111	UR	011000	TM
001000	ULX	011001	OD
001001	ULR	011010	OL
001010	URX	011011	OR
001011	URR	011100	OU
001100	LX	011101	CN
001101	LR	011110	SD
001110	LDX	011111	SL
001111	LDR	100000	SR
010000	LUX	100001	SU

Figure 5: Code assignments for instruction set of self-replicator. Codes for construction arm articulation come in four groups of six signals. The six signals are, for each group, of a symmetrical nature. For instance, RX is *right extend*, and RR is *right retract*. Similarly, DX is *down extend*, and DR is *down retract*. For rounding corners, we see that ULX is *up-to-left extend*, ULR is *up-to-left retract*, and LDR is *left-to-down retract*. TM is the *tape mark*, and partitions the code assignment list into two parts, with construction arm articulation codes coming before the codes for configuration construction. Extension always increases the length of the construction arm, and retraction always reduces the length of the construction arm.

3 ONTOGENY OF A CONSTRUCTOR

We have now to address the length of instructions on tape. For the example self-replicator, it happens that universal articulation (over all quadrants) is necessary to our model of ontogeny. Therefore, it is necessary that the constructor support a total of 24 different motions, thus necessitating at least 24 different instruction codes on tape. The need for constructing nine state types increases the required instruction code count to 33, and the addition of a Tape Mark symbol stretches the number to 34 different codes required to express a configuration description on the tape. We see in figure 5 the instruction codes assigned for these 34 different operations; hence the six levels of the binary tree suggested in figure 3.

The six von Neumann states necessary to construction of stages employed in the TL, TL′ and C Arm organs are the four ordinary transmission states $\{\leftarrow, \uparrow, \rightarrow, \downarrow\}$, the confluent $\{C\}$ state and the downward pointing special transmission state $\{\Downarrow\}$. Therefore, those stages that correspond to the instruction codes for the special transmission states $\{\Leftarrow, \Uparrow, \Rightarrow\}$ need not be represented in the TL, TL′ and C Arm organs. Indeed, the TL stages corresponding to the signals $\{TL_2, TL_5, TL_{11}, TL_{23}, TL_{47}\}$ need not be constructed prior to configuration start, nor need the TL′ stages corresponding to these signals be constructed prior to configuration start. Further, the TL′ stage corresponding to signal $\{TL'_{46}\}$ does not need to be constructed prior to configuration start. Clearly, corresponding stages from the C Arm organ also need not be constructed prior to configuration start, for a total of 19 stages that need not be constructed prior to the start of configuration behavior.

Thus, the machine begins its behavior with construction competence restricted to those states of which TL, TL′ and C Arm stages are comprised, and completes its behavior having acquired unrestricted construction competence

By placing upon the tape a description of these stages (that are missing from the initial state of the configuration), all of these stages can be added to the configuration post-initiation of behavior. This yields an increase in the number of instructions acceptable from the tape. Careful design of the interface of stage to signal line allows the stage to be fully constructed before it is linked into the corresponding organ, and the acceptance of signal in a highly discriminatory way ensures that no spurious signal is generated during ontogeny. The configuration remains well behaved throughout any and all ontogeny.

4 DISCUSSION

Simply put, ontogeny is genome-governed development.

Development is the acquisition of new features, be they physical or otherwise. For biological organisms, ontogeny is very complex, with many sources of information giving their affect ultimately to biological metabolism, and this metabolism yielding emergent features, like hands and eyes and legs and hearts. It is commonly understood that biology sees the genome not as a blueprint but as a recipe, and yet we know that those recipes are sufficiently regular that resemblances between generations of individuals is strong, if not uncanny. We suggest that there is within that recipe a hint of blueprint, yet.

This leads to justification of our model. In this case, the blueprint analogy is strong. Indeed, for typical von Neumann self-replicators, the description is exactly a blueprint; the state of every cell is strictly mapped, and instructions to construct these cells are placed within a bed of other instructions that direct space articulation of the construction arm. It becomes a real challenge to show how such a machine can develop from an immature state into a mature state. The use of stages to represent the means to control machine function allows the machine to be partition-able down to the level of the stage; the proper function of any one stage is not dependent upon the proper function of any other stage. Stages are mutually independent, and yet by combining them, higher-order functionality is obtainable, all according to the programming (accepted and emitted signals) represented within constructed stages; self-replication becomes an emergent property of the machine.

In the von Neumann model of machine self-replication, machine M has a description of itself D expressed in a language L that is accepted by M, with acceptance of D by M yielding construction of another M and another D. Further, the (daughter) copies of M and D are placed adjacent to each other in the same pose as was assumed by the original (or parent) M and D. The important point is that D is a complete description of M; it has not more nor less information than is needed to describe M in the language L. M and D represent a distribution of total complexity for the system M(D).

In our model, we alter that complexity distribution, by placing more information about M into the description D, thus reducing the complexity of M and increasing the complexity of D, and we do so in such a way that M is able still to construct modifications to itself. We suggest that the development of biological zygotes is more than analogous with the ontogeny expressed in our model; the chief differences are perhaps in complexity of process as opposed to fundamental difference of process.

5 CONCLUSIONS

We have presented in abstract a self-replicating machine that observes ontogeny, demonstrating a direct link between development and learning within automata. We have also shown that there are pathways of construction that facilitate the development of constructors from a state of restricted construction competence to a state of unrestricted (general) construction competence.

The architecture of our example self-replicator is sufficiently flexible that it may provide a useful framework for the modeling of open-ended evolution within machines.

One may see also within the architecture of our example self-replicator the suggestion of an alternative cellular automata architecture, one based upon cells that either implement the functionality of a stage, or of a simple signal line. Such a transition of automata definition might well improve computational performance sufficient to make practicable the use of such automata in more general study of biological processes.

ACKNOWLEDGMENTS

The work in this paper responds to the reservations of Daniel Mange regarding the ability of the model given in our paper Computational Ontogeny to support further machine decomposition, and particularly decomposition of the constructor (Mange, 2005).

Many thanks are extended to Bruce H. Weber and David Depew for their many helpful suggestions and comments, particularly regarding details of biological ontogeny.

To Cosma for his wisdom and his notebooks.

REFERENCES

Buckley, W. R. (2008a). Computational ontogeny. *Biological Theory*, 3(1):3–6.

Buckley, W. R. (2008b). Signal-crossing solutions in von neumann self-replicating cellular automata. In Adamatzky, A., Alonso-Sanz, R., Lawniczak, A., Martinez, G. J., Morita, K., and Worsch, T., editors, *AUTOMATA-2008, Proceedings of the 14th Workshop of Working Group 1.5 (Cellular Automata and Machines) Bristol, UK, June 12-14, 2008*, pages 453–502. Luniver.

Buckley, W. R. and Mukherjee, A. (2005). Constructibility of signal-crossing solutions in von neumann 29-state cellular automata. In Sunderam, V. S., van Albada, G. D., Sloot, P. M. A., and Dongarra, J., editors, *International Conference on Computational Science (2)*, volume 3515 of *Lecture Notes in Computer Science*, pages 395–403. Springer.

Burks, A. (1970). Von neumann's self-reproducing automata. In Burks, A., editor, *Essays on Cellular Automata*, pages 3–64. University of Illinois Press, Urbana and London. Available from the DeepBlue server of the University of Michigan, on the internet at the persistent URL (URI) http://hdl.handle.net/2027.42/3954.

Jeffress, L. A., editor (1951). *Cerebral mechanisms in behavior: the Hixon symposium, September 20, 1948, Pasadena*, New York, NY, USA; London, UK; Sydney, Australia. John Wiley and Sons. Reprinted 1967.

Mange, D. (2005). Private email correspondence. Dated July 15, 2005.

McMullin, B. (2000). John von neumann and the evolutionary growth of complexity: Looking backwards, looking forwards. *Artificial Life*, 6(4):347–361.

Shalizi, C. R. (2012). Cellular automata. Dated 23 Jun 2012 - Retrieved 20120818 - Available online at the URL http://cscs.umich.edu/ crshalizi/notebooks/cellular-automata.html.

Smith, J. M. (1986). *The problems of biology*. Oxford University Press, New York.

Thatcher, J. W. (1970). Universality in the von neumann cellular model. In Burks, A. W., editor, *Essays on Cellular Automata*, pages 132–186. University of Illinois Press. Available from the DeepBlue server of the University of Michigan, on the internet at the persistent URL (URI) http://hdl.handle.net/2027.42/7923.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana, IL, USA. Von Neumann's work on self-reproducing automata, completed and edited after his death by Arthur Walter Burks. Also includes transcripts of von Neumann's 1949 University of Illinois lectures on the "Theory and Organization of Complicated Automata".