What *is* a Universal Constructor?

B. McMullin

School of Electronic Engineering
Dublin City University
Dublin 9
Ireland

E-mail: McMullinB@DCU.IE

Abstract

John von Neumann's seminal investigations into the theory of complex automata (von Neumann 1951; Burks 1966b) arguably marked the original birth of the field now called *Artificial Life*. In the years since, von Neumann's work has been studied and elaborated in many ways; but it is my view that it has also been, to a significant extent, misunderstood. One facet of this misunderstanding revolves around von Neumann's concept of the *Universal Constructor*. Von Neumann introduced this concept as a preliminary step in tackling the problem of realising a spontaneous and open-ended growth in automaton complexity. He formulated the concept by an analogy with Turing's earlier notion of a *Universal (Computing) Machine* (Turing 1936). This paper attempts to identify precisely what should, and, more importantly, should *not*, be read into this analogy.

Introduction

The seminal work carried out by John von Neumann in the theory of complex automata is usually characterised as having been concerned with the problem of realising *non-trivial self-reproducing machines*, where "non-trivial" is interpreted as requiring that the machine contain an embedded universal computer. This interpretation of von Neumann's work seems to have originated with A.W. Burks, in his "completion" of von Neumann's unpublished manuscript *Theory of Automata: Construction, Reproduction, Homogeneity* (Burks 1966a). In any case, it has become canonical (e.g. Codd 1968, Burks 1970b, Moore 1970, Thatcher 1970, Herman 1973, Langton 1984, Poundstone 1985, Smith 1992).

I, on the other hand, take the view that this interpretation is (and probably always was) fundamentally mistaken. I view it as a self-perpetuating *myth.*[1]

Firstly, as far as I can establish, this interpretation is nowhere to be found in von Neumann's *own* writings. But even as a myth, it is of dubious coherence, as has been known for at least 20 years. Thus, Herman (1973) exhibited a self-reproducing "machine", meeting the technical desiderata of the von Neumann myth, but which was, nonetheless, manifestly *trivial.* Langton (1984) also independently intimated that something must be amiss with the conventional interpretation of von Neumann's work (though his diagnosis of precisely *what* seems to have been orthogonal to that of Herman). Despite such difficulties, the myth has survived largely intact—albeit Smith (1992), for example, is reduced to an appeal to "historical precedent" for his continued advocation of it. However, I have detailed my objections to the von Neumann myth elsewhere, and at length (the reader with sufficient stamina may wish to refer to McMullin 1992, Chapter 4); this paper is not directly concerned with that, but rather with a *prolegomenon* to it.

Nonetheless, it *is* necessary that I make clear my rejection of the traditional interpretation of von Neumann's work at the outset, for that rejection provides an essential context for the matters I *will* discuss. For example, Smith has recently concluded that von Neumann's concept of a *universal constructor* is essentially redundant and can be dispensed with (Smith 1992, p. 719); whereas my paper is devoted to an analysis, and attempted clarification, of this very concept. Now there can be little point in further refining concepts which are already otiose, so there is evidently something not quite right here. The explanation for my apparent obduracy is simply that Smith and I are operating within different (perhaps even "incomensurable"!) problem situations: Smith endorses the von Neumann myth (and I will happily stipulate that his conclusions are valid in that context); but I reject the von Neumann myth, and this leads me to believe that von Neumann's concept of the universal constructor is not only still useful, but absolutely pivotal to a proper understanding of von Neumann's real achievement. Indeed, my belief is that this proper understanding may be blocked if the concept is not very carefully clarified; and providing such clarification is precisely my objective here.

In brief then, my approach will be this. I first introduce some specialised terminology. This may represent a degree of overkill for the limited purposes of this paper: but it is essential to the more detailed discussions the paper potentially

[1] It is not my intention here, or elsewhere, to be merely polemical; rather, by being forthright, I hope to facilitate subsequent criticism (and, no doubt, correction) of my views.

leads on to, elsewhere, and I therefore prefer to retain as much consistency as possible—even at the risk of complicating this preliminary presentation. I then state my own view of the problem von Neumann was actually originally attempting to solve (which, of course, differs fundamentally from the von Neumann myth); and I go on, in the context of this (admittedly iconoclastic) view of von Neumann's problem, to review and re-present his concept of the *universal constructor.*

Some Terminology

I shall refer to some particular formalisation of abstract automata as defining an *A-system.* Within the context of such a particular A-system I shall refer to the entities which are to be regarded as "automata" as *A-machines.* The possible "primitive" (irreducible) parts of an A-machine will be called *A-parts.* Some A-machines may operate so as to acquire further A-parts, and assemble them into new A-machines. I shall call these *A-constructors.* Some A-constructors may, in turn, be capable of constructing offspring which are "identical" to themselves. I shall call these *A-reproducers.*

Von Neumann's Problem (P_v)

> Everyone knows that a machine tool is more complicated than the elements which can be made with it, and that, generally speaking, an automaton A, which can make an automaton B, must contain a complete description of B and also rules on how to behave while effecting the synthesis. So, one gets a very strong impression that complication, or productive potentiality in an organization, is degenerative, that an organization which synthesizes something is necessarily more complicated, of a higher order, than the organization it synthesizes.
>
> von Neumann (1966a, p. 79)

If this were really so it would represent, at the very least, a severe difficulty for the continued application of mechanistic theories in evolutionary biology—it is evidently an issue of considerable importance. Von Neumann proposed to resolve this issue by a "proof-of-principle" argument: i.e. by demonstrating the possibility of a spontaneous, open-ended, growth of automaton "complexity" within a rigidly mechanistic, and completely formalised, framework. The problem of achieving this is what I refer to as *Von Neumann's Problem* or P_v. It seems to me that Von Neumann's crucial insight was to recognise that there *is* a way whereby P_v can be solved (at least in principle), and solved relatively easily at that. However, in this paper, I concentrate only on one preliminary step in von Neumann's solution schema: the concept of a *universal constructor.*

The A_T-system

Von Neumann's attempted solution to P_v was heavily, and explicitly, influenced by Turing's formulation and analysis of a certain formalised class of "computing machines" (Turing 1936). Turing's analysis had the following general structure. He first introduced a basic formalisation of the notion of a *computing* machine. In my terms, this corresponds to the definition of a (more or less)

specific A-system. I shall distinguish references to this with a subscript T, thus: A_T-system, A_T-machine etc. What I term an A_T-machine is, of course, what is more commonly referred to as a *Turing Machine* (e.g. Minsky 1967).

One of Turing's major results was that, in a perfectly definite sense, certain particular A_T-machines can be so configured that they can *simulate* the (computational) operations of *any* A_T-machine—and can thus, in a definite sense, realise the same "computation" as any A_T-machine.

Turing called any A_T-machine having this property a *universal* (computing) machine. Von Neumann referred to this same property as "logical universality" (von Neumann 1966b, p. 92). It should be clear that this *concept* (though not any particular automaton) can be generalised across *any* A-system which supports a notion of "computing automaton", in the following way. Call any "computation" which can be carried out by some A-machine an *A-computation*; then, a "universal logical (computational) machine", which I shall call a *ULM*, is a single A-machine which, when suitably "configured", can carry out *any* A-computation.

Note carefully that (so far, at least), there is no claim about any relationship which might exist between A-computations (and thus ULMs) in *different* A-systems. The ULM concept is well defined only relative to a particular A-system (and especially the particular notion of A-computation incorporated in that A-system).

We may restate Turing's claim then as a specific claim for the existence of at least one ULM within the A_T-system—i.e. the existence of a ULM_T. Again, what I call a ULM_T is now most commonly referred to as a *Universal Turing Machine* (Minsky 1967). An essential concept in Turing's formulation of his ULM_T is that its operations are "programmed" by a list of "instructions" and that, as long as a fairly small basis set of instructions are supported, it is possible to completely describe the computational behaviour of an arbitrary A_T-machine in terms of a finite sequence of such instructions. That is, a ULM_T is made to simulate the computations of any arbitrary A_T-machine simply by providing it with an appropriately coded *description* of that machine.

In itself, Turing's claim for the existence of at least one ULM_T is neutral as to whether ULM's can or do exist in any other A-system, or whether "computing machines" in general share any interesting properties across different A-systems. These are important issues, which were central to the problem which Turing was attempting to solve. They will be taken up again in due course. For the moment, however, note that although von Neumann was inspired by Turing's work on the A_T-system, his *problem* was entirely different from Turing's problem; and, as a result, these issues prove to be more or less irrelevant to von Neumann's work.

Universal the First

Turing formulated the A_T-machines specifically as *computing* machines; the things which they can manipulate or operate upon are not at all the same kinds of things as they are made of. There are no such things as A_T-constructors or, more particularly, A_T-reproducers.

Von Neumann's basic idea was to generalise Turing's analysis by considering abstract machines which *could* operate on, or manipulate, things of the "same sort" as those of which they are themselves constructed. He saw that, by generalising Turing's analysis in this way, it would be possible to solve P_v in a very

definite, and rather elegant, way. In fact, von Neumann considered a number of distinct A-systems, which are not "equivalent" in any general way, and which were not always completely formalised in any case. However, a key thread running throughout all this work was to introduce something roughly analogous to the general concept of a ULM, but defined relative to some notion of "construction" rather than "computation". Von Neumann's new concept refers to a particular kind of A-machine which he called a *universal constructor*; I shall call this a "universal constructing machine", or *UCM*.

The analogy between the ULM and UCM concepts is precisely as follows. Like a ULM, the behaviour of a UCM can be "programmed", in a rather general way, via a list of "instructions". In particular, these instructions may provide, in a suitably encoded form, a *description* of some A-machine; and in that case, the effect of "programming" the UCM with that description will be to cause it to *construct* the described A-machine (assuming some suitable "environmental" conditions). Thus, just as a ULM can "simulate the computation of" *any* A-machine (when once furnished with a description of it), so a UCM should be able to "construct" *any* A-machine (again, when once furnished with a description of it, and, of course, always working within a particular formalisation of "A-machine", which is to say a particular A-system). We may trivially note that since there do not exist any A_T-constructors at all, there certainly does not exist a UCM_T, i.e. a UCM within the A_T-system.

I emphasise strongly here that it was precisely, and solely, the *spanning of all A-machines in a particular A-system* that mandated Turing's original usage of the word "universal" (in "universal machine", or ULM_T in my terms), and which therefore also mandated von Neumann's analogous usage (in "universal constructor", or UCM in my terms). The typical *operations* of the two kinds of machine (computation and construction, respectively) are, of course, quite different.

In Turing's original paper (Turing 1936) he argued, *inter alia*, that there exists a ULM_T. This is a technical, formal, result—a *theorem* in short—which Turing *proved* by actually exhibiting an example of a specific A_T-machine having this property. We shall see that von Neumann sought to achieve an essentially analogous, perfectly formal, result for a UCM—i.e. to prove the existence of such things, at least within some "reasonable" A-system, and to do so by precisely paralleling Turing's procedure, which is to say by actually exhibiting one. At this level, the analogy between these two developments is very strong and direct, and the word "universal" has a clearly related implication in both "UCM" and "ULM" within their respective domains. However, a problem arises because the "universal" in "ULM" actually admits of a number of quite distinct interpretations or connotations—only *one* of which is the one described above as being legitimately preserved in von Neumann's intended analogy. If one mistakenly supposes that any of the *other* connotations should be preserved (as well as, or instead of, the correct one) then the result can be serious confusion, if not outright error.

Universal the Second

The second interpretation of "universal"—and the first which it would be erroneous to impute to the UCM—revolves around the idea that what makes a ULM "universal" is not *just* that there exists *some* relationship between it and some

complete set of A-machines, but that there exists a very *particular* relationship—namely that of being able, when suitably programmed, to carry out the same A-computations. To put it another way, the "universality" of the ULM is seen to be *inseparably* bound up with the idea of "computation", so that it is not so much a matter of spanning a set of (A-)machines, but rather to be specifically about spanning a set of (A-)computations. Now this is not an entirely *un*reasonable interpretation of "universal"—as long as we restrict attention to U*LM*'s; because, in that case, it is entirely compatible with the original interpretation. However, in the case of a U*CM* this interpretation is deeply problematic. If we try to force it, we come up with something like this: given any (A-)computation, a UCM can, when suitably programmed, construct an A-machine which could, in turn, carry out that (A-)computation.

Now this is a most abstruse, and unlikely, interpretation. After all, von Neumann's whole point is to talk about automata which can construct automata *like* themselves; whereas, under this interpretation the definition of a UCM would make no reference at all to its ability to construct automata "like itself" (i.e. which could, in their turn, also construct further automata "like" themselves), but would instead talk about the ability of a UCM to construct automata of a *different* kind—namely, "computing" automata. Nonetheless, precisely this interpretation *has* been adopted in some of the literature, as we shall see. To explain how, and perhaps why, this arises, it is first useful to distinguish three variants on the idea, which differ in exactly how the "universal" set of "computations", which is to be spanned by the offspring of the UCM, is defined:

- In the simplest case, we assume that the A-system, in which the putative UCM exists, itself supports some definite notion of computation, which is to say it defines *some* set of A-computations. We then require only that the offspring of the UCM span this set. We place no *a priori* constraints or requirements on what should qualify as an "A-computation".

- In the second case, we require that the set of A-computations of the A-system be such that, in some well defined sense, for every A_T-computation there must be at least one "equivalent" A-computation. Assuming that such a relationship could somehow be established, we then require that the offspring of the UCM span some set of A-computations which is "equivalent" to the set of A_T-computations (this may, or may not, be the complete set of all A-computations). On this interpretation, a UCM is related not to the "general" notion of a ULM, but to the specific case of a ULM_T.

- Finally, we might require that the set of A-computations of the A-system be such that, in some well defined sense, for every "computation" of *any* sort, which can be effectively carried out at all, there must be some "equivalent" A-computation. Assuming, again, that such a relationship could somehow be established, we then require that the offspring of the UCM span some set of A-computations which is "equivalent" to the set of all effective computations (and again, this may, or may not, be the complete set of all A-computations).

I refer to all three of these (sub-)interpretations of the "universal" in UCM as being "computational". The first two of these could, in principle at least, be

completely formalised in particular A-systems, so that the existence of a UCM in these (somewhat peculiar) senses would, at least, be a matter of fact, which might admit of proof or disproof. However, the third computational interpretation relies on the informal notion of what constitutes an "effective computation", and will always be a matter of opinion or convention rather than fact; there is no possibility of the existence (or otherwise) of a UCM, in *this* sense, being decisively established for *any* A-system.

Having said that, Turing (1936) argued (informally, of course) that the A_T-system already captures everything that could "reasonably" be regarded as an effective computation. This proposal is now referred to as the *Church-Turing thesis*. Due to its necessarily informal nature, it is a *thesis* not a *theorem*; nonetheless it is now widely regarded as being well founded (e.g. Minsky 1967, Chapter 5).

Now *if* the Church-Turing Thesis is accepted, then the third (computational) interpretation of UCM becomes exactly equivalent to the second. Indeed, one may say that the only reasonable basis for introducing the second computational interpretation at all is on the understanding that the Church-Turing thesis holds, because this implies that the A_T-computations provide an absolute benchmark of *all* kinds of computation. If this were *not* the case, then it would appear rather arbitrary to single out *this* set of computations for special significance relative to the notion of UCM.

More generally, it seems to me that it is *only* in the context of the Church-Turing Thesis that a strictly computational interpretation of the "universal" in UCM suggests itself at all. The point is that a ULM_T is (by definition) capable of carrying out all A_T-computations; and therefore, under the conditions of the Church-Turing Thesis, a ULM_T is, in fact, capable of carrying out all effective computations. We should perhaps say that a ULM_T is *doubly* universal: it is firstly universal with respect to all A_T-computations (which gave it its original title); but this then turns out (at least if the Church-Turing Thesis is accepted) to mean that it is universal with respect to the computations of *any* effective computing system whatsoever, not "just" those of the A_T-system. To make this completely clear, we should perhaps refer to a UULM, or U^2LM; but, since there is apparently no conflict between these two distinct attributions of universal (i.e. since the Church-Turing Thesis asserts that they are synonymous) it has become conventional not to bother to distinguish them; the single "U" in ULM_T (i.e. in "universal Turing machine") is, today, flexibly interpreted in either or both of these two senses, as the context may demand, without any further comment. I suggest that it is *only* because these two connotations of "universal" in ULM_T are not normally distinguished, that a strictly computational interpretation of "universal construction", or UCM, (i.e. any of the three such interpretations I have distinguished above) is typically entertained at all.

I stated that computational interpretation(s) of UCM have appeared in the literature. It is not always possible to isolate exactly which of the three identified sub-cases are intended. In any case, the most explicit (and, to the best of my knowledge, the earliest) advocate of a computational view of the UCM concept is E.F. Codd, and his proposal is quite precise, corresponding exactly to what I identified above as the second computational interpretation:

> The notion of construction universality which we are about to formalise demands of a space the existence of configurations with the ability to con-

> struct a rich enough set of computers such that with this set any Turing-computable partial function on a Turing domain can be computed in the space.
>
> Codd (1968, p. 13)

Codd's interpretation of UCM has been explicitly repeated by Herman (1973). Langton (1984) does not explicitly endorse Codd's interpretation as such, but does equate Codd's concept with von Neumann's, which I consider to be mistaken.

I should admit that the position, typified here by Codd, is not quite as perverse as I have painted it. Codd had special reasons for his particular approach,[2] and, even aside from these, it *can* ultimately prove useful to say something about the "computational" powers of A-constructors and/or their offspring, in the overall solution schema for P_v (McMullin 1992, Chapter 4). However, my claim is that such powers need form no part of the essential *definition* of the UCM concept; in particular, they seem to be no part of von Neumann's *analogy* between the ULM and the UCM. While Codd's definition cannot be said to be "wrong", it is certainly *different*, in a substantive way, from von Neumann's; more seriously, it seems to me that adopting such an interpretation fatally undermines von Neumann's proposed solution to P_v. To see why this is so, note that the Church-Turing thesis was proposed for a very definite reason. Both Church and Turing were attempting to solve the so-called *Entscheidungsproblem*, the *decision* problem of (meta-)mathematics. The statement of this problem explicitly referred to the (informal) notion of a "definite method", or an "effective procedure" as it is now called; thus Turing's work could conceivably be regarded as a solution of this problem *only* if the Church-Turing thesis were accepted. The thesis was thus absolutely central and essential to Turing's analysis. Von Neumann's problem, on the other hand (at least in my formulation as P_v), makes *no* reference whatsoever to computation, "effective" or otherwise; so I suggest that the Church-Turing thesis, and *computational* universality as such, can have no *essential* rôle to play in its solution.

Universal the Third

I now come (briefly) to a third conceivable interpretation of "universal" (in UCM). This again involves the Church-Turing thesis, but in a way which is quite different from the strictly computational interpretations just outlined. Roughly speaking, the Church-Turing thesis says that the computations of which A_T-machines are capable are universal with respect to *all* computational systems—regardless, for example, of their "material" structure. We could thus attempt to carry over this whole thesis, through von Neumann's analogy, to say something, not about *computational* systems in general, but *constructional* systems in general.

The point here is that the analogy between the ULM and UCM concepts is so strong that one might be easily lulled into supposing that there *is* some obvious generalisation of the Church-Turing thesis; which would imply, in turn, that a

[2] He was *inter alia* interested in the the design of massively parallel *computers*.

UCM, in *any* "sufficiently powerful" A-system, captures something important about the powers of *all* automata, in *all* formal frameworks, and, by implication, about the powers of all "real" (physical) automata. It is important to emphasise that von Neumann himself never asserted, much less argued for, any such thesis; and that, for what it is worth, it seems unlikely (to me) that such a thesis could be defended. Conversely, to *assume* that some such thesis holds will be confusing at the very least, and also liable to lead to actual error in interpreting the implications of von Neumann's work.

As far as I am aware, no worker has ever *explicitly* argued for such a generalisation of the Church-Turing thesis—but there are some indications of its having been at least implicitly assumed. Thus, Thatcher (1970, pp. 153, 186) makes passing reference to such a possibility, though he does not explore it in any detail. More substantively, while Tipler (1981; 1982) does not explicitly mention the Church-Turing thesis, he does interpret von Neumann's work as having extremely wide-ranging applicability, well outside anything actually mentioned by von Neumann himself. In brief, Tipler cites von Neumann as establishing that a "real", physical, UCM, which can construct *any* physical object or device whatsoever (given an appropriate description, sufficient raw materials, energy, and, presumably, time), can be built. I suggest that such a claim must implicitly rely *inter alia* on something like a generalised Church-Turing Thesis; it is, in any case, directly contrary to von Neumann's comment, in discussing the general nature of his theory, that "Any result one might reach in this manner will depend quite essentially on how one has chosen to define the elementary parts" (von Neumann 1966a, p. 70).

Conclusion

Von Neumann introduced the notion of a UCM, by analogy with Turing's ULM_T, as a particular kind of A-machine which could, when suitably programmed, construct *any* "machine"; but this notion only becomes precise in the context of a particular formalisation of "machine", i.e. a particular A-system. I claim that the UCM concept, as originally formulated by von Neumann, does not *inherently* involve any comment about the "computational" powers either of itself or of its offspring, and does not involve or imply any "natural" generalisation of the Church-Turing Thesis.

Acknowledgements

This paper is a severely abridged version of selected material first presented in Chapter 4 of my Ph.D. Thesis (McMullin 1992). My Ph.D. work was made possible by generous support from the School of Electronic Engineering in Dublin City University. I am grateful to Noel Murphy of DCU, and John Kelly of University College Dublin, for discussions relating to the paper. Chris Langton of the Los Alamos National Laboratory was also generous in responding to my correspondence. The ECAL-93 reviewers provided very useful criticism of an earlier draft, and I hope the paper has now been improved as a result.

References

Burks, A. W. 1966a. Automata Self-Reproduction. *Pages 251–296 of:* (Burks 1966b).

Burks, A. W. (ed). 1966b. *Theory of Self-Reproducing Automata [by] John von Neumann.* Urbana: University of Illinois Press.

Burks, A. W. (ed). 1970a. *Essays on Cellular Automata.* Urbana: University of Illinois Press.

Burks, A. W. 1970b. Von Neumann's Self-Reproducing Automata. *Pages 3–64 of:* (Burks 1970a).

Codd, E. F. 1968. *Cellular Automata.* New York: Academic Press Inc.

Herman, G. T. 1973. On Universal Computer-Constructors. *Information Processing Letters*, **2**, 61–64.

Langton, C. G. 1984. Self-Reproduction in Cellular Automata. *Physica*, **10D**, 135–144.

McMullin, B. 1992. *Artificial Knowledge: An Evolutionary Approach.* Ph.D. thesis, Ollscoil na hÉireann, The National University of Ireland, University College Dublin, Department of Computer Science.

Minsky, M. L. 1967. *Computation: Finite and Infinite Machines.* New Jersey: Prentice-Hall Inc.

Moore, E. F. 1970. Machine Models of Self-Reproduction. *Pages 187–203 of:* (Burks 1970a).

Poundstone, W. 1985. *The Recursive Universe.* Oxford: Oxford University Press.

Smith, A. R. 1992. Simple Nontrivial Self-Reproducing Machines. *Pages 815–838 of:* Langton, C. G., *et al.* (eds), *Artifical Life II.* California: Addison-Wesley Inc.

Taub, A. H. (ed). 1961. *John von Neumann: Collected Works. Volume V: Design of Computers, Theory of Automata and Numerical Analysis.* Oxford: Pergamon Press.

Thatcher, J. W. 1970. Universality in the von Neumann Cellular Model. *Pages 132–186 of:* (Burks 1970a).

Tipler, F. J. 1981. Extraterrestrial Intelligent Beings Do Not Exist. *Physics Today*, **32**(4), 9, 70–71.

Tipler, F. J. 1982. We are Alone in our Galaxy. *New Scientist*, **96**(1326), 33–35.

Turing, Alan. 1936. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proc. London Math. Soc.*, **Series 2, Vol. 42**, 230–265.

von Neumann, J. 1951. The General and Logical Theory of Automata. *Chap. 9, pages 288–328 of:* (Taub 1961). First published 1951 as *pages 1–41 of:* L. Jeffress, A. (ed), *Cerebral Mechanisms in Behavior—The Hixon Symposium*, New York: John Wiley.

von Neumann, J. 1966a. Theory and Organization of Complicated Automata. *Pages 29–87 of:* (Burks 1966b). Based on lectures delivered at the University of Illinois, in December 1949. Edited by A.W. Burks.

von Neumann, J. 1966b. The Theory of Automata: Construction, Reproduction, Homogeneity. *Pages 89–250 of:* (Burks 1966b). Based on an unfinished manuscript by von Neumann. Edited by A.W. Burks.