Chapter 5

Artificial Genesis

5.1 Introduction

> ... AI as a field is starving for a few carefully documented failures. Anyone can think of several theses that could be improved stylistically and substantively by being rephrased as reports on failures. I can learn more by just being told why a technique won't work than by being made to read between the lines.
>
> McDermott (1976, p. 159)

I have argued that the central outstanding problem in the realisation of a substantive growth of A-knowledge via a process of Artificial Darwinism is that of exhibiting A-machines which are not only self-reproducing but also *robust*, in the face of a "hostile" environment. By "self-reproducing" I mean, of course, the von Neumann sense of supporting "heritable mutation"; that is, our A-reproducer should be a member of a set of A-reproducers, which span an indefinitely large range of A-complexity, where this set is connected under some form of A-mutation. The von Neumann schema of *genetic* self-reproduction shows, in outline at least, how this condition can be satisfied.

This outstanding problem, which I have labelled P_a, is still very informal; nonetheless, I have suggested that, to date, there has been little tangible progress toward a solution. Without attempting to prejudge the ultimate prospects for solving P_a by the "direct" route (i.e. by directly designing robust A-reproducers within some "reasonable" A-system) I have pointed out that there may be an alternative "indirect" approach—namely attempting to exhibit the *spontaneous* emergence of (viable) A-reproducers. The latter approach is inspired by the

(conjectured) spontaneous genesis of life in the biological world.

This chapter is concerned with a critical investigation of a class of A-systems which (it has been suggested) might indeed exhibit something like the spontaneous origin of Artificial Life.[1] I should warn in advance that the results to be presented here are largely negative: it will turn out that, contrary to expectations, the single specific A-system which will be examined in detail *cannot* support phenomena of this sort. However, I shall argue that the mechanisms of failure are not without interest.

The A-system presented here is an example of an α-Universe. The α-Universes are a class of artificial system originally proposed by Holland (1976). Holland made this proposal in a particular context, related to, but by no means identical with, my P_a. It is therefore useful to briefly review the problem situation which Holland *intended* to address with the α-Universes.

Holland's stated objective was to rebut certain criticisms of the neo-Darwinian interpretation of evolution. The situation was roughly as follows (for a more detailed discussion Holland cites Moorhead & Kaplan 1967):

- Darwinian evolution is predicated on the prior existence of entities having a wide behavioural repertoire which includes, among other things, the ability to self-reproduce in a manner which supports heritable mutation. Following Gould, I have previously called such entities Darwinian actors, or simply *D-actors* (Gould 1982; McMullin 1992a).

- Biological Darwinism should therefore be accompanied by some complementary theory to explain the advent of the *initial* D-actors in the so-called "primordial soup". I stress that the problem being presented here is *not* that of Biological Darwinism itself—i.e. whether the latter provides an adequate theory of the growth of biological complexity *once an initial set of biological D-actors is postulated*. It is the prior problem of whether Darwinian processes could have (spontaneously) started in the first place: the problem of the original genesis of life.

[1] A preliminary version of some of the material presented in this chapter has been previously published (McMullin 1992d). However, the treatment given here is much more detailed and extensive, and includes more recent experimental results: it may be regarded as the definitive version.

- This complementary theory should not draw on any new causal principles, over and above those assumed by biological Darwinism in the first place (it should not, for example, be theistic)—for otherwise, Biological Darwinism itself would evidently be undermined.

- A first (and naïve) tentative solution is the conjecture that, prior to the emergence of the initial D-actors, conventional physical effects (thermal and electrical agitation of the unorganised chemical soup), will result in the generation of a wide variety of physically feasible structures in an "unbiased" manner (i.e. we do not suppose that D-actors are any more likely to be generated than other structures of comparable "size"). Provided sufficient time is available, this process might eventually result (with probability approaching one?) in the emergence of the required, initial, D-actors. Darwinian evolution then takes over (or not, as the case may be; that is a separate problem).

- This conjecture seems, however, to be refuted by quantitative calculations of the expected time to emergence of D-actors, based on such an unbiased search process.[2] Even allowing for a substantial margin of uncertainty in the parameters for these calculations, the result is an emergence time so large that it seems entirely incompatible with such emergence having occurred in the lifetime of planet Earth.

- The next proposed solution is to retain the conjecture that conventional physical effects will result in the generation of a wide variety of structures, but to suppose that this generation process may be (or may become) strongly *biased.* In particular, in analogy with conventional Darwinian theory, it is conjectured that there may be *incremental* progress toward fully qualified D-actors. That is, there may be structures, typically much simpler than the ultimate D-actors (i.e. much simpler than even the simplest of contemporary *organisms*), which thus have plausible emergence times, but which might still have a long term effect on the subsequent generation

[2]This is a variant of the infamous "monkeys typing Shakespeare" kind of argument (e.g. Dawkins 1986, pp. 46–49).

process—biasing it in such a way that fully qualified D-actors can, feasibly, have emerged within the available time. To put it another way, we abandon any notion of a strict, binary distinction between structures which are "D-actors" and those which are not, and accept that there may be a continuum. Instead of supposing that there was a more or less dramatic or catastrophic change between unbiased generation of structures, and Darwinian evolution of structures, we suppose that what we now recognise as Darwinian evolution may have emerged gradually, reinforcing itself as it became established. We might call this a *bootstrap* theory of the emergence of D-actors (and Darwinian evolution).

- This serves to rescue a materialist theory of the origin of life and thus, of Biological Darwinism itself; but at the cost of becoming vague, qualitative, and, in this form, virtually untestable. In Popperian terms, it has become, not so much a theory, as a *metaphysical research program* (e.g. Popper 1976, Section 33, pp. 148–151). That is, it is a framework for the development of detailed theories, which detailed theories *might* then be capable of making testable predictions.

- It is extremely difficult to improve on this situation. Ideally, we would analyse and/or simulate or duplicate a quantitative model of the dynamics of the postulated primordial "soup" and thereby formulate a detailed, quantitative, testable, theory of the emergence of fully qualified D-actors. Indeed, considerable effort has been expended along these lines, with some degree of success (e.g. Oparin 1953; Eigen & Schuster 1979; Dyson 1985); however, these efforts are seriously limited by the size and complexity of the system (the primordial soup) being investigated. Thus: the detailed composition of the system is quite uncertain; the basic chemical interactions are varied, complex, and non-linear; the system is extremely large (say the total number of active chemical components on the planet Earth, during the pre-biotic epoch), and the duration potentially available for the significant processes is extremely long.

- Holland proposed an alternative approach (still within the same basic metaphysical research program). This is to investigate the behaviour of very much simplified systems, in order to provide a "proof-of-principle". The idea is this: suppose we can formulate a relatively simple system, which is, nonetheless, capable of demonstrating analogous phenomena to those being postulated for the primordial soup. Specifically, the system should be such that some form of D-actors can be sustained in the system, if they once emerge; that there is some kind of continuum of behaviour from that of simpler structures to that of fully qualified D-actors; and that, in the absence of bias due to the behaviours of the structures already present, there will be an unbiased generation of structures over some large set of "feasible" structures. We can calculate the naïve emergence time for D-actors, in an analogous manner to that for the "real" primordial soup; but additionally, if the system is sufficiently simple, we may be able to demonstrate analytically and/or experimentally, that D-actors can actually emerge in a much shorter time, by virtue of the generation process becoming progressively biased. If this could be achieved, it would constitute a proof of the principle that D-actors, and Darwinian evolution, could establish themselves spontaneously. This would not, of itself, "prove", or even "verify", the theory that some analogous process occurred in the real primordial soup; but it would increase our *preference* for such theories, by refuting the implicit alternative (that such behaviours are entirely impossible).

- To this end, Holland introduced the α-Universes as a class of simple A-system which could exhibit at least some of the required properties; he went on to identify one particular α-Universe (which I shall denote α_0) for which he was able to present detailed, closed form, analytic results. On Holland's analysis, α_0 demonstrates precisely the result sought: simple D-actors appear to have an expected emergence time much shorter than would be predicted by a naïve assumption of unbiased generation.

At first sight, Holland's analysis (if it survives critical testing) would solve not just the problem he was directly addressing, but P_a also: for, as I have described the situation, Holland seems to claim, *inter alia,* that α_0 can support D-actors—

i.e. *identified*, robust, A-reproducers. The fact that such D-actors should emerge spontaneously would, in that scenario, be an added bonus, but would be quite inessential to the solution of P_a.

However, on closer analysis the situation proves to be rather more complicated than this. While Holland does identify putatively robust, "self-replicating", A-machines, which can be embedded in α_0, these are not properly self-reproducing in the von Neumann sense; although they do involve a von Neumann style genetic mechanism, the complete set of related A-reproducers is essentially trivial—it certainly does *not* span a wide range of A-complexity.[3] Thus, α_0 certainly cannot offer an *immediate* solution to P_a.

The real relevance of α_0 to P_a is the following: if Holland's analysis of α_0 is correct, then it suggests that some more "powerful" α-Universe might support a set of D-actors (A-reproducers) spanning a satisfactory range of A-complexity, *while still retaining the property that such D-actors would spontaneously emerge.* If this were so, it might allow, as previously anticipated, an experimental solution of P_a *without* the need for an *a priori* design of any initial, robust, A-reproducers.

The question which immediately arises is whether Holland's analysis of α_0 is, in fact, correct. Although α_0 is extremely simple compared to real chemical systems, its analysis is by no means trivial, and could conceivably be mistaken. Holland therefore noted that his analysis could feasibly be tested by instantiating an α-Universe in a suitable, high speed, digital computer. However, Holland himself did not report on such tests, and, as far as I am aware, no such test program was ever carried out (Holland, Langton, personal communication).[4]

This chapter will therefore present original results from just such a program. I preface this with a more detailed and formal definition of α_0 than that originally presented by Holland, and an account of relevant aspects of the particular implementation.

As already indicated, these results will be negative: it transpires that Holland's analysis was indeed mistaken (through being oversimplified). In fact, even

[3] In this respect, the A-reproducers in α_0 are essentially similar to an A-reproducer proposed by Langton (1984) (in a rather different A-system); see my previous discussion in Chapter 4, section 4.2.7.

[4] Indeed, I have been able to identify only two substantive discussions of any kind of (Holland 1976). I shall return to these in section 5.5.8 below.

the extremely impoverished D-actors proposed by Holland for α_0 prove *not* to be robust; the question of their spontaneous emergence (never mind the emergence of more powerful D-actors in some alternative α-Universe) is thereby rendered irrelevant. This outcome will serve principally to reiterate again the seemingly intractable nature of P_a; but it will nonetheless also suggest some useful new insights into the problem.

5.2 The Universe α_0

5.2.1 Outline

Firstly, let me note that α_0 is not strictly a single, unique, α-Universe but denotes instead a parameterised family of related α-Universes. I shall identify such parameters as they arise, but otherwise it will be convenient to continue to refer to α_0 in the singular.

I should emphasise that Holland's original definition of α_0 was not complete; that is, many detailed aspects of its operation were left unspecified. The implication is that these details should not affect the ultimate outcome; nonetheless, in any realisation of α_0 it is still necessary to fill in all such details in some particular way. This section thus serves both to re-present Holland's original definition and also to specify, in detail, how this definition was extended and completed to allow a practical realisation.

Loosely speaking, α_0 consists of some fixed number of discrete *atoms*.[5] The total number of atoms is a parameter of α_0, and is denoted R; in general, Holland does not stipulate a specific size in his analysis. While he does refer to what he calls a "region" he gives no precise definition of "region". The experimental work described below will be based on a total size of $R = 10^4$ atoms, this being the size of "region" used by Holland for numerical calculations.

Atoms are classified into six distinct kinds, or *elements*. Of these, one has an especially distinguished rôle, and is referred to as the *null* element; the remaining

[5]Holland strictly speaks in terms of an underlying "cellular automaton", each cell of which can effectively "contain" one atom. The cells then remain "fixed" while the atoms "move" among the cells. However, this underlying cellular automaton *per se* plays no rôle either in Holland's analysis or the implementation to be described here; further discussion of it is therefore omitted.

five elements are collectively referred to as *material* elements. α_0 supports a detailed principle of "matter" conservation, in that atoms cannot be transmuted from one element to another, and thus the numbers of atoms of each element remain constant. The "densities" of each element (the number of atoms of the element divided by the total number of atoms R) are thus further parameters of α_0.

Each atom has, associated with it, one *bond*, connecting it to one other atom. A bond may be in either of two states: *strong* or *weak*. This state may be dynamically altered under the action of certain α_0 operators (with the exception that a bond originating with a null atom, or connecting to a null atom, cannot become strong).

As long as a given bond is strong, it cannot be disconnected. By contrast, a weak bond, may, in certain circumstances, be disconnected and re-connected to a different atom; in this way the connections between atoms may change in time. However, it is characteristic of α_0 that bonds may only be "transiently" disconnected; that is, every operator which involves disconnecting a bond also involves re-connecting it (to a different atom) all within a single time step. It follows that, before and after the operation of any allowed α_0 operator, every atom must be connected to precisely one other atom, and, therefore, that all the atoms in α_0 must form a single connected chain. Assuming that the number of atoms (R) is finite (as it must be for any practical realisation) this further implies that the chain of atoms must be closed on itself—i.e. it must form a single closed *loop*.

Any arbitrary connected series or sequence of atoms in α_0 will be called a *segment*. In effect, all α_0 operators which change the relative ordering of the atoms will do so in two stages (both completed in a single time step): a segment (which may consist of a single atom) is first cut out of one part of the loop, thus *transiently* dividing α_0 into a (smaller) loop and a separate, disconnected segment; this disconnected segment is then spliced back in, at some other point, reforming α_0 into a single closed loop of R atoms again.

A segment consisting of a null atom, followed by one or more material atoms, and then terminating in another null atom, is called a *structure*; a structure

containing exactly one material atom will also sometimes be referred to as a *free atom*.

A *complex* is a set of interacting structures: it is (for the time being at least) the kind of entity which we shall recognise as an *A-machine* in α_0. The structures making up a complex need not, in general, have definite connections with each other (a complex is not a segment *per se*); however, for a complex to exhibit interesting properties it is generally necessary that all the component structures be more or less "close" to each other.

The α_0 dynamics progress in discrete time steps; that is, the operators are all defined in terms of their effect in a single such time step.

Holland anticipated that, for a computer realisation of α_0, a time step might be accomplished in about 1ms of real time; however, he gave no indication of the kind of platform he assumed to achieve this. In any case, in the experiments to be described below, a more typical value actually achieved was of the order of 500 ms per time step (based on an Intel 80386 CPU with 33 MHz clock)—though this varied very considerably with the actual state of the universe.

The dynamic behaviour of α_0 is stochastic, and is defined in terms of two groups of operators: the *primitive* operators, and the *emergent* operators. The primitive operators are context *insensitive*—i.e. they apply throughout α_0 without regard to its sequential configuration. They are the abstract counterparts of diffusion and activation in real chemical systems. The emergent operators are context *sensitive*—i.e. their operation is sensitive to the sequential configuration of α_0. In effect, certain structures (should they arise) have special dynamic properties. They are termed "emergent" operators precisely because they are contingent on such structures—they "emerge" iff some matter in α_0 "happens" (under the action of the primitive operators or otherwise) to adopt some such special configuration. These are the abstract counterparts of catalysts (particularly enzymes) in real (bio-)chemical systems.

In the study of real chemical systems it is of interest to seek an explanation of the properties and characteristics of catalysis in terms of more fundamental (atomic) interactions. However, for the particular uses we wish to make of the α_0 dynamics, such a more fundamental analysis would be superfluous, and is not attempted. Instead, the properties of emergent operators are simply imposed by

fiat.[6] I may note in passing that the notions of matter conservation and coherent movement (of strongly bonded material segments) make α_0 somewhat reminiscent of von Neumann's (1966a) *kinematic* A-system—though α_0 is, of course, very much simpler.

Holland takes "self-replication" as diagnostic of "life"; the dynamics of α_0 are such that certain complexes (should they arise) may exhibit primitive (but still loosely *genetic*) self-reproducing behaviours.

5.2.2 A Little Formality

In what follows, I shall freely use relevant terminology and notation from the formal theory of computation, as presented, for example, by Lewis & Papadimitriou (1981, especially Section 1.8).

Atoms in α_0 are *formally* defined as *symbols*; the closed loop of atoms is defined as a *string* of exactly R such atomic symbols, which will be referred to as the *state string*; segments and structures are also *strings* (of length less than R) over this same atomic symbol alphabet, normally occurring as *substrings* of the state string;[7] and the operators are *production rules* specifying particular transformations of the state string.

In more detail, the alphabet of atomic symbols is defined as $Z = X \times Y$ (i.e. Z is the cartesian product of two "simple" alphabets X and Y), where:

$$\begin{aligned} X &= \{\mathtt{0}, \mathtt{1}, :, N_0, N_1, -\} \\ Y &= \{\mathtt{s}, \mathtt{w}\} \end{aligned}$$

$$\begin{aligned} \Rightarrow Z = \{ \quad & (\mathtt{0}, \mathtt{s}), (\mathtt{1}, \mathtt{s}), (:, \mathtt{s}), (N_0, \mathtt{s}), (N_1, \mathtt{s}), (-, \mathtt{s}), \\ & (\mathtt{0}, \mathtt{w}), (\mathtt{1}, \mathtt{w}), (:, \mathtt{w}), (N_0, \mathtt{w}), (N_1, \mathtt{w}), (-, \mathtt{w}) \quad \} \end{aligned}$$

[6] This is, in itself, an unusual and interesting (metaphysical) position. The α_0 dynamics might be said to be *irreducible*, to the extent that the properties and behaviours of structures in α_0 are not reducible to properties or behaviours of their "constituent" atoms. However, it should be added that this is still a very weak form of irreducibility, compared to, say, Rosen's (1985b) "complex" systems or Popper's Worlds 1, 2 and 3 (e.g. Popper & Eccles 1977).

[7] Strictly, a segment or structure will not satisfy the technical definition of a "substring" if it spans the atom which is (arbitrarily) designated as the initial atom of the state string. It should be clear that this can be overcome by a minor adjustment to the formal definition of "substring" and this technicality will not, therefore, be discussed further.

We see that each atom (i.e. each atomic symbol) is actually an *ordered pair* of simple symbols, the first taken from the X-alphabet (denoting the element) and the second from the Y-alphabet (denoting the bond state). The state string is then a string of these atoms where each atom is (implicitly) bonded to the next atom to the right. The state string will, of course, be exactly R atoms (ordered pairs) in length.

For many purposes in discussing the α_0 dynamics it will be necessary to refer to just the elements (X-symbols) or the bond states (Y-symbols) in a segment. Two functions are introduced to facilitate this. The first, denoted $\chi()$, extracts the X-symbols from a segment; the second, denoted $\varphi()$, extracts the Y-symbols from a segment. that is, reading the segment (Z-string) from left to right, $\chi()$ maps each Z-symbol onto its X-component:

$$(x, y) \mapsto x$$

and, similarly, $\varphi()$ maps each Z-symbol onto its Y-component:

$$(x, y) \mapsto y$$

5.2.2.1 The Elements

Certain of the elements have similar or related characteristics with respect to the α_0 dynamics. It will therefore prove convenient to group the element (X) symbols into several partially overlapping families or (sub-)alphabets as follows:

$$\begin{aligned} N &\equiv \{N_0, N_1\} \\ A &\equiv \{\mathtt{0}, \mathtt{1}, \mathtt{:}\} \\ B &\equiv \{\mathtt{0}, \mathtt{1}\} \\ M &\equiv N \cup A \\ D &\equiv M - \{\mathtt{:}\} \end{aligned}$$

"–" identifies the *null* element, previously mentioned, and is not a member of any of these sub-alphabets.

The N-alphabet serves primarily in the construction of more or less static data storage structures (similar in concept to the A-tapes of the previous chapter); the name N indicates a crude analogy to the function of *nucleotides* in molecular

biology. The A-alphabet serves primarily in the realisation of active structures (emergent operators); the name A indicates a crude analogy to the function of *amino acids*. The B-alphabet is a subset of the A-alphabet; this alphabet is used within emergent operators to code for the operator type and arguments. The name B is a mnemonic for *binary*. The N-alphabet is also, of course, a form of binary alphabet: we shall see that these two distinct binary alphabets are closely related, and this fact partially motivated the particular choice of symbols to represent them. The M-alphabet serves simply to group all the material elements (i.e. as a more concise name for $N \cup A$); the name M is a mnemonic for *material*. Finally, the D-alphabet groups the material elements *other* than the colon element ("`:`"); the name D has no mnemonic significance whatever.

Henceforth I shall refer to atoms whose X-symbol is from the A-alphabet as A-atoms, those whose X-symbol is from the N-alphabet as N-atoms etc. Similarly, a segment consisting exclusively of A-atoms will be called an A-segment etc.

The densities of the separate elements (number of atoms of that element divided by the total number of atoms, R) are parameters of α_0, denoted $\rho(\mathtt{0}), \rho(\mathtt{1})$ etc. The total density of the material elements (M-atoms) is denoted simply by ρ; we must therefore have $\rho(-) = (1 - \rho)$. Typical numerical values, suggested by Holland, which will be used in the empirical investigation are as follows:

$$\rho = \rho(-) = 0.5$$
$$\rho(\mathtt{0}) = \rho(\mathtt{1}) = \rho(:) = \rho(N_0) = \rho(N_1) = 0.1$$

5.2.2.2 The Bond States

Every atom in α_0 is bonded to the next atom to the right. The state of the bond is denoted by the Y-symbol of each atom: "`s`" denoting a *strong* bond and "`w`" denoting a *weak* bond. Note that a strong bond cannot connect with a null atom; this fact constrains the state string in two distinct ways. Firstly, a null atom cannot originate a strong bond, which is to say that the atom $(-, \mathtt{s}) \in Z$ cannot, in fact, arise in α_0. Thus, null atoms will actually occur in α_0 *only* in the form $(-, \mathtt{w})$; we now give this distinguished atomic symbol the special name $z_{\_}$. Secondly, a null atom cannot terminate a strong bond and thus no segment can arise in α_0 consisting of an atom with a strong bond immediately followed by a null atom (i.e. a segment of the form $zz_{\_}$ where $\varphi(z) = \mathtt{s}$).

5.2.3 The Primitive Operators

There are two primitive operators: Bond Modification (`BM`) and Exchange (`EX`). `BM` is the abstract counterpart of activation, and `EX` is the abstract counterpart of diffusion.

5.2.3.1 Bond Modification (`BM`)

`BM` was originally defined by Holland as follows: on each time step, every bond state in α_0 is (stochastically) updated: each strong bond decays (becoming weak), with probability r; each weak bond becomes strong with probability λr. r and λ are parameters of α_0.

However, as it stands, this is not consistent with the proviso, already stated, that a strong bond cannot connect with a null atom. This does not affect the decay aspect, from strong to weak; but the transformation of weak bonds to strong must strictly be qualified as applying *only* to bonds connecting material atoms (all other bonds, namely the weak bonds connecting null atoms to each other or to material atoms, are thus unaffected by `BM`).

More formally, `BM` is defined in terms of two stochastic transformations, or production rules, affecting the state string. The first is the decay from strong to weak:

$$(x, \mathtt{s}) \mapsto (x, \mathtt{w}),\ x \in X$$

This is applied, with probability r, to every atom matching the left hand side (i.e. every atom having a strong bond). The second is the transformation from weak to strong:

$$(x_a, \mathtt{w})(x_b, y) \mapsto (x_a, \mathtt{s})(x_b, y),\ x_a, x_b \in M,\ y \in Y$$

This is applied, with probability λr, to every segment matching the left hand side. r represents bond "stability" (i.e. the probability of a bond decaying from strong to weak). Thus, once a strong bond forms, its lifetime is simply a geometric random variable with parameter r, and the *expected lifetime* of a strong bond is just $1/r$ (neglecting the possible effects of operators other than `BM`). The typical numerical value used is $r = 10^{-4}$ giving an expected lifetime of a strong bond

of 10^4 time steps. There is no directly analogous result for *weak* bonds because a given weak bond would most likely not be eligible for modification to strong (because it connects to a null atom, transiently or otherwise) throughout its lifetime—and therefore its lifetime could *not* be modeled by any simple geometric random variable.

λ determines (roughly) the "equilibrium ratio" (fixed point of the Markov process implied by `BM`) between weak bonds and strong bonds—*provided* this is interpreted as referring only to bonds connecting material atoms (i.e. which *are* eligible to be strong). Holland argues (in his Theorem 1^8) that the distribution of structures generated by primitive operators, in isolation, will be *unbiased* (in a precise sense, defined by Holland) iff λ is specified as follows:

$$\lambda \simeq \frac{5(1-\rho)}{3\rho^2} - 1$$

The typical numerical value for λ therefore follows from the value already specified for ρ (0.5), yielding $\lambda = 7/3$. Thus, among all bonds eligible to be strong, approximately two thirds will be expected to be strong, and one third weak, at any given time (at least to the extent that this ratio is determined by `BM`).

In any case, note that the formation and decay of strong bonds are only stochastically related (as represented by λ). The number of strong (or weak) bonds is *not* constant in α_0. To give an extreme example, there is a very small (but still non-zero) probability that, even within a single step, *all* bonds could become weak; or all eligible bonds could become strong, for that matter.

5.2.3.2 Exchange (`EX`)

The function of `EX` is to provide for a randomised motion or relative rearrangement of the atoms in α_0, with the proviso that any segment consisting exclusively of (necessarily material) atoms which are strongly bonded together will move as a unit.

[8] I should warn that this theorem relies, in turn, on Holland's Lemma 2, and that there are grounds for thinking that the latter is mistaken, both in its result and its derivation—see also section 5.5.2 below. However, it will ultimately be clear that nothing critical relies on this, so I shall accept Holland's analysis at face value just here.

In brief, the idea is that, on each time step, some pairs of adjacent segments, having weak external bonds, exchange positions. The internal bonds of a segment being exchanged may be weak or strong. No bond *states* (internal or external) are altered by the `EX` operator; what *is* altered is the connectivity between atoms.

However, the details of the `EX` operator are somewhat complex, as follows.

On each time step, each atom with a weak bond serves, with probability m_1, as the *pivot* for an *exchange operation*. An exchange operation consists of two steps. Firstly, two other atoms with weak bonds are identified as the left and right *limits* of the exchange. This is done by considering first the next atom with a weak bond, to the right of the pivot; this is selected as the right limit, with probability m_2; if it is not selected, then the next atom with a weak bond to the right again is similarly considered, and so on until a limit is determined. The left limit is then established by counting the *same* number of atoms with weak bonds, to the left of the pivot.[9]

In this way, two disjoint, contiguous segments are identified: the left segment consists of all atoms between the left limit and the pivot (excluding the limit, but including the pivot); the right segment consists of all atoms between the pivot and the right limit (excluding the pivot, but including the limit); From the definition, the external bonds of these segments—being those of the left limit, the pivot, and the right limit— are necessarily weak.

If the entire universe is scanned without establishing valid limits (i.e. without identifying two disjoint segments to be exchanged) the exchange operation is aborted—but this should be extremely rare with the typical parameter values.

Conversely, in the normal case, valid left and right segments *are* identified, and these are then swapped (exchanged) with each other, preserving the left to right ordering (and bond states) within the segments. Note that the left and right segments contain the same number of weak bonds but are not, in general, of equal length.

Informally, we may think of an exchange operation as being implemented by "cutting" the right segment out of α_0, and then "splicing" it back into α_0

[9]More concisely: a geometric random variable, with parameter m_2, is sampled; the right and left limits are then established by counting that number of weak bonds to the right and left of the pivot respectively.

immediately to the left of the left segment. These cutting, exchanging, and splicing operations must be pictured as taking place in some space of higher dimensionality, in which α_0 is embedded.

Formally, of course, an exchange operation is a string transformation of the form:

$$z_a z_b z_c z_d z_e \mapsto z_a z_d z_e z_b z_c$$

where:

$$\begin{aligned} z_a, z_c, z_e &\in Z \\ \varphi(z_a), \varphi(z_c), \varphi(z_e) &= \mathtt{w} \\ z_b, z_d &\in Z^* \end{aligned}$$

z_a, z_c, z_e denote, respectively, the left limit, the pivot, and the right limit; $z_b z_c$ is then the left segment, and $z_d z_e$ the right segment, of the exchange operation.

m_1 and m_2 are parameters of α_0. m_1 is roughly analogous to "mean velocity" or "temperature" in chemical systems; the typical value used is 10^{-2}, which is to say that any given atom will serve as a pivot for `EX` about once in every 100 time steps on which it has a weak bond. m_2 is roughly analogous to "mean free path". Holland does not specify a typical or unique value for m_2 as he suggests that his results are relatively insensitive to it. I shall discuss this in more detail in section 5.5 below.

Note that the `EX` operator does *not* emulate anything even approximating to Newtonian mechanics. Force, velocity, momentum or kinetic energy are not even meaningful concepts in α_0. In particular, there is no notion of conservation of (kinetic) energy.

Since exchange operations with distinct pivots might interfere with each other (if their limits overlapped) Holland stipulated that the various exchanges should occur sequentially from the "leftmost" pivot to the "rightmost".

This stipulation implicitly requires that α_0 be finite: otherwise a single time step of this (strictly sequential) `EX` operator could never be completed; but this is not a substantive issue since any practical realisation would have to be finite anyway. More significantly, this stipulation also implicitly requires that α_0 be *bounded* (and not, for example, circular as assumed up to now); otherwise "leftmost" and "rightmost" atoms would not be well defined. However, Holland gives

no detailed discussion of the exact behaviour of α_0 at these implied boundaries, neither in the specific context of `EX`, nor elsewhere.

Furthermore, this mechanism for the ordering of the exchange operations still requires one further clarification. As stated, it is not clear whether, for the purposes of `EX`, α_0 should be scanned left to right just once (deciding, at each atom with a weak bond, whether to carry out an exchange, and, if so, then immediately carrying it out) or twice (first to decide which atoms should serve as pivots, and then a second time to actually carry out the exchanges). The former approach seems more straightforward, but suffers from a subtle defect: with such an approach, on any single time step, some atoms may not be even *considered* as candidate pivots for an exchange operation, and some others may actually be considered several times. This would arise whenever the outer limits are more than one weak bond away from the pivot: for then some atom(s) with weak bonds, which have not yet been considered as candidate pivots, will be shifted to the left of the current pivot, and will therefore be passed over; and conversely, some atoms with weak bonds to the left of the pivot, which have (presumably) already been considered as possible pivots, will be shifted to the right of the current pivot and will be considered again. Even more convoluted scenarios can, of course, be imagined. This defect is avoided with the alternative approach of scanning the complete space twice, for then the pivots are *all* identified before *any* exchanges are carried out. While Holland did not discuss this issue in such detail, it seems that this must have been his intended ordering mechanism.

In any case, we shall subsequently see that, in the implementation of α_0 to be described, there are good reasons for ultimately adopting a somewhat different (and simpler) approach to the `EX` operator. This will, of course, retain the required statistical characteristics, and will still involve implementing potentially conflicting exchange operations sequentially; but it will improve the execution speed, and, as an added bonus, it will transpire to be immune to the kind of ordering difficulties just described, so that the issue of scanning from a leftmost boundary to a rightmost boundary (however many times) will not arise. This will sidestep the requirement for special "boundary" behaviour completely, and allow a symmetrical (unbounded) treatment of α_0 as previously assumed. Similar comments apply to certain aspects of the behaviour of the emergent operators.

5.2.4 The Emergent Operators

The "activation" conditions for BM and EX are such that they are bound to operate in arbitrary states of α_0: BM is guaranteed to establish a population of atoms with weak bond states, and this, in turn, guarantees that the conditions for EX to operate will be satisfied. In particular, this means that these operators will be effective even in the most "disordered" or "primitive" states of α_0—and it is for this reason that they are termed *primitive.*

In contrast to this, the remaining operators are contingent on more complicated activation conditions; it seems therefore that they will have a substantive effect on the α_0 dynamics only if such (comparatively) special states should occur.[10] It is for this reason that these other operators are termed *emergent,* in the sense that they (or their effects) will not be manifest in arbitrary states of α_0, but may become manifest (emerge) if pre-existing operators (the primitive operators, in the first instance) should happen to cause the relevant special states to arise.

As already mentioned, the effects of emergent operators are roughly analogous to the functions of catalysts and enzymes in real biochemical systems. Formally, we shall identify or label the emergent operators with certain relatively invariant aspects of their activation conditions—specifically, with certain segments, or classes of segments, which remain essentially unaltered through a cycle of transformations associated with a particular operator. We shall then say that these distinguished segments *are* the emergent operators, or E-OPs.

While, in principle, the set of distinct E-OPs is infinite, they are all more or less similar in operation. This will allow the specification of their behaviour to be streamlined. In particular, the E-OPs will all be classified into just two major groups, each characterised by a certain "typical" cycle of transformations, or "reaction cycle". Loosely speaking, an E-OP can be considered as a finite state automaton, embedded in α_0, which will automatically transit through a certain typical cycle of "states".

However, it is important to emphasise, even at this point, that these "typical" cycles are simply a *device,* adopted to allow a more concise and systematic

[10]The task of quantifying just how special, or otherwise, these conditions are is a major element in any attempted analysis of α_0.

description of the E-OPs: it should not be taken to imply that, *in fact*, E-OPs can only arise or emerge in some "initial" state, or that an E-OP "reaction cycle" will necessarily run to completion. In particular, an E-OP may be spontaneously created, modified, or destroyed in any arbitrary state—i.e. at any arbitrary point in its "cycle"—by the effects of the EX *primitive* operator. This fact is referred to only obliquely by Holland; but it will transpire that it is critical to the overall behaviour of α_0.

5.2.4.1 The *Codon String* Function $\pi()$

Before attempting to characterise the E-OPs proper it is necessary to define another, different, kind of object, termed a *codon structure*.

Informally, a codon structure is the analog in α-Universe of a polynucleotide in molecular biology. It is loosely defined as any structure (null delimited segment) whose leftmost material atom (at least) is an N-atom. More formally, we define the set of codon structures as a language, $L_{\mathtt{CS}} \subset Z^*$, as follows:

$$\begin{aligned} L_{\mathtt{CS}} \equiv \{ \quad & z_{\mathtt{CS}} \in Z^*, \\ & z_{\mathtt{CS}} = z_{-} z_a z_b z_{-}, \\ & \chi(z_a) \in N^+, \\ & \chi(z_b) \in (M^* - NM^*) \quad \} \end{aligned}$$

(Note that the notation N^+ is a short form for NN^*, which is to say the set of all strings over N having at least one symbol.)

In words: a codon structure is any structure whose material segment has an N-segment prefix (z_a above). This prefix will, necessarily, be uniquely delimited at the right—either by the null atom (z_{-}) terminating the structure or by an A-atom (i.e. some M-atom which is not an N-atom). In the latter case, this A-atom, and any M-atoms following it, will be referred to as a "garbage suffix" to the codon structure (z_b above).

It is convenient to define the function:

$$\begin{aligned} \pi : L_{\mathtt{CS}} \quad &\rightarrow \quad N^+ \\ z_{\mathtt{CS}} = z_{-} z_a z_b z_{-} \quad &\mapsto \quad \chi(z_a) \end{aligned}$$

where the decomposition $z_{\mathtt{CS}} = z_{-} z_a z_b z_{-}$ is as introduced in the definition of $L_{\mathtt{CS}}$

above. That is, $\pi(z_{\texttt{CS}})$ yields the N-string corresponding to the N-segment prefix of the codon structure. Such an N-string will be termed a *codon string*.

Examples of codon structures (members of $L_{\texttt{CS}}$), and their corresponding codon strings, might be the following:

$z_{\texttt{CS}}$	$\chi(z_{\texttt{CS}})$	$\pi(z_{\texttt{CS}})$
$z_{-}(N_1,\texttt{s})(N_0,\texttt{w})z_{-}$	$-N_1N_0-$	N_1N_0
$z_{-}(N_1,\texttt{w})(\texttt{0},\texttt{s})(N_1,\texttt{s})(N_1,\texttt{w})z_{-}$	$N_1\texttt{0}N_1N_1$	N_1
$z_{-}(N_0,\texttt{w})(\texttt{:},\texttt{w})(\texttt{0},\texttt{w})(\texttt{0},\texttt{w})(\texttt{:},\texttt{w})z_{-}$	$N_0\texttt{:00:}$	N_0
$z_{-}(N_0,\texttt{s})(N_1,\texttt{s})(N_0,\texttt{s})(\texttt{1},\texttt{s})(N_1,\texttt{w})z_{-}$	$N_0N_1N_0\texttt{1}N_1$	$N_0N_1N_0$

Codon structures and E-OPs are mutually exclusive—i.e. no codon structure is an E-OP and vice versa. Codon structures are thus more or less "static"—in the sense that the only dynamic behaviour they exhibit *in themselves* is that implied by the primitive operators. We shall see that codon structures (or, at least, the codon strings thereof) play the rôle of "data" objects operated upon by the "programs" represented by the E-OPs. As mentioned earlier, they are somewhat analogous to the "A-tapes" of the previous chapter.

5.2.4.2 The *Binding* Function $\alpha()$

In general, if an E-OP is to operate on some codon structure, it must first identify a "suitable" structure, and then attach this structure (or, at least, the material segment of it) onto itself. I shall refer to this process as *selective binding*. Binding is *selective* in that there will be an embedded B-segment within the E-OP (termed the *argument* of the E-OP) which will constrain the binding—the selected codon structure must "match" this B-segment. This matching is insensitive to bond states either in the E-OP or the codon structure—it is based purely on the X-symbols in both cases. The matching condition is expressed via a mapping from N-strings (specifically, codon strings) to B-strings, called the *binding function*,[11] and denoted by:

$$\alpha : N^* \rightarrow B^*$$

[11]Holland actually describes this function as yielding an "anticode" for a given N-string; I find this confusing, as it seems to imply some relationship with the "coding" function $\gamma()$ (to be described in the next section), whereas there is no such relationship. $\alpha()$ and $\gamma()$ are quite independent both in definition and application. In particular, it is *not* the case that $\alpha = \gamma^{-1}$. Thus I prefer not to propagate the term "anticode" further.

$\alpha()$ is defined as follows: reading the N-string from left to right, each N-symbol is mapped onto a single B-symbol, according to:

$$\begin{aligned} N_0 &\mapsto \texttt{0} \\ N_1 &\mapsto \texttt{1} \end{aligned}$$

Thus, we can say, for example:

$$\begin{aligned} \alpha(N_0N_1N_0N_1) &= \texttt{0101} \\ \alpha(N_0N_1N_1N_0N_0) &= \texttt{01100} \end{aligned}$$

It is, of course, the relationship implied by $\alpha()$ which originally motivated the particular choice of symbols for the N-alphabet. A codon structure $z_{\texttt{CS}} \in L_{\texttt{CS}}$ will then be said to match a relevant B-segment u iff $\alpha(\pi(z_{\texttt{CS}}))$ contains $\chi(u)$ as a prefix. This condition will be denoted $z_{\texttt{CS}} \bowtie u$. More concisely, we require that:

$$z_{\texttt{CS}} \bowtie u \iff \alpha(\pi(z_{\texttt{CS}})) = x_a x_b,\ x_a = \chi(u) \in B^+,\ x_b \in B^*$$

I may note in passing that, since $\alpha()$ is bijective, its inverse, $\alpha^{-1}()$, is well defined; this proves convenient when it comes to practical realisation of selective binding, as it is somewhat easier to implement the matching condition $z_{\texttt{CS}} \bowtie u$ by expressing it, in terms of the segments defined above, as the condition that $\pi(z_{\texttt{CS}})$ must contain $\alpha^{-1}(\chi(u))$ as a prefix.

5.2.4.3 The *Decoding* Function $\gamma^{-1}()$

The definition of certain `E-OP`s (the *decode* type) will involve "constructing" an A-segment based on interpreting a given codon string as a "description" of it. The general idea is that codon strings may act as (more or less) quiescent descriptions of certain A-segments; these A-segments themselves are, in general, *not* quiescent (i.e. they may be, or become, `E-OP`s). The mapping from an A-segment to its description (codon string) is referred to in (Holland 1976) as the "coding" function, and is denoted by $\gamma()$. This function specifies a coding *only* for the X-symbols of the atoms in the A-segment—it is insensitive to the bond states, and is thus of the form:

$$\gamma : A^* \to N^*$$

i.e. a mapping from an A-string to an N-string.

The definition[12] of $\gamma()$ is that, reading the A-string from left to right, each A-symbol maps onto a pair of N-symbols according to:

$$\begin{aligned} \texttt{0} &\mapsto N_0N_0 \\ \texttt{1} &\mapsto N_0N_1 \\ \texttt{:} &\mapsto N_1N_0 \end{aligned}$$

Thus, we can say, for example:

$$\begin{aligned} \gamma(\texttt{010:}) &= N_0N_0N_0N_1N_0N_0N_1N_0 \\ \gamma(\texttt{0:100}) &= N_0N_0N_1N_0N_0N_1N_0N_0N_0N_0 \\ \gamma(\texttt{::::}) &= N_1N_0N_1N_0N_1N_0N_1N_0 \end{aligned}$$

This relationship between A-strings and N-strings—i.e. this particular coding of A-strings into N-strings—"exists" *only* in the sense that certain `E-OP` dynamics reflect it; nonetheless, it is convenient to define it here, separately from the detailed description of the `E-OP`s.

Of course, the function that must be used in constructing an A-segment from its description is not $\gamma()$, but its inverse—i.e. the *decoding* function, $\gamma^{-1}()$. Unfortunately, $\gamma()$, as so far defined (following Holland), is not bijective—there are many N-strings which do not code for any A-string. This arises, for example, if the N-string has an odd length, or if, when it is split into pairs from left to right, the pair N_1N_1 occurs. Thus, the "correct" definition of $\gamma^{-1}()$ is somewhat arbitrary. Holland explicitly specifies that $\gamma^{-1}()$ should map odd length N-strings by ignoring the final N-symbol in the string (thus effectively making it of even length). However, Holland does *not* specify how $\gamma^{-1}()$ should deal with the pair N_1N_1. For the work described here N_1N_1 was (arbitrarily) mapped onto ":"—thus, both N_1N_0 and N_1N_1 effectively code for the same A-symbol. The definition of $\gamma^{-1}()$ is then fully characterised by the following mappings:

$$\begin{aligned} N_0N_0 &\mapsto \texttt{0} \\ N_0N_1 &\mapsto \texttt{1} \\ N_1N_0 &\mapsto \texttt{:} \\ N_1N_1 &\mapsto \texttt{:} \end{aligned}$$

[12]This is not literally the definition given by Holland: the latter evidently incorporated a typographical error, as it showed the *same* coding for both `0` and `1`.

5.2.4.4 Searching for Raw Materials

At various points it will be necessary for an E-OP to search for "raw materials", so that it can effect particular transformations while still respecting the matter conservation "laws" of α_0. This will involve searching for a segment of a particular, specified, form. A standard searching procedure is used, which is somewhat similar to that described for establishing the limits of an exchange operation in section 5.2.3.2 above.

The search proceeds outward from the E-OP (concurrently to both the left and right); when a suitable segment is located (on the left *or* right), then, with probability m_2, this segment is selected and the search terminates; otherwise the search continues outward. Effectively a geometric random variable, with parameter m_2, is sampled, yielding a count—say k; the k'th nearest suitable segment is then selected (if it exists).

This procedure is completed within a single time step, regardless of how far the search has to proceed—up to, and including, searching the complete space. I may note that this ability to search arbitrarily far in a constant (α_0) time is a particularly counter-intuitive feature of α_0. Holland does suggest that consideration should ultimately be given to more "realistic" procedures (Holland 1976, Section 5), though I shall not pursue that here.

This search procedure may *fail*; that is, the search may exhaust α_0 without having selected any segment. This would certainly occur if *no* suitable segment existed in the (finite) α_0, and might occur even if one or more suitable segments exist but they are all passed over (which can happen with probability $1 - m_2$ for each such segment).

In any case it is stipulated that if a search fails, for whatever reason, the E-OP in question will have no effect on α_0 for that time step; this will typically mean that the search procedure will then be repeated, afresh, on the next time step.[13]

[13]Holland states this point *explicitly* only for one special case; but I have applied the same principle, *mutatis mutandis,* to all comparable cases.

5.2.4.5 Outline E-OP Syntax

All E-OPs supported in α_0 are structures and are, more particularly, members of the language $L_{\texttt{E-OP}}$, formally defined as follows:

$$\begin{aligned} L_{\texttt{E-OP}} \equiv \{ \quad & z_{\texttt{E-OP}} \in Z^*, \\ & z_{\texttt{E-OP}} = z\_uvz\_, \\ & u = u_a u_b u_c, \\ & \chi(u_a) \in B, \\ & \chi(u_b) = :, \\ & \chi(u_c) \in B^+, \\ & \chi(v) \in (M^* - BM^*) \quad \} \end{aligned}$$

Less formally, an E-OP is a structure (a null delimited segment), whose material segment can be (uniquely) decomposed into two distinct parts, u and v. We shall see that u remains essentially invariant through the cycle of transformations associated with the E-OP, and represents its fixed "program" part. u can be further decomposed into a B-atom u_a, called the operator *type*, a colon atom u_b which is just a separator,[14] and a B-segment u_c, called the operator *argument*. u_c is stipulated not to be empty ($e \notin B^+$) but is otherwise an arbitrary B-segment.[15]

The material segment v is called the *operand*; it is progressively modified through the cycle of transformations and may be regarded (roughly) as the "data" being processed by the E-OP, or as the record of the instantaneous "state" of the E-OP. At this point v is permitted to be an arbitrary material segment ($\chi(v) \in M^*$) *except* that its leftmost atom (if any) must not be a B-atom ($\chi(v) \notin BM^*$). In this way we are guaranteed that, whether v is empty or not, the argument u_c (and thus u itself) will be uniquely delimited on the right, being immediately followed by an atom whose X-symbol is *not* in B (namely the first atom in v, if it exists, or, if v is empty, the right null atom delimiter of the structure). This ensures that the decomposition of the E-OP as $z\_uvz\_$ is unambiguous.

[14] In fact, this separator is not strictly required in α_0, because the length of u_a is fixed; however, Holland intended this syntax to be extensible to more complicated α-Universes, where u_a would be of variable length ($u_a \in B^+$), hence the inclusion of u_b even in α_0.

[15] There are conceivable alternatives to requiring simply that the argument be non-empty: we might permit an empty argument, or we might impose a minimum length greater than one. Holland is not entirely clear on this issue: he explicitly refers to the possible use of an argument length of two, so the minimum should not be more that this—but it could still be zero, one or two without contradicting anything in Holland's original paper. I simply note the convention I have adopted—i.e. that the minimum valid argument length is one.

In practice, the definition of the E-OP dynamics is such that only a small number of really distinct kinds of transformation can be effected by a given E-OP. Thus, even though the set of possible operands v is very large, these will be classified into a small number of distinct classes for the purposes of defining the resulting transformations. This classes will, of course, be disjoint. Operands which do not fall into *any* of these classes are considered, by default, to be representatives of a "halt" class—which is to say that an E-OP, in such a state, will not cause any transformations at all.

Examples of members of $L_{\texttt{E-OP}}$ would be segments having the following images under $\chi()$:

$$-\texttt{0:1}-$$
$$-\texttt{1:0}N_0N_1\texttt{::}-$$
$$-\texttt{0:0000010}N_0\texttt{:}N_1N_1\texttt{10:::}-$$
$$-\texttt{1:100:100}-$$

E-OPs are classified into types on the basis of the X-symbol of the B-atom u_a in the definition above; since $B = \{\texttt{0}, \texttt{1}\}$, this yields just two distinct types. An E-OP with $\chi(u_a) = \texttt{0}$ will be called a *copy* type (henceforth denoted CP), and an E-OP with $\chi(u_a) = \texttt{1}$ will be called a *decode* type (henceforth DC). The behaviour of E-OPs of these two types will be defined in detail in the following sections.

5.2.4.6 The CP E-OPs

In brief, the CP reaction cycle consists of locating and incorporating a codon structure which matches the argument; copying the N-segment prefix part of this (effectively the codon string proper), by locating and incorporating free N-atoms in the appropriate order; and finally, dividing (incorporating null atoms) in such a way as to reconstitute the original codon structure and delimit the copy thereof; this also returns the E-OP to the initial state in the cycle. In principle, a given E-OP may repeat this cycle indefinitely although, of course, it will sooner or later be broken up through the action of the other operators (especially BM and EX).

Loosely then, the essential behaviour of a CP E-OP can be best understood by examining its image under $\chi()$, through a complete reaction cycle. A "typical"

sequence of this sort would be as follows:

$$
\begin{array}{ll}
t & -\texttt{0:010}- \\
t+1 & -\texttt{0:010:}- \\
t+2 & -\texttt{0:010:}N_0N_1N_0N_0- \\
t+3 & -\texttt{0:010:}N_0N_1N_0N_0\texttt{:}- \\
t+4 & -\texttt{0:010}N_0\texttt{:}N_1N_0N_0\texttt{:}N_0- \\
t+5 & -\texttt{0:010}N_0N_1\texttt{:}N_0N_0\texttt{:}N_0N_1- \\
t+6 & -\texttt{0:010}N_0N_1N_0\texttt{:}N_0\texttt{:}N_0N_1N_0- \\
t+7 & -\texttt{0:010}N_0N_1N_0N_0\texttt{::}N_0N_1N_0N_0- \\
t+8 & -\texttt{0:010}-N_0N_1N_0N_0-\texttt{:}-\texttt{:}-N_0N_1N_0N_0-
\end{array}
$$

Note that, in general, a CP E-OP will require at least $\ell + 4$ time steps to complete its reaction cycle, where ℓ is the length of the N-segment prefix in the codon structure which is copied ($\ell = 4$ in the example above). The time required may, of course, be longer than this—for example if the search for relevant raw materials should fail at any point.

This outline of the "normal" reaction cycle of a CP E-OP is, in fact, *all* that Holland specified of its behavior. As already noted, however, if one is interested in building a practical realisation of α_0, it is necessary to consider not just this kind of normal reaction cycle, but also all the other possible E-OP states which might conceivably arise. The rest of the discussion of the CP E-OPs below is therefore concerned with giving a *complete* and *formal* definition of the transformations they effect, such that reaction cycles of the kind loosely implied above will, in fact, result.

From the discussion already given, all CP E-OPs must be members of the language $L_{\texttt{CP}} \subset L_{\texttt{E-OP}}$ defined by:

$$
\begin{array}{lll}
L_{\texttt{CP}} \equiv \{ & z_{\texttt{CP}} \in Z^*, \\
& z_{\texttt{CP}} = z_- uvz_-, \\
& u = u_a u_b u_c, \\
& \chi(u_a) = \texttt{0}, \\
& \chi(u_b) = \texttt{:}, \\
& \chi(u_c) \in B^+, \\
& \chi(v) \in (M^* - BM^*) \quad \}
\end{array}
$$

The particular transformations implemented by a `CP E-OP` are determined by the instantaneous state—i.e. by the operand segment v. The transformations associated with any given operand are completed in one time step. This will result in a new operand; this will "typically" either be a member of the next operand class in the reaction cycle, or will still be a member of the same operand class. However, I emphasise again that an `E-OP` can, in principle, be "initially" formed with an operand of arbitrary class, and the relevant transformations will, of course, then proceed from there; and equally, the operand may fail to be transformed from one class to to the next in the reaction cycle for various reasons, despite the cycle being described as "typical" or "normal".

5.2.4.6.1 `CP` Operand Class 0

There is exactly one class 0 operand, namely the empty string:

$$v = e$$

Informally, the class 0 transformations involve locating a free colon atom and splicing this into the `E-OP`, strongly bonding it to the right end of the structure (i.e. between the rightmost existing M-atom, and the right null atom delimiter). This colon atom will be subsequently used to mark the current position in an N-segment (derived from a codon structure) while it is being copied; it will be referred to below as the *position marker*.

The "location" of the free colon atom is an example of a search for "raw materials", which has already been detailed in section 5.2.4.4. As noted there, this procedure may, in general *fail*; and, should that happen, then *none* of the transformations described here will be effected. In particular, the operand itself will remain unchanged, and typically, therefore, the procedure will be attempted afresh on the next time step (etc.). These considerations will apply in every case where reference is made to locating particular kinds of segment, and will not be repeated further.

More formally, let the decomposition of the segment u of the `E-OP`, into its constituent atoms, be denoted in the normal way by:

$$u = u(1)u(2)\ldots\ u(|u|),\ u(i) \in Z$$

Note that $|u|$ denotes the length of u. $u(|u|)$ will thus denotes the rightmost atom of u. It is necessarily the case that $\varphi(u(|u|)) = \mathtt{w}$ (i.e. this atom has a weak bond state) because the next atom to the right is null ($= z_{\_}$).

Let z denote a free colon atom:

$$z = z_{\_}(:, \mathtt{w})z_{\_}$$

The following transformations are then triggered by a class 0 operand:

$$\begin{aligned} z &\mapsto z_{\_}z_{\_} \\ v &\mapsto (:, \mathtt{w}) \\ \varphi(u(|u|)) &\mapsto \mathtt{s} \end{aligned}$$

5.2.4.6.2 CP Operand Class 1

Again, there is exactly one class 1 operand, namely a single colon atom (necessarily with a weak bond):

$$v = (:, \mathtt{w})$$

Informally, the class 1 transformations involve locating a codon structure matching the argument of the E-OP, and splicing the material segment of this into the E-OP, strongly bonding it to the right of the colon atom v. The internal bonding of the material segment extracted from the codon structure is not altered.

More formally, let $z_{\mathtt{CS}}$ be a codon structure:

$$\begin{aligned} z_{\mathtt{CS}} = \; & z_{\_}z_a z_b z_{\_}, \\ & \chi(z_a) \in N^+, \\ & \chi(z_b) \in (M^* - NM^*) \end{aligned}$$

We require that $z_{\mathtt{CS}}$ match the E-OP argument, in the sense of the $\alpha()$ function as described in section 5.2.4.2; that is:

$$z_{\mathtt{CS}} \bowtie u_c$$

The following transformations are then triggered by a class 1 operand:

$$\begin{aligned} z_{\mathtt{CS}} &\mapsto z_{\_}z_{\_} \\ v &\mapsto (:, \mathtt{s})z_a z_b \end{aligned}$$

5.2.4.6.3 `CP` Operand Class 2

Class 2 operands are defined by the condition:

$$\begin{aligned} v = \quad & v_a v_b v_c v_d, \\ & \chi(v_a) \in (D^* - BD^*) \\ & \chi(v_b) = :, \\ & \chi(v_c) \in ND^*, \\ & v_d = e \end{aligned}$$

(The reason for including v_d here will become clear subsequently.)

It should be clear that this decomposition, and similar ones which follow in subsequent sections, will be unique. This point will not, therefore, be repeated.

In words, the operand is of class 2 if it has exactly one colon atom in it, and the atom immediately to the right of this is an N-atom.

Note carefully that the segment denoted v_c must not contain any colon atoms (recall that D is the set of material elements *exclusive* of the colon element). Now, there is no constraint in the definition of codon structures which guarantees that their material segments (which is what v_c is typically derived from) will not contain colon atom(s) in the garbage suffix. Thus, the "typical" transition from a class 1 operand to a class 2 operand might be subverted. It would, arguably, be preferable to define the class 1 transformations to reduce or eliminate this possibility, by stipulating that *only* the N-segment prefix of a codon structure should be incorporated into the `E-OP`, rather than the complete material segment (i.e. the garbage suffix, if any, would be discarded by the action of the class 1 operand). This would also, incidentally, slightly simplify the definitions of later operand classes and their transformations. However, while Holland is not completely unambiguous on this point, there is a strong implication in his treatment that the garbage suffix should not be discarded in this way, so I leave the class 1 transformations as they have already been stated.

Informally, the class 2 transformations are similar to those for the class 0 operand: a free colon atom is located and spliced into the `E-OP`, strongly bonding it to the right of the rightmost existing M-atom. This colon atom will be subsequently used to separate the original N-segment from the copy.

More formally, let z denote a free colon atom:

$$z = z_{\_}(:, \mathtt{w})z_{\_}$$

The following transformations are then triggered by a class 2 operand:

$$\begin{aligned} z &\mapsto z_{\_}z_{\_} \\ v_d &\mapsto (:, \mathtt{w}) \\ \varphi(v_c(|v_c|)) &\mapsto \mathtt{s} \end{aligned}$$

(The reason for including $v_d = e$ in the prior decomposition of v now becomes apparent: it allows the second of these transformations to be expressed relatively concisely.)

5.2.4.6.4 `CP` Operand Class 3

Class 3 operands are defined by the condition:

$$\begin{aligned} v = \; & v_a v_b v_c v_d v_e v_f v_g, \\ & \chi(v_a) \in (D^* - BD^*), \\ & \chi(v_b) = :, \\ & \chi(v_c) \in N, \\ & \chi(v_d) \in D^*, \\ & \chi(v_e) = :, \\ & \chi(v_f) \in M^*, \\ & \chi(v_g) = e \end{aligned}$$

In words, the operand is of class 3 if it contains at least two colon atoms, and the atom immediately to the right of the first colon atom (the position marker) is an N-atom.

Informally, the class 3 transformations involve "copying" a single N-atom; that is, a free N-atom is located, matching the N-atom immediately to the right of the position marker, and this is spliced into the `E-OP`, strongly bonding it to the right of the rightmost existing M-atom. The position marker is also exchanged with the N-atom which has just been copied. In general, this will produce an operand which is still of class 3 unless and until the position marker is moved to a point where the next atom is not an N-atom (typically, it will either be the

colon atom added by the class 2 operand, or the first atom of the garbage suffix of the original codon structure). In this way, the `E-OP` should continue to have a class 3 operand, and will continue copying one N-atom per time step, until the embedded codon string is completely copied.

More formally, let z denote the required free N-atom:

$$z = z_-(\chi(v_c), \mathtt{w})z_-$$

The following transformations are then triggered by a class 3 operand:

$$\begin{aligned} z &\mapsto z_-z_- \\ v_g &\mapsto (\chi(v_c), \mathtt{w}) \\ \varphi(v_f(|v_f|)) &\mapsto \mathtt{s} \\ v_c &\mapsto v_b \\ v_b &\mapsto v_c \end{aligned}$$

5.2.4.6.5 `CP` Operand Class 4

Class 4 operands are defined by the condition:

$$\begin{aligned} v = \; & v_a v_b v_c v_d v_e, \\ & \chi(v_a) \in (D^* - BD^*), \\ & \chi(v_b) = :, \\ & \chi(v_c) \in (D^* - ND^*), \\ & \chi(v_d) = :, \\ & \chi(v_e) \in M^* \end{aligned}$$

In words, the operand is of class 4 if it contains at least two colon atoms, and the atom immediately to the right of the first colon atom (the position marker) is *not* an N-atom.

Informally, the class 4 transformations involve breaking up and ejecting the operand, typically establishing two distinct, but identical (in the sense of $\pi()$), codon structures (and possibly an additional garbage structure), separated from the `E-OP`. One of the codon structures is effectively the "original", and the other is the newly constructed "copy". By ejecting the existing operand, the transformed operand (namely the empty string e) will be of class 0 again, and the reaction cycle has been closed.

These transformations are achieved by locating "free" null atoms and splicing them into the (former) `E-OP` in appropriate locations. Bond states will be forced to weak wherever necessary. The precise details vary depending on whether certain substrings which, by the definition above, are allowed to be empty, are, in fact, empty (since clearly, in such a case, there is no point in "delimiting" such empty strings by null atoms).

The transformations associated with the class 4 operands are the most complicated of all, and involve at least two, and perhaps as many as five, distinct sets of related transformations. These sets must be implemented sequentially, and each separately involves a "location" procedure, which may potentially fail. It is stipulated however, that *all* required location procedures must be successfully completed before *any* transformations are carried out; and if any location procedure actually fails then *no* transformations will be carried out.

It is convenient to regard each distinct set of transformations as a case of a generic "null-insertion" set of transformations which inserts a single null atom immediately to the right of a specified segment, say z. Let $z = z_a z_b$ where $z_b = e$. A segment $z_c = z_- z_-$ is first located. The null-insertion transformations, applied to z, are then defined as follows:

$$\begin{aligned} z_c &\mapsto z_- \\ z_b &\mapsto z_- \\ \varphi(z_a(|z_a|)) &\mapsto \texttt{w} \end{aligned}$$

The complete set of transformations triggered by a class 4 operand then consists of the sequential application of the null-insertion transformations to the segments u, v_a (iff $v_a \neq e$), v_b, v_c (iff $v_c \neq e$), and v_d (iff $v_e \neq e$), in that order.

5.2.4.7 The `DC E-OP`s

The behaviours of the `DC E-OP`s are extremely similar to those of the `CP E-OP`s. The discussion here will therefore concentrate just on those aspects in which the two `E-OP` types *differ*.

The DC reaction cycle consists of locating and incorporating a codon structure which matches the argument; decoding (in the sense of $\gamma^{-1}()$) the N-segment prefix part of this (effectively the codon string proper), by locating and incorporating free A-atoms in the appropriate order; and finally, dividing (incorporating null atoms) in such a way as to reconstitute the original codon structure and delimit the decoded version thereof; this also returns the E-OP to the initial state in the cycle. In principle, a given E-OP may repeat this cycle indefinitely although, of course, it will sooner or later be broken up through the action of the other operators (especially BM and EX).

As for CP, the essential behaviour of a DC E-OP can be best understood by examining its image under $\chi()$, through a complete reaction cycle. A "typical" sequence of this sort would be as follows:

$$\begin{array}{ll}
t & -\texttt{0:010}- \\
t+1 & -\texttt{0:010:}- \\
t+2 & -\texttt{0:010:}N_0N_1N_0N_0- \\
t+3 & -\texttt{0:010:}N_0N_1N_0N_0\texttt{:}- \\
t+4 & -\texttt{0:010}N_0N_1\texttt{:}N_0N_0\texttt{:1}- \\
t+5 & -\texttt{0:010}N_0N_1N_0N_0\texttt{::10}- \\
t+6 & -\texttt{0:010}-N_0N_1N_0N_0-\texttt{:}-\texttt{:}-\texttt{10}-
\end{array}$$

Thus, in general, a DC E-OP will require at least $\ell/2+4$ time steps to complete its reaction cycle, where ℓ is the length of the N-segment prefix in the codon structure which is decoded ($\ell = 4$ in the example above). Again, the time actually required may be longer than this.

All DC E-OPs must be members of the language $L_{\texttt{DC}} \subset L_{\texttt{E-OP}}$ defined by:

$$\begin{array}{ll}
L_{\texttt{DC}} \equiv \{ & z_{\texttt{DC}} \in Z^*, \\
& z_{\texttt{DC}} = z_- uvz_-, \\
& u = u_a u_b u_c, \\
& \chi(u_a) = \texttt{1}, \\
& \chi(u_b) = \texttt{:}, \\
& \chi(u_c) \in B^+, \\
& \chi(v) \in (M^* - BM^*) \quad \}
\end{array}$$

As with CP, the operands of the DC E-OP are grouped into classes which effect essentially similar transformations. The operand classes 0, 1 and 2, and the resulting transformations, are identical for both CP and DC and will not be repeated. The definitions of the remaining 2 classes, and their transformations, differ slightly, and will be discussed individually.

5.2.4.7.1 DC Operand Class 3

The definition of this class is identical to CP class 3, *except* that the codon string part must consist of at least two atoms, where one sufficed in the CP case. In terms of the decomposition defined for CP, we require that:

$$\chi(v_c) \in N^2$$

The resulting transformations are somewhat similar to those of CP, but with several significant differences. A free A-atom (rather than N-atom) is located and spliced into the E-OP. The X-symbol of this A-atom must equal the decoded version of the next pair of N-atoms in the codon string part of the E-OP (in the sense of the $\gamma^{-1}()$ function). Finally, the position marker must be moved *two* positions to the right rather than just one. As with CP, the new operand will again typically be of (DC) class 3, and transformations of this sort will be iterated until the codon string part of the E-OP is exhausted (has a length less than two in this case).

More formally, let z denote the required free A-atom:

$$z = z_-(\gamma^{-1}(\chi(v_c)), \mathtt{w})z_-$$

The following transformations are then triggered by a DC class 3 operand:

$$\begin{aligned} z &\mapsto z_-z_- \\ v_g &\mapsto (\gamma^{-1}(\chi(v_c)), \mathtt{w}) \\ \varphi(v_f(|v_f|)) &\mapsto \mathtt{s} \\ v_c &\mapsto v_b \\ v_b &\mapsto v_c \end{aligned}$$

5.2.4.7.2 DC Operand Class 4

The definition of this class is identical to CP class 4, except that the garbage remaining from the codon string is now allowed to include a single initial N-atom. In terms of the decomposition defined for CP class 4, we require that:

$$\chi(v_c) \in (D^* - N^2 D^*)$$

With this modification of the definition of v_c, the resulting transformations are then identical to those defined for CP Class 4 operands.

5.3 "Life" in α_0?

Consider a complex in α_0 comprising 8 structures having the following images under $\chi()$:

$-$0:001$-$	$-N_0N_0N_1N_0N_0N_0N_0N_0N_0N_1-$
$-$0:011$-$	$-N_0N_0N_1N_0N_0N_0N_0N_1N_0N_1-$
$-$1:001$-$	$-N_0N_1N_1N_0N_0N_0N_0N_0N_0N_1-$
$-$1:011$-$	$-N_0N_1N_1N_0N_0N_0N_0N_1N_0N_1-$

For reasons which will become more clear subsequently, this complex will be referred to as FullSR.

It will be observed that the structures represented in the left column are all E-OPs, the first two of type CP, the second two of type DC, while the structures represented in the right column are all codon structures. The first CP can bind with the first two codon structures and is thus capable of copying them; the second CP can bind to, and thus copy, the remaining two codon structures. Similarly, the first DC can bind to, and decode, the first two codon structures, and the second DC can bind to, and decode, the remaining two codon structures. Finally, it can be easily verified that, when the four codon structures are decoded they actually yield precisely the four E-OPs represented in the left column.

Prima facie, then, the complex FullSR is capable of a form of "self-reproduction": it is an example, in α_0, of a roughly von Neumann style, genetically based, *A-reproducer*.

Admittedly, complexes in α_0 lack some of the coherence or unity of the A-machines considered in the previous chapter. That is, given a population of

structures there is a certain arbitrariness in identifying which of these constitute distinct complexes. When we say that `FullSR` is self-reproducing, what we mean is that, given a single instance of it, this should result in the generation of a large population of structures which contains many instances of each of the component structures of `FullSR`; but it is then quite arbitrary which of these structures should be grouped together as "instances" of `FullSR` itself. There is actually a rather fundamental issue at stake here: to anticipate somewhat, we can roughly regard the recursive interaction of the structures making up `FullSR` as realising a form of genetic self-reproduction *or* as realising a quasi-autopoietic organisation—*but not as doing both*. I shall eventually return to examine this question again in section 5.5.8 below; for the present I shall continue to take the former view, which regards `FullSR` simply as realising a limited form of genetic self-reproduction. The crucial feature of `FullSR`, for my purposes (i.e. in terms of P_a), is that it seems to be *robustly* self-reproducing, and, in this respect, it might represent a substantive advance over the various A-reproducers previously considered in Chapter 4.

As it happens, the reproduction mechanism of `FullSR` is also *genetic*—i.e. it is of the same general kind as formulated by von Neumann in the solution of P_v. In particular, we can roughly interpret the set of `E-OP`s in `FullSR` as a basic *Genetic Machine* (GM) g_0 as introduced in the previous chapter; the set of codon structures then collectively constitute a particular (dashed) A-descriptor, namely $d'(g_0)$, and self-reproduction follows. Indeed, although Holland does not explicitly mention von Neumann's work, it seems likely that Holland's particular definition of the `E-OP`s in α_0 was motivated precisely with this outcome in mind.

However, it should be emphasised that the similarities between α_0 and the (A-)systems introduced by von Neumann are very limited. While it is true that one can formulate an indefinitely large set of self-reproducing complexes, all "related" to `FullSR`, this has no particular significance in the context of α_0. Because the set of `E-OP`s defined in α_0 has been (deliberately) impoverished (to facilitate the analysis of this "proof of principle" system), the range of *A-behaviours* spanned by this set of A-reproducers seems to be extremely limited. In particular, since α_0 does not support universal computation, we cannot even expect this set of A-reproducers to meet the weak (behaviour spanning) criterion of allowing

the embedding of an arbitrarily programmed universal logical machine (ULM).

Thus, we may say that α_0 does indeed support a form of genetic self-reproduction, leading to an indefinitely large set of related A-reproducers (though even here, the details would diverge somewhat from von Neumann's concept), but this set of A-reproducers clearly does *not* span a significant range of complexity. α_0 would *not* therefore serve as a vehicle for the solution of P_v (by von Neumann's schema or otherwise).

While it is well to be clear about this divergence between α_0 and the von Neumann A-systems, it is not particularly surprising. The *problem* being addressed by Holland is (ostensibly) quite different from that tackled by von Neumann. Indeed, to the extent that von Neumann did, indeed, solve his identified problem, it is neither here nor there whether α_0 might provide another "alternative" solution to that same problem. No, the point of α_0 (for Holland at least) is not to consider the potential for evolution from `FullSR` to *more* complex A-reproducers, but rather to consider how even this initial, extremely basic, A-reproducer could itself arise from *less* complex precursors. In particular, Holland estimates that if the complex `FullSR` had to spontaneously emerge *solely* as a result of the unbiased generation due to the primitive operators, the expected emergence time would be of the order of 10^{43} time steps. Holland comments:

> This is such a large number that, for all practical purposes, we can reject the possibility of spontaneous emergence, if indeed the system [`FullSR`] must emerge in one fell swoop.
>
> Holland (1976, p. 399)

We now note that, on the face of it, `FullSR` itself is already of greater "complexity" (or, at least, bigger) than is strictly necessary. It would seem that we could shorten the arguments of the `E-OP`s of `FullSR`, while still retaining self-reproduction. Thus, shortening the arguments by one atom yields a complex of the form:

$$\begin{aligned}
-\texttt{0:00}- \quad & -N_0N_0N_1N_0N_0N_0N_0N_0- \\
-\texttt{0:01}- \quad & -N_0N_0N_1N_0N_0N_0N_0N_1- \\
-\texttt{1:00}- \quad & -N_0N_1N_1N_0N_0N_0N_0N_0- \\
-\texttt{1:01}- \quad & -N_0N_1N_1N_0N_0N_0N_0N_1-
\end{aligned}$$

Or, even more dramatically, if we shorten the arguments by a further atom (thus reducing them to the minimum length of just a single atom in each case),

it turns out that only one distinct argument (0) is required, and the complex can be reduced to just four distinct structures in total:

$$-\texttt{0:0}- \quad -N_0N_0N_1N_0N_0N_0-$$
$$-\texttt{1:0}- \quad -N_0N_1N_1N_0N_0N_0-$$

These simplified complexes share with FullSR the fact that they each define a set of possible transformations which, if they all occur, will result in the reproduction of the original complex.

But the issue here is not just which transformations are possible, but also which will actually occur. It should be clear that the simplifications of FullSR, suggested above, may be counterproductive: by shortening the arguments of the E-OPs one is making it more likely that E-OPs will bind with "random" codon strings (i.e. *not* belonging to the complex). If such "mis"-binding events are too common, then the complex will fail to achieve self-reproduction after all. In the event, Holland argues that FullSR represents a *minimal* complex in α_0 which could effectively self-reproduce.

However Holland goes on to identify complexes, significantly simpler than FullSR, which are not properly capable of self-reproduction (in the manner of FullSR) but which could nonetheless achieve a kind of "partial" (self?)-reproduction; he then argues that this phenomenon might be sufficient to strongly bias the subsequent generation of new structures (and complexes), and might ultimately provide a plausible route for the emergence of FullSR proper (for example).

Holland introduces firstly the following complex, consisting of three structures:

$$-\texttt{0:100}- \quad -N_1N_0N_0N_1N_0N_0N_0N_0-$$
$$-\texttt{1:100}-$$

The key point here is that the single codon structure in this complex, which can be copied and decoded by the two given E-OPs, does not code completely for either E-OP, but does code *partially* for *both*:

$$\gamma^{-1}(N_1N_0N_0N_1N_0N_0N_0N_0) = \texttt{:100}$$

The arguments of the two E-OPs are still long enough (arguably) to ensure that they will almost certainly bind to this codon string if it is available. So, if

this complex should arise, it should result in the generation of a high density of copies of the codon structure, *plus* a high density of E-OP fragments of the form:

$$-\texttt{:100}-$$

Now these fragments can be transformed into one or the other of the E-OPs of the original complex, simply by a B-atom being added in front of the colon atom—and it seems possible, at least, that this could happen spontaneously, with reasonable frequency, just by the background operation of the EX operator. This would effectively complete the reproduction of the original complex.[16]

Holland suggests that, indeed, this kind of process could occur and sustain itself in α_0. Indeed, he goes further and suggests that, if a large density of the relevant codon structure could be *initially* established, then this kind of process could actually be effective even if the arguments to the E-OPs were reduced from three atoms to just two. Again, this would allow the codon structure itself to be shorter also, so we identify the following complex as also "partially" self-reproducing:

$$-\texttt{0:10}- \quad -N_1N_0N_0N_1N_0N_0-$$
$$-\texttt{1:10}-$$

I shall refer to this complex as PartSR.

It is important to note carefully here the condition which allows the reduction in size of the E-OP arguments—namely that the complex (or, at least, the single codon structure within it) *already* exists in high density. More specifically, the claim is that if, by whatever means, instances of this codon structure, with strong internal bonding, should achieve high density, then a high density of the complex PartSR should spontaneously form and sustain itself indefinitely thereafter.

In more detail, the idea is that the initial large population of codon structures is expected to persist long enough that it is likely that an instance of the DC E-OP of PartSR will spontaneously form, even while the density of codon structures is still high. This will then result in the formation of a large population of the relevant E-OP fragments—i.e. fragments of the form:

$$-\texttt{:10}-$$

[16]It is perhaps debatable whether this should still be termed "self" reproduction—but the precise terminology is not important here. More generally, it does not matter for my purposes which, if any, of the complexes described are actually labelled as "living".

After this, the E-OP fragments could get spontaneously transformed into the required E-OPs belonging to PartSR, with sufficient frequency that a large population of PartSR does, indeed, form and sustain itself.

All this leaves open the question of how a high density of instances of the PartSR codon structure (with strong internal bonding) could be formed in the first place. To this end, Holland finally directs attention to the following complex with just two structures:

$$-\texttt{0:100}- \quad -N_1N_0N_0N_1N_0N_0-$$

Holland argues that if even a single instance of this complex should spontaneously form (with both structures having strong internal bonding) then this will result precisely in the formation of a high density of the PartSR codon structure, as required. Note the CP E-OP in this final complex is not the CP E-OP of the PartSR complex as such: it has an extra atom in the argument to ensure that it will preferentially bind, with "sufficient" probability, to the PartSR codon structure, even while the latter is present only at low density.

This final complex will be referred to as the Seed complex: it apparently has the property that, if a single instance should (spontaneously) form, then (with high probability) a population of PartSR complexes should arise, and subsequently sustain itself indefinitely (unless and until it is displaced by some other complex—perhaps even FullSR— which is more efficient in its reproduction).

Holland's "proof-of-principle" can then be stated as follows. A naïve view of the origin of "life" (in α_0) would assume that FullSR (or something of comparable complexity) must spontaneously form purely from the "unbiased" generation of "random" structures by the primitive operators. But, in fact, the complex Seed would spontaneously form at a much earlier stage (since it is so much simpler than FullSR). Holland specifically estimates the expected emergence time for Seed as only about 4×10^8 time steps, which would make such emergence quite feasible. Once this occurs, the subsequent generation of structures would be strongly biased, in a way which could dramatically accelerate the emergence of FullSR. Indeed, as indicated above, the process subsequent to the spontaneous formation of Seed would *already* take on at least some of the flavour of Darwinian evolution.

It may be noted that there is no claim that the complexes `Seed`, `PartSR`, or even `FullSR`, are *unique* in the rôles they play here. There may well be other complexes which, if they should spontaneously form, would strongly bias the subsequent generation of structures in a manner similar to that of `Seed` and `PartSR`, such that `FullSR`, or some other similarly complex A-reproducer, could then quite plausibly emerge. Holland's point is to give a "proof-of-principle": for this it is sufficient to exhibit *one* family of complexes (namely `Seed`, `PartSR` and `FullSR`) having the required properties. To whatever extent (if any) other alternative complexes could have an equivalent effect, Holland's argument could only be strengthened.

In this section I have been concerned solely with outlining the *conclusions* of Holland's analysis of α_0. This is necessarily qualitative—and entirely unconvincing in itself. Holland, of course, accompanies this discussion with a detailed quantitative analysis to support his conclusions. I shall not consider this analysis at this point. Rather, I shall turn to the more direct approach: simply testing whether the phenomena which have been qualitatively described here do, in fact, occur in a particular implementation of α_0.

5.4 `AV0`: A Realisation of α_0

This section describes a package called `AV0`, which is, in effect, a computer based realisation of α_0.

`AV0` has been written entirely in the C language (ANSI X3J11). Original development was carried out on an IBM PC compatible platform, running MS-DOS, and Turbo-C V2.0. However, the empirical results reported below were recorded with an alternative version, still running on an IBM PC compatible platform, but compiled under `GNU cc`, and executed in 80386 protected mode, to allow access to a large, linear, 32-bit address space; the latter was required to allow universes of size greater than about 2×10^3 atoms to be realised. As far as possible, the package has been written to be "easily" portable (machine dependencies are encapsulated by conditional compilation). The source code comprises about 5000 lines, in roughly 60 files. This source code has been placed

in the public domain, and is available to interested researchers.[17]

The primary documentation for `AV0` is the source code itself. This section is intended only to provide background information which might significantly ease the understanding of the source code.

5.4.1 The Programs

`AV0` is organised into four executable programs.

`av0run` realises the α_0 dynamics proper. It offers facilities for loading and saving disk file images of α_0 universes and for executing the α_0 dynamic operators over any specified number of time steps, including dynamic display of a window onto the state string. There is also support for the extraction and logging of various statistical measures evaluated on the state string.

The other three programs are utilities for generating state strings with particular characteristics, as follows:

- `randmat`: This yields a completely "randomised" state string.
- `partsr`: This divides all material atoms more or less equally between instances of the three structures making up the `PartSR` complex, and free atoms.
- `fullsr`: This divides all material atoms more or less equally between instances of the eight structures making up the `FullSR` complex, and free atoms.

5.4.2 The Disk Images

A disk image of a particular α_0 minimally comprises three files, grouped by having the same name (up to 8 characters). The three files are distinguished by their extensions as follows:

- `.mat`: This contains an image of the state string.
- `.prm`: This contains all parameters not implicit in the state string—namely r, λ, m_1 and m_2. It also contains a set of flags which allow each of the

[17] Requests should be directed to the author, in the first instance.

operator types (`BM`, `EX`, `CP`, `DC`) to be selectively enabled or disabled. Finally, this file contains parameters specifying the interval (in time steps) at which log records should be emitted, and (separately) the interval at which the console display should be updated.

- `.stt`: This contains three "state" variables not contained in the state string: the current "time" (in α_0, not wall clock time), the current "seed" for the pseudo-random number generator, and a "count" of the number of pseudo-random number evaluations so far carried out (the latter is maintained as a check against possible cycling of the generator).

When `av0run` is executing it maintains a log file with the extension `.log`. Each log record contains a summary of certain statistics on the state string at a particular time step. When an image is loaded by `av0run` a log file will be created if one does not already exist; otherwise the new log records will be simply appended to the existing log file.

All disk files associated with the image of a particular α_0 are simple ASCII encoded text, so that they can be examined (and modified, if necessary) using normal text editing tools.

5.4.3 The State String

The primary data structure required is the state string. A closed doubly linked list is used. The size is dynamically determined whenever a new image is loaded, but remains static otherwise. This corresponds to a finite, but unbounded (circular) organisation, where the size is set at "initialisation". Essentially, the size is determined by an argument to whichever program is used to generate a `.mat` file; thereafter it is a constant of the α_0. The linked list is effectively superimposed on a simple, static, array of atoms.

A closed organisation was chosen primarily to avoid having to introduce special code to deal with behaviour at boundaries. Note that this means that the state string lacks any "absolute" position reference.

The linked list arrangement was chosen so that the locations *in memory* of each atom would not change (the relative locations in the state string do, of course, change). This makes the implementation of "movement" (which arises

both from the EX operator, and the E-OPs) reasonably efficient: a segment can be arbitrarily relocated just by rearranging pointers, rather than actually moving atomic symbols around in memory.

The underlying, static, array organisation makes it possible to efficiently maintain indices of specified kinds of atoms, which, in turn, can significantly speed up the execution of certain operators. However, many aspects of the α_0 dynamics still require segments to be scanned, generally in either direction. The double linking of the list makes this reasonably efficient.

5.4.4 Pseudo-random Number Generator

The AVO package involves a number of "stochastic" processes. A pseudo-random number generator is used to support these. A variety of pseudo-random number generators are available—there is generally one included with the standard C library. However, as reported by Park & Miller (1988), the quality of these generators is highly variable—where "quality" reflects some statistical measure(s) on the generated numbers. Furthermore, these statistical properties are not generally documented for the generators supplied as standard in a C library.

It was therefore decided to implement the "minimal standard generator", identified in (Park & Miller 1988), whose characteristics would, at least, be known, and would also meet certain minimal quality criteria.

5.4.5 Primitive Operators

Prima facie, the BM and EX operators both involve sampling a "large" number of independent Bernoulli random variables at each time step. This would be computationally expensive, and an alternative, statistically equivalent, algorithm was developed.

The key point is that, in each case, the probability of success for each Bernoulli random variable is quite small (typically 10^{-4} in the case of BM, and 10^{-2} in the case of EX). That is, "most" of the Bernoulli trials would usually come up with failure. The approach adopted was to first decide *how many* successes there should be on each time step, and then select *which* members of the population (of trials) will actually be the successes.

More formally, let $\mathbf{X}$ be an n-dimensional random variable, consisting of n identical, independent, Bernoulli random variables.[18] Interpreting a component value of 1 as indicating that the corresponding object should be operated on, and 0 as indicating that it should not, then the selection processes associated with the `BM` and `EX` operators are equivalent to making trials of a suitable $\mathbf{X}$.

Let $\mathbf{x}$ denote a trial of $\mathbf{X}$. Let M denote the number of 1's in $\mathbf{X}$. M, being a function of the random variable $\mathbf{X}$, is, formally, another random variable, which is jointly distributed with $\mathbf{X}$. Let m denote the number of 1's in $\mathbf{x}$—i.e. m denotes the result of the trial of M (note that, by the definitions of $\mathbf{X}$ and M, M is binomially distributed, with parameters n and p; this will prove significant later).

Let $q = 1 - p$. Given that the components of $\mathbf{X}$ are identical, independent, Bernoulli random variables with parameter p, the (marginal) probability function for any component is simply:

$$\begin{aligned} P(X_i = 1) &= p \\ P(X_i = 0) &= q \end{aligned}$$

Given that the components of $\mathbf{X}$ are independent, the probability function of $\mathbf{X}$ (i.e. the joint probability function of the components) is given simply by the product of the marginals. The event $\mathbf{x}$ denotes the situation that, of n independent Bernoulli random variables, m resulted in the value 1 (with probability p in each case) and $n-m$ resulted in the value 0 (with probability q in each case). The probability of this event is the product of the separate probabilities. Concisely, the probability function for $\mathbf{X}$ is, therefore:

$$p_{\mathbf{X}}(\mathbf{x}) = p^m q^{n-m}$$

This, then, is the objective: we wish to formulate an alternative procedure for evaluating $\mathbf{X}$, such that the probability function still matches this expression but which will be computationally more efficient (at least in the cases of interest in the α-Universe) than evaluating n independent Bernoulli random variables.

Now consider a new random variable, $\mathbf{X}'$. As with $\mathbf{X}$, $\mathbf{X}'$ is an n-dimensional random variable, where the sample space for each component is $\{0, 1\}$. $\mathbf{X}'$ is evaluated as follows. Let M' be a binomial random variable with parameters n

[18] The notation introduced here follows that of Larson (1974).

and p. Make a trial of M'; let m be the result. Randomly select m distinct numbers from the set $\{0..(n-1)\}$ (with all possible such selections being equally likely). For all i in this set assign the value 1 to the corresponding components, X'_i, of $\mathbf{X}'$; assign the value 0 to all other components of $\mathbf{X}'$.

Note carefully the distinction between M and M'. Though they both have the same probability function, M is defined (and therefore evaluated) *indirectly*—as a function of $\mathbf{X}$; whereas M' is directly defined (and evaluated) in its own right. The precise mechanism for directly evaluating M' will be discussed subsequently; for the moment, the important point is that a binomial random variable *can* be directly generated—we do not *have* to resort to the indirect method of generating n independent trials of a Bernoulli random variable.

I claim that, with this new procedure, $\mathbf{X}'$ will have precisely the same probability function as $\mathbf{X}$ (and may therefore be evaluated in place of it), but is computationally much more efficient (the quantitative improvement in efficiency will depend on p: the smaller p is, the greater the improvement). Informally, the idea is this: with the original procedure, we *always* had to evaluate n independent random variables—even though, "on average", only np of them resulted in a "success". With the new procedure, the number of random variable evaluations is, on average, $1 + np$ (M' is always evaluated, and then, on average, a further np evaluations are necessary to randomly select the "lucky" components).

We now prove that this new procedure does indeed produce the same probability function as the original procedure.

M' is defined to be binomial, with parameters n and p; its probability function is therefore:

$$p_{M'}(m) = \left(\begin{array}{c} n \\ m \end{array} \right) p^m q^{n-m}$$

Now consider the event $\mathbf{X}' = \mathbf{x}$, where $\mathbf{x}$ contains exactly m 1's. This can occur only if, firstly, the result of evaluating M' is m (probability $p_{M'}(m)$ per the expression above), *and* the "correct" m elements of the set $\{0..(n-1)\}$ are selected. For the latter event, there are $\left(\begin{array}{c} n \\ m \end{array} \right)$ possible distinct outcomes, all equally likely—so the probability of the particular outcome specified is just the reciprocal of this. Since the two experiments (sampling M', and selecting the m

components) are defined to be independent, we can multiply the probabilities of the two events to get the probability of $\mathbf{x}$:

$$\begin{aligned} p_{\mathbf{X}'}(\mathbf{x}) &= p_{M'}(m) \cdot \frac{1}{\binom{n}{m}} \\ &= \frac{\binom{n}{m} p^m q^{n-m}}{\binom{n}{m}} \\ &= p^m q^{n-m} \\ &= p_{\mathbf{X}}(\mathbf{x}) \quad \text{QED} \end{aligned}$$

Assuming that the computational cost of evaluating a single (scalar) random variable is (approximately) independent of its probability function (and this assumption may have to be justified), then, clearly, it is computationally cheaper to evaluate only $1 + np$ random variables, instead of n. To quantify this, the new procedure will be, on average, computationally cheaper by a factor $n/(1 + np)$. In the case of `AVO`, we typically have $p < 10^{-2}$, and the computational advantage is substantial—say $> 10^2$.

In implementing this procedure in `AVO`, there are two distinct steps: evaluating M' (yielding m, the *number* of components to be selected) and then actually choosing the particular m components.

To sample a random variable of given probability function, a (pseudo-)random number is generated (with a uniform probability function over a given range) and this is then passed through an inverse, cumulative, version of the desired probability function. This distorts the uniform probability function of the original (pseudo-)random variable into just the shape of the desired probability function. Thus, the assumption that the cost of evaluating a random variable is independent of its probability function reduces to an assumption that the cost of evaluating the inverse, cumulative, probability function is negligible (at least compared to the cost of generating the pseudo-random number itself).

In the particular case of interest (evaluating M'—binomially distributed with parameters n and p) the computation of the inverse, cumulative function is quite

demanding, but this is overcome by using look up tables which are computed once-off per run of `av0run`, and imposes no on-going computational costs per time step. There is one lingering difficulty which is that the n values relevant to the various α_0 operators vary dynamically—they correspond essentially to the number of atoms having bonds of an appropriate sort (only atoms with weak bonds qualify as potential pivots for `EX` etc.). In practice this is overcome by fixing $n = R$ in *all* cases; then, m atoms are "provisionally" selected, but the operator is applied only to those for which it is allowed. This means that somewhat more atoms are "provisionally" considered for the application of the operator than is strictly necessary; but the net gain in computational efficiency is still judged to be worthwhile. The probability functions for selection of the "eligible" atoms are not altered by this procedure.[19]

The final issue here is, having generated a trial m of M, how to pick the appropriate m atoms. This is done in practice by repeatedly picking a random value in the range $0..(R-1)$ and using this as an index into the state string (viewed now as an array rather than a linked list). In itself, this runs the risk that a single atom could be selected more than once for a given operation within a single time step (it represents selection with replacement, rather than without replacement as required). This problem is overcome by tagging each atom which has been selected (for the given operation, on this time step); if such an atom is, by chance, reselected, that reselection is discarded, and another attempt made etc. Obviously, this process could become very inefficient if m could be comparable to R—but this does not arise in the cases of interest in `AVO`. The tagging process is made efficient by using a time stamp rather than a simple binary tag: this obviates the need to clear the tags on each time step.

[19]There is one minor qualification of this. The two distinct phases of `BM` (strong-to-weak and weak-to-strong) are executed sequentially. To keep the respective probability functions precisely as described by Holland it is technically necessary to ensure that a bond made weak in the first phase is not made strong again in the second phase of the same time step (albeit, with the typical parameter values, such an event would be extremely rare anyway). This is achieved by tagging each bond which is actually modified during the strong-to-weak phase; this tag can then be checked during the weak-to-strong phase. This tag is qualified by a (strong-to-weak) time stamp, so that it need not be explicitly cleared on each time step.

5.4.6 Emergent Operators

The simplest approach to the implementation of the emergent operators is to scan the entire state string, identifying (and processing) any emergent operators encountered. This is, indeed, the basic approach adopted in AVO. However, this scheme is modified in two ways, in order to improve the execution speed.

First note that, from the definition of the E-OPs, they all necessarily incorporate at least one colon atom. Now, typically, only one in 10 atoms in AVO are colon atoms. Thus, instead of scanning the entire state string on each time step it is significantly more efficient to selectively target the scan onto the colon atoms. This is achieved by creating an index of all colon atoms; this need only be done once per run, at initialisation, as the "positions" of these atoms, in the *array* view of the state string, will not change thereafter. Then, on each time step, it is sufficient to inspect the immediate neighbourhood of each colon atom, in turn, to establish whether it is part of an E-OP. Again, a time stamp mechanism is used to ensure that a given E-OP (which may contain more than one colon atom) is not processed multiple times on any single time step.

The second, and ultimately more significant, optimisation is concerned with the location procedure for raw materials, which is an essential part of the execution of all E-OPs. Recall that there is no limit on how far this search may extend, possibly spanning the entire state string. The lower the density of the required raw materials, the more extended these searches will become. It turns out, for reasons to be explained later, that the densities of free atoms, particularly colon atoms, tend to quickly become quite small, and can be zero for a significant fraction of the time. Clearly, if there are no free atoms of a particular kind present in the state string at all, then the location procedure (for such a free atom) is guaranteed to fail; and should not even be initiated. This is arranged by establishing and maintaining counts of the free atoms of each element currently present in the state string. These must be correctly updated by the EX operator, and all E-OPs. For experiments in which the density of E-OPs is high, this optimisation has been found to yield approximately a five-fold increase in execution speed; in other cases the improvement is less dramatic, but is still worthwhile.

5.4.7 Tracking Complexes

In the experiments to be described it is essential to track the densities, in the state string, of specified complexes (specifically of `PartSR` and `FullSR`). In fact, it turns out to be sufficient to track an upper bound on these densities, which proves somewhat simpler to implement.

A first point to note is that, instead of literally attempting to assess the density of a complete complex, we track only the density of some one structure in the complex; clearly this does establish an upper bound on the density of the entire complex.

Secondly, note that we must decide *which* structure to track for any given complex. For both complexes of interest here, codon structures have been (arbitrarily) selected for tracking. In the case of `PartSR` the complex only includes a single codon structure (`-`$N_1N_0N_0N_1N_0N_0$`-`) so no further choice is necessary; in the case of `FullSR` the particular codon tracked was arbitrarily selected as `-`$N_0N_0N_1N_0N_0N_0N_0N_0N_0N_1$`-`.

However, some care must still be taken in identifying instances of these codon structures. Since codon structures are dynamically incorporated into the `E-OP`s and are punctuated by colon atoms (used as position markers), the tracking algorithm is designed to recognise the codon strings regardless of surrounding context, and regardless of the presence of a (single) embedded colon atom. Again, while this is not a completely reliable procedure, it will clearly yield a satisfactory upper bound for the density of the codon strings, and thus of the complexes, of interest.

5.5 Playing God

5.5.1 The Predictions

To recap, there are three substantive elements to Holland's predictions:

1. The `Seed` complex will spontaneously appear within a relatively short time (of the order of 10^9 time steps).

2. Once the `Seed` complex *does* appear, a population of `PartSR` complexes will be established, and will maintain themselves.

3. Conventional Darwinian evolution can then optimise the reproducing ability of the complexes quite quickly, up to and including the possible emergence of the `FullSR` complex (or something comparable).

Of these, the first potentially requires a substantial amount of (real) time to test; the second can be easily tested (by "playing God"—directly inserting an instance of the `Seed` complex); and the third can be tested only when (or if) testing of the second has been successful (i.e. after prediction 2 has been verified). Therefore, testing concentrated, in the first instance, on prediction 2—whether the `Seed` complex can establish a viable population of `PartSR` complexes.

5.5.2 Parameter Values

As already discussed in the detailed definition of α_0, Holland (1976) stipulated particular values for the α_0 parameters, which he then used in his numerical calculations. In summary, these values are as follows:

$$\begin{aligned} R &= 10^4 \\ \rho &= 0.5 \ \ (\Rightarrow \rho(-) = 0.5) \\ \rho(\texttt{0}) &= \rho(\texttt{1}) = \rho(:) = \rho(N_0) = \rho(N_1) = 0.1 \\ r &= 10^{-4} \\ \lambda &= 7/3 \\ m_1 &= 10^{-2} \end{aligned}$$

With minor exceptions, which will be noted below, these values were consistently adhered to in all the experiments to be described.

Holland did not specify any numerical value for the parameter m_2; in his Lemma 2 he suggests that his results will be insensitive to its exact value provided only that $m_2 < 1/b$ where b is the number of weak internal bonds in the structure(s) of interest. The longest structures to which Holland subsequently even loosely applies this analysis are the codons of the `FullSR` complex, which are each 10 atoms long, and thus have no more than 9 weak internal bonds. The condition $m_2 < 1/b$ is thus guaranteed to be satisfied, in all relevant cases, if we have $m_2 < 1/9$. It is desirable not to make m_2 much smaller than necessary, as this progressively slows the execution of the `EX` operator, and of `E-OP`s in general. Bearing this in mind, a value of $m_2 = 0.1$ in used in all the experiments described below.

I should note here that, in any case, I have been unable to follow the derivation of Holland's Lemma 2; and that I have carried out both theoretical and empirical analyses which suggest that it may be mistaken in detail. However, this seems to be a relatively minor issue, which would not critically affect Holland's predictions; therefore it will not be pursued further.

Four distinct experiments will be described. In each case, results are presented for two distinct runs of `av0run`; these runs are distinguished only in that the pseudo-random number generator was seeded with a different value, both in the randomisation of the initial configuration and the actual execution of `av0run`. The distinct seeds were chosen such that there was no overlap in the sections of the pseudo-random number cycle traversed within each pair of corresponding runs; that is, the runs used completely *distinct* sequences of pseudo-random numbers. The only purpose of this procedure is to demonstrate that, in each case, the essential pattern of the results does not rely on any artefact of the particular pattern of pseudo-random numbers encountered.

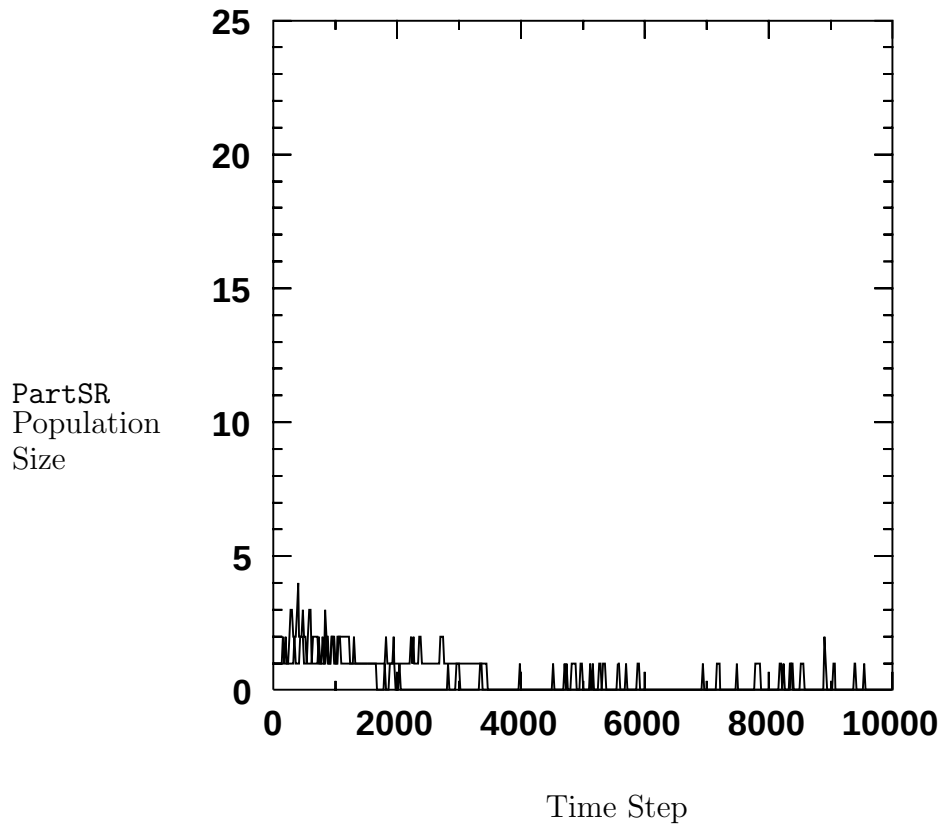

Figure 5.1: *The* `Seed` *Complex:* In this case, a randomised configuration is initially generated, and a single instance of the `Seed` complex is artificially inserted. The graph shows a superimposition of results from two runs, with disjoint initialisation of the pseudo-random number generator. Contrary to the original prediction, a large population of `PartSR` is *not* established.

5.5.3 Experiment 1: The `Seed` Complex

An α_0 of 10^4 cells was generated with a random initial configuration. A single instance of `Seed` was then inserted (this increases R slightly, and also slightly alters the element densities; but the effect is negligible). 10^4 steps were executed. This was repeated with the alternative seeding of the pseudo-random number generator.

The results of the two runs are shown (superimposed) in Figure 5.1. It shows the number of `PartSR` complexes present over time (as tracked by the number of `PartSR` codon structures). It is seen that, contrary to expectations, a significant population of `PartSR` was *not* generated. Indeed, the greatest `PartSR` codon string population achieved (itself an overestimate of the density of the complex proper) was only 4 instances.

5.5.4 Experiment 2: The *Modified* `Seed` Complex

On examining the detailed behaviour of the `Seed` complex it was found that, quite typically, the `CP E-OP` was *failing* to bind the `PartSR` codon, but was, rather, binding other "garbage" codons. But, reviewing Holland's analysis of `Seed`, we find a claim that the 3-atom argument of the `CP E-OP` *should* be sufficient "to assure that it will preferentially attach to the single copy of $\gamma(:\alpha(N_1N_0))$" (Holland 1976, p. 399).

In fact, it seems that Holland's analysis here is mistaken. Following the logic of Holland's own Lemma 1 (though not the precise result) the expected density of structures having the prefix `100` is $0.5 \times 0.1^3 = 5 \times 10^{-4}$. Thus, in a "region" of 10^4 atoms we would expect there to be approximately 5 distinct, garbage, codon structures which can be potentially bound by the `Seed CP E-OP`. Thus `Seed` could well fail to reliably reproduce the correct (`PartSR`) codon structure, and this indeed seems to be what is happening. Note that this effect is *not* offset by initially placing the two structures of the `Seed` complex immediately beside each other: with the relatively small value of m_2 in use (0.1), proximity has only a very limited effect on the binding probability.

To investigate this further, the previous experiment was repeated, but with the `Seed` complex *modified* by adding a further atom to the argument of the `E-OP` (the argument now becoming `1001`). The expected number of garbage codon structures, matching this argument, in a universe of 10^4 atoms, now falls to 0.5, so the desired specificity should be achieved. It should be noted that this modification will also increase the expected emergence time of the modified `Seed` complex (though that is not, of course, at issue for these particular experiments); and that the expected lifetime of the modified `Seed` complex will be somewhat reduced thus reducing the maximum density of the `PartSR` complex which could be generated. But, for the moment, the important requirement is to establish a significant density even of the `PartSR` codon structure.

Figure 5.2 is a plot of the outcome of this experiment. While there *is* now a significant generation of `PartSR` complexes (or, at least, of its codon structure), it is clear that this effect is still very limited. The greatest `PartSR` codon string population achieved was 18 instances, whereas this particular universe has a

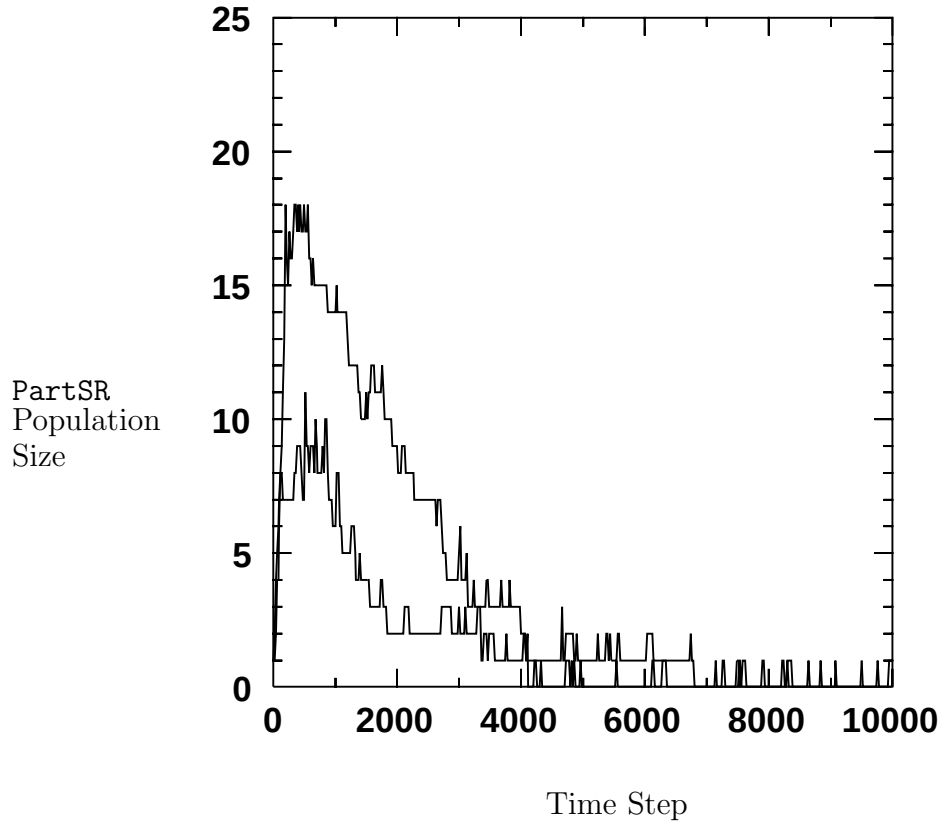

Figure 5.2: *The Modified* `Seed` *Complex:* In this case, a randomised configuration is initially generated, and a single instance of the *modified* `Seed` complex (see text) is artificially inserted. Again, the graph shows a superimposition of results from two runs, with disjoint initialisation of the pseudo-random number generator. While a population of the `PartSR` complex (or, at least, its codon string) *is* now initially built up, it subsequently dies out again relatively quickly.

theoretical capacity for 250 instances. It is clear that the modified `Seed` complex does not come close to saturating the universe in this sense. Furthermore, after the initial transient, the population rapidly dies out.

5.5.5 Experiment 3: The `PartSR` Complex

At this point it was clear that the `Seed` complex was not capable of carrying out the function anticipated by Holland—i.e. to establish a viable population of `PartSR` complexes. However, it was not clear whether this was merely a problem of the relatively limited size of `PartSR` population which the initial instance of `Seed` was managing to generate, or whether the `PartSR` complex would not be viable even in an established population.

To test this, an α_0 was generated (via the `partsr` program) with a highly artificial initial configuration—essentially (80%) saturated with instances of the

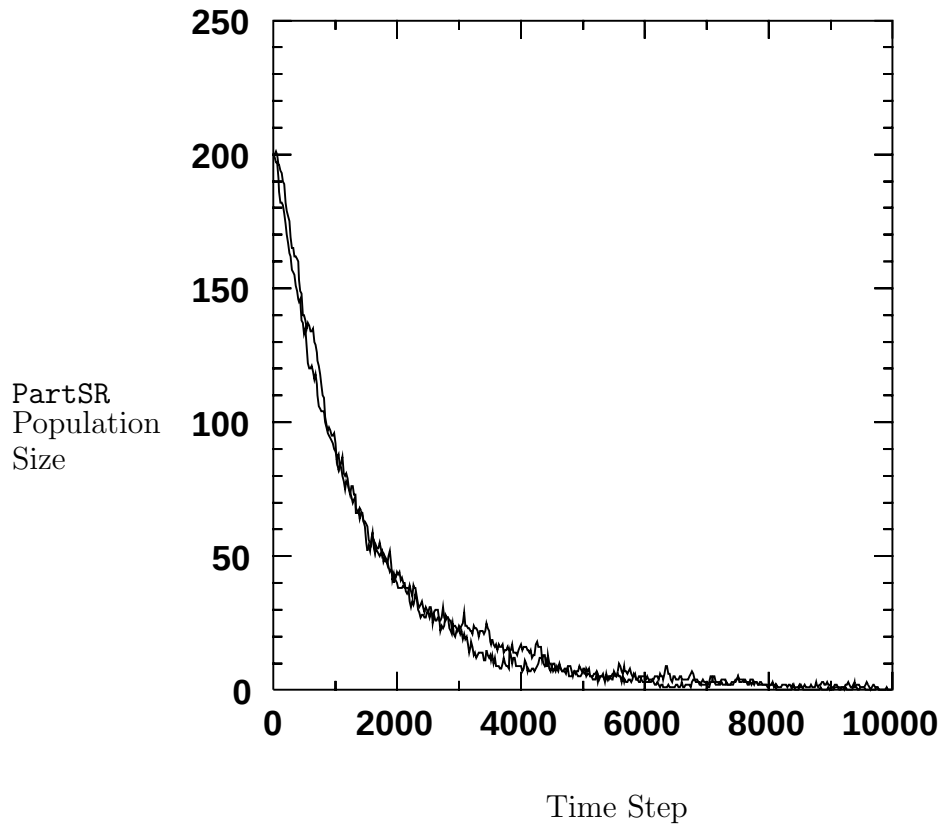

Figure 5.3: *The* `PartSR` *Complex:* In this case, an initial configuration is generated by the program `partsr`; this consists solely of instances of the structures making up the `PartSR` complex and of free atoms. Again, the graph shows a superimposition of results from two runs, with disjoint initialisation of the pseudo-random number generator. Contrary to the original prediction, the population of `PartSR` is *not* sustained, but dies out rapidly.

`PartSR` complex. This was again executed for 10^4 steps. Figure 5.3 shows the outcome. It is seen that, even with this "most favourable" configuration, the population still rapidly goes extinct.

5.5.6 Experiment 4: The `FullSR` Complex

It will be recalled that the `PartSR` complex has the property of not being "fully" self-reproducing: it relies on the primitive operators to complete its cycle of reproduction. A final experiment was carried out to test whether this was a critical factor in the failure of `PartSR` to be viable. In this case, an α_0 was generated (via the `fullsr` program) which was saturated with instances of the `FullSR` complex (the maximum capacity is 35 instances).[20] This was again executed for 10^4

[20]The size of this universe was made marginally larger than 10^4. A basic complex consisting of a single instance of `FullSR` plus sufficient free atoms to correctly set the relative element

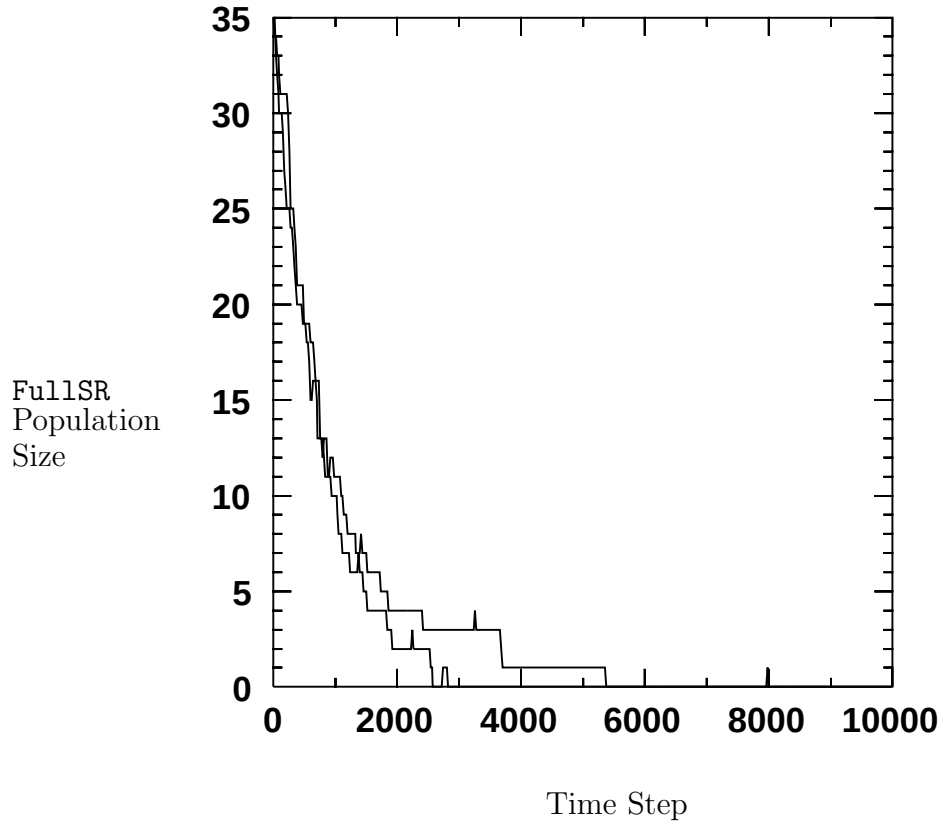

Figure 5.4: *The* `FullSR` *Complex:* In this case, an initial configuration is generated by the program `fullsr`; this consists solely of instances of the structures making up the `FullSR` complex and of free atoms. Again, the graph shows a superimposition of results from two runs, with disjoint initialisation of the pseudo-random number generator. As with the `PartSR` complex, and contrary to the original prediction, the population of `FullSR` is *not* sustained, but dies out rapidly.

steps. Figure 5.4 shows the outcome. It is seen that, even with this "fully" self-reproducing complex, the population still rapidly goes extinct; indeed, the extinction is, if anything, somewhat more rapid for this complex.

5.5.7 What's going wrong?

From the experiments described above, it was clear that α_0 was simply not capable of supporting the "life-like" behaviour postulated by Holland. There was thus no point in pursuing the question of spontaneous emergence. But these experiments do not, in themselves, give any indication of how deep rooted (or otherwise) the deficiencies of α_0 may be.

densities turns out to require 290 atoms. The size of the universe must then be made an integral multiple of this $(10, 150)$ to correctly retain these densities.

A series of informal studies were then carried out, which involved simply monitoring a dynamic display of a window onto the α_0 state string, over many different variations in the configuration and parameters of the universe. Based on this exercise it was possible to identify at least some specific, proximate, causes of failure (there can, of course, be no guarantee that *all* relevant factors were identified by this process).

Note that Holland's analysis of the `PartSR` dynamics relies on its being composed of structures which are, internally, "strongly bonded"—which is to say *long lived*. He then estimates the average "productivity" over this lifetime, to come up with a net positive rate of change for the density of the complex (once a threshold is reached). However, in practice, there are (at least) three factors which severely disturb the behaviour of the complex, and which are not allowed for in Holland's analysis:

- Raw materials (free atoms, in particular) quickly become scarce (due to usage by random, garbage, emergent operators). The effect is to drastically reduce the rate at which all emergent operators function in practice, thus reducing the *fecundity* of any putatively self-reproducing complexes.

- Even when a structure is strongly bonded internally, there is nothing to stop random garbage moving into a position immediately adjacent to it. At the very least this interrupts or suspends the progress of an emergent operator. Thus, it turns out that complexes can only be *active* for a limited portion of their total lifetimes (regardless of the availability of free atoms); again this severely limits fecundity.

- But, at worst, this random arrival of a garbage structure beside an emergent operator can have much more severe effects. If it arrives on the right hand side it can corrupt the output of the operator (introducing a high "mutation" rate, and further reducing fecundity). If it arrives on the left hand side it can result in the formation of a different, garbage, emergent operator which forcibly, and prematurely, breaks up the original operator. This has actually been observed to occur on a number of occasions. Thus, as well as reduced fecundity, complexes also have higher mortality than expected.

So: compared to Holland's analysis, the lifetimes of the structures are shorter than expected, they are only active for a fraction of this time, and their products are quite frequently corrupted. The net effect is that mortality exceeds fecundity (by a significant margin), the putatively reproducing complexes cannot make up for their own natural decay rate, and thus become extinct quickly. These effects are directly related to the time required to complete a reproduction cycle, and thus to the size of the complexes. This explains the even worse performance of the `FullSR` complex compared to `PartSR`.

Note that I have not presented any experimental investigation of the sensitivity of my quantitative results (experiments 1–4 above) to the parameters of α_0, even though this would be a straightforward (if tedious) exercise. The reason for this omission should now be clear: Holland's analysis has been demonstrated to be greatly oversimplified, and defective as a result. The failure of the predictions is not dependent on the particular parameter values used, but is, rather, representative of the fact that several significant factors have been entirely neglected in the analysis. This effectively destroys the assumed theoretical basis for the empirical investigation; in this situation a random search through the α_0 parameter space for a system which would show some "interesting" behaviour seems to me quite futile, and I do not pursue it.

Thus, even though the original objective was to design an artificial system which *would* be simple enough to allow closed form analysis, it turns out that α_0 is just not that simple. At this point it seems doubtful to me that a system which was genuinely simple enough to allow the kind of analysis envisaged by Holland, would actually support any of the phenomena of relevance—though this remains an open question. I should note that Holland's own diagnosis of the situation is not quite as pessimistic as mine; commenting on the preliminary results of the current work (McMullin 1992d) he stated:

> When I wrote my original version I didn't make any allowance for diversionary "precipitates" (and the like) that would sequester needed intermediates ... several of my colleagues in various places have said in one way or another that this is a central problem in the origin of life. I think some revisions in the model would "fix this up" but it takes some thought (and I would make no guarantees).
>
> John Holland (personal communication)

In any case, leaving aside the question of closed form analysis, a naïve attempt at solution of the particular problems observed in α_0 might be to simply reduce the "temperature" of the universe (reduce the rate at which bonds decay, and structures get randomly moved around). It seems likely that the `FullSR` complex could be made "viable" in this way (in the limit, if the primitive operators are disabled entirely, `FullSR` should be able to expand to the capacity set by whatever free atoms are initially available; thereafter, of course, all dynamic activity would cease); however, this is not at all the case for `PartSR`, which relies on the primitive operators to complete its reproduction. It is quite possible (though no proof is currently available) that `PartSR` would not be viable at *any* "temperature". However, in any case, from the point of view of the problem originally posed by Holland (i.e. that of spontaneous emergence) any reduction in "temperature" would be accompanied by an increase in the expected emergence time for any particular structure or complex, and thus may be completely counterproductive.

But there is a more general point here: my purpose in studying α_0 was not concern for the problem of spontaneous emergence *per se*, but as an avenue to the solution of P_a—the demonstration of A-reproducers which are *robust* in the face of environmental perturbations (including interactions with each other). Thus, while one might well be able to improve the viability of certain complexes in α_0 by *ad hoc* measures which "protect" them from interference, this would be to undermine completely the purpose for which I turned to the α-Universes in the first place. The specific perturbations identified above, which arise in α_0, are precisely the kinds of things we *want* to allow. Again, as noted at the end of the previous chapter, we already *know* that we can achieve "viable" A-reproducers if we rule out, or rigidly constrain, their interactions with each other and their common environment, so modifications of α_0 which move in that direction are fundamentally of limited interest.

In summary, α_0 does *not* yet provide any substantive advance toward a solution of P_a. The A-reproducers in α_0, such as they are, are just as fragile as in any of the A-systems considered in the previous chapter; contrary to Holland's analysis, α_0 does not provide any prospect for the spontaneous emergence of *robust* A-reproducers, and does not, therefore, provide a basis for the realisation of Artificial Darwinism.

5.5.8 Can We Fix it?

I should note here that the original paper (Holland 1976) seems to have been largely ignored since its publication. I have been able to identify only two substantive discussions of it: by Martinez (1979) and Kampis (1991, Section 5.1.2). In both these cases the correctness of Holland's analysis was *assumed*, and further discussion was then predicated on that. Given the results which have been presented here, this assumption was not justified; I shall not, therefore, comment further on these works.

I do not, of course, *know* how one might best proceed in the light of the results which have been presented here; but there are two distinct avenues which seem to me worth considering further.

Firstly, it seems that at least one part of the deficiency of α_0 hinges on the fact that von Neumann style reproduction involves *copying* and *decoding* an information carrier, where the decoding must be such as to generate (at least) a copy of the required copying and decoding "machinery". α_0 fails to sustain this kind of behaviour because (*inter alia*) the maximum information capacity of its carriers (in the face of the various sources of disruption) seems to be of the order of perhaps 10 bits, which is insufficient to code for any worthwhile machinery—even the relatively simple copying and decoding machinery constructible in α_0.

A more plausible model for the spontaneous emergence of properly genetic A-reproducers *might* therefore involve a universe in which certain information carriers, of capacity (say) an order of magnitude larger than that required to code for minimal decoding machinery (in the particular universe), can be copied *without any specialised machinery at all.* In such a system there may be potential for a Darwinian evolutionary process to begin more or less immediately, in which more sophisticated phenotypic properties might, incrementally, become associated with the information carriers—possibly then culminating in a full blown "decoding" (or embryology).

This is, of course, rather speculative; but, as it happens, it is closely related to a general model for the origin of *terrestrial* life which has been championed by Cairns-Smith (1982). This is based on *inorganic* information carriers, which could conceivably be replicated without the relatively complex apparatus required

for RNA or DNA replication. It seems to me, in the light of the experimental results presented here, that it would now be a promising research program to adopt Holland's original *strategy* (which is to design relatively simplified model chemistries, loosely based on cellular automata, in which to examine the origin of "life"), but to replace his detailed models (the α-Universes) with models based on different theoretical considerations—such as those of Cairns-Smith.

The second avenue I can envisage for challenging the limitations of α_0 turns on a point which is both subtle and fundamental. I had already raised, or at least anticipated, this issue in general terms in the previous chapter (section 4.3.4), and I referred to it again, albeit obliquely, earlier in the present chapter (section 5.3). In the specific context of α_0 it may be expressed in the form of a question: should complexes, such as `FullSR`, be properly regarded as realising self-reproduction (as I have done up to this) or, instead, as realising a primitive form of autopoiesis?

Briefly, the situation is this. As long as we consider an instance of an A-machine in α_0 as corresponding to a particular, fixed, set of structures, then it makes sense to regard the mutually recursive relations of production between these structures as realising a form of self-reproduction—such a set of structures is (in principle at least) capable of bringing new and separate instances of such sets into existence. But this is not the only possible way of looking at things. We could, instead, regard an A-machine in α_0 as corresponding to the set of recursive relations of production rather than a particular set of structures which happen to realise these relations. In effect, an A-machine is then identified with what I have previously regarded as a *population* of structures (or complexes) in α_0. These relations of production are then recognised as being autopoietic: such a population is (or, at least, should be) capable of sustaining itself, by virtue of this autopoietic organisation, despite turnover of some or all of its constituent structures.

From this perspective, the phenomena studied in α_0 can now be recognised as fundamentally related to phenomena occurring in, say, the `VENUS` (Rasmussen *et al.* 1990) or `Tierra` (Ray 1992) systems, discussed in the previous chapter (see sections 4.3.1 and 4.3.3). In the present chapter I have very loosely talked in terms of the putative A-reproducers in α_0 as being potentially "robust" or "viable"; but the fact is that, as long as by "A-reproducer" I meant a single fixed

set of structures, there was never any possibility of their being "autonomous" in the strong sense of being *autopoietic.* As it happens, the putative α_0 A-reproducers turned out not be "viable" anyway (just like the A-machine `MICE` in `VENUS`); but, even if they had been "viable", it seems that it could only have been, at best, the cosseted "viability" of the A-reproducers in `Tierra` with their inviolable memory allocations. By definition, no *static* set of structures in α_0 can realise the *dynamic* homeostasis of its own identity, which would be characteristic of properly autopoietic viability or autonomy.

By contrast, if we turn our attention to "populations" of structures in α_0—the equivalent of considering "organisms" in `VENUS` or "sociality" in `Tierra` (see Chapter 4, section 4.3.4)—we *can* encounter the possibility of properly autopoietic organisation. Granted, in α_0 as it stands, the autopoiesis is not effective—such populations actually die out—but (with the example of `Tierra` before us) we may anticipate that some modified α-Universe could overcome this. The point is that the kinds of entities which we might properly regard as autonomous are *not* the kinds of entities which could be regarded as self-reproducing; and, moreover, the "higher level", properly autonomous entities, are not, in general, self-reproducing in any sense, and are *certainly* not genetically self-reproducing in the von Neumann sense of permitting an open-ended growth in complexity.

Can we envisage a path toward making the properly autonomous entities ("organisms" in `VENUS`, "social systems" of `Tierra`, "populations" in α_0) self-reproducing, in the von Neumann sense?

Well, the first point is that to have *any* kind of self-reproduction, we would probably need some mechanism for the formation and maintenance of *boundaries* by the autopoietic entities. Some kind of boundary formation is actually part of the definition of fully fledged autopoiesis. Furthermore, a boundary seems to be logically necessary if we wish to talk about self-reproduction: unless the entities establish well defined boundaries then it is entirely unclear what could possibly qualify as self-reproduction. In `VENUS`, `Tierra`, or α_0, as they stand, there are no such mechanisms for boundary formation (capable of bounding the *relevant* entities). Boundary formation has, of course, been exhibited in the A-systems pioneered by Varela *et al.* (1974). These systems, by contrast to `VENUS`, `Tierra` and α_0, are *two* dimensional rather than linear. On the other hand, the

introduction of a kind of boundary mechanism has been previously outlined by Martinez (1979), in a modification of α_0 which would still be one-dimensional. Thus, while two-dimensionality is probably not essential here, it certainly provides conceptual simplification, and makes visualisation much easier.

Incidentally, it seems plausible that the introduction of an appropriate boundary mechanism could positively help in overcoming the primary deficiency of α_0 identified by the empirical tests described above, that even the putatively autopoietic populations cannot actually sustain themselves.

In any case, assuming the introduction of mechanisms allowing for the construction and maintenance of such boundaries, it is clear that self-reproducing autopoietic entities can be established, in the manner already described by Zeleny (1977). Briefly, once one has a bounded autopoietic entity of any sort then, since it already incorporates processes capable of reestablishing all its component relationships, it should be a relatively trivial matter to arrange for it to progressively grow *larger.* Once this is possible, then one need only add a mechanism for the boundary to rupture in such a way that it can be reformed into two closed parts, and a primitive form of self-reproduction is achieved. There seems no reason, in principle, why this general kind of process cannot be achieved in A-systems derived from the `VENUS`, `Tierra` or α_0 models.

Doing this based on the `VENUS` or `Tierra` models would yield a form of self-reproduction which might still be said to be *impoverished* in the sense that, insofar as "information carriers" are being reproduced, this is occurring by self-inspection, without any overt genotype/phenotype distinction, or von Neumann style *decoding.* Still, although I have arrived at this from a completely distinct direction, this idea actually corresponds rather closely to the first suggestion which I outlined in this section, following Cairns-Smith (1982), of arranging for the possible existence of reasonably high capacity "information carriers" which could be "reproduced" without the aid of any special or elaborate machinery. It may thus be a useful, and perhaps even essential, step toward more sophisticated self-reproduction techniques.

Conversely, if we used α_0 as our starting point, and succeeded in modifying it to support reproduction of bounded, autopoietic, "populations", then we would have entities which *do* exhibit a "von Neumann style decoding"; but, of course,

they would be impoverished in a different manner, namely that the functionality available in α_0 is extremely impoverished anyway and there certainly could not exist a space of such autopoietic A-reproducers which would span a wide range of A-complexity (see the previous discussion of this point in section 5.3 above).

This is all rather vague and informal, and I do not pretend that it has more than heuristic value. Nonetheless, it seems that there may be some limited grounds for optimism here. If the various phenomena which have been separately exhibited in this diverse range of A-systems can be consolidated into a single system, then it seems that some significant progress may then be possible in the solution of P_a.

5.6 Conclusion

> The popular view that scientists proceed inexorably from well-established fact to well-established fact, never being influenced by any unproved conjecture, is quite mistaken. Provided it is made clear which are proved facts and which are conjectures no harm can result. Conjectures are of great importance since they suggest useful lines of research.
>
> Turing (1950, p. 442)

I should like to emphasise the debt which the work reported here owes to John Holland's original formulation and analysis of the problem of spontaneous emergence of self-reproducing behaviour. While it has been possible to point to defects in that analysis, this was with the benefit of hindsight, and prompted by experimental evidence not available to Holland. It does not detract in any way from Holland's creative achievement in formulating the *possibility* of such an investigation in the first place.

In conclusion, this chapter is a report on failure—but, I suggest, in the very best and most productive sense of that word. As enunciated by McDermott in the quotation with which I opened the chapter—and, indeed, as encapsulated in the Popperian theory of the evolutionary growth of knowledge—failure, or experimental refutation of predictions, is the very stuff of the so-called "scientific method". Although the model universe α_0 fails to demonstrate the phenomena originally hoped for, its particular mechanisms of failure are interesting in their own right. These show that P_a continues to be a deep and intractable problem *even* in a universe which is extremely simplified, and where the dynamics

have been deliberately tailored to make von Neumann style genetic reproduction "easy" to realise.

However: all human works are finite, the end of the rainbow steadily recedes, and we have finally reached that otherwise arbitrary point where a halt must be called to this cycle of conjecture and refutation. It only remains, in our concluding chapter, to briefly look back and consider a final, distinctive, view, which can be made available now that we have arrived at the end point of this particular journey.