Contents

Chapter 4

You Can't Get There from Here

In 1930, Kurt Gödel announced the proof of a theorem of logic that shocked the mathematical community because of its profound implications.[1] He proved that in any consistent system of logic, there are true statements that can be properly stated within the constraints of the system but which cannot be shown to be true without going outside the system.

A conceptual way to understand Gödel's theorem is to imagine all the true statements that can be made as points on a graph (or locations on a road map), and logical inferences as connections between the nodes (or roads between towns). The inferences are one-way connections between nodes, allowing movement from one to another, but not necessarily in the reverse direction. Gödel's theorem is a statement that on such a graph (or on such a road map) there are places you can be that you can't get to from anywhere else. Some statements are true but cannot be proven to be true. It is the opposite of the statement that all roads lead to Rome, namely, there is always a point away from which all roads lead.

The implications of the incompleteness theorem are profound for several reasons. Perhaps most striking, the hope of the rationalist, to accept as true only things that can be proven, cannot be achieved. In more practical terms, there are things that you and I think are true but which can never be conclusively proven as true. This is certainly the case with many features of Darwinism. There are some things about Darwin's Computer that are true or false but which can never be proven as such. Some questions cannot be resolved; some controversies cannot be settled.

The incompleteness theorem of Gödel took on even greater significance with

[1] K. Gödel, Uber formal unentscheidbare Satze per Principia Mathematica und verwandter Systeme I. Monatshefte fur Mathematik und Physik, 38, 173-98 (1931). See also K. Gödel, Collected Works, Vol. I (publications 1929-1936) (ed. by S. Feferman et al.) Oxford University Press (1986).

the work of Alan Turing. Turing was interested in, among other things, what we now call computational complexity. In the appendix to this chapter, I've included Turing's proof of Gödel's theorem, because it is elegant and demonstrates some issues related to computational complexity.

One interesting feature of computational complexity is that, even when problems are solvable, it is often easier to verify that you have a solution to a problem than it is to find the solution. Suppose it is Saturday evening and you arrive at a big party. Feeling shy, you wonder whether you already know anyone in the room? Your host proposes that you must certainly know the lady in the corner next to the dessert tray. In a fraction of a second you are able to cast a glance and verify that your host is correct. However, in the absence of such a suggestion, you are obliged to make a tour of the whole room, checking each person one by one, to see if there is anyone you recognize.

A more complicated version of this problem is to determine the shortest chain of personal acquaintances (defined by handshakes) that links any two people in the world. Similarly, mathematicians have defined the Erdös number to be the smallest number of published collaborations (coauthors) that connect to the famed number theorist Paul Erdös (1913-1996).

Think for a minute about computing my Erdös number. First, you must make a list of all my collaborators[2], then you must add to the list all of their collaborators, and then all of their collaborators, and so on. The computation ends when Erdös shows up on the list or the list ceases to grow. At each level the computation increases in complexity exponentially, and although an Erdös number is not too large, the computation of it is difficult. [3]

The computation to link two people in the world by handshakes is much more difficult, since the number of people in the world is roughly 5 billion, and there are many more possible connecting routes to check, even though the length of the shortest connecting path is believed to be fairly small, less than ten.

There are other problems which, although solvable in a finite amount of time, this time is larger than the computational resources of the universe, and so from a human point of view, i.e., within the constraints of our ability to compute, cannot be solved. For example, there are prime numbers which are so large that they cannot be verified as prime, since the algorithm to check that they are prime requires more computational resources than is or can ever be available.

Other algorithms fail because they can solve problems within a certain class, but are useless for problems outside of that class. Thus, a bisection algorithm (sometimes also called "divide and conquer") is not appropriate to determine tomorrow's weather, because the governing equations cannot be formulated in a way that the bisection algorithm applies.

Since the Darwinian mechanism is an algorithm, there are some things that

---

[2] of which there are 27.

[3] It is known that the median Erdös number is 5, and the largest is 15. There are 208,201 people with an Erdös number, and 45,142 mathematicians who have co-authored a paper but do not have an Erdös number. For more information about the Erdös number and some data see http://www.oakland.edu/ grossman/erdoshp.html.

it does well and limitations on what it can compute or construct. Any numerical analyst or computer scientist understands these kinds of limitations. There are some problems that Darwin's computer simply cannot solve, given its computational restrictions.

To illustrate how an understanding of the limitations of an algorithm can be used, suppose I assign a problem to my math class, due the following day. The assigned problem is to solve a particular system of equations, in this case a very large system of linear algebraic equations, say, a million equations with a million unknowns. The data for these equations is contained in a computer file that the students must access.

The next day, after students hand in their solutions (via a computer file which I access on the network), suppose that I interview each of them briefly to ascertain the method by which they found their proposed solution. Some students respond that they used an iterative method in which one guesses a solution and then refines the guess by a specific algorithm. Because such methods are known to work well, I am content with the student's answer. Similarly, if the student responds that he used a multi-grid technique or some other appropriate buzzword, I am contented. However, if the student replies that he/she used a direct solution technique (called Gaussian elimination), then I am dubious, because for this algorithm it is known how long the computation will take (about 1000 years on a fast workstation!), and there is clearly not enough time in a day for this to have been accomplished.

Matters are even worse for the student who responds that he/she guessed the solution and it just happened to be correct. Call it incredible luck if you wish, but I am not convinced.

The student who claims to have used the Darwinian algorithm is in equally bad shape. First, the student will have to convince me that the algorithm can be made to apply to this problem and then I will have to be convinced that the algorithm as applied would actually work in the amount of time allotted. I am not impressed with the answer, "Well, I guess I just got lucky." It is more likely that the student had some form of help from some source that he/she is not willing to admit.

Since we know something about how random search algorithms work, and since we know something about the structure of the genome, we can know the kinds of computations for which Darwin's computer is ideally suited and those which are beyond its capability. In a later chapter, we will discuss some computations for which Darwin's algorithm is ideally suited and works extremely well. In this chapter, however, my goal is to give a sense of the computational complexity associated with Darwin's algorithm and the development of life forms. In fact, it is my contention that, because of insurmountable computational complexity, the Darwinian algorithm is not the mechanism by which all existing life forms came to be.

To see why this is the case, we need an understanding of the problem that needs to be solved. The Darwinian process can be described as an undirected random walk on a highly irregular fitness landscape.[4] Different possible arrangements of DNA are viewed as points in a space, and as the population changes and as mutations occur, different regions of that space are occupied. The intrinsic biochemical fitness of the different arrangements gives a topography or landscape to the space. That is, those arrangements of DNA that are biochemically viable are identified of as high spots and those that are not biochemically viable are low spots.

This genomic space is subdivided into islands where survival is possible, much like Monument Valley in Northern Arizona, with randomly scattered fitness "mesas" that poke up above the uninhabitable desert floor. Specifically, by a fitness mesas, I mean the set of all DNA configurations that permits a viable, reproducing organism, fit in the sense that it is biochemically functional. Biochemical fitness is observable in a laboratory, where all of the external, environmental needs of the organism are met, independent of other organisms, and is therefore a feature that is independent of time.[5] Biochemical fitness implies nothing at all about environmental or cooperative/competitive fitness. The fitness desert is that region of DNA configuration space that, for biochemical reasons, does not permit a viable, reproducing organism. A single mesa is used to depict any set of viable points that are connected by single beneficial or neutral mutations, while the boundary of a mesa consists of those points where a single mutation is fatal. Notice that the topography of this fitness surface is quite contorted, as every point on a mesa is a boundary point (since there are single mutations that are fatal to any organism[6]). However, there are many points of the fitness desert that are not adjacent to mesa points (since an arbitrary non-viable strand of DNA probably requires many mutations to render it viable). In fact, the vast majority of points are desert points, and mesa points are few and far between, and extraordinarily difficult to locate by a random search. The density of viable points in this genomic space is not known, but it is, without doubt[7], quite small.

The second feature of this genomic space is that currently existing forms occupy mesas that are separated by a large number of mutations. Some species

[4]The metaphor of a fitness landscape for Darwinian evolution is quite old, created in 1932 by Sewell Wright, The roles of mutation, inbreeding, crossbreeding and selection in evolution, in: Int. Proc. 6th Int. Cong. on Genetics, ed. D. F. Jones, Vol. 1, 356-366 (1932). This metaphor is extremely powerful and useful in many areas of science. See, for example, the proceedings of the 16th Annual International Conference of the Center for Nonlinear Studies at Los Alamos National Laboratory, published in Physica D 107, pp.2-4, 1997.

[5]Here I am assuming that fundamental physics and chemistry has not changed during the history of life, so that an identical collection of chemicals reacted exactly the same four billion years ago as today.

[6]Although many mutations are neutral, I contend that for any organism there is a single mutation that will kill it. For example, of the 61 codons, there are 19 which are one mutation away from a codon for STOP. If one of these mutations occurs, it will prematurely end the formation of the corresponding protein, rendering it nonfunctional. If this occurs in the code for a protein that is vital for life, then the offspring organism with this mutation will not be viable.

[7]This follows from proof by confident assertion. Sorry about that. Not only is this density not known, it cannot be known. By the way, any time someone says that something is true without doubt, it is a fair question to ask the method of proof by which all doubt was removed.

like house cats and bobcats, or mice and rats, are certainly closely related. Others like butterflies and pine trees are separated by a vast desert, and that there is a viable connecting path between the two is not obvious.

To quantify this separation, one might try to calculate a "Darwin Number". The Darwin Number, like the Erdös number, specifies the smallest number of mutations that connect two species with viable organisms at each mutational step. However, the Darwin Number is not simply the smallest number of mutations needed to connect two species, but the shortest mutational path along which all organisms are viable. Much of the work in genetics these days to establish phylogenetic trees is based on the thesis of maximal parsimony, namely, that the closest relative is the one with the most similar genome. However, this will not suffice for the Darwin Number, since maximal parsimony does not establish the crucial feature of viability of all intermediates.

So the questions raised by this chapter could be stated as follows:

1. What is the Darwin Number connecting any two currently existing species, and in particular, does it exist?

2. Is Darwin's computer capable of performing this calculation?

First, we must understand what it means that Darwin's algorithm is a random walk. A random walk is a process by which an individual or population moves in random fashion around some space that it occupies. Random walks work just fine if the underlying space is relatively smooth. A blind man can use a random search algorithm to walk up to the top of a hill, by using his stick to determine which way is up or down. Similarly, it is well known that bacteria use random walks to find food. *E. coli* use combinations of swimming and tumbling to randomly change directions to find food sources. Theirs is a directed random walk because they are able to detect gradual improvement and thereby adjust the time they spend swimming rather than tumbling. Lady bugs locate aphids by a random search in which the more sated they are the less they move. Leukocytes find bacterial infection by randomly extending pseudopods, by extending more of them in the direction of bacterial chemical products. In all ot these situation, there is some information about direction that feeds back to the seeker. But no such guidance is available to mutations of the genome.

A scene from the movie UHF provides a good ilustration of how a random search algorithm works. As the camera slowly pans across a small park setting, we hear a voice repeatedly asking, "Is this it?", followed each time by the response "No!". As the camera continues to pan, it picks up two men on a park bench, one of them blind and holding a Rubick's cube. He gives it a twist, holds it up to his friend and the query-response sequence is repeated.

This depiction of a blind search algorithm needs to be modified slightly. With Darwin's algorithm, it is as if the blind man starts with one configuration of Rubic's cube and asks "Is this one better?". If so, then he uses the new configuration as the starting point, but if not, returns to the previous configuration and tries again.

There are two inherent difficulties with this space. A difficulty of crushing proportions is the size of the space about which Darwin's Computer must randomly roam. In fact, this space is so large that it cannot be explored - ever.

To illustrate, suppose we wanted to determine the shortest genome that can sustain a living organism. Unfortunately, there is no efficient way to search for the simplest organism.[8]

Two strategies might be considered. We might start with the simplest known living organism, and make random mutations that shorten its genome, to see if a shorter viable genome can be found. If a shorter one is found, we then try another random mutation to see if it can be shortened again. In this way we follow the mutational paths until a dead end is reached. However, after we have surveyed all of the dead end paths we still do not know if a still shorter viable genome that is not connected by a continuous mutational path might exist. In other words, this algorithm cannot solve the problem.

The only way to be certain that the shortest viable genome has been found is to check all possible configurations of that length and shorter.

Where should we start? The simplest known bacterium is *mycoplasma genitalium*, with 500,000 base pairs in its genome and about 470 genes.[9] A genome with 500,000 base pairs has $4^{500,000} = 10^{300,000}$ possible arrangements. That is a lot of possibilities to check, so maybe we could start by checking strands with a few base pairs and build up from there until we find the shortest viable one. To check all sequences with n or less base pairs requires that we check $4 + 4^2 + 4^3 ... + 4^n = (4^{n+1} - 1)/3$, which is about $(4/3) \times 10^{.6n}$ possible arrangements. We know (actually, we believe, but do not know in the strict sense) ahead of time that most of these are not viable, but we must check all of them to be sure that we haven't missed something.

To check all the genomes of length, say, 10,000, (not even close to 500,000, but we have to start somewhere) and to make the computation go as fast as possible, suppose we take all of the atoms in the universe and turned them into possible genomes of length 10,000. (Recall that is estimated that there are no more than 10^{87} elementary particles in the universe.) If we use each elementary particle as a two-state device, we need two elementary particles to represent the four possible states of a single base pair. We need 20,000 elementary particles to build a "genomic processor" to represent the state of the genome. To build the genome with real organic molecules requires many more elementary particles, since a base pair uses about 70 molecules of carbon, oxygen, hydrogen, nitrogen and phosphate. But ignoring this detail, we can build no more than 10^{83} genomic processors. Now let us suppose that for each of these processors we can make genomes and check them for viability at the rate of the Planck time, the smallest

---

[8]According to the "No free lunch" Theorem, there is no algorithm that is more efficient at optimization of a random function than blind search. See D. H. Wolpert and W. G. McCready, No free lunch theorems for optimization, IEEE Trans. on Evolutionary Computation, 1, 67-82 (1997), and J. C. Culberson, On the futility of blind search: An algorithmic view of "No Free Lunch", Evolutionary Computation, 6, 109-127 (199?)

[9]C. M. Fraser, The minimal gene complement of mycoplasma genitalium, Science, 270, 397-403 (1993).

physically meaningful time in the universe, 10^{-45}, so that each computational genome can check 10^{45} combinations (or chemical arrangements) per second. The fastest modern computers have clock speeds which are significantly slower, at 10^{16} cycles per second, and the shortestlifetime of an elementary particle is 10^{-33} seconds. The universe has existed for 10-20 billion years, no more than 10^{18} seconds. So, in all the time provided by the existence of the universe, with all the elementary particles turned into genomic processors having 10,000 base pairs, each checking one combination per Planck time, the total number of genomes checked cannot exceed 10^{146}. At most one in 10^{5861} of the 10^{6000} possibilities of length 10,000 can be checked for viability. In other words, a finite state machine (our hypothetical collection of genomic processors) cannot determine with certainty that life cannot be sustained with a genome of less than 10,000 base pairs.

Of course, in the history of life on earth, far fewer genomes have been tested. Suppose instead that we use a more realistic test facility and actually test the viability of a genome by making a cell with that genome. So, suppose that the earth were covered one kilometer thick with bacteria that each reproduction cycle created and tested a new genome for viability. The reproductive cycle of a bacterium (the clock speed) is typically between 20 and 60 minutes. In the 4 billion years of life history on earth, that allows a total of 10^{14} cell cycles. If the crust of the earth's surface (5.1×10^8 km^2) were covered 1 km thick with nothing but bacterial cells (with volume about 2×10^{-12} cm^3), there would be about 10^{35} bacterial cells. If with each cycle a different genome were made and checked for viability, then no more than $10^{41} = 4^{81}$ different genomes could be checked in the entire history of life on earth.[10] After four billion years of testing, the best that could be confidently asserted is that genomes of less than 81 base pairs do not support life.

What this means is that, if it is to be explored by *any* finite state computer in our universe, which Darwin's computer is, then the range of possibilities of a genome, or of any modest sized protein, is unknowable.

Problems of this nature are intractable. The solution exists (there is, after all, a shortest genome that can support life), and an algorithm that would solve it is known to exist, but the solution cannot be realized using the computational resources of the known universe. There are true statements about genomes that are not provable using any finite state machine that could be constructed in the universe. There are facts about genomes that can be known only by entities that can perform computational functions beyond that of any finite state machine in the universe and whose access to knowledge therefore transcends the physical universe.[11]

Biology is replete with intractable problems. The difficulty arises from the fact that the dimension of the DNA search space is incomprehensibly large. In our toy problem to find a simple bacterium, the search space was $4^{10,000} =$

[10] Arguments of this sort are used to examine the evolution of computer viruses in M. Ludwig, Computer Viruses, Artificial Life, and Evolution, American Eagle Publications, Tucson, AZ, 1993.

[11] R. Penrose, Shadows of the Mind, Oxford University Press, 1994.

$10^{6,000}$ sequences. But in real life, genomes are much larger. The well-known bacterium *E. coli* has 5×10^6 base pairs and the human genome has 3×10^9 base pairs, making the size of these search spaces incomprehensibly large. The number of possible bacterial genomes is on the order of $10^{840,000}$ sequences, and far fewer than 10^{41} of these possibilities have been tested in the entire history of the earth.

The ability of a random search algorithm to find things depends both on how big the search space is and on how many things there are that can be found. In more prosaic terms, the search for a needle in a haystack depends both on the size of the haystack and on the size and number of needles.

The size of the viable part of the genomic space is simultaneously unknowable and hotly debated. If the density of viable genomes is small, then it is not reasonable to believe that Darwin's algorithm found them. Conversely, those who believe that Darwin's algorithm was responsible for finding all of the life forms that currently exist are forced to conclude that the density of viable genomes is quite large.

To get some idea of how density arguments work, let's consider the question of how hard it would be for a biochemist to synthesize cytochrome c from scratch. Cytochrome c is a simple protein that is found in many organisms and is of fundamental importance in the conversion of glucose to ATP, an important energy transaction vital to cell survival. Wherever it is found, cytochrome c has the same chemical function, although with slightly different chemical arrangements. These arrangements, which are called homologous, are possible because only the active sites are highly specific and much of the remaining structure is composed of amino acid "filler", and is readily replaced by different amino acids. By comparing the structure of a variety of homologous cytochrome c proteins, Yockey estimated that there are about 10^{94} possible ways to arrange the amino acid sequence for cytochrome c.[12] There is, of course, no way to ever check if all of these variations actually function correctly as cytochrome c, or that these are the only possibilities.

Cytochrome c is constructed from 110 amino acids. There are 20 amino acids that can be used at any of the positions, so that there are 10^{143} possible amino acid arrangements with the same length as cytochrome c. Now we can estimate the density of cytochrome c molecules among all amino acid chains of length 110 to be 1 in 10^{49}. Thus, we estimate that the probability of a biochemist synthesizing amino acid chains of length 110 finding a protein that functions like cytochrome c is 1 in 10^{49}.

But the Darwinian algorithm is only required to find connecting, viable paths, from one organism to another. So the proper question to ask is how hard it is to find connecting paths between the needles in our haystack.

This problem has been studied in some detail for one specific transformation in the early history of life[13], that being the transformation from an anaerobic

---

[12] H. P. Yockey, Information Theory and Molecular Biology, Cambridge University Press, 1992.

[13] S. Scherer, Basic functional states in the evolution of light-driven cyclic electron transport, J. Theor. Biol. 104, 289-299 (1983).

heterotrophic microorganism lacking membrane-associated electron transport to a microorganism with a light-driven cyclic electron transport process. This particular system was chosen for study because the electron transport process is a well-studied biological system, and because its development is one of the most fundamental in the history of bioenergetic processes.

In this study, Sherer presents a possible scenario for this transformation in which the new organism is both functional and slightly more efficient at each step, in order to achieve some selective advantage. He proposes five steps between basic functional states and discusses the nature of the mutations required to achieve these transformations. A sample of his analysis for the first step in which an indirect proton transport is formed is summarized as follows: ATP hydrolase (which is assumed to already exist) becomes reversible, permitting ATP synthesis, and fumarate reductase becomes membrane-associated. Fumarate reductase then changes in its specificity, accepting electrons from formate dehydrogenase and not from NAD(H). Gene duplication of a gene coding for a membrane bound protein and creation of a new catalytic site, binding and cleaving formate are necesary to form formate dehydrogenase. Formation of a second new catalytic site at formate hydrogenase, transducing electrons to fumarate reductase, is needed. This first step is estimated to require a minimum of 14 point mutations.

The other four steps are subjected to a similar analysis, and the minimal number of necessary point mutations and gene duplications estimated. From this it is possible to estimate the probability that these basic functional states arose and then to estimate the probability that they can be found using random mutations in 10^9 years in a laboratory the size of the earth. At a mutation rate of 10^{-4} per gene per generation, which is extremely high, he found the probability of step one to be about 10^{-16}. Three of the 4 other steps had probabilities between 10^{-32} and 10^{-64}. Only one step had a probability close to one. However, at the more realistic mutation rate of 10^{-6}, the probabilities for each of the steps ranged between 10^{-20} and 10^{-116}.

The point of this example is to illustrate that pathfinding using the Darwinian algorithm is extremely slow, partly because of the size of the search space and partly because of the narrow specifications on the connecting path.

But if this is not enough of a difficulty for Darwin's Computer, I contend that there is an even greater obstacle for it to overcome, and that is that that life as we know it occupies several isolated mesas and that there is not a smooth genetic connection, with biochemically viable organisms at each step, between all living organisms.

Suppose that possible life forms are segregated onto several isolated biochemical fitness mesas, each separated by a fitness desert. There are only two ways to attempt to cross a fitness desert, with small steps or with big steps. Small steps, corresponding to a few mutations per generation, fail because all offspring stay on the same mesa or land in the desert and cannot survive. Big steps, corresponding to many mutations per generation, would work if an organism could, like Neil Armstrong, take "a giant leap for mankind", or like superman, "leap tall buildings with a single bound". But with random leaping, there must be lots of huge catastrophes. A trial and error method requires (on average) lots of errors, and in the case of crossing a fitness desert, the number of required trials is astronomical. Lucky guesses are not a sufficient explanation.

This leads us to two observations about organisms that permit a large number of mutations per reproduction cycle. First, they necessarily have a large number of genetic stillborn or miscarriages. In fact, this is a necessary feature of an evolving population for which possible changes are more than minor. Said another way, to accomplish a big change with a random walk, a population must take risks, and with risk there will be mostly failure. Evidence of failure is production of a seemingly limitless number of "hopeful monsters", hopeful perhaps, but very dead. Without reproductive infidelity, grand discoveries cannot be made, and for a random process, a grand discovery requires an enormous number of hopeful, but unsuccessful, mutations.

The large number of eggs or seeds produced by an organism that fail to hatch or germinate do not qualify as genetic stillborn, because they are not genetic mutants. Thus, even though the maple tree in my yard produces thousands of seeds and hundreds of seedlings every year, they are all genetically faithful offspring, with no evidence whatever of attempting to move to a different fitness island. [14]

The second problem, however, with genetic infidelity is that it is a harmful inherited trait. An organism that produces failed offspring is at a serious disadvantage compared to one that can reproduce faithfully. If an organism makes enough mistakes in its reproduction process to create hopeful monsters, then it rarely, if ever, produces offspring that are identical to itself.[15] An organism that expends energy to produce many stillborns is certain to be selected against and become extinct, in favor of an organism that reproduces faithfully.

In the words of my friendly car mechanic, "if it ain't broke, don't fix it", and in the world of genetics, if it works, don't change it. The dual goals of survival via faithful reproduction and exploration of the genetic topography are incompatible. Biochemical viability is a valuable commodity, easily lost, hard to find, and therefore must be preserved, if possible. Genetic faithfulness is vastly preferred to unfaithful reproduction. In fact, there are a number of remarkable biochemical repair mechanisms to prevent mutations. Hopeful monsters are selected against, but without hopeful monsters, new fitness islands cannot be found.

Yet a third problem with genetic infidelity is that if a new viable organism is found, and if it survives to reproduce, then its chance of establishing a community of similar organisms is impossibly small. This is not only because it is the sole example of this particular genetic organism, but also because it is

[14] Reproduction is not error-free, but the number of errors is small. In higher organisms the usually quoted mutation rate is 10^{-6}/gene/generation, which for humans with 10^5 genes, is on the order of ten mutations per generation. See J. W. Drake, The Molecular Basis of Mutation, Holden-Day, Inc., San Francisco, Table 5-3.

[15] Here I am assuming that replication errors are produced by some standard random process, such as a Poisson process. If a large number of replication errors is possible for one progeny, then it is likely for all.

the offspring of an unfaithful reproducer, a tendency which it surely inherited. Because it is unlikely to faithfully replicate itself, its continued survival is even more unlikely. The organism will be killed by the very feature that enabled it to find the new mesa in the first place.

This feature that mutation rates cannot be too large is well known in evolutionary theory as a genotypic error threshold. At mutation rates above this threshold, inheritance breaks down since the number of correct copies of even the fittest genotype is steadily decreasing until it is eventually lost.[16]

I conclude that isolated fitness mesas cannot be discovered and populated by the Darwinian algorithm. The search space is far too large to find anything significant by a random walk, and any organism that tries to explore this space is eliminated from the population.

But clearly, this argument has force only if there are, in fact, different isolated fitness mesas that are currently, or have been, populated by life forms.

There are many contenders for isolated fitness islands. Differences between cell types (bacterial vs. animal, plant and fungal), differences in metabolism (plants vs. animals), differences in organization (single cell vs. multicellular), differences in reproduction mechanism (asexual vs. sexual), differences in organ structure (birds vs. fish vs. mammals), are all features that appear to have have vast genetic separation.

There are several features to look for in trying to decide if organisms inhabit separate fitness mesas. One is any function that depends in some crucial way on an integer. The challenge with integers is mathematically obvious. There is no way to gradually change between two integers. But if viability and reproduction depend in some way on those integers, then there is no smooth connecting path (with gradual mutational steps) along which all organisms are viable.

Any complex structure that cannot be built up in an incremental fashion, with functionality at each step, cannot be found by Darwin's Computer. Here we mention some examples, but in the next chapter we describe other examples of "irreducible complexity".

The two obvious contenders for integer relatedness are sexual reproduction and chromosomal numbers. In asexual reproduction, an organism replicates its DNA and the replicated DNA is separated into separate compartments, which are then split apart into two identical copies of the cell. With sexual reproduction, the organism makes a copy of half of its DNA and separates that portion into a special capsule, (an ovum or a sperm cell). Reproduction is completed when a sperm cell contacts an ovum and the two recombine to provide the full genome of the offspring. I suggest that sexual and asexual organisms are separated by a vast biochemical fitness desert; the minimal number of mutations

[16]The threshold mutation rate is that number μ which satisfies $2(1-\mu)^N = 1$, where N is the number of base pairs in the genomic sequence, and μ is the mutation rate per base pair per generation, calculated in T. B. Kepler and A. S. Perelson, Somatic Hypermutation in B cells: An optimal control treatment, preprint, Santa Fe Institute, 1992. See also P. Schuster, Landscapes and molecular evolution, Physica D 107, 351-365 (1997). Mutation rates on the order of 10^{-5} per base pair are fatal to the organism, although high mutation rates of 10−4/gene/generation may have been possible for protobionts with primitive, non-fixed genetic codes, cf. R. W. Kaplan, Der Ursprung des Lebens, Stuttgart, Thieme.

between the two is quite large; there is no gradual genetic transition between the two.

In higher organisms, the genome is separated into smaller subunits called chromosomes. Humans have 46 chromosomes; all other mammals have a different integer even number of chromosomes; gorillas and chimpanzees have 48, mice have 40. Frogs have 26 and peas have 14 chromosomes. In sexual reproduction with chromosomes, the chromosomes are replicated and packaged, half by the male in the sperm cell and half by the female in the ovum. However, the chromosomes are tagged so that after sperm and ovum share their DNA, the chromosomes are appropriately re-paired. This is why it is impossible for organisms with different numbers of chromosomes to form a mating pair and produce offspring; there is always an unmatched chromosome and all the genetic information carried thereby is unavailable so that the offspring is not viable.

It is easy to believe that organisms with different chromosomal numbers reside on separate fitness mesas, since the minimal number of mutations between the two would seem to be quite large. Consider, for example, the problem of one organism with one number of chromosomes establishing a population with a different number of chromosome pairs.[17] At a minimum, there must be a parent organism that mistakenly (and randomly) produces a different number of chromosomes. This new chromosome set must contain meaningful information, and leave the organism viable.[18] Simultaneously in space and time, there must be another parent donor of the opposite sex that also mistakenly produces a modified set of chromosomes. This chromosomal set must also contain meaningful information, and be compatible with (i.e., code for the same genes as) the chromosomal set produced by the partner, so that the offspring is viable, not stillborn. That is, the two sets must be sufficiently similar so that they can pair up properly. Further, there must be yet another happy coincidence that produces a viable offspring of the opposite sex. The two lucky children must meet, mate and be fortunate enough to be genetically compatible.[19] Keep in mind that for every successful formation of a new chromosomal set by an undirected random process, there must be untold billions of genetic stillborns and miscarriages. (A trial and error method requires, on average, lots of errors for each success. Evidence of randomness is a large number of errors.) For organisms with small population numbers and small litter sizes and long reproduction cycles (like mammals), this is not merely improbable, it is impossible.

[17] The discussion that follows does not apply to plants capable of self-fertilization, which changes the reproductive mechanism. In plants, a change of chromosomal numbers (via doubling) can occur in two ways: 1) autopolyploidy, which applies to hermaphrodite species capable of self-fertilization and 2) allopolyploidy, which may occur as the result of hybridization of two slightly different plant species.

[18] There are disorders that yield differing numbers of chromosomes. For example, humans with Down's syndrome have an extra single (unpaired) copy of chromosome 21 resulting from an error during meiosis (cell division) in which one cell received two copies of chromosome 21 and its sister received none. The sister cell without chromosome 21 was not viable and died before fertilization; the Down's syndrome individual cannot reproduce.

[19] Perhaps these could be fraternal twins, but this does not make their individual births and biochemical viabilities any more likely.

On this basis, it appears that mammals with different chromosomal numbers occupy different fitness mesas.

The preceding arguments lead me to conclude that the Darwinian algorithm could not have found all of the life forms that currently exist. The genomic space is too large, the Darwinian computation far too slow, the distances to travel between organisms are far too long and constrained, and there appear to be complex structures and organisms that occupy separate fitness mesas that are therefore not explained by the Darwinian mechanism. Darwin's algorithm by itself did not get us where we are today.

4.0.1 Appendix: Turing's Proof of Gödel's Theorem.

This discussion is about numerical algorithms, that is, specific sets of instructions that are operations on numbers. An algorithm starts with a number on input and carries out a calculation. Some algoritms terminate at completion, but others never terminate. We can make a list of all possible algorithms that start with the integer n, as $C_1(n), C_2(n), C_3(n), C_4(n), \ldots$ etc.

Now suppose that there is a test (a numerical algorithm) to determine if an algorithm terminates. We denote this test by $T(p, n)$ to indicate that it tests to see if the pth algorithm acting on the integer n, $C_p(n)$, terminates or not. If $T(p, n)$ terminates then $C_p(n)$ will not terminate.

Now a trick. We set $p = n$, so that $T(n, n)$ is an algorithm acting on the integer n. This means that $T(n, n)$ is somewhere in our list of algorithms, $T(n, n) = C_k(n)$ for some value of k. It follows that $T(k, k) = C_k(k)$. But this is a contradiction, since $T(k, k)$ cannot both terminate and not terminate.

Apparently the test T does not exist. This means that there are some algorithms for which it cannot be decided if they will terminate.