Entropy **2009**, *11*, 85-110; doi:10.3390/e11010085

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Review

The Insights of Algorithmic Entropy

Sean Devine

Victoria Management School, Victoria University, PO Box 600, Wellington, New Zealand;
E-mail: sean.devine@vuw.ac.nz

Received: 19 November 2008 / Accepted: 20 February 2009 / Published: 4 March 2009

Abstract: The algorithmic entropy of a system, the length of the shortest algorithm that specifies the system's exact state adds some missing pieces to the entropy jigsaw. Because the approach embodies the traditional entropies as a special case, problematic issues such as the coarse graining framework of the Gibbs' entropy manifest themselves in a different and more manageable form, appearing as the description of the system and the choice of the universal computing machine. The provisional algorithmic entropy combines the best information about the state of the system together with any underlying uncertainty; the latter represents the Shannon entropy. The algorithmic approach also specifies structure that the traditional entropies take as given. Furthermore, algorithmic entropy provides insights into how a system can maintain itself off equilibrium, leading to Ashby's law of requisite variety. This review shows how the algorithmic approach can provide insights into real world systems, by outlining recent work on how replicating structures that generate order can evolve to maintain a system far from equilibrium.

Keywords: Algorithmic entropy; Non equilibrium thermodynamics; Complex systems; Requisite variety.

Classification: PACS 89.70Cf; 05.70.Ln; 89.75.Fb

1. Introduction

The traditional understandings of entropy include the thermodynamic approach of Clausius, the statistical mechanics approaches of Boltzmann and Gibbs and the information theory approach of Shannon. As these different approaches refer to the same quantity, with some provisos (allowing for the units used and recognising that the Boltzmann entropy is a special case of the Gibbs entropy), one might well ask what are the advantages in developing another entropy based on Algorithmic Information Theory. This paper argues that algorithmic entropy, based on Algorithmic Information Theory (AIT) adds a piece to the entropy jigsaw puzzle; providing insights into the issues surrounding the Gibbs' entropy and the second law of thermodynamics (section 2.6.); providing a measure for an individual configuration that is consistent whether it is an equilibrium or non equilibrium state; and finally consistently tracking entropy flows between a system and its environment in a way that clarifies the role of Maxwell's demon (section 4. and references therein).

While the basic idea behind AIT was first articulated in embryonic form by Solomonoff [1], the approach has been put on a robust footing by Kolmogorov [2] and independently by Chaitin [3]. The extension to prefix coding by Levin [4], Gács [5], and later Chaitin [6], provided the tools to better align the approach with Shannon's information theory. The length in bits of the shortest computer programme or algorithm that generates the string defining the particular configuration of a system represents the system's algorithmic entropy or its information content. This length is known as the "Kolmogorov complexity", the "algorithmic complexity" or the "program-size complexity". In effect, those structures or configurations that show order or pattern will be described by much shorter algorithms than those that are random. For example, Ratsaby [7] argues that the algorithmic complexity (*i.e.* the algorithmic entropy) of a static structure is a measure of its order. The description of the momentum and position of the molecules in a container of water at instant of time are random, but the algorithm that specifies the position and momentum of the particles of frozen water is much shorter, as ice shows an ordered structure.

While the length of the description will depend on the computational method used to define the structure, as is argued later in section 2.1., this entropy measure is not particularly dependent on the choice of the reference abstract machine. Thus the algorithmic entropy provides a useful measure of the entropy of a physical situation. However, in contrast to the traditional entropies, the algorithmic entropy measure applies to a particular configuration of a system. Nevertheless, once allowance is made for the units used, the expectation value of the algorithmic entropy for a set of strings belonging to an ensemble is virtually the same as the Shannon entropy, or the Boltzmann or Gibbs entropies derived for a set of states. Where the statistical physics entropies might use the phrase "equilibrium configuration", the algorithmic approach would use the phrase "typical string". An atypical string would have a very low probability of representing an actual state in an equilibrium configuration.

Two subsequent developments have helped to align the algorithmic approach with the traditional approaches.

- As the algorithmic description is an exact description of a system, the underlying determinants of the system must in principle be specified. In the traditional approaches to entropy these are taken as given. Nevertheless, as entropy is a state function, any information common to the states of a system does not affect the entropy differences between the states. For example, the algorithmic description of the position and momentum coordinates of the planets in the solar system at an instant of time can take the physical laws as given, as the algorithmic description of the physical laws can be embodied in subroutines that are common to all configurations. From a physical point of view, the useful algorithmic entropy measure can ignore these long and usually unspecifiable strings that make no difference in comparing configurations.

- Until recently the algorithmic entropy concept seemed limited to ideal situations as the algorithm must provide an exact description of the string at an instant of time; the entropy would be meaningless where random variations dominated the pattern. This problem was resolved by recognising (see section 2.4.) that the best algorithmic description of a patterned, but noisy, string involves combining the algorithm that identifies the particular string in the set of patterned strings with the algorithm that specifies the structure of the pattern or model that defines the set. Rather than being a problem, this resolution reinforces the connection between algorithmic entropy and the traditional entropies in a way that allows one to slip from one approach to another. When no pattern is recognised, the algorithmic entropy of a string is virtually the Shannon entropy, whereas with further information, if a pattern is recognised, the description can be shortened as is outlined in section 3..

As is discussed in more detail in section 2.6., the algorithmic approach provides a methodology to deal with the issues such as phase space resolution and course graining that arise with the traditional statistical thermodynamic approach to entropy. Also, for the interested reader Algorithmic Information Theory is closely related to the Minimum Description Length in its ideal form [8] or its stochastic form [9–11] and can be used to justify the maximum entropy approach of Jaynes [12, 13].

While Algorithmic Information Theory provides insights into the idea of entropy, the technique has not been readily accessible to the scientific community as most of the development has taken place in mathematics with a focus on randomness, Kolmogorov probability, inference and non computability. The hope is that this article will allow interested researchers to get a sufficient understanding of the issues underpinning the approach, that they can pursue the significant mathematical developments when appropriate.

2. Algorithmic Information Theory

The idea behind Algorithmic Information Theory was first articulated by Solomonoff in embryonic form [1]. Solomonoff recognised that it was much simpler to send a coded message of structured information such as the first 100 digits of π by sending the algorithm that generates π, rather than sending the actual string of digits. On the other hand, as a random string cannot be coded simply, the complete string must be transmitted. Later Kolmogorov [2] with the insights of Levin [14] and independently of both, Chaitin [3] developed Algorithmic Information Theory (AIT) to provide a measure of the complexity, or the algorithmic entropy of a system. As a system can be represented by a string s, the complexity or algorithmic entropy of the system is defined as the length of the shortest algorithm p^* that is able to generate string s. The length of this algorithm, which is usually in binary form, is denoted by $|p^*|$, where the vertical lines denote the number of characters in the enclosed algorithmic description. In mathematical terms, systems and their string representations that show no pattern or order are the most complex - they appear random and, because their description cannot be compressed, the algorithmic entropy is maximum. On the other hand ordered strings have low algorithmic entropy or, in mathematical terms, low complexity and can be described by a shorter algorithm. As scientists use the word "complex system" in a different sense to mean a system that is not random, but is highly complicated, the phrase "algorithmic entropy" rather than "algorithmic complexity" is the preferred term in this paper (see section 2.5.).

While in principle, any structure can be described by a string and its entropy measured by $|p^*|$, p^* is non computable. As a consequence of Turing's halting theorem there is no certainty that a particular description is the shortest possible as the shortest description cannot be effectively computed. Nevertheless, where symmetry is observed, or models have been discovered that explain much of the data, some order clearly exists and the data can be compressed into a shorter algorithm. Once pattern or structure is identified, by whatever means, compression is possible. The obviously patterned string $s = 111\ldots1$ with N repeated 1's illustrates this. It can be represented by the following algorithm or programme, p'.

$$p' = PRINT \text{ “1” } N \, times. \tag{1}$$

In general, if a binary algorithm for p' generates s on a computer C, the length of the shortest possible algorithm $|p^*|$ must be $\leq |p'|$; *i.e.*

$$\begin{aligned} |p^*| \leq |p'| = \;& |N| + the\, size\, of\, the\, code\, for\, the\, PRINT \\ & \quad instruction \\ + \;& the\, number\, of\, bits\, to\, specify\, the\, object, \end{aligned} \tag{2}$$

where $|N|$ represents the length of the description of N. The length of the above algorithm in binary notation is much shorter than the original string and is therefore a compressed or coded form of the string. Even though the above programme may not be the shortest possible, for large N the length of the shortest description will be dominated by the $|N|$ which is $\log_2 N$ rounded up, usually denoted by $\lceil \log_2 N \rceil$. In this case,

$$|p^*| \approx |p'| = \lceil \log_2 N \rceil + |1| + O(1), \tag{3}$$

In the literature, $\lceil \log_2 N \rceil$ is often represented by $\log_2 N$ for large N. The $O(1)$, or order 1 term, is independent of N and represents the string defining the “PRINT” instruction.

On the other hand a random string of the same length will show no pattern. A representative random string might be something like $10111\ldots01011\ldots101$. Such a string, if random, cannot be represented by a simple algorithm. In which case the algorithm that generates the string must itself specify the string exactly and the shortest description $|p^*|$ is given by

$$\begin{aligned} |p*| \quad & = |10111\ldots01011\ldots101| \\ & +size\ \ of\ \ print\ \ instructions\ \ etc. \end{aligned} \tag{4}$$

I.e. as strings with no pattern cannot be compressed, their information content, or algorithmic entropy must be slightly greater than the length of the string itself.

In AIT, the word “code” can refer to each binary instruction that makes up the algorithm. It also can refer to a compressed version of a complete string. For example the string $s_i = 1y1y1y1y1y$, where y is a 0 or a 1, can be coded by replacing each 10 by a 0 and each 11 by a 1. If the original string has length N, the compressed string has length $N/2$. However, the coded string itself is not an algorithmic measure. In order to be an actual algorithm, the code must be given together with the decoding process that replaces each '1' by a '11' and each '0' by a '10'. The algorithmic entropy of the string is then made up of

$$\begin{aligned} |p*| &\leq |algorithmic\, code\, for\, s_i| \\ &+|decoding\, routine| \end{aligned} \tag{5}$$

Here again, the vertical lines denote the length of the instruction string enclosed by the lines.

2.1. Self-delimiting coding and the Invariance Theorem

If the algorithmic entropy of a string is to be defined in terms of the size $|p*|$ of the minimal programme that generates the string, and if the definition is to be aligned with the definition of Shannon entropy, it is necessary to ensure that the instructions and numbers in binary form, are encoded in a way that it is clear where one instruction finishes and the other starts. In such a formalism, no codeword is a prefix of any other. This can be achieved by using self-delimiting coding, codes that can be read instantaneously [5, 6, 14] (see also [15]). (Note, as all such codewords must come from a prefix-free set, these codes are sometimes called prefix codes which is somewhat confusing.)

The self-delimiting requirement adds about $\log_2 N$ to the specification of a string of length N. If there are n words to be coded, and if $|x_i|$ is the length of the code of the ith word x_i, then the code lengths satisfy the Kraft inequality; *i.e.*

$$\sum_i^n 2^{-|x_i|} \leq 1. \tag{6}$$

Conversely, given a set of codeword lengths that satisfy this inequality, there exists a self-delimiting code with these word lengths. This inequality can be seen if one generates the tree of all possible binary codewords. Whenever a codeword such as 1011 is assigned to a word, no code starting with 1011 is legitimate so that the code tree terminates at that point. Similarly 101 cannot be a code as any branch of the coding tree starting with 101 would already have terminated. A simple proof is to be found in Calude [16]. Chaitin extended the Kraft inequality [6] to infinite, recursively enumerable sets of source words, not just finite codes.

Where the computation only accepts self-delimiting instructions, the program-size complexity of string s is often denoted by $K(s)$ or, where the input string i is given, $K(s|i)$. However, in this paper, the algorithmic entropy, which denotes the information content of the string, denoted by $H_{algo}(s)$ is the preferred term. *i.e.* $H_{algo}(s) = K(s)$. The conditional algorithmic entropy, given input string i, is the extra information required for the shortest algorithm on the reference UTM able to produce the output string s; *i.e.* the length of the shortest programme p^* such that $U(p^*, i) = s$. The conditional algorithmic entropy is denoted by $H_{algo}(s|i)$. Here, the subscript "algo" distinguishes the algorithmic entropy measure from the Shannon entropy H_s.

Chaitin [6] has shown that the Kraft inequality holds if the algorithmic entropy or information content of the string s_i is defined as the length of the algorithm that generates s_i on a reference Universal Turing Machine (UTM) that only accepts inputs from a prefix-free, or self-delimiting, set of codes [16]. As such a reference UTM machine can simulate any other Turing machine [6, 15, 16] the computer dependence of the algorithmic entropy can be mostly eliminated. Denoting $U(p)$ as the computation using programme p on UTM U;

$$H_U(s_i) = minimum\ |p|\ such\ that\ U(p) = s_i$$

While the definition is somewhat dependent on the UTM used, the use of an alternative UTM will add an $O(1)$ constant, corresponding to the size of the algorithm that will simulate one UTM on another. This leads to the following relationship known as the Invariance Theorem;

$$H_{algo}(s) \leq H_U(s) + O(1). \tag{7}$$

While there are an infinite number of possible UTMs, by choosing the reference UTM to be one with an appropriately chosen instruction set, the O(1) term can be made small relative to the algorithmic entropy of a string that, for example, describes the microstate of a thermodynamic system. Marvin Minzky has described a 7-state, 4-symbol UTM [17] and the 2-state 3 colour Turing Machine of Stephen Wolfram has recently been shown to be universal by Alex Smith [18]. In practice, as Li and Vitányi (page 199) [15] using combinatory logic and Chaitin [19] using LISP have shown the $O(1)$ term need only be a few hundred bits. Recently Tromp [20] has outlined two simple computational devices; one based on a 210 bit lambda calculus self-interpreter, and the other a 272 bit binary combinatory logic self-interpreter. Tromp has given an example where the O(1) term is 130 bits. Nevertheless, as entropy difference is usually the critical measure, machine dependence becomes irrelevant.

2.2. *Efficient coding*

If one wishes to code a message efficiently, it makes sense to use the shortest codes for the message symbols that occur most frequently. Shannon's noiseless coding theorem shows how this can be done. Consider a finite set of message symbols s_1, s_2, s_3, s_4, s_5,. . s_k that occur in the expected message with probabilities p_1, p_2, p_3, p_4, p_5, ... p_k. Let the self-delimiting binary code words for these be $\hat{s}_1$, $\hat{s}_2$, $\hat{s}_3$,$\hat{s}_4$, $\hat{s}_5$, ...$\hat{s}_k$. Shannon - Fano coding is an efficient coding methodology that satisfies the Kraft inequality with the length of the code words constrained by:

$$-\log_2 p_k \leq |\hat{s}_k| \leq 1 - \log_2 p_k. \tag{8}$$

This implies that $2^{-|\hat{s}_k|} \leq p_k \leq 2^{1-|\hat{s}_k|}$. The Shannon Fano coding can be implemented by ordering the message symbols from the most probable to the least probable using the binary tree expansion to assign the most probable symbols to the shortest code lengths in the tree. As the codes are to be prefix-free, once a code has been assigned, codes belonging to that particular branch are no longer available.

An alternative, Huffman coding, combines probabilities to form even more efficient coding, while an arithmetic code [21] is slightly better. Hence, given the probabilities of symbols in the message, the average length of coding can be made very close to the Shannon entropy, $H_s = -\Sigma_k p_k \log_2 p_k$, of the source; *i.e.* the entropy based on the expected occurrence of symbols.

It follows from Equation 8 above that, the expected code length per symbol (*i.e.* $\sum p_k|\hat{s}_k|$,) in a message is given by

$$H_s \leq \sum p_k|\hat{s}_k| \leq H_s + 1 \tag{9}$$

As a consequence, the expected length of a message consisting of N symbols can be made close NH_s. This equation is known as Shannon's noiseless coding theorem. A code where the expected code length per symbol equals the Shannon entropy is optimal for the given probability distribution in that it cannot be bettered.

However, there is an ambiguity in coding a natural number n in binary form as "01" is the same as "1". Lexicographic ordering (E.g. the integers from 0 to 6 etc are coded as $\oslash$, 0, 1, 00, 01, 10,11, etc, where $\oslash$ is the empty string.) can be used to avoid this ambiguity as the length of n's description becomes $\lfloor \log_2(n+1) \rfloor$. Here the floor notation denotes the greatest integer $\leq$ to the enclosed term. This integer specifying the length of string n will be represented by $l(n)$, which is no more than one bit less than $\lceil \log_2 n \rceil$, where the ceiling notation means rounding up.

However where n comes from an unknown probability distribution, and where a self-delimiting code is needed for n, the length of the code must be explicitly included within the code [15]. One possibility is to use $code(n) = 1^{l(n)}0n$, which has an overall length of $2l(n)+1$. While there is no simple expression for the most compressed code, more sophisticated coding procedures can produce a shorter self-delimiting description of n by including the length of the code for n or, better still, the length of the length of the code for n in the description. In practice $|code(n)| \leq l(n) + 2l(l(n))$ with two iterations and $|code(n)| \leq l(n) + l(l(n)) + 2l(l(l(n)))$ with three iterations and so on. Tromp [20] has a different process for generating a self delimiting code but with the same result.

A common application is where all members of a finite set of N strings occur with probability $1/N$. Shannon's coding theorem shows that any string x_i in the set can be represented by a code that is self-delimiting having length $\lceil \log_2 N \rceil$. In what follows, e.g. in the discussion on provisional entropy, as is customary, $\log_2 N$ will be used to represent this integer. (One could choose a logarithmic base that ensured $\log_Q N$ was an integer, and the result converted to base 2. In which case, $\log_2 N$ would be the converted value.) In general, the decoding routine associated with the code word for a string x_i must contain information about the length of $code(x_i)$ to know when the code ends. I.e. the routine must read each character in turn and decide whether it has read sufficient characters or not. As the overall routine will include a term specifying the length of $code(x_i)$, the contribution to the entropy will include not just $|code(x_i)|$ but also $||code(x_i)||$.

2.3. *Entropy relative to the common framework*

Where there are two strings x and y, and the information about one string can be used as an input to calculate the other, there are three different situations.

- In general $H_{algo}(x) \leq H_{algo}(y) + H_{algo}(x|y) + O(1)$. If there is mutual information between the strings, knowing y reduces $H_{algo}(x|y)$ and if y actually contains x, the conditional terms is zero.

- The entropy measure based on the algorithm that computes both x and y is given by $H_{algo}(x, y) = H_{algo}(y) + H_{algo}(x|y, H_{algo}(y)) + O(1) = H_{algo}(y) + H_{algo}(x|y^*) + O(1)$. Gács [5] and later Chaitin [6] have shown that either y^* the compressed version of y, or both y and $H_{algo}(y)$ are needed, otherwise small logarithmic corrections are needed to make the equation consistent.

- In the particular case where y is the optimally compressed subroutine needed in the calculation of

x, then y =y^*, and the two routines can be concatenated (joined) to generate x. The algorithmic entropy defined by combining two compressed routines is $H_{algo}(x) = H_{algo}(y) + H_{algo}(x|y) + O(1)$. Here the $O(1)$ refers to a simple instruction to link the subroutine to the main programme [22, 23]. This allows one to consistently nest subroutines within subroutines in a way that allows models or physical laws to be nested within higher level models or laws.

As entropy is a state function only differences in entropy are important. Subroutine strings such as "$PRINT$" and "$FOR/NEXT$" etc., and the $O(1)$ uncertainty related to the reference UTM that are common to different states can be taken as given. Similarly, the complete description of a physical system may require the physical laws that determine the particular state of the system to be specified, together with the description of the system such as the specification of the coarse graining regime [24–26]. Common subroutines that specify the relevant physical laws or describe the physical system, can be taken as given. This makes the algorithmic approach tractable for many actual situations.

Let the common instruction string be represented by 'CI' [27] and, given the common instructions or common subroutines, the physically significant entropy will be denoted by the conditional algorithmic entropy $H_{algo}(x|CI)$. In what follows unless specifically stated otherwise, $H_{algo}(x)$ can be used to represent $H_{algo}(x|CI)$. I.e. the zero of entropy is chosen by subtracting $H_{algo}(CI)$, from the full entropy.

2.4. *Provisional entropy*

Until recently AIT seemed restricted, as the algorithm must describe the string exactly; including both the structure and any noise or randomness. As most real world configurations, even if ordered, show noise and variation, it was thought that the random components would dominate the length of the generating algorithm obscuring any pattern [28].

Devine [27] has shown that where pattern in a noisy sequence is recognized there is an implicit reference to a set containing all similar strings exhibiting the pattern. As this set will be recursively enumerable (*i.e.* a procedure exists to generate the set), a noisy string that exhibits this pattern can be generated by an algorithm that consists of two routines; one that enumerates the patterned set of strings S containing the string of interest, and one that specifies which particular string in the set is the string of interest. The algorithmic entropy of the string x_i in the set is $H_{algo}(S) + H_{algo}(x_i|S)$; the first term is the length of the routine that defines the set consistent with the pattern or model and the second is the length of the routine that identifies the particular string within the set. Let there be N_S equally probable members of the set S. As was shown in the previous section, a particular member can be identified by an algorithmic code of length $H_{algo}(x_i|S) = \log_2 N_S$. However as there is always the possibility that more structure can be recognised the algorithmic description above is only a provisional entropy measure and will be denoted by $H_{prov}(x_i)$. It is the best estimate of the entropy of a particular noisy patterned string representing the state of a physical system. Devine [27] used the phrase "revealed entropy" instead of "provisional entropy" to indicate the value depended on the observed or revealed pattern.

The provisional entropy is derived by combining the length of these two routines [27]. Hence

$$H_{prov}(x_i) = H_{algo}(S) + \log_2 N_S + O(1).$$

Or, in effect;

$$H_{prov}(x_i) \quad = H_{algo}(description\ of\ the\ set's\ structure)$$

$$+H_{algo}(identifying\,the\,string\,in\,the\,set). \tag{10}$$

The second contribution is equivalent to the Shannon entropy of the set. In other words the provisional algorithmic entropy depends on the uncertainty in defining the member of the set together with the algorithmic description of the set's pattern. The provisional entropy is the entropy of a typical member of the patterned set. It is the upper measure of the algorithmic entropy for that member and is the best estimate of the length of the minimum description of the physical system given the available information. However a very few atypical members of the set may be further compressed, or some unidentified pattern may exist. For example if y is taken to represent 0 or 1 selected on a random basis, the provisional algorithmic entropy of the noisy period two string $s = 1y1y1y1\ldots 1y1y$ is (see Devine [27])

$$H_{prov}(s) \cong N/2 + \log_2(N/2) + |1| + |0|.$$

Nevertheless the string $s' = 11111\ldots 11$ from the same set has the much lower entropy; $H_{algo}(s') \cong \log_2 N + |1|$. I.e. $H_{algo}(s') \leq H_{prov}(s)$. Whenever the description of a member of a set can be compressed further, a more refined model is needed to capture the pattern in those particular strings.

Kolmogorov introduced the algorithmic equivalent of the minimum sufficient statistic concept known as the Kolmogorov Minimum Sufficient Statistic, or the Algorithmic Minimum Sufficient Statistic (AMSS). This provides the basis of a methodology that, when applied to strings with a recognised pattern, gives an identical result to that above. The approach is outlined in the earlier work of Gács *et al.*, and Vereshchagin and Vitányi [29]. A string showing pattern or structure is a member of a finite set of similar strings and can be specified by the algorithm that generates the set of strings, coupled with an algorithm that specifies the particular string in the set. However this description may not be the shortest. The shortest description involves finding the optimum set where the string is a typical member; *i.e.* relative to other members in the set it is random. In which case all the structure embodied in the string is captured by the algorithm that specifies the set. The result is the same as Equation 10 as the two part algorithm first specifies the set, and then the string within the set by $\log_2 N_S$. It is worth noting that the ideal version of the Minimum Description Length [8] gives a similar result provided a short decoding routine is appended to the minimum description.

2.5. Relationship with logical depth

As has been mentioned scientists, in contrast to mathematicians, use the word "complexity" to characterise a sophisticated structure. Bennett [23, 30] developed the concept of ”logical depth” to provide a basis for this intuitive idea. Logical depth is a measure of how difficult it is, in terms of time, for a universal computer to generate a particular outcome; in effect it is a measure of the effort to derive the outcome from a hypothetical cause. However such a measure cannot just be the time (as measured by the number of cycles) taken for the shortest programme to generate the output, as a slightly longer programme might be significantly more efficient than the shortest one. Bennett [23, 30] (see also reference [15] page 510) allows for these more efficient programmes in terms of a significance level s. The depth of a string representing an object at significance level s is the least time to compute the string by a programme that is compressible by no more than s bits. Particularly useful is the logical depth relative to the length of the string itself as the input to the computation. A random string has minimal logical depth as the computation must specify each character. On the other hand a string of repeated ones has a short computation time and therefore shows little depth. Readers should not that logical depth does not compete with algorithmic entropy as a measure of order as it is a measure of the process, not the outcome. It addresses a different issue, as the most interesting biological and physical structures appear to be characterised by programmes that are logically deep; *i.e.* the algorithmic description of the structures require significant computational processing time. Like program-size complexity the result is not particularly machine dependent.

2.6. Relationship with other entropies

The equilibrium approach of statistical thermodynamics relates the entropy to the logarithm of the number of states in the system. In practice, this is taken to be the logarithm of the number of equilibrium states as these overwhelm the statistical entropy measure. When no pattern is recognised, the algorithmic entropy of a string is virtually $\log_2 N_S$, *i.e.* the Shannon entropy, rounded up to the nearest integer. Whereas if a pattern is recognised or discovered as information is gathered, the description may be shortened as is outlined in section 2.4.. The approach is effective for any string that exhibits a noisy pattern reflecting data derived from a model. While the algorithmic approach requires the underlying structure and physical situation to be defined, including the physical laws, the graining structure, and the computational system, the Shannon entropy only defines the remaining uncertainty. In the algorithmic case this remaining uncertainty measure is captured by a term that corresponds to the length of the algorithm required to identify a particular string in the set of all strings. The resolution of the problem of noisy strings reinforces the connection between the algorithmic entropy and the traditional entropies in a way that allows one to slip from one approach to another.

However the algorithmic measure of entropy, is conceptually different from the Shannon entropy as it returns a value for the entropy of the actual state of a physical system at an instant of time, and has meaning for a non equilibrium configuration. On the other hand, the traditional entropies return a value for a set of states which, in the thermodynamic case, is the set dominated by the equilibrium states. Nevertheless, allowing for the differences in the units (bits or $k_B ln2$ where k_B is Boltzmann's constant), the expectation value of the algorithmic entropy, $\langle H_{algo} \rangle$, for a typical or equilibrium state is asymptotically identical to the traditional entropies, once allowance is made for differences of $O(1)$ [24, 31] due to the computational overheads. This is because an equilibrium state is a typical state and cannot be compressed to less than the $\log_2 W$ where W is the number of possible equilibrium states. One can say that the Shannon, or (allowing for units) the thermodynamic entropy, provide the most likely value of the algorithmic entropy of an individual state in the set of states.

Consider a classical physical system, initially isolated from the rest of the universe where the instantaneous microstate of the system is represented by a point in the multidimensional state space. Over time the state point of the system will move through state space under the operation of classical physical laws. As is discussed in section 3., the instantaneous microstate of the system can be represented by a binary string (e.g. Zurek [24] and Li and Vitányi [15]). The size of the most concise algorithmic description of the configuration's microstate gives the algorithmic entropy.

Just as in the statistical mechanics description of a thermodynamic system such as a gas, the algorithmic entropy measure depends on the resolution chosen to specify a microstate of the system. In both cases, the quantum limit provides the ultimate possible resolution. However as was outlined in section 2.3., the algorithmic approach can take the physical situation, such as the resolution of the phase space, as given when comparing the entropy of different states. Furthermore, Zurek has shown how the algorithmic approach can resolve the difficulties with the coarse graining issue that arises from the Gibbs approach. In the algorithmic version, the coarse graining must be specified in principle as part of the description of the system. The requirement that the description must be minimal constrains the allowable coarse grain shape for a given Universal Turing computer that implements the algorithm. Zurek points out, the subjectivity associated with coarse graining emerges in a different form; namely the definition of the given universal computer. In this case the problem is easier to deal with conceptually as simple UTMs provide similar algorithmic descriptions of the physical situation and any measure on a more sophisticated Universal Machine can always be translated into a measure on a simple one.

Again, Gács [26] (see Li and Vitányi [15]), by using a different approach, shows that an algorithmic entropy can be defined that converges to a specific value as the resolution of the system implied by the grain size increases.

2.7. *The algorithmic equivalent of the noiseless coding theorem*

Let the algorithmic halting probability $Q_U(x)$ be the probability that a randomly generated programme p will halt on the reference UTM with x as its output. I.e. all programmes generated by the toss of a coin that halt giving x as an output are included in $Q_U(x)$. Thus $Q_U(x)$ is defined by

$$Q_U(x) = \sum_{U(p)=x} 2^{-|p|}. \tag{11}$$

From the Kraft inequality, the sum of all such probabilities cannot be greater than one. Chaitin [6] defines this sum as $\Omega = \sum_x Q_U(x) \leq 1$. As the sum Ω may not reach 1, $Q_U(x)$ is not a true probability. In measure theory the term used for a defective probability of this nature is semi measure. Because it is not a true probability Q rather than P is used here to denote this measure. Furthermore, there is no gain in normalising $Q_U(x)$ by dividing by Ω as there is no halting computational procedure to evaluate Ω. The value of Ω is highly machine dependent and it can be proven in some cases that not even 1 bit of Ω can be computed [32]. Calude and Dineen [33] outline the issues and, using a specific UTM, calculate the first 43 bits of Ω in base 16 and the first 40 bits in base 2. Despite not being a true probability, the halting probability measures relative probabilities and furthermore there is a very important connection with the algorithmic entropy of a string. The argument is as follows.

- As $Q_U(x)$ must contain the programme that specifies $H_U(x)$ on the reference UTM as outlined in section 2.1., $Q_U(x) \geq 2^{-H_U(x)}$ or,

$$H_U(x) \geq -\log_2 Q_U(x) \tag{12}$$

- But, $H_U(x)$ cannot be much greater than $-\log_2 Q_U(x)$ as there are few short programmes able to generate x. It can he shown that all longer descriptions contribute little to the sum [6] so that

$$H_U(x) \leq \lceil -\log_2 Q_U(x) \rceil + c \tag{13}$$

leading to

$$H_U(x) = -\log_2 Q_U(x) + c'. \tag{14}$$

The halting probability, and the algorithmic entropy of string x can be related to a semi measure $m(x)$ whose negative logarithm is defined to be a universal code for x; one that is asymptotically optimal in coding a message independently of the distribution of the source words. For a discrete sample space, there exists such an optimal universal enumerable semi measure. It is one that multiplicatively dominates all other constructive semi measures; *i.e.* all semi measures that can be computed from below [34]. This leads to the algorithmic equivalent of the Shannon's noiseless coding theorem outlined in section 2.2., but in this case the distribution is not the distribution of source words, but the halting probability of the source word as the input to the UTM. In other words, if $m(x)$ is taken to be the universal enumerable semi measure then $m(x) \geq c\mu(x)$ where $\mu(x)$ is any other semi measure. The significance of this will become apparent, but firstly it should be noted that the universal property only holds for enumerable semi measures (see Li and Vitányi page 246 [15]).

It can be shown the $Q(x)$ is a universal semi measure as the set of all random inputs giving rise to $Q(x)$ will include the simulation of every Turing Machine and every programme p that runs on each Turing machine T_i. Because $Q(x)$ includes every semi measure, it multiplicatively dominates them all. However, $2^{-H(x)}$ is also a constructive semi measure, and because $H(x)$ is equal to $-\log_2 Q(x)$ to within a multiplicative constant, $2^{-H(x)}$ can also be taken to be a universal semi measure. In other words, as universal semi measures, $m(x)$, $Q(x)$ and $2^{-H(x)}$ are all equal to within a multiplicative constant.

This gives rise to the algorithmic equivalent of the coding theorem [4]. The following equality holds to within an additive constant or an O(1) term.

$$H(x) = -\log_2 m(x) = -\log_2 Q(x)$$

This allows $2^{-H(x)}$ to be taken to be the universal semi measure $m(x)$. This semi measure is a measure of maximum ignorance about a situation as it assigns maximal probability to all objects (as it dominates other distributions up to a multiplicative constant) and is in effect a universal distribution. As a universal distribution, the universal semi measure provides insights into defining a typical member of a set and provides a basis for induction based on Bayesian principles.

3. Information uncertainty and the algorithmic description

The microstate of a Boltzmann gas of N particles in a container can be specified by a string representing the exact configuration of the system in terms of the coordinates of each particle in the 6N dimensional phase space known as Γ space. The 6 dimensional position and momentum coordinates of each particle are specified in binary form to a required degree of precision. Generally a macrostate of the system will embody many microstates. However in principle, with sufficient precision, this process can impose a fine grained cellular structure in the phase space that can distinguish between the different microstates of the system. As the string is binary, doubling the resolution of each axis of the Γ space is equivalent to adding an extra significant figure to the representation of each position and momentum value. An alternative description outlined by Zurek [24] defines the resolution and structure of the Γ space first, and then specifies the configuration by a sequence of 0's and 1's, where a 0 represents a cell in Γ space that is empty and a 1 a cell that is occupied; the size of the cell depending on the required resolution. These two specifications can be interchanged using a simple algorithm. However, the Zurek approach has advantages that, where the phase space resolution or structure can be taken as given and assumed to be an optimum description, the contribution can be ignored as part of the common information. The algorithmic entropy of the configuration is then the shortest programme that generates the sequence of 0's and 1's.

The algorithmic description of an equilibrium state will be no shorter than $\log_2 W$ where W is the number of available states; *i.e.* the algorithmic entropy coincides with the Shannon entropy. However, where a measurement shows that the microstate is partially ordered, the algorithmic entropy will be lower than the Shannon entropy; as would be the case if all particles were moving in the same direction. No amount of extra information can lower the algorithmic entropy of an equilibrium configuration, but where measurements narrow the configuration of a microstate to an ordered subset, the algorithmic description can be compressed.

This insight prompted Zurek [24] to define the 'Physical Entropy' $\mathcal{S}_d$ of a microstate. This is the sum of (a) the most concise but partial algorithmic description based on the available information, and (b) a Shannon entropy term to allow for the remaining uncertainty. Comparison with section 2.4. shows that the physical entropy is virtually equivalent to the provisional algorithmic entropy. I.e.

$$\mathcal{S}_d = H_{prov} = H_{algo}(specification\ of\ the\ set\ structure) + H_{algo}(identification\ of\ \ string\ in\ set).$$

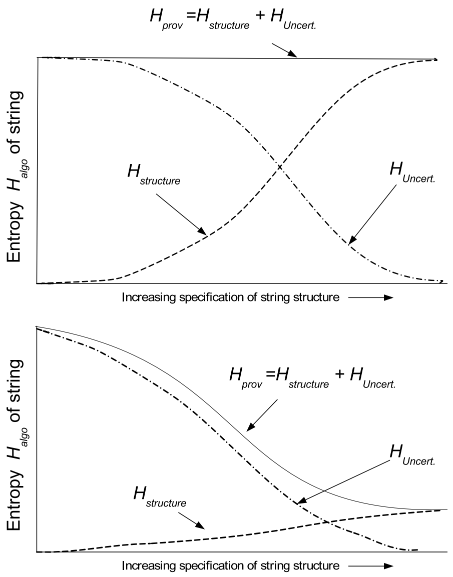
This can be summarised as $H_{structure} + H_{uncertainty}$ (see Figure 1). As the last term measures the uncertainty, the provisional algorithmic entropy decreases when the uncertainty decreases with further information; *i.e.* the provisional algorithmic entropy will drop for ordered states, but remain constant for equilibrium states.

Zurek [24] originally articulated this argument in some detail in terms of physical entropy. Our Figure 1 illustrates the two cases using an identical argument based on the provisional entropy in (a) and (b).

1. When the microstate is an equilibrium or typical state (Figure 1a), additional information cannot provide a lower value of the provisional entropy. The shortest description of the state returns the same value as the Shannon entropy despite increasing the available information.

2. When the microstate is ordered as in Figure 1b, the extra information about the actual state leads to a reduction in provisional entropy as the extent of the ordering is ascertained. This is equivalent to refining a model when further information narrows down the uncertainty. Li and Vitányi [15] describe the particular case of ascertaining the microstate of a gas. At low resolution there is only sufficient information to identify the macrostate and the corresponding provisional entropy is high because of the uncertainty. Increasing the measurement resolution of the system decreases the algorithmic entropy as the uncertainty is less. The process of doubling the resolution for each of the 6N axes corresponds to adding one more significant (binary) Figure to the specification of the coordinate of that axis. This process will ultimately identify that the configuration is an ordered one because the provisional algorithmic entropy will decrease with increasing resolution finally approaching the true algorithmic entropy.

Figure 1. Provisional entropy (a) Where the string is a typical member of the set (b) Where the string comes from an ordered configuration.



4. Maxwell's Demon, Irreversibility and Information processes

A historic difficulty with the statistical approach to thermodynamics is illustrated by the thought experiment involving an informed agent, commonly known as Maxwell's demon, who is able to extract work at no cost by judiciously manipulating the particles of a gas in a container. One version of the thought experiment makes the argument that a container of a gas can be divided into two halves both at the same temperature. Maxwell's demon opens a trap door in the dividing wall to allow faster moving particles to collect on one side of the container and closes the door to stop the faster particles returning, thus keeping the slower moving particles on the other side. Over time, the side with the faster moving particles will be at a higher temperature than the side with the slower moving particles. Work can then be extracted which violates the second law of thermodynamics.

Initially it was understood (see Szilard [35, 36] and Brillouin [37]) that, as the demon needed to measure the position-momentum coordinate of a particle to know when to open the trap door, it was necessary for the demon to interact with the system under study. It was argued that the measurement process would require an entropy increase corresponding to $k_B ln2$ for each bit of information obtained.

Landauer [38] showed that this explanation was not completely satisfactory as a thermodynamically reversible process cannot lead to an overall increase in entropy. The demon is a computational device. Landauer [39] points out that any computing process is constrained by physical laws. A conventional computation is a process which uses physical laws, under the control of a programmer, to map the input to the output in the desired way. However, the input and the output from a computational point of view are not the digits written on an input/output device such as a tape, but the initial configuration and final configuration of bits stored within the computer at the start and finish of the computation. It is these input bits that are processed by the bits in the programme, and the programme itself defines a trajectory through a state space represented by on or off bits in the computer.

While a computer can map the evolution of states in the natural world, the natural world itself can be envisaged as a Turing computer. Similarly, the information processing of the real world computation is encapsulated in what Landauer [39] and Bennett [40] call the Information bearing Degrees of Freedom (IBDF). The input string to the computation is stored in the states of the atoms and molecules at the start of the physical process.

The simple ballistic computer of Fredkin [41] illustrates how a physical process can act as a computation. However a more relevant example that illustrates the relationship between a physical or chemical process and a computation is the Brownian computer of Bennett [31, 42]. This is the computation embodied in the process by which RNA polymerase copies a complement of a DNA string. Reversibility is only attained at zero speeds as the computation randomly walks through possible computational paths. Indeed, because the process is driven forward by irreversible error correction routines that underpin natural DNA copying, the process is no longer strictly reversible.

In practice, most real world systems are open systems and information and material may enter or leave the system. As Landauer [38] and subsequently Bennett [23, 31, 42] and Zurek [24] discuss, when energy is removed, either because an information state of the system is changed, or because molecules are removed, the computational possibilities of the system are altered. Because the Hamiltonian dynamics of the system conserves information, whenever a process becomes irreversible, information must be lost. Landauer showed that the discarding 1 bit of information from the IBDF contributes about k_BTln2 joules to the environment. Both logical and thermodynamic irreversibility correspond to the situation where information is removed from the computation. This removal leads to a reduction of entropy of the system, matched by an increase of entropy to the universe.

Alternatively where information appears to be wiped (e.g. such as a chain of aligned spins randomised) a full algorithmic description of the system and the environment would need to account for the degrees of freedom in the environment acting on the system to wipe the information.

Bennett [23, 31, 40, 42], Zurek [43] show the demon cannot violate the second law of thermodynamics as it is part of the total system. Bennett (see also [43]) illustrates the argument with a gas of one particle showing how the demon traps the particle on one side of a partition in order to extract work. The entropy loss occurs when the demon memory is reset to allow the process to cycle. However, there are subtleties as Bennett [40] points out. Where information initially is random, the entropy of the system drops if the random information is erased, but the entropy of the environment still increases. Setting a random string of bits to zero is an ordering process. The key papers expressing the different interpretations of the Maxwell demon problem, and a discussion on the "new resolution" involving erasure and logical irreversibility, are to be found in Leff and Rex [44].

The Algorithmic Information Theory approach provides a framework to understand the thermodynamic cost of a real world computation as is discussed in the next section. The erasure of k_Bln2 entropy units per bit is the thermodynamic cost of the physical process. Similarly the algorithmic entropy increases by k_Bln2 per bit when information enters the system.

The consistency of the argument can be seen by considering the following examples.

- An adiabatic expansion of an ideal gas against a piston, increases the disorder. The movement of the piston in effect is like a programme, changing the computational trajectory of the components of the gas. The algorithmic entropy is derived from the shortest description of the instantaneous configuration of the gas in terms of the position and momentum coordinates of each gas particle. In algorithmic terms, while the contribution to the algorithmic entropy of the position coordinates increases through disordering, the algorithmic entropy of the momentum coordinates decreases to compensate. This decrease corresponds to a drop in the temperature. Adiabatic compression or adiabatic demagnetisation are the converse effect. Adiabatic demagnetization randomises a set of aligned spins by transferring entropy embodied in the thermal degrees of freedom to the ordered magnetic degrees of freedom thereby lowering the temperature. While a system remains isolated no erasure of information, or total change in entropy occurs.

- A closed system, such as a set of aligned spins, can be disordered or randomised by the equivalent of the isothermal expansion of a gas against a piston. The algorithmic description of the system, in contrast to the adiabatic case, increases as heat flows in; *i.e.* the thermodynamic entropy increases by $k_B ln2$ per bit. In a sense, the order is the fuel that does work in the randomising process. The converse ordering process is analogous to isothermal compression. E.g. the ordering process requires each random spin, which was previously either 0 or a 1 (representing up or down), becoming a 0 indicating that all spins are now aligned. As two states must map on to one, this process loses $k_B T ln2$ joules per spin to the environment.

- In an open, real world system, the computational processes that reset the states of the computational elements (*i.e.* the atoms, molecules, spins etc.) require energy to be expended or released. This may occur through electrical energy where pulses reset bits, or where magnetic field gradients align spins, or where material enters or leaves the system. Energy sources within the system can also be redistributed to reset the computational states as, for example, when hydrogen and oxygen are converted to water by these real world computational processes. In this case, the energy released is passed to the kinetic energy states of the system increasing the algorithmic description of the momentum states while reducing the algorithmic specification of the position and composition of the species involved. The physical (or computational) processes within the system provide an entropy gradient. Higher entropy regions, such as those embodied in the kinetic energy degrees of freedom can then pass excess entropy to the environment leaving the more ordered regions behind. Ordering only occurs when entropy is ejected; for example when heat is passed to a low entropy sink in the environment or high entropy molecular fragments escape the system.

4.1. Reversibility

Physical processes are in principle reversible and can map on to a logically reversible Turing machine that operates on the input string (representing the initial state) to produce an output string (representing the final state). The computation takes place through the operation of the physical laws determining the trajectory of the system through its states. It is only when information is discarded that the process becomes irreversible. However where this information is able to be stored in memory, reversibility can be maintained by keeping the computational history. This allows information to be inserted at critical steps when the computation is reversed. However, a Universal Turing Machine is not in general reversible, as more than one prior state can lead to a given state [40]., unless the history of the computation is stored. Nevertheless, a reversible process can be mapped on to a non reversible UTM by using a reversible algorithm that works in both the forward and the reverse direction, inserting the information that otherwise would be lost at the irreversible computational steps [13, 24, 42].

In general, the shortest algorithm able to describe the system will be less than the length of a reversible algorithm or an algorithm embodied in a real world computation. For example, the programme implicit in the Brownian computer described by Bennett [31, 42] may be able to be shortened if a path involving, say, a catalyst could be replaced by a more direct computing path. The speed of such a process might be less, but the outcome would be the same. Several authors [13, 45, 46] have considered the trade off between computer storage and number of computing steps to reproduce a given output. The shorter the computation the more information storage is required to achieve the reversible computation. For on going computations, the stored information must be erased and entropy is lost to the environment, making the process more difficult to reverse.

5. The cost of cycling an irreversible computation

These understandings allow Algorithmic Information Theory to describe the behaviour of real world systems and the maintenance of order. Maintaining order is a recycling process with a thermodynamic cost. Zurek [24] and Bennett *et al.* [47] have shown that the minimum entropy passed from the physical system to the universe in an irreversible process is $H_{algo}(i) - H_{algo}(o)$, noting that $H_{algo}(i)$ represents the entropy or information content of the initial state and $H_{algo}(o)$ that of the final state. However, a process in which a complex string is ordered is not in itself an erasure, if the history of the process is stored elsewhere in the system. It is only when the informational bits capturing the history of the process, are discarded that erasure occurs. The change in entropy represents the difference between the minimal bits added to specify the original input string, and the minimal bits discarded in the final description of the output. Reversibility requires that this lost information must be restored. Zurek [24] argues that if one knew the exact description of these states, a cyclic process based on this knowledge would be maximally efficient for extracting work. In practice the exact state is seldom known. However the provisional entropies $H_{prov}(i)$ and $H_{prov}(o)$ provide the best information available about the states i and o. If the system is to return to an equivalent state i' which has the same provisional algorithmic entropy as i, the entropy change from i to o must be compensated for.

This approach has led to an algorithmic equivalent of Ashby's law of "requisite variety" [48] which is the governing principle behind the maintenance of homeostasis by an open autonomous system. If variety denotes the total number of available states, Ashby's law of requisite variety states that to achieve control, the variety in the regulatory or control system must equal, or be larger than, the variety of the perturbations. In effect the regulatory part of the system must generate sufficient internal variety to match the variety disturbances from the external environment (See Casti [49] for a review). The law is sufficiently general that it can be applied to any complex system whether such a system is a firm, an economy or a living system.

While the law can be derived from a game theory approach, Ashby developed a more general approach using the logarithm of the number of available states to measure variety [48]. In this case variety becomes the Shannon entropy H_s. If regulation is to be effective, there must be an appropriate state in the regulating set of states R that can react to every disturbance in in the set $\mathcal{D}$. This implies, given the set of disturbances $\mathcal{D}$, that $H_s(\mathcal{R}|\mathcal{D})$ is zero - there is no uncertainty about the state of the regulation component. Ashby showed that $H_s(\mathcal{D}) = H_s(\mathcal{R})$; *i.e.* the logarithmic measure of variety given by the number of states in the regulatory system must match the logarithmic measure of the variety of the disturbances.

From an algorithmic point of view, any disturbance has the capacity to shift the state ϵ_i in the set $\mathcal{E}$, the current state in the viable region, to a state η_j outside the viable region. The disturbance resets the system trajectory to give the output η_j and the algorithmic change will be $H_{prov}(\epsilon_i) - H_{prov}(\eta_j)$. The regulatory system must compensate for this effectively providing a computational path that redirects the trajectory to another viable state ϵ_f having the same provisional entropy. If ϵ_i and ϵ_f are typical states in the viable region, they will both have the same provisional algorithmic entropy and belong to the same macroscopic state of the system, analogous to the traditional entropy for an equilibrium state. This leads to the algorithmic equivalent of Ashby's law. For a system to maintain itself off equilibrium, Ashby's law becomes the requirement that the provisional algorithmic entropy of the regulating system match that of any disturbance. This is a necessary, but not a sufficient condition for regulation, as the regulator must have an appropriate, or in Ashby's terms, a requisite response. Ashby's law can then be rewritten as :

If a regulatory process of a system is to have an effective response to any external disturbance, the provisional algorithmic entropy of an appropriate regulatory process must match the provisional algorithmic entropy of any disturbance.

As the algorithmic approach focuses on individual disturbances and individual responses, whenever the description of the disturbance can be compressed because of pattern or structure, the provisional algorithmic entropy of the regulating programme will be less than where no order is recognised. If the regulatory system can model the disturbance, greater regulation capability exists than where the disturbances appear to be random. While the approach is appropriate for individual disturbances and responses, it also applies to sets of disturbances. However, as the concept of variety only has meaning for sets of states, it is not an appropriate term to describe the entropy of an individual state used in the algorithmic formulation of the law.

6. Replication and Algorithmic Information Theory

If the universe started in a highly ordered state, how did the far from equilibrium local order emerge as the universe trended towards equilibrium? Algorithmic Information Theory suggests that the physical process of replication accesses existing order, usually in the form of high grade energy, and ejects disorder to create new ordered structures. Replication, to some extent offsets the disorder arising from the free expansion of the universe.

A replicating system can be an autocatalytic set, bacteria growing in an environment of nutrients, a crystal that forms from a melt, a set of spins that align in a magnetic material, or coherent photons that emerge through stimulated emission. These are physical or biological structures that can reproduce by utilizing energy and resources from an external environment. In a resource rich environment, where the probability that a structure will replicate increases with the number of existing structures, replicated structures are more likely to be observed than alternative structures. For example, molecules are more likely to solidify on a seed crystal in a melt and, given one strand of DNA in the right environment, the probability of a second strand of DNA appearing is comparatively high. Where resources are limited, the number of replicates grows over time until a state of homeostasis is reached; a state where the set of replicates and the environment reach a long-term stable relationship. Here the noun 'replicate' is used to distinguish the copied structure, such as the DNA information string, from the full replicating system which, in the DNA case, is the whole cell.

Structures consisting of replicates have low entropy as they can be simply described by an algorithmic "Repeat replicate N times". The algorithmic entropy of such a system.

$$H_{algo}(system) = |N| + |replicate\ description| + |CI|,$$

where CI refers to the common information.

Algorithmic Information Theory makes the following points [50, 51]

- Structures generated by replication processes are highly ordered, having low algorithmic entropy, and are more likely to emerge than similar structures produced by non replicating or random physical processes.

- Replication processes can maintain an ordered system away from equilibrium in an attractor-like region of the system's dynamical state space, where all the states have the same provisional algorithmic entropy.

- Variation in replicated structures leads to an increase in the algorithmic entropy. Nevertheless, variation allows the dynamical system to maintain homeostasis in a changing environment by providing a mechanism for the system to evolve to a more restricted region of its state space - *i.e.* diversity in the replication process can maintain the system in a stable configuration through adaptive evolutionary-like processes.

- Coupled replicator systems create greater stability against change by co evolving.

6.1. Replication

Consider a physical system, initially isolated from the rest of the universe where the instantaneous microstate of the system is represented by a point in the multidimensional state space. Over time the state point of the system will move through state space under the operation of classical physical laws. As the probability of replicates appearing increases with their occurrence, replication will drive the state point to a region of state space dominated by a large number of repeated replicated structures. Other physical processes, such as collisions or chemical reactions, will destroy replicates and the system will eventually settle in a region of long term stability.

6.2. Examples of replication

A fixed structure of identical replicates is simple to describe algorithmically. Consider a simple replicating system of coherent photons produced by a set of atoms with an excited state and a ground state [50, 51]. Assuming that initially all the atoms are in the excited state, once a photon is emitted, that photon can stimulate others to generate a coherent photon system. If the system is isolated the algorithmic description of an instantaneous configuration of the total system after a period of time will consist of the description of the atomic states in terms of whether they are excited or in the ground state; a description of the state of each incoherent photons and finally the description of the coherent photon states which are replicates of the original stimulating photon. Because the process is like a free expansion, without replication of coherent photons, the entropy of the total system would increase until equilibrium is reached. While the replication of coherent photons creates local order, disorder must be passed to the momentum states of the atoms. In general a replication process creates order by passing the disorder elsewhere; effectively partially randomizing other degrees of freedom. If the system is isolated for long periods, the system will settle in a region of state space where replicates die and are born as the total system moves through possible configurations. Nevertheless, the system is still in principle reversible. It is only when entropy embodied in the thermal degrees of freedom is passed to other degrees of freedom in an external sink, that the system will settle in an attractor region of the state space. The length of the algorithmic description is reduced and the entropy cost of this is $k_B ln2$ per bit.

However if photons can escape, the system will need to access a low entropy source to re-create some excited state to allow the photon loss to be replenished. In other words to maintain such a system off equilibrium the order needs to be replaced, as for example would happen if external photons can excite the atomic states. The cost of maintaining the system in such a configuration was discussed in section 5.

6.3. The entropy cost of replication with variations

Many real world systems are made of non-identical replicates. For example the expression of a cell in the eye is not identical to one in the liver. In this case the provisional algorithmic entropy provides a measure of the entropy increase due to the uncertainty of the variations; the algorithmic description is longer because of the uncertainty. The provisional entropy approach provides a tool to measure the increase in entropy due to variation. As Equation 10 implies, if there are $\mathcal{V}$ members in the set of strings where variations in the replicate are possible, the provisional entropy is:

$$H_{prov}(o) \cong \log_2 \mathcal{V} + |description\ of\ pattern\ of\ replicates|. \tag{15}$$

For example, if there are $\mathcal{M}$ different variations of the replicate and these form a sequence of L replicates, there will be $\mathcal{M}^L$ members in the set of possible strings.

The algorithm that produces the replicate string must include the specification of the number of different variations $\mathcal{M}^L$, the number of replicates in the sequence (L), while P, the size of the replicate description usually needs to be specified to define the set of replicates. In which case the provisional entropy for a string o of variable replicates is:

$$H_{prov}(o) \cong L(\log_2 M) + \log_2 L + \log_2 P + |description\ of\ pattern\ of\ subset\ given\ P| + O(1). \tag{16}$$

Where the replicates die and are born, non replicate structures will need to be specified [50, 51]. If information or entropy is being lost, the attractor region needs to be maintained off equilibrium by accessing order from a higher energy source.

6.4. Adaptation of replicates in an open system

The new resources flowing into an open system are seen as additions to the input string expanding its state space. Similarly, resources flowing out of the system lead to a loss of information and a contraction of the state space. Where a changing input mix creates new computational paths some variations of the replicate may become less likely, while others may become more likely. As the replicates that dominate will represent only a subset of possible replicates. The attractor region of the state space will contract.

7. The second law of thermodynamics

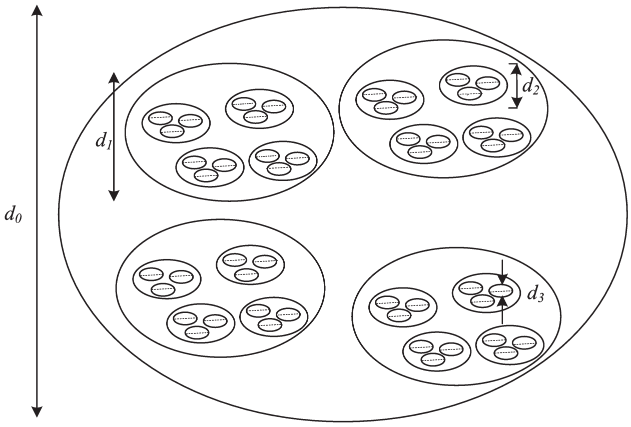
Devine [50, 51] has discussed how the replicating system follows a trajectory towards an attractor region of its state space. An account of all the entropy or information flowing in and out of a system implies that information is neither created nor destroyed, but is conserved. The question is whether this is consistent with the second law of thermodynamics. That the approach is consistent can be seen by considering a simple model of a classical universe which assumes the universe evolves through a series of discrete computational steps from an initial state s_0, (which for convenience is taken to be one shortly following the Big Bang) through an incomprehensibly large, but finite, set of discrete states W.

Zurek Appendix C [24] has argued that a closed dynamical system evolves from a low entropy initial state s_0 where $H_{algo}(s_0) << \log_2 W$, and where $\log_2 W$ corresponds to the Boltzmann entropy of a system with W states. Such a system will eventually return to the initial state in the Poincaré repeat number of steps which, for an ergodic system will be W. While from an observers point of view such a system is chaotic, from the computational point of view the universe itself is the computer and, ignoring quantum effects, can be considered deterministic. In this process, the traditional entropy increases as the universe initially undergoes a free expansion into a discrete set of states that were previously not accessible and ultimately for most of the existence, the universe will be in an equilibrium configuration.

Following this approach, the algorithm describing the present state of the universe after t steps from the beginning is of the form:

```
$STATE = s_0$
$FOR\ STEP = 0\ to\ t$
  $Compute\ next\ STATE.$
$NEXT\ STEP$ .                                              (17)
```

Figure 2. Nested structures at scales d_0, d_1, d_2 and d_3.



The length of the reversible algorithm is $\approx |initial\ state| + |t^*| + |physical\ laws|$ where t^* is the shortest representation of the number of computational steps t undertaken. At the early stages of the evolution of the universe, the algorithmic entropy will be dominated by $|t^*|$. While this fluctuates, as Li and Vitányi [15] point out, $|t^*|$ mostly hugs $\log_2 t$. However, if the universe is assumed closed, after a time t' where, $\log_2 t' \gg \log_2 t$ the most likely configurations are the typical equilibrium states. In which case as $\log_2 t$ approaches $\log_2 W$, the typical sized algorithm will be dominated by $\log_2 W$. As some highly ordered states exist in the set of equilibrium states, spontaneous fluctuations from equilibrium can occur.

8. Coupling of replicator systems

When entropy or information flows out of a system that is not at equilibrium, this must be replenished if the system is to maintain itself. However, where a system, such as a replicating system, accesses resources from another replicating system (for example where the input string of one replicating system accesses the output of another) the systems are coupled. As entropy flows between the systems the overall less entropy is lost and there is a lower throughput of energy. In which case the replicating systems become interdependent and tend to stabilize each other.

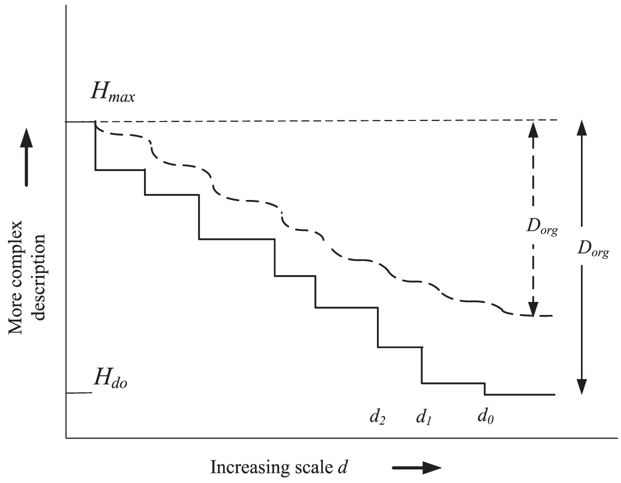
A simple example, is where photons from one laser system create a population inversion in another laser system, there is less information lost to the environment. Where sufficient variety in the systems' states occur, each system may adapt by settling in more restricted areas of its state space; the coupled systems co evolve by using resources more efficiently. In a resource constrained environment, dependence will emerge in preference to alternatives, as the coupled systems are more stable against input perturbations. Their mutual attractor region will not drift through state space at the same rate as similar, but uncoupled, systems.

8.1. Nested systems

As nested structures, such as cells nested within higher structures, can be described by nested algorithms they are more ordered. Chaitin's [52] concept of 'd-diameter complexity', which quantifies order at different levels of scale, applies to nested systems.

Figure 2 illustrates a nested system of replicates. Let H_{d0} represent the algorithmic entropy of the

Figure 3. Variation of d-diameter complexity with scale; — nested replicators with no variation; – – – – – nested replicators with variation; - - - - No organization at any scale.



system, based on its minimal description at the largest scale d_0 [50, 51]. However, where the system is dismantled so that the large scale structure is suppressed, the system is perceived as a collection of structures at scale $d_1 < d_0$. As the scale reduces, the algorithmic entropy cannot decrease. However whenever large scale pattern that previously could be specified by a compressed algorithm is lost, the algorithmic entropy increases as the structure now must be specified in detail. I.e. the algorithmic description where the large scale structure is suppressed, must specify each substructure at this scale and how the substructures are to be assembled. This invariably will be more lengthy than a description using the algorithm that specified the overall pattern.

Referring to Figure 3, at the scale level where $d_1 \leq d \leq d_0$, $H_d > H_{d0}$. The entropy remains about the same until the scale reduces below that of d_1, the next level of nesting. The algorithmic description at the scale $d_2 < d_1$ must include more detailed specifications and assembly instructions. At the smallest scale, the algorithmic entropy reaches a maximum.

Figure 3 captures how the algorithmic entropy decreases as the scale is increased. The stepped bold line shows an ideal system where the nested systems at each level of scale are identical. As more pattern is recognised at increasing scales, allowing shorter algorithmic descriptions to be found, the algorithmic entropy decreases. The dashed line in Figure 3 shows how variations in replicates at a given scale smooth out the steps in the ideal case, leading to a lower decrease of algorithmic entropy with scale. However, where no organization at all exists the algorithmic entropy is the same at all levels of scale as shown by the dotted horizontal line, H_{max} in Figure 3. Chaitin [52] quantifies the degree of organization (D_{org}) of structure X by:

$$D_{org} = H_{max}(X) - H_{d0}(X).$$

The degree of organization, corresponds to Kolmogorov's "deficiency in randomness'. This is a measure of how far a system is from equilibrium. While systems with high D_{org}, have lower entropy, they do not have the same flexibility to adapt as systems with variation and lower D_{org}. It would appear that nesting can increase organization and thereby decrease entropy faster than the entropy cost of the variation.

Indeed, this may be an inevitable consequence of selection processes acting on structures. Interestingly, as software variation occurs at lower levels of scale, it would appear to be algorithmically more efficient to generate variation through software (e.g. variation in DNA) rather than directly.

9. Conclusion

This review show that the concept of algorithmic entropy, despite some awkward features, is consistent with conventional understandings of entropy. The approach provides particularly useful insights into reversibility, the cost of maintaining a system off equilibrium, and how order might emerge in natural systems. Because the approach highlights how order is often nested within order algorithmic entropy provides a tool for focussing on the order at any level of scale, While it is too difficult to provide a detailed algorithmic entropy measure of most real systems, the approach may be useful in understanding incremental changes and as well provide broad descriptions of system behaviour.

References and Notes

1. (a) Solomonoff, R. J. A formal theory of inductive inference; Part 1. *Information and Control* **1964**, *7*, 1–22; (b) Solomonoff, R. J. A formal theory of inductive inference; Part 2. *Information and Control* **1964**, *7*, 224–254.
2. Kolmogorov, K. Three approaches to the quantitative definition of information. *Prob. Info. Trans.* **1965**, *1*, 1–7.
3. Chaitin, G. On the length of programs for computing finite binary sequences. *J. ACM* **1966**, *13*, 547–569.
4. Levin, L. Laws of information (nongrowth) and aspects of the foundation of probability theory. *Problems Inf. Transm.* **1974**, *10*, 206–210.
5. Gács, P. On the symmetry of algorithmic information. *Sov. Math.-Doklady* **1974**, *15*, 1477–1780. Correction: ibid., 15 (1974) 1480.
6. Chaitin, G. A theory of program size formally identical to information theory. *J. ACM* **1975**, *22*, 329–340.
7. Ratsaby, J. An algorithmic Complexity interpretation of Lin's third law of Information Theory. *Entropy* **2008**, *10*, 6–14.
8. Vitányi, P. M. B.; Li, M. Minimum description length induction, Bayesianism, and Kolmogorov Complexity. *IEEE Trans. Inf. Theory* **2000**, *46*, 446–464.
9. Rissanen, J. Modeling by the shortest data description. *Automatica* **1978**, *14*, 465–471.
10. Rissanen, J. Stochastic complexity. *J. Royal Stat. Soc.* **1987**, *49B*, 223–265 and 252–265.
11. Rissanen, J. *Stochastic Complexity in Statistical Inquiry*; World Scientific: New Jersey, USA, 1989.
12. Rissanen, J. A universal prior for integers and estimation by minimum description length. *Ann. Stat.* **1983**, *11*, 416–431.
13. Li, M.; Vitanyi, P. Reversibility and adiabatic computation: trading time and space for energy. *Proc. Royal Soc. London, Series A* **1996**, *452*, 769–789.
14. Zvonkin, A.; Levin, L. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russ. Math. Survs.* **1970**,

25, 83–124.

15. Li, M.; Vitanyi, P. M. B. *An introduction to Kolmogorov Complexity and its Applications*, 2^{nd} Ed.; Springer-Verlag: New York, NY, USA, 1997.
16. Calude, C. *Information and Randomness:An Algorithmic perspective*, 2^{nd} Ed.; Springer-Verlag: Berlin, Germany, 2002.
17. Minsky, M. Size and structure of a Universal Turing Machine using tag systems. In *Recursive Function Theory*, Proc. Symposium in Pure Mathematics; AMS: Providence, RI, USA, 1962; Vol. 5, pp. 229–238.
18. Wolfram Institute. Wolfram's 2,3 Turing Machine is universal, 2007.
19. Chaitin, G. Two philosophical applications of algorithmic information theory. In Proc. DMTCS'03, Springer Lecture Notes in Computer Science; Calude, C. S., Dinneen, M. J., Vajnovszki, V., Eds.; Springer-Verlag: Berlin, Germany, 2003; Vol. 2731, pp. 1–10.
20. Tromp, J. Binary lambda calculus and combinatory logic. http://homepages.cwi.nl/~tromp/cl/LC.pdf, 2009.
21. Rissanen, J. Generalized Kraft inequality and arithmetic coding. *IBM J. Res. Develop.* **1976**, *20*, 198–203.
22. Chaitin, G. J. Algorithmic Information Theory. *IBM J. Res. Develop.* **1977**, *21*, 350–359,496.
23. Bennett, C. H. Logical depth and physical complexity, In *The Universal Turing Machine- a Half-Century Survey*; Herken, R., Ed.; Oxford University Press: Oxford, UK, 1988; pp. 227–257.
24. Zurek, W. H. Algorithmic randomness and physical entropy. *Phys. Rev. A* **1989**, *40*, 4731–4751.
25. Gács, P. The Boltzmann entropy and randomness tests- extended abstract. In Proc. Workshop on Physics and Computation; IEEE Computer Society Press, 1994; pp. 209–216.
26. Gács, P. The Boltzmann entropy and random tests. http://www.cs.bu.edfaculty/gacs/papers/ent-paper.pdf, 2004.
27. Devine, S. D. The application of Algorithmic Information Theory to noisy patterned strings. *Complexity* **2006**, *12*, 52–58.
28. Shalizi, C. R.; Crutchfield, J. P. Pattern discovery and computational mechanics. [arxiv:cs/0001027v1], 2000.
29. Vereshchagin, N. K.; Vitányi, P. M. B. Kolmogorov's structure function and model selection. *IEEE Trans. Inf. Theory* **2004**, *50*, 3265–3290.
30. Bennett, C. H. How to define complexity in physics, and why, In *Complexity, Entropy and the Physics of Information*; Zurek, W. H., Ed.; Addison-Wesley, Redwood City, CA, USA, 1990; pp. 137–148.
31. Bennett, C. H. Thermodynamics of computation- a review. *Int. J. Theor. Phys.* **1982**, *21*, 905–940.
32. Solovay, R. M. A version of O for which ZFC can not predict a single bit, In *Finite Versus Infinite. Contributions to an Eternal a Dilemma*; Calude, C. S., Paun, G., Eds.; Springer-Verlag: London, UK, 2000.
33. Calude, C. S.; Dineen, M. J. Exact approximation of Omega numbers. *Int. J. Bifurcation Chaos* **2007**, *17*, 1–18.
34. Gács, P. Lecture notes on descriptional complexity and randomness. Technical report, Boston University Computer Science Department, 1988.

35. Szilard, S. Uber die Entropieverminderung in einnem thermodynamischen System bei Eingriffen intelligenter Wesen. *Zeitschr. f. Phys.* **1929**, *53*, 840–856 (in German).
36. Szilard, L. On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings, 2003.
37. Brillouin, L. *Science and Information Theory*, 2^{nd} Ed.; Academic Press: New York, NY, USA, 1962.
38. Landauer, R. Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.* **1961**, *5*, 183–191.
39. Landauer, R. Information is physical. In Proc. PhysComp 1992, Los Alamitos, 1992; IEEE Computer Society Press, Oxford, 1992; pp. 1–4.
40. Bennett, C. H. Notes on Landauer's principle, reversible computation, and Maxwell's demon. http://xxx.lanl.gov/PS_cache/physics/pdf/0210/0210005.pdf, 2003.
41. Fredkin, E.; Toffoli, T. Conservative logic. *Int. J. Theor. Phys.* **1982**, *21*, 219–253.
42. Bennett, C. H. Logical reversibility of computation. *IBM J. Res. Develop.* **1973**, *17*, 525–532.
43. Zurek, W. H. Thermodynamics of of computation, algorithmic complexity and the information metric. *Nature* **1989**, *341*, 119–124.
44. Leff, H. S.; Rex, A. F. *Maxwell's Demon: Entropy, Information, computing*; Princeton University Press, Princeton, 1990.
45. Buhrman, H.; Tromp, J.; Vitányi, P. Time and space bounds for reversible simulation. *J. Phys. A: Math. General* **2001**, *34:35*, 6821–6830.
46. Vitányi, P. Time space and energy in reversible computing. In Proc. 2005 ACM International Conference on Computing Frontiers, Ischia, Italy, 2005; pp. 435–444.
47. Bennett, C. H.; Gács, P.; Li, M.; Vitányi, P. M. B.; Zurek, W. H. Information distance. *IEEE Trans. Inform. Theory* **1998**, *44*, 1407–1423.
48. Ashby, W. R. *Introduction to Cybernetics*; University Paperbacks: London, UK, 1964.
49. Casti, J. The great Ashby:complexity, variety, and information. *Complexity* **1996**, *2*, 7–9.
50. Devine, S. D. An algorithmic information theory approach to the emergence of order using simple replication models. http://arxiv.org/PS_cache/arxiv/pdf/0807/0807.0048v3.pdf, 2008.
51. Devine, S. D. An algorithmic information theory approach to the emergence of order using simple replication models, In *First International Conference on the Evolution and Development of the Universe*, 2008.
52. Chaitin, G. Toward a mathematical definition of "Life", In *The Maximum Entropy formalism*; Levine, R. D., Tribus, M., Eds.; MIT Press: Boston, MA, USA, 1979; pp. 477–498.