Thermodynamic cost of computation, algorithmic complexity and the information metric

W. H. Zurek

Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

Algorithmic complexity is a measure of randomness. In contrast to Shannon's entropy it is defined without a recourse to probabilities; for a binary string *s* it is given by the size, in bits, of the shortest computer program with the output *s*. I show that algorithmic complexity sets limits on the thermodynamic cost of computations, casts a new light on the limitations of Maxwell's demon and can be used to define distance between binary strings.

REVERSIBLE computation can be accomplished only by using computer memory to keep track of the exact logical path from the input to the output. This conclusion[1-3], is based on the observation that thermodynamic irreversibility is inevitable only in presence of logically irreversible operations[4-6]: If several input states lead to the same output, the loss of information in such a many-to-one mapping makes it impossible reversibly to 'backtrack' the machinery of the computer. To allow reversible operation the computer must retain this additional information (the history of all the logically irreversible steps) at least temporarily, and it must retain at the end of the computation at least enough extra information to assure unambiguous backtracking[1,2,6,7]. Thus, reversible computation can be achieved only at the expense of filling up computer memory with historical records, aptly named 'garbage'[2,6].

But suppose one insists on replacement of the input with the output in the computer memory: How irreversible would the resulting computation be? This question about the 'replacement computation' leads to simple, yet interesting consequences, among them an information-theoretic measure of distance between binary sequences (which may represent physical configurations) or between statistically characterized ensembles. Thermodynamic efficiency of various information processing operations is of interest in physics of computation[1-9], and replacement computation is relevant for the foundations of thermodynamics[10].

Consider a simply describable universal computer C. Computations start with some program i as the input, and with the computer in a definite initial state. After the computer stops, only the output o should be present on the output tape and the computer should end up in some unique 'halting state.' As no permanent (only temporary) storage of 'garbage' is allowed, there is no possibility of retaining the history. In general, the computation of o from i will be logically and thermodynamically irreversible.

I shall demonstrate that the least increase of entropy $\delta S(i \rightarrow o)$ associated with the replacement depends on the input and output strings and is at least equal to the difference in length of the shortest programs i^* and o^* to compute i and o on C

$$\delta S(i \rightarrow o) \gtrsim |i^*| - |o^*| \tag{1}$$

Vertical lines denote the size of programs in the number of symbols. When the computer alphabet is binary the difference of entropy is given in bits.

Equation (1), the first main result of this paper, is derived below, where I also show that the optimum thermodynamic efficiency is difficult to attain: the minimal programs needed to guarantee it cannot be found systematically because of the undecidability of the halting problem—a Turing-machine version of Gödel's theorem.

The total amount of information exchanged in an update $s \rightarrow t$ (the minimal number of erased plus the minimal number of added bits needed to transform file s into file t) provides a natural measure of the information distance between two strings. I also show that this information distance satisfies conditions expected of a metric for both algorithmic and statistical measures of information content. Finally, discussion of the physical significance of the results shows how the logical irreversibility of replacement computations limits thermodynamic efficiency of Maxwell's demon. I propose that algorithmic complexity can be used to define the entropy from the point of view of the demon-like 'intelligent observer', in a manner that takes into account both the lack of complete information and the degree of randomness of the available data about the physical system.

Algorithmic information theory

Algorithmic randomness[11-22] of a string $K(s)$, also known as algorithmic complexity or algorithmic information content, is defined by the size of the minimal program s^* that, when executed on a universal computer, yields output s

$$K(s) \equiv |s^*| \tag{2}$$

Any message can be represented as a binary string (a sequence of 0s and 1s), but some messages can be reproduced from more concise descriptions: a binary string consisting of 10^6 0s could not be printed in this paper but its content can be (indeed, already has been) communicated to the reader. By contrast, a typical random string of 10^6 0s and 1s obtained by flipping a coin cannot be compressed to a similarly concise description. Incompressible strings are called algorithmically random, and strings that can be characterized by messages small compared to their length in bits are called algorithmically simple.

Equation (2) captures the intuitive difference between the information content and the size of a string (a universal computer is stipulated only to make the definition reasonably independent of the addressee of the message). A string that appears random may nevertheless have a concise description. Binary representations of π, or $\sqrt{2}$, are good examples of apparently random, but, in fact, algorithmically simple strings. In general, no algorithm can distinguish random from simple strings, a difficulty related to Gödel's undecidability[16-18,22]. As I show below, however, $K(s)$ can be estimated from above by a simple algorithm for any finite s. Moreover, most strings cannot be compressed, so the typical algorithmic randomness of a string is, to leading order, given by its length in bits $|s|$. For example, the algorithmic randomness of a natural number n is typically $\log_2 n$. Certain classes of string are algorithmically random or simple; readers are invited to demonstrate that minimal programs used in the definition of algorithmic complexity are algorithmically random.

As in Shannon's information theory, one can define algorithmic randomness for sets of strings. Joint algorithmic complexity of a pair of strings (s, t) is the size of the smallest program required to print them both on the output tape. It satisfies the 'commuting' property

$$K(s, t) = K(t, s) + O(1) \tag{3}$$

and the inequality

$$K(s, t) \leq K(s) + K(t) + O(1) \tag{4}$$

The error term of O(1) comes from the slight computer dependence of the algorithmic information content[13-19]. I shall henceforth omit O(1) and replace equality and inequality signs with their approximate versions ($= \to \simeq$; $\geq \to \gtrsim$, for example) to simplify the notation.

Conditional algorithmic information $K(s|t)$ is the size of the smallest program $q^*_{s|t}$ that can compute string s from the string t (or, equivalently, both s and t from t)

$$K(s|t) \equiv |q^*_{s|t}| \tag{5}$$

Conditional information will be central in our discussion. By analogy with the statistical information theory one would expect $K(s|t)$ to obey

$$K(s, t) = |t^*| + |q^*_{s|t}| \simeq K(t) + K(s|t) \tag{6}$$

This equation is satisfied, but only approximately[19]. Correction terms of order of $O(\log_2 K(t))$ may arise, with their exact form dependent on the adopted definition of $K(s|t)$. However, equation (6) will hold with no more than O(1) errors if, in addition to string t, one is willing to provide its algorithmic randomness $K(t)$ as the supplementary datum

$$K(s, t) \simeq |t^*| + |q^*_{s|t,K(t)}| \simeq K(t) + K(s|t, K(t)) \tag{6a}$$

Moreover, conditional algorithmic randomness of s given t^* (rather than simply t)

$$K(s|t^*) \equiv |q^*_{s|t^*}| \tag{5a}$$

also satisfies equation (6) without logarithmic errors[20]

$$K(s, t) \simeq |t^*| + |q^*_{s|t^*}| \simeq K(t) + K(s|t^*) \tag{6b}$$

Rigorous definition of algorithmic randomness imposes certain requirements on the properties of the minimal programs. In particular, it is convenient to require that minimal programs be self-delimiting[19-21]—they must contain no special symbol to mark their ends, and must therefore contain information about their size. For instance, algorithmic randomness of a typical string defined through a self-delimiting minimal program is given by $|s| + \log_2 |s|$, where the logarithmic term enters because of the necessity to communicate, in addition to the content of s, its size.

Reversibility of compression

What is the least thermodynamic expense of generating string s from its minimal program s^*? This question emerges in the context of compressing information: by definition, the minimal program s^* can be used to reproduce the more 'verbose' record s, so the operation of replacing s by s^* will preserve the content, but will free $|s| - |s^*|$ bits of memory.

I shall demonstrate that, given s^*, the replacement of s by s^* can be achieved reversibly, with no cost in terms of entropy increase. Note that a reversible program can compute s from s^* and in the process generate history $g(s^*, s)$, which allows it to backtrack from s to s^* reversibly. The process of getting rid of s and replacing it with s^* is:

■ Initially, the computer memory contains s and s^*; The complete input consists of $\{s, s^*\}$.

■ The reversible execution of s^* yields output s and history $g(s^*, s)$. The computer memory now contains two copies of s and $g(s^*, s)$, but does not contain s^*, which was used up, that is, transformed into s in a reversible computation.

■ The newly generated string s can reversibly cancel the 'old' s: $(s, s \to 0 \ldots 0, s)$. This frees up $|s|$ bits, but to make this operation logically reversible one copy of string s must be retained[2]. The memory now contains string s and 'garbage' $g(s^*, s)$.

■ Finally, the computation of s from s^* can be reversed. The computer has all the directions, $g(s^*, s)$, to undo the original computation of s from s^* reversibly.

■ At the end of the operation the computer memory contains only s^*. With no loss of information content, and at no thermodynamic cost, $|s| - |s^*|$ memory bits were liberated.

The string substituting s did not have to be minimal, or even shorter than s: all that was needed was the assumption that the execution of the input s' results in the output s.[1,2,7]

Lemma 1—Program s', which generates string s as an output, can be used to reversibly erase s from the memory of the computer, $(s', s) \to s'$. Minimal program s^* can be employed in this manner to reversibly compress information contained in s.

The essential criterion that makes compression of s possible is that the information in s must be available in the form of some other string s'. So far, I have assumed that the concise s' was already available. To accomplish compression, this s' was used in an algorithm outlined in the proof of Lemma 1. But it is more natural to consider a computer that 'looks for' some convenient and concise s' given the original file s. Lemma 1 then sets the limit on the compressibility of s.

A simple instance of such compressibility occurs when there are two copies of s (ref. 2). Availability of a program s' generating s as an output can be regarded as equivalent to the availability of an implicit copy of s. One can also imagine a more complicated situation where the information needed to generate s resides in a collection of several strings s', s'', ... which jointly generate s.

The ability of one binary string s to share information content with another string t can be expressed in terms of mutual algorithmic information defined by[14,20]

$$K(s:t) = K(s) + K(t) - K(s, t) \tag{7}$$

Two strings are algorithmically independent when $K(s:t) \simeq 0$. Two copies of the same string are algorithmically redundant: $K(s:s) \simeq K(s)$. Moreover, string s is algorithmically redundant with respect to string t when

$$K(t) > K(s) \tag{8a}$$

and

$$K(s:t) \simeq K(s) \tag{8b}$$

or equivalently

$$K(s, t) \simeq K(t) \tag{8c}$$

In other words, s is algorithmically redundant with repect to t when it contains no additional information not already included in t. In this sense, the output must be redundant with respect to the input.

Reversibility of the replacement operation $(s', s) \to s'$ relies on redundancy of the output string s with respect to the input s'. Lemma 1 is an example of the more general fact established in ref. 1: if $s \to f(s)$ is an arbitrary computable function, operations $s \to (s, f(s))$ and $(s, f(s)) \to s$ can be carried out reversibly because the inputs and the outputs are mutually redundant. What is new here is the application: Lemma 1 uses algorithmic complexity to establish the limit on the thermodynamically reversible compressibility of information.

Is redundancy not just a necessary but also a sufficient condition for the reversibility of the replacement operation $(s', s) \to s'$? It is true that whenever $K(s', s) - K(s) \simeq 0$ there must exist, by definition of conditional algorithmic information, a short minimal conditional string $q^*_{s|s'}$ capable of turning s' into a program with output s. But the existence of a concise program generating s does not guarantee its availability because minimal programs are not recursively computable functions of their respective outputs[13-20]. The possibility of a reversible compression of s into the s^* or the existence of $q^*_{s|s'}$ thus establishes limits on thermodynamic efficiency, but does not guarantee that optimal efficiency can be attained. Nevertheless, the absolute thermodynamic cost of computations is of interest for the foundations of statistical mechanics and, as demonstrated in more detail below, can be settled through arguments involving existence rather than computability.

Thermodynamic cost of replacement computation

I now investigate the least thermodynamic price (the optimal attainable efficiency) of a general replacement computation $i \rightarrow o$. The output is redundant with respect to the input, and therefore typically contains less information. The replacement computation of o from i thus disposes of some information, and must be logically irreversible. These logically irreversible steps can be either 'paid for' at once, the history of the computation being erased as the computation is carried out, or reversibly recorded, which delays thermodynamically costly erasures until after the computation is completed.

The cost of erasures will not increase if they are delayed by making a record $g(i, o)$ of all the irreversible steps and erasing it after the computation is completed—a bit of information is erased at the same cost regardless of when the erasure is done. One can therefore consider reversible computations $i \rightarrow o, g(i, o)$ and, without loss of generality, establish the lower bound on their efficiency by finding the least thermodynamic cost of erasure of $g(i, o)$ in the presence of o. Because the history will be generally algorithmically non-random and may contain information already present in the output, any concise program $q_{i|o}$ that can reproduce i from o and thereby reconstruct the history $g(i, o)$ can be also used, by Lemma 1, to reversibly compress $g(i, o)$. This line of reasoning can be developed into:

Theorem 1—The thermodynamic cost of computation which turns input i into output o, retaining no memory of i or of the intermediate steps cannot be made lower than

$$\delta S(i \rightarrow o) = K(i|o) \tag{9}$$

where $K(i|o)$ is the size of the minimal conditional string $q^*_{i|o}$.

The proof is accomplished by contradiction. Let the history $\tilde{g}(i, o)$ allow computation of o from i, and suppose that it contains fewer than $K(i|o)$ bits, $|\tilde{g}(i, o)| < K(i|o)$. Then $\tilde{g}(i, o)$ could be used to reproduce i from o with a program shorter than the minimal conditional program $q^*_{i|o}$, which is impossible. Therefore $q^*_{i|o}$, containing $K(i|o)$ bits, is the shortest possible record of the history.

Erasure of the conditional string cannot be accomplished at a cost of less than $\delta S(i \rightarrow o) = |q^*_{i|o}|$ bits of entropy, which demonstrates Theorem 1. It also establishes:

Corollary—There exists an abbreviated version $q^*_{i|o}$ of the history of the computation $i \rightarrow o$ with $K(i|o)$ bits. It is typically much shorter than the original history $g(i, o)$ but, like g, it enables i to be computed from o and can hence be used to reversibly reconstruct g itself. Unlike g, however, it is not necessarily computable from the input.

Uncomputability means that although Theorem 1 and its corollary establish a firm lower bound on the size of the history and demonstrate that this bound can be (not by recursive computation, but perhaps by sheer luck) actually met, they offer little help in the task of minimizing thermodynamic expenditures.

The lower bound of Theorem 1 can be used to derive equation (1), a slightly weaker but more direct bound expressed in terms of the algorithmic complexities of i and o. Because the output is redundant with respect to the input, $K(i, o) \simeq K(i)$, and from equation ($6a$), $K(i|o, K(o)) \simeq K(i) - K(o)$. The size of the minimal program that computes i from o alone cannot be less than $K(i) - K(o)$; $K(i|o) \gtrsim K(i|o, K(o))$, so:

Lemma 2—The thermodynamic cost of a replacement computation is greater than the difference in the algorithmic information content between input and output

$$\delta S(i \rightarrow o) \gtrsim K(i) - K(o) \tag{10}$$

This simple equation gives the lower bound in terms of the individual complexities of the two strings. Moreover, it is the absolute limiting efficiency for the case when o^*, the minimal program for the output, is at hand. This follows from equation ($6b$) for the conditional information given o^*: $K(i|o^*) \simeq K(i) - K(o)$ without logarithmic corrections.

The disadvantage of equation (10) is that it is in general somewhat weaker than Theorem 1, which contained directly the size of the minimal conditional string. The reason for this slight difference touches some of the fascinating (and somewhat paradoxical) aspects of algorithmic information theory. Consider computation of a string s from its minimal program s^*. The algorithmic randomness of s and s^* is almost ($\sim O(1)$) identical: $K(s) = |s^*|$. Moreover, s^* is algorithmically random, as the reader verified above. Hence, the shortest program needed to obtain s^* cannot be shorter than $K(s)$. In addition, it cannot be longer than $K(s)$ by more than a few bits, as it must be shorter than the binary version of 'Print s^*'. Consequently, $K(s) \simeq K(s^*)$. Yet, s^* cannot be computed from s in a finite number of steps because of gödelian undecidability[16-18,22].

This comment should not be misunderstood. It is not difficult to suggest a concise and straightforward method of finding the shortest program that will generate string s within a pre-determined (arbitrarily large, but finite) number of steps N. One can test on C all of the conceivable programs (all the strings) shorter than the binary version of some program that obviously generates s. ($s_p \equiv$ Print s with the self-delimiting correction, if necessary, would do the job.) The list

0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, ...

of all binary strings shorter than $|s_p|$, which must include all candidate programs, is very large ($\sim 2^{|s_p|}$) but finite, and can be easily supplied by a concise algorithm.

Each of the candidate programs can be run for N steps in a finite time $\leqslant N \times 2^{|s_p|}$. Among the programs that have halted within N steps some may give an output s. The first of them in the lexicographic order, $\tilde{s}$, is easily identified, and if $\tilde{s} < s$ it can be used to accomplish a partial compression of the information. If none of the programs that have halted produces s, s_p is still the shortest known program to generate s, and information compression has not been found possible.

This algorithm can be implemented reversibly, so the thermodynamic cost of finding $\tilde{s}$ is, in principle, negligible. But this does not disprove the fundamental undecidability of the algorithmic randomness of binary strings[16-18,22], because the algorithm used to find $\tilde{s}$ is primitive recursive[23,24]; it must halt after a finite number of steps $\leqslant N \times 2^{|s_p|}$. Identity of $\tilde{s}$ with s^* could be established only in the $N \rightarrow \infty$ limit.

Indeed, if it were known beforehand which programs never halt, they could be eliminated from the lexicographic list. Now the shortest remaining program with output s would be guaranteed to yield the correct value of $K(s)$, and the minimal program s^* could be easily found. This clever plan cannot be used: Turing's halting theorem[25], the computer-theoretic equivalent of the Gödel's theorem, establishes that there is no way to distinguish halting from non-halting programs. Undecidability thus thwarts attempts to systematically derive, in a finite number of steps, the most efficient algorithm. Worse than that, the minimal thermodynamic cost of computation is given in terms of algorithmic complexities, which are themselves uncomputable. Not only is it impossible to find the best procedure for optimizing a computation, it is also impossible to find out how well one is doing.

Advance knowledge of the algorithmic complexity $K(s)$ of s would allow one to modify the search so that it would be guaranteed to halt after a finite number of steps. A short program for s, equal in length to the minimal s^*, would therefore become recursively computable. The uncomputability of the optimal efficiency is thus essential for the consistency of the above arguments, but it remains true that infinite computations can discover minimal programs.

The optimal algorithm could conceivably be found quickly. The halting theorem does not disallow the best solution being found after a finite number of iterations. One might also find the best solution by a lucky guess. It is only certain that one will never know how lucky one was.

Although the optimally efficient algorithm cannot be systematically discovered, it is always possible to limit the number of the erased bits to the size of the input[1,2,7]

$$\delta S(i \to o) \leq |i| \tag{11}$$

This 'guaranteed efficiency' can be easily justified[1]: it is always possible to perform the computation $i \to o$, $g(i, o)$, duplicate o, and use $g(i, o)$ to reproduce i. One is now left with o and i. Erasure of i can always be carried out at the cost of $|i|$ bits, so the thermodynamic cost can always be limited to no more than $|i|$ bits.

Although the δS guaranteed by equation (11) is somewhat trivial, there are instances when it coincides with the lower bound given by Theorem 1 and Lemma 2. For example, replacement of a random string with an equally long string of zeros cannot be carried out at a price lower than the guaranteed efficiency, equation (11).

These results complement and extend the proof[1] that a computation can be always performed reversibly[1-3,5-9] by establishing the minimal price of such reversibility for replacement computations. The price is given by the algorithmic complexity of the history of the computation given its output. Retaining $g(i, o)$ or i, leaves one the option to undo the computation at no thermodynamic expense. Erasing the history and the input incurs at least $\delta S(i \to o) \simeq |q^*_{i|o}| \geqslant K(i) - K(o) = |i^*| - |o^*|$ of entropy increase.

One might argue that the distinction between these two options—erasure and storage—is artificial: erasure can be accomplished by storing the relevant information in a physical system outside the computer memory. This point of view, which identifies increase of entropy with the erased information, must be further extended in a discussion of an update. Consider two files, s and t, each containing some information not present in the other $K(s) \neq K(s:t) \neq K(t)$. To compute t from s one must supply conditional information $K(t|s)$. A replacement computation requires erasure of the spurious information $K(s|t)$.

It is natural to count the thermodynamic cost of the supplied information with the opposite sign from the cost of the erased information. This leads to the net flow of algorithmic information between the computer and the rest of the Universe

$$\delta K = K(s|t) - K(t|s) \tag{12}$$

which (with the logarithmic correction terms) could be reduced to the right hand side of equation (10). But δK cannot immediately be identified with the minimal dissipation. The arguments of ref. 4 imply only that erasure of information is associated with an increase of entropy. δK would be a valid expression for the thermodynamic cost of an update only if the transfer of information from outśide the computer caused a corresponding decrease of entropy. This assumption may be not unreasonable, (acquisition of information can be regarded as a measurement, and measurements are thought to decrease statistical entropy) I shall not attempt to discuss conditions for its validity here. Moreover, neither of the two ingredients of δK is recursively computable, so the actual net number of transferred bits could be larger or smaller than δK. Thus, these arguments do not allow δK, equation (12), to be regarded as a minimal thermodynamic cost in the sense of equation (1) and Theorem 1.

I shall return to the thermodynamic significance of replacement computation later, in a discussion of measurements and the laws of thermodynamics. A more complete analysis of related issues can be found in ref. 10.

Algorithmic distance and the information metric

The extent of the update can be measured by the least amount of information involved in transforming s into t. Instead of counting the $K(s|t)$ and $K(t|s)$ with signs that depend on the direction of their flow between C and the rest of the Universe, as was done to obtain equation (12), one can add them to establish the total amount information needed to carry out the replacement. This sum defines the information distance between s and t

$$\Delta(s, t) \equiv K(s|t) + K(t|s) \tag{13a}$$

Algorithmic information distance can also be approximately expressed as

$$\Delta'(s, t) \equiv K(s) + K(t) - 2K(s:t) \tag{13b}$$

$$\equiv K(s, t) - K(s:t) \tag{13c}$$

$$= 2K(s, t) - K(s) - K(t) \tag{13d}$$

The equality between $\Delta(s, t)$ and $\Delta'(s, t)$ would be exact only if equation (6) were satisfied exactly. For statistical (Shannon) entropy, analogues of Δ and Δ' are identically equal, but logarithmic correction terms in the algorithmic version of equation (6) imply similar (logarithmic) corrections going from the original definition, equation (13a), to its alternatives, equations (13b-d). One could get rid of these corrections by using one of the alternative but less natural definitions of conditional algorithmic information (equation (6a) or (6b)).

The information-theoretic quantity $\Delta(s, t)$ defined by equation (13a) satisfies the three conditions required to allow its use as a distance; (i) positivity

$$\Delta(s, t) \geqslant 0 \tag{14}$$

with the equality holding only when $s = t$, (ii) symmetry

$$\Delta(s, t) = \Delta(t, s) \tag{15}$$

and (iii) the triangle inequality

$$\Delta(s, t) + \Delta(t, u) \geqslant \Delta(s, u) \tag{16}$$

which follows from noting that the conditional minimal program $q^*_{u|s}$ for computing u from s cannot be longer than the program that computes u from s using t (that is, the concatenated program $q^*_{u|t}q^*_{t|s}$). Consequently

$$K(u|t) + K(t|s) \geqslant K(u|s) \tag{17a}$$

The same argument holds for computation of s from u. Thus

$$K(t|u) + K(s|t) \geqslant K(s|u) \tag{17b}$$

Addition of these two inequalities with the help of equation (13a) yields the triangle inequality, equation (16). Hence:

Theorem 2—Information distance $\Delta(s, t)$ provides a metric for the space of binary sequences.

Another proof of the triangle inequality, more immediately applicable to the statistical (Shannon) version of the information distance, follows from

$$K(s, t, u) \geqslant K(s, u) \tag{18}$$

Both sides of equation (18) can be multiplied by two, and after adding $K(s, t) - K(s, t) + K(t, u) - K(t, u) = 0$ to the left-hand side and subtracting $(K(s) + K(u))$ from both sides one arrives at $K(s|t, u) + K(t|s) + K(t|u) + K(u|s, t) \geqslant K(s|u) + K(u|s)$. Now, ignoring the usual logarithmic correction terms and using the fact that the left-hand side of the above inequality cannot decrease if $K(s|t, u)$ and $K(u|s, t)$ are exchanged with $K(s|t)$ and $K(u|t)$, respectively, leads to the triangle inequality in the form in which Δ is expressed through conditional information, equation (13a).

All the steps of this second proof apply, without any logarithmic corrections, to Shannon's entropy H of classical ensembles, S, T and U. There is then no difference between Δ and Δ', and the distance given by any of the versions of equation (13) provides a metric for a collection of statistical ensembles characterized by their probability distributions. But it should be noted that the 'self-evident' inequality, equation (18), should not be taken for granted: it is not satisfied for statistical entropy H for systems that exhibit non-separable quantum correlations.

It is not difficult to show that $\Delta(s, t)$ satisfies the inequality $K(s, t) \geqslant \Delta(s, t) \geqslant |K(s) - K(t)|$. Obvious applications of the information distance include characterization of the relationship of the probability distributions. To some degree, mutual information has been already employed in that role[26]. Algorithmic-information distance is perhaps less immediately applicable to practical issues, but its fundamental implications may be far-reaching.

Complexity, Maxwell's demon and entropy

We have found that replacement computation can be done at a thermodynamic expense (an increase of entropy) greater than or equal to the difference in the algorithmic randomness between input and output. Entropy increase is thus a direct consequence of the decrease in the algorithmic information content. So far, this relation has been applied to set the thermodynamic limit on the reversibility of replacement computation. But there is also an inverse problem: what limits on statistical mechanics and thermodynamics can be found from algorithmic concepts? This is discussed in detail elsewhere[10]; here, I shall briefly outline the basic argument to provide physical motivation for interest in algorithmic randomness.

Consider a computerized engine consisting of a physical system in contact with a thermal reservoir at temperature T and controlled by universal computer C. The system is initially described by some density matrix ρ. Its standard ensemble entropy, measured in bits, is given by

$$H(\rho) = -\mathrm{Tr}\, \rho \log_2 \rho \tag{19}$$

The computer initiates a cycle of the engine by performing a dissipationless measurement[2,10,27] on the system. (In the context of quantum theory this implies that the measured observable must commute with the entropy, a condition which is always satisfied in the classical context.) Each of the outcomes, which are assumed to be mutually exclusive, is described by a density matrix ρ_k and occurs with probability p_k, so that $\rho = \Sigma_k p_k \rho_k$. The entropy of each outcome $H(\rho_k)$ is smaller than $H(\rho)$, and the average decrease of entropy is equal to

$$\langle \Delta H \rangle = H(\rho) - \sum_k p_k H(\rho_k) = -\sum_k p_k \log_2 p_k \tag{20}$$

The computer can convert this gain of information into useful work, at no thermodynamic expense, by constraining the system to the part of the phase space that coincides with ρ_k. Subsequently, useful work

$$\Delta W^+ = T(H(\rho) - H(\rho_k)) = T\Delta H_k \tag{21a}$$

could be extracted by allowing the constraint to relax in a slow isothermal fashion, so that ρ_k is transformed into ρ (Boltzmann's constant $k_B = 1$). The cycle can be repeated. If the process were truly cyclic, it would result in a violation of the second law by allowing an average gain $T\langle \Delta H \rangle$ of useful work per cycle. Szilard's engine provides a gedanken experiment realization of this idea[2,27-29].

Although the computerized "Maxwell's demon" can seemingly extract useful work, the second law of thermodynamics is not in danger. The cycle is never completed[2,10,27-29], as the memory of the computer does not return to its initial state: useless information about the outcomes of the past measurements piles up. The process uses computer memory as a zero-entropy reservoir. To make the process truly cyclic, the memory has to be periodically erased, and the cost of erasure must be subtracted to calculate the actual amount of useful work extracted. When the data is not random, a 'smart' erasing mechanism can take advantage of the regularities and compress the information before the erasure, whereas a 'dumb' erasing mechanism incapable of information compression would have to use $k_B T$ of work per bit, regularities notwithstanding. Data compression can be done separately, however, before the result is passed on to the 'dumb' erasing mechanism. One can thus separate reversible compression from the final irreversible erasure, and the cost of erasure can be in principle made as low as $k_B T$ times the algorithmic information content, rather than size in bits, of the record in question. This efficiency is attained by prior maximum (reversible) compression of the data to be erased (that is, in case of the computerized demon, of the conditional information $K(\rho_k|\rho)$). From Theorem 1 and Lemma 2, the thermodynamic cost of the most efficient erasure cannot be less than

$$\Delta W^- \simeq T\Delta K_k, \tag{21b}$$

where $\Delta K_k = [K(\rho) - K(\rho_k)] < 0$ is the difference between the algorithmic information content of the initial state and of the measurement outcome k.

Here I have used the version of the lower limit, $\delta S(\rho_k \to \rho) = K(\rho_k) - K(\rho)$, appropriate for a computation starting with the description of ρ_k as the input and ending with the.minimal description of ρ as the output. This shows that even if the demon was able to guess minimal programs, it would still not endanger the second law. Without minimal descriptions, the cost of erasure would increase and the efficiency would decrease. The second law is therefore satisfied for computer-operated engines independently of the issue of computability, but uncomputability of minimal strings will cause additional dissipation.

The increase in the potential to do useful work as a result of the information gain[10] must be measured not just by the decrease in the ensemble entropy ΔH_k, but by the sum of ΔH_k and ΔK_k, which gives the least number of bits needed to write down the outcome of the measurement. The net amount of work that can be gained following measurement with the outcome k is

$$\Delta W_k = \Delta W_k^+ + \Delta W_k^- = T(\Delta H_k + \Delta K_k) \tag{22}$$

Consequently, what determines the net amount of extractable work is the sum of the usual ensemble entropy, which specifies the missing information, plus the algorithmic randomness of the available information[10]

$$\mathcal{S} = H + K \tag{23}$$

Physical entropy $\mathcal{S}$ extends the applicability of arguments based on entropy beyond the usual ensemble setting by allowing discussion of definite outcomes of measurements and their consequences[10]. In the usual limit it is dominated by the ensemble Boltzmann-Gibbs-Shannon component H. When the state of the system is known sufficiently well, however, algorithmic randomness K supplies the key contribution. Moreover, only when this state is algorithmically simple can one extract a sizable amount of useful energy.

The second law formulated in terms of physical entropy, equation (23), holds even during the measurements because the average information gain $\langle \Delta H \rangle$ is offset in $\mathcal{S}$ by the least average size of the code $\langle \Delta K \rangle = \Sigma_k p_k \Delta K_k$ required to specify the outcome. Information and coding theory implies that[30,31]

$$\langle \Delta H \rangle \leqslant \langle \Delta K \rangle \tag{24}$$

So, recording the outcome of the measurement performed on a system in a thermodynamic equilibrium will, on the average, use up at least as many memory bits as the corresponding decrease of ignorance ΔH. Fluctuations with $|\Delta K_k| < \Delta H_k$ are possible, but they occur sufficiently infrequently that equation

(24), and the second law formulated in terms of $\mathcal{S}$ are satisfied for the ensemble. Erasure of information can be represented, in terms of the $\mathcal{S}$, as the act of storing the information from the memory of the computer in some external system. This point of view was mentioned earlier.

Theorem 1 establishes an interesting connection between algorithmic complexity and the thermodynamic depth recently proposed as a measure of complexity[32]. Thermodynamic depth of an object o is the amount of information discarded in course of the process that led to its formation. As defined, thermodynamic depth is a function of a specific history[32], but it is of interest to define a minimal thermodynamic depth that is solely a function of the initial conditions i and of the final result o. Theorem 1 demonstrates that such minimal thermodynamic depth is given by the size of the smallest program that can reconstruct i from o and is closely approximated by the difference of their algorithmic information contents. Further discussion of the applications of algorithmic concepts to complexity can be found in ref. 33.

Conclusions

The central idea in this paper is a computational counterpart of the second law of thermodynamics. A computation that replaces the input (program) file with the output causes an increase of entropy by no less than $\delta S(i \to o) \simeq |q^*_{i|o}| \geqslant K(i) - K(o)$ bits, where $q^*_{i|o}$ is the minimal conditional string, the shortest conceivable version of history of the logically irreversible transformation $i \to o$. This implies that a compression of information, replacement of a file s with a known program $\tilde{s}$ that computes s, can be accomplished reversibly, and could save as much as, but no more than $|s| - K(s)$ bits of memory.

The second law of algorithmic dynamics sets limits on the efficiency of Maxwell's demon: Even if the demon could acquire the relevant information about the state of the physical system without dissipation, and correctly guess the most concise method of representing its knowledge, the necessity to periodically erase from its memory the records of 'used up' past measurements implies that at least $\delta S \geqslant K(\text{measurement outcome}) - K$ (initial state) bits would have to be erased in each successful cycle. This logical irreversibility of replacement computations is enough to guarantee the safety of the second law of thermodynamics even in the presence of information gathering and utilizing systems.

The demon's inability to circumvent the second law by any combination involving measurements and computation is especially obvious when the process of energy extraction is discussed in terms of physical entropy defined as a sum of known disorder (measured by the algorithmic randomness) and remaining ignorance (measured by Shannon's entropy).

Smoluchowski[34], in his discussion of the famous 'trapdoor' (an automated version of Maxwell's demon) showed that "no automatic, permanently effective perpetual-motion machine" can violate the second law by taking advantage of statistical fluctuations, but he suggested that "such device might perhaps function if it were operated by intelligent beings".

I have demonstrated that the second law is safe even from 'intelligent beings', as long as their abilities to process information are subject to the same laws as these of universal Turing machines. (This is the so-called 'Church-Turing thesis', which can be succinctly restated as[24] "What is human-computable is also machine-computable".) Moreover, it is unlikely that a demon dealing with an equilibrium system will 'break even', because optimal efficiency of replacement computations can be attained only with the necessary minimal programs (s^*, $q^*_{i|o}$, for example). Unfortunately, minimal programs cannot be systematically computed—Turing's halting theorem implies that the information required to attain maximum efficiency can be secured only through an infinitely long computation. Thus, Gödel's undecidability can be regarded as an additional source of dissipation.

Algorithmic distance between binary strings can be defined as a difference between their joint and mutual complexities. It has all the properties required of a metric. Moreover, an analogous formula—the difference between Shannon's joint entropy and mutual information—can be used to define distances between statistical ensembles. □

Received 14 September 1988; accepted 2 August 1989.

1. Bennett, C. H. *IBM J. Res. Dev.* **17,** 525–532 (1973).
2. Bennett, C. H. *Int. J. theor. Phys.* **21,** 305–340 (1982).
3. Bennett, C. H. *IBM J. Res. Dev.* **32,** 16–25 (1988).
4. Landauer. R. *IBM J. Res. Dev.* **3,** 113–131 (1961).
5. Landauer, R. in *Signal Processing* (ed. Haykin, S.) 18–47 (Prentice-Hall, New York, 1989).
6. Bennett, C. H. & Landauer, R. *Scient. Am.* **253,** 48–56 (1985).
7. Fredkin, E. & Toffoli, T. *Int. J. theor. Phys.* **21,** 219–253 (1982).
8. Zurek, W. H. *Phys. Rev. Lett.* **53,** 391–394 (1984).
9. Bennett, C. H. *SIAM J. Comput.* **18,** 766–776 (1989).
10. Zurek, W. H. *Phys. Rev.* A (in the press).
11. Solomonoff, R. J. *Inf. Control* **7,** 1–22 (1964).
12. Kolmogorov, A. N. *Inf. Transmission* **1,** 3–11 (1965).
13. Kolmogorov, A. N. *IEEE Trans Inf. Theory* **14,** 662–664 (1968).
14. Kolmogorov, A. N. *Usp. mat. Nauk* **25,** 602–613 (1970).
15. Chaitin, G. J. *J. Ass. Comput. Mach.* **13,** 547–569 (1966).
16. Chaitin, G. J. *Scient. Am.* **232**(5), 47–52 (1975).
17. Chaitin, G. J. *IBM J. Res. Dev.* **21,** 350–359 (1977).
18. Chaitin, G. J. *Algorithmic Information Theory* (Cambridge University Press, 1987).
19. Gacs, P. *Soviet Math. Dokl.* **15,** 1477–1478 (1974).
20. Chaitin, G. J. *J. Ass. Comput. Mach.* **22,** 245–268 (1975).
21. Levin, L. A. *Soviet Math. Dokl.* **17,** 522–526 (1976).
22. Stewart, I. *Nature* **332,** 115–116 (1988).
23. Davies, M. *Computability and Undecidability* (McGraw-Hill, 1958).
24. Hofstadter, D. *Gödel, Escher, Bach* (Random House, New York, 1979).
25. Turing, A. M. *Proc. Lond. math Soc.* **42,** 230–265 (1936).
26. Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. *Numerical Recipes* Section 13.6 (Cambridge University Press, 1987).
27. Bennett, C. H. *Scient. Am.* **255**(11), 108–117 (1987).
28. *Szilard, L. Z. Phys.* **53,** 840–856 (1929).
29. Zurek, W. H. in *Frontiers of Nonequilibrium Statistical Mechanics,* (eds More, T. G. & Scully, M. O.) 151–161 (Plenum, New York, 1986).
30. Shannon, C. E. & Weaver, W. *The Mathematical Theory of Communication* (University of Illinois Press, 1949).
31. Hamming, R. W. *Coding and Information Theory* (Prentice-Hall, Englewood Cliffs, 1987).
32. Lloyd, S. & Pagels, H. *Ann. Phys.* **188,** 186–213 (1988).
33. (ed. Zurek, W. H.) *Proc. Santa Fe Inst. Workshop "Complexity, Entropy, and Physics of Information"* (Addison-Wesley, in the press).
34. Smoluchowski, M. in *Vortgäge über die Kinetische Theorie der materie und der Elektrizität* (Teubner, Leipzig, 1914).

ACKNOWLEDGEMENTS. I would like to thank C. H. Bennett for comments and suggestions, M. Gell-Mann and S. Lloyd for discussions, and the Santa Fe Institute and the Institute for Theoretical Physics in Santa Barbara for hospitality. This research was supported in part by the NSF and the National Aeronautics and Space Administration.