

The Relationship of Entropy to Message Size in Binary Symbol Strings

1.0 Introduction.

Early in the twentieth century, quantum physics demonstrated that the universe is random in its essential nature. More recently, Gregory Chaitin has shown that any set of symbols, such as the set of positive integers, can be used as input to generate an infinite set of random binary digits or bits. His conclusion is that there is infinitely more randomness than order in the universe. However, the opposite is also true. *With only a few exceptions, any string of bits, random or otherwise, can be shown to have an information content that will always be expressible using fewer bits than the original string.* This is another way of saying that, at the very least, order is coequal with randomness in its authority and effect.

In information theory, a string of symbols transmitted from one place to another is commonly referred to as a message. Entropy is a term used to represent the information content of a message. The calculation of entropy for a given message (i.e. string of symbols) represents the actual information content of that message as measured in bits. If the calculation of entropy for a given message results in a number value that is less than the number value (as measured in bits) of the original message, the message is considered to be compressible. Compressibility is a measure of randomness in a message. A purely random binary symbol string is considered uncompressible. As a corollary, a purely random binary symbol string should have entropy equal to the number of bits in the symbol string. If the entropy for a binary symbol string is less than the number of bits in the symbol string, the symbol string is not considered random.

When entropy is calculated by viewing a message as a collection of binary symbols with each symbol being one bit in length, the following assertions prove the entropy for any string of bits or symbols to be less than the number of bits in the string for all strings 5 bits or greater in length. Given an infinite number of possible messages, the only exceptions to these assertions are the binary symbol strings 01, 10, 0011, 0101, 0110, 1010, 1001 and 1100. This means that almost all messages, random binary data included, can be shown to have an information content less than the number of bits in the original string or message.

2.0 Definitions.

Entropy is defined as the minimum number of binary digits necessary to encode a message.

A message is defined as a string S of symbols.

Each symbol in S is a member or element of a set T .

The number of elements in T , also referred to as the base, is designated $|T|$ and is equal to the number of unique symbols occurring in S .

The entropy for a given symbol s in S is calculated as:

$$- \log |T| (s_n/S_m)$$

where

s_n equals the number of occurrences of s in S and
 S_m equals the total number of symbols in S

The entropy for a collection of like symbols s in S is calculated as:

$$- \log |T| (s_n/S_m)(s_n)$$

The entropy for S is calculated as the sum of the entropies for all collections of like symbols in S .

3.0 Assertions and Proofs.

The following assertions and proofs demonstrate the relationship between entropy and message size in binary symbols strings as being one in which the entropy is less than the number of binary symbols in the symbol string. In all cases, a binary symbol takes the form of and is equivalent to a binary digit.

3.1 Assertion A.

For any string of binary symbols of odd length greater than or equal to 3 that contains at least one of each type of binary symbol, the sum of the entropies of the two types of binary symbols in the string is less than the total number of binary symbols in the string.

3.2 Proof of Assertion A.

Given a string S of n binary symbols,

Where:

- (1) n is both odd and an integer greater than or equal to 3,
- (2) each instance of a binary symbol is equivalent to a binary digit or bit and
- (3) both types of symbols are represented in the string,

It follows that

- (4) the string S is divisible into two sets of binary symbols of like type and
- (5) one of the two sets of binary symbols will always contain at least one (1) more member than the remaining set of symbols.

Therefore, given (v), calculating the entropy for each set of binary symbols and adding the resulting values will always result in a number less than the total number of binary symbols in the string.

For example:

Where $n = 3$ the possible combinations of three bit symbol strings are:

000, 001, 010, 011, 100, 101, 110, 111

With the exception of 000 and 111 (whose members are of like type and the calculation of whose entropy, because it requires division by 0, is undefined), the calculation of entropy for the remaining combinations of binary symbols results in the following values: A sample calculation for symbol string 001 is:

$$(-\log((2/3), \text{base } 2)*2) + (-\log((1/3), \text{base } 2)*1) = 2.754888$$

In all cases the number of binary digits in the original message (3) is greater than the sum of the calculated entropies for the collections of like symbols contained in the original message (2.754888).

In summary,

Where:

$$n = (1, 2, 3, \dots),$$

A = Assertion A and

$$N = 2(n) + 1,$$

It follows that

A(3) is true and therefore A(N) is true.

Note A: In all cases in which a binary symbol string contains only one type of symbol, the calculation of entropy, because it results in division by 0, is undefined. One solution is to add a single symbol of a second type to the string. Entropy can then be calculated using |T| equal to 2, or base 2. For example, the binary string 000 can be converted to 0001. Using this method, all symbol strings of like type with a length of 4 or greater will have entropy less than the total number of symbols in the string. For example, the binary string 1111, after placing a symbol of a different type anywhere in the string (e.g. 01111), will have entropy of 3.60964.

3.3. Assertion B.

For any string of binary symbols of length n , where n is both even and a whole number greater than or equal to 6, by dividing the string into two substrings of odd length and applying Assertion A, the sum of the entropy of the symbols can be shown to be less than the total number of symbols.

3.4 Proof of Assertion B.

Given a string of binary symbols of length n , where n is both even and a whole number greater than or equal to 6,

It follows that

- (6) since the string is divisible into two substrings, one substring of length 3 and the remaining substring of length $n - 3$ and
- (7) since the sum of two odd numbers is always even and the sum of an odd and even number is always odd, the remaining substring must be an odd number of binary symbols in length.

Therefore, given Assertion A, calculating the entropy for each substring and adding the resulting values will always result in a number less than the total number of binary symbols in the original string.

In summary,

Where:

$$n = (3, 4, 5, \dots),$$

$$A = \text{Assertion A,}$$

$$B = \text{Assertion B and}$$

$$N = 2(n),$$

It follows that

$$A \text{ is true and } B(6) \text{ is true and therefore } B(N) \text{ is true.}$$

3.5. Assertion C.

For any string of binary symbols of infinite length, such as those representing irrational numbers, by dividing the string into two substrings, the first of which is of odd length greater than or equal to 5 containing at least one of each type of binary symbol (see Note A), and the second of which is the remaining symbols in the string, by applying Assertion A, the sum of the entropy of the symbols can be shown to be less than the total number of symbols in the string.

3.6 Proof of Assertion C.

Given a string of binary symbols of infinite length,

It follows that,

- (8) the string is divisible into two substrings, one substring of odd length 5 or greater (see Note A) and the remaining substring of infinite length.

Therefore, given Assertion A, by first calculating the entropy for the first substring and then successively adding the value of each succeeding binary digit (i.e., 1) in the second substring to the sum, the total will always result in a number less than the total number of binary symbols in the original string.

In summary, where:

$$n = (2, 3, 4, \dots),$$

$$A = \text{Assertion A},$$

$$N = 2(n) + 1$$

It follows that

A is true, A(N) is true and therefore C is true.