Complex and unpredictable binary strings

Tom Rochette <tom.rochette@coreteks.org>

August 30, 2025 — 861fb9d0

1 Question

What is the most complex sequence that can be made from a n-long binary string?

2 Answer

$0\ldots0$ and $1\ldots1$ are easily compressible, thus what is the most unpredictable sequence?

Let's proceed by induction.

2.1 1-long binary string

0 and 1 have a 50% to be selected in a 1-long binary string, thus they are both the most complex examples of a 1-long binary string.

2.2 2-long binary string

string	pattern
00	repeating
01	alternating
10	alternating
11	repeating

With two bits, we introduce two patterns: repeating and alternating. Repeating means that the bits are repeated for the complete string. Alternating means that the bits are alternating between 0 and 1 given a certain periodicity.

From our initial observation we suggested that the repeating pattern was likely to be the simplest because it consists of defining the symbol (0 or 1) that is repeated and the number of repetitions. We can say that this function has two parameters: length and symbol.

The same can be said about the alternating pattern. If you provide the initial symbol and the length of the pattern, you've defined its parameters. We can say that this function also has two parameters: length and starting symbol.

Using those two descriptions, we can observe that the alternating pattern is in fact a special case of the repeating pattern, where the symbol is 2 bits instead of one.

$00 = \text{repeating}(2, 0)$
$01 = \text{alternating}(2, 0) = \text{repeating}(1, 01)$
$10 = \text{alternating}(2, 1) = \text{repeating}(1, 10)$
$11 = \text{repeating}(2, 1)$

2.3   3-long binary string

string	pattern
000	repeating
001	complex, or alternating, periodicity=2
010	alternating
011	complex, or alternating, periodicity=1,2
100	complex, or alternating, periodicity=1,2
101	alternating
110	complex, or alternating, periodicity=2
111	repeating

001, 011, 100, 110 can be described as alternating and repeating, which is more complex than the previous examples as it requires the composition of two operations.

4 complex examples (001, 011, 100, 110) have been introduced at this stage.

2.4   4-long binary string

string	pattern
0000	repeating
0001	complex, or fill & flip
0010	complex, or fill & flip
0011	alternating, periodicity=2
0100	complex, or fill & flip
0101	alternating, periodicity=1
0110	mirror
0111	complex, or fill & flip
1000	complex, or fill & flip
1001	mirror
1010	alternating, periodicity=1
1011	complex, or fill & flip
1100	alternating, periodicity=2
1101	complex, or fill & flip
1110	complex, or fill & flip
1111	repeating

Here we first observe the presence of the alternating pattern with a periodicity of 2. We also observe the first case of a mirror pattern with 0110 and 1001. It could be claimed that 010 and 101 is the uneven equivalent of the mirror pattern for 3-long binary strings. The remaining "complex" examples are again displaying alternating and repeating patterns at the lower level.

One observation is that the complex patterns appear to be the variants where 1 bit is the opposite of the remaining bits, the only special case so far being for the 3-long binary string where the pattern is considered alternating. Instead of considering it a complex pattern, it could be described as the operation "fill X bits with 0/1, then flip by Y".

From those few inductive iterations, we can see that the first and last string are always the repeating pattern. We also observe that the first half of the strings exhibit the same pattern of the second half of the strings, which is expected due to the binary nature of the string.

At this point there is still not enough information to be able to derive an answer to the original question, given that the strings can be described using 4 patterns: repeating, alternating, mirror, and fill & flip.

8 complex examples are found at this stage.

2.5 5-long binary string

string	pattern
00000	repeating
00001	fill & flip
00010	fill & flip
00011	alternating, periodicity=3,2
00100	fill & flip
00101	complex, or 0 and alternating, periodicity=1
00110	complex, or alternating, periodicity=2
00111	alternating, periodicity=2,3
01000	fill & flip
01001	complex, or 0 and mirror
01010	alternating
01011	complex, or 0 and fill & flip
01100	complex, or alternating, periodicity=1,2,2
01101	complex, or 0 and fill & flip
01110	complex, or alternating, periodicity=1,3,1
01111	fill & flip

The second half of this table was not written down due to the previous observation that the patterns are the same, simply inverted.

At this stage, the complex patterns that emerge are difficult to express using the prior language. Composition needs to be used, e.g. 00101 can be described as 00 then 101.

14 complex examples are found at this stage.

At this point in my study, I would summarize as follows:

- Repeating, alternating and fill & flip are low complexity. For a n-long binary string, this implies 2 repeating patterns, 2n fill & flip patterns, 2n alternating patterns of periodicity=1.
- Complexity appears to emerge from alternating patterns with varying periodicity. As we're working with longer and longer strings, the periodic patterns will start to become periodic themselves (I believe this is leading to fractal).
- The fill & flip pattern appears to be able to cover most of the complex cases that appear as the length of the binary string increases.
- Some of the patterns can be expressed as a few alternatives, for example 01110 as alternating, periodicity=1,3,1, but also as 0 and fill & flip (or even an "odd" mirror). My intuition would lead me to think that those cases that can be expressed in multiple ways are less complex.
- It wasn't done during this analysis, but it may be possible to apply operations overlayed on one another, such as two alternating patterns, one with a periodicity of 2 and one with a periodicity of 3, and use logic operators such as AND/OR/XOR (and their negative variants) to recreate the pattern
- The sequence of complex examples seems to be as follows: 0, 4, 8, 14, . . .
- It would be interesting to look at those strings from a logic circuit point of view. For example, 00000 can be produced from a single signal (0) routed to 5 exits, 00001 would be a single signal (0) routed to 4 exits and to one not gate which is then routed to an exit. Assuming the input signal is always 0, producing 11111 would require 1 not gate.