We shall define boolean strings $\mathbb{B}_{\mathbb{Z}}$ as:

$$\mathbb{B} = \{False, True\}$$

$$B_1 = \{b_0 : b_0 \in \mathbb{B}\}$$

$$B_2 = \{b_0 b_1 : b_0 \in \mathbb{B} \land b_1 \in \mathbb{B}\}$$

$$...$$

$$B_n = \{b_0 b_1 ... b_{n-1} : b_0 ... b_{n-1} \in \mathbb{B}\}$$

$$\mathbb{B}_{\mathbb{Z}} = \{B_1, B_2, ..., B_\infty\}$$

Pretty sure I don't need $\mathbb{B}_{\mathbb{Z}}$, but it's pretty. I think B_n is what I really want.

Thus, a boolean string is a non-terminating, positionally-ordered set of $\{False, True\}$ taking the form *eg* 00100011000....

While strings do not terminate, they may have an infinite number of trailing $False$ entries. Such strings can be simply represented in a computer's memory, but the full expansion of $b \in \mathbb{B}_{\mathbb{Z}}$ cannot.

This definition removes much of the utility of boolean strings in computers, but does allow them to fully represent $\mathbb{R}$, for example.

We shall attempt to show that, through techniques such as Dedekind cuts, one can embed all spaces within $\mathbb{B}_{\mathbb{Z}}$.