Author: Keith Stellyes

Proof that `k << 1 = k * 2` where `<<` is the bitshift left operation and k is an integer represented as a series of bits.

The bit string ordered set B is equal to the integer $I(B)$ where:

$$I(B) = \sum_{k=0}^{|B|-1} 2^k \times B_k$$

For convenience, we define a function, $bi(b, k)$ where $b = 0 \oplus b = 1$ where:

$$bi(b, k) = 2^k \times b$$

Which lets us re-express $I(B)$ as:

$$I(B) = \sum_{k=0}^{|B|-1} bi(B_k, k)$$

The bit string representing the left bitshift is defined by the function $lshift(B)$ where:

$$lshift(B)_k = \begin{cases} 0 & \text{if } k \leq 0 \\ B_{k-1} & \text{if } k > 0 \end{cases}$$

We want to prove the following:

$$I(lshift(B)) = 2 \times I(B) =$$

$$\sum_{k=0}^{|B|-1} bi(B_k, k) = 2 \times \sum_{k=0}^{|B|-1} bi(lshift(B)_k, k)$$

For $k > 0$

$$bi(m, n+1) = 2 \times bi(m, n)$$

$$2 \times 2^n \times m = 2^{n+1} \times m$$

$k + 1$ to maintain the magnitude...:

$$bi(lshift(B)_k, k) = bi(B_{k-1}, k+1) = 2 \times bi(B_{k-1}, k)$$

Finally...

$$\sum_{k=0}^{|B|-1} bi(B_k, k) = 2 \times \sum_{k=0}^{|B|-1} bi(B_{k-1}, k) =$$

$$2 \times \sum_{k=0}^{|B|-1} bi(leftshift(B_k), k)$$