Computability of S

Daniel Filan, daniel.filan@anu.edu.au

We define S the same way as in Schmidhuber's paper, except having the definition on monotone Turing machines. It is clear that S is lower semicomputable, using the FAST algorithm. We show that $S(x)$ is computable for all x by induction.

The base case is the string $x = \epsilon$. Since $S(\epsilon) = 1$ by definition, it is computable.

Now, suppose all strings of length n are computable. Then, let x be a string of length n and $y = x0$. We will show that $S(x) - S(x0) - S(x1)$ is lower semicomputable. Then, since $S(x1)$ is lower semicomputable and $S(x)$ is computable,

$$S(x0) = -(S(x) - S(x0) - S(x1)) + S(x) - S(x1)$$

is upper semicomputable, meaning that it is computable. Therefore, all strings of length $n+1$ are computable.

To show that $S(x) - S(x0) - S(x1)$ is lower semicomputable, we note that

$$S(x) - S(x0) - S(x1) = \sum_{i=1}^{\infty} 2^{-i} \left(\sum_{p \to_i x} 2^{-|p|} - \sum_{\substack{p \to_i x0 \\ p \to_i x1}} 2^{-|p|} \right)$$

To lower semicompute this quantity, we can just run the FAST algorithm for k steps, which will give us the first k terms in this sum. Eventually this will converge to the required quantity, and each term in brackets is positive (since $p \to_i x0$ or $p \to_i x1$ implies $p \to_i x$), meaning that this is a lower semicomputation.