Stopping time complexity

A.Shen

March 13, 2017

Consider a bit string x written on the input one-directional tape of some Turing machine. We want the machine to stop reading the tape exactly when x is read. How much information should be communicated to this machine? We may call this amount "stopping time complexity" of x.

This quantity (in the context of prediction theory) was considered by Vovk and Pavlovic (see https://arxiv.org/abs/1603.04283), and we try to perform more systematic analysis of it in the language of Kolmogorov complexity (joint work with M. Andreev, G. Posobin).

One can consider the plain version of stopping time complexity (minimal plain complexity of a Turing machine that stops at x). It turns out to be equivalent to monotone-conditional complexity $C(x|x*)$ where the condition x is considered as a prefix of the string. There is also a quantitative characterization as a minimal upper semicomputable function such that on every path there is at most 2^n points where the function drops below n.

We show also that one should be careful: for the general case of $C(x|y*)$ we should consider monotone (prefix-stable), not prefix-free functions of y.

A similar theory can be constructed for prefix versions of stopping time complexity. We answer the question asked by Vovk–Pavlovic and show that the minimal prefix complexity of a program stopping at x, the quantity $K(x|x*)$ and the logarithm of stopping time semimeasure, introduced by Vovk and Pavlovic, are all different. Also we show that the stopping time semimeasure has a natural probabilistic interpretation while for the general case $m(x|y*)$ the natural interpretation is no longer valid.