PAUL BEAME, University of Washington

Making Branching Programs Oblivious Requires Superlogarithmic Overhead

An algorithm is oblivious if and only if its sequence of operations is determined independent of its input. Certain models of computation, such as circuits or straight-line programs are inherently oblivious. However, many computing models such as Turing machines and random access machines (RAMs) are not. Fairly efficient simulations of these general models by their more restricted oblivious variants are known.

We show that either a superlogarithmic increase in time or a large increase in space is necessary for any randomized oblivious simulation of general RAMs. In particular we show a $T = \Omega(nlog(n/S)loglog(n/S))$ lower bound for any randomized oblivious RAM determining out-degree 1 graph reachability which has an easy linear time logspace non-oblivious algorithm. This is the largest time-space tradeoff lower bound known for any randomized non-uniform model. We show similar, though incomparable, results for Boolean branching programs.

Joint work with Widad Machmouchi