Tung Anh Vu
tung@kam.mff.cuni.cz
presented:

Derandomization from Space-Time Tradeoffs

Oliver Korten
https://eccc.weizmann.ac.il/report/2022/025/

Definitions

- A *RAM machine* is a Turing machine which has an oracle to an infinite "random access memory". The oracle has two tapes, an *addressing tape* and an *auxiliary tape*. We refer to this pair of tapes as *linear tapes*. The random access memory stores a bit in each cell. We can only[1] update and read a cell of the memory. To read a cell, we write the address, that is logarithmically many bits long in the value of the address, on the addressing tape and the contents appear under the head of the auxiliary tape. Update is done analogously.
- Let $T, S, G : \mathbb{N} \to \mathbb{N}$ be functions. $\mathsf{1\text{-}NTISPG}[T(n), \mathrm{S}(n), \mathrm{G}(n)]$ is the set of languages decidable by 1-tape nondeterministic Turing machines which on an input of length n on every computation path spends time $\mathcal{O}(T(n))$, space $\mathcal{O}(S(n))$, and makes $\mathcal{O}(G(n))$ nondeterministic guesses. If we omit "N","G", and the function G, then we require the machine to be deterministic. If we omit "SP", and the function S, then we do not restrict the space usage. If we omit "1-", then we do not limit the number of work tapes.
- Let $C : \{0,1\}^n \to \{0,1\}^m, D : \{0,1\}^m \to \{0,1\}^n$, where $m < n$; we call such a C a "*compressor*", such a D a "*decompressor*", and collectively we will refer to (C, D) as a "*compression scheme.*" We may occasionally refer to m as *code length* and to n as *message length.* By the pigeonhole principle, we know that there must exist some $x \in \{0,1\}^n$ such that $D(C(x)) \neq x$; we call such an x "*incompressible*" with respect to the scheme (C, D). In the task of finding incompressible strings, we may without loss of generality assume that $2m = n$.[2]
- We call C a "*proper compressor*" for D if C, D[3] satisfy the following for all $x \in \{0,1\}^n$: if there exists an $y \in \{0,1\}^m$ such that $D(y) = x$, then $D(C(x)) = x$.

Theorem

- Let $\mathcal{C}, \mathcal{D} = \{C_n\}_{n\in\mathbb{N}}, \{D_n\}_{n\in\mathbb{N}}$ be a uniform compression scheme. Then one of the following must hold.
  1. There is a polynomial algorithm with oracle access to $\mathcal{C}, \mathcal{D}$ which, for infinitely many n outputs an incompressible string for C_n, D_n on input 1^n.
  2. For every exponential time bound T[4], every language $L \in \mathsf{RAM\text{-}TIME}[T(n)]$, and every $\varepsilon > 0$, there is a 1-tape Turing machine with oracle access to $\mathcal{C}, \mathcal{D}$ which decides L in time $T(n)^{1+\varepsilon}$, uses space at most $T(n)^{\varepsilon}$, and makes oracle calls of length at most $T(n)^{\varepsilon}$.[5]

Main results

Hypothesis: For some exponential time bound $T := T(n)$ and some $\varepsilon > 0$	**Implication:**
$\mathsf{RAM\text{-}TIME}[T] \subsetneq \mathsf{1\text{-}TISP}[T^{1+\varepsilon}, T^{\varepsilon}]$	If the compression problem has a polynomial time algorithm, then incompressible strings can be constructed in polynomial time.
$\mathsf{NTIME}[T] \subsetneq \mathsf{1\text{-}NTISPG}[\mathsf{T}^{1+\varepsilon}, \mathsf{T}^{\varepsilon}, \mathsf{T}^{\varepsilon}]$	Incompressible strings can be constructed in polynomial time with an NP-oracle. In particular, $\mathsf{E}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}[2^n/2n]$.
$\mathsf{RAM\text{-}TIME}[T] \subsetneq \mathsf{1\text{-}NTISPG}[T^{1+\varepsilon}, T^{\varepsilon}, T^{\varepsilon}]$	Incompressible strings can be constructed in polynomial time with an oracle for the compression problem.

[1] This is a simplification to avoid defining/restricting addition/multiplication/XOR/... of numbers.
[2] And we shall do so.
[3] We start omitting parentheses from now.
[4] There exist constants α, β such that $2^{\beta n} \leq T(n) \leq 2^{\alpha n}$ for every $n \in \mathbb{N}$.
[5] First row of the table below follows from this theorem.