Some Games on Turing Machines and Power from Random Strings

Alexey Milovanov

24.07.2023 / Batumi, CiE

What is "Power from Random Strings"

- Denote by $\mathrm{K}_U(x)$ the Kolmogorov complexity of x with respect to a universal decompressor U—the minimal length of a program that outputs x.
- Denote by R_U the oracle function that outputs $\mathrm{K}_U(x)$ on input x.
- In the sense of computational complexity, how strong is this oracle?
- Consider, for example, $\mathrm{P}^R := \bigcap_U \mathrm{P}^{R_U}$. What are upper and lower bounds for this class? The same questions are arouses for $\mathrm{BPP}^R, \mathrm{P}^R_{\mathrm{tt}}, \ldots$
- Partial answers to this questions were done in works Allender, Lempp, Hirahara, Fortnow,. . . with co-authors.

Motivation: why should we research P^R, BPP^R?..

- This examination raises interesting questions in areas such as derandomization and interactive proofs within computational complexity.
- This research can help to understand the complexity of Minimal Circuit Size Problem (MCSP):
  given the truth-table of a Boolean function f and a number k, does there exist a Boolean circuit of size at most k computing f?
- Open problem: Is MCSP NP-complete?
- MSCP is close to the following notion in resource-bounded Kolmogorov complexity KT.
  $KT(x) := \min\{|p| + t : \forall i \leq |x| + 1, \forall b \in \{0, 1, *\} :$
  $U(p, i, b) = 1 \iff b = x_i$ and U works in time $t\}$.
- Usually, problems in plain Kolmogorov complexity are easier than the same problems in resource-bounded Kolmogorov complexity.

Lower bounds I

Theorem (2005;Allender, Buhrman, Koucky, van Melkebeek, Ronneburger)

- $\mathrm{P}^R = \mathrm{BPP}^R$;
- $\mathrm{PSPACE} \subseteq \mathrm{P}^R$.

Idea of the proof: oracle R allows to distinguish random strings from pseudo-random. In fact authors do not need oracle function, they used $\{x | \mathrm{K}(x) > \frac{|x|}{2}\}$.

Theorem (2006; Allender, Buhrman, Koucky)

- $\mathrm{H} \in \mathrm{P/poly}^R$;
- $NEXP \subseteq \mathrm{NP}^R$.

Here H is the Halting problem.

The proofs used interactive proofs and KT.

Lower bounds II

Theorem (2010, Buhrman, Fortnow, Koucky, Loff)

$\mathrm{BPP} \subseteq \mathrm{P}^{R}_{tt}$. Here tt means truth-table reductions.

Theorem (2020, Hirahara)

- $\mathrm{EXP}^{\mathrm{NP}} \subseteq \mathrm{P}^{R}$;
- $\mathrm{NEXP} \subseteq \mathrm{BPP}^{R}_{tt}$.

The proofs use local-decoding codes, pseudo-random generators, interactive proofs.
These proofs use oracle-function R (not just the set of random strings). The results are still valid if the oracle-function gives the value with logarithmic precision.

Upper bounds

Theorem (2014, M. Cai, R. Downey, R. Epstein, S. Lempp, and J. Miller)

Classes P^R and NP^R contains only decidable languages.
Here R is the oracle function for plain or prefix complexity.

Theorem (2013, Allender, Friedman, Gasarch)

- $\mathrm{P}^R_{tt} \subseteq \mathrm{PSPACE}$;
- $\mathrm{P}^R, \mathrm{NP}^R \subseteq \mathrm{EXPSPACE}$.
  Here R is the oracle function for prefix complexity.

The idea is to use the main theorem about prefix complexity—its connection with universal semi-measures. The statements above are reduced to some game on Turing machines. The authors show that defining a winning player in such games belongs to PSPACE/EXPSPACE. This allows (by some reasons) to prove the upper bounds.

Game for tt-reduction

- Let M be a polynomial time Turing machine that has access to oracle O. This machine implements tt-reduction, i.e. on inputs of length n the machine M asks poly(n) questions to oracle O. After this machine outputs 1 or 0.
- Initially O is empty. Let x be some string. Consider the following game. The goal of Alice is $M^O(x) = 1$, the goal of Bob is $M^O(x) = 0$. Alice and Bob can add strings to O for some cost. Specifically, adding string y costs $v(y)$ for some function v.
- The players take turns, but they can skip their turn if the current value $M^O(x)$ is acceptable for them. Initially Alice has c_A dollars, Bob has c_B dollars.
- The problem is to decide the winner by (x, c_A, c_B).

New results

Theorem

$\mathrm{BPP}_{tt}^{R} \subseteq \mathrm{AEXP}^{\mathrm{poly}}$.
Here $\mathrm{AEXP}^{\mathrm{poly}}$ is the class of languages decidable in exponential time by an alternating Turing machine that switches from an existential to a universal state or vice versa at most polynomial times.

Theorem (Informal)

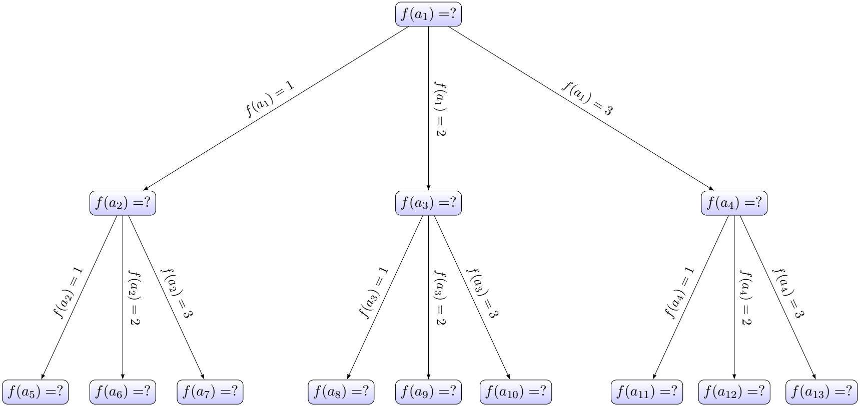
The games that appears in the proof of the following statements $\mathrm{P}_{tt}^{R} \subseteq \mathrm{PSPACE}$ and $\mathrm{P}^{R}, \mathrm{NP}^{R} \subseteq \mathrm{EXPSPACE}$ are PSPACE- and EXPSPACE-complete.

This means that current methods can not provide better upper bounds for P^{R}, NP^{R} and P_{tt}^{R} than known.

Sub-adaptive reductions

Definition

A machine M with an oracle access is called sub-adaptive if for every input all nodes in the reduction tree (i.e., all the oracle queries) are different.



Bounds for sub-adaptive reductions

Theorem

$\mathrm{P}_{sa}^{R} \subseteq \mathrm{EXP}$.

Recall that for adaptive reduction the upper bound is EXPSPACE, for tt-reduction the upper bound is PSPACE. We can consider a “mixture” of tt-reduction and sub-adaptive reduction and also get EXP as an upper bound.

Open problem

Is there any non-trivial lower bounds for P_{sa}^{R}?

გმადლობთ ყურადღებისთვის!