Allender's conjecture and its impacts on meta-complexity

Shuichi Hirahara

National Institute of Informatics, Japan

Eric Allender's day @Simons Institute

Feb 13, 2023

CiE 2012 (Computability in Europe)

Curiouser and Curiouser: The Link between Incompressibility and Complexity

Eric Allender

- Eric's invited paper at CiE 2012

Conjecture 1. NEXP = NP^R.

Conjecture 2. BPP is the class of problems non-adaptively polynomial-time reducible to R.

More specifically, let R denote the set of Kolmogorov-random strings. Let BPP denote the class of problems that can be solved with negligible error by probabilistic polynomial-time computations, and let NEXP denote the class of problems solvable in nondeterministic exponential time.

Conjecture 1. NEXP = NP^R.

Conjecture 2. BPP is the class of problems non-adaptively polynomial-time reducible to R.

These conjectures are not only audacious; they are obviously false! R is not a decidable set, and thus it is absurd to suggest that the class of problems reducible to it constitutes a complexity class.

The absurdity fades if, for example, we interpret "NP^R" to be "the class of problems that are NP-Turing reducible to R, no matter which universal machine we use in defining Kolmogorov complexity". The lecture will survey the body of work (some of it quite recent) that suggests that, when interpreted properly, the conjectures may actually be true.

- My first paper was about this conjecture.
  (MFCS 2014, joint work with Akitoshi Kawamura)

- Today: Recent progress and its impacts

"Randomness" in Complexity and Computability

- Complexity Theory

$\mathrm{BPP} :=$ {The class of problems solvable efficiently with **randomness**}.

Example: Polynomial Identity Testing $\in$ BPP

Try to understand BPP through the lens of computability!

- Computability Theory

$R_{\mathrm{K}_U} :=$ {The set of Kolmogorov-**random** strings } is not computable.

Example: $0000000000 \notin R_{\mathrm{K}_U}$, $10110100111 \in R_{\mathrm{K}_U}$.

Kolmogorov complexity

- The *Kolmogorov complexity* $\mathrm{K}_U(x)$ of a finite string x is the "shortest program" size to print x.

Examples

- $000000000000000000000000000000$ $(= 0^n)$ can be compressed into a program "$\mathrm{print}\ '0' \times n$".

$$\mathrm{K}_U(0^n) = \log n + O(1).$$

- $r \coloneqq 101011011001101010001010 10$ chosen uniformly and randomly.

$\mathrm{K}_U(r) \approx |r|$ with high probability.

Formal Definition and Universality

- We need to fix an "interpreter" U.

Definition: Kolmogorov complexity

The *Kolmogorov complexity* of a finite string $x \in \{0,1\}^*$ with respect to a Turing machine U is defined as

$$\mathrm{K}_U(x) \coloneqq \min\{\, |d| : U \text{ outputs } x \text{ on input } d \,\}.$$

- U can be chosen so that K_U is smallest up to an additive $O(1)$ term.

Definition: Universality of U

A machine U is said to be *universal* if for any M and any x,

$$\mathrm{K}_U(x) \leq \mathrm{K}_M(x) + O(1).$$

Kolmogorov-Randomness

- Kolmogorov complexity gives rise to the notion of randomness.

Definition: Kolmogorov-Randomness

- A string x is said to be Kolmogorov-random if $\mathrm{K}_U(x) \geq |x|$.
- Define $R_{\mathrm{K}_U} := \{\, x \mid \mathrm{K}_U(x) \geq |x| \,\}$.

Examples

- $0000000000000000000000000000 \notin R_{\mathrm{K}_U}$: not random
- $101011011001101010001010 \in R_{\mathrm{K}_U}$: random

Complexity vs. Computability Theory

- Complexity Theory

$\mathrm{BPP} :=$ {The class of problems solvable efficiently with randomness}.

Bridge the gap via an efficient reduction to R_{K_U}.

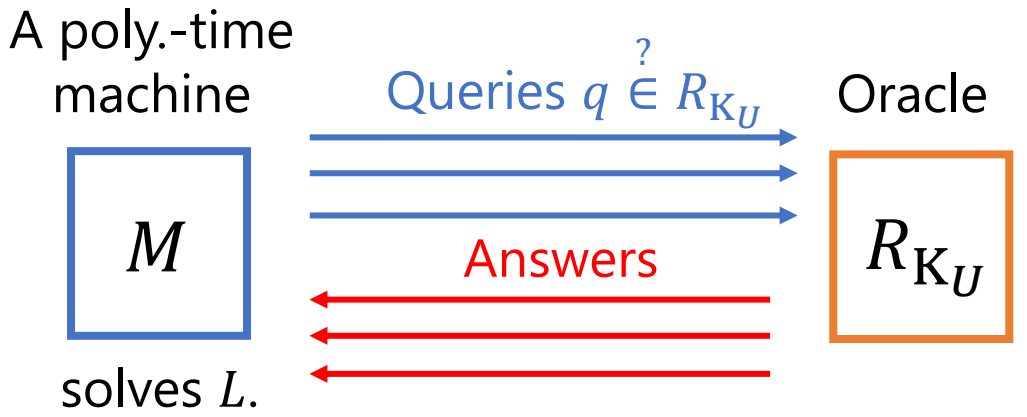
- Computability Theory

$R_{\mathrm{K}_U} :=$ {The set of Kolmogorov-random strings} is **not computable**.

Reductions to Kolmogorov-random strings

- What can be solved efficiently by nonadaptively asking whether $q \in R_{\mathrm{K}_U}$ or not? (no matter what U is)

$$\mathrm{P}_{\|}^{R_{\mathrm{K}_U}} = \{ L \mid$$



$$\}$$

[Buhrman, Fortnow, Koucký & Loff (2010)]

$$\mathrm{BPP} \subseteq \mathrm{P}_{\|}^{R_{\mathrm{K}_U}} \subseteq \mathrm{PSPACE}$$

[Allender, Friedman & Gasarch (2013)]

$$\mathrm{BPP} \quad \subseteq \quad \bigcap_{U \text{ universal}} \mathrm{P}_{\|}^{R_{\mathrm{K}U}} \quad \subseteq \quad \mathrm{PSPACE}$$

Allender's Conjecture (2012)

$$\mathrm{BPP} = \bigcap_{U} \mathrm{P}_{\|}^{R_{\mathrm{K}U}}.$$

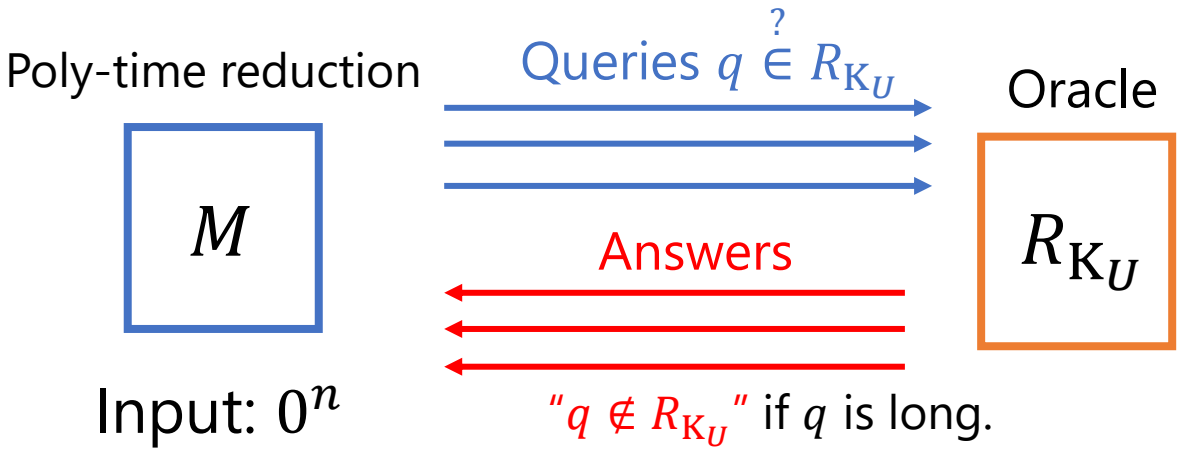
Motivation:

- $\mathrm{MCSP}^{\mathrm{HALT}} \approx R_{\mathrm{K}_U}$. [Allender, Buhrman, Koucky, van Melkebeek, Ronneburger'06]
  (Minimum HALT-Oralce Circuit Size Problem)
- An approach towards the $\mathrm{P} = \mathrm{BPP}$ conjecture?

$$\bigcap_{U} \mathrm{P}_{\mathrm{dtt}}^{R_{\mathrm{K}U}} = \mathrm{P}$$ [Allender, Buhrman, Koucky 2006]

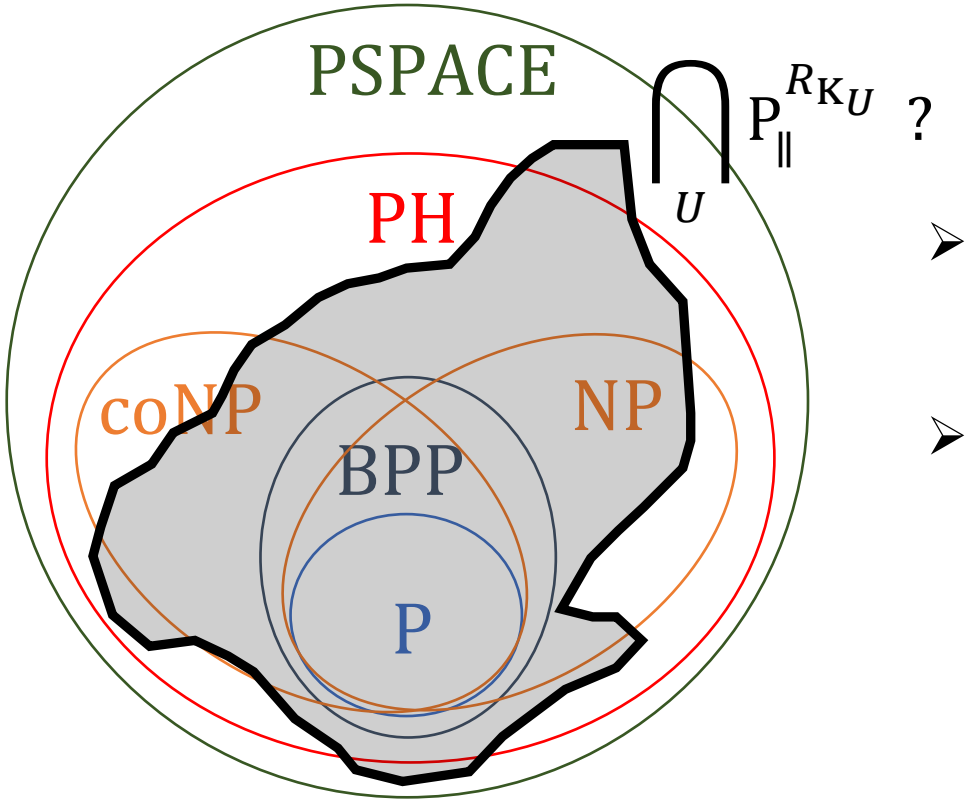
(dtt: disjunctive truth table reductions)

The Intuition behind Allender's Conjecture



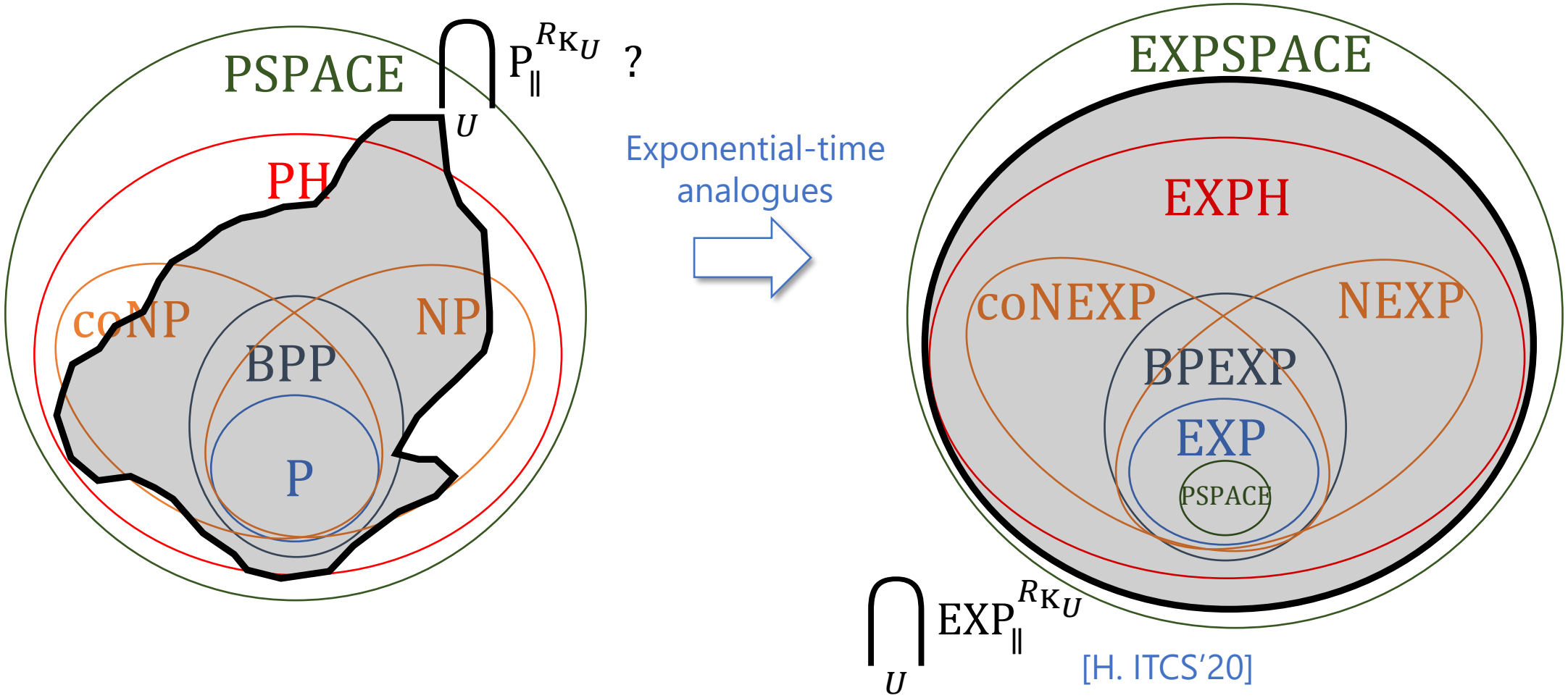
- $\mathrm{K}_U(q) = O(\log n)$ for any query q.
- If $|q| > O(\log n)$, then $q \notin R_{\mathrm{K}_U}$ (i.e. the answer is "No").
- On input 0^n, a polynomial-time nonadaptive reduction cannot make use of any query of length $> O(\log n)$.

Complexity Classes



- Let us focus on input 0^n.
- This is essentially equivalent to considering an exponential-time analogue.

Exponential-Time Classes



Allender's conjecture is false (unless EXPH collapses)

Theorem [H. ITCS'20]

For every universal Turing machine U,

$$\mathrm{EXPH} \subseteq \mathrm{EXP}_{\|}^{R_{\mathrm{K}_U}}.$$

A padding argument

Corollary

Allender's conjecture is false (i.e., $\mathrm{BPP} \neq \bigcap_U \mathrm{P}_{\|}^{R_{\mathrm{K}_U}}$) unless $\mathrm{EXPH} \subseteq \mathrm{BPEXP}$.

Other conjectures

[Allender, Buhrman, Koucký, van Melkebeek, and Ronneburger (2006)]

- $\mathrm{PSPACE} \subseteq \bigcap_U \mathrm{P}^{R_{\mathrm{K}U}} \subseteq \mathrm{EXPSPACE}.$

[Allender, Friedman, Gasarch (2013)]

- $\mathrm{NEXP} \subseteq \bigcap_U \mathrm{NP}^{R_{\mathrm{K}U}} \subseteq \mathrm{EXPSPACE}.$

[Allender, Buhrman, Koucký (2006)]

Hypothesis [Allender, Buhrman, Koucký (2006)]

$$\mathrm{PSPACE} = \bigcap_U \mathrm{P}^{R_{\mathrm{K}U}}.$$

Conjecture [Allender (2012)]

$$\mathrm{NEXP} = \bigcap_U \mathrm{NP}^{R_{\mathrm{K}U}}.$$

Unexpected Power of Random Strings

Main Theorem [H. STOC'20]

$$\mathrm{EXP}^{\mathrm{NP}} \subseteq \mathrm{P}^{R_{\mathrm{K}_U}}$$

- It follows that $\mathrm{PSPACE} \neq \bigcap_U \mathrm{P}^{R_{\mathrm{K}_U}}$ and $\mathrm{NEXP} \neq \bigcap_U \mathrm{NP}^{R_{\mathrm{K}_U}}$ unless $\mathrm{EXP}^{\mathrm{NP}} \subseteq \mathrm{NEXP}$.

$$\mathrm{EXP}^{\mathrm{NP}} \subseteq \bigcap_U \mathrm{P}^{R_{\mathrm{K}_U}} \subseteq \mathrm{EXPSPACE}$$

NP-hardness of K_U under nonadaptive reductions

Theorem [H. STOC'20]

For every universal Turing machine U,

$$\mathrm{NP} \subseteq \mathrm{quasiP}_{\|}^{\mathrm{K}_U}.$$

The oracle returns $\mathrm{K}_U(q)$ on query q.

- The proof is inspired by [H. ITCS'20].
- Proof Idea:
  - Symmetry of information

$$\mathrm{K}(y \mid x) + \mathrm{K}(x) \approx \mathrm{K}(x, y) \approx \mathrm{K}(x \mid y) + \mathrm{K}(y).$$

  - k-wise direct product generator DP_k

k-Wise Direct Product Generator [H. STOC'20]

$$\mathrm{DP}_k : \{0,1\}^n \times \left(\{0,1\}^\lambda\right)^k \to \{0,1\}^{\lambda k + k}$$

$$\mathrm{DP}_k(x; z_1, \ldots, z_k) = \left(z_1, \ldots, z_k, \mathrm{Enc}(x)_{z_1}, \ldots, \mathrm{Enc}(x)_{z_k}\right)$$

A pseudorandom generator construction based on a "hard" truth table x that extends seed z by k bits.

- $\mathrm{Enc}: \{0,1\}^n \to \{0,1\}^{2^\lambda}$, a list-decodable error-correcting code
- We may choose Enc so that $\lambda = O(\log n)$.

<u>A Reconstruction Property of DP_k:</u>

For every oracle $D: \{0,1\}^{\lambda k + k} \to \{0,1\}$ and every $x \in \{0,1\}^n$, if

$$\Pr_{z \sim \{0,1\}^{nk}}\left[D\left(\mathbf{DP_k(x; z)}\right) = 1\right] - \Pr_{w \sim \{0,1\}^{nk+k}}\left[D(\boldsymbol{w}) = 1\right] \geq \frac{1}{2},$$

then $\mathrm{K}^{\mathrm{poly}(n),D}(x) \leq O(\lambda k)$.

∃ a poly-time program M^D of length $O(\lambda k)$ that prints x.

Claim: $\mathrm{NP} \subseteq \mathrm{quasiP}_{\|}^{\mathrm{K}}$

- Let $L \in \mathrm{NP}$ and V be a verifier for L.
- Let $x \in \{0,1\}^n$ be an input.
- Let $y_x \in \{0,1\}^{\mathrm{poly}(n)}$ be the lexicographically first certificate (if $x \in L$). $(V(x, y_x) = 1.)$
- **Goal**: To distinguish $\mathrm{DP}_k(y_x; -)$ from the uniform distribution.
  - $\mathrm{K}\big(x, \mathrm{DP}_k(y_x; z)\big) \leq \mathrm{K}(x) + |z| + O(\log k)$ $\quad \because \mathrm{K}(y_x|x) = O(1)$
    $\leq \mathrm{K}(x) + k\lambda + O(\log k).$ $\quad \lambda = O(\log n)$
  - $\mathrm{K}(x, w) \geq \mathrm{K}(x) + \mathrm{K}(w|x) - O(\log n)$ $\quad \because$ Symmetry of Information
    $\geq \mathrm{K}(x) + k\lambda + k - O(\log n)$ $\quad \because \mathrm{K}(w|x) \approx |w|$ with high probability
- K can distinguish $\mathrm{DP}_k(y_x; -)$ for a sufficiently large $k := O(\log n)$.

Claim: $\mathrm{NP} \subseteq \mathrm{quasiP}_{\|}^{\mathrm{K}}$ (continued)

- Define $D_x(w) \coloneqq 1$ iff $\mathrm{K}(x, w) \leq \boldsymbol{s}$.
  - $\Pr_z\left[D_x\left(\mathrm{DP}_k(y_x; z)\right) = 1\right] = 1.$
  - $\Pr_w[D_x(w) = 1] \leq \frac{1}{2}.$

  $\Longrightarrow \mathrm{K}^{\mathrm{poly}(n), D_x}(y_x) \leq O(k\lambda) = O\left(\log^2 n\right).$ ∵ The reconstruction property

- **Goal**: To distinguish $\mathrm{DP}_k(y_x; -)$ from the uniform distribution.
  - $\mathrm{K}\left(x, \mathrm{DP}_k(y_x; z)\right) \leq \mathrm{K}(x) + |z| + O(\log k)$ ∵ $\mathrm{K}(y_x|x) = O(1)$

    $\leq \mathrm{K}(x) + k\lambda + O(\log k) \coloneqq \boldsymbol{s}.$ $\lambda = O(\log n)$
  - $\mathrm{K}(x, w) \geq \mathrm{K}(x) + \mathrm{K}(w|x) - O(\log n)$ ∵ Symmetry of Information

    $\geq \mathrm{K}(x) + k\lambda + k - O(\log n)$ ∵ $\mathrm{K}(w|x) \approx |w|$ with high probability
- K can distinguish $\mathrm{DP}_k(y_x; -)$ for a sufficiently large $k \coloneqq O(\log n)$.

Claim: $\mathrm{NP} \subseteq \mathrm{quasiP}_{\|}^{\mathrm{K}}$ (continued)

- Define $D_x(w) \coloneqq 1$ iff $\mathrm{K}(x, w) \leq \boldsymbol{s}$.
  - $\Pr_z\left[D_x\left(\mathrm{DP}_k(y_x; z)\right) = 1\right] = 1.$
  - $\Pr_w[D_x(w) = 1] \leq \frac{1}{2}.$

$\Longrightarrow \mathrm{K}^{\mathrm{poly}(n), D_x}(y_x) \leq O(k\lambda) = O\left(\log^2 n\right).$

- Given x, enumerate all poly-time programs d of size $O(\log^2 n)$ and check if $U^{D_x}(d)$ is a certificate for $x \in L$.

■

Remark

- If $x = 0^n$, we obtain $\mathrm{NP} \cap \mathrm{Tally} \subseteq \mathrm{quasiP}_{\|}^{R_{\mathrm{K}_U}}$. [H. ITCS'20]
- If $x \sim \{0,1\}^n$, we obtain $\mathrm{NP} \times \{\mathcal{U}\} \subseteq \mathrm{Heur}\ \mathrm{quasiP}_{\|}^{R_{\mathrm{K}_U}}$.

$\because$ the threshold $s \approx |x| + |w|$ if $x \sim \{0,1\}^n$

Two mysteries of complexity theory

1. What is the (errorless) average-case complexity of PH? [Levin'86]

   e.g., $\mathrm{DistPH} \not\subseteq \mathrm{AvgP}$? (Is PH hard on average?)

Main Theorem [H. FOCS'20]: The equivalence of these two questions

2. What is the complexity of computing time-bounded Kolmogorov complexity?
   = meta-complexity [Ko'91]

   e.g., $\mathrm{GapMINKT}^{\mathrm{PH}} \notin \mathrm{P}$? (Is it hard to approximate PH-oracle Kolmogorov complexity?)

Main Theorem

- An interdisciplinary link between average-case complexity and worst-case meta-complexity

Main Theorem [H. FOCS'20]

(Average-case hardness of PH) (Worst-case hardness of meta-complexity)

$$\mathrm{DistPH} \not\subseteq \mathrm{AvgP} \iff \mathrm{GapMINKT}^{\mathrm{PH}} \notin \mathrm{P}$$

Significance:

Average-Case Complexity

- It provides new proof techniques of analyzing average-case complexity via meta-complexity.

Meta-Complexity

- It classifies the complexity of $\mathrm{GapMINKT}^{\mathrm{PH}}$ as a "DistPH-complete" problem.

Application: a hardness amplification theorem for PH

Meta-Complexity – Complexity of Complexity

- Examples of meta-computational problems: $\mathrm{MCSP}, \mathrm{MKTP}, \mathrm{MINKT}, \ldots$

MINKT [Ko'91] = "Compute the time-bounded Kolmogorov complexity"

- t-time-bounded Kolmogorov complexity of x

$$\mathrm{K}^t(x) := (\text{the length of a shortest program that prints } x \text{ in } t \text{ steps})$$

- $\mathrm{MINKT} = \{(x, 1^t, 1^s) \mid \mathrm{K}^t(x) \leq s\}.$

Meta-Complexity – Complexity of Complexity

➢ Examples of meta-computational problems: $\mathrm{MCSP}, \mathrm{MKTP}, \mathrm{MINKT}, \ldots$

MINKT^A [Ko'91] = "Compute the A-oracle time-bounded Kolmogorov complexity"

- A-oracle t-time-bounded Kolmogorov complexity of x

  $\mathrm{K}^{t,A}(x) \coloneqq$ (the length of a shortest program M^A that prints x in t steps)

- $\mathrm{MINKT}^A = \{(x, 1^t, 1^s) \mid \mathrm{K}^{t,A}(x) \leq s\}$.

Remark: In general, we may have $A \not\leq_m^p \mathrm{MINKT}^A$.

It is easy to see $\mathrm{MINKT}^A \in \mathrm{NP}^A$.

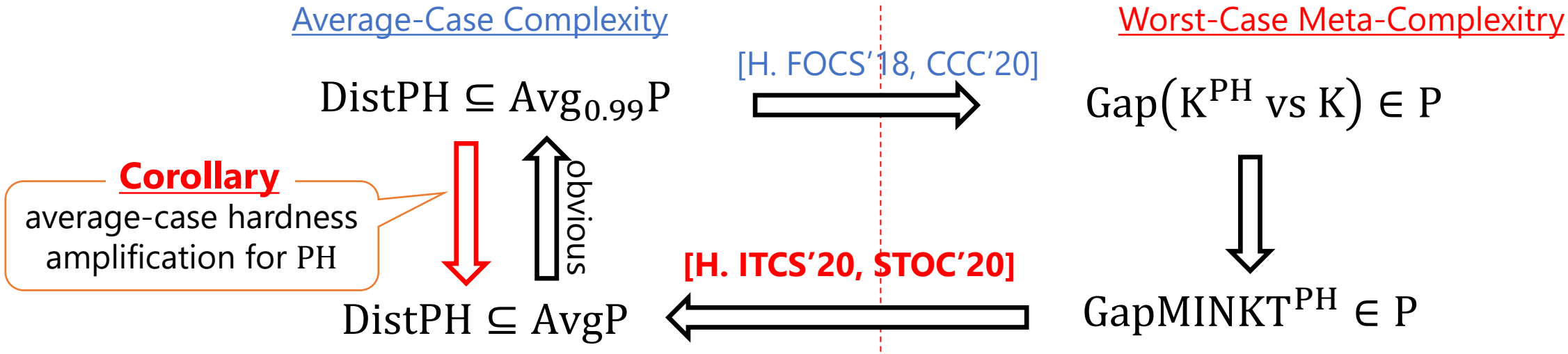
Average-Case Complexity = Meta-Complexity

Theorem [H. (FOCS'20)]

$$\mathrm{DistPH} \subseteq \mathrm{AvgP} \iff \mathrm{GapMINKT}^{\mathrm{PH}} \in \mathrm{P}$$

For every $A \in \mathrm{PH}$, $\mathrm{GapMINKT}^{A} \in \mathrm{P}$

- $\mathrm{GapMINKT}^{A}$: an $O(\log n)$-additive approximation version of MINKT^{A}.
- **Corollary:** A new technique of analyzing **average-case complexity** by **meta-complexity.**



Claim: $\mathrm{GapMINKT}^{\mathrm{NP}} \in \mathrm{P} \Longrightarrow \mathrm{DistNP} \subseteq \mathrm{AvgP}$

- Let $L \in \mathrm{NP}$ and V be a verifier for L.
- Let $x \sim \{0,1\}^n$ be a random input.
- Let $y_x \in \{0,1\}^{\mathrm{poly}(n)}$ be the lexicographically first certificate (if $x \in L$). $(V(x, y_x) = 1.)$
- **Goal**: To distinguish $\mathrm{DP}_k(y_x; -)$ from the uniform distribution.
  - $\mathrm{K}^{2t,\mathrm{SAT}}\big(x, \mathrm{DP}_k(y_x; z)\big) \leq \mathrm{K}^{t,\mathrm{SAT}}(x) + |z| + O(\log k)$ $\because \mathrm{K}^{t,\mathrm{SAT}}(y_x|x) = O(1)$
    $\leq n + k\lambda + O(\log k).$ $\lambda = O(\log n)$
  - $\mathrm{K}^{2t,\mathrm{SAT}}(x, w) \geq \mathrm{K}(x, w) \geq |x| + |w| - O(1)$ with high probability over $x, w \sim \{0,1\}^*$
    $\geq n + k\lambda + k - O(\log n)$ $\because \mathrm{K}(w|x) \approx |w|$ with high probability
- $\mathrm{K}^{2t,\mathrm{SAT}}$ can distinguish $\mathrm{DP}_k(y_x; -)$ for a sufficiently large $k \coloneqq O(\log n)$.

Claim: $\mathrm{GapMINKT}^{\mathrm{NP}} \in \mathrm{P} \Longrightarrow \mathrm{DistNP} \subseteq \mathrm{AvgP}$

- Define $D_x(w) \coloneqq 1$ iff $\mathrm{K}^{2t,\mathrm{SAT}}(x,w) \leq n + k\lambda + O(\log k)$.
  - $\Pr_z\left[D_x\left(\mathrm{DP}_k(y_x;z)\right) = 1\right] = 1.$
  - $\Pr_w[D_x(w) = 1] \leq \frac{1}{2}.$

$\Longrightarrow \mathrm{K}^{\mathrm{poly}(n),D_x}(y_x) \leq k + O(\log n)$ ~~$O(k\lambda) = O\left(\log^2 n\right)$~~.

Can be improved using complexity-theoretic PRG.

- Given x, enumerate all poly-time programs d of size $O(\log n)$ and check if

$U^{D_x}(d)$ is a certificate for $x \in L$.

$\Longrightarrow$ a heuristic polynomial-time algorithm for $L \in \mathrm{NP}$. ■

New Worst- to Average-Case Connections

Main Theorems [H. STOC'21]

(1) $\mathrm{UP} \not\subseteq \mathrm{DTIME}\left(2^{O(n/\log n)}\right) \implies \mathrm{DistNP} \not\subseteq \mathrm{AvgP}$

$\mathrm{PH} \not\subseteq \mathrm{DTIME}\left(2^{O(n/\log n)}\right) \implies \mathrm{DistPH} \not\subseteq \mathrm{AvgP}$

(3) $\mathrm{NP} \not\subseteq \mathrm{DTIME}\left(2^{O(n/\log n)}\right) \implies \mathrm{DistNP} \not\subseteq \mathrm{Avg_PP}$

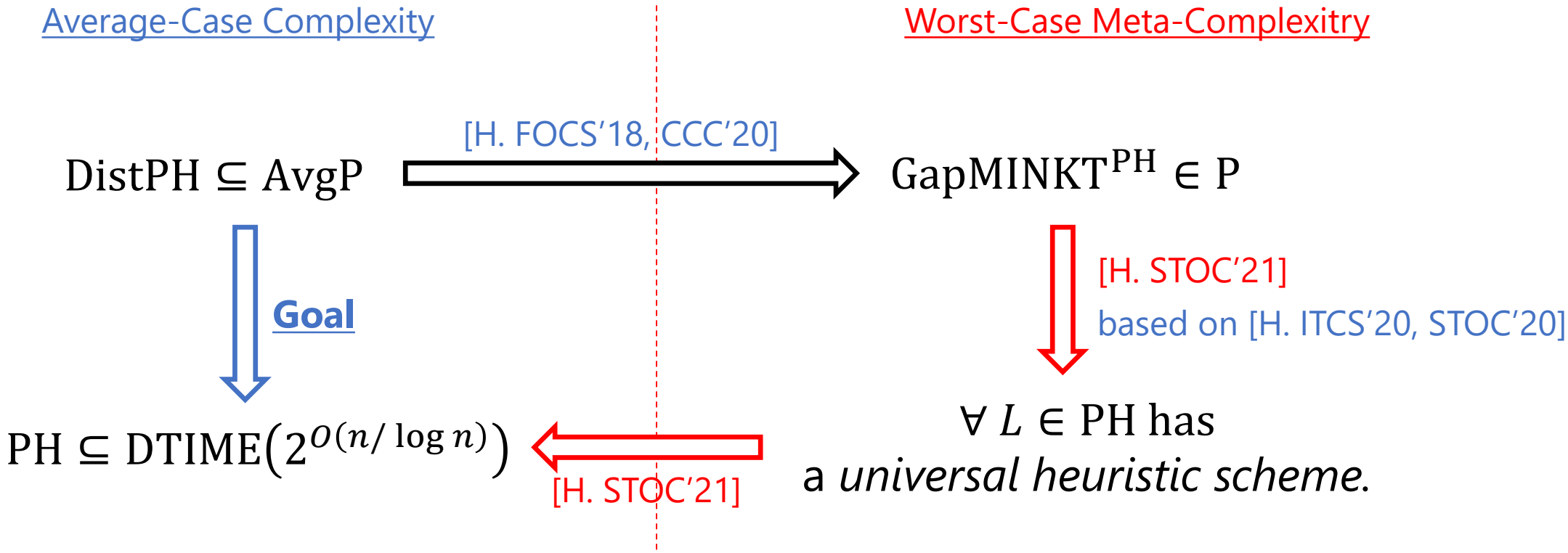
A strong variant of Heuristica doesn't exist!

P-computable average-case polynomial-time

- n denotes the length of inputs (encoded as binary strings).
- $\mathrm{Avg_PP}$ ($\subseteq \mathrm{AvgP}$): the class of $(L, \mathcal{D})$ solvable by average-case polynomial-time algorithms whose running time can be "estimated."

Theorem [H. STOC'21]

(2) $\mathrm{PH} \not\subseteq \mathrm{DTIME}\left(2^{O(n/\log n)}\right) \Longrightarrow \mathrm{DistPH} \not\subseteq \mathrm{AvgP}$



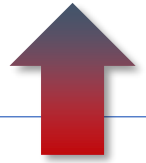
- See my survey in EATCS Bulletin for the proof!

Complexity vs. Computability Theory

- Complexity Theory

~~BPP := {The class of problems solvable efficiently with randomness}.~~

$$\text{NISZK} := \left\{ \begin{matrix} \text{The class of problems that have} \\ \text{non interactive statistical zero knowlege} \end{matrix} \right\}.$$



Bridge the gap via an efficient reduction to R_{K_U}.

- Computability Theory

R_{K_U} := {The set of Kolmogorov-random strings} is **not computable**.

Kolmogorov complexity characterizes NISZK

Theorem [Allender-H.-Tirumala (ITCS'23)]

Approximation of R_{K}

$$\mathrm{NISZK} = \left\{ A\colon \text{decidable} \;\middle|\; A \leq_m^{\mathrm{BPP}} \widetilde{R_{\mathrm{K}}} \right\}$$

$$\mathrm{NISZK}_L = \left\{ A\colon \text{decidable} \;\middle|\; A \leq_m^{\mathrm{RNC}^0} \widetilde{R_{\mathrm{K}}} \right\}$$

Open Question

- Can this new characterization give a new insight into NISZK?

In Conclusion

- I am very influenced by Eric's work.
- Thank you for fundamental contributions to meta-complexity!