Exposition on HMO26:

"Failure of symmetry of information for randomized computations"

Hanlin Ren
IAS

February 18, 2026

This is an exposition of [HMO26].

We say that SoI for rKt holds with error $e(n)$ if for every large enough n and for all strings $x, y \in \{0,1\}^n$, we have

$$\mathrm{rKt}(x \mid y) \leq \mathrm{rKt}(x, y) - \mathrm{rKt}(y) + e(n).$$

1 Theorem 1: SoI to Meta-complexity

Theorem 1. *Suppose that SoI for* rKt *holds with error $e(n)$. Then given $x \in \{0,1\}^n$, the value* $\mathrm{rKt}(x)$ *can be approximated up to an additive term of $\gamma(n) := O(e(n) \cdot \log n)$ in randomized $2^{O(e(n) \cdot \log n)}$ time.*

Proof. Assume that SoI for rKt holds with error $e(n)$. We proceed in the following steps.

1. We need some notations:
   - $M_{\mathrm{opt}}(x \mid y)$ is the rKt-optimal program for generating x given y.
   - $t_{\mathrm{opt}}(x \mid y)$ is the running time of $M_{\mathrm{opt}}(x \mid y)$.
2. **For any length-n strings x, y, one of $t_{\mathrm{opt}}(x)$ and $t_{\mathrm{opt}}(y \mid x)$ is at most $2^{e(n)}\mathrm{poly}(n)$.**

   In fact, this follows from a cute trick regarding Kt-style measures.

   Let $s_1 = |M_{\mathrm{opt}}(x)|$, $t_1 = t_{\mathrm{opt}}(x)$, $s_2 = |M_{\mathrm{opt}}(y \mid x)|$, $t_2 = t_{\mathrm{opt}}(y \mid x)$.

$x = \boxed{}$	$y = \boxed{}$
rKt-optimal program: size s_1, time t_1	rKt-optimal program (given x): size s_2, time t_2

   If we generate x first and then generate y, then we have

   $$\mathrm{rKt}(xy) \leq s_1 + s_2 + \log(t_1 + t_2) + O(\log n).$$

   On the other hand, by SoI, we have

   $$\mathrm{rKt}(xy) \geq s_1 + s_2 + \log(t_1) + \log(t_2) - e(n).$$

   Therefore, $\log(t_1 + t_2) + O(\log n) + e(n) \geq \log(t_1) + \log(t_2)$.

   It follows that $\min\{t_1, t_2\} \leq 2^{e(n)} \cdot \mathrm{poly}(n)$.

3. **rKt is approximated by rK^t for some $t(n) \approx 2^{e(n)}$.**

   In fact, let x denote a string.

   Let i denote the smallest integer such that $t_{\mathrm{opt}}(x_{1\sim i}) \geq 2^{e(n)} \cdot \mathrm{poly}(n)$.

   - That is, the rKt-optimal program for printing $x_{1\sim i}$ takes more than $2^{e(n)} \cdot \mathrm{poly}(n)$ time.

   Split x as follows.

$$x = \overbrace{\boxed{}}^{x_L := x_{1\sim(i-1)}}\underset{i}{\boxed{}}\overbrace{\boxed{}}^{x_R := x_{(i+1)\sim n}}$$

   By Item 2 above, $t_{\mathrm{opt}}(x_R \mid x_{1\sim i}) \leq 2^{e(n)} \cdot \mathrm{poly}(n)$.

   - That is: given the prefix $x_{1\sim i}$, the suffix x_R can be constructed in $2^{e(n)} \cdot \mathrm{poly}(n)$ time.

   By the minimality of i, $t_{\mathrm{opt}}(x_L) < 2^{e(n)} \cdot \mathrm{poly}(n)$.

   Now, by hardwiring i and x_i, we obtain a program of length $|M_{\mathrm{opt}}(x_L)| + |M_{\mathrm{opt}}(x_R \mid x_{1\sim i})| + O(\log n)$ that computes x in $2^{e(n)} \cdot \mathrm{poly}(n)$ time.

   SoI implies that this program has length at most $\mathrm{rKt}(x) + e(n) + O(\log n)$. In fact:

$$\begin{aligned}
\mathrm{rKt}(x) &\geq \mathrm{rKt}(x_L, x_R) - O(\log n) \\
&\geq \mathrm{rKt}(x_L) + \mathrm{rKt}(x_R \mid x_{1\sim i}) - e(n) - O(\log n) && \text{(by SoI)} \\
&\geq |M_{\mathrm{opt}}(x_L)| + |M_{\mathrm{opt}}(x_R \mid x_{1\sim i})| - e(n) - O(\log n).
\end{aligned}$$

4. **The $2^{e(n)}$-time near-optimal program for x is rKt-easy given x.**

   Say that m is a program of length $|m|$ and time t that outputs x. Then

$$\begin{aligned}
\mathrm{rKt}(m \mid x) &\leq \mathrm{rKt}(m, x) - \mathrm{rKt}(x) + e(n) && \text{(by SoI)} \\
&\leq |m| + \log t - \mathrm{rKt}(x) + e(n) + O(\log n).
\end{aligned}$$

   When m is the program defined in Item 3 above, $|m| + \log t - \mathrm{rKt}(x) \leq e(n) + O(\log n)$.

   Hence $\mathrm{rKt}(m \mid x) \leq 2e(n) + O(\log n)$.

5. **To estimate $\mathrm{rKt}(x)$, simply enumerate all the programs.**

   Enumerate all programs m such that $\mathrm{rKt}(m \mid x) \leq 2e(n) + O(\log n)$ and simulate them for $2^{e(n)} \cdot \mathrm{poly}(n)$ steps.

   This takes $\mathrm{poly}(n, 2^{e(n)})$ time. □

References

[HMO26] Jinqiao Hu, Yahel Manor, and Igor C. Oliveira. Failure of symmetry of information for randomized computations. In *STOC*, 2026.