Randomness Extractor Seminar **October, 14, 2016**

Introduction

Leader: KM Chung **Notes: Chi-Ning Chou**

We are going to see that motivation and basic formulation of randomness extractors. Some background knowledge about applied probability will be provided as prilliminary materials for future.

In this note, we will start with an example to get some feeling of how randomness plays in our daily life and see informal definition of extractors and its applications. Next, some common random sources will be introduced. Then, we'll discuss the concept of *how random* a random source is in a quantitative way. Finally, the mathematical definition of extractors will be introduced and we'll see several properties and applications.

1 Motivation

In many applications, we see the power of randomness. For instance, the polynomial identity problem, there exists a simple and fast randomized algorithm while no efficient deterministic algorithm is known. In computational complexity, we used the complexity class **BPP** to capture the problems that can be solved with polynomially many random bits in polynomial time. Concretely, if $L \in \mathbf{BPP}$, then exists algorithm $A(\cdot, \cdot)$ with error $\gamma < 1/2$ such that for any integer n exist $m = \text{poly}(n)$ such that for input $x \in \{0,1\}^n$,

- If $x \in L$, then $\mathbb{P}_{r \leftarrow \{0,1\}^m}[A(x,r) = 1] \geq 1 - \gamma$.
- If $x \notin L$, then $\mathbb{P}_{r \leftarrow \{0,1\}^m}[A(x,r) = 1] \leq \gamma$.

The randomness used in the framework of randomized algorithm is *uniform*. That is, we assume the algorithm can access uniformly distributed random bits. However, what if one cannot access uniform randomness? What if we only have *weak* random source in the sense that it has certain level of randomness? The goal of randomness extractor is then to extract such randomness into the randomness that is close to uniform distribution so that we can replace uniform bits with them.

2 Random sources

From the vocabulary game example, one can see that there are actually two potential bad properties lie in a random sources: *inconsistence* and *correlation*. Formally, if a random source is inconsistent, we would call it **non-identical**, and if it is correlated, we would call it **dependent**. See Table 1.

	Independent	Dependent
Identical	IID	SV source, etc.
Non-identical	IndBits	

Table 1: Random sources.

In the following, we will start with the simplest random source: IID source, and see how to extract uniform randomness from it. Then, we will introduce weaker sources such as IndBits and SV source. In Section 4, we will see how to extract uniform randomness from these sources and take a look at a negative result.

2.1 IID

The simplest random source taught in a probability course would be the *independent and identical (IID)* sources. Let $X_1, \ldots, X_n$ be a sequence of random variables, they are IID if the probability distribution of each X_i are the same and independent to one an another. For instance, $X_1, \ldots, X_n$ could be a sequence of coin flipping results where the probabilities of head are p. At first glance, this random source might not be as uniformly random as we would expect. It turns out that there's a simple way to convert such IID source to uniform randomness, *i.e.,* the pure randomness we want. The idea is as follows. Combine two flips into one, if the result is HT then output 0, if the result is TH then output 1; otherwise, give up and flip the coin again. From Table 2, we can see that the probability of 0 and 1 to show up is the same, *i.e.,* we produce uniform randomness, while the probability of doing again is $1 - 2p$.

	H	T
H	p^2	$p \cdot (1-p)$
T	$p \cdot (1-p)$	$(1-p)^2$

Table 2: Turning IID coin flipping into uniform randomness.

2.2 IndBits

Now, let's take a look at another interesting random source in which the identical property no longer holds. A sequence of random variables $X_1, \ldots, X_n$ is called IndBits source if they are independent and $\mathbb{P}[X_i = 1] = \delta_i$ where $\delta \leq \delta_i \leq 1 - \delta$ for all $i \in [n]$ and a constant δ. That is to say, we only know that the probability of X_i to be a head is within δ and $1 - \delta$ while different i has different probability.

An interesting exercise is that when we xor n such bits, the resulting bit will actually distribute more and more close to an uniform distribution. Concretely, $|\mathbb{P}[\oplus_{i \in [n]} X_i = 1] - \frac{1}{2}| \leq 2^{-\Omega(n)}$. Intuitively, although we cannot be guaranteed to have a pure uniform randomness, we can be arbitrarily close to an uniformly distributed bit exponentially fast. Here, we think of the "closeness" between a random source and uniform distribution as the probability difference of being 1, in next section, we will formally generalize this notion and show that it has nice practical implication.

2.3 SV source

Santha-Vazirani source (SV source), a.k.a. unpredictable source is a source that the i-th bit is unpredicted by the previous $i - 1$ bits in the following sense.

$$\delta \leq \mathbb{P}[X_i = 1 \mid X_1 = x_1, \ldots, X_{i-1} = x_{i-1}] \leq 1 - \delta, \tag{1}$$

for some constant $0 < \delta < 1/20$. We denote the source with n bits satisfying the above condition as $\text{UnpredBits}_{n,\delta}$. At first glance, SV source is more less looked similar to the previous IID source

and IndBits source, however, it turns out that for $\epsilon < 1/2 - \delta$, there's no deterministic ϵ-extractor for SV source.

3 Measures of randomness

In Section 2, we see several types of random sources. Intuitively, each of them has different *level* of randomness. Now, it's time for us to give a *quantitative* analysis on measuring the randomness of a random source.

3.1 Random source as distribution

First of all, an important concept is to think of a random source of m bits as a *probability distribution* over $\{0,1\}^m$. Take our favorite uniform distribution as example, we can think of it as an uniform probability distribution over $\{0,1\}^m$, *i.e.,* for any $r \in \{0,1\}^m$, the probability of r to be drawn from the random source is $1/2^m$. As to arbitrary random source, we can record the probability of every possible realization $r \in \{0,1\}^m$ into a probability distribution. That is to say, to discuss how random a source is, it is equivalent to work on the probability distribution.

3.2 Difference between random source

Since our goal is to extract the randomness in a weak random source into a source that is *close* to the uniform distribution. Here, we need to formally define the measure of closeness between two random sources.

Definition 1 (statistical distance)**.** *Given two random sources X and Y over $\{0,1\}^m$, the statistical distance between them is*

$$\Delta(X,Y) := \max_{S \subseteq \{0,1\}^m} |\mathbb{P}[X \in S] - \mathbb{P}[Y \in S]| \tag{2}$$

Furthermore, we say X is ϵ-close to Y if $\Delta(X,Y) \leq \epsilon$.

It turns out that statistical distance enjoys several nice properties as follows.

Lemma 2 (properties of statistical distance)**.** *Let X, Y, and Z be random variables over $\mathcal{X}$.*

- *It's a distance measure for probability distribution, i.e.,*
  - *Non-negativity: $\Delta(X,Y) \geq 0$.*
  - *Identity: $\Delta(X,Y) = 0$ iff $X = Y$.*
  - *Symmetry: $\Delta(X,Y) = \Delta(Y,X)$.*
  - *Triangle inequality: $\Delta(X,Z) \leq \Delta(X,Y) + \Delta(Y,Z)$.*
- *Data processing inequality: Let $f : \mathcal{X} \to \mathcal{Y}$, then $\Delta(f(X), f(Y)) \leq \Delta(X,Y)$.*
- *Tensorization: Let $\{X_i\}_{i\in[m]}$, and $\{Y_i\}_{i\in[m]}$ be random variables, then $\Delta((X_1,\ldots,X_m),(Y_1,\ldots,Y_m)) \leq \sum_{i\in[m]} \Delta(X_i,Y_i)$.*

With the notion of statistical distance at hand, now we can go back to the randomized algorithm and see how a random source that is ϵ-close to uniform distribution can help us.

Proposition 3. *Let $A(\cdot,\cdot)$ be a randomized algorithm for language $L \in$ **BPP** with error γ, for any integer n and $m = poly(n)$, if X is an m bits random source that is ϵ-close to U_m, then $A(\cdot, X)$ has error at most $\epsilon + \gamma$.*

Proof. Without loss of generality, we consider input $x \in L$. Let $S_x := \{r \in \{0,1\}^m \mid A(x,r) = 0\}$ be the set of bad random strings to x, we have

$$\begin{aligned}\mathbb{P}_X[A(x,X)=0] &= \mathbb{P}_X[A(x,X)=0] = \mathbb{P}[X \in S_x] \\ &\leq \mathbb{P}_{U_m}[U_m \in S_x] + \epsilon = \epsilon + \gamma.\end{aligned}$$

□

3.3 Entropies

Once we view a random source as a probability distribution, it would be natural to think about using *entropies* as a measure of randomness. Entropies capture two important properties in computer science: *additive*, and *bits*, which, in my opinion, is the fundamental reason why we like to use entropy to measure randomness.

Name	Definition
Shannon entropy	$H_{Sh}(X) = \mathbb{E}_{x \leftarrow X}[\log \frac{1}{\mathbb{P}}]$
Rényi entropy	$H_2(X) = \log \frac{1}{\mathbb{E}_{x \leftarrow X}[\mathbb{P}[X=x]]}$
Min-entropy	$H_\infty(X) = \min_{x \in \text{supp}(X)} \log \frac{1}{\mathbb{P}[X=x]} = \log \frac{1}{\max_{x \in \text{supp}(X)} \mathbb{P}[X=x]}$

Table 3: Four common entropies.

The reader might be confused with the three entropies at first glance, but don't be afraid. They are all defined with respect to the $\log 1/x$ function with slightly different settings.

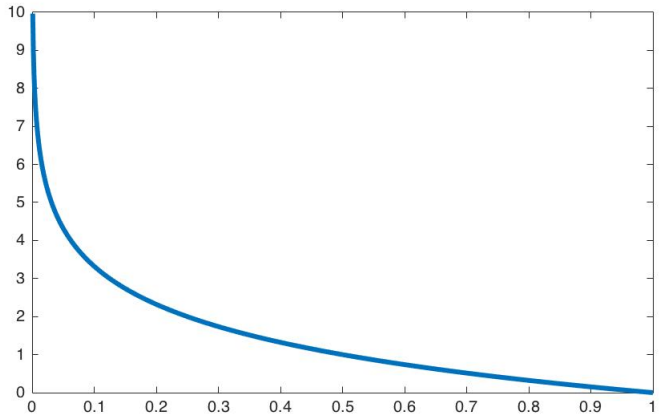

Figure 1: The $\log 1/x$ function.

For any $x \in \text{Supp}(X)$, it has a corresponding probability $\mathbb{P}[X = x]$. We can put it into the $\log 1/x$ function and see the value. Intuitively, the smaller the probability of x showing up means that once it shows up, you receive lots of information. Thus, $\log 1/\mathbb{P}[X = x]$ is somehow being regarded as the *information* we get when seeing x.

With this intuition in mind, we can now reinterpret the three entropies above. The Shannon entropy is the weighted average of the individual information. The Rényi entropy is the information of averaging probability. Finally, the min entropy is the smallest information.

4 Extractors

Finally, it's time for us to formally define the notion of extractors.

Definition 4 (extractors)**.** *Let $\mathcal{C}$ be a class of n bits random sources, and $\epsilon > 0$. An ϵ-extractor for $\mathcal{C}$ is a function $Ext : \{0,1\}^n \to \{0,1\}^m$ such that $\forall X \in \mathcal{C}$, $\Delta(Ext(X), U_m) \leq \epsilon$.*

Note that the extractors defined here is **deterministic** in the sense that it is a fixed function.

4.1 Impossibility result for deterministic extractors

Here, we are going to introduce two impossibility results for deterministic extractors defined above. The main idea to show the impossibility of extractors is to **construct** a bad random source such that previously chosen extractor performs bad on it.

The first example showed that there's no deterministic extractor for $(m-1)$-source.

Proposition 5 (no deterministic extractor for $(m-1)$-source)**.** *For any deterministic function $Ext : \{0,1\}^n \to \{0,1\}^m$, there exists an $(m-1)$-source X such that $Ext(X)$ is constant.*

Proof. As we fix the extractor Ext, without loss of generality, assume $|\text{Ext}^{-1}(0)| \leq |\text{Ext}^{-1}(1)|$, construct the following source:

$$\mathbb{P}[X = x] = \begin{cases} \frac{1}{|\text{Ext}^{-1}(1)|}, & \text{if } \text{Ext}(x) = 1 \\ 0, & \text{else.} \end{cases}$$

One can verify that X is a $(m-1)$-source since the support size $|\text{Ext}^{-1}(1)| \geq 2^{m-1}$ and $\text{Ext}(X)$ is a constant. □

As k-source is rather a unstructured source, it is not too surprising that there does not exist a deterministic extractor for it. However, the following showed that for a much more structured source, the SV source, there also does not exist a deterministic extractor.

Proposition 6. *For any $n \in \mathbb{N}$, $0 < \delta < 1/2$, and a **fixed** extractor $Ext : \{0,1\}^n \to \{0,1\}$, there exists a source $X \in UnpredBits_{n,\delta}$ such that Ext is not an $(1/2 - \delta)-$extractor for X.*

Proof. We prove the theorem with the following two steps.

(1) Let X be a source taking value in $\{0,1\}^n$ such that for all x, y, $\mathbb{P}[X = x]/\mathbb{P}[X = y] \leq (1-\delta)/\delta$, then $X \in \text{UnpredBits}_{n,\delta}$.

(2) For any fixed $\mathrm{Ext} : \{0,1\}^n \to \{0,1\}$, and $0 < \delta < 1/2$, there exists $X \in \mathrm{UnpredBits}_{n,\delta}$ such that $\mathbb{P}[\mathrm{Ext}(X) = 1] \leq \delta$ or $\mathbb{P}[\mathrm{Ext}(X) = 1] \geq 1 - \delta$.

(1) First, observe that for any $i \in [n]$ and $x_{1,\dots,i}, y_{1,\dots,i} \in \{0,1\}^i$, denote the first i bits of X as $X_{1,\dots,i}$, we have

$$\begin{aligned}\delta \cdot \mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}] &\leq \delta \cdot \sum_{x_{i+1},\dots,x_n \in \{0,1\}} \mathbb{P}[X = x] \\ &\leq (1-\delta) \cdot \sum_{y_{i+1},\dots,y_n \in \{0,1\}} \mathbb{P}[Y = y] \\ &= (1-\delta) \cdot \mathbb{P}[Y_{1,\dots,i} = y_{1,\dots,i}]\end{aligned}$$

That is, the inequality also holds for considering only the first i-th bits. Next, let $\bar{x}_{1,\dots,i} = x_{1,\dots,i}$ except flipping the i-th bits, by Bayes rule,

$$\begin{aligned}\mathbb{P}[X_i = x_i \mid X_{1,\dots,i-1} = x_{1,\cdots-1}] &= \frac{\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}]}{\mathbb{P}[X_{1,\dots,-1} = x_{1,\dots,i-1}]} \\ &= \frac{\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}]}{\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}] + \mathbb{P}[X_{1,\dots,i} = \bar{x}_{1,\dots,i}]}\end{aligned}$$

From our previous observation, $\frac{\mathbb{P}[X_{1,\dots,i}=x_{1,\dots,i}]}{\mathbb{P}[X_{1,\dots,i}=\bar{x}_{1,\dots,i}]} \leq \frac{1-\delta}{\delta}$, *i.e.*,

$$\begin{aligned}\mathbb{P}[X_i = x_i \mid X_{1,\dots,i-1} = x_{1,\cdots-1}] &\leq \frac{\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}]}{\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}] + \frac{\delta}{1-\delta}\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}]} \\ &= 1 - \delta,\end{aligned}$$

and $\frac{\mathbb{P}[X_{1,\dots,i}=x_{1,\dots,i}]}{\mathbb{P}[X_{1,\dots,i}=\bar{x}_{1,\dots,i}]} \geq \frac{1-\delta}{\delta}$

$$\begin{aligned}\mathbb{P}[X_i = x_i \mid X_{1,\dots,i-1} = x_{1,\cdots-1}] &\geq \frac{\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}]}{\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}] + \frac{1-\delta}{\delta}\mathbb{P}[X_{1,\dots,i} = x_{1,\dots,i}]} \\ &= \delta,\end{aligned}$$

(2) Fix an extractor $\mathrm{Ext} : \{0,1\}^n \to \{0,1\}$, take a source X by the following probability. For any $x \in \{0,1\}^n$,

$$\mathbb{P}[X = x] = \begin{cases} \frac{\delta}{|\mathrm{Ext}^{-1}(0)|}, & \text{if } \mathrm{Ext}(x) = 0 \\ \frac{1-\delta}{|\mathrm{Ext}^{-1}(1)|}, & \text{if } \mathrm{Ext}(x) = 1 \end{cases}$$

One can easily verify that X satisfy the inequality in (1) and thus $X \in \mathrm{UnpredBits}_{n,\delta}$.

□

Corollary 7. *There is no deterministic ϵ-extractor for $\mathrm{UnpredBits}_{n,\delta}$ for any $\epsilon \leq 1/2 - \delta$.*

4.2 Seeded extractors

From the previous impossibility results, one can see that once we fix a extractor Ext, there can always find a bad random source X for Ext such that $\mathrm{Ext}(X)$ is not uniform enough. However, this is a really **worst-case** scenario. That is, when fixing Ext, it performs well on most of the random sources. Intuitively, we can think of

Proposition 8. *For any $n, m, k \in \mathbb{N}$, and $\epsilon > 0$. For arbitrary flat k-source X, if we randomly pick a function Ext : $\{0,1\}^n \to \{0,1\}^m$ with $m = k - 2\log 1/\epsilon - O(1)$, then Ext$(X)$ will be ϵ-close to U_m with probability $2^{-\Omega(K\epsilon^2)}$.*

Before we formally prove Proposition 8, let's take a look at Figure 2.

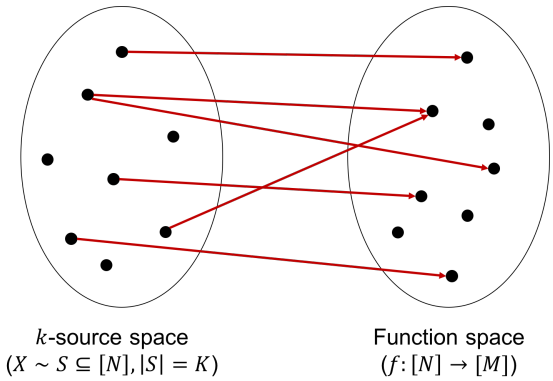

Figure 2: Illustration for Proposition 8. For any fixed k-source, there are only a small portion of function that is not an ϵ-extractor for it.

Proof. Consider arbitrary $T \subseteq [M]$, define random variable $Y_T = \mathbb{P}_X[\mathrm{Ext}(X) \in T] - \mathbb{P}_{U_m}[U_m \in T]$. Note that the randomness of Y_T came from both Ext and X.

- $\mathbb{E}_{\mathrm{Ext}}[Y_T] = 0$.
- $|Y_T| \leq \epsilon$ with probability $1 - 2^{-\Omega(K\epsilon^2)}$.

As a result, after applying union bound on all possible T, we have

$$\begin{aligned}\mathbb{P}[\exists T \subseteq [M],\ |Y_T| > \epsilon] &\leq \sum_{T \subseteq [M]} \mathbb{P}[|Y_T| > \epsilon] \\ &\leq 2^M \cdot 2^{-\Omega(K\epsilon^2)} = 2^{-\Omega(K\epsilon^2)}.\end{aligned}$$

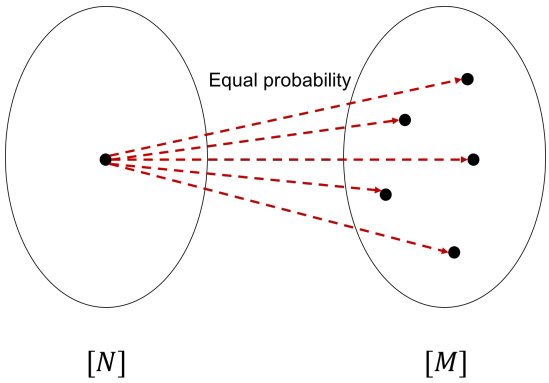

Figure 3: Random function from $[N]$ to $[M]$. For any point $x \in [N]$, the probability of x being mapped to a point $y \in [M]$ by a random function is uniform.

See Figure 3 for some intuition. Now, we go back to prove the previous two propositions.

- $\mathbb{E}_{\text{Ext}}[Y_T] = 0$.
  Intuitively, there are two ways to think about a random function $\text{Ext} : [N] \to [M]$:[1]

  (1) For any $x \in [N]$, and for any $y \in [M]$, the probability of $\text{Ext}(x) = y$ is $1/M$.
  (2) For any $y \in [M]$, and for any $x \in [N]$, the probability of $\text{Ext}(x) = y$ is $1/M$.

  Here, we use the second one. That is, for a fixed $T \subseteq [M]$, the probability of any $x \in [N]$ such that $\text{Ext}(x) \in T$ is $|T|/M$. As a result, $\mathbb{P}_X[\text{Ext}(X) \in T] = |T|/M$, which is exactly the same as uniform distribution.

- $|Y_T| \leq \epsilon$ with probability $1 - 2^{-\Omega(K\epsilon^2)}$.
  Note that here there are two sources of randomness: the choice of Ext and X. First, we split Y_T into $[K]$ random variable as follow.

$$Y_T = \frac{1}{K} \cdot \sum_{x \in \text{Supp}(X)} \left(\mathbf{1}_{x \in T} - \frac{1}{K} \right)$$

  Observe that for any $x \in \text{Supp}(X)$, $\mathbf{1}_{x \in T}$ has mean $1/K$. By Chernoff bound, we have

$$\mathbb{P}_{\text{Ext}}[|Y_T| > \epsilon] \leq 2^{-\Omega(K\epsilon^2)}. \tag{3}$$

□

From Proposition 8, for a fixed flat k-source, a random extractor can successfully extract $m = k - 2\log 1/\epsilon + O(1)$ randomness with high probability. However, when it comes to **all** flat k-source, as there are $\binom{N}{K} \approx N^K$ distinct flat k-sources, union bound on the error probability won't work.

To resolve this tragedy, people then came up with a brilliant construction of extractor: *the deeded extractors*, which utilize additional **pure** random bits to boost the performance. Let's first formally define the notion of seeded extractors.

[1] Be aware of the quantifiers.

Definition 9 (seeded extractors)**.** *A function* $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k,ϵ)*-seeded extractor for some* $\epsilon > 0$ *if for every* k*-source* X*,* $Ext(X, U_d)$ *is* ϵ*-close to* U_m*.*

It turns out that with this definition, we can prove that there **exists** an (k,ϵ)-seeded extractor that extracts almost all the min entropy of the input k-source.

Proposition 10 (existence of seeded extractor)**.** *For* $n \in \mathbb{N}$*,* $k \in [n]$*, and* $\epsilon > 0$*. There exists* (k,ϵ)*-seeded extractor with* $m = k + d - 2\log 1/\epsilon - O(1)$ *and* $d = \log(n-k) + \log 1/\epsilon + O(1)$*.*

Before go into the proof, let's have some interpretation on the parameters in Proposition 10. First, the m here is optimal since it means that the extractor has extracted all the min entropy in both the k-source and the seed except $2\log 1/\epsilon$ cost. As to the seed length d, one can see from the proof that it is necessary to be at least logarithmic in $n-k$.

Proof. Basically the proof is similar to that of Proposition 8 except having a larger domain $[ND]$ and requiring to apply union bound on all k-sources. That is to say, when fixed a flat k-source X and randomly pick a function $\text{Ext} : [N] \times [D] \to [M]$, the probability of Ext **not** being a (k,ϵ)-seeded extractor for X is at most $2^{-\Omega(KD\epsilon^2)}$. Apply union bound we have

$$\begin{aligned}
\mathbb{P}_{\text{Ext}}[\text{Ext fails on one } k\text{-source}] &= \sum_{S \subseteq [N], |S|=K} \mathbb{P}_{\text{Ext}}[\text{Ext fails on the flat } k \text{ source with support } S] \\
&\leq \binom{N}{K} \cdot 2^{-\Omega(KD\epsilon^2)} \\
(\because \text{Stirling's formula}) &\leq (\frac{Ne}{K})^K \cdot 2^{-\Omega(KD\epsilon^2)}.
\end{aligned}$$

When we properly choose $d \geq \log(n-k) + 2\log 1/\epsilon + O(1)$, the above error probability will be less than 1. Thus, by the probabilistic method, we have showed that there **exists** a seeded extractor with logarithmic seed length that can extract ϵ-close randomness only losing $2\log 1/\epsilon$ min entropy. □

5 Conclusion

In this post, we have a brief introduction to randomness extractors. Seeing its applications, the first impossibility result, and the existence of good seeded extractor (log-length seed). In the future, we are going to see how to explicitly construct a good extractor and other variants of extractors.