1 Introduction

In this paper, we review the lecture note number 4 made by Pascal Lafourcade the 28th september 2009. In this paper, we first introduce some definitions, like Probability Ensembles, Polynomial-Time Indistinguishability, and Efficiently Constructible Ensembles. We present then the notion of hybrid technique to prove some theorems, and in the next part, we study an application of these hybrid technique applied with pseudo-random generators. And we conclude in the last part.

2 Definitions

2.1 Probability Ensemble

We define what is a Probability Ensemble. Let $I$ be a countable index set. An ensemble indexed by $I$ is a sequence of random variable indexed by $I$. Namely, any $X = \{X_i\}_{i \in I}$, where each $X_i$ is a random variable, is an ensemble indexed by $I$.

Notations

- $X = \{X_n\}_{n \in \mathbb{N}}$ has each $X_n$ ranging over strings of length $\text{poly}(n)$.
- $X = \{X_w\}_{w \in \{0,1\}^*}$ has each $X_w$ ranging over string of length $\text{poly}(|w|)$.

Example: Sequences $\{x_n\}_{n \in \mathbb{N}}$ and $\{y_n\}_{n \in \mathbb{N}}$ are said to be computationally indistinguishable if no efficient procedure can tell them apart.

2.2 Polynomial-Time Indistinguishability

We say that two ensembles, $X := \{X_n\}_{n \in \mathbb{N}}$ and $Y := \{Y_n\}_{n \in \mathbb{N}}$, are indistinguishable in polynomial time if for every probabilistic polynomial-time algorithm $D$, every positive
polynomial $p(\cdot)$, and all sufficiently large $n$'s,

$$\left| Pr[D(X_n, 1^n) = 1] - Pr[D(Y_n, 1^n) = 1] \right| < \frac{1}{p(n)}$$

In the same way, two ensembles, $X := \{X_w\}_{w \in S}$ and $Y := \{Y_w\}_{w \in S}$, are indistinguishable in polynomial time if for every probabilistic polynomial-time algorithm $D$, every positive polynomial $p(\cdot)$, and all sufficiently long $w \in S$,

$$\left| Pr[D(X_w, w) = 1] - Pr[D(Y_w, w) = 1] \right| < \frac{1}{p(|w|)}$$

### 2.3 Example

Let $b$ be a string generated by flipping a "fair" coin until head appears (head = 1). Let $X$ be random variable which represents the size of $b$. Define random variables $B_1, B_2, \ldots$, where $B_i$ represents the value of the bit assigned to $b$ in the $i^{th}$ flip, if $X \geq i$, and $\star$ otherwise.

**Note**: exactly one $B_i$ will take the value 1, in which case $X$ takes the value $i$. Evidently, for each $i \geq 1$, then $B_i$ is uniformly distributed over $\{0, 1\}$, and otherwise, $B_i = \star$

$$P[B_i = 0 | X \geq i] = \frac{1}{2}$$

$$P[B_i = 1 | X \geq i] = \frac{1}{2}$$

$$P[B_i \neq \star | X < i] = 1$$

$$P[X \geq 1] = 1$$

$$P[X \geq 2] = P[B_1 = 0 | X \geq 1]P[X \geq 1] = \frac{1}{2}$$

$$P[X \geq 3] = P[B_2 = 0 | X \geq 2]P[X \geq 2] = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

By induction on $i$

$$P[X \geq i] = P[B_{i-1} = 0 | X \geq i - 1]P[X \geq i - 1] = \frac{1}{2} \cdot \frac{1}{2^{i-2}} = \frac{1}{2^{i-1}}$$

$X$ has a geometric distribution with success $\frac{1}{2}$.

The following simple probabilistic algorithm corresponds to flipping a coin until head appears:

```
repeat
    b \leftarrow_r \{0, 1\}
until b = 1
```
Here are two exercises:

Exercice 1
Consider the algorithm $D_1$ which flips a coin and outputs its outcome (0-1), with probability $\frac{1}{2}$. Prove that

$$|Pr[D_1(X) = 1] - Pr[D_1(Y) = 1]|$$

is negligible.

Where $X$ is the event obtain 1 and $Y$ obtain 0.

Exercice 2
Consider the algorithm $D_2$ that outputs 1 iff the input string contains more zeros than ones. If $D_2$ can be implemented in polynomial time, then prove that $X$ and $Y$ are polynomial-time-indistinguishable.

Now we define notion of transitivity of indistinguishability for three ensembles:

Transitivity
Let $X := \{X_n\}_{n \in \mathbb{N}}$, $Y := \{Y_n\}_{n \in \mathbb{N}}$ and $Z := \{Z_n\}_{n \in \mathbb{N}}$ three ensembles. If $X$ and $Y$ are indistinguishable in polynomial time, $Y$ and $Z$ are indistinguishable in polynomial time then $X$ and $Z$ are indistinguishable in polynomial time.

2.4 Indistinguishability by Repeated Sampling
Two ensembles, $X := \{X_n\}_{n \in \mathbb{N}}$ and $Y := \{Y_n\}_{n \in \mathbb{N}}$ are indistinguishable by polynomial-time sampling if for every probabilistic polynomial-time algorithm $D$, every positive polynomials $m(.)$ and $p(.)$, and all sufficiently large $n$’s:

$$|Pr[D(X_n^1, ..., X_n^{m(n)}) = 1] - Pr[D(Y_n^1, ..., Y_n^{m(n)}) = 1]| < \frac{1}{p(n)}$$

where $X_n^1$ through $X_n^{m(n)}$ and $Y_n^1$ through $Y_n^{m(n)}$ are independent random variables, with each $X_n^i$ identical to $X_n$ and $Y_n^i$ identical to $Y_n$.

2.5 Efficiently Constructible Ensembles
An ensemble $X := \{X_n\}_{n \in \mathbb{N}}$ is said to be polynomial-time-constructible if there exists a probabilistic polynomial-time algorithm $S$ such that for every $n$, the random variables $S(1^n)$ and $X_n$ are identically distributed.
3 Hybrid Technique

3.1 Theorem

Theorem 1

Let \( X := \{X_n\}_{n \in \mathbb{N}} \) and \( Y := \{Y_n\}_{n \in \mathbb{N}} \) be two polynomial-time-constructible ensemble, and suppose that \( X \) and \( Y \) are indistinguishable in polynomial time. Then \( X \) and \( Y \) are indistinguishable by polynomial-time sampling.

3.2 Proof by contradiction

We prove that the existence of an efficient algorithm that distinguishes \( X \) and \( Y \) using several samples implies the existence of an efficient algorithm which distinguishes the ensembles \( X \) and \( Y \).

3.3 Proof

We assume that there is \( D \) a polynomial-time algorithm such that for many \( n \)’s holds :

\[
\Delta(n) := |Pr[D(X_n^{(1)}, \ldots, X_n^{(m)}) = 1] - Pr[D(Y_n^{(1)}, \ldots, Y_n^{(m)}) = 1]| > \frac{1}{p(n)}
\]

where \( m := m(n) \) and the \( X_n^{(i)} \) and \( Y_n^{(i)} \) are defined by repeated sampling.

**GOAL:** Finding a probabilistic polynomial-time algorithm \( D' \) that distinguishes \( X \) and \( Y \).

3.4 Introducing \( H_n^k \)

For every \( 0 \leq k \leq m \), we define the hybrid random variable

\[
H_n^k := (X_n^{(1)}, \ldots, X_n^{(k)}, Y_n^{(k+1)}, \ldots, Y_n^{(m)})
\]

where \( X_n^{(1)} \) through \( X_n^{m(n)} \) and \( Y_n^{(1)} \) through \( Y_n^{m(n)} \) are independent random variables, with each \( X_n^{(i)} \) identical to \( X_n \) and \( Y_n^{(i)} \) identical to \( Y_n \). Clearly we have

\[
H_n^m := (X_n^{(1)}, \ldots, X_n^{(m)})
\]

and

\[
H_n^0 := (Y_n^{(1)}, \ldots, Y_n^{(m)})
\]
3.5 Idea of the Proof

By hypothesis, $D$ distinguishes $H_n^0$ and $H_n^m$. We use $D$ to build $D'$ which distinguishes $X$ and $Y$:

1. selects $k$ uniformly in the set $\{0, 1, ..., m - 1\}$.
2. generates $k$ independent samples of $X_n$ denoted $x^1, ..., x^k$.
3. generates $m - k - 1$ independent samples of $Y_n$ denoted $y^{k+2}, ..., y^m$.
4. invokes $D$ with the input $\alpha$ and halts with the output
   
   $$D'(\alpha) = D(x^1, ..., x^k, \alpha, y^{k+2}, ..., y^m)$$

Claim 1

$$Pr[D'(X_n) = 1] = \frac{1}{m} \sum_{k=0}^{m-1} Pr[D(H_n^{k+1}) = 1]$$

and

$$Pr[D'(Y_n) = 1] = \frac{1}{m} \sum_{k=0}^{m-1} Pr[D(H_n^k) = 1]$$

Remark

- $\sum_{k=0}^{m-1} Pr[D(H_n^{k+1}) = 1]$ corresponds to all $H_n^i$ except $H_n^0$
- $\sum_{k=0}^{m-1} Pr[D(H_n^k) = 1]$ corresponds to all $H_n^i$ except $H_n^m$

Proof of Claim 1: By construction of the algorithm $D'$, we have

$$D'(\alpha) = D(X_n^{(1)}, ..., X_n^{(k)}, \alpha, Y_n^{(k+2)}, ..., Y_n^{(m)})$$

where $k$ is uniformly distributed in $\{0, 1, ..., m - 1\}$.

$$Pr[D'(X_n) = 1] = \sum_{l=0}^{m-1} Pr[k = l] Pr[D(X_n^{(1)}, ..., X_n^{(k)}, X_n^{(l)}, Y_n^{(k+2)}, ..., Y_n^{(m)}) = 1]$$

Using the definition of the hybrids $H_n^k$, the claim follows.

$$Pr[D'(X_n) = 1] = \frac{1}{m} \sum_{l=0}^{m-1} Pr[D(H_n^{k+1}) = 1]$$
Claim 2
For $\Delta(n)$ we have:

$$|\Pr[D'(X_n) = 1] - \Pr[D'(Y_n) = 1]| = \frac{\Delta(n)}{m(n)}$$

where

$$\Delta(n) := |\Pr[D(X_n^{(1)}, ..., X_n^{(m)}) = 1] - \Pr[D(Y_n^{(1)}, ..., Y_n^{(m)}) = 1]|$$

where $m := m(n)$ and the $X_i^n$ and $Y_i^n$ are defined by repeated sampling.

Proof of Claim 2: Using Claim 1 we get,

$$|\Pr[D'(X_n) = 1] - \Pr[D'(Y_n) = 1]| = \frac{1}{m} \left| \sum_{k=0}^{m-1} \Pr[D(H_n^{k+1}) = 1] - \sum_{k=0}^{m-1} \Pr[D(H_n^k) = 1] \right|$$

$$= \frac{1}{m} \left| \Pr[D(H_n^m) = 1] - \Pr[D(H_n^0) = 1] \right|$$

$$= \frac{\Delta(n)}{m}$$

where the last equality follows by recalling that:

$$H_n^m := (X_n^{(1)}, ..., X_n^{(m)})$$

$$H_n^0 := (Y_n^{(1)}, ..., Y_n^{(m)})$$

Using the definition of $\Delta(n)$

Our hypotheses said that $\Delta(n) > \frac{1}{p(n)}$ for infinitely many $n$'s, hence $D'$ distinguishes $X$ and $Y$, which contradicts the hypothesis of the theorem.

3.6 Hybrid Argument: A digest
• Extreme hybrids collide with the complex ensembles
• Neighboring hybrids are easily related to the basic ensembles
• Number of hybrid is ”small” (polynomial)

4 Application: Pseudo-Random Generators
4.1 Pseudo-random Ensembles
Definition
The ensemble $X = \{X_n\}_{n \in N}$ is called pseudo random ensemble if there exists a uniform ensemble $U = \{U_{(n)}\}_{n \in N}$ such that $X$ and $U$ are indistinguishable in polynomial
4.2 Pseudo-random Generator

Definition

A pseudo-random generator is a deterministic polynomial-time algorithm \( G \) satisfying:

- **Expansion:** There exists a function \( l : \mathbb{N} \rightarrow \mathbb{N} \) such that \( l(n) > n \) for all \( n \in \mathbb{N} \) and \( |G(s)| = l(|s|) \) for all \( s \in \{0,1\}^* \).

- **Pseudo-randomness:** The ensemble \( \{G(U_n)\}_{n \in \mathbb{N}} \) is pseudo-random.

\( l \) is called the expansion factor of \( G \).

4.3 Increasing the Expansion Factor

Given a pseudo-random generator \( G_1 \) with expansion function \( l_1(n) = n + 1 \), we construct a PRG \( G \) with arbitrary polynomial expansion factor.

**Construction**

Let \( G_1 \) be a deterministic polynomial-time algorithm mapping strings of length \( n \) into strings of length \( n + 1 \), and let \( p(.) \) be a polynomial. Define \( G(s) = \sigma_1\sigma_2...\sigma_{p(|s|)} \) where \( s_0 = s \), the bit \( \sigma_i \) is the first bit of \( G_1(s_{i-1}) \), and \( s_i \) is the \( |s| \)-bit-long suffix of \( G_1(s_{i-1}) \) for every \( 1 \leq i \leq p(|s|) \).

\[
G(s_0) \leftarrow \sigma_1\sigma_2...\sigma_{p(n)} \\
\text{Let } s_0 = s \text{ and } n = |s| \\
\text{for } i = 1 \text{ to } p(n) \text{ do} \\
\quad \sigma_i s_i \leftarrow G_1(s_{i-1}) \{ \text{where } \sigma_i \in \{0,1\} \text{ and } |s_i| = |s_{i-1}| \} \\
\text{end for} \\
\text{Output } \sigma_1\sigma_2...\sigma_{p(n)}
\]

4.4 Application of Hybrid Argument

**Theorem 2**

Let \( G_1, p(.) \), and \( G \) defined as in previous construction such that \( p(n) > n \). If \( G_1 \) is a PRG the \( G \) is also a PRG.

Proof uses an hybrid argument.

Intuitively, we can see that each application of \( G_1 \) can be replaced by a random process. The indistinguishability of each applications of \( G_1 \) implies that polynomially many applications of \( G_1 \) are indistinguishable from a random process.
4.5 Idea of the proof

To the contrary, suppose $G$ is not a PRG then $\{G(U_n)\}_{n \in \mathbb{N}}$ and $\{U_p(n)\}_{n \in \mathbb{N}}$ are indistinguishable, i.e.

$$\Delta(n) = |Pr[D(G(U_n))] = 1] - Pr[D(U_p(n))]| > \frac{1}{q(n)}$$

It will contradict the fact that $G_1$ is a PRG.

Hybrid Term

We define $\forall k, 0 \leq k \leq p(n)$

$$H^k_n = U^{(1)}_{k}.pref_{p(n)-k}(G(U^{(2)}_n))$$

where $U^{(1)}_k$ and $U^{(2)}$ are independent random variables

4.6 Other Representation of $H^k_n$

It is clear that:

- $H^0_n = G(U_n)$
- $H^{p(n)}_n = U_{p(n)}$

4.7 Proof

Idea: If an algorithm $D$ can distinguish extrem hybrid, it can do it for two neighboring hybrids.

By construction

$$pref_{j+1}(G(x)) = pref_{j}(G_1(x)).pref_{j}(G(suff_{p}(G_1(x))))$$
\[ H_k^n = U^{(1)}_k \cdot \text{pref}_{p(n)} - k - 1 (G(U_n^{(2)})) \]
\[ H_{n+1}^k = U^{(1)}_{k+1} \cdot \text{pref}_{p(n)} - (k+1) (G(U_n^{(2)})) \]

Notations
\[ f_{p(n)-k}(\alpha) = \text{pref}_1(\alpha), \text{pref}_{p(n)-k-1}(G(\text{suff}_1(G_1(U_n^{(2)})))) \]

4.8 Claims

Two Easy Claims

Claim 3
\[ H_n^k \text{ is distributed identically to } U^{(1)}_k \cdot f_{p(n)-k}(G_1(U_n^{(2)})) \]

Proof of Claim 3:
\[ H_n^k = U^{(1)}_k \cdot \text{pref}_{p(n)-k-1} + 1 (G(U_n^{(2)})) \]
\[ = U^{(1)}_k \cdot \text{pref}_1(G_1(U_n^{(2)})), \text{pref}_{p(n)-k-1}(G_1(\text{suff}_1(G_1(U_n^{(2)})))) \]
\[ = U^{(1)}_k \cdot f_{p(n)-k}(G_1(U_n^{(2)})) \]

Claim 4
\[ H_{n+1}^k \text{ is distributed identically to } U^{(1)}_k \cdot f_{p(n)-k}(G_1(U_{n+1}^{(2)})) \]
Proof of Claim 3:
\[
H_n^{k+1} = U_{k+1}^{(1)}\text{pref}_{(p(n))-(k+1)}(G(U_n^{(2)})) \\
= U_k^{(1')} U_1^{(1'')}\text{pref}_{(p(n))-(k-1)}(G(suff_{n}(U_{n+1}^{(2)}))) \\
= U_k^{(1')}\text{pref}_1(U_{n+1}^{(2')}\text{pref}_{(p(n))-(k-1)}(G(suff_{n}(U_{n+1}^{(2')}))) \\
= U_k^{(1')}f_{p(n)-k}(G(U_n^{(2')}))
\]

We derive now from \(D'\) an algorithm that distinguishes \(G_1(U_n)\) from \(U_{n+1}\).

Algorithm \(D'\)

Input \(\alpha \in \{0, 1\}^{n+1}\)

1. \(D'\) selects an integer \(k\) in \(\{0, 1, ..., p(n) - 1\}\)
2. \(D'\) selects \(\beta\) uniformly in \(\{0, 1\}^k\)
3. \(D'\) halts with output \(D(\beta.f_{p(n)-k}(\alpha))\)

4.9 Two Last Claims

Claim 5

\[
Pr[D'(G_1(U_n)) = 1] = \frac{1}{p(n)} \sum_{k=0}^{p(n)-1} Pr[D(H_n^k) = 1]
\]

Claim 6

\[
Pr[D'(U_{n+1}) = 1] = \frac{1}{p(n)} \sum_{k=0}^{p(n)-1} Pr[D(H_n^{k+1}) = 1]
\]

Proof of the two claims: By construction of \(D'\), we get for every \(\alpha \in \{0, 1\}^{n+1}\)

\[
Pr[D'(\alpha) = 1] = \frac{1}{p(n)} \sum_{k=0}^{p(n)-1} Pr[D(U_k.f_{p(n)-k}(\alpha)) = 1]
\]

Finally, we get:

\[
\Delta = |Pr[D'(G_1(U_n)) = 1] - Pr[D'(U_{n+1}) = 1]| \\
= \frac{1}{p(n)} | \sum_{k=0}^{p(n)-1} Pr[D(H_n^k) = 1] - \sum_{k=0}^{p(n)-1} Pr[D(H_n^{k+1}) = 1]| \\
= \frac{1}{p(n)} |Pr[D(G(U_n)) = 1] - Pr[D(U_{p(n)}) = 1]| \\
= \frac{\Delta(n)}{p(n)} > \frac{1}{q(n)p(n)}
\]

Then we have the contradiction, and the proof is done.
5 Conclusion

In this lecture, we have first seen some definition, to understand what is an hybrid argument in the second section. We apply then the hybrid technique to prove that we can increase the expansion factor of a pseudo-random generator, to get an another pseudo-random generator.