6.842 Randomness and Computation April 28, 2008

Lecture 21

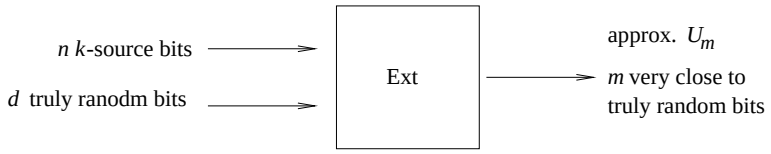
Lecturer: Ronitt Rubinfeld *Scribe: Megumi Ando*

Today's Lecture

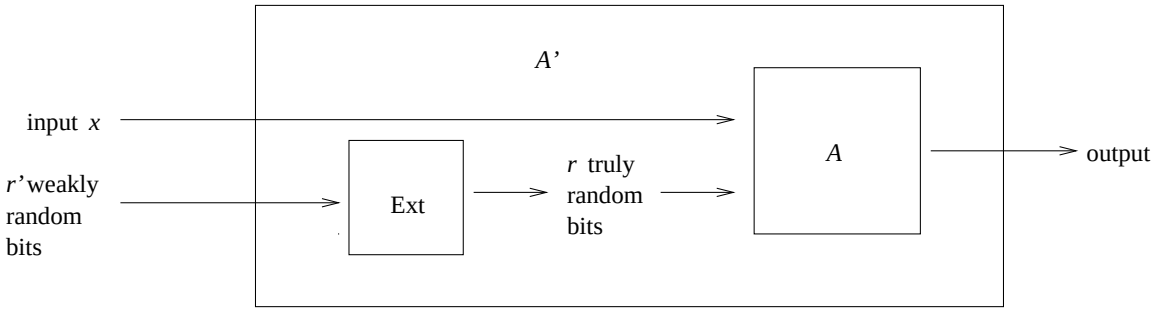
- Random Weak Sources
- Randomness Extractors

Extractor Functions

The idea of a Extractor Function is that it takes a large number of weakly random bits and outputs a smaller number of (very close to) truly random bits. So, Ext : $\{0,1\}^n \to \{0,1\}^m$, where $n > m$. For example, we often study the case for when $m = 1$.



How do we use an Extractor Function? The purpose of an Extractor Function is that it allows us to construct an algorithm that relies on only weakly random bits instead of truly random bits. From an efficient algorithm A that uses truly random bits, we would like to construct an efficient algorithm A' with the same IO behavior that uses only weakly random bits.



Desired Properties of A':

- Same IO behavior as A
- r' is poly(r)
- Only one query for r'
- Same source so as to not introduce independence

"δ-sources"

Our first example of a source for weakly random bits, is the δ-source defined as:

$$\forall i,\ \Pr[x_i = 1] = \delta_i \text{ for } 0 < \delta \leq \delta_i \leq 1 - \delta < 1.$$

Due to the independence of the bits, the δ-source is a fairly strong source. In particular, we can show that:

$$\left| \Pr\left[\bigoplus_{i=1}^{l} x_i = 1 \right] - \frac{1}{2} \right| = 2^{-\Omega(l)}.$$

"SV-sources" [Santha-Vazirani]

The SV-source is defined as:
Given δ, $\forall i$, $\forall x_1, ..., x_n \in \{0,1\}$,

$$\delta \leq \Pr[x_i = 1 | X_1 = x_1, X_2 = x_2, ..., X_{i-1} = x_{i-1}] \leq 1 - \delta.$$

Because of the bits' dependence, the SV-source is not as strong a source as the δ-source. For one, we can show that $\left| \Pr\left[\bigoplus_{i=1}^{l} x_i = 1 \right] - \frac{1}{2} \right|$ could be as large as δ. Even worse, we claim the following:

For all fixed boolean functions $\text{Ext} : \{0,1\}^n \to \{0,1\}$, there exists an SV-source such that,

$$\Pr[\text{Ext}(x) = 1] \leq \delta, \text{ or } \Pr[\text{Ext}(x) = 1] \geq 1 - \delta.$$

"k-sources"

A k-source outputs n bits such that the probability of any output is at most 2^{-k}.

Examples of k-sources:

1. *Oblivious bit fixing sources.* k iid bits and $(n-k)$ fixed bits.
2. *Adaptive bit fixing sources* k iid bits and $(n-k)$ bits depending arbitrarily on k bits.
3. SV-source with $k = \log \frac{1}{(1-\delta)^n}$.
   Probability that any specific string is output is at most $(1-\delta)^n$.
4. *Flat source.* Uniform distribution on subset $S \leq \{0,1\}^n$, such that: $|S| = 2^k$.

Theorem 1 *Every k-source is a convex combination of flat k-sources.*

Proof: $N = 2^n$. Every k-source is a distribution on N. (Think of them as vectors in R^N.)

X is a k-source if and only if:

- $\sum x(i) = 1$ (definition of probability), and
- $\forall i,\ 0 \leq x(i) \leq 2^{-k}$ (definition of k-source).

The linear constraints (the intersection of the hypercube $[0, 2^{-k}]^N$ and the hyper-plane $\sum x_i = 1$) define a convex polytope. Because the polytope is convex, any point on the polytope is a convex combination of vertices, points which make the maximal number of inequalities tight. E.g. $x(i) = 2^{-k}$ for 2^k $i's$, and $x(i) = 0$ for the others.

This is equivalent to a convex combination of flat k-sources. ■

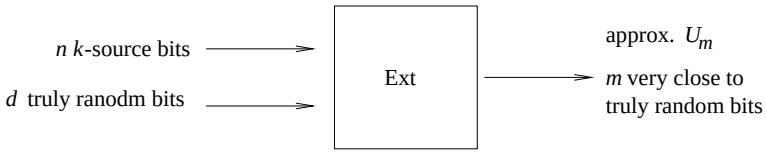
Theorem 2 *For all functions Ext* $: \{0,1\}^n \to \{0,1\}$*, there exists a* $(n-1)$*-source* X*, such that Ext*(X) *is constant.*

Proof: There exists a bit b such that $|\text{Ext}^{-1}(b)| \geq 2^{n-1}$. Let X be uniform on $\text{Ext}^{-1}(b)$. ■

Claim 3 *For all* n*, for all* $k \leq n$*, for all flat* k*-sources* X*, there exists an Extractor Function Ext* : $\{0,1\}^n \to \{0,1\}^m$ *with* $m = k - 2\log 1/\epsilon - O(1)$*, such that Ext*$(X)$ *is* ϵ*-close to* U_m*.*[1]

Seeded Extractor Functions

The function Ext : $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ϵ)-extractor if for all k-source X on $\{0,1\}^n$, $\text{Ext}(x, U_d)$ is ϵ-close to U_m.



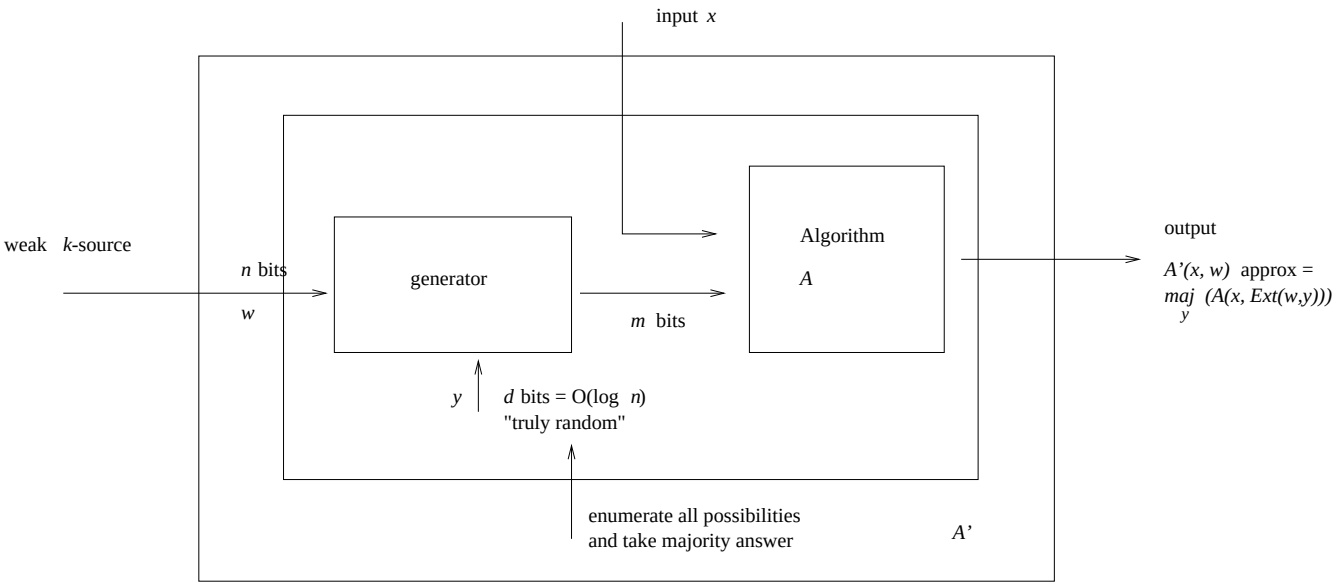
- $m > d$. We better be able to output more truly random bits than were inputted.
- $m > d + n$? Unlikely.
- $m > d + k$? Close.

Useful parameters: $k = \delta n$, $\delta = 0.01$ (a small constant), $\epsilon = 0.01$ (a small constant)

Theorem 4 *For all* x*, for all* $k \leq n$*,* $\epsilon > 0$*, there exists a* (k,ϵ)*-seed extractor Ext* $: \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$*, with* $m = k + d - 2\log(1/\epsilon) - O(1)$ *and* $d = \log(n-k) + 2\log(1/\epsilon) + O(1)$*.*

How can this theorem be applied? Given a polynomial-time A using truly random bits, we can construct a polynomial-time A' using k-source bits.

[1] U_m is the uniform distribution of $\{0,1\}^m$.



Run Time. $2^d \cdot O(\text{RT for } A)$. So, we need approx. $O(logn)$.
Behavior.

$$\begin{aligned} \Pr[A(x, U_m) \text{ incorrect}] &\equiv \gamma \\ \Pr[A(x, \text{Ext}(w, U_d)) \text{ incorrect}] &\leq \gamma + \epsilon \\ \Pr_w[A'(x, w) \text{ incorrect}] &\leq \Pr[w \text{ such that majority of } y's \text{ bad for } (x, w)] \leq 2(\gamma + \epsilon) \end{aligned}$$