Weak Random Sources

What if you don't have truly random bits?

imperfect sources:

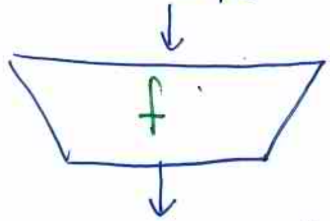
biased coin – Von Neumann trick

lsb's of fine clock

What perfect sources do we have?

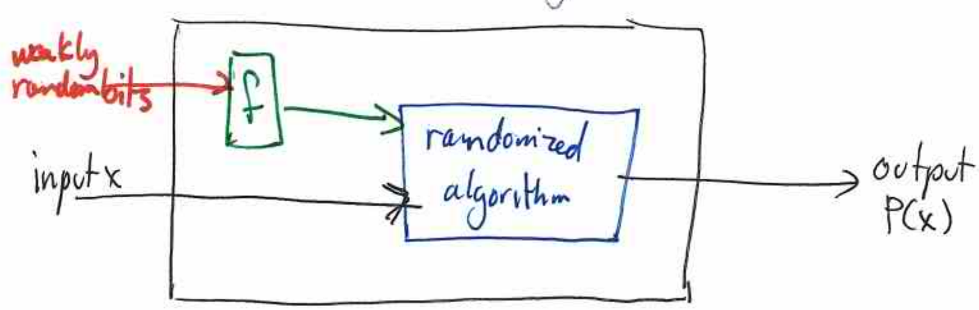
First try at defn:

$X: \{0,1\}^n$ ← comes from weak source



$f(x) \in \{0,1\}^m$ ← should be close to uniform

would allow us to use weakly random bits to simulate randomized algorithms



Examples of weakly random sources:

- biased coins

$$\forall i \quad \Pr[X_i = 1] = \delta_i \qquad 0 < \delta \le \delta_i \le 1-\delta < 1$$

parity of ℓ such bits approaches perfect coin flip exponentially fast i.e. bias $= 2^{-\Omega(\ell)}$

- "SV"-source [Santha Vazirani]

$$\forall i \quad \forall\, x_1 \cdots x_n \ \text{ + for const } \delta$$

$$\delta \le \Pr[X_i = 1 \mid X_1 = x_1,\ X_2 = x_2 \ldots X_{i-1} = x_{i-1}] \le 1-\delta$$

problem - for each deterministic f
$\exists$ SV-source which causes f to have agreement at least $1-\delta$ with either 1 or 0
so can't improve bias

- k-source
output n bits s.t. $\forall\, x = x_1 \cdots x_n \quad \Pr[\text{output } x] \le 2^{-k}$

examples:

1) k iid bits + $n-k$ fixed bits } oblivious bit fixing source

2) k iid bits + $n-k$ bits depend on 1st k } adaptive bit fixing source

3) SV source with $k = \log \frac{1}{(1-\delta)^n} = \Theta(\delta n)$

$$\Pr[\text{any } n\text{-bit string output in SV source}] \le (1-\delta)^n$$
$$= \left(2^{\log(1-\delta)}\right)^n$$
$$= 2^{-n \log \frac{1}{1-\delta}}$$

4) Unif dist on $S \subseteq \{0,1\}^n$ s.t. $|S| = 2^k$

"flat" - k-source

Flat k-sources are interesting:

<u>Claim</u> Every k-source is a convex combination of flat k-sources

i.e. $p = \sum a_i p_i$

$0 \le a_i \le 1$, $\sum a_i = 1$; p_i: flat k-source

why? see H.W.

how does this help?

any k-source = 1st select i according to a_i's
then output according to p_i

Extracting Randomness from k-sources

For every extractor fctn, $\exists$ bad k-source:

Claim $\forall$ fctns $Ext : \{0,1\}^n \to \{0,1\}$ ← this is "f" in the picture

$\exists$ $(n-1)$-source X s.t. $Ext(X)$ is constant.

Pf. HW.

$\Rightarrow$ Bad news – can't extract i.e. no one extractor works for all sources
but each source has an extractor that works for it

Claim $\forall n$ $\forall k \le n$ + $\forall$ flat k-source X,

$\exists$ fctn $Ext : \{0,1\}^n \to \{0,1\}^m$ with $\underbrace{m = k - 2\log\frac{1}{\varepsilon} - O(1)}$

note k would be perfect. This is almost there.

s.t. $Ext(X)$ is $\underbrace{\varepsilon\text{-close}}$ to U_m

statistical distance ($\frac{1}{2} \cdot L_1$ distance)

↑ uniform on m bits

Pf. H.W. (hint pick Ext randomly)

So, will change the model:
two kinds of randomness
1) truly random (a little bit)
2) weakly random (more)

Seeded Extractors

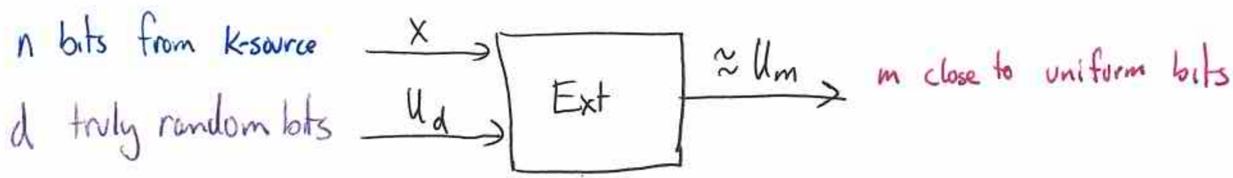
<u>def.</u> Seeded extractor

$$f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

is (k,ε)-extractor if $\forall$ k-source X on $\{0,1\}^n$, $f(X, U_d)$ is ε-close to U_m

U_d: unif on d bits

ε-close: L_1 norm



<u>Goal</u>: minimize d, maximize m

$m > d$ (otherwise trivial)

~~$m > d + n$~~ (impossible)

$m > d + k$? goal

<u>Thm</u> $\forall\ n, k \le n, \varepsilon > 0$

$\exists$ (k,ε)-extractor $f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$

with
$$m = k + d - 2\log\tfrac{1}{\varepsilon} - O(1)$$
$$d = \log(n-k) + 2\log\tfrac{1}{\varepsilon} + O(1)$$

almost reaches goal for d that is quite small

<u>Pf.</u> H.W.

Comments

1) if truly random bits only ε' close to uniform,
then get $\varepsilon' + \varepsilon$ close to U_m

2) can use to simulate randomized algorithms ← good for truly random bits
with only weak sources
without truly random bits
since d is logarithmic so
can try all possibilities

Our first extractor construction:

Leftover Hash Lemma

Explicitly construct extractor from p.i. hash $\mathcal{H}$

Use random hash fctn as a seed

e.g. can throw out last $n-\ell$ bits of output or can use different construction

Thm. (LHL)

if $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^\ell\}$ is p.i. $\left(\forall\, x \neq y,\ \forall\, a,b \quad \Pr_{h \in \mathcal{H}}[h(x) = a \wedge h(y) = b] = \frac{1}{(2^\ell)^2}\right)$

for $\ell = k - 2\log\frac{1}{\varepsilon} - O(1)$

then $\mathrm{Ext}(x, h) \equiv (h, h(x))$ # bits $= d + \ell = d + k - 2\log\frac{1}{\varepsilon} - O(1)$

$d = O(n)$ random bits

n k-source bits

is (k, ε)-extractor

Note
- large seed length
- output length is good

Preliminaries

def. $\mathrm{Coll}(p) = \sum p_i^2$ collision probability

Fact $\mathrm{Coll}(p) \le p_{max} \sum p_i$
$\le p_{max}$

Proof of LHL:

X is arbitrary k-source ← use weak random bits to pick X

$H \in_R \mathcal{H}$ ← use totally random bits to pick hash fctn H

show $(H, H(x))$ is ε-close to $U_d \times U_\ell$?

Three steps:

1) $\mathrm{Coll}((H, H(x))) \approx \mathrm{Coll}(U_d \times U_\ell)$

2) $\Leftrightarrow \quad \| (H, H(x)) - U_d \times U_\ell \|_2$ small

3) $\Leftrightarrow \quad \| (H, H(x)) - U_d \times U_\ell \|_1$ small

Step 1 the "meat"

happens iff $H = H'$ & either $X = X'$ or $X \neq X'$ but still $H(x) = H'(x') = H(x')$ (since $H = H'$)

$$\mathrm{Coll}(H, H(x)) \equiv \Pr_{X, X', H, H'} \left[(H, H(x)) = (H', H'(x')) \right]$$

$$= \underbrace{\Pr[H = H']}_{\frac{1}{2^d} \text{ since seed length is } d} \cdot \Big[\underbrace{\Pr[X = X' \mid H = H']}_{= \Pr[X = X'] \text{ since } X \text{ ind of } H \;\; \leq \frac{1}{2^k} \text{ since } k\text{-source \& using earlier note}} + \underbrace{\Pr[H(x) = H'(x) \mid X \neq X' \,\&\, H = H']}_{\leq \frac{1}{2^\ell} \text{ by def of p.i.}} \cdot \underbrace{\Pr[X \neq X' \mid H = H']}_{\leq 1} \Big]$$

$$\leq \frac{1}{2^d} \left(\frac{1}{2^k} + \frac{1}{2^\ell} \right)$$

$$= \frac{1}{2^{d+\ell}} \left(1 + \frac{1}{2^{\underbrace{2 \log \frac{1}{\varepsilon} + O(1)}_{k - \ell}}} \right) \leq \frac{1}{2^{d+\ell}} \left(1 + \varepsilon^2 \right)$$

$$= \mathrm{coll}[U_d \times U_\ell](1 + \varepsilon^2)$$

Step 2

$$\|(H, H(x)) - U_d \times U_\ell\|_2^2$$

Note: $\|p-q\|_2^2 = \Sigma(p_i - q_i)^2 = \underbrace{\Sigma p_i^2}_{\text{Coll}(p)} - 2\underbrace{\Sigma p_i q_i}_{\text{Coll}(p,q)} + \underbrace{\Sigma q_i^2}_{\text{Coll}(q)}$

$$= \text{Coll}(H, H(x)) + \text{Coll}(U_d \times U_\ell) - 2\,\text{Coll}((H, H(x)), U_d \times U_\ell)$$

Step 1 gives

$\leq \text{Coll}(U_d \times U_\ell)(1+\varepsilon^2)$

$= \frac{1}{2^{d+\ell}}(1+\varepsilon^2)$

$q = U_{d+\ell}$

r = size of domain

Lemma $\forall\, p,\ \text{Coll}(p, U_r)$

$= \Sigma p_i \cdot \frac{1}{r}$

$= \frac{1}{r} \cdot \underbrace{\Sigma p_i}_{=1}$

$= \frac{1}{r}$

so both Collision prob are $\frac{1}{2^{d+\ell}}$

$$\leq \frac{1}{2^{d+\ell}}\left[1 + \varepsilon^2 + 1 - 2\right]$$

$$= \frac{\varepsilon^2}{2^{d+\ell}}$$

Step 3 use $L_2 \leq L_1 \leq \sqrt{\text{domainsize}} \cdot L_2$

$$\|(H, H(x)) - U_d \times U_\ell\|_1 \leq \sqrt{2^{d+\ell}} \cdot \sqrt{\frac{\varepsilon^2}{2^{d+\ell}}} = \varepsilon$$

∎