Entropy, Randomness, and Information

Lecture 23

November 13, 2015

Part I

Entropy

Quote

"If only once - only once - no matter where, no matter before what audience - I could better the record of the great Rastelli and juggle with thirteen balls, instead of my usual twelve, I would feel that I had truly accomplished something for my country. But I am not getting any younger, and although I am still at the peak of my powers there are moments - why deny it? - when I begin to doubt - and there is a time limit on all of us."
–Romain Gary, The talent scout.

Entropy: Definition

Definition

The **entropy** in bits of a discrete random variable X is

$$\mathbb{H}(X) = -\sum_x \Pr\Big[X = x\Big] \lg \Pr\Big[X = x\Big].$$

Equivalently, $\mathbb{H}(X) = \mathbf{E}\Big[\lg \frac{1}{\Pr[X]}\Big]$.

Entropy intuition...

Intuition...

$\mathbb{H}(X)$ is the number of **fair** coin flips that one gets when getting the value of X.

Interpretation from last lecture...

Consider a (huge) string $S = s_1 s_2 \ldots s_n$ formed by picking characters independently according to X. Then

$$|S|\, \mathbb{H}(X) = n\mathbb{H}(X)$$

is the minimum number of bits one needs to store the string S.

Binary entropy

$$\mathbb{H}(X) = -\sum_x \Pr\Big[X = x\Big] \lg \Pr\Big[X = x\Big]$$

$\Longrightarrow$

Definition

The **binary entropy** function $\mathbb{H}(p)$ for a random binary variable that is 1 with probability p, is $\mathbb{H}(p) = -p \lg p - (1-p) \lg(1-p)$. We define $\mathbb{H}(0) = \mathbb{H}(1) = 0$.

Q: How many truly random bits are there when given the result of flipping a single coin with probability p for heads?

Binary entropy

$$\mathbb{H}(X) = -\sum_x \Pr\Big[X = x\Big] \lg \Pr\Big[X = x\Big]$$
$\Longrightarrow$

Definition

The **binary entropy** function $\mathbb{H}(p)$ for a random binary variable that is 1 with probability p, is $\mathbb{H}(p) = -p \lg p - (1-p) \lg(1-p)$. We define $\mathbb{H}(0) = \mathbb{H}(1) = 0$.

Q: How many truly random bits are there when given the result of flipping a single coin with probability p for heads?

Binary entropy

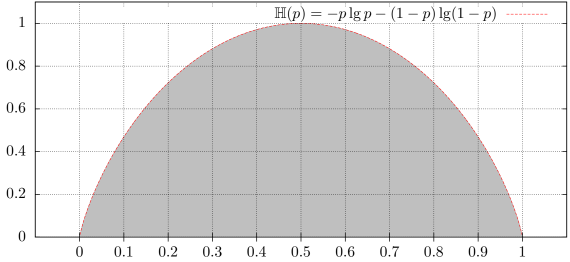
$$\mathbb{H}(X) = -\sum_x \Pr\Big[X = x\Big] \lg \Pr\Big[X = x\Big]$$
$\implies$

Definition

The **binary entropy** function $\mathbb{H}(p)$ for a random binary variable that is 1 with probability p, is $\mathbb{H}(p) = -p \lg p - (1-p) \lg(1-p)$. We define $\mathbb{H}(0) = \mathbb{H}(1) = 0$.

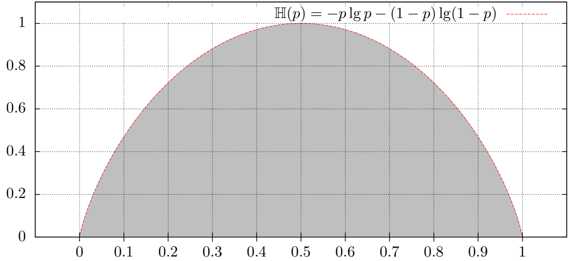
Q: How many truly random bits are there when given the result of flipping a single coin with probability p for heads?

Binary entropy: $\mathbb{H}(p) = -p \lg p - (1-p) \lg(1-p)$



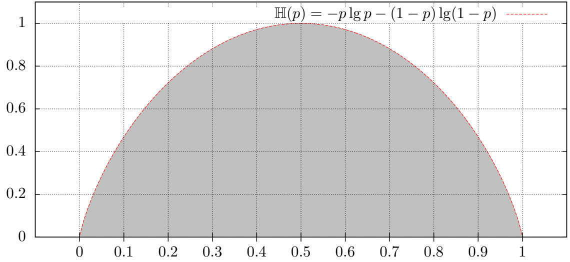
1. $\mathbb{H}(p)$ is a concave symmetric around $1/2$ on the interval $[0, 1]$.
2. maximum at $1/2$.
3. $\mathbb{H}(3/4) \approx 0.8113$ and $\mathbb{H}(7/8) \approx 0.5436$.
4. $\implies$ coin that has $3/4$ probably to be heads have higher amount of "randomness" in it than a coin that has probability $7/8$ for heads.

Binary entropy: $\mathbb{H}(p) = -p \lg p - (1 - p) \lg(1 - p)$



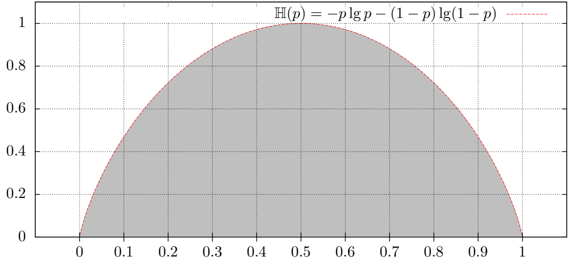
1. $\mathbb{H}(p)$ is a concave symmetric around $1/2$ on the interval $[0, 1]$.
2. maximum at $1/2$.
3. $\mathbb{H}(3/4) \approx 0.8113$ and $\mathbb{H}(7/8) \approx 0.5436$.
4. $\implies$ coin that has $3/4$ probably to be heads have higher amount of "randomness" in it than a coin that has probability $7/8$ for heads.

Binary entropy: $\mathbb{H}(p) = -p \lg p - (1-p) \lg(1-p)$



1. $\mathbb{H}(p)$ is a concave symmetric around $1/2$ on the interval $[0, 1]$.
2. maximum at $1/2$.
3. $\mathbb{H}(3/4) \approx 0.8113$ and $\mathbb{H}(7/8) \approx 0.5436$.
4. $\implies$ coin that has $3/4$ probably to be heads have higher amount of "randomness" in it than a coin that has probability $7/8$ for heads.

Binary entropy: $\mathbb{H}(p) = -p \lg p - (1-p)\lg(1-p)$



1. $\mathbb{H}(p)$ is a concave symmetric around $1/2$ on the interval $[0, 1]$.
2. maximum at $1/2$.
3. $\mathbb{H}(3/4) \approx 0.8113$ and $\mathbb{H}(7/8) \approx 0.5436$.
4. $\implies$ coin that has $3/4$ probably to be heads have higher amount of "randomness" in it than a coin that has probability $7/8$ for heads.

And now for some unnecessary math

1. $\mathbb{H}(p) = -p \lg p - (1-p) \lg(1-p)$
2. $\mathbb{H}'(p) = -\lg p + \lg(1-p) = \lg \frac{1-p}{p}$
3. $\mathbb{H}''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)}$.
4. $\implies \mathbb{H}''(p) \leq 0$, for all $p \in (0,1)$, and the $\mathbb{H}(\cdot)$ is concave.
5. $\mathbb{H}'(1/2) = 0 \implies \mathbb{H}(1/2) = 1$ $\max$ of binary entropy.
6. $\implies$ balanced coin has the largest amount of randomness in it.

And now for some unnecessary math

1. $\mathbb{H}(p) = -p \lg p - (1-p) \lg(1-p)$
2. $\mathbb{H}'(p) = -\lg p + \lg(1-p) = \lg \frac{1-p}{p}$
3. $\mathbb{H}''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)}$.
4. $\implies \mathbb{H}''(p) \leq 0$, for all $p \in (0,1)$, and the $\mathbb{H}(\cdot)$ is concave.
5. $\mathbb{H}'(1/2) = 0 \implies \mathbb{H}(1/2) = 1$ $\max$ of binary entropy.
6. $\implies$ balanced coin has the largest amount of randomness in it.

And now for some unnecessary math

1. $\mathbb{H}(p) = -p \lg p - (1-p)\lg(1-p)$
2. $\mathbb{H}'(p) = -\lg p + \lg(1-p) = \lg \frac{1-p}{p}$
3. $\mathbb{H}''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)}$.
4. $\implies \mathbb{H}''(p) \le 0$, for all $p \in (0,1)$, and the $\mathbb{H}(\cdot)$ is concave.
5. $\mathbb{H}'(1/2) = 0 \implies \mathbb{H}(1/2) = 1$ $\max$ of binary entropy.
6. $\implies$ balanced coin has the largest amount of randomness in it.

And now for some unnecessary math

1. $\mathbb{H}(p) = -p \lg p - (1-p)\lg(1-p)$
2. $\mathbb{H}'(p) = -\lg p + \lg(1-p) = \lg \frac{1-p}{p}$
3. $\mathbb{H}''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)}$.
4. $\implies \mathbb{H}''(p) \leq 0$, for all $p \in (0,1)$, and the $\mathbb{H}(\cdot)$ is concave.
5. $\mathbb{H}'(1/2) = 0 \implies \mathbb{H}(1/2) = 1$ $\max$ of binary entropy.
6. $\implies$ balanced coin has the largest amount of randomness in it.

And now for some unnecessary math

1. $\mathbb{H}(p) = -p \lg p - (1-p)\lg(1-p)$
2. $\mathbb{H}'(p) = -\lg p + \lg(1-p) = \lg \frac{1-p}{p}$
3. $\mathbb{H}''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)}$.
4. $\implies \mathbb{H}''(p) \leq 0$, for all $p \in (0,1)$, and the $\mathbb{H}(\cdot)$ is concave.
5. $\mathbb{H}'(1/2) = 0 \implies \mathbb{H}(1/2) = 1$ $\max$ of binary entropy.
6. $\implies$ balanced coin has the largest amount of randomness in it.

And now for some unnecessary math

1. $\mathbb{H}(p) = -p \lg p - (1-p)\lg(1-p)$
2. $\mathbb{H}'(p) = -\lg p + \lg(1-p) = \lg \frac{1-p}{p}$
3. $\mathbb{H}''(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)}$.
4. $\implies \mathbb{H}''(p) \leq 0$, for all $p \in (0,1)$, and the $\mathbb{H}(\cdot)$ is concave.
5. $\mathbb{H}'(1/2) = 0 \implies \mathbb{H}(1/2) = 1$ $\max$ of binary entropy.
6. $\implies$ balanced coin has the largest amount of randomness in it.

Task at hand: Squeezing good random bits...

...out of bad random bits...

1. $b_1, \ldots, b_n$: result of n coin flips...
2. From a faulty coin!
3. p: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q:** How many truly random bits can we extract?

Task at hand: Squeezing good random bits...

...out of bad random bits...

1. $b_1, \ldots, b_n$: result of n coin flips...
2. From a faulty coin!
3. p: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q:** How many truly random bits can we extract?

Task at hand: Squeezing good random bits...

...out of bad random bits...

1. $b_1, \ldots, b_n$: result of n coin flips...
2. From a faulty coin!
3. p: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q:** How many truly random bits can we extract?

Task at hand: Squeezing good random bits...

...out of bad random bits...

1. $b_1, \ldots, b_n$: result of n coin flips...
2. From a faulty coin!
3. p: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q:** How many truly random bits can we extract?

Task at hand: Squeezing good random bits...

...out of bad random bits...

1. $b_1, \ldots, b_n$: result of n coin flips...
2. From a faulty coin!
3. p: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q:** How many truly random bits can we extract?

Task at hand: Squeezing good random bits...

...out of bad random bits...

1. $b_1, \ldots, b_n$: result of n coin flips...
2. From a faulty coin!
3. p: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q:** How many truly random bits can we extract?

Task at hand: Squeezing good random bits...

...out of bad random bits...

1. $b_1, \ldots, b_n$: result of n coin flips...
2. From a faulty coin!
3. p: probability for head.
4. We need fair bit coins!
5. Convert $b_1, \ldots, b_n \implies b'_1, \ldots, b'_m$.
6. **New bits must be truly random**: Probability for head is $1/2$.
7. **Q:** How many truly random bits can we extract?

Intuitively...

Squeezing good random bits out of bad random bits...

Question...

Given the result of n coin flips: $b_1, \ldots, b_n$ from a faulty coin, with head with probability p, how many truly random bits can we extract?

If believe intuition about entropy, then this number should be $\approx n\mathbb{H}(p)$.

Back to Entropy

1. **entropy** of X is $\mathbb{H}(X) = -\sum_x \Pr\left[X = x\right] \lg \Pr\left[X = x\right]$.
2. Entropy of uniform variable..

Example

A random variable X that has probability $1/n$ to be i, for $i = 1, \ldots, n$, has entropy $\mathbb{H}(X) = -\sum_{i=1}^{n} \frac{1}{n} \lg \frac{1}{n} = \lg n$.

3. Entropy is oblivious to the exact values random variable can have.
4. $\implies$ random variables over $-1, +1$ with equal probability has the same entropy (i.e., 1) as a fair coin.

Back to Entropy

1. **entropy** of X is $\mathbb{H}(X) = -\sum_x \Pr\left[X = x\right] \lg \Pr\left[X = x\right]$.
2. Entropy of uniform variable..

Example

A random variable X that has probability $1/n$ to be i, for $i = 1, \ldots, n$, has entropy $\mathbb{H}(X) = -\sum_{i=1}^{n} \frac{1}{n} \lg \frac{1}{n} = \lg n$.

3. Entropy is oblivious to the exact values random variable can have.
4. $\implies$ random variables over $-1, +1$ with equal probability has the same entropy (i.e., 1) as a fair coin.

Back to Entropy

1. **entropy** of X is $\mathbb{H}(X) = -\sum_x \Pr\left[X = x\right] \lg \Pr\left[X = x\right]$.
2. Entropy of uniform variable..

Example

A random variable X that has probability $1/n$ to be i, for $i = 1, \ldots, n$, has entropy $\mathbb{H}(X) = -\sum_{i=1}^{n} \frac{1}{n} \lg \frac{1}{n} = \lg n$.

3. Entropy is oblivious to the exact values random variable can have.
4. $\implies$ random variables over $-1, +1$ with equal probability has the same entropy (i.e., 1) as a fair coin.

Back to Entropy

1. **entropy** of X is $\mathbb{H}(X) = -\sum_x \mathbf{Pr}\Big[X = x\Big] \lg \mathbf{Pr}\Big[X = x\Big]$.
2. Entropy of uniform variable..

Example

A random variable X that has probability $1/n$ to be i, for $i = 1, \ldots, n$, has entropy $\mathbb{H}(X) = -\sum_{i=1}^{n} \frac{1}{n} \lg \frac{1}{n} = \lg n$.

3. Entropy is oblivious to the exact values random variable can have.
4. $\implies$ random variables over $-1, +1$ with equal probability has the same entropy (i.e., 1) as a fair coin.

Back to Entropy

1. **entropy** of X is $\mathbb{H}(X) = -\sum_x \Pr\left[X = x\right] \lg \Pr\left[X = x\right]$.
2. Entropy of uniform variable..

Example

A random variable X that has probability $1/n$ to be i, for $i = 1, \ldots, n$, has entropy $\mathbb{H}(X) = -\sum_{i=1}^{n} \frac{1}{n} \lg \frac{1}{n} = \lg n$.

3. Entropy is oblivious to the exact values random variable can have.
4. $\implies$ random variables over $-1, +1$ with equal probability has the same entropy (i.e., 1) as a fair coin.

Lemma: Entropy additive for independent variables

Lemma

Let X and Y be two independent random variables, and let Z be the random variable (X, Y). Then $\mathbb{H}(Z) = \mathbb{H}(X) + \mathbb{H}(Y)$.

Proof

In the following, summation are over all possible values that the variables can have. By the independence of X and Y we have

$$\begin{aligned}
\mathbb{H}(Z) &= \sum_{x,y} \Pr\Big[(X,Y) = (x,y)\Big] \lg \frac{1}{\Pr[(X,Y) = (x,y)]} \\
&= \sum_{x,y} \Pr\Big[X = x\Big] \Pr\Big[Y = y\Big] \lg \frac{1}{\Pr[X = x] \Pr[Y = y]} \\
&= \sum_{x} \sum_{y} \Pr[X = x] \Pr[Y = y] \lg \frac{1}{\Pr[X = x]} \\
&\qquad + \sum_{y} \sum_{x} \Pr[X = x] \Pr[Y = y] \lg \frac{1}{\Pr[Y = y]}
\end{aligned}$$

Proof continued

$$
\begin{aligned}
\mathbb{H}(Z) &= \sum_x \sum_y \Pr[X = x] \Pr[Y = y] \lg \frac{1}{\Pr[X = x]} \\
&\quad + \sum_y \sum_x \Pr[X = x] \Pr[Y = y] \lg \frac{1}{\Pr[Y = y]} \\
&= \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} \\
&\quad + \sum_y \Pr[Y = y] \lg \frac{1}{\Pr[Y = y]} \\
&= \mathbb{H}(X) + \mathbb{H}(Y).
\end{aligned}
$$

■

Bounding the binomial coefficient using entropy

Lemma

$q \in [0, 1]$
nq *is integer in the range* $[0, n]$.
Then

$$\frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{nq} \leq 2^{n\mathbb{H}(q)}.$$

Proof

Holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We have

$$\binom{n}{nq} q^{nq}(1-q)^{n-nq} \leq (q + (1-q))^n = 1.$$

We also have:
$q^{-nq}(1-q)^{-(1-q)n} = 2^{n\,(-q \lg q - (1-q)\lg(1-q))} = 2^{n\mathbb{H}(q)}$, we have

$$\binom{n}{nq} \leq q^{-nq}(1-q)^{-(1-q)n} = 2^{n\mathbb{H}(q)}.$$

Proof

Holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We have

$$\binom{n}{nq} q^{nq}(1-q)^{n-nq} \leq (q + (1-q))^n = 1.$$

We also have:
$q^{-nq}(1-q)^{-(1-q)n} = 2^{n\,(-q \lg q - (1-q)\lg(1-q))} = 2^{n\mathbb{H}(q)}$, we have

$$\binom{n}{nq} \leq q^{-nq}(1-q)^{-(1-q)n} = 2^{n\mathbb{H}(q)}.$$

Proof

Holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We have

$$\binom{n}{nq} q^{nq}(1-q)^{n-nq} \leq (q + (1-q))^n = 1.$$

We also have:
$q^{-nq}(1-q)^{-(1-q)n} = 2^{n\,(-q\lg q - (1-q)\lg(1-q))} = 2^{n\mathbb{H}(q)}$, we have

$$\binom{n}{nq} \leq q^{-nq}(1-q)^{-(1-q)n} = 2^{n\mathbb{H}(q)}.$$

Proof

Holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We have

$$\binom{n}{nq} q^{nq} (1-q)^{n-nq} \le (q + (1-q))^n = 1.$$

We also have:
$q^{-nq}(1-q)^{-(1-q)n} = 2^{n\,(-q\lg q - (1-q)\lg(1-q))} = 2^{n\mathbb{H}(q)}$, we have

$$\binom{n}{nq} \le q^{-nq}(1-q)^{-(1-q)n} = 2^{n\mathbb{H}(q)}.$$

Proof continued

Other direction...

1. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
2. $\sum_{i=0}^{n} \binom{n}{i} q^i (1-q)^{n-i} = \sum_{i=0}^{n} \mu(i)$.
3. Claim: $\mu(nq) = \binom{n}{nq} q^{nq} (1-q)^{n-nq}$ largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
4. $\Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1-q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right)$,
5. sign of Δ_k = size of last term...
6. $\mathrm{sign}(\Delta_k) = \mathrm{sign}\left(1 - \frac{(n-k)q}{(k+1)(1-q)}\right)$
$= \mathrm{sign}\left(\frac{(k+1)(1-q) - (n-k)q}{(k+1)(1-q)}\right)$.

Proof continued

Other direction...

1. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
2. $\sum_{i=0}^{n} \binom{n}{i} q^i (1-q)^{n-i} = \sum_{i=0}^{n} \mu(i)$.
3. Claim: $\mu(nq) = \binom{n}{nq} q^{nq} (1-q)^{n-nq}$ largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
4. $\Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1-q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right)$,
5. sign of Δ_k = size of last term...
6. $$\mathrm{sign}(\Delta_k) = \mathrm{sign}\left(1 - \frac{(n-k)q}{(k+1)(1-q)}\right) = \mathrm{sign}\left(\frac{(k+1)(1-q) - (n-k)q}{(k+1)(1-q)}\right).$$

Proof continued

Other direction...

1. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
2. $\sum_{i=0}^{n} \binom{n}{i} q^i (1-q)^{n-i} = \sum_{i=0}^{n} \mu(i)$.
3. Claim: $\mu(nq) = \binom{n}{nq} q^{nq} (1-q)^{n-nq}$ largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
4. $\Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1-q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right)$,
5. sign of Δ_k = size of last term...
6. $\mathrm{sign}(\Delta_k) = \mathrm{sign}\left(1 - \frac{(n-k)q}{(k+1)(1-q)}\right)$

$$= \mathrm{sign}\left(\frac{(k+1)(1-q) - (n-k)q}{(k+1)(1-q)}\right).$$

Proof continued

Other direction...

1. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
2. $\sum_{i=0}^{n} \binom{n}{i} q^i (1-q)^{n-i} = \sum_{i=0}^{n} \mu(i)$.
3. Claim: $\mu(nq) = \binom{n}{nq} q^{nq} (1-q)^{n-nq}$ largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
4. $\Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1-q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right)$,
5. sign of Δ_k = size of last term...
6. $\mathrm{sign}(\Delta_k) = \mathrm{sign}\left(1 - \frac{(n-k)q}{(k+1)(1-q)}\right)$
   $= \mathrm{sign}\left(\frac{(k+1)(1-q)-(n-k)q}{(k+1)(1-q)}\right)$.

Proof continued

Other direction...

1. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
2. $\sum_{i=0}^{n} \binom{n}{i} q^i (1-q)^{n-i} = \sum_{i=0}^{n} \mu(i)$.
3. Claim: $\mu(nq) = \binom{n}{nq} q^{nq} (1-q)^{n-nq}$ largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
4. $\Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1-q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right)$,
5. sign of Δ_k = size of last term...
6. $\mathrm{sign}(\Delta_k) = \mathrm{sign}\left(1 - \frac{(n-k)q}{(k+1)(1-q)}\right)$
$= \mathrm{sign}\left(\frac{(k+1)(1-q)-(n-k)q}{(k+1)(1-q)}\right)$.

Proof continued

Other direction...

1. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
2. $\sum_{i=0}^{n} \binom{n}{i} q^i (1-q)^{n-i} = \sum_{i=0}^{n} \mu(i)$.
3. Claim: $\mu(nq) = \binom{n}{nq} q^{nq} (1-q)^{n-nq}$ largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
4. $\Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1-q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right)$,
5. sign of Δ_k = size of last term...
6. $\mathrm{sign}(\Delta_k) = \mathrm{sign}\left(1 - \frac{(n-k)q}{(k+1)(1-q)}\right)$

$$= \mathrm{sign}\left(\frac{(k+1)(1-q) - (n-k)q}{(k+1)(1-q)}\right).$$

Proof continued

Other direction...

1. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
2. $\sum_{i=0}^{n} \binom{n}{i} q^i (1-q)^{n-i} = \sum_{i=0}^{n} \mu(i)$.
3. Claim: $\mu(nq) = \binom{n}{nq} q^{nq} (1-q)^{n-nq}$ largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
4. $\Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1-q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right)$,
5. sign of Δ_k = size of last term...
6. $$\begin{aligned} \operatorname{sign}(\Delta_k) &= \operatorname{sign}\left(1 - \frac{(n-k)q}{(k+1)(1-q)}\right) \\ &= \operatorname{sign}\left(\frac{(k+1)(1-q)-(n-k)q}{(k+1)(1-q)}\right). \end{aligned}$$

Proof continued

1. $(k+1)(1-q) - (n-k)q =$
   $k + 1 - kq - q - nq + kq = 1 + k - q - nq$.
2. $\implies \Delta_k \geq 0$ when $k \geq nq + q - 1$
   $\Delta_k < 0$ otherwise.
3. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
4. $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$.
5. $\implies \mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
6. $\mu(nq)$ larger than the average in sum.
7. $\implies \binom{n}{k} q^k (1-q)^{n-k} \geq \frac{1}{n+1}$.
8. $\implies \binom{n}{nq} \geq \frac{1}{n+1} q^{-nq} (1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}$. ∎

Proof continued

1. $(k+1)(1-q) - (n-k)q =$
   $k + 1 - kq - q - nq + kq = 1 + k - q - nq.$
2. $\implies \Delta_k \geq 0$ when $k \geq nq + q - 1$
   $\Delta_k < 0$ otherwise.
3. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
4. $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$.
5. $\implies \mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
6. $\mu(nq)$ larger than the average in sum.
7. $\implies \binom{n}{k} q^k (1-q)^{n-k} \geq \frac{1}{n+1}$.
8. $\implies \binom{n}{nq} \geq \frac{1}{n+1} q^{-nq} (1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}$.

Proof continued

1. $(k+1)(1-q)-(n-k)q =$
   $k+1-kq-q-nq+kq = 1+k-q-nq$.
2. $\implies \Delta_k \geq 0$ when $k \geq nq+q-1$
   $\Delta_k < 0$ otherwise.
3. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
4. $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$.
5. $\implies \mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
6. $\mu(nq)$ larger than the average in sum.
7. $\implies \binom{n}{k} q^k (1-q)^{n-k} \geq \frac{1}{n+1}$.
8. $\implies \binom{n}{nq} \geq \frac{1}{n+1} q^{-nq}(1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}$.

Proof continued

1. $(k+1)(1-q)-(n-k)q =$
   $k+1-kq-q-nq+kq = 1+k-q-nq$.
2. $\implies \Delta_k \geq 0$ when $k \geq nq+q-1$
   $\Delta_k < 0$ otherwise.
3. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
4. $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$.
5. $\implies \mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
6. $\mu(nq)$ larger than the average in sum.
7. $\implies \binom{n}{k} q^k (1-q)^{n-k} \geq \frac{1}{n+1}$.
8. $\implies \binom{n}{nq} \geq \frac{1}{n+1} q^{-nq} (1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}$. ■

Proof continued

1. $(k+1)(1-q) - (n-k)q =$
   $k + 1 - kq - q - nq + kq = 1 + k - q - nq$.
2. $\implies \Delta_k \geq 0$ when $k \geq nq + q - 1$
   $\Delta_k < 0$ otherwise.
3. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
4. $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$.
5. $\implies \mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
6. $\mu(nq)$ larger than the average in sum.
7. $\implies \binom{n}{k} q^k (1-q)^{n-k} \geq \frac{1}{n+1}$.
8. $\implies \binom{n}{nq} \geq \frac{1}{n+1} q^{-nq} (1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}$.

Proof continued

1. $(k+1)(1-q)-(n-k)q =$
   $k+1-kq-q-nq+kq = 1+k-q-nq$.
2. $\implies \Delta_k \geq 0$ when $k \geq nq+q-1$
   $\Delta_k < 0$ otherwise.
3. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
4. $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$.
5. $\implies \mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
6. $\mu(nq)$ larger than the average in sum.
7. $\implies \binom{n}{k} q^k (1-q)^{n-k} \geq \frac{1}{n+1}$.
8. $\implies \binom{n}{nq} \geq \frac{1}{n+1} q^{-nq} (1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}$.

Proof continued

1. $(k+1)(1-q) - (n-k)q =$
   $k + 1 - kq - q - nq + kq = 1 + k - q - nq$.
2. $\implies \Delta_k \geq 0$ when $k \geq nq + q - 1$
   $\Delta_k < 0$ otherwise.
3. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
4. $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$.
5. $\implies \mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
6. $\mu(nq)$ larger than the average in sum.
7. $\implies \binom{n}{k} q^k (1-q)^{n-k} \geq \frac{1}{n+1}$.
8. $\implies \binom{n}{nq} \geq \frac{1}{n+1} q^{-nq} (1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}$.

Proof continued

1. $(k+1)(1-q) - (n-k)q =$
   $k + 1 - kq - q - nq + kq = 1 + k - q - nq$.
2. $\implies \Delta_k \geq 0$ when $k \geq nq + q - 1$
   $\Delta_k < 0$ otherwise.
3. $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$
4. $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$.
5. $\implies \mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$.
6. $\mu(nq)$ larger than the average in sum.
7. $\implies \binom{n}{k} q^k (1-q)^{n-k} \geq \frac{1}{n+1}$.
8. $\implies \binom{n}{nq} \geq \frac{1}{n+1} q^{-nq} (1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}$. ■

Generalization...

Corollary

We have:
(i) $q \in [0, 1/2] \Rightarrow \binom{n}{\lfloor nq \rfloor} \leq 2^{n\mathbb{H}(q)}$.
(ii) $q \in [1/2, 1]$ $\binom{n}{\lceil nq \rceil} \leq 2^{n\mathbb{H}(q)}$.
(iii) $q \in [1/2, 1] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{\lfloor nq \rfloor}$.
(iv) $q \in [0, 1/2] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{\lceil nq \rceil}$.

Proof is straightforward but tedious.

What we have...

1. Proved that $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$.
2. Estimate is loose.
3. Sanity check...
   (I) A sequence of n bits generated by coin with probability q for head.
   (II) By Chernoff inequality... roughly nq heads in this sequence.
   (III) Generated sequence Y belongs to $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$ possible sequences .
   (IV) ...of similar probability.
   (V) $\implies \mathbb{H}(Y) = n\mathbb{H}(q) \approx \lg \binom{n}{nq}$.

What we have...

1. Proved that $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$.
2. Estimate is loose.
3. Sanity check...
   (I) A sequence of n bits generated by coin with probability q for head.
   (II) By Chernoff inequality... roughly nq heads in this sequence.
   (III) Generated sequence Y belongs to $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$ possible sequences .
   (IV) ...of similar probability.
   (V) $\implies \mathbb{H}(Y) = n\mathbb{H}(q) \approx \lg \binom{n}{nq}$.

What we have...

1. Proved that $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$.
2. Estimate is loose.
3. Sanity check...
   (I) A sequence of n bits generated by coin with probability q for head.
   (II) By Chernoff inequality... roughly nq heads in this sequence.
   (III) Generated sequence Y belongs to $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$ possible sequences .
   (IV) ...of similar probability.
   (V) $\implies \mathbb{H}(Y) = n\mathbb{H}(q) \approx \lg \binom{n}{nq}$.

What we have...

1. Proved that $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$.
2. Estimate is loose.
3. Sanity check...
   (I) A sequence of n bits generated by coin with probability q for head.
   (II) By Chernoff inequality... roughly nq heads in this sequence.
   (III) Generated sequence Y belongs to $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$ possible sequences .
   (IV) ...of similar probability.
   (V) $\implies \mathbb{H}(Y) = n\mathbb{H}(q) \approx \lg \binom{n}{nq}$.

What we have...

1. Proved that $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$.
2. Estimate is loose.
3. Sanity check...
   (I) A sequence of n bits generated by coin with probability q for head.
   (II) By Chernoff inequality... roughly nq heads in this sequence.
   (III) Generated sequence Y belongs to $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$ possible sequences .
   (IV) ...of similar probability.
   (V) $\implies \mathbb{H}(Y) = n\mathbb{H}(q) \approx \lg \binom{n}{nq}$.

Just one bit...

question

Given a coin C with:
p: Probability for head.
$q = 1 - p$: Probability for tail.
Q: How to get **one** true random bit, by flipping C.

Describe an algorithm!

Extracting randomness...

Entropy can be interpreted as the amount of unbiased random coin flips can be extracted from a random variable.

Definition

An extraction function **Ext** takes as input the value of a random variable X and outputs a sequence of bits y, such that $\Pr\Big[\mathsf{Ext}(X) = y \;\Big|\; |y| = k\Big] = \frac{1}{2^k}$, whenever $\Pr[|y| = k] > 0$, where $|y|$ denotes the length of y.

Extracting randomness...

1. X: uniform random integer variable out of $0, \ldots, 7$.
2. $\mathsf{Ext}(X)$: binary representation of x.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
    1. X: uniform random integer variable $0, \ldots, 11$.
    2. $\mathsf{Ext}(x)$: output the binary representation for x if $0 \leq x \leq 7$.
    3. If x is between 8 and 11?
    4. Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
$\Pr\left[\mathsf{Ext}(X) = 00 \;\middle|\; |\mathsf{Ext}(X)| = 2\right] = \frac{1}{4}$,

Extracting randomness...

1. X: uniform random integer variable out of $0, \ldots, 7$.
2. **Ext**(X): binary representation of x.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   1. X: uniform random integer variable $0, \ldots, 11$.
   2. **Ext**(x): output the binary representation for x if $0 \leq x \leq 7$.
   3. If x is between 8 and 11?
   4. Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
   $\Pr\left[\textbf{Ext}(X) = 00 \;\middle|\; |\textbf{Ext}(X)| = 2\right] = \frac{1}{4}$,

Extracting randomness...

1. X: uniform random integer variable out of $0, \ldots, 7$.
2. **Ext**(X): binary representation of x.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
    1. X: uniform random integer variable $0, \ldots, 11$.
    2. **Ext**(x): output the binary representation for x if $0 \leq x \leq 7$.
    3. If x is between 8 and 11?
    4. Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
   $\Pr\left[\textbf{Ext}(X) = 00 \;\middle|\; |\textbf{Ext}(X)| = 2\right] = \frac{1}{4}$,

Extracting randomness...

1. X: uniform random integer variable out of $0, \ldots, 7$.
2. $\mathsf{Ext}(X)$: binary representation of x.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
   1. X: uniform random integer variable $0, \ldots, 11$.
   2. $\mathsf{Ext}(x)$: output the binary representation for x if $0 \leq x \leq 7$.
   3. If x is between 8 and 11?
   4. Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
   $\Pr\left[\mathsf{Ext}(X) = 00 \;\middle|\; |\mathsf{Ext}(X)| = 2\right] = \frac{1}{4}$,

Extracting randomness...

1. X: uniform random integer variable out of $0, \ldots, 7$.
2. Ext(X): binary representation of x.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
    1. X: uniform random integer variable $0, \ldots, 11$.
    2. Ext(x): output the binary representation for x if $0 \leq x \leq 7$.
    3. If x is between 8 and 11?
    4. Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
   $\Pr\left[\text{Ext}(X) = 00 \;\middle|\; |\text{Ext}(X)| = 2\right] = \frac{1}{4}$,

Extracting randomness...

1. X: uniform random integer variable out of $0, \ldots, 7$.
2. **Ext**(X): binary representation of x.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
    1. X: uniform random integer variable $0, \ldots, 11$.
    2. **Ext**(x): output the binary representation for x if $0 \leq x \leq 7$.
    3. If x is between 8 and 11?
    4. Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
$\Pr\left[\mathsf{Ext}(X) = 00 \;\middle|\; |\mathsf{Ext}(X)| = 2\right] = \frac{1}{4},$

Extracting randomness...

1. X: uniform random integer variable out of $0, \ldots, 7$.
2. $\textsf{Ext}(X)$: binary representation of x.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
    1. X: uniform random integer variable $0, \ldots, 11$.
    2. $\textsf{Ext}(x)$: output the binary representation for x if $0 \leq x \leq 7$.
    3. If x is between 8 and 11?
    4. Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
   $\Pr\left[\textsf{Ext}(X) = 00 \;\middle|\; |\textsf{Ext}(X)| = 2\right] = \frac{1}{4}$,

Extracting randomness...

1. X: uniform random integer variable out of $0, \ldots, 7$.
2. $\mathsf{Ext}(X)$: binary representation of x.
3. Def. subtle: all extracted seqs of same len have same probability.
4. Another example of extraction scheme:
    1. X: uniform random integer variable $0, \ldots, 11$.
    2. $\mathsf{Ext}(x)$: output the binary representation for x if $0 \leq x \leq 7$.
    3. If x is between 8 and 11?
    4. Idea... Output binary representation of $x - 8$ as a two bit number.
5. A valid extractor...
   $\Pr\left[\mathsf{Ext}(X) = 00 \;\middle|\; |\mathsf{Ext}(X)| = 2\right] = \frac{1}{4}$,

Technical lemma

The following is obvious, but we provide a proof anyway.

Lemma

Let x/y be a faction, such that $x/y < 1$. Then, for any i, we have $x/y < (x+i)/(y+i)$.

Proof.

We need to prove that $x(y+i) - (x+i)y < 0$. The left size is equal to $i(x-y)$, but since $y > x$ (as $x/y < 1$), this quantity is negative, as required. □

A uniform variable extractor...

Theorem

1. X: *random variable chosen uniformly at random from* $\{0, \ldots, m-1\}$.
2. *Then there is an extraction function for* X:
   1. *outputs on average at least*

$$\lfloor \lg m \rfloor - 1 = \lfloor \mathbb{H}(X) \rfloor - 1$$

   independent and unbiased bits.

A uniform variable extractor...

Theorem

1. X: *random variable chosen uniformly at random from* $\{0, \ldots, m-1\}$.
2. *Then there is an extraction function for* X:
    1. *outputs on average at least*

    $$\lfloor \lg m \rfloor - 1 = \lfloor \mathbb{H}(X) \rfloor - 1$$

    independent and unbiased bits.

A uniform variable extractor...

Theorem

1. X: *random variable chosen uniformly at random from* $\{0, \ldots, m-1\}$.
2. *Then there is an extraction function for* X:
   1. *outputs on average at least*

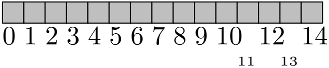
$$\lfloor \lg m \rfloor - 1 = \lfloor \mathbb{H}(X) \rfloor - 1$$

   independent and unbiased bits.

Proof

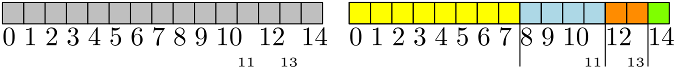
1. m: A sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$.
2. Example:
3. decomposed $\{0, \ldots, m-1\}$ into disjoint union of blocks sizes are powers of 2.
4. If x is in block 2^k, output its relative location in the block in binary representation.
5. Example: $x = 10$:
   then falls into block 2^2...
   x relative location is 2. Output 2 written using two bits,
   Output: "10".

Proof

1. m: A sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0,1\}$.
2. Example: 

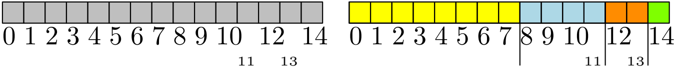
3. decomposed $\{0, \ldots, m-1\}$ into disjoint union of blocks sizes are powers of 2.
4. If x is in block 2^k, output its relative location in the block in binary representation.
5. Example: $x = 10$:
then falls into block 2^2...
x relative location is 2. Output 2 written using two bits, Output: "10".

Proof

1. m: A sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$.
2. Example: 

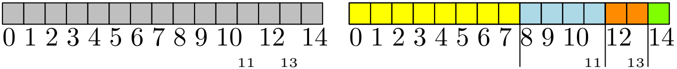
3. decomposed $\{0, \ldots, m-1\}$ into disjoint union of blocks sizes are powers of 2.
4. If x is in block 2^k, output its relative location in the block in binary representation.
5. Example: $x = 10$:
   then falls into block 2^2...
   x relative location is 2. Output 2 written using two bits,
   Output: "10".

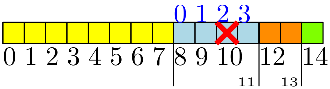
Proof

1. m: A sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$.
2. Example: 

3. decomposed $\{0, \ldots, m-1\}$ into disjoint union of blocks sizes are powers of 2.
4. If x is in block 2^k, output its relative location in the block in binary representation.
5. Example: $x = 10$:
then falls into block 2^2...
x relative location is 2. Output 2 written using two bits,
Output: "10".

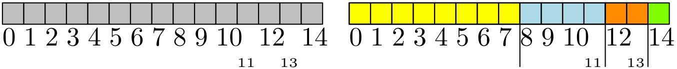
Proof

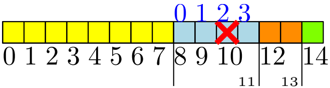
1. m: A sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$.
2. Example: 

3. decomposed $\{0, \ldots, m-1\}$ into disjoint union of blocks sizes are powers of 2.
4. If x is in block 2^k, output its relative location in the block in binary representation.
5. Example: $x = 10$: 

then falls into block 2^2...
x relative location is 2. Output 2 written using two bits,
Output: "10".

Proof

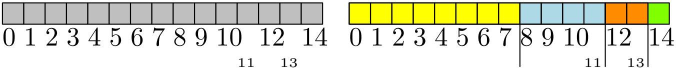
1. m: A sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$.
2. Example: 

3. decomposed $\{0, \ldots, m-1\}$ into disjoint union of blocks sizes are powers of 2.
4. If x is in block 2^k, output its relative location in the block in binary representation.
5. 

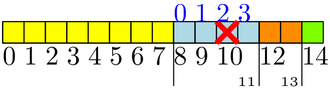
Example: $x = 10$:
then falls into block 2^2...
x relative location is 2. Output 2 written using two bits,
Output: "10".

Proof

1. m: A sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$.
2. Example:

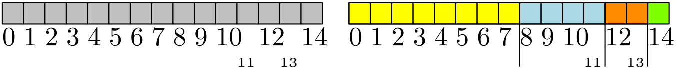


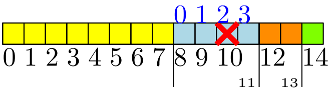
3. decomposed $\{0, \ldots, m-1\}$ into disjoint union of blocks sizes are powers of 2.
4. If x is in block 2^k, output its relative location in the block in binary representation.



5. Example: $x = 10$:
   then falls into block 2^2...
   x relative location is 2. Output 2 written using two bits,
   Output: "10".

Proof

1. m: A sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$.
2. Example: 

3. decomposed $\{0, \ldots, m-1\}$ into disjoint union of blocks sizes are powers of 2.
4. If x is in block 2^k, output its relative location in the block in binary representation.
5. Example: $x = 10$: 

then falls into block 2^2...
x relative location is 2. Output 2 written using two bits,
Output: "10".

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..
   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..
   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..

   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.

   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..

   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.

   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..
   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..

   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.

   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..
   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..
   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..
   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..
   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.
   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. Valid extractor...
2. Theorem holds if m is a power of two. Only one block.
3. m not a power of 2...
4. X falls in block of size 2^k: then output k complete random bits..

   ... entropy is k.
5. Let $2^k < m < 2^{k+1}$ biggest block.
6. $u = \lfloor \lg(m - 2^k) \rfloor < k$.

   There must be a block of size u in the decomposition of m.
7. two blocks in decomposition of m: sizes 2^k and 2^u.
8. Largest two blocks...
9. $2^k + 2 * 2^u > m \implies 2^{u+1} + 2^k - m > 0$.
10. Y: random variable = number of bits output by extractor.

Proof continued

1. By lemma, since $\frac{m-2^k}{m} < 1$:

$$\frac{m-2^k}{m} \le \frac{m-2^k + \left(2^{u+1}+2^k-m\right)}{m \qquad + \left(2^{u+1}+2^k-m\right)} = \frac{2^{u+1}}{2^{u+1}+2^k}.$$

2. By induction (assumed holds for all numbers smaller than m):

$$\mathbf{E}[Y] \ge \frac{2^k}{m}k + \frac{m-2^k}{m}\Bigl(\underbrace{\lfloor \lg(m-2^k) \rfloor}_{u} - 1\Bigr)$$

$$= \frac{2^k}{m}k + \frac{m-2^k}{m}(\underbrace{k-k}_{=0} + u - 1)$$

$$= k + \frac{m-2^k}{m}(u-k-1)$$

Proof continued

1. By lemma, since $\frac{m-2^k}{m} < 1$:
$$\frac{m-2^k}{m} \leq \frac{m-2^k+(2^{u+1}+2^k-m)}{m \qquad +(2^{u+1}+2^k-m)} = \frac{2^{u+1}}{2^{u+1}+2^k}.$$

2. By induction (assumed holds for all numbers smaller than m):
$$\mathbf{E}[Y] \geq \frac{2^k}{m}k + \frac{m-2^k}{m}\left(\underbrace{\lfloor \lg(m-2^k)\rfloor}_{u} - 1\right)$$
$$= \frac{2^k}{m}k + \frac{m-2^k}{m}(\underbrace{k-k}_{=0} + u - 1)$$
$$= k + \frac{m-2^k}{m}(u-k-1)$$

Proof continued

1. By lemma, since $\frac{m-2^k}{m} < 1$:
$$\frac{m-2^k}{m} \leq \frac{m-2^k+\left(2^{u+1}+2^k-m\right)}{m \qquad +\left(2^{u+1}+2^k-m\right)} = \frac{2^{u+1}}{2^{u+1}+2^k}.$$

2. By induction (assumed holds for all numbers smaller than m):
$$\mathbf{E}[Y] \geq \frac{2^k}{m}k + \frac{m-2^k}{m}\Bigl(\underbrace{\left\lfloor \lg(m-2^k)\right\rfloor}_{u} -1\Bigr)$$
$$= \frac{2^k}{m}k + \frac{m-2^k}{m}(\underbrace{k-k}_{=0}+u-1)$$
$$= k + \frac{m-2^k}{m}(u-k-1)$$

Proof continued

1. By lemma, since $\frac{m-2^k}{m} < 1$:
$$\frac{m-2^k}{m} \le \frac{m-2^k+\left(2^{u+1}+2^k-m\right)}{m \qquad + \left(2^{u+1}+2^k-m\right)} = \frac{2^{u+1}}{2^{u+1}+2^k}.$$
2. By induction (assumed holds for all numbers smaller than m):
$$\mathbf{E}[Y] \ge \frac{2^k}{m}k + \frac{m-2^k}{m}\left(\underbrace{\left\lfloor \lg(m-2^k)\right\rfloor}_{u} - 1\right)$$
$$= \frac{2^k}{m}k + \frac{m-2^k}{m}(\underbrace{k-k}_{=0} + u - 1)$$
$$= k + \frac{m-2^k}{m}(u-k-1)$$

Proof continued..

1. We have:

$$\mathbf{E}\left[Y\right] \geq k + \frac{m - 2^k}{m}(u - k - 1)$$
$$\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k}(u - k - 1)$$
$$= k - \frac{2^{u+1}}{2^{u+1} + 2^k}(1 + k - u),$$

since $u - k - 1 \leq 0$ as $k > u$.

2. If $u = k - 1$, then $\mathbf{E}[Y] \geq k - \frac{1}{2} \cdot 2 = k - 1$, as required.
3. If $u = k - 2$ then $\mathbf{E}[Y] \geq k - \frac{1}{3} \cdot 3 = k - 1$.

Proof continued..

1. We have:

$$\mathbf{E}\Big[Y\Big] \geq k + \frac{m - 2^k}{m}(u - k - 1)$$
$$\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k}(u - k - 1)$$
$$= k - \frac{2^{u+1}}{2^{u+1} + 2^k}(1 + k - u),$$

   since $u - k - 1 \leq 0$ as $k > u$.

2. If $u = k - 1$, then $\mathbf{E}[Y] \geq k - \frac{1}{2} \cdot 2 = k - 1$, as required.
3. If $u = k - 2$ then $\mathbf{E}[Y] \geq k - \frac{1}{3} \cdot 3 = k - 1$.

Proof continued..

1. We have:

$$\mathbf{E}\Big[Y\Big] \geq k + \frac{m - 2^k}{m}(u - k - 1)$$
$$\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k}(u - k - 1)$$
$$= k - \frac{2^{u+1}}{2^{u+1} + 2^k}(1 + k - u),$$

   since $u - k - 1 \leq 0$ as $k > u$.
2. If $u = k - 1$, then $\mathbf{E}[Y] \geq k - \frac{1}{2} \cdot 2 = k - 1$, as required.
3. If $u = k - 2$ then $\mathbf{E}[Y] \geq k - \frac{1}{3} \cdot 3 = k - 1$.

Proof continued..

1. We have:

$$\mathbf{E}\Big[Y\Big] \geq k + \frac{m - 2^k}{m}(u - k - 1)$$
$$\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k}(u - k - 1)$$
$$= k - \frac{2^{u+1}}{2^{u+1} + 2^k}(1 + k - u),$$

   since $u - k - 1 \leq 0$ as $k > u$.

2. If $u = k - 1$, then $\mathbf{E}[Y] \geq k - \frac{1}{2} \cdot 2 = k - 1$, as required.
3. If $u = k - 2$ then $\mathbf{E}[Y] \geq k - \frac{1}{3} \cdot 3 = k - 1$.

Proof continued.....

1. $\mathbf{E}[Y] \geq k - \frac{2^{u+1}}{2^{u+1}+2^k}(1+k-u)$.
   And $u - k - 1 \leq 0$ as $k > u$.
2. If $u < k - 2$ then

$$\begin{aligned} \mathbf{E}[Y] &\geq k - \frac{2^{u+1}}{2^k}(1+k-u) \\ &= k - \frac{k-u+1}{2^{k-u-1}} \\ &= k - \frac{2+(k-u-1)}{2^{k-u-1}} \\ &\geq k-1, \end{aligned}$$

since $(2+i)/2^i \leq 1$ for $i \geq 2$.

Proof continued.....

1. $\mathbf{E}[Y] \geq k - \frac{2^{u+1}}{2^{u+1}+2^k}(1+k-u)$.
   And $u-k-1 \leq 0$ as $k > u$.
2. If $u < k-2$ then

$$\begin{aligned}
\mathbf{E}[Y] &\geq k - \frac{2^{u+1}}{2^k}(1+k-u) \\
&= k - \frac{k-u+1}{2^{k-u-1}} \\
&= k - \frac{2+(k-u-1)}{2^{k-u-1}} \\
&\geq k-1,
\end{aligned}$$

since $(2+i)/2^i \leq 1$ for $i \geq 2$.

Notes

Notes

Notes

Notes