cpts 515. 10/19/2020.

How to measure "randomness" in a r.v.

Suppose that X takes vals $a_1, \ldots, a_k$ with probability $p_1, \ldots, p_k$, resp. Shannon's entropy

$$H(X) = -\sum_i p_i \log p_i \geq 0.$$

Example: fair coin X:

$$\text{prob}(X = \text{head}) = 1/2$$

$$\text{prob}(X = \text{tail}) = 1/2.$$

$$H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1 \text{ bit}$$

A memory cell that holds

1 with probability 90%

and 0 with probability 10%

is of memory size $\ll$ 1 bit !!!

In practice, the distribution of X is very hard to obtain; that is, the $p_1, \ldots, p_k$ are unknown. Then, how to estimate the randomness $H(X)$ of X?

many ways:

① sampling. (read/take prob & statistics).

② Lempel-Ziv gives the answer.

// Note: Lempel-Ziv has a condition: The sampled
// sequence is "taken" or "generated" by an ergodic
// Markov Chain.

step 1. produce a "random" sequence of values

$x_1, \ldots, x_n$; each from dist. of X.

(classic sampling.) // n shall be large — half million.

// Example. if you know X is a coin with
// unknown distribution, Then you toss it for n
// times, and obtain The outcomes $x_1, \ldots, x_n$.

step 2. Run Lempel-Ziv (LZ) on $\cancel{X_1, \ldots, X_n}$ $x_1, \ldots, x_n$.

step 3. Compression Ratio can be used to estimate $H(\mathcal{X})$.

Remark. ① LZ is the, for example, LZMA in 7z.

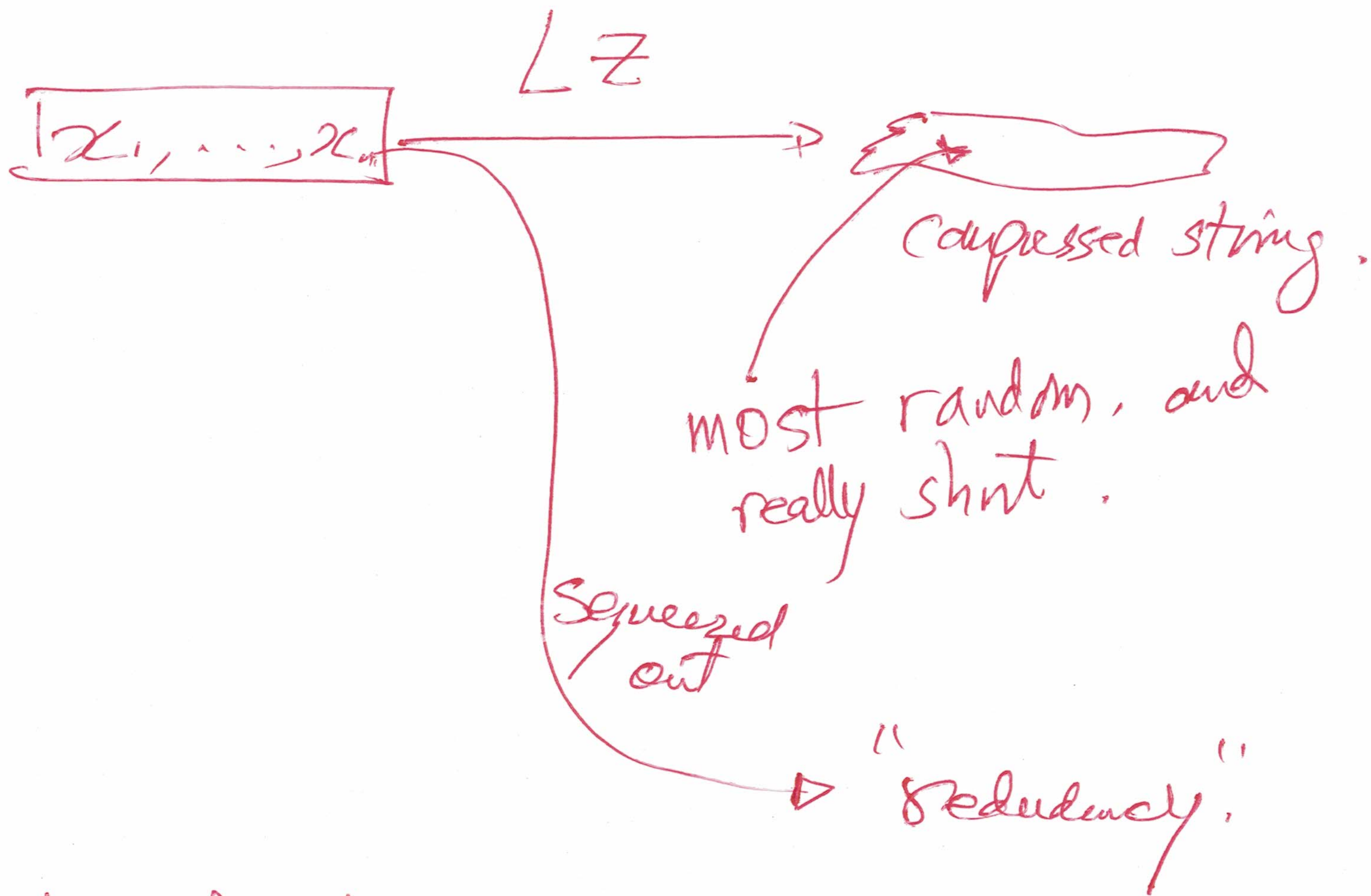
or gzip, ...

② LZ is optimal. (with side condition).

③ In a sequence

$x_1, \ldots, x_n$,

randomness (entropy) is what is left when "redundancy" is squeezed out.



④ LZ is lossless.

Read: "Losslness of communication channels modeled by transducers".

Hash, Universal Hash, Locality-Sensitive hashing & Bloom Filter.

Big picture.

a few things chosen (S) from many things (U).

Want: a short name for each thing that is chosen. How?

names are $[M]$.

Mathematical setup.

$[M] = \{0, 1, 2, \dots, M-1\}$ ← M indices.

U: universe, a very large set of <u>keys</u>.

(e.g., $U = \Sigma^k$ with size $|\Sigma|^k$).

Ex: $\Sigma = \{0, 1\}$, $k = 100$. Then

U is of all 100-bit strings. how may?

$$|U| = |\Sigma|^{100} = 2^{100}.$$

$S \subseteq U$, a subset of the Universe. (keys)

Let $|S| = N$. $(N \ll |U|)$.

Example. Consider U as the set of all 4-bit strings. So, $U = \{0000, 0001, \ldots, 1111\}$.

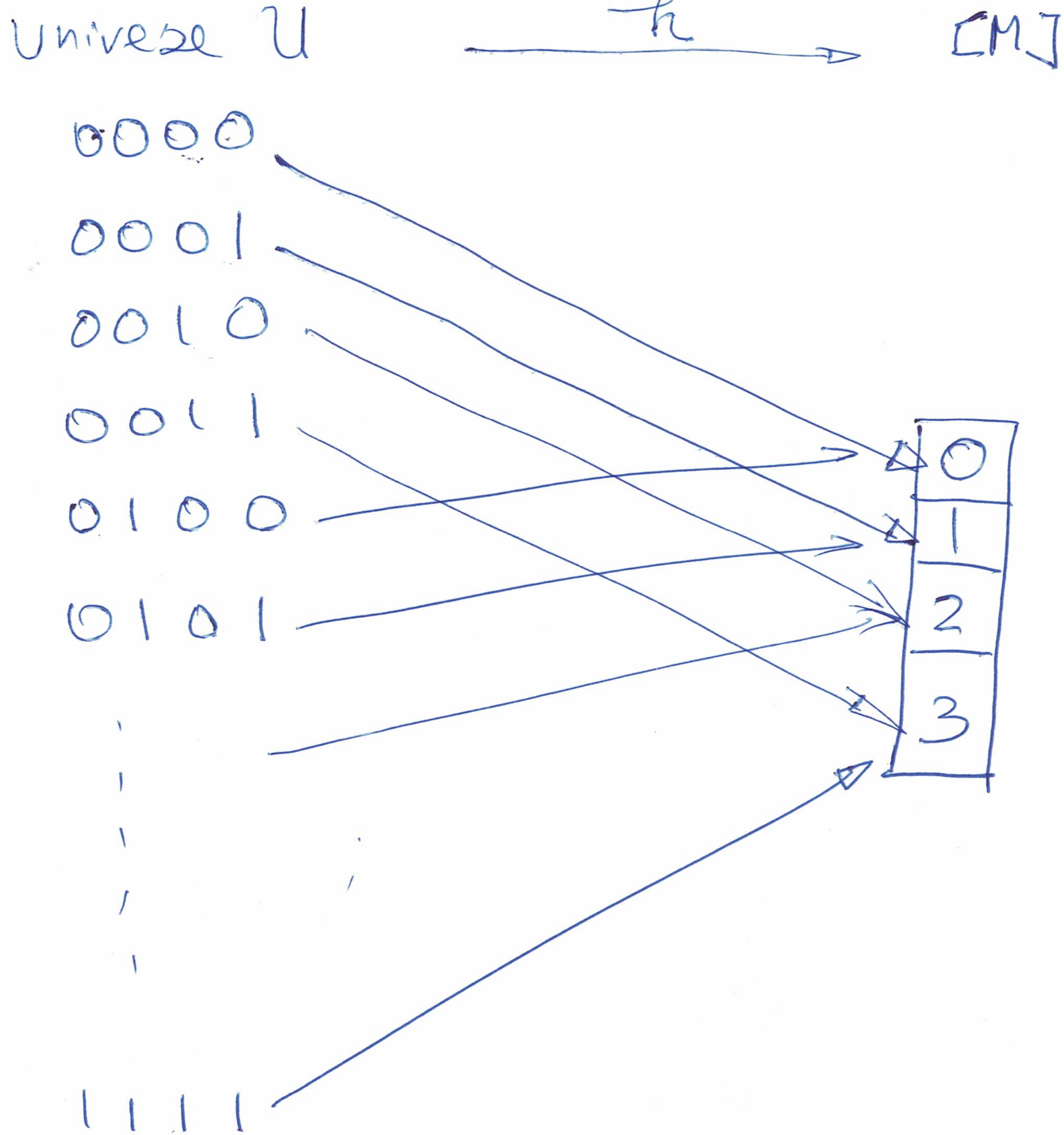
$$|U| = 2^4 = 16.$$

Let $M = 4$. (4 indices).

$h : U \longrightarrow [M] = \{0, 1, 2, 3\}$ defined as

$h(a_1 a_2 a_3 a_4) = a_3 a_4$. (taking the last two bits).

For example, $h(0010) = 10_{\text{in binary}} = 2$.



Universe U
h
[M]
0000
0001
0010
0011
0100
0101
1111
0
1
2
3

Nightmare:

take $S = \{0000, 0100, 1000\}$ with

$$N = |S| = 3.$$

we have $h(x) = 00$ for <u>each</u> $x \in S$.

That is, all elements (keys) in S share the same index. A collision refers to this scenario:

$$\underbrace{h(x)}_{\text{same index}} = h(y) \quad \text{but} \quad x = y.$$

Ideal case:

$$h(x) \neq h(y) \quad \text{whenever} \quad x \neq y.$$

for all $x, y \in S$, and for all $S \subseteq U$

when $|S| = N \approx M \ll |U|$.

Intuive Explanation:

universe.

h

In a family of 16 kids. I wan to find a way to ssign short names to each kid (e.g., each kid's name has only two bits)

$N = M$

such that kids in any subset of 4 kids would have distinct names. Possible? NO.

Theorem. Let $|\mathcal{U}| \geq (N-1)M + 1$. Then for each hash function h, there is a subset S with $|S| = N$ and

$$h(x) = h(y) \text{ for all } x, y \in S.$$

Proof. Pigeon-hole. Read book.

Reasoning: we try to ~~have~~ map $|S| = N$ keys to different locations/indices with minimal # of collisions possible. However, if h is known, we can <u>always</u> pick a set of keys, S, to let h fail terribly as in the theorem.

Solution: to make h unknown. How?

But we have to know h. What to do?

"h is known But it's random".