Tool: Error Correcting Codes

$\boxed{\text{WC hard ckts}} \longrightarrow \boxed{\text{AC hard ckts}}$

" An error correcting code maps strings into slightly larger strings in such a way that it "amplifies differences" [~hashing] such that every two distinct strings maps to far strings."

$\mapsto$

Definition
For $x, y \in \Sigma^m$ the Fractional Hamming distance between x & y denoted $\Delta(x,y)$, is equal to $\frac{1}{m} |\{i : x_i \neq y_i\}|$.

For every $\delta \in [0,1]$ a function $E : \Sigma^n \to \Sigma^m$ is an error-correcting code with distance $\delta > 0$ if $\forall\, x \neq y \in \Sigma^n$, we have $\Delta(E(x), E(y)) \geq \delta$.
The set $Im(E) = \{ E(x) : x \in \Sigma^n \}$ is the set of codewords of E.

$\mapsto$

Note: $m > n$.
Note: $|Im(E)| = 2^n$.

Note: "Strings with large HD will be mapped farther away from each other than strings with smaller HD." Can we show this? Is it false?

$\mapsto$.

Suppose H.D.$(x,y) = 1$. Then H.D.$(E(x), E(y)) = \Delta(E(x), E(y)) \cdot m$
$\geq \delta m$.

(In order that δm is significant, (for example > 1), δ can be at most $\frac{1}{m^{o(1)}}$.)

Canonical Application

$$\text{Alice} \xrightarrow{\text{noisy channel}} \text{Bob}$$

Alice: $x \in \Sigma^n$ to be sent to Bob

Say the channel may corrupt up to 10% of the bits.

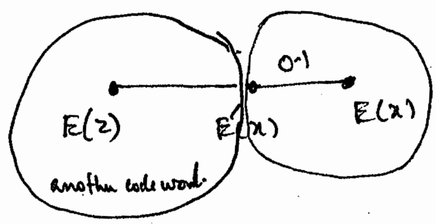
If she sends x, the only guarantee is that Bob receives $x' \in \Sigma^n$ where $\Delta(x, x') \leq 0.1$

Suppose Alice instead uses an ECC $E : \Sigma^n \to \Sigma^m$ with $\delta \geq 0.2$.

She sends $E(x)$.

Bob receives some $E(x)'$ with $\Delta\big(E(x)', E(x)\big) \leq 0.1$

Then Bob can uniquely decode $E(x)$, since it is the only encoded codeword within 0.1 fractional Hamming distance of $E(x)'$.



[How to decode fast?]

The following lemma says that good error-correcting codes exist.

Lemma (Gilbert – Varshamov bound)

$\forall\, \delta < \frac{1}{2}$ $\quad \forall^{\infty} n$ $\quad \exists\, E : \Sigma^n \to \Sigma^{\frac{n}{(1-H(\delta))}}$, there is an ECC with fractional Hamming distance δ, where $H(\delta) = \delta \log\left(\frac{1}{\delta}\right) + (1-\delta)\log\left(\frac{1}{1-\delta}\right)$.

[How fast does this grow with δ? Is it upper bounded by poly$(1/\delta)$?].

We prove a slightly weaker variant where the length of the codeword is $\frac{2n}{1-H(\delta)}$

Proof strategy
existential argument. Not that violation of conclusion leads to a contradiction.
But that a ~~[illegible]~~ code works.

Proof

We show the existence of a δ-ECC $E : \Sigma^n \to \Sigma^{\frac{2n}{1-H(\delta)}}$.

$\boxed{P(\text{no code}) \lneq 1 \quad \text{So code exists}}$

Choose E at random.

i.e. choose 2^n random strings $y_1, \ldots, y_{2^n} \in \Sigma^{\boxed{m}}$

E maps $x_1, \ldots, x_{2^n}$ to $y_1, \ldots, y_{2^n}$ respectively.

We need to show that $\Delta(y_i, y_j) < \delta$ for every $i < j$.

For every y_i, the number of strings of distance $\le \delta$ is ~~at most~~

$$\binom{m}{\lceil \delta m \rceil} \le 0.99 \; 2^{H(\delta) m} \quad \text{for all } m \text{ sufficiently large.}$$

So for every $j > i$, the probability that $\Delta(y_i, y_j) \le \delta$ is at most

$$\frac{0.99 \; 2^{H(\delta)m}}{2^m} \Leftarrow \frac{\#\text{ such strings}}{\text{total }\#\text{ of } m \text{ length strings}}$$

[no independence required]

There are at most 2^{2n} such pairs (i, j). $\quad 2^n \times 2^n$

[count the concatenations of 2 strings $i \cdot j$ $\in \Sigma^m \cdot \Sigma^m = \Sigma^{2m}$].

It suffices to show
$$0.99 \; 2^{2n} \frac{2^{H(\delta)m}}{2^m} < 1$$

i.e. $$0.99 \; 2^{2n - m + H(\delta) m} < 1$$

i.e. $$0.99 \; 2^{2n - m(1 - H(\delta))} < 1$$

i.e. $$2n - m(1 - H(\delta)) < \log(0.99)$$

i.e. $$2n - \log(0.99) < m(1 - H(\delta))$$

i.e. $$\frac{2n - \log(0.99)}{1 - H(\delta)} < m, \quad \text{which is true.}$$

$\square$

[Are there, in fact, polar codes?]

As $\delta \to 0$, there do exist codes with $m < \frac{n}{1-H(\delta)}$

$\boxed{\text{Open}}$: As $\delta \to 1/2$, is the bound in the lemma optimal?

Why half?

$\delta = 1/2$: codes exist only if m is exponentially larger than n

[Interesting threshold dichotomy: for $\delta < 1/2$, linear in n codes exist.
for $\delta = \frac{1}{2}$, all ECCs must be exponentially longer than n]

$\forall \delta > 1/2, \exists n_0\ \forall n > n_0$ no ECC exists for Σ^n.

x——x

The above lemma does not give any explicit ECC. It only shows that random ECCs in a sufficiently large space work. (Actually the proof shows that random points in $\Sigma^{\frac{1}{1-H(\delta)}}$ work. There is actually no encoding.)

⟼

We will first see explicit codes. Efficient decoding is also important for our scenario, so we will cover that afterwards.

A code $E: \Sigma^n \to \Sigma^m$, a δ-ECC is explicit if

- encoding runs in poly(n) time.
- decoding: $\exists \rho < \frac{\delta}{2}$ $\exists$ poly(m) algorithm to compute x from any y such that $\Delta(y, E(x)) < \rho$.

We see:

1. Walsh Hadamard code.
2. Reed-Solomon code
3. Reed-Muller code.
4. Concatenated code.

Summary.

	m	δ	Encoding time	Decoding Time.	Local?	List dec?
(binary) Walsh Hadamard	2^n	$\frac{1}{2}$	poly(n)			
(field) Reed Solomon	$\leq \lvert\mathbb{F}\rvert$, $\geq n$	$\left(1-\frac{n}{m}\right)$				
(field) Reed Muller	$\lvert\mathbb{F}\rvert^{l}$	$1-\frac{d}{\lvert\mathbb{F}\rvert}$				
(binary) Concatenated	$n \log \lvert\mathbb{F}\rvert \mapsto m \lvert\mathbb{F}\rvert$	$\frac{1}{2}\left(1-\frac{n}{m}\right)$.				

In Reed Muller code: d: total degree of the polynomial (max (sum of individual degrees in each monomial)) check

1. Walsh-Hadamard code. (dot product of x with all strings in Σ^n)

For $x, y \in \Sigma^n$, define $x \odot y = x_1 y_1 \oplus x_2 y_2 \oplus \cdots \oplus x_n y_n$.

The Walsh Hadamard code for $x \in \Sigma^n$, $WH: \Sigma^n \to \Sigma^{2^n}$ is defined by

$$WH(x) = \boxed{x \odot y^{(1)} \mid x \odot y^{(2)} \mid \quad \cdots \quad \mid x \odot y^{(2^n)}}$$

where $y^{(1)}, \ldots, y^{(2^n)}$ is the lexicographic ordering of Σ^n.

i.e. $WH(x) = z \in \Sigma^{2^n}$ where $z_y = x \odot y$ for every $y \in \Sigma^n$
(using strings as indices in lexicographic ordering)

Claim $WH: \Sigma^n \to \Sigma^{2^n}$ is error correcting with $\delta = \frac{1}{2}$.

Proof

First

$$WH(x \oplus y) = \boxed{(x \oplus y) \odot z^{(1)} \mid \quad \cdots \quad \mid x \oplus y \odot z^{(2^n)}}$$

$$= \boxed{(x \oplus y)_1 \cdot z_1^{(1)} \oplus (x \oplus y)_2\, z_2^{(1)} \oplus \cdots + (x \oplus y)_n\, z_n^{(1)}} \cdots \boxed{\quad}$$

$$= \boxed{(x_1 \oplus y_1) z_1^{(1)} \oplus \cdots \oplus (x_n \oplus y_n) \cdot z_n^{(1)}} \cdots \boxed{\quad}$$

$$= \boxed{[x_1 z_1^{(1)} \oplus y_1 z_1^{(1)}] \oplus \cdots \oplus [x_n z_n^{(1)} \oplus \text{[illegible]}]} \cdots \boxed{\quad} = WH(x) \oplus WH(y)$$

Thus $\forall x, y \in \Sigma^n$ $x \neq y$, the number of 1's in $WH(x) \oplus WH(y)$ is the number of positions in which $WH(x)$ & $WH(y)$ differ. Since $WH(x) \oplus WH(y) = WH(x \oplus y)$, we conclude that the number of 1 bits in $WH(x \oplus y)$ is the Hamming distance b/w $WH(x)$ & $WH(y)$

~~HD(WH(x), WH(y)) = 2 HD(x,y)~~.

Thus to show $WH(x) \oplus WH(y)$ have half ones, it suffices to show that for every nonzero string w (the zero string can never be an XOR of two distinct strings), $WH(w)$ has at least ½ ones.

Each a w is always an XOR of 2 strings & vice versa.

This follows from the $\boxed{\text{random subset principle}}$ which says that

$$\Pr\left[w \odot y = 1 \text{ for } y \sim \Sigma^n\right] = \frac{1}{2}.$$

2. Reed – Solomon Code.

The codelength of the Walsh Hadamard code is exponential in n. We can do much better for the following code, later shown to be explicit.

Definition

$\forall$ finite set Σ, & $a, b \in \Sigma^{m}$, we define $\Delta(a, b) = \frac{1}{m} \left|\{ i : a_i \neq b_i \}\right|$

Note: This is no longer restricted to binary.

We say that for $\delta > 0$, a function $E : \Sigma^n \to \Sigma^m$ is an ECC with distance δ over Σ if $\forall x, y \in \Sigma^{n}$

$$\Delta\left(E(x), E(y)\right) \geq \delta.$$

think it should be

Enlarging the alphabet simplifies the construction of ECCs which are more succinct. (See the discussion in Pages 382-383 for how alphabet sizes affect the Hamming distance in one way.)

<u>Definition.</u>

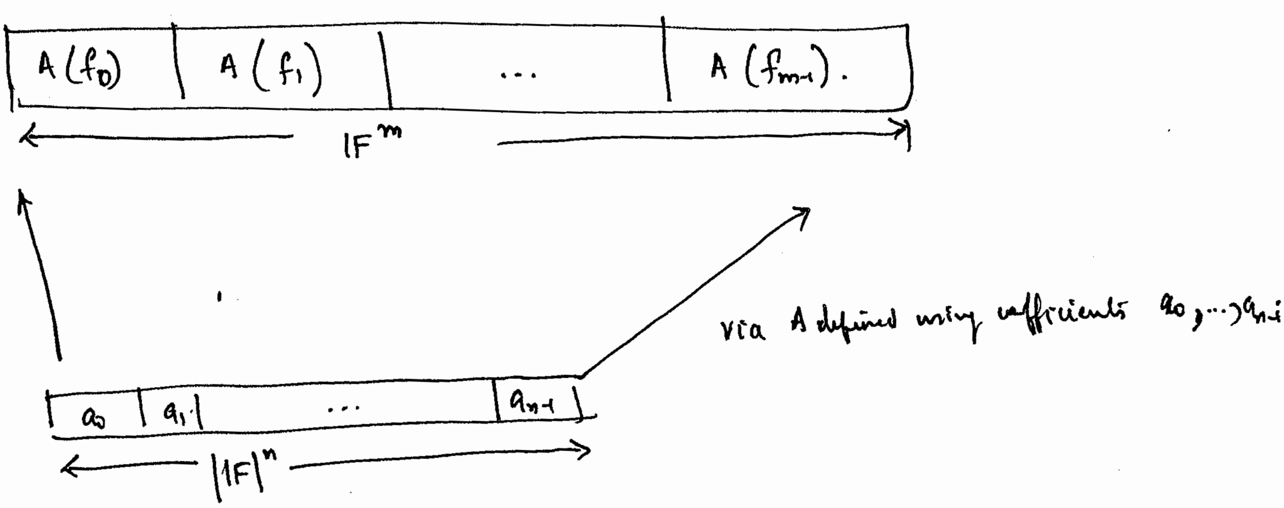
Let $\mathbb{F}$ be a finite field, $0 \leq n \leq m \leq |\mathbb{F}|$. The Reed-Solomon code from $\mathbb{F}^n \to \mathbb{F}^m$ is a function $RS: \mathbb{F}^n \to \mathbb{F}^m$ defined by

$$RS(a_0 \cdots a_{n-1}) = z_0 \cdots z_{m-1} \quad \text{where} \quad z_j = \sum_{i=0}^{n-1} a_i \, f_j^i$$

field $\mathbb{F}$ addition operation

Where f_j is the j^{th} element of $\mathbb{F}$ under some ordering.

Equivalently, given a description of an $\boxed{n-1}$? -degree univariate polynomial $A(x) = \sum_{i=\boxed{0}}^{n-1} a_i x^i$, the Reed Solomon code $RS(a_0 \cdots a_{n-1})$ is the evaluation of the polynomial A on the points $f_0, \ldots, f_{m-1}$



Lemma. The Reed Solomon code $RS: \mathbb{F}^n \to \mathbb{F}^m$ has distance $1 - \frac{n}{m}$.

Proof

RS also obeys $\forall a, b \in \mathbb{F}^n$ $RS(a+b) = RS(a) + RS(b)$:

componentwise sum

$$RS(a+b) = \sum_{i=0}^{n-1} (a+b)_i \; f_j^i$$

$$= \sum_{i=0}^{n-1} a_i f_j^i + \sum_{i=0}^{n-1} b_i f_j^i$$

$$= RS(a) + RS(b).$$

The componentwise sum of two distinct elements in $\mathbb{F}^n$ can never be 0. Every other element is the componentwise sum of some $a \neq b$, and ~~vice versa~~ conversely.

Thus ~~we~~ it suffices to show that $\forall a \in \mathbb{F}^n$ $RS(a)$ has at most n coordinates which are 0.

[because $RS(a) = RS(a+0) = RS(a) + RS(0)$]

Then $\Delta(RS(a), RS(0)) \leq \frac{n}{m}$.

Since this is the componentwise sum, $=$ # nonzero positions in a

But this follows from the fact that a nonzero $n-1$ degree polynomial has at most n distinct roots (i.e. places where it evaluates to 0.) □

true even in finite fields.

Reed Muller Codes

We now generalize both Walsh Hadamard codes & Reed Solomon codes

- generalize WH by going to a larger field (WH is a multivariate poly over $\mathbb{F}_2$)
- generalize RS by going to multivariate polynomials

	$\mathbb{F}_2$	general $\mathbb{F}$
univ.		RS
multiv.	WH	RM

general

Definition (Reed Muller Codes)

Let $\mathbb{F}$ be a finite field, and l, d be numbers with $d < |\mathbb{F}|$. The Reed Muller code with parameters $\mathbb{F}, d, l$

$$RM: \mathbb{F}^{\binom{l+d}{d}} \to \mathbb{F}^{|\mathbb{F}|^l}$$

is a function that maps every polynomial P over $\mathbb{F}$ of total degree d to the values of P on all the inputs in $\mathbb{F}^l$.

Concatenated codes

WH : drawback: exponential sized output.

RS: drawback: non binary

We now combine both to avoid either's drawbacks.

<u>Definition</u>

If $RS: \mathbb{F}^n \to \mathbb{F}^m$ is the Reed Solomon code and

$WH: \Sigma^{\log|\mathbb{F}|} \to \Sigma^{2^{\log|\mathbb{F}|}}$ is the Walsh Hadamard code, then

$WH \circ RS : \Sigma^{n \log|\mathbb{F}|} \to \Sigma^{m|\mathbb{F}|}$ is defined by:

(1) View RS as a code from $\Sigma^{n\log|\mathbb{F}|}$ to $\mathbb{F}^m$ and WH as a code from $\mathbb{F} \to \Sigma^{|\mathbb{F}|}$ using the canonical binary representation of elements in $\mathbb{F}$ as strings in $\Sigma^{\log|\mathbb{F}|}$.

(2) $\forall x \in \Sigma^{n\log|\mathbb{F}|}$ $\quad WH \circ RS(x) = WH(RS(x)_1), \ldots, WH(RS(x)_m)$

where $RS(x)_i$ represents the i^{th} symbol of $RS(x)$.

$$\begin{tikzcd}
\Sigma^{n\log|\mathbb{F}|} \arrow[r, "RS"] & \mathbb{F}^m \arrow[d] \\
x \arrow[u, "binary"] & \Sigma^{m|\mathbb{F}|}
\end{tikzcd}$$

__Claim__

Let $\delta_1 = 1 - \frac{n}{m}$ be the distance of RS and $\delta_2 = \frac{1}{2}$, of WH.

Then $WH \circ RS$ is an ECC of distance $\delta_1 \delta_2 = \frac{1}{2}\left(1 - \frac{n}{m}\right) = \frac{1}{2} - \frac{n}{2m}$.

__Proof__

Let x, y be distinct strings in $\Sigma^{n \log |\mathbb{F}|}$.

Let $x' = RS(x)$, $y' = RS(y)$, with $\Delta(x', y') \geq \delta_1$.

If $x'' = WH(x'_1); WH(x'_2), \ldots, WH(x'_m)$, (resp y'').

Suppose for some position $1 \leq i \leq m$, we have

$x'_i \neq y'_i$.

(The number of such distinct positions where they differ is $\geq \delta_1$)

For each such i, note that $WH(x'_i) \neq WH(y'_i)$, and

further, $\Delta\left(WH(x'_i), WH(y'_i)\right) \geq \frac{1}{2}$.

(← __Insight__ positionwise encoding of code's output produces [illegible] error correction)

Thus the fractional HD of the concatenated ECC = fraction of differing positions in (x', y')

$\times \frac{1}{2}$

$= \delta_1 \delta_2$

$\square$.

For every $k \in \mathbb{N}$, $\exists$ finite field $\ni |\mathbb{F}| \in [k, 2k]$ (take a prime.

It exists by Bertrand's postulate, or a power of 2 - a prime power).