Lecture 14

Procedure to test if a code is uniquely decodable:

(1) Construct a list of all (distinct) codewords. N distinct codewords

(2) Examine all pairs of the codeword, to see if a codeword is a prefix of another codeword.

$$\binom{N}{2} = \frac{N \times (N-1)}{2}$$

(3) If such a pair is found, then add the dangling suffix to the list, unless the you have added the same dangling suffix to the list in previous iterations.

(4) Repeat the above procedure using the longer list, and continue in this fashion until one of the following cases happens:

Either
(a) Get a dangling suffix that is **a codeword in the original list** => The code is NOT uniquely decodable;
Or,
(b) There is no more dangling suffix. => The code is uniquely decodable.

Example:
Code: {0, 01, 10} , dangling suffix between {0, 01} is '1'

$$\begin{array}{ccc} 0 & 01 & 10 \\ \downarrow & \downarrow & \downarrow \\ a_1 & a_2 & a_3 \end{array}$$

$\Downarrow$

{0, 01, 10, 1}

dangling suffix between {10, 1} is '0', which is an existing codeword in the original list

$\Downarrow$

The code is NOT uniquely decodable.

Another example:

Code: {0, 01, 11}

$\uparrow$ $\uparrow$ $\uparrow$

a_1 a_2 a_3

Procedure:

The pair {0, 01} results in a dangling suffix of '1', which is added to the list:

{0, 01, 11, 1}, which results in a dangling suffix of '1', between {11, 1}, but '1' has been added in the previous step.

(b) There is no more dangling suffix. => The code is uniquely decodable.

Yet another example:

Code: {1, 01, 000, 0010, 0011} => No dangling suffix => Code is uniquely decodable.

- Instantaneous Codes
  During the decoding phase, we do not have to wait until the beginning of the next codeword before knowing the current codeword is complete.

Example: Alphabet = {a1, a2, a3, a4}

Letter	Code 1	Code 2	Code 3
a_1	0	0	0
a_2	1	10	01
a_3	00	110	011
a_4	11	111	0111

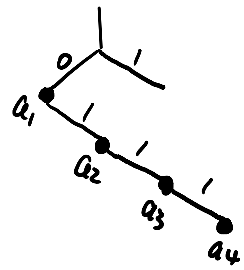
Code 1: {0, 1, 00, 11} is NOT uniquely decodable, using the procedure discussed earlier.

Code 2: {0, 10, 110, 111} is uniquely decodable, and instantaneous.

'1' '11'

Code 3: {0, 01, 011, 0111} is **uniquely decodable**

'1' '11' '1' '111'



{0, 01, 011, 0111, 1, 11, 111}

Code 3 is NOT instantaneous.
Reason: suppose there is coded bitstream: 010, which can be decoded as: $a_2\ a_1$
Thus, the decoder has to wait until the beginning of the next codeword before knowing the current codeword is complete.

- **Prefix** code (where no codeword is a prefix of another codeword, thus there is no dangling suffix)
  For example:

  Code 2: {0, 10, 110, 111} is **uniquely decodable**, and **instantaneous**.

Next, answer the following question:
Are we losing on the *coding efficiency* (in terms of average codeword length) if we restrict ourselves to prefix codes?

Kraft-McMillan Inequality (K-M inequality):

(1) If a code C is uniquely decodable, then K(C) = $\sum_{i=1}^{N} 2^{-l_i} \leq 1$
Where N is the number of codewords in code C, and $l_1, l_2, \cdots, l_N$
Are the codeword lengths.

(2) If K(C) ≤ 1, then we can always construct a prefix code with codeword lengths being $l_1, l_2, \cdots, l_N$.

Letter	Code 1	Code 2	Code 3
a_1	0	0	0
a_2	1	10	01
a_3	00	110	011
a_4	11	111	0111

$K(\text{Code 1}) = 2^{-1} + 2^{-1} + 2^{-2} + 2^{-2} = 1.5 > 1$, NOT uniquely decodable!

$K(\text{Code 2}) = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = 1 \leqslant 1$, **uniquely decodable**

$K(\text{Code 3}) \begin{cases} = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} < 1 \\ = 2^{-4}\underbrace{(2^3 + 2^2 + 2 + 1)}_{8+4+2+1} = \frac{15}{16} \end{cases}$, **uniquely decodable**

Midterm Exam:
March 3, 2025 (Monday) 1:00 pm – 2:20 pm

Scope: Lecture 1 -- Lecture 14 (inclusive), HW1 -- HW3.

Closed-book, closed-notes, no internet search; no Matlab.

Bring a calculator (log function)
You can bring a formula sheet (letter-sized, one page; can be two-sided).

- Information-theoretic metrics, H(X1,X2), H(X1|X2), …….
- Discrete-time first-order Markov chain.
- Check if a code is unique decodable, instantaneous, prefix code, etc.
- Huffman code