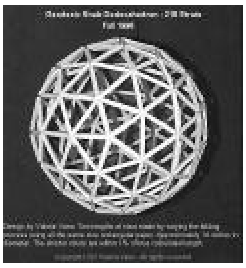
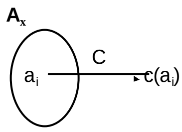
# The Revenge of a Student - Symbol Codes



## Symbol codes

- Notation: $\{0,1\}^{+}=\{0,1,00,01,10,11,000,...\}$
- A symbol code C is a mapping from $\mathbf{A_x}$ to $\{0,1\}^{+}$

$c^{+}(x_1x_2x_3...x_N) = c(x_1)c(x_2)c(x_3)...\ c(x_N)$




$l(x) = |x|$



## Decoding of symbol codes

- A code C(X) is uniquely decodable if

$$\forall \mathbf{x},\mathbf{y} \in A_X^{+}, \mathbf{x} \neq \mathbf{y} \Rightarrow c^{+}(\mathbf{x}) \neq c^{+}(\mathbf{y})$$

- A code C(X) is a prefix code if no codeword is a prefix of any other codeword
- The expected length L(C,X) of a symbol code C for ensemble X is

$$L(C,X) = \sum_{x \in A_x} P(x) l(x)$$



## Example

$\mathbf{A_x} = \{1,2,3,4\}$, $P_X = \{1/2,1/4,1/8,1/8\}$

C: c(1) = 0, c(2) = 10, c(3) = 110, c(4) = 111

The entropy of X is 1.75 bits: L(C,X) is also 1.75 bits

Obs!

$l_i = \log_2(1/p_i),\ p_i = 2^{-l_i}$



## Kraft inequality

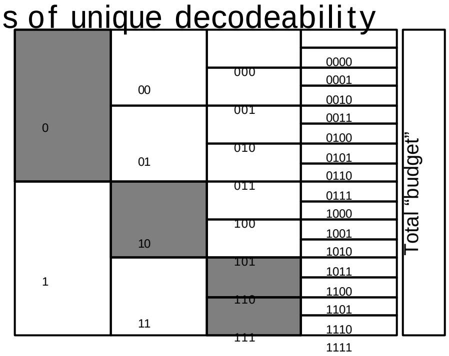
- Given a list of integer $\{l_i\}$, does there exist a uniquely decodable code with $\{l_i\}$?
- "Market model": total budget 1; cost per codeword of length *l* is $2^{-l}$.

**Kraft inequality**: For any uniquely decodeable code C over the binary alphabet {0,1}, the codeword lengths must satisfy: $\sum_i 2^{-l_i} \leq 1$

Conversely, given a set of codeword lengths that satisfy this inequality, there exists a uniquely decodable prefix code with these codelengths.



## Limits of unique decodeability



## What can we hope for?

**Lower bound on expected length**: The expected length L(C,X) of a uniquely decodable code is bounded below by H(X).

**Compression limit of symbol codes:** For an ensemble X there exists a prefix code

$$H(X) \le L(C,X) < H(X) + 1.$$



## "Proof-map" of the lower bound

Define $q_i \equiv 2^{-l_i}/z$, where $z = \sum_i 2^{-l_i}$ (By the definition of log)

Thus $l_i = \log 1/q_i - \log z$ (Substitution)

$$L(C,X) = \sum_i p_i l_i = \sum_i p_i \log 1/q_i - \log z$$

$$\ge \sum_i p_i \log 1/p_i - \log z$$

(Gibbs inequality)

$$\ge H(X)$$

$-\log z \ge 0$ (Kraft inequality)



## (What happens if we use the "wrong" code?)

Assume the "true probability distribution" is $\{p_i\}$. If we use a complete code with lengths $l_i$, they define a probabilistic model $q_i = 2^{-l_i}$. The average length is

$$L(C,X) = H(X) + \sum_i p_i \log p_i / q_i$$

Kullback-Leibler divergence $D_{KL}(p||q)$

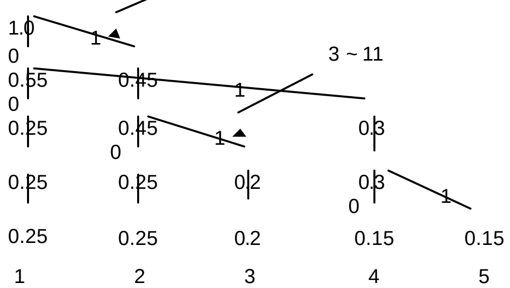


## "Optimal" symbol code: Huffman coding

- Take two least probable symbols in the alphabet as defined by $\{p_i\}$.
- Combine these symbols into a single symbol, $p_{new} = p_1 + p_2$. Repeat (until one symbol)



## Huffman in practice

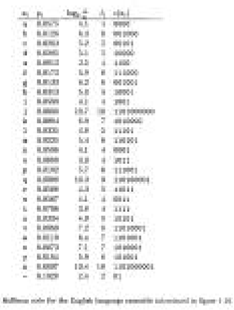





## Huffman for the Linux manual

L(C,X) = 4.15 bits

H(X) = 4.11 bits

## Why is this not the end of the story?

- Adaptation: what if the ensemble X changes? (as it does…)
  - ✓calculate probabilities in one pass
  - ✓communicate code + the Huffman-coded message
- "The extra bit": what if H(X) ~1 bit?
  - ✓Group symbols to blocks and design a "Huffman block code"