

CS 240 – Data Structures and Data Management

Module 10: Compression - Enriched

T. Biedl É. Schost O. Veksler

Based on lecture notes by many previous cs240 instructors

David R. Cheriton School of Computer Science, University of Waterloo

Winter 2021

Outline

- 1 Compression
 - Arithmetic compression

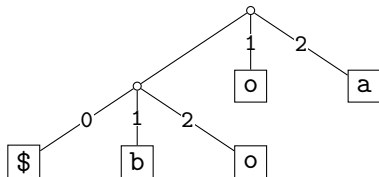
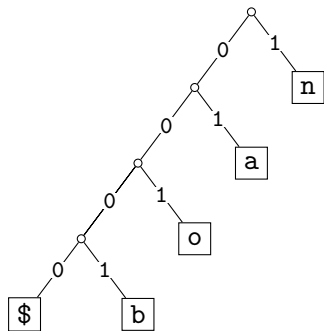
Outline

- 1 Compression
 - Arithmetic compression

Huffman with a different base

Example text: nobanana\$, $\Sigma_S = \{\$, b, o, a, n\}$

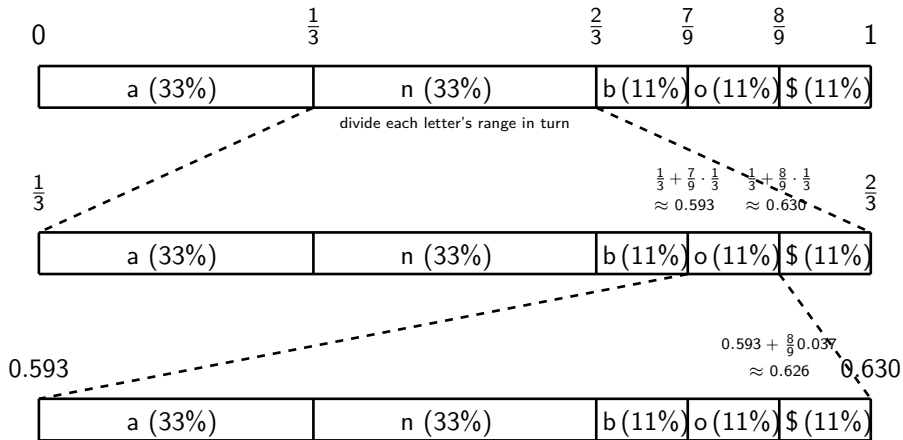
Character frequencies: \$: 1, b : 1, o : 1, a : 3, n : 3



$$\left(\underbrace{10010001011011010000}_{20 \text{ bits}} \right) \text{ vs. } (202011212100)_3 = \left(\underbrace{1100000111110010110}_{19 \text{ bits}} \right)$$

Arithmetic compression

Subdivide $[0, 1]$ by frequencies of letters.



Any number in interval $(0.626, 0.63)$ represents 'no\$'.

Arithmetic compression code

ArithmeticEncoding(f, S)

f : frequencies of Σ_S , S : text to encode (ends with \$)

1. $I \leftarrow [0, 1]$ // interval
2. **for** $i = 0 \dots |S| - 1$
3. subdivide I relative to frequencies f
4. $I \leftarrow$ interval that corresponds to $S[i]$
5. $C \leftarrow$ number in I that needs few bits
6. **return** C (or its binary encoding)

ArithmeticDecoding(f, C)

f : frequencies of Σ_S , C : number to decode

1. $S \leftarrow$ empty string, $I \leftarrow [0, 1]$
2. **repeat**
3. subdivide I relative to frequencies f
4. $S.append$ (character whose interval contains C)
5. **until** $c = \$$
6. **return** S