Permutation coding using divide-and-conquer strategy

Kun Tu* and Dariusz Puchala**

*School of Mathematical Sciences
Yangzhou University
Yangzhou, China

**Institute of Information Technology
Lodz University of Technology
Lodz, Poland

Introduction

In computer science **permutations** are used in pattern searching, duplicate documents detection and data compression tasks.

For this reason, **redundancy reduction** leading to a concise representation of permutations is very important.

In this paper, we introduce a novel method for **succinct representation** of permutations where the average number of bits per element is **close to the theoretic limit**.

The theoretical limit

We consider n-element permutations $\pi(n)$ for n being **integer powers of 2**. Theoretical entropy limit can be calculated as (**equal probabilities** of $\pi(n)$):

$$H(n) = \frac{1}{n}\log_2(n!).$$

And using Stirling's approximation formula we can write:

$$H(n) = \log_2 n - \log_2\left(\frac{n}{\sqrt[n]{n!}}\right) \approx \log_2 n - 1.443.$$

The proposed method

The proposed method is based on **divide-and-conquer strategy**.

The outline of the method.

1. The method follows the divide-and-conquer strategy and at each stage the currently considered permutation is divided into two equal halves (bins).
2. Binary encoding is used to describe elements-to-bins assignment (’0’-first, ’1’-second bin).
3. Depending on a form of permutation some bits can be omitted, which leads to succinct representation.

The proposed method

For example, we have the following permutation $\pi_2 = (0,2,1,3,7,6,4,5)$ we want to encode.

1. $\pi_1 = (0,1,2,3,4,5,6,7)$.

$(0,1,2,3,4,5,6,7)$

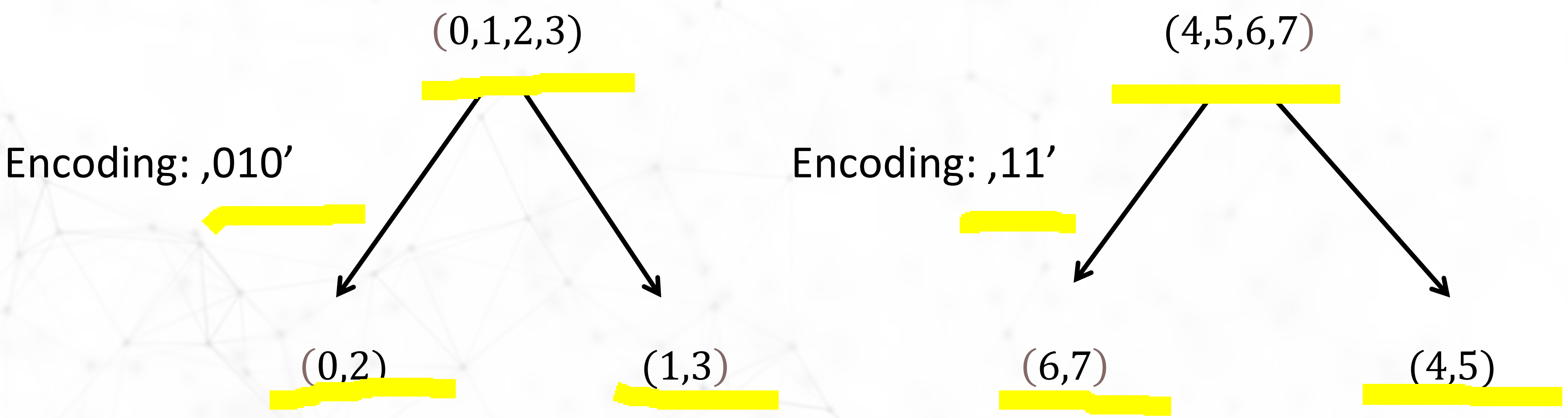
Encoding: ‚0000’

$(0,1,2,3)$ $(4,5,6,7)$

The proposed method

2.

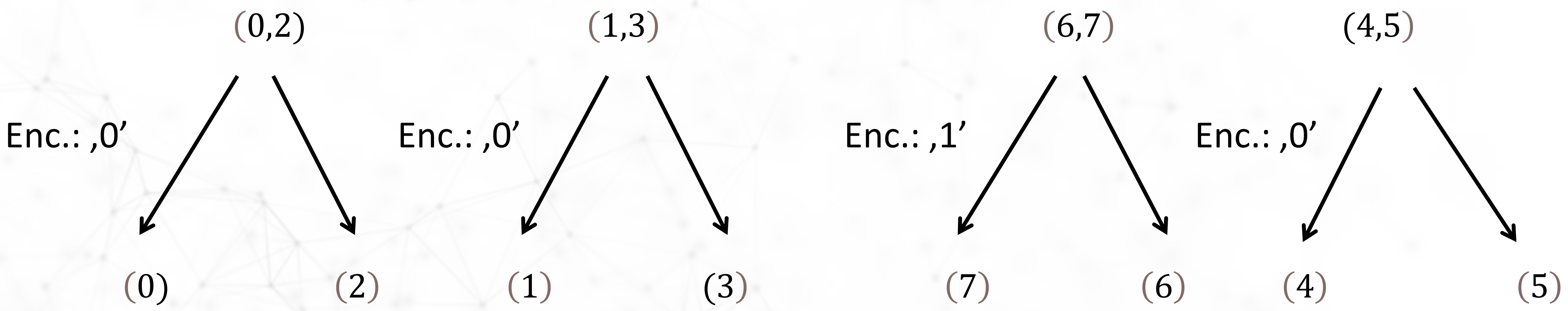
$$\pi_1 = (0,1,2,3|4,5,6,7).$$



The proposed method

3.

$$\pi_1 = (0,2|1,3|6,7|4,5).$$



$$\pi_2 = (0,2,1,3,7,6,4,5)$$

The proposed method

Hence, the following permutation $\pi_2 = (0,2,1,3,7,6,4,5)$ can be encoded as the sequence of concatenated stage encodings.

We have than: $C_2 = 0000010110010$. This gives a numer of 13 bits. The numer of bits can change depending on permutation.

This encoding is **unique**.

The following encodings can be **concatenated** and written as a stream of bits/bytes.

The bounds and average number of bits

For the proposed method the minimum number of bits equals:

$$G^{min}(n) = \frac{1}{2}\log_2 n\,.$$

The maximum number of bits equals:

$$G^{max}(n) = \log_2 n - \left(1 - \frac{1}{n}\right).$$

The average number of bits (with $n \to \infty$):

$$G(n) = \log_2 n - \left(\sum_{i=0}^{\log_2 n - 1} \frac{1}{2^l + 1}\right) \approx \log_2 n - 1.269.$$

Experimental results

Comparison of the theoretical $H(n)$ limit with the efficiency of the proposed method $G(n)$.

Table 1: The values of $G(n)$ and $H(n)$ calculated for different values of n.

n	2	4	8	16	32	64
$G(n)$	0.5	1.167	1.967	2.856	3.797	4.766
$H(n)$	0.5	1.146	1.912	2.766	3.677	4.625

Thank you for watching our video.