UNIVERSITY OF LIVERPOOL

LEVERHULME TRUST

The Leverhulme Research Centre for Functional Materials Design

# Multidimensional Necklaces:

## Enumeration, Generation, Ranking and Unranking

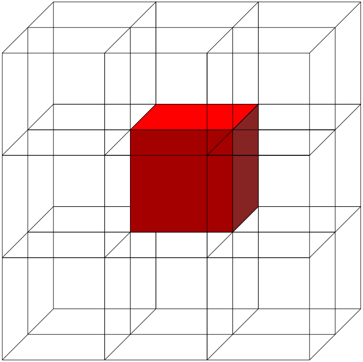
**Duncan Adamson**, Argyrios Deligkas, Vladimir V. Gusev and Igor Potapov
University of Liverpool, Department of Computer Science

April 10, 2020

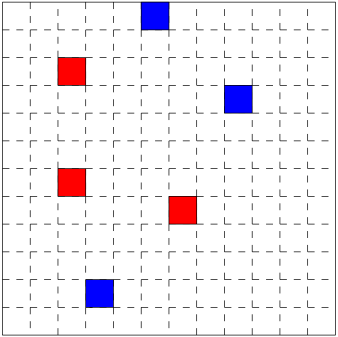
diamond Hartree Centre Imperial College London ICNS King Abdullah University of Science and Technology University of Southampton

# Crystals

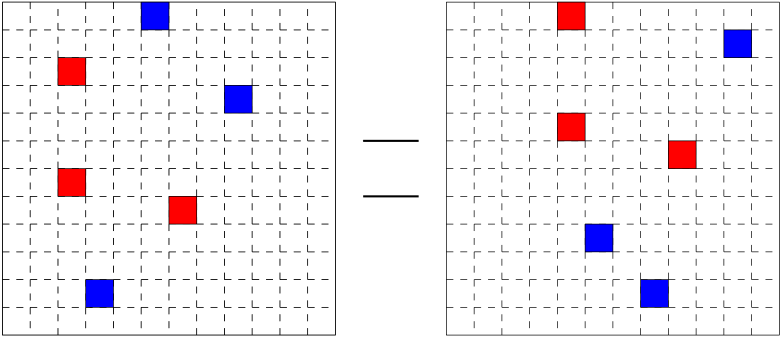
Crystals are a fundamental material structure defined by an infinitely repeating unit cell.



# Unit Cells



# Unit Cells



# The Problem

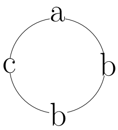
- Given a unit cell in $d$ dimensions of size $N_1 \times N_2 \times \ldots \times N_d$, and $k-1$ types of ions, how many ways of arraigning ions in the cell are there up to translational equivalence?
- We assume that the space is discrete and there is no limits on how many of each type of ion can be placed.

# The Problem

- Given a unit cell in $d$ dimensions of size $N_1 \times N_2 \times \ldots \times N_d$, and $k - 1$ types of ions, how many ways of arraigning ions in the cell are there up to translational equivalence?
- We assume that the space is discrete and there is no limits on how many of each type of ion can be placed.
- **Idea:** Represent each unit cell as a multidimensional string.
- Represent each ion as a character plus one for blank space.
  - This gives an alphabet of size $k$.

# Necklaces

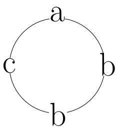
- In one dimension, counting the number of arraignments corresponds to counting the number of **Necklaces** of size $N_1$ over an alphabet of size $k$.
  - A **necklace** is the lexicographically smallest representation of a cyclic string.
  - This means every necklace is unique under cyclic rotation.




**abbc**
bbca
bcab
cabb

# Necklaces

- In one dimension, counting the number of arraignments corresponds to counting the number of **Necklaces** of size $N_1$ over an alphabet of size $k$.
  - A **necklace** is the lexicographically smallest representation of a cyclic string.
  - This means every necklace is unique under cyclic rotation.




**abbc**
bbca
bcab
cabb

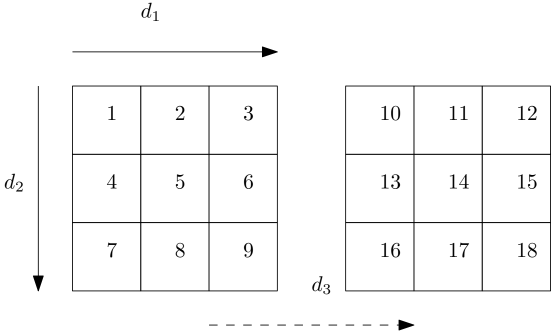
- **Problem**
  - Crystals are not 1d.
  - How can we generalise the concept of necklaces to multiple dimensions?

# What is known about necklaces?

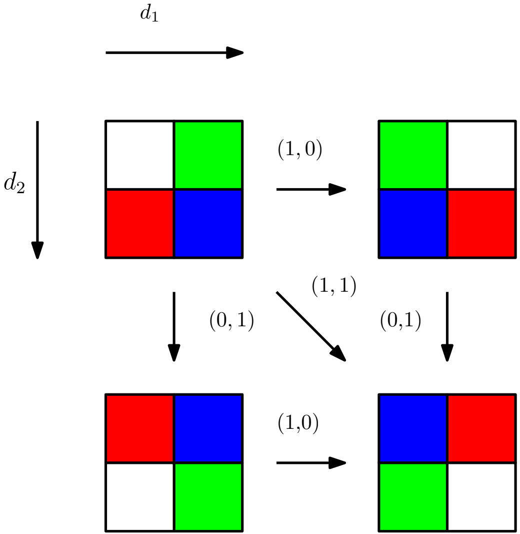
- Necklaces are a heavily studied combinatorial object, the main results are:
  - **Enumeration**: How many necklaces are there for a given alphabet?
  - **Generation**: How can we quickly generate all necklaces in order?
  - **Ranking**: How many necklaces are there smaller than a given necklace?
  - **Unranking**: How can we generate the necklace corresponding to a given rank?

# Multidimensional Necklaces

- A multidimensional cyclic string is the generalisation of a cyclic string into more than 1 dimension.
- Here we may rotate along one or more dimension.
- A multidimensional necklace is the lexicographically smallest rotation of a cyclic string.
  - The lexicographical ordering will be "book" style.



# Multidimensional Necklaces



# Multidimensional Necklaces

- Each necklace will be defined over some alphabet $\Sigma$ of length $k$.
- The largest dimension will be denoted $d$.
- The length of the necklace in dimension $i$ will be $N_i$.
  - This gives the total size as $N_1 \times N_2 \times \ldots \times N_d$.
  - We will use $m$ to denote the total number of positions, i.e. $m = N_1 \times N_2 \times \ldots \times N_d$.
- The set of necklaces of size $N_1 \times N_2 \times \ldots \times N_d$ over alphabet $k$ will be denoted $N_k^{N_1, N_2, \ldots, N_d}$.

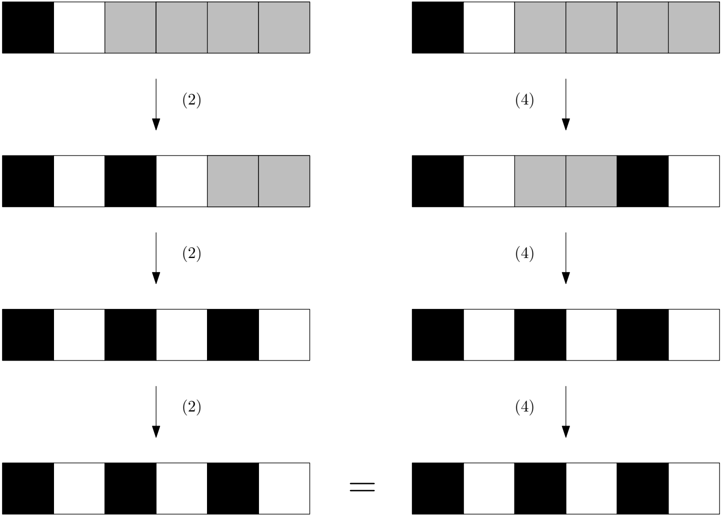
# Enumeration

- The symmetry of this space may be captured by the direct product of the cyclic groups $\mathbf{Z}_{N_i}$ for each $i$ from 1 to $d$.
  - $G = \times_{i=1}^{d} \mathbf{Z}_{N_i}$
- This can be used with the Pólya ennumeration theorem to giving:

$$\left| N_k^{N_1, N_2, \dots, N_d} \right| = \frac{1}{|G|} \sum_{g \in G} k^{c(g)}$$

- Here $c(g)$ returns the number of cycles of group operation $g$.
- Note also that $|G| = m$ by definition.

# Example



# $c(g)$

- Given an action $g \in G$ with a rotation by $g_i$ in dimension $i$, what is $c(g)$?
- Let $l_i$ be the length of the cycles in dimension $i$.
- In one dimension, we want the smallest $l_i$ such that $l_i \cdot g_i \bmod N_i \equiv 0$.
- This is given by $l_i = \frac{N_i}{GCD(N_i, g_i)}$.
- As all cycles will have the same length under translation, this gives (for one dimension):

$$c(i) = \frac{N_i}{l_i} = GCD(N_i, g_i)$$

# $c(g)$ in multiple dimensions

- Observe that the rotation in each dimension acts independently.
- For the cycle to be complete, we need the smallest $j$ such that, for each $i$ from 1 to $d$, $j \cdot g_i \bmod N_i \equiv 0$.
- Note that each $l_i$ must be a factor of $j$ so that $j \cdot g_i \bmod N_i \equiv 0$.
- Therefore the smallest $j$ will be the least common multiple of $l_1, l_2, \ldots, l_d$.
- Thus:

$$c(g) = \frac{m}{j} = \frac{m}{LCM(GCD(N_1, g_1), \ldots, GCD(N_d, g_d))}$$

# Enumeration - Can we do better

- If we can work out quickly how many times each value of $c(g)$ occurs, we remove a large amount of computation.
- In the $1d$ case, this is quite straight forward:
    - Given two rotations, by $i$ and $j$, $c(i) = c(j)$ iff $GCD(N_1, i) = GCD(N_1, j)$.
    - This means we only need to consider factors of $N_1$, as for any rotation by $i$, $GCD(N_1, i)$ must be a factor of $N_1$.
- Given $GCD(N_1, i) = l$, the number of groups where $c(g) = l$ is given by $\phi\left(\frac{N_1}{l}\right)$.
- Putting these observations together gives:

$$\left|N_k^{N_1}\right| = \frac{1}{N_1} \sum_{f|N_1} \phi\left(\frac{N_1}{f}\right) k^f$$

# Number of cycles in multiple dimensions

- As in one dimension, in multiple we only care about factors of each dimension.
  - Again the number of times each factor $f$ occurs will be $\phi(f)$.
- As each dimension is independent of each other, the number of times each combination occurs will be $\phi(f_1) \times \phi(f_2) \times \ldots \times \phi(f_d)$.
- Therefore the number of necklaces will be:

$$N_k^{N_1,N_2,\ldots,N_d} = \frac{1}{m} \sum_{f_1 | N_1} \phi\left(\frac{N_1}{f_1}\right) \sum_{f_2 | N_2} \phi\left(\frac{N_2}{f_2}\right) \cdots \sum_{f_d | N_d} \left(\frac{N_d}{f_d}\right) k^{c(g)}$$

# Generation

- The generation of necklaces is a well studied problem.
- Despite the exponential number of necklaces, this can be done relatively efficiently.
    - One notable result is a constant amortized time algorithm [1].
- We have extend this to an algorithm with an average time of $O(d)$ to generate each necklace.

# Ranking

- Ranking (also known as indexing) necklaces is the problem of determining how many necklaces there are smaller than some given string.
- The idea of ranking necklaces comes originates from the problem of ranking de Bruijn Sequences [2].
- In one dimension it is possible to rank necklaces is $O(m^2)$ [3].
- Our generalisation of this to higher dimensions require $O(d^3m^4)$ time.

| Necklace | Rank | Necklace | Rank | Necklace | Rank |
|---|---|---|---|---|---|
| aaa | 1 | abc | 5 | bbc | 9 |
| aab | 2 | acb | 6 | bcc | 10 |
| aac | 3 | acc | 7 | ccc | 11 |
| abb | 4 | bbb | 8 | | |

# Unranking

- Unranking is the complimentary process to ranking, taking a rank and finding the corresponding necklace.
- A binary search based algorithm can be used in combination with the ranking algorithm to solve this in $O(m^3 \log(k))$ time [3].
- The same algorithm can be adapted to the multidimensional case to do unranking in $O(d^3 m^6 \log(k))$

# Fixed Content necklaces

- One notable generalisation of necklaces is when the number of occurrences of each character is fixed.
- This known as a **fixed** content necklaces.
- There exists variations on our existing algorithms for the enumeration, generation, ranking and unranking of fixed content multi-dimensional necklaces, requiring at most a factor of $O(k)$ longer.

# Next Steps

- **Concerning Rotations**: Beyond just translational symmetry, we would want to consider rotational and reflective symmetry.
- This is a much harder problem, even in 1 dimension, it is not known how to rank when taking into account reflective symmetry.

# Thank you for listening

Feel free to email any questions to
duncan.adamson@liverpool.ac.uk

Kevin Cattell, Frank Ruskey, Joe Sawada, Micaela Serra, and C. Robert Miers.
Fast Algorithms to Generate Necklaces, Unlabeled Necklaces, and Irreducible Polynomials over GF(2).
*Journal of Algorithms*, 37(2):267–282, nov 2000.

Tomasz Kociumaka, Jakub Radoszewski, and Wojciech Rytter.
Computing k-th Lyndon Word and Decoding Lexicographically Minimal de Bruijn Sequence.
pages 202–211. Springer, Cham, 2014.
doi:10.1007/978-3-319-07566-2_21.

Joe Sawada and Aaron Williams.
Practical algorithms to rank necklaces, Lyndon words, and de Bruijn sequences.
*Journal of Discrete Algorithms*, 43:95–110, mar 2017.