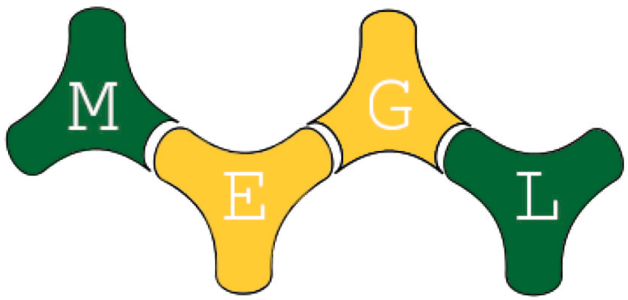
Universal Cycles in Higher Dimensions

William Carey, Matthew Kearney, Stefan Popescu, Charles Landreaux, Dr. Rachel Kirsch

George Mason University, MEGL

December 1, 2023



De Bruijn Cycles

A *De Bruijn* cycle for A^n is a cyclic sequence over an alphabet A that contains each word of length n exactly once. [2] [6]

For example, we will denote cycles of word length 2 over a binary alphabet: $\{0, 1\}^2$

Four possible words: 00, 01, 10, 11

De Bruijn Cycle: 0011

De Bruijn Cycles

A *De Bruijn* cycle for A^n is a cyclic sequence over an alphabet A that contains each word of length n exactly once. [2]

For example, we will denote cycles of word length 2 over a binary alphabet: $\{0, 1\}^2$

Four possible words: 00, 01, 10, 11

De Bruijn Cycle: 0011

De Bruijn Cycles

A *De Bruijn* cycle for A^n is a cyclic sequence over an alphabet A that contains each word of length n exactly once. [2]

For example, we will denote cycles of word length 2 over a binary alphabet: $\{0, 1\}^2$

Four possible words: 00, 01, 10, 11

De Bruijn Cycle: 0011

De Bruijn Cycles

A *De Bruijn* cycle for A^n is a cyclic sequence over an alphabet A that contains each word of length n exactly once. [2]

For example, we will denote cycles of word length 2 over a binary alphabet: $\{0, 1\}^2$

Four possible words: 00, 01, 10, 11

De Bruijn Cycle: 0011

De Bruijn Cycles

A *De Bruijn* cycle for A^n is a cyclic sequence over an alphabet A that contains each word of length n exactly once. [2]

For example, we will denote cycles of word length 2 over a binary alphabet: $\{0, 1\}^2$

Four possible words: 00, 01, 10, 11

De Bruijn Cycle: 0011

De Bruijn Cycles

A *De Bruijn* cycle for A^n is a cyclic sequence over an alphabet A that contains each word of length n exactly once. [2]

For example, we will denote cycles of word length 2 over a binary alphabet: $\{0, 1\}^2$

Four possible words: 00, 01, 10, 11

De Bruijn Cycle: 0011

De Bruijn cycles exist for any finite non-empty alphabet and every finite subword length.

De Bruijn Torus

Binary alphabet, word dimensions 2×2: $(4, 4, 2, 2)_2$

Sixteen possible words:

$$\begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix} \quad \begin{matrix} 0 & 0 \\ 0 & 1 \end{matrix} \quad \begin{matrix} 0 & 0 \\ 1 & 0 \end{matrix} \quad \begin{matrix} 1 & 0 \\ 0 & 0 \end{matrix} \quad \begin{matrix} 1 & 1 \\ 0 & 0 \end{matrix} \quad \begin{matrix} 0 & 1 \\ 0 & 1 \end{matrix} \quad \begin{matrix} 0 & 0 \\ 1 & 1 \end{matrix} \quad \begin{matrix} 1 & 0 \\ 1 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \quad \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \quad \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix} \quad \begin{matrix} 1 & 1 \\ 0 & 1 \end{matrix} \quad \begin{matrix} 0 & 1 \\ 1 & 1 \end{matrix} \quad \begin{matrix} 1 & 0 \\ 1 & 1 \end{matrix} \quad \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} \quad \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$$

De Bruijn Torus:

$$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{matrix}$$

De Bruijn Torus

Binary alphabet, word dimensions 2×2: $(4, 4, 2, 2)_2$

Sixteen possible words:

0 1	0 0	0 0	1 0	1 1	0 1	0 0	1 0
0 0	0 1	1 0	0 0	0 0	0 1	1 1	1 0

---

1 0	0 1	1 1	1 1	0 1	1 0	1 1	0 0
0 1	1 0	1 0	0 1	1 1	1 1	1 1	0 0

De Bruijn Torus:

$$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{matrix}$$

Higher Dimensional Analogues

Binary alphabet, word dimensions 2×2: $(4, 4, 2, 2)_2$

Sixteen possible words:

$$\begin{matrix}0&1\\0&0\end{matrix} \quad \begin{matrix}0&0\\0&1\end{matrix} \quad \begin{matrix}0&0\\1&0\end{matrix} \quad \begin{matrix}1&0\\0&0\end{matrix} \quad \begin{matrix}1&1\\0&0\end{matrix} \quad \begin{matrix}0&1\\0&1\end{matrix} \quad \begin{matrix}0&0\\1&1\end{matrix} \quad \begin{matrix}1&0\\1&0\end{matrix}$$

$$\begin{matrix}1&0\\0&1\end{matrix} \quad \begin{matrix}0&1\\1&0\end{matrix} \quad \begin{matrix}1&1\\1&0\end{matrix} \quad \begin{matrix}1&1\\0&1\end{matrix} \quad \begin{matrix}0&1\\1&1\end{matrix} \quad \begin{matrix}1&0\\1&1\end{matrix} \quad \begin{matrix}1&1\\1&1\end{matrix} \quad \begin{matrix}0&0\\0&0\end{matrix}$$

De Bruijn Torus:

$$\begin{matrix}0&1&0&0\\0&1&1&1\\1&1&1&0\\0&0&1&0\end{matrix}$$

Higher Dimensional Analogues

Binary alphabet, word dimensions 2×2: $(4, 4, 2, 2)_2$

Sixteen possible words:

$$\begin{matrix}0&1\\0&0\end{matrix} \quad \begin{matrix}0&0\\0&1\end{matrix} \quad \begin{matrix}0&0\\1&0\end{matrix} \quad \begin{matrix}1&0\\0&0\end{matrix} \quad \begin{matrix}1&1\\0&0\end{matrix} \quad \begin{matrix}0&1\\0&1\end{matrix} \quad \begin{matrix}0&0\\1&1\end{matrix} \quad \begin{matrix}1&0\\1&0\end{matrix}$$

$$\begin{matrix}1&0\\0&1\end{matrix} \quad \begin{matrix}0&1\\1&0\end{matrix} \quad \begin{matrix}1&1\\1&0\end{matrix} \quad \begin{matrix}1&1\\0&1\end{matrix} \quad \begin{matrix}0&1\\1&1\end{matrix} \quad \begin{matrix}1&0\\1&1\end{matrix} \quad \begin{matrix}1&1\\1&1\end{matrix} \quad \begin{matrix}0&0\\0&0\end{matrix}$$

De Bruijn Torus:

$$\begin{matrix}0&1&0&0\\0&1&1&1\\1&1&1&0\\0&0&1&0\end{matrix}$$

Higher Dimensional Analogues

Binary alphabet, word dimensions 2×2: $(4, 4, 2, 2)_2$

Sixteen possible words:

0 1	0 0	0 0	1 0	1 1	0 1	0 0	1 0
0 0	0 1	1 0	0 0	0 0	0 1	1 1	1 0

---

1 0	0 1	1 1	1 1	0 1	1 0	1 1	0 0
0 1	1 0	1 0	0 1	1 1	1 1	1 1	0 0

De Bruijn Torus:

$$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{matrix}$$

Higher Dimensional Analogues

Binary alphabet, word dimensions 2×2: $(4, 4, 2, 2)_2$

Sixteen possible words:

0 1	0 0	0 0	1 0	1 1	0 1	0 0	1 0
0 0	0 1	1 0	0 0	0 0	0 1	1 1	1 0

---

1 0	0 1	1 1	1 1	0 1	1 0	1 1	0 0
0 1	1 0	1 0	0 1	1 1	1 1	1 1	0 0

De Bruijn Torus:

$$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{matrix}$$

Visualisation of the toroidal properties of a DB torus

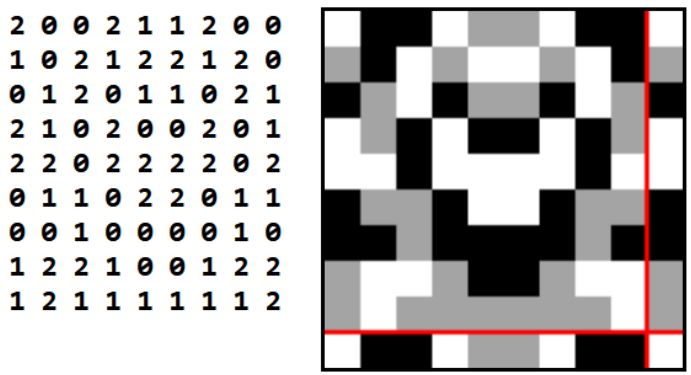
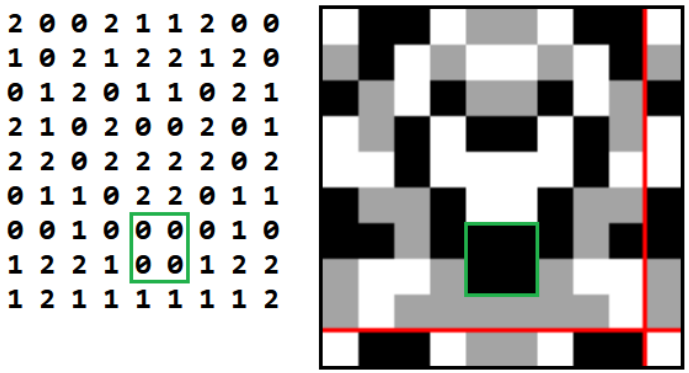


Figure: A visualization of a de Bruijn torus with shaded pixels. The de Bruijn torus is extended to the right and downwards so that sub-arrays can be identified more easily [1]

Visualisation of the toroidal properties of a DB torus

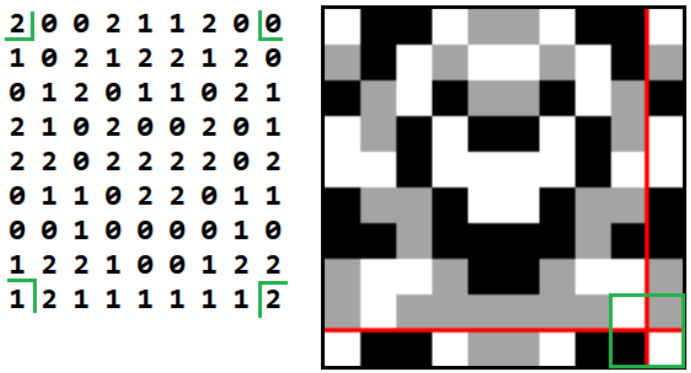


The matrix

$$\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$$

is easy to find as a submatrix.

Visualisation of the toroidal properties of a DB torus



The matrix

$$\begin{matrix} 2 & 1 \\ 0 & 2 \end{matrix}$$

is present in the torus when the torus is viewed cyclically.

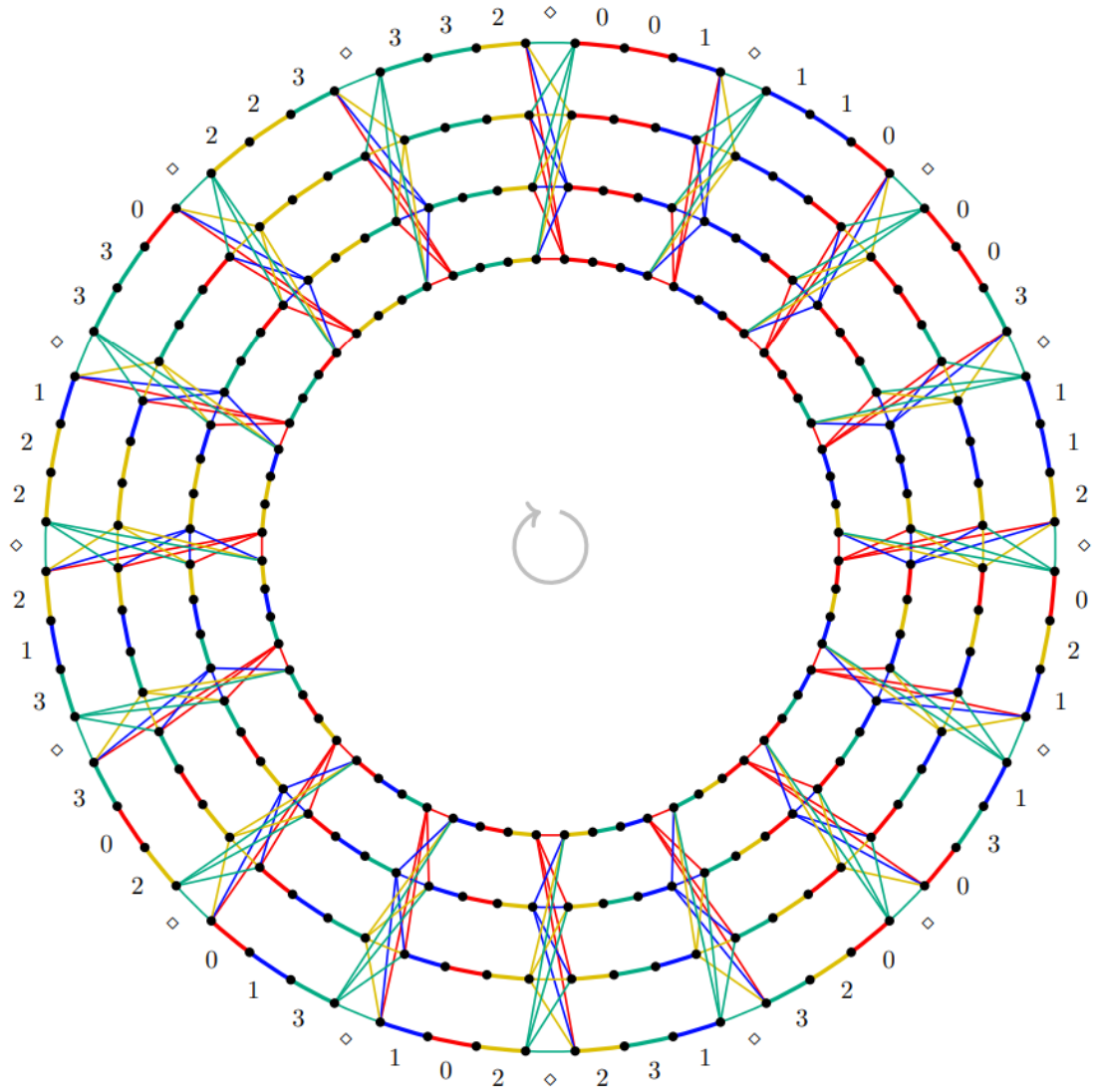
Universal Partial Cycles

Binary alphabet, word length 2: $\{0,1\}^2$, wildcard $\diamond$

$1\diamond 0$ covers 110 *and* 100

$001\diamond 110\diamond$ is a universal partial cycle [4] for $\{0,1\}^4$.

Universal Partial Cycles



A visualization of a much larger upcycle. [3]

Research Results

1. Universal partial tori exist.
2. We have a method for constructing universal partial tori from universal partial cycles.

Are there higher dimensional analogues of the De Bruijn Torus?

Hypertorus #39477527777731683095 with 64 unique matrices. We have not found any new *universal* 3-dimensional hypertori (yet).

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

0	1	1	0
1	1	0	0
1	0	0	1
0	0	1	1

1	1	0	0
0	0	1	1
1	1	0	0
0	0	0	1

1	0	0	1
1	1	0	0
0	0	1	0
0	1	1	1

A Minimal Universal Partial Torus

Consider $(4, 3, 2, 2)_2^{\diamond=1}$:

$\diamond$	0	0	1
1	1	0	0
1	1	0	0

$\diamond$	1	0	0
1	0	0	1
1	0	0	1

$\diamond$	0	1	1
0	1	1	0
0	1	1	0

$\diamond$	1	1	0
0	0	1	1
0	0	1	1

This covers the minimal non-trivial alphabet: $\{0, 1\}$ with the minimal number of wildcards, in the smallest number of cells.

We discovered this by a computer search of the solution space, which happily had only 2^{11} possible configurations.

Is there a way to construct a De Bruijn Torus without searching the whole solution space?

Yes!

There seem to be many different ways, one way we have focused on was a method developed by Matthew Kreitzer, Mihai Nica, Rajesh Pereira in source [5]. This method involves using something called *alternating De Bruijn cycles* and *De Bruijn families*.

Alternating words and Alternating De Bruijn cycles

Alternating words

Let A and B be two finite alphabets. An alternating word on the alphabet pair (A, B) is a string of length $2n + 1$ which alternates between elements of A and elements of B (beginning and ending with an element of A). The set of all $2n + 1$ alternating words is the set $A \times B \times A \times \ldots \times A \times B \times A$ with $n + 1$ elements from A and n elements from B. [5]

Alternating De Bruijn Cycles

Let A and B be two finite alphabets. An alternating de Bruijn sequence of order $2n+1$ on the alphabet pair (A, B), is a cyclic string which alternates between elements of A and elements of B and contains every alternating word on (A, B) of length $2n+1$ exactly once as a substring. We denote the set of all alternating de Bruijn sequences with these parameters as AdeBS(A,B,$2n + 1$). [5]

De Bruijn Families

A De Bruijn Family for a De Bruijn sequence is a set that contains cyclic strings where every subword of the original DB sequence appears exactly once in exactly one of the family members. Where each family member must have the same length (and same alphabet).

For example $\{0001, 1110\}$ is the family for the binary DB sequence of subword length 3.

0001 contains 000, 001, 010, 100 1110 contains 111, 110, 101, 011

We also have adopted the definition of DB families which are for DB sequences to DB up-families which are for up-DB sequences.

Partial Families

A "De Bruijn Partial Family" is a structure similar to a De Bruijn Family that may also include wildcards.

For example,

$\{110\diamond003\diamond, 332\diamond001\diamond, 330\diamond223\diamond, 312\diamond221\diamond,$

$310\diamond203\diamond, 132\diamond201\diamond, 130\diamond023\diamond, 112\diamond021\diamond\}$

is a "partial family" for subwords of length 4 within the alphabet $\{0, 1, 2, 3\}$.

Rotation Functions

Because the objects in question here are cyclic, they can be rotated. Let π_x be a forward (i.e. rightward) rotation by x elements with x considered mod the length of the cyclic string. For example,

$\pi_1(0100) = 0010$

$\pi_{-1}(0100) = 1000$

$\pi_5(0100) = \pi_1(0100) = 0010$

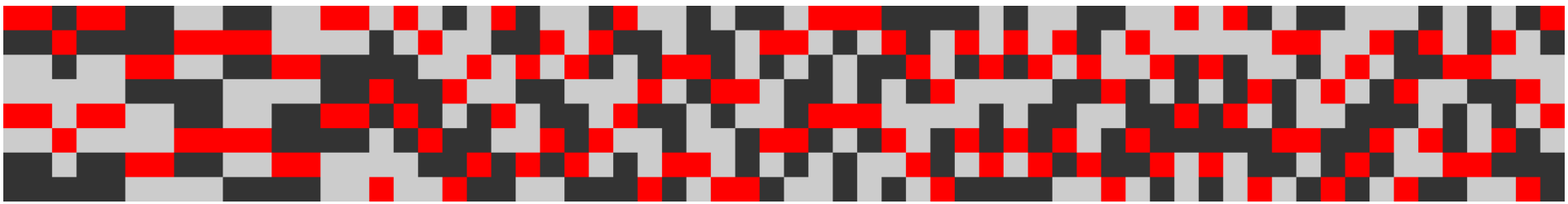
A method for constructing De Bruijn tori

Kreitzer et al. [5] constructed de Bruijn tori using alternating de Bruijn sequences and de Bruijn families. The alphabet pair that the alternating DB sequence uses is $(\mathcal{S}, \Pi)$, where $\mathcal{S}$ is the de Bruijn family and Π is the set of rotations on each element of $\mathcal{S}$. The construction involves cyclically shifting elements of $\mathcal{S}$ according to the alternating de Bruijn sequence and then stacking them.

Can we use this method to create an uptorus?

We found that the method works when $\mathcal{S}$ is replaced with a universal partial family. The result includes wildcards and should be an uptorus. An upcycle can be viewed as a universal partial family with a single element.

The upcycle 001$\diamond$110$\diamond$ (mentioned in [3, 4]) was used. A de Bruijn cycle generated at [6] was converted to an alternating de Bruijn sequence and used in the construction of the uptorus illustrated below.



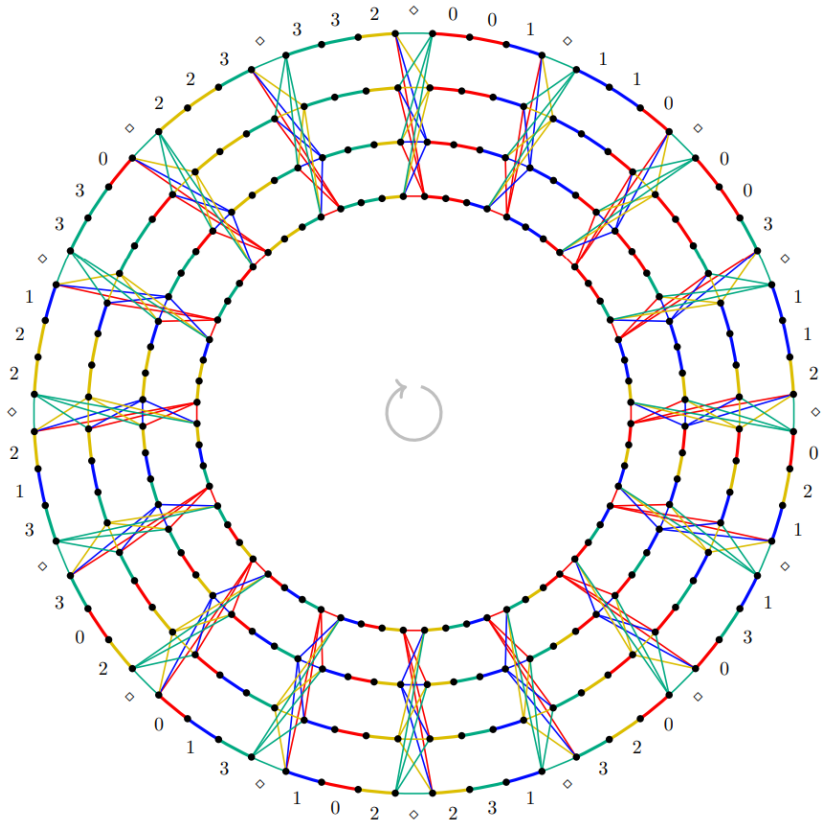
Can this method be used in general?

We don't know yet!

We do believe that the method above used above can be applied analogously to how it was used in [5], which would then mean there are infinite uptori!

This is what we want to prove next semester.

Partial Families and The Pizza Slicing Conjecture



Conjecture: Cutting up the universal partial cycle from this example into eight equal portions eight characters long leads to a universal partial family, and we conjecture that universal partial families can be created like this for all similar universal partial cycles with larger alphabets. Picture is from source [3]

Quasi-Families

A "De Bruijn Quasi-Family" is a structure similar to a De Bruijn Family, but waiving the requirement that all the elements must have the same length.

For example $\{001111, 00001, 01011\}$ is a "quasi-family" for the binary DB sequence of subword length 4.

The elements can even be shorter than the subword length - for example, $\{001, 000011, 0101111\}$ could also be considered a "quasi-family" for subwords of length 4.

De Bruijn Partial Quasi-Families

Analogues of the "Pizza Slicing" from the previous example have been used to create "quasi-families" that have wildcards in them.

For example,

$\{110\diamond003\diamond332\diamond001\diamond, 330\diamond223\diamond, 312\diamond221\diamond,$

$310\diamond203\diamond, 132\diamond201\diamond, 130\diamond023\diamond, 112\diamond021\diamond\}$

is just one example of a family that satisfies the constraints, but two of the "pizza slices" have been fused together.

This pattern does not always work to create universal partial quasi-families. It is unknown which cuttings of the cycle will always lead to a family.

Acknowledgements

We would like to give a thanks to everyone at MEGL for allowing us to do this research.

We would also like to thank Prof. Kirsch, who led our group and worked with us, and Matthew Kearney, our graduate mentor.

References I

Minimal arrays containing all sub-array combinations of symbols: De bruijn sequences and tori.

URL: `https://web.archive.org/web/20140527202958/http://lcni.uoregon.edu/~dow/Geek_art/Minimal_combinatorics/Minimal_arrays_containing_all_combinations.html`.

N.G. Bruijn, de.

A combinatorial problem.

Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam, 49(7):758–764, 1946.

Dylan Fillmore, Bennet Goeckner, Rachel Kirsch, Kirin Martin, and Daniel McGinnis.

The existence and structure of universal partial cycles, 2023.

arXiv:2310.13067.

References II

Bennet Goeckner, Corbin Groothuis, Cyrus Hettle, Brian Kell, Pamela Kirkpatrick, Rachel Kirsch, and Ryan Solava.

Universal partial words over non-binary alphabets.

Theoretical Computer Science, 713:56–65, 2018.

doi:10.1016/j.tcs.2017.12.022.

Matthew Kreitzer, Mihai Nica, and Rajesh Pereira.

Using alternating de Bruijn sequences to construct de Bruijn tori, 2023.

arXiv:2306.01498.

Joe Sawada.

De Bruijn sequence and universal cycle constructions, 2023.

URL: http://debruijnsequence.org.