Unclustered BWTs of any Length over Non-Binary Alphabets

WCTA 2025

Gabriele Fici[1], **Estéban Gabory**[1],Giuseppe Romana[1],Marinella Sciortino[1]

[1]Università di Palermo, Italy

September 11th, 2025

Outline

BWT: definition

$$u = \mathtt{0100111}$$

BWT: definition

$$u = \mathtt{0100111}$$

$$\begin{pmatrix} \mathtt{0100111} \\ \mathtt{1010011} \\ \mathtt{1101001} \\ \mathtt{1110100} \\ \mathtt{0111010} \\ \mathtt{0011101} \\ \mathtt{1001110} \end{pmatrix}$$

BWT: definition

$$u = 0100111$$

$$\begin{pmatrix} 0011101 \\ 0100111 \\ 0111010 \\ 1001110 \\ 1010011 \\ 1101001 \\ 1110100 \end{pmatrix}$$

BWT: definition

$$u = \mathtt{0100111}$$

$$\begin{pmatrix} \mathtt{0011101} \\ \mathtt{0100111} \\ \mathtt{0111010} \\ \mathtt{1001110} \\ \mathtt{1010011} \\ \mathtt{1101001} \\ \mathtt{1110100} \end{pmatrix} BWT\ matrix$$

BWT: definition

$$u = \mathtt{0100111}$$

$$\begin{pmatrix} \mathtt{001110}\color{red}{\mathtt{1}} \\ \mathtt{010011}\color{red}{\mathtt{1}} \\ \mathtt{011101}\color{red}{\mathtt{0}} \\ \mathtt{100111}\color{red}{\mathtt{0}} \\ \mathtt{101001}\color{red}{\mathtt{1}} \\ \mathtt{110100}\color{red}{\mathtt{1}} \\ \mathtt{111010}\color{red}{\mathtt{0}} \end{pmatrix} \quad BWT\ matrix$$

BWT: definition

$$u = 0100111$$

$$\begin{pmatrix} 001110\color{red}{1} \\ 010011\color{red}{1} \\ 011101\color{red}{0} \\ 100111\color{red}{0} \\ 101001\color{red}{1} \\ 110100\color{red}{1} \\ 111010\color{red}{0} \end{pmatrix} \quad BWT\ matrix$$

$$BWT(u) = 1100110$$

BWT and clusterization

The BWT is widely used in data compression because of its *clustering effect*

BWT and clusterization

The BWT is widely used in data compression because of its *clustering effect*

Example

$$u = 1010210201$$

BWT and clusterization

The BWT is widely used in data compression because of its *clustering effect*

Example

$$u = 1010210201$$
$$\mathrm{BWT}(u) = 1211120000$$

BWT and clusterization

The BWT is widely used in data compression because of its *clustering effect*

Example

$$\begin{aligned} u &= 1010210201 \\ \mathrm{BWT}(u) &= 1211120000 \\ RLE(BWT(u)) &= 1^1 2^1 1^3 2^1 0^4 \end{aligned}$$

BWT and clusterization

The BWT is widely used in data compression because of its *clustering effect*

Example

$$\begin{aligned} u &= 1010210201 \\ \mathrm{BWT}(u) &= 1211120000 \\ RLE(BWT(u)) &= 1^1 2^1 1^3 2^1 0^4 \end{aligned}$$

However, there exist texts for which no such clustering occurs:

Example

$$u' = 00211$$

BWT and clusterization

The BWT is widely used in data compression because of its *clustering effect*

Example

$$u = 1010210201$$
$$\mathrm{BWT}(u) = 1211120000$$
$$RLE(BWT(u)) = 1^1 2^1 1^3 2^1 0^4$$

However, there exist texts for which no such clustering occurs:

Example

$$u' = 00211$$
$$\mathrm{BWT}(u') = 10120$$

BWT and clusterization

The BWT is widely used in data compression because of its *clustering effect*

Example

$$
\begin{aligned}
u &= 1010210201 \\
\mathrm{BWT}(u) &= 1211120000 \\
RLE(BWT(u)) &= 1^1 2^1 1^3 2^1 0^4
\end{aligned}
$$

However, there exist texts for which no such clustering occurs:

Example

$$
\begin{aligned}
u' &= 00211 \\
\mathrm{BWT}(u') &= 10120 \\
RLE(BWT(u')) &= 1^1 0^1 1^1 2^1 0^1
\end{aligned}
$$

Totally unclustered BWTs

We study the following problem:

Problem

Let $k \leq n$ be a pair of integers. Is there a word w of length n over k letters such that $\mathrm{BWT}(w)$ has no run longer than 1?

Totally unclustered BWTs

We study the following problem:

Problem

Let $k \leq n$ be a pair of integers. Is there a word w of length n over k letters such that $\mathrm{BWT}(w)$ has no run longer than 1?

We say that such a word has a *totally unclustered BWT*.

Totally unclustered BWTs

We study the following problem:

Problem

Let $k \leq n$ be a pair of integers. Is there a word w of length n over k letters such that $\mathrm{BWT}(w)$ has no run longer than 1?

We say that such a word has a *totally unclustered BWT*.
(Note: We can already ignore words that are not primitive, because $\mathrm{BWT}(u^r)$ always have runs of length $\geq r$.)

Totally unclustered BWTs

We study the following problem:

Problem

Let $k \leq n$ be a pair of integers. Is there a word w of length n over k letters such that $\mathrm{BWT}(w)$ has no run longer than 1?

We say that such a word has a *totally unclustered BWT*.
(Note: We can already ignore words that are not primitive, because $\mathrm{BWT}(u^r)$ always have runs of length $\geq r$.)
Motivation: This is the worst case of BWT clustering. Great attention has been given to the best case, in which the BWT produces the fewest number of clusters. (Simpson and Puglisi, 2008; Restivo and Rosone, 2009; Mantaci et al., 2003; Ferenczi and Zamboni, 2013).

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ & & & & & & & & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & & & & & & & & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & & & & & & & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & & & & & & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & & & & & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & & & & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & & & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1\textcolor{red}{2}11120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & \end{pmatrix}$$

Inverse standard permutation: definition

Definition

The *inverse standard permutation* (or *FL mapping*) of a word s is the permutation π_s^{-1} defined by $\pi_s^{-1}(i) = p$ if p is the position of the letter with lexicographical rank p in s.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT of some primitive word u if and only if π_u^{-1} is a cyclic permutation.

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT of some primitive word u if and only if π_u^{-1} is a cyclic permutation.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if* π_u^{-1} *is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if* π_u^{-1} *is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if* π_u^{-1} *is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if π_u^{-1} is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if* π_u^{-1} *is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9\ 5$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if* π_u^{-1} *is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9\ 5\ 2$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if π_u^{-1} is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9\ 5\ 2\ 8$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT of some primitive word u if and only if π_u^{-1} is a cyclic permutation.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9\ 5\ 2\ 8\ 1$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if* π_u^{-1} *is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9\ 5\ 2\ 8\ 1\ 7$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if* π_u^{-1} *is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & \color{red}{7} & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & \color{red}{4} & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9\ 5\ 2\ 8\ 1\ 7\ 4$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT of some primitive word u if and only if π_u^{-1} is a cyclic permutation.

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9\ 5\ 2\ 8\ 1\ 7\ 4)$$

Inverse standard permutation and BWT

Lemma (Folklore)

A n-length word w is a BWT *of some primitive word u if and only if* π_u^{-1} *is a cyclic permutation.*

Example

Let $w = 1211120000$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 6\ 3\ 9\ 5\ 2\ 8\ 1\ 7\ 4)$$

Hence, w is the BWT of a word ($w = \mathrm{BWT}(1010210201)$).

Inverse standard permutation and BWT

Example

Let $w = 210102$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 3 & 0 & 5 \end{pmatrix}$$

Inverse standard permutation and BWT

Example

Let $w = 210102$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 3 & 0 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 2\ 1\ 4)$$

Inverse standard permutation and BWT

Example

Let $w = 210102$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 3 & 0 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 2\ 1\ 4)(3)$$

Inverse standard permutation and BWT

Example

Let $w = 210102$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 3 & 0 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 2\ 1\ 4)(3)(5)$$

Inverse standard permutation and BWT

Example

Let $w = 210102$. Then, we have:

$$\pi_w^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 3 & 0 & 5 \end{pmatrix}$$

$$\pi_w^{-1} = (0\ 2\ 1\ 4)(3)(5)$$

Hence, w is *not* the BWT of any word u.

Problem reformulation

We can reformulate our problem as follows:

Problem

Let $k \leq n$ be a pair of integers. Is there a word u of length n over k letters such that (i) π_u^{-1} is a cycle, and (ii) u has no run longer than 1?

Outline

The binary case: Mantaci et al., 2017

Theorem (Mantaci et al., 2017)

There exists a $2n$-length word with totally unclustered BWT if and only if $2n+1$ is an odd prime and 2 generates the cyclic group $\mathbb{Z}^*_{2n+1}$.

The binary case: Mantaci et al., 2017

Theorem (Mantaci et al., 2017)

There exists a $2n$-length word with totally unclustered BWT if and only if $2n+1$ is an odd prime and 2 generates the cyclic group $\mathbb{Z}^*_{2n+1}$.

Sketch.

A binary word starting with 0 or ending with 1 is not a BWT image.

The binary case: Mantaci et al., 2017

Theorem (Mantaci et al., 2017)

There exists a $2n$-length word with totally unclustered BWT if and only if $2n+1$ is an odd prime and 2 generates the cyclic group $\mathbb{Z}^*_{2n+1}$.

Sketch.

A binary word starting with 0 or ending with 1 is not a BWT image.
Hence, the only candidate is $s_{2n} = (10)^n$

The binary case: Mantaci et al., 2017

Theorem (Mantaci et al., 2017)

There exists a 2n-length word with totally unclustered BWT *if and only if* $2n+1$ *is an odd prime and* 2 *generates the cyclic group* $\mathbb{Z}^*_{2n+1}$.

Sketch.

A binary word starting with $\mathtt{0}$ or ending with $\mathtt{1}$ is not a BWT image. Hence, the only candidate is $s_{2n} = (\mathtt{10})^n$ One can rewrite $\pi^{-1}_{s_{2n}}(i) = 2^{i-1} \mod 2n+1$. □

The binary case: Mantaci et al., 2017

Theorem (Mantaci et al., 2017)

There exists a $2n$-length word with totally unclustered BWT if and only if $2n+1$ is an odd prime and 2 generates the cyclic group $\mathbb{Z}^*_{2n+1}$.

Sketch.

A binary word starting with 0 or ending with 1 is not a BWT image. Hence, the only candidate is $s_{2n} = (10)^n$ One can rewrite $\pi^{-1}_{s_{2n}}(i) = 2^{i-1} \mod 2n+1$ (up to reindexing). $\square$

The binary case: Mantaci et al., 2017

Theorem (Mantaci et al., 2017)

There exists a $2n$*-length word with totally unclustered* BWT *if and only if* $2n+1$ *is an odd prime and* 2 *generates the cyclic group* $\mathbb{Z}^*_{2n+1}$.

Sketch.

A binary word starting with 0 or ending with 1 is not a BWT image.
Hence, the only candidate is $s_{2n} = (10)^n$ One can rewrite $\pi^{-1}_{s_{2n}}(i) = 2^{i-1} \mod 2n+1$ (up to reindexing). $\square$

Our problem: what happens after the binary case?

Outline

DeBruijn word, definition

Definition

A word w is a *deBruijn word* of order n over Σ_k if it contains (cyclically) each word of Σ_k^n exactly once.

DeBruijn word, definition

Definition

A word w is a *deBruijn word* of order n over Σ_k if it contains (cyclically) each word of Σ_k^n exactly once.

Lemma

For every $k \leq n$, there exists a deBruijn word of order n over Σ_k and it has length k^n.

DeBruijn word, definition

Definition

A word w is a *deBruijn word* of order n over Σ_k if it contains (cyclically) each word of Σ_k^n exactly once.

Theorem (Higgins, 2012)

A word u of length k^n is a deBruijn word if and only if $w = \mathrm{BWT}(u)$ can be written as a concatenation of permutations of the alphabet.

DeBruijn word, definition

Definition

A word w is a *deBruijn word* of order n over Σ_k if it contains (cyclically) each word of Σ_k^n exactly once.

Theorem (Higgins, 2012)

A word u of length k^n is a deBruijn word if and only if $w = \mathrm{BWT}(u)$ can be written as a concatenation of permutations of the alphabet.

Example

$$u = 001022112$$

DeBruijn word, definition

Definition

A word w is a *deBruijn word* of order n over Σ_k if it contains (cyclically) each word of Σ_k^n exactly once.

Theorem (Higgins, 2012)

A word u of length k^n is a deBruijn word if and only if $w = \mathrm{BWT}(u)$ can be written as a concatenation of permutations of the alphabet.

Example

$$u = 001022112$$
$$\mathrm{BWT}(u) = 201021120$$

DeBruijn word, definition

Definition

A word w is a *deBruijn word* of order n over Σ_k if it contains (cyclically) each word of Σ_k^n exactly once.

Theorem (Higgins, 2012)

A word u of length k^n is a deBruijn word if and only if $w = \mathrm{BWT}(u)$ can be written as a concatenation of permutations of the alphabet.

Example

$$u = 001022112$$
$$\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$$

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?
2. de Bruijn words do not *exactly* have totally unclustered BWT.

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?
2. de Bruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?
2. de Bruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$

DeBruijn words and graphs, revisited

Alphabet size $k = 2$

DeBruijn words and graphs, revisited

Alphabet size	$k = 2$
Length	$n = 3$

DeBruijn words and graphs, revisited

$$\begin{array}{ll} \text{Alphabet size} & k = 2 \\ \text{Length} & n = 3 \end{array}$$

$$V = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

DeBruijn words and graphs, revisited

$$\begin{array}{ll} \text{Alphabet size} & k = 2 \\ \text{Length} & n = 3 \end{array}$$

$$V = \{\mathtt{000}, \mathtt{001}, \mathtt{010}, \mathtt{011}, \mathtt{100}, \mathtt{101}, \mathtt{110}, \mathtt{111}\}$$

$$E = \{\text{pairs of strings with length } n - 1 = 2 \text{ overlap }\}$$

DeBruijn words and graphs, revisited

$$\begin{array}{ll} \text{Alphabet size} & k = 2 \\ \text{Length} & n = 3 \end{array}$$

$$V = \{\texttt{000}, \texttt{001}, \texttt{010}, \texttt{011}, \texttt{100}, \texttt{101}, \texttt{110}, \texttt{111}\}$$
$$E = \{\text{pairs of strings with length } n - 1 = 2 \text{ overlap } \}$$

$\texttt{010}$

DeBruijn words and graphs, revisited

$$\begin{array}{ll} \text{Alphabet size} & k = 2 \\ \text{Length} & n = 3 \end{array}$$

$$V = \{\texttt{000}, \texttt{001}, \texttt{010}, \texttt{011}, \texttt{100}, \texttt{101}, \texttt{110}, \texttt{111}\}$$

$$E = \{\text{pairs of strings with length } n - 1 = 2 \text{ overlap }\}$$

```
 100
010
 101
```

DeBruijn words and graphs, revisited

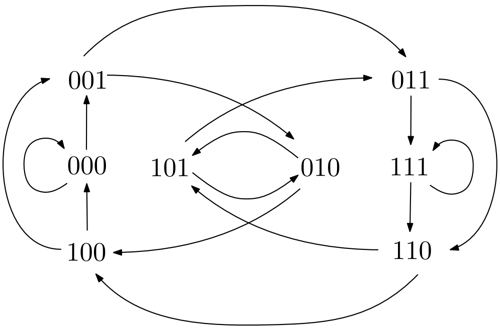
Alphabet size	$k = 2$
Length	$n = 3$

$$V = \{\texttt{000}, \texttt{001}, \texttt{010}, \texttt{011}, \texttt{100}, \texttt{101}, \texttt{110}, \texttt{111}\}$$

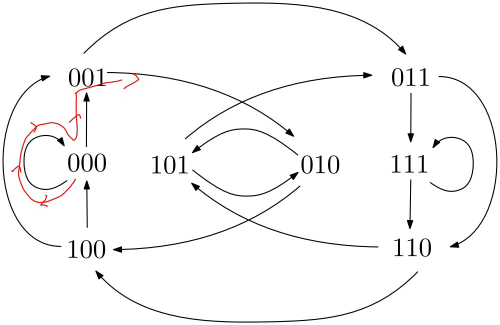
$$E = \{\text{pairs of strings with length } n - 1 = 2 \text{ overlap }\}$$

$$\texttt{010} \to \texttt{100}$$
$$\texttt{010} \to \texttt{101}$$

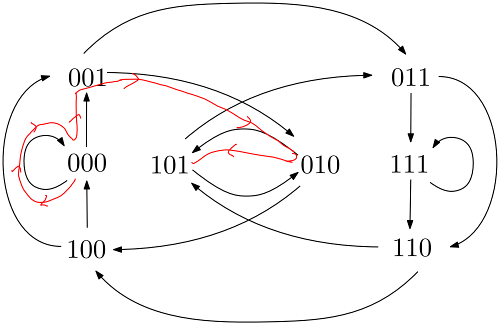
DeBruijn words and graphs, revisited



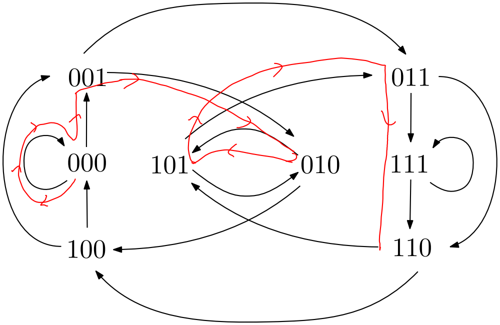
DeBruijn words and graphs, revisited



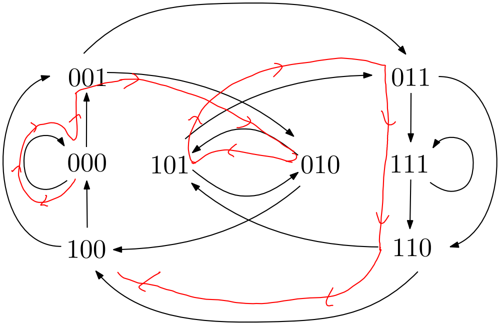
DeBruijn words and graphs, revisited



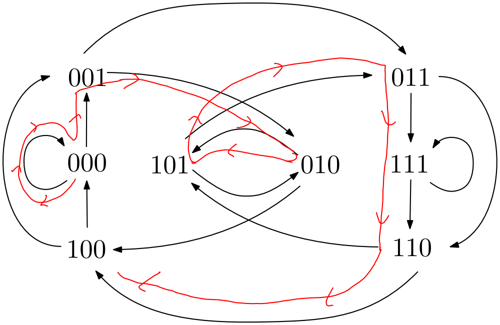
DeBruijn words and graphs, revisited



DeBruijn words and graphs, revisited



DeBruijn words and graphs, revisited



De Bruijn Word $w = \texttt{0001011100}$

DeBruijn words and graphs, revisited

Alphabet size $k = 2$

Length $n = 3$

$$V = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

$$E = \{\text{pairs of strings with length } n - 1 = 2 \text{ overlap } \}$$

DeBruijn words and graphs, revisited

$$\begin{aligned} \text{Alphabet size} \quad & k = 2 \\ \text{Length} \quad & n = 3 \end{aligned}$$

$$V = \{\texttt{0,1,2,3,4,5,6,7}\}$$

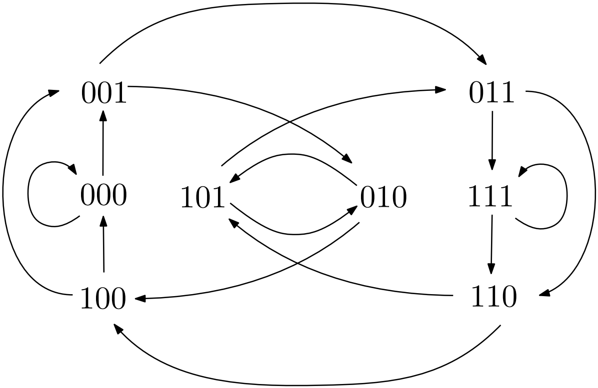
$$E = \{\text{pairs of strings with length } n - 1 = 2 \text{ overlap }\}$$

DeBruijn words and graphs, revisited

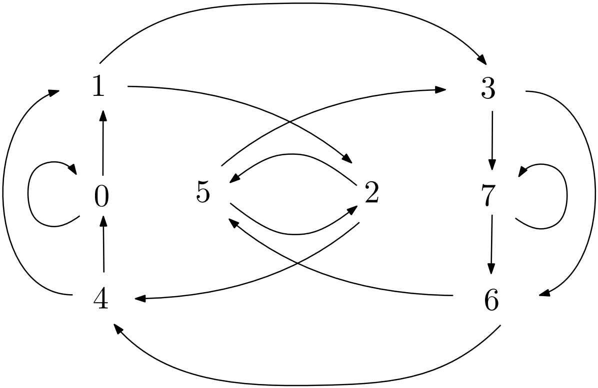
$$\begin{array}{ll} \text{Alphabet size} & k = 2 \\ \text{Length} & n = 3 \end{array}$$

$$V = \{\mathtt{0},\mathtt{1},\mathtt{2},\mathtt{3},\mathtt{4},\mathtt{5},\mathtt{6},\mathtt{7}\}$$
$$E = \{(i, j),\ j \in \{2i, 2i + 1\} \bmod 8\}$$

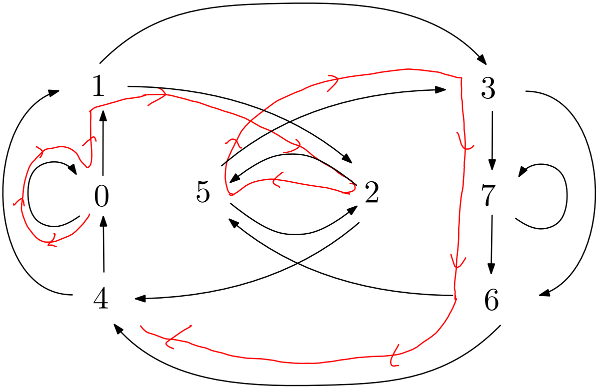
DeBruijn words and graphs, revisited



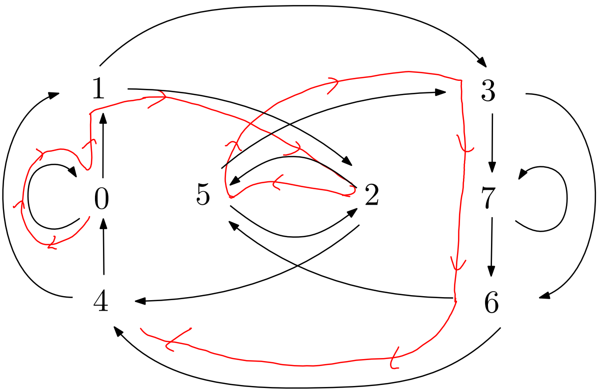
DeBruijn words and graphs, revisited



DeBruijn words and graphs, revisited

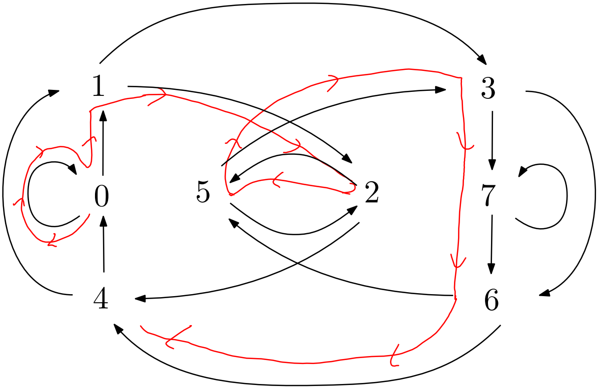


DeBruijn words and graphs, revisited



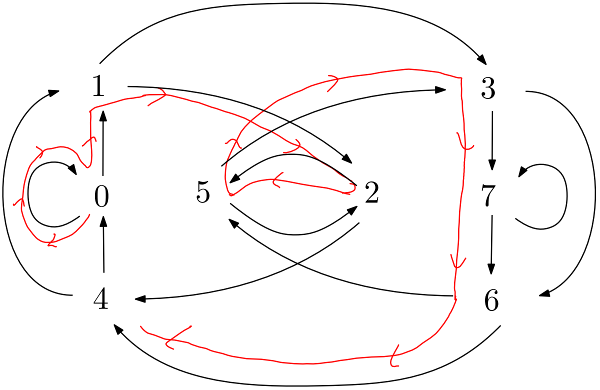
$(0\ 1\ 2\ 5\ 3\ 7\ 6\ 4)$

DeBruijn words and graphs, revisited



$$w = 0001011100 \quad ? \quad (0\ 1\ 2\ 5\ 3\ 7\ 6\ 4)$$

DeBruijn words and graphs, revisited



$$\pi_{BWT(w)}^{-1} = (0\ 1\ 2\ 5\ 3\ 7\ 6\ 4)$$

DeBruijn words and graphs, revisited

$$\pi_{BWT(w)}^{-1} = \texttt{01253764}$$

Because the letters of the de Bruijn word w are `00001111`, and in the graph, the strings starting with 0 are sent to $[0, 3]$, and the strings starting with 0 are sent to $[4, 7]$

DeBruijn words and graphs, revisited

Lemma

A word w is a de Bruijn word of order n over k letters if and only if $\pi^{-1}_{\mathrm{BWT}(w)}$ (written as a cycle) is an Hamiltonian cycle of the de Bruijn graph $\mathrm{DB}(k, kn)$.

Generalized de Bruijn words

Definition (Generalized de Bruijn graphs. Imase and Itoh, 1981; Reddy et al., 1980)

Let k, n be two integers. The *Generalized de Bruijn graph* $\mathrm{DB}(k,n)$ is defined by:

- $V(\mathrm{DB}(k,n)) = \{0, \ldots n-1\}$
- $E(\mathrm{DB}(k,n)) = \{(i, ki + j \mod n),\ i \in V\ j \in \{0, \ldots k-1\}\}$

Generalized de Bruijn words

Definition (Generalized de Bruijn graphs. Imase and Itoh, 1981; Reddy et al., 1980)

Let k, n be two integers. The *Generalized de Bruijn graph* $\mathrm{DB}(k,n)$ is defined by:

- $V(\mathrm{DB}(k,n)) = \{0, \ldots n-1\}$
- $E(\mathrm{DB}(k,n)) = \{(i, ki + j \mod n),\ i \in V\ j \in \{0, \ldots k-1\}\}$

Theorem (Fici and Gabory, 2025)

Let $w \in \Sigma_k^{kn}$. The following assertions are equivalent:

- *The permutation $\pi^{-1}_{\mathrm{BWT}(w)}$, written as a cycle, is an Hamiltonian cycle of the de Bruijn graph $\mathrm{DB}(k,kn)$*
- *The word $\mathrm{BWT}(w)$ is a concatenation of permutations of Σ_k*

In that case, we say that w is a generalized de Bruijn word.

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths? With generalized de Bruijn words, we can get words of length kn instead of k^n.

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths? With generalized de Bruijn words, we can get words of length kn instead of k^n.
2. DeBruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$.

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths? With generalized de Bruijn words, we can get words of length kn instead of k^n.
2. DeBruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120.$

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths? With generalized de Bruijn words, we can get words of length kn instead of k^n.
2. DeBruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$. We call such configurations *ties*.

Outline

Untying generalized deBruijn words

$u = [001022112]$

Untying generalized deBruijn words

$$u = [001022112]$$

w=BWT(u)=201021120

Untying generalized deBruijn words

$$u = [001022112]$$

$$\text{w=BWT(u)=201021120}$$

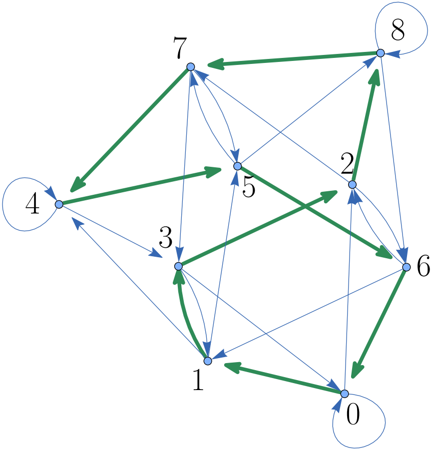
$$\sigma = \pi_w^{-1} = (0, 1, 3, 2, 8, 7, 4, 5, 6)$$

Untying generalized deBruijn words

$u = [001022112]$

w=BWT(u)=201021120

$\sigma = \pi_w^{-1} = (0, 1, 3, 2, 8, 7, 4, 5, 6)$

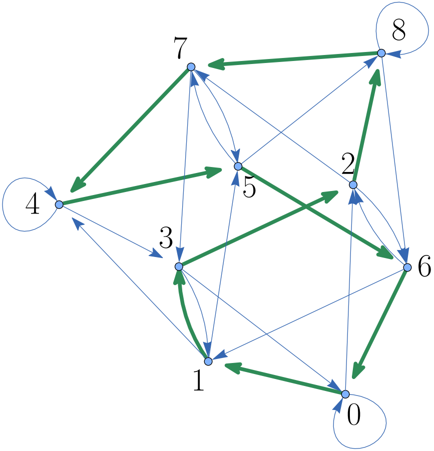


Untying generalized deBruijn words

$$u = [001022112]$$

w=BWT(u)=201021120

$$\sigma = \pi_w^{-1} = (0, 1, 3, 2, 8, 7, 4, 5, 6)$$



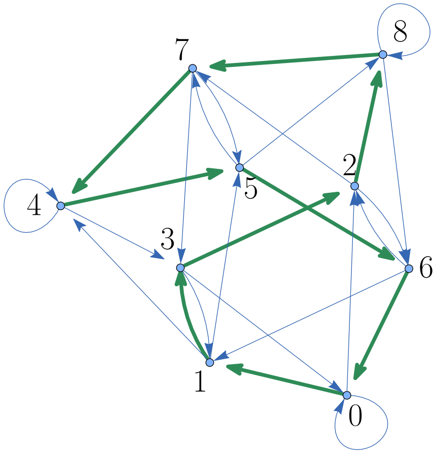
Untying generalized deBruijn words

$$u = [\texttt{001022112}]$$

$$\text{w=BWT(u)=}\texttt{20102112 0}$$

$$\sigma = \pi_w^{-1} = (0, 1, 3, 2, 8, 7, 4, 5, 6)$$

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 8 & 2 & 5 & 6 & 0 & 4 & 7 \end{pmatrix}$$



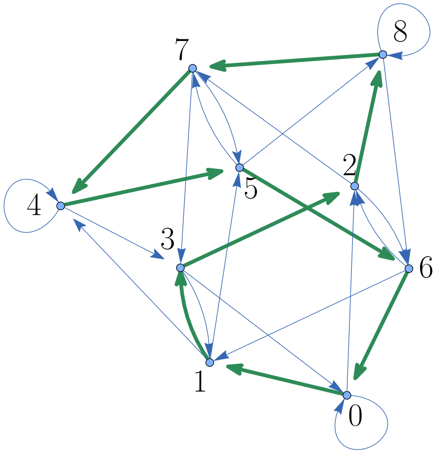
Untying generalized deBruijn words

$$u = [001022112]$$

$$\text{w=BWT(u)=}\mathtt{201021120}$$

$$\sigma = \pi_w^{-1} = (0,1,3,2,8,7,4,5,6)$$

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 8 & 2 & 5 & 6 & 0 & 4 & 7 \end{pmatrix}$$



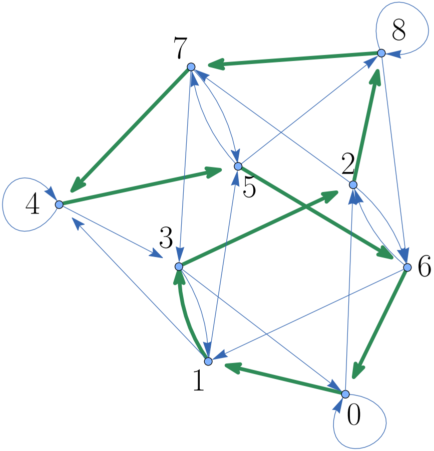
Untying generalized deBruijn words

$$u = [001022112]$$

w=BWT(u)=201021120

$$\sigma = \pi_w^{-1} = (0,1,3,2,8,7,\underline{4,5},6)$$

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 8 & 2 & 5 & 6 & 0 & 4 & 7 \end{pmatrix}$$



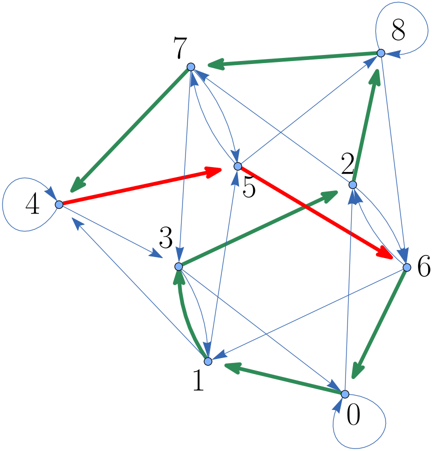
Untying generalized deBruijn words

$$u = [001022112]$$

$$\text{w=BWT(u)=}201021120$$

$$\sigma = \pi_w^{-1} = (0,1,3,2,8,7,\underline{4,5},\underline{6})$$

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 8 & 2 & 5 & 6 & 0 & 4 & 7 \end{pmatrix}$$



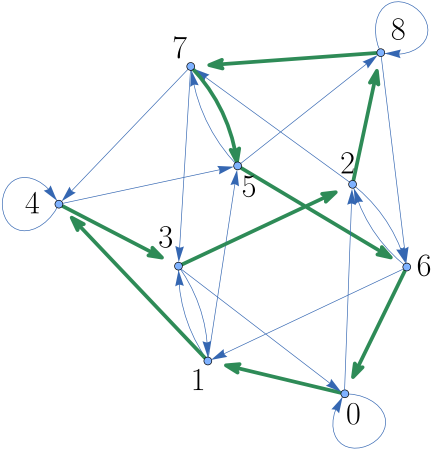
Untying generalized deBruijn words

$$u = [001102212]$$

w=BWT(u)=201102120

$$\sigma = \pi_w^{-1} = (0, 1, 4, 3, 2, 8, 7, 5, 6)$$

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 4 & 8 & 2 & 3 & 6 & 0 & 5 & 7 \end{pmatrix}$$



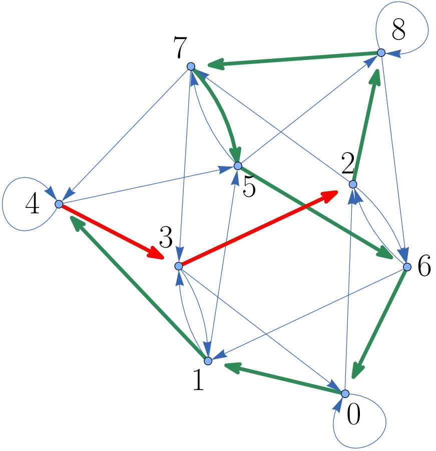
Untying generalized deBruijn words

$$u = [\texttt{001102212}]$$

$$\text{w=BWT(u)=}\texttt{201102120}$$

$$\sigma = \pi_w^{-1} = (0, 1, 4, 3, 2, 8, 7, 5, 6)$$

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 4 & 8 & 2 & 3 & 6 & 0 & 5 & 7 \end{pmatrix}$$



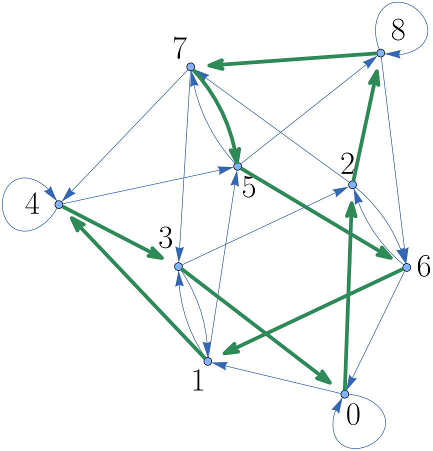
Untying generalized deBruijn words

$$u = [\texttt{002212011}]$$

$$\text{w=BWT(u)=}\texttt{120102120}$$

$$\sigma = \pi_w^{-1} = (0, 2, 8, 7, 5, 6, 1, 4, 3)$$

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 4 & 8 & 0 & 3 & 6 & 1 & 5 & 7 \end{pmatrix}$$



Why does a rerouting always exists?

Sketch on the whiteboard

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?With generalized de Bruijn words, we can get words of length kn instead of k^n.
2. DeBruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?With generalized de Bruijn words, we can get words of length kn instead of k^n.
2. DeBruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$ We can construct words with totally unclustered BWT by resolving ties, as described.

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?With generalized de Bruijn words, we can get words of length kn instead of k^n.
2. DeBruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$ We can construct words with totally unclustered BWT by resolving ties, as described.

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?With generalized de Bruijn words, we can get words of length kn instead of k^n.
2. DeBruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$ We can construct words with totally unclustered BWT by resolving ties, as described.

De Bruijn words and unclustered BWT

DeBruijn words are promising, but 2 things remain to solve:

1. What about arbitrary lengths?With generalized de Bruijn words, we can get words of length kn instead of k^n.
2. DeBruijn words do not *exactly* have totally unclustered BWT.
   - $\mathrm{BWT}(u) = 201 \cdot 021 \cdot 120$ We can construct words with totally unclustered BWT by resolving ties, as described.

Outline

The Burrows-Wheeler Transform

Unclustered BWT: Previous results

De Bruijn words

The ternary case: existence

Extension to all alphabets and lengths

The ternary case: lower bounds

Looking back at the binary case

Wrapping up

For alphabets of size $k \geq 3$, we have generalized de Bruijn word with totally unclustered BWT of length kn for every n.

Wrapping up

For alphabets of size $k \geq 3$, we have generalized de Bruijn word with totally unclustered BWT of length kn for every n.
Of course, ternary words are particular cases of alphabet on k letters for every $k \geq 3$.

Wrapping up

For alphabets of size $k \geq 3$, we have generalized de Bruijn word with totally unclustered BWT of length kn for every n.
Of course, ternary words are particular cases of alphabet on k letters for every $k \geq 3$.
Hence for alphabets of size $k \geq 3$, we have generalized de Bruijn word with totally unclustered BWT of length $3n$ for every n.

Wrapping up

For alphabets of size $k \geq 3$, we have generalized de Bruijn word with totally unclustered BWT of length kn for every n.
Of course, ternary words are particular cases of alphabet on k letters for every $k \geq 3$.
Hence for alphabets of size $k \geq 3$, we have generalized de Bruijn word with totally unclustered BWT of length $3n$ for every n.
What about other lengths?

Wrapping up

For alphabets of size $k \geq 3$, we have generalized de Bruijn word with totally unclustered BWT of length kn for every n.
Of course, ternary words are particular cases of alphabet on k letters for every $k \geq 3$.
Hence for alphabets of size $k \geq 3$, we have generalized de Bruijn word with totally unclustered BWT of length $3n$ for every n.
What about other lengths?
What if we want words that use *exactly* $k \geq 3$ letters?

Extending to full alphabets

0	0	1	0	2	1	1	2	2
0	1	0	2	1	1	2	2	0
0	2	1	1	2	2	0	0	1
1	0	2	1	1	2	2	0	0
1	1	2	2	0	0	1	0	2
1	2	2	0	0	1	0	2	1
2	0	0	1	0	2	1	1	2
2	1	1	2	2	0	0	1	0
2	2	0	0	1	0	2	1	1

Extending to full alphabets

0	0	1	0	2	1	1	2	2
0	1	0	2	1	1	2	2	0
0	2	1	1	2	2	0	0	1
1	0	2	1	1	2	2	0	0
1	1	2	2	0	0	1	0	2
1	2	2	0	0	1	0	2	1
2	0	0	1	0	2	1	1	2
2	1	1	2	2	0	0	1	0
2	2	0	0	1	0	2	1	1

0	0	1	0	4	1	2	5	3
0	1	0	4	1	2	5	3	0
0	4	1	2	5	3	0	0	1
1	0	4	1	2	5	3	0	0
1	2	5	3	0	0	1	0	4
2	5	3	0	0	1	0	4	1
3	0	0	1	0	4	1	2	5
4	1	2	5	3	0	0	1	0
5	3	0	0	1	0	4	1	2

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.
   - e.g. $\mathrm{BWT}(u) = 201021120$.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.
    - e.g. $\mathrm{BWT}(u) = 201021120$. Then 20102110 is still a BWT image.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.
   - e.g. $\mathrm{BWT}(u) = 201021120$. Then 20102110 is still a BWT image.
2. Either the last 2 is at the antepenultimate position.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.
   - e.g. $\mathrm{BWT}(u) = 201021120$. Then 20102110 is still a BWT image.
2. Either the last 2 is at the antepenultimate position. Then, adding a 2 just before the last position still yields a totally unclustered BWT image, of length $3n + 1$.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.
   - e.g. $\mathrm{BWT}(u) = 201021120$. Then 20102110 is still a BWT image.
2. Either the last 2 is at the antepenultimate position. Then, adding a 2 just before the last position still yields a totally unclustered BWT image, of length $3n + 1$.
   - e.g. $\mathrm{BWT}(u) = 201021201$.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.
   - e.g. $\mathrm{BWT}(u) = 201021120$. Then 20102110 is still a BWT image.
2. Either the last 2 is at the antepenultimate position. Then, adding a 2 just before the last position still yields a totally unclustered BWT image, of length $3n + 1$.
   - e.g. $\mathrm{BWT}(u) = 201021201$. Then 2010212021 is still a BWT image.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.
   - e.g. $\mathrm{BWT}(u) = 201021120$. Then 20102110 is still a BWT image.
2. Either the last 2 is at the antepenultimate position. Then, adding a 2 just before the last position still yields a totally unclustered BWT image, of length $3n + 1$.
   - e.g. $\mathrm{BWT}(u) = 201021201$. Then 2010212021 is still a BWT image.

Lastly, we can show that both cases occur for at least one word for every n.

Extension to all length

Let $n \geq 1$ and consider a de Bruijn word u with totally unclustered BWT, $w = \mathrm{BWT}(u)$. Two cases:

1. Either the last 2 is at the penultimate position. Then, removing the last 2 still yields a totally unclustered BWT image, of length $3n - 1$.
   - e.g. $\mathrm{BWT}(u) = 201021120$. Then 20102110 is still a BWT image.
2. Either the last 2 is at the antepenultimate position. Then, adding a 2 just before the last position still yields a totally unclustered BWT image, of length $3n + 1$.
   - e.g. $\mathrm{BWT}(u) = 201021201$. Then 2010212021 is still a BWT image.

Lastly, we can show that both cases occur for at least one word for every n. We finally obtained:

Theorem

For every integer $n > 0$, there exists a word u of length n over the alphabet Σ_3 having totally unclustered BWT.

Outline

generalized Euler's totient function

Definition

Let p be a prime number. The generalized Euler's totient function counts the number of polynomials over $\mathbb{F}_p$ of degree smaller than n and coprime with $X^n - 1$. It can be computed using the formula

$$\Phi_p(n) = p^n \prod_{d|(n/\lambda_p(n))} \left(1 - \frac{1}{p^{\mathrm{ord}_p(d)}}\right)^{\frac{\phi(d)}{\mathrm{ord}_p(d)}} \qquad (1)$$

generalized Euler's totient function

Definition

Let p be a prime number. The generalized Euler's totient function counts the number of polynomials over $\mathbb{F}_p$ of degree smaller than n and coprime with $X^n - 1$. It can be computed using the formula

$$\Phi_p(n) = p^n \prod_{d|(n/\lambda_p(n))} \left(1 - \frac{1}{p^{\mathrm{ord}_p(d)}}\right)^{\frac{\phi(d)}{\mathrm{ord}_p(d)}} \qquad (1)$$

Theorem (Fici and Gabory, 2025.)

The number of generalized de Bruijn words of length n over Σ_3 is $2^{n/3-1} \cdot \frac{\Phi_3(n/3)}{n/3}$

Finding the lower bound

When constructing generalized de Bruijn word with totally unclustered BWT, we untie blocks.

Finding the lower bound

When constructing generalized de Bruijn word with totally unclustered BWT, we untie blocks.
How many choices of the set of blocks to untie?

Finding the lower bound

When constructing generalized de Bruijn word with totally unclustered BWT, we untie blocks.
How many choices of the set of blocks to untie? Exactly 2^n.

Finding the lower bound

When constructing generalized de Bruijn word with totally unclustered BWT, we untie blocks.
How many choices of the set of blocks to untie? Exactly 2^n.
How many generalized de Bruijn word with totally unclustered BWT?

Finding the lower bound

When constructing generalized de Bruijn word with totally unclustered BWT, we untie blocks.
How many choices of the set of blocks to untie? Exactly 2^n.
How many generalized de Bruijn word with totally unclustered BWT? At least $\frac{1}{2} \cdot \frac{\Phi_3(n/3)}{n/3}$.

Finding the lower bound

When constructing generalized de Bruijn word with totally unclustered BWT, we untie blocks.
How many choices of the set of blocks to untie? Exactly 2^n.
How many generalized de Bruijn word with totally unclustered BWT? At least $\frac{1}{2} \cdot \frac{\Phi_3(n/3)}{n/3}$.

Theorem

For every integer $n > 0$, there exist at least $\Phi_3(\frac{n}{3})/2n = \Omega(2^{\frac{n}{3}}/n)$ word of length n over Σ_3 having a totally unclustered BWT.

Outline

Theorem (Mantaci et al., 2017)

There exists a 2n-length word with totally unclustered BWT *if and only if* $2n+1$ *is an odd prime and* 2 *generates the cyclic group* $\mathbb{Z}^*_{2n+1}$.

Theorem (Mantaci et al., 2017)

There exists a 2n-length word with totally unclustered BWT *if and only if* $2n+1$ *is an odd prime and* 2 *generates the cyclic group* $\mathbb{Z}^*_{2n+1}$.

This was because the only candidate in the binary case was $(10)^n$.

Theorem (Mantaci et al., 2017)

There exists a $2n$-length word with totally unclustered BWT if and only if $2n+1$ is an odd prime and 2 generates the cyclic group $\mathbb{Z}^*_{2n+1}$.

This was because the only candidate in the binary case was $(10)^n$.
For larger alphabets, this changes. But we can still obtain:

Theorem (Mantaci et al., 2017)

There exists a $2n$-length word with totally unclustered BWT if and only if $2n+1$ is an odd prime and 2 generates the cyclic group $\mathbb{Z}^*_{2n+1}$.

This was because the only candidate in the binary case was $(10)^n$.
For larger alphabets, this changes. But we can still obtain:

Theorem

Let $k \geq 2$, and $\alpha = k-1\ldots0$, the word obtained by concatenating each letter of Σ_k in decreasing lexicographic order. For every integer n, the word α^n is a BWT image if and only if $kn+1$ is a prime and k generates the cyclic group $\mathbb{Z}^*_{kn+1}$.

Theorem (Mantaci et al., 2017)

There exists a $2n$-length word with totally unclustered BWT if and only if $2n+1$ is an odd prime and 2 generates the cyclic group $\mathbb{Z}^*_{2n+1}$.

This was because the only candidate in the binary case was $(10)^n$.
For larger alphabets, this changes. But we can still obtain:

Theorem

Let $k \geq 2$, and $\alpha = k-1\ldots0$, the word obtained by concatenating each letter of Σ_k in decreasing lexicographic order. For every integer n, the word α^n is a BWT image if and only if $kn+1$ is a prime and k generates the cyclic group $\mathbb{Z}^*_{kn+1}$.

This is interesting, because of its relation with the following conjecture:

Conjecture (Artin's conjecture)

Let a be an integer that is not a square number and not -1. The set of prime number k such that k generates the cyclic group $\mathbb{Z}^*_a$ is infinite.

Conclusion

We obtained the following:

Conclusion

We obtained the following:

- For every $n \geq 1$, $k \geq 3$, there exists words with totally unclustered BWT of length n over k letters, and such that every letter occurs at least once.

Conclusion

We obtained the following:

- For every $n \geq 1$, $k \geq 3$, there exists words with totally unclustered BWT of length n over k letters, and such that every letter occurs at least once.
- For every $n \geq 1$, $k \geq 3$, there exists $\Omega(2^{\frac{n}{3}}/n)$ words with totally unclustered BWT of length n over k letters.

Conclusion

We obtained the following:

- For every $n \geq 1$, $k \geq 3$, there exists words with totally unclustered BWT of length n over k letters, and such that every letter occurs at least once.
- For every $n \geq 1$, $k \geq 3$, there exists $\Omega(2^{\frac{n}{3}}/n)$ words with totally unclustered BWT of length n over k letters.

You can find our preprint on arXiv:

Conclusion

We obtained the following:

- For every $n \geq 1$, $k \geq 3$, there exists words with totally unclustered BWT of length n over k letters, and such that every letter occurs at least once.
- For every $n \geq 1$, $k \geq 3$, there exists $\Omega(2^{\frac{n}{3}}/n)$ words with totally unclustered BWT of length n over k letters.

You can find our preprint on arXiv:

Thank you! Questions?