HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

ICE-TCS
Icelandic Centre of Excellence
in Theoretical Computer Science

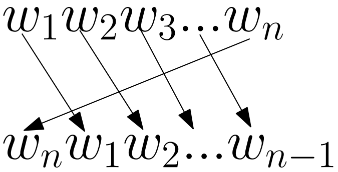
Ranking Binary Unlabelled Necklaces

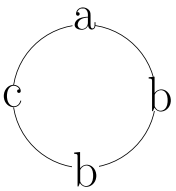
Duncan Adamson
Reykjavik University, Department of Computer Science

November 24, 2022

Necklaces

- A **necklace** is an equivalence class of words under the **cyclic shift** operation.
- The canonical representative of a necklace V is the **lexicographically smallest word** in the equivalence class, denoted $\langle V \rangle$.





abbc
bbca
bcab
cabb

Labelling

- A **labelling** of a word over the alphabet Σ is a bijective function of the form $\psi(x) : \Sigma \mapsto \Sigma$.
- The **Identity** labelling function $I(x)$ is defined as $I(x) = x$.
- The set of all labelling functions over an alphabet Σ is denoted $\mathbf{\Psi}(\Sigma)$.
- An **unlabelled word** is an equivalence class of words from Σ under the set of all labelling functions in $\mathbf{\Psi}(\Sigma)$.
- For binary alphabets $\Sigma = \{0, 1\}$, $\mathbf{\Psi}(\Sigma)$ contains two functions:
    - The **identity function** $I(x) = x$.
    - The **switch function** $S(x) = \begin{cases} 1 & x = 0 \\ 0 & x = 1 \end{cases}$

Unlabelled Word Example

Function ψ	$\psi(w)$
$\psi_1(0,1,2) = (0,1,2)$	012021120
$\psi_2(0,1,2) = (0,2,1)$	021012210
$\psi_3(0,1,2) = (1,0,2)$	102120021
$\psi_4(0,1,2) = (1,2,0)$	120102201
$\psi_5(0,1,2) = (2,0,1)$	201210012
$\psi_6(0,1,2) = (2,1,0)$	210201102

Unlabelled Necklaces

- An **unlabelled necklace** is an equivalence class over both the cyclic shift operation, and the labelling operations.
- The **canonical representation** of an unlabelled necklace is the smallest word in the equvivelence class.
- An unlabelled necklace can be thought of either as:
  - an equivalence class of unlabelled words under the cyclic shift operation.
  - an equivalence class of necklaces under the labelling operation.
- For the binary alphabet, we can partition the set of unlabelled necklaces into two subsets:
  - Unlabelled necklaces that contain two necklace classes.
  - Unlabelled necklaces that contain one necklace class.

Unlabelled Necklaces

w	$S(w)$
00011011*	11100100
00110110	11001001
01101100	10010011
11011000	**00100111**
10110001	01001110
01100011	10011100
11000110	00111001
10001101	01110010

What is ranking?

- The ranking problem asks, given an object o and a strictly ordered set **S** where each element is comparable to o, how many members of **S** are smaller than or equal to o.
- For the set of unlabelled necklaces of a given length n over an alphabet of size 2, the ordering is defined over the canonical representations.

1.	00000000	8.	00001101	15.	00011011
2.	00000001	9.	00001111	16.	00100101
3.	00000011	10.	00010001	17.	00101011
4.	00000101	10.	00010011	18.	00101101
5.	00000111	11.	00010101	19.	00110011
6.	00001001	13.	00010111	20.	01010101
7.	00001011	14.	00011001		

What is ranking?

- The ranking problem asks, given an object o and a strictly ordered set S where each element is comparable to o, how many members of **S** are smaller than or equal to o.
- For the set of unlabelled necklaces of a given length n over an alphabet of size 2, the ordering is defined over the canonical representations.

1.	00000000	8.	00001101	**15.**	*00011011*
2.	00000001	9.	00001111	16.	00100101
3.	00000011	10.	00010001	17.	00101011
4.	00000101	10.	00010011	18.	00101101
5.	00000111	11.	00010101	19.	00110011
6.	00001001	13.	00010111	20.	01010101
7.	00001011	14.	00011001		

Ranking Cyclic Words

- The problem of ranking classes of cyclic words originates from the problem of ranking de Bruijn Sequences [3].
- The first class of cyclic words to be ranked was **Lyndon words** (aperiodic necklaces).
- This was generalised to ranking necklaces [4, 5] in quadratic time, **Fixed density** necklaces in cubic time [2] and **bracelets** in $O(k^2 n^4)$ time [1].

Class	Solved by	Best Run time
Lyndon words	Kociumaka et. al. [3]	$O(n^2)$ ([5])
Necklaces	Kopparty et. al. [4]	$O(n^2)$ ([5])
Fixed Density Necklaces	Hartman and Sawada	$O(n^3)$ ([2])
Bracelets	Adamson et al.	$O(k^2 n^4)$ [1]

Ranking Unlabelled Necklaces

Idea: Rather than try to directly compute the rank of a word w within the set of unlabelled necklaces, we want to partition the set of unlabelled necklaces in to three subsets and rank w within each of them separately.

Symmetric, Asymmetric and Enclosing Necklaces

Definition

An unlabelled necklace U is **symmetric** if for every pair of words $w, v \in U$, there exists some index i such that $w = v_{i+1}, v_{i+2} \dots v_n v_1 v_2 \dots v_i$. Let $\mathcal{S}^n$ denote the set of length n symmetric necklaces.

Definition

An unlabelled necklace is **asymmetric** if it is not symmetric. Let $\mathcal{A}^n$ denote the set of length n asymetric necklaces.

Definition

An asymmetric, unlabelled necklace $U = V \cup S(V)$ **encloses** a word w if $\langle V \rangle < w < \langle S(V) \rangle$. Let $\mathcal{E}^n(w)$ denote the set of length n unlabelled necklaces enclosing w.

Ranking Unlabelled Necklaces

- Let $RN(w)$ be the number of binary (labelled) necklaces that are smaller than w and let $RU(w)$ be the number of binary unlabelled necklaces that are smaller than w.
- Each unlabelled symmetric necklace that is smaller than w corresponds to exactly one necklace that is smaller than w. Let $RS(w)$ be the number of symmetric binary necklaces that are smaller than w.
- Each unlabelled asymmetric necklace that does **not** enclose w, corresponds to exactly two necklaces that are smaller than w. Let $RA(w)$ be the number of asymmetric binary necklaces that are smaller than w and do not enclose w.
- Each unlabelled necklace that encloses w corresponds to exactly one necklace that is smaller than w. Let $RE(w)$ be the number of binary unlabelled necklaces that enclose w.

The Main Equation(s)

$$
\begin{aligned}
RU(w) &= RS(w) + RA(w) + RE(w)\\
RN(w) &= RS(w) + 2 \cdot RA(w) + RE(w)\\
RS(w) &= RN(w) - (2 \cdot RA(w) + RE(w))\\
RA(w) &= \left(RN(w) - (RS(w) + RE(w))\right)/2\\
RE(w) &= RN(w) - (RS(w) + 2 \cdot RA(w))
\end{aligned}
$$

The Main Equation(s)

$$
\begin{aligned}
\mathbf{RU}(\mathbf{w}) &= \mathbf{RS}(\mathbf{w}) + \mathbf{RA}(\mathbf{w}) + \mathbf{RE}(\mathbf{w})\\
RN(w) &= RS(w) + 2 \cdot RA(w) + RE(w)\\
RS(w) &= RN(w) - (2 \cdot RA(w) + RE(w))\\
\mathbf{RA}(\mathbf{w}) &= (\mathbf{RN}(\mathbf{w}) - (\mathbf{RS}(\mathbf{w}) + \mathbf{RE}(\mathbf{w})))\,/\mathbf{2}\\
RE(w) &= RN(w) - (RS(w) + 2 \cdot RA(w))\\
\mathbf{RU}(\mathbf{w}) &= (\mathbf{RN}(\mathbf{w}) + \mathbf{RS}(\mathbf{w}) + \mathbf{RE}(\mathbf{w}))\,/\mathbf{2}
\end{aligned}
$$

Bounding Subwords

- **Bounding subwords** have provided the key tool for ranking Unlabelled Necklaces.
- Bounding subwords provide a way of partitioning the words in Σ^m by lexicographic value relative to a (longer) word w.

$$\underbrace{1}_{\emptyset}\ \underbrace{2\ \ 3}_{\text{AAAB}}\ \underbrace{4\ \ 5\ \ 6\ \ 7}_{\text{AABB}}\ \underbrace{8}_{\text{ABBB}}\ \underbrace{9\ \ 10\ \ 11\ \ 12}_{\text{BAAA}}\ \underbrace{13\ \ 14}_{\text{BBAA}}\ \underbrace{15\ \ 16}_{\text{BBBA}}$$

Figure 1: The set of words $\{A, B\}^4$ bound by $AAABBB$.

Definition

A word $v \in \Sigma^m$ is **bound** with respect to $w \in \Sigma^n$ by the subword $u \sqsubseteq_m w$ such that:

- $u \leq v$.
- There exists no other subword $u' \sqsubseteq_m w$ such that $u < u' \leq v$.

Bounding Subwords and Ranking

- Bounding subwords are extremely useful in the problem of ranking classes of cyclic words.
- In particular, bounding subwords are very useful for **dynamic programming techniques**.
- For cyclic words, they are particularly useful for "closing" the cycle.

Lemma

Let v be a word of length $n-1$ such that:

- *$s \sqsubseteq_{n-1} w$ strictly bounds v.*
- *j is the largest value for which $v_{[n-1-j,n-1]} = w_{[1,j]}$.*

Then vx belongs to a necklace class greater than w if and only if $w_{[1,j]}xs \geq w$.

Symmetric Necklaces

Proposition

A necklace V represented by the word $w \in \Sigma^n$ is symmetric if and only if there exists some $r \in [n]$ s.t. $w_i = S(w_{i+r \bmod n})$ for every $i \in [n]$. Further, the period of w equals $2 \cdot r$ where r is the smallest rotation such that $w_{r+1}w_{r+2}\dots w_n w_1 \dots w_r = S(w)$.

Idea. We can use the above proposition to partition the set of symmetric necklaces in to subsets based on their periods.

Appraoch

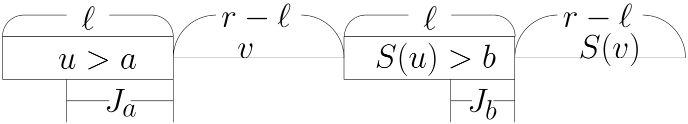
- Using the structural proposition, we can count the number of symmetric words with a period of at most $2r$ with the equation 2^r.
- This can be used to give the number of symmetric necklaces with a period of exactly $2r$ as $\frac{1}{r}\sum\limits_{l|r} \mu\left(\frac{r}{l}\right) 2^l$, and the number of necklaces with a period of at most $2r$ as $\frac{1}{r}\sum\limits_{l|r} \phi\left(\frac{r}{l}\right) 2^l$.
- Using this equation, if we can count the number of symmetric necklaces that are **larger** than w, we can get the number of words belonging to a necklace class **smaller** than w.

Symmetric Necklaces larger than w

Goal. Count the number of words of length r that are greater than w under *every* rotation.

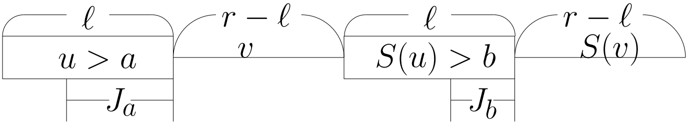
Outline.

- We achieve this through an **iterative** approach.
- The idea is to count the number of **prefixes** of length ℓ from the number of prefixes of length $\ell - 1$.
- By repeating this r-times, we get the number of words of length r the belong to a necklace class larger than w.



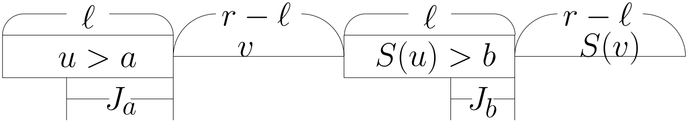
After the first ℓ symbols, we know the number of ℓ-length prefixes where:

- The first ℓ-symbols (forming the prefix u) are bound by the word $a \sqsubseteq_\ell w$.
- The symbols form $r + 1$ to $r + \ell$ (equal to $S(u)$) are bound by the word $v \sqsubseteq_\ell w$.
- J_a is the length of the longest suffix of u that is a prefix of w.
- J_b is the length of the longest suffix of $S(u)$ that is a prefix of w.



We need to work out the number of values of v such that $u : v : S(u) : S(v) > w$, which holds if and only if:

- $w_1 w_2 \dots w_{J_a} : v : b : S(v) : a > w$.
- $w_1 w_2 \dots w_{J_b} : S(v) : a : v : b > w$.



Using a dynamic programming technique, we can determine the number of words belonging to a necklace greater than w in $O(n^5)$ time due to:

- $O(n)$ possible prefix lengths.
- $O(n)$ possible values of a.
- $O(n)$ possible values of J_a.
- $O(n)$ possible values of b.
- $O(n)$ possible values of J_b.

Combined with an additional factor of $O(n)$ for the number of factors gives a total complexity of $O(n^6)$

Enclosing Necklaces

- Enclosing necklaces are ranked via a similar technical solution.
- At a high level, we use the same tools as in the symmetric case.
- We need a small amount of extra work in order to compute the number of enclosing necklaces, giving a total complexity of $O(n^6 \log n)$.

Putting it all together

- Using the equation $RU(w) = (RN(w) + RS(w) + RE(w))\,/2$ we can compute:
  - $RN(w)$ in $O(n^2)$ time.
  - $RS(w)$ in $O(n^6)$ time.
  - $RE(w)$ in $O(n^6 \log n)$ time.
- As each of these complexities are independent, the total complexity of computing $RU(w)$ is $O(n^6 \log n)$

Future Work on ranking

- $O(n^6 \log n)$ seems surprisingly high. Can it be brought down?
- Can unlabelled necklaces of an arbitrary (finite) alphabet be ranked in polynomial time?
- Can Chord diagrams be ranked in polynomial time?