Permutrees Sorting

Daniel Tamayo Jiménez

Université Paris-Saclay (LISN)

Joint work with Vincent Pilaud and Viviane Pons

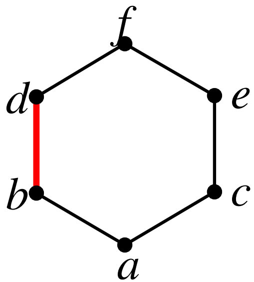
Formal Power Series and Algebraic Combinatorics (FPSAC 2021)
Bar-Ilan University, Ramat-Gan (online)
January 19, 2021

Lattice Quotients

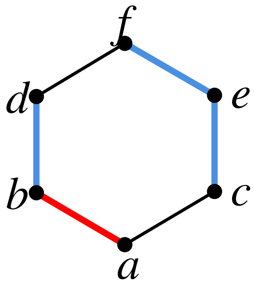
A lattice quotient is an equivalence relationship preserving meets and joins as follows

$$x \equiv x' \text{ and } y \equiv y' \Rightarrow \begin{array}{l} x \vee y \equiv x' \vee y', \\ x \wedge y \equiv x' \wedge y'. \end{array}$$

In the case of the weak order:

- 

  No other relation is affected.

- 

$$e \vee b \equiv e \vee a \implies f \equiv e,$$

$$c \vee b \equiv c \vee a \implies f \equiv c,$$

$$d \wedge f \equiv d \wedge e \implies d \equiv a.$$

Lattice Quotients

Our quotients are generated by relations $s_i - s_i s_j$ where $s_i = (i\, i+1)$ is a simple transposition.

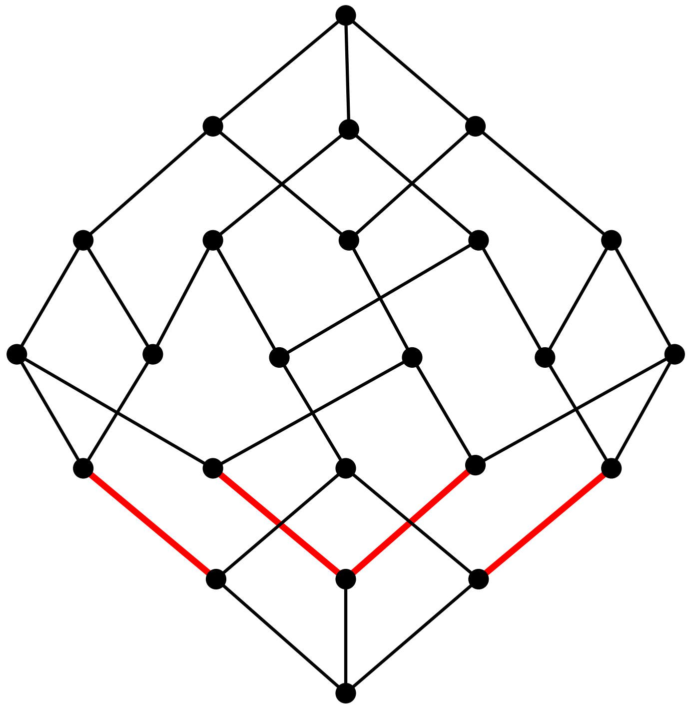
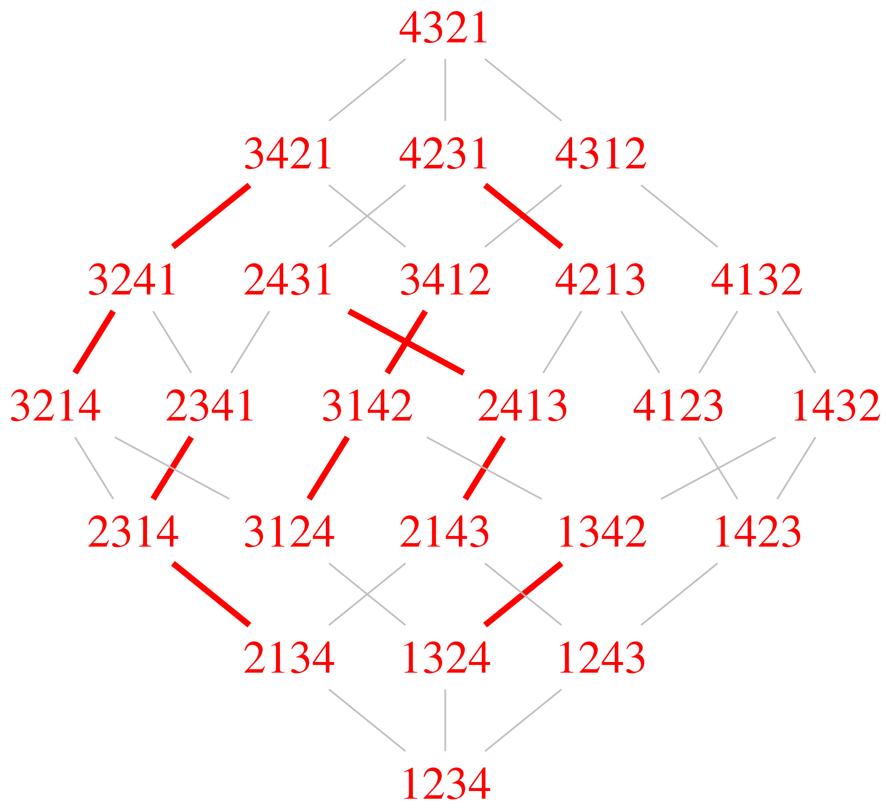


Figure 1: Possible generators of lattice quotients.

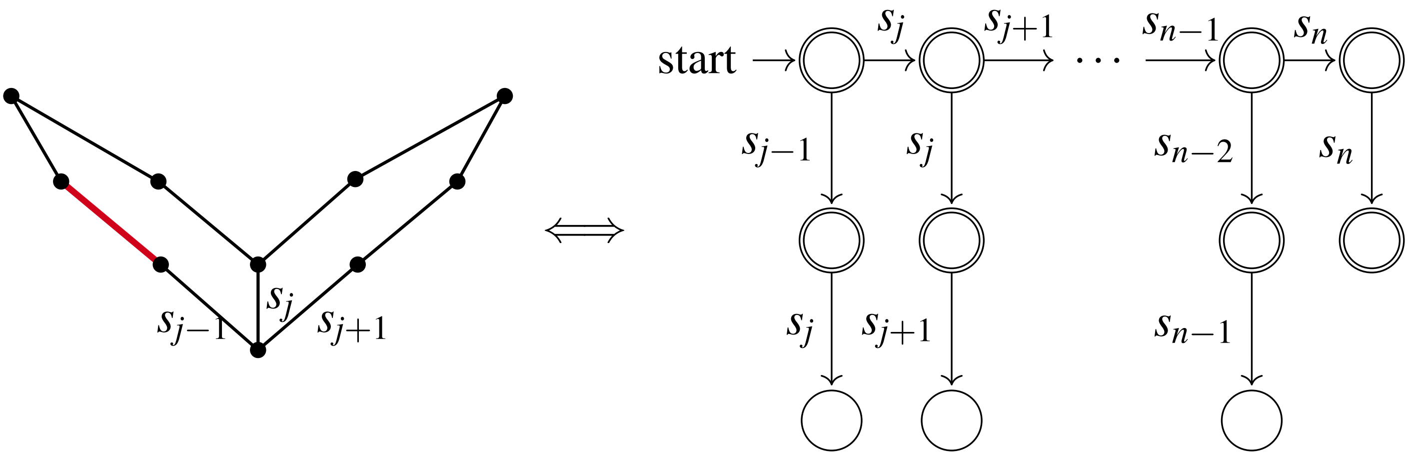
Example

The Tamari lattice is a particular case through the Sylvester congruence.



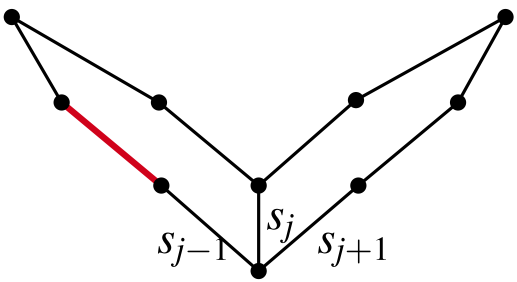
Automata

We define our automata in a recursive way depending on which relation of the braid relation $(s_i s_j)^3 = e$ is considered.

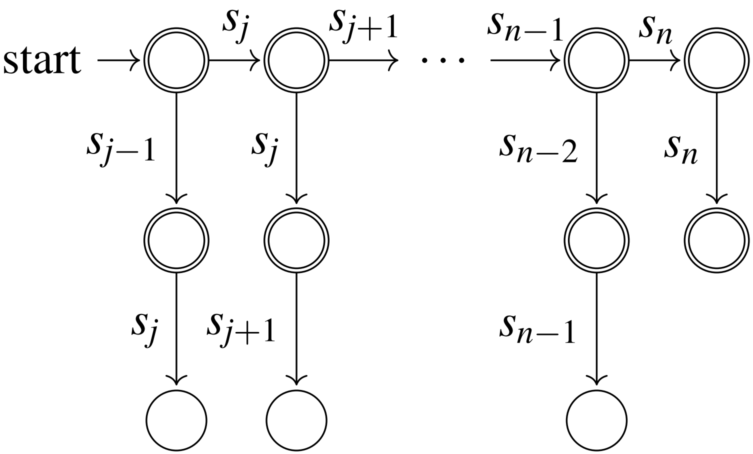


Automata

We define our automata in a recursive way depending on which relation of the braid relation $(s_i s_j)^3 = e$ is considered.



$\iff$



Theorem. [Pilaud, Pons, T. '20]
A permutation has a reduced word accepted by our automaton if and only if it is minimal in a congruence class of the corresponding permutree quotient.

Generating Trees

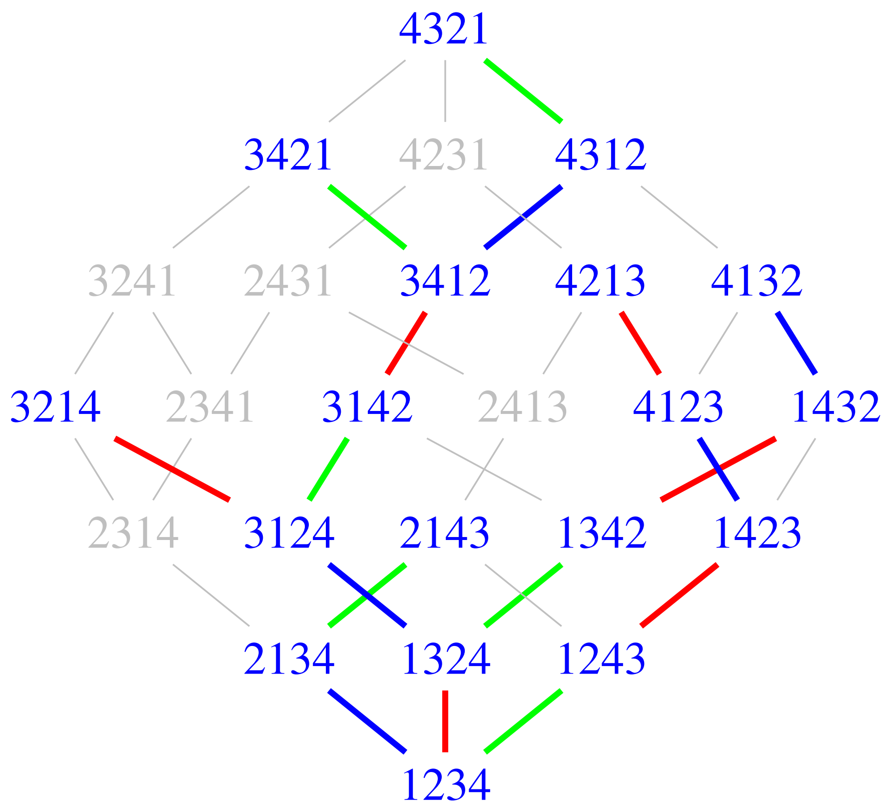


Figure 2: The set of reduced words accepted by our automata.

Properties

Accepted reduced words of a permutation

- describe a permutation that avoids certain patterns (e.g. $2ki$ for $1 \leq i < 2 < k \leq 4$ in the block above).
- are closed by prefix.
- are all accepted in the same state.
- include a special one that prioritizes the top of the automaton.

Sorting Permutrees

We define a sorting algorithm that determines if a permutation is minimal in its corresponding permutree congruence class.

Given a permutation π and an integer j, we compute a candidate reduced word for π that is accepted by the automata which we call the **permutree sorting word** of π.

π	w	j	ℓ
3421	ε	2	2
2431	s_2	3	1
1432	$s_2 \cdot s_1$	3	3
1342	$s_2 \cdot s_1 \cdot s_3$	4	2
1243	$s_2 \cdot s_1 \cdot s_3 \cdot s_2$	4	3
1234	$s_2 \cdot s_1 \cdot s_3 \cdot s_2 \cdot s_3$	4	

Table 1: Permutree sorting of $\pi = 3421$ with $j = 2$.

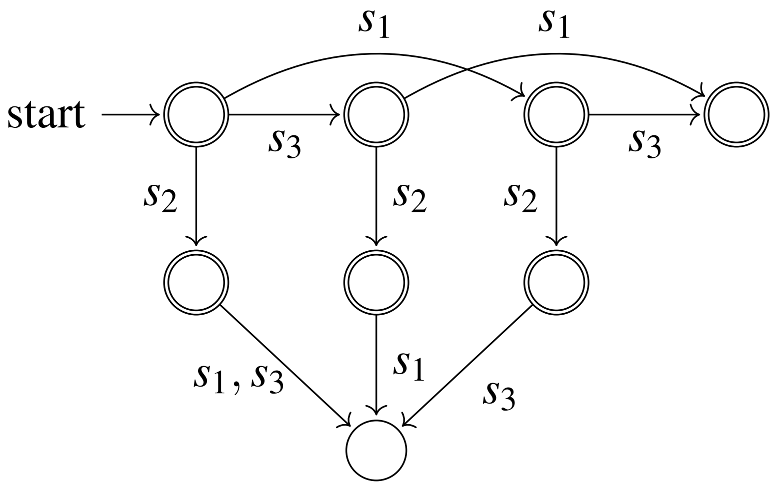
Algorithmic Characterization

Theorem. [Pilaud, Pons, T. '20]

The permutree sorting word of a permutation π is a reduced word of π if and only if π is minimal in its congruence class.

Multiple Relations

A similar automata exists for congruence relations generated by several braid relations



References

Donald E. Knuth.
The art of computer programming. Volume 3.
Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1973.
Sorting and searching, Addison-Wesley Series in Computer Science and Information Processing.

Vincent Pilaud and Viviane Pons.
Permutrees.
Algebraic Combinatorics, 1(2):173–224, 2018.

Nathan Reading.
Clusters, Coxeter-sortable elements and noncrossing partitions.
Trans. Amer. Math. Soc., 359(12):5931–5958, 2007.

Some developments

- Generalizable to type B.
- Similarities in Coxeter D and H.
- Computationally faster than doing lattice congruences in SageMath (albeit some details).

Coxeter Groups

It is a group generated by elements s_i that satisfy relations $(s_i s_j)^{m_{ij}} = e$ encoded as vertices and edges of the following graphs:

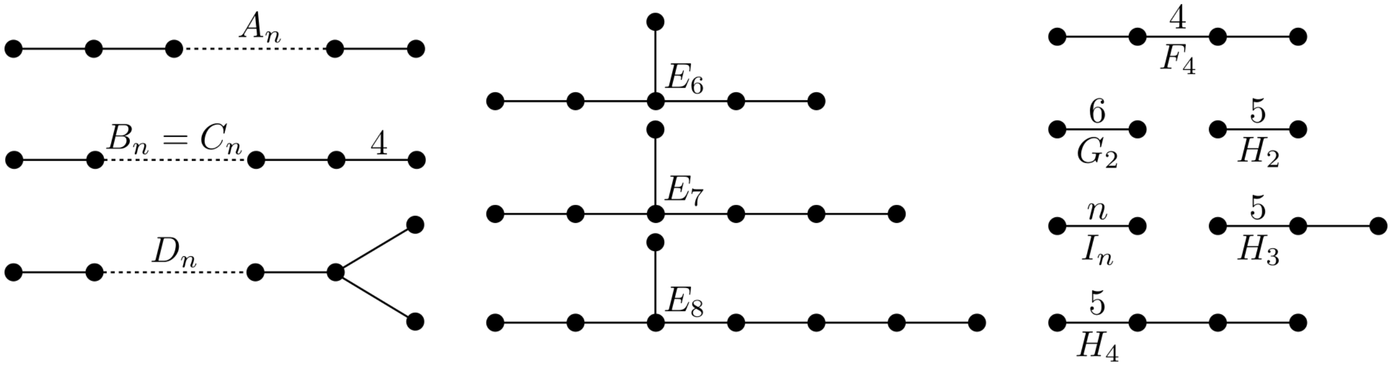


Figure 3: Finite Coxeter Groups (source: Wikipedia)

Coxeter Groups

It is a group generated by elements s_i that satisfy relations $(s_i s_j)^{m_{ij}} = e$ encoded as vertices and edges of the following graphs:

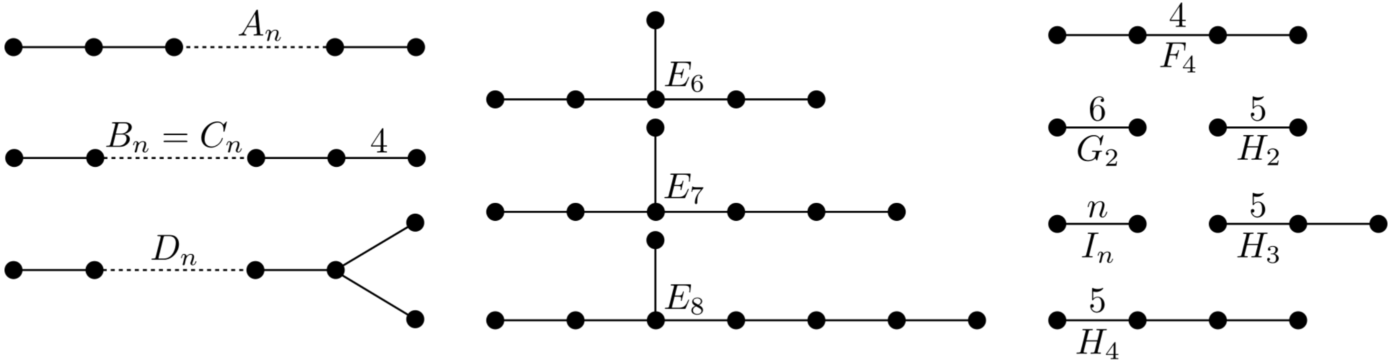


Figure 3: Finite Coxeter Groups (source: Wikipedia)

Each Coxeter Group has a weak order lattice associated to it where we can define permutree congruences.

Other types

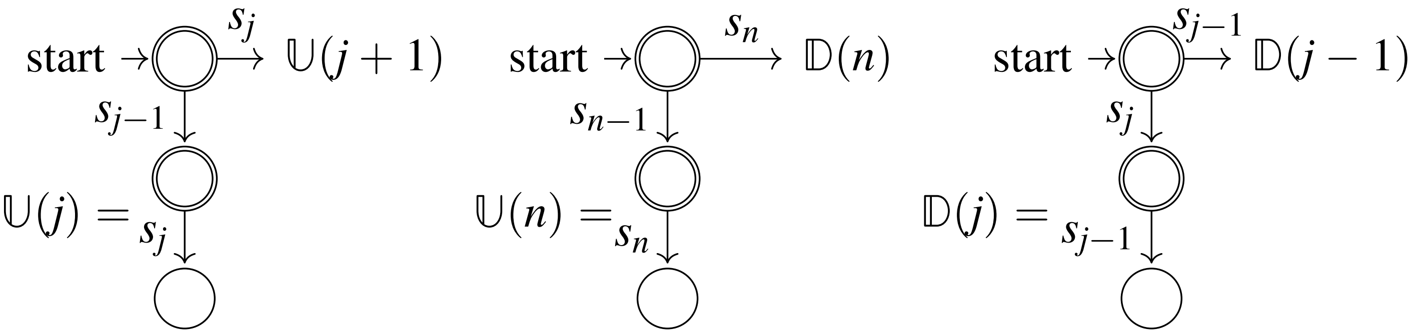


Figure 4: Recursive definition of the automata $\mathbb{U}(j)$ and $\mathbb{D}(j)$ in type B (above) and the corresponding automata that form $\mathbb{U}(2)$ in D_4 (below).