From tidying books to shuffling cards

Bérénice Delcroix-Oger
joint with Jovana Obradović (Serbian Academy of Science)
and Pierre-Louis Curien (CNRS-IRIF, Université Paris Cité)

Conférence MIPS
29 mai 2026

Outline

Tidying books

Outline

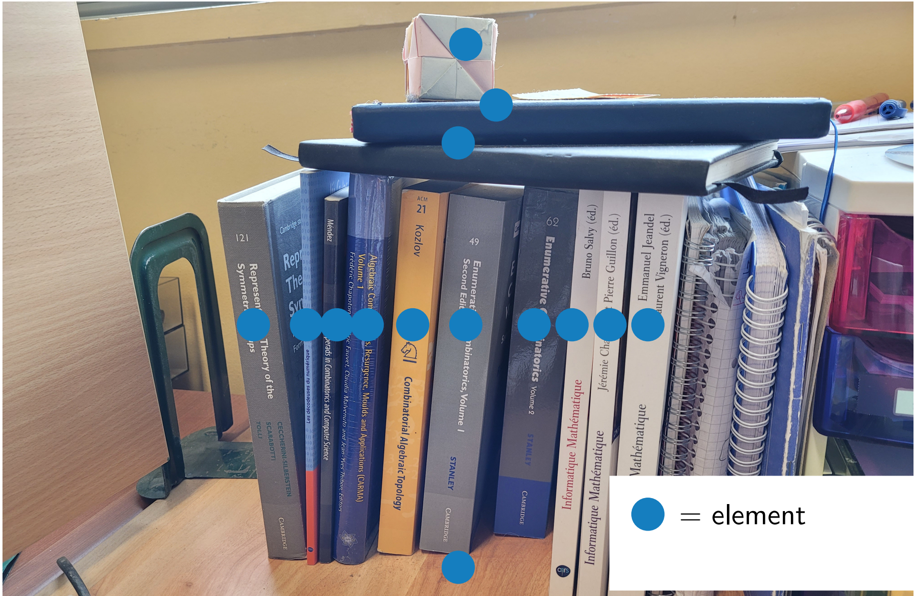
Tidying books

Tidying books

Tidying books

Tidying books

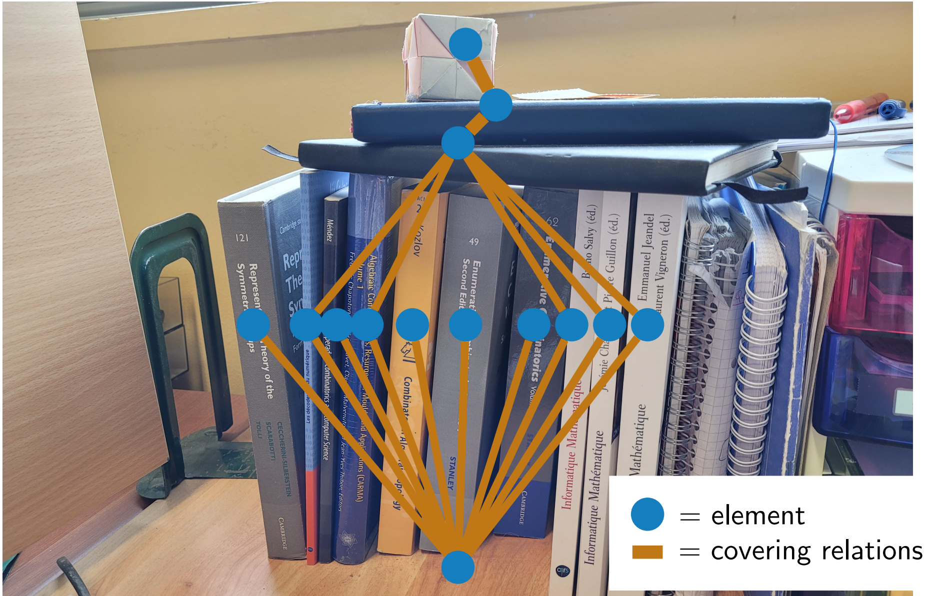
Tidying books



Tidying books

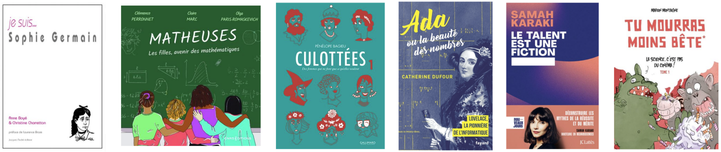
Tidying books

Tidying books



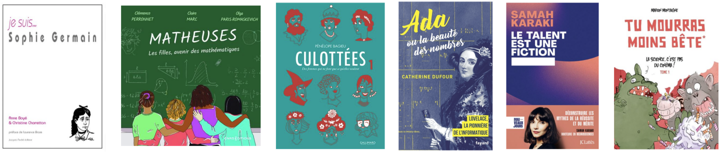
Sorting books

- On a shelf, the books are **totally ordered**



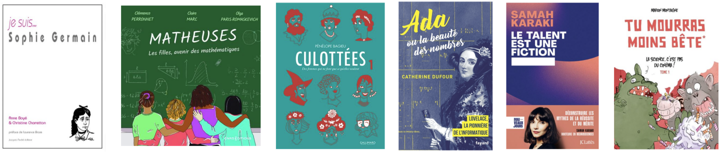
Sorting books

- On a shelf, the books are **totally ordered**
- Here, we want to sort them in the alphabetic order



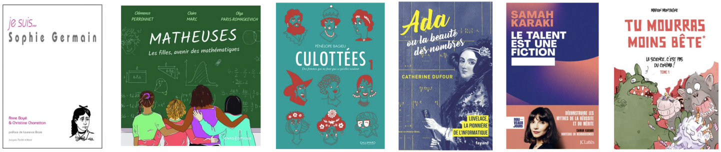
Sorting books

- On a shelf, the books are **totally ordered**
- Here, we want to sort them in the alphabetic order
- Rule : we can only exchange consecutive books



Sorting books

- On a shelf, the books are **totally ordered**
- Here, we want to sort them in the alphabetic order
- Rule : we can only exchange consecutive books



Main questions :

- How many comparisons ? Is it efficient ?

Why do we need efficient sorting algorithms ?

Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

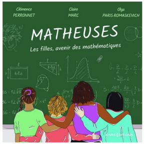
Sorting books : The one-handed librarian's sort aka. Bubble sort

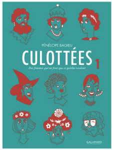
Algorithm

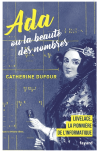
Going from left to right, the librarian

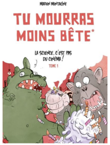
- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning









Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

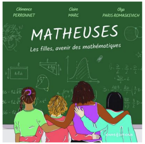
Sorting books : The one-handed librarian's sort aka. Bubble sort

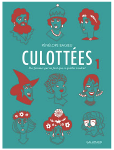
Algorithm

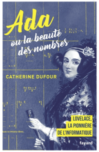
Going from left to right, the librarian

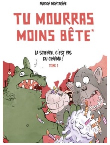
- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning









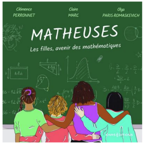
Sorting books : The one-handed librarian's sort aka. Bubble sort

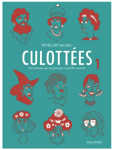
Algorithm

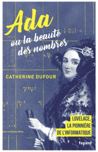
Going from left to right, the librarian

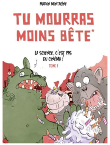
- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning









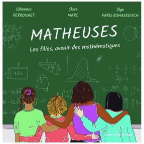
Sorting books : The one-handed librarian's sort aka. Bubble sort

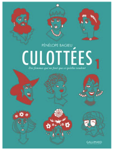
Algorithm

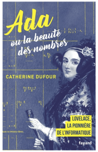
Going from left to right, the librarian

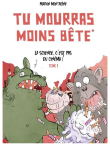
- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning









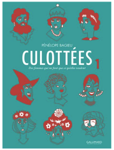
Sorting books : The one-handed librarian's sort aka. Bubble sort

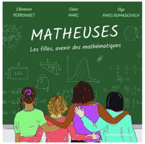
Algorithm

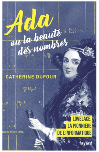
Going from left to right, the librarian

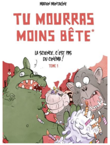
- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning









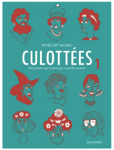
Sorting books : The one-handed librarian's sort aka. Bubble sort

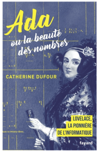
Algorithm

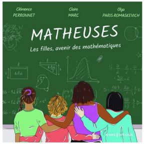
Going from left to right, the librarian

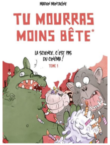
- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning









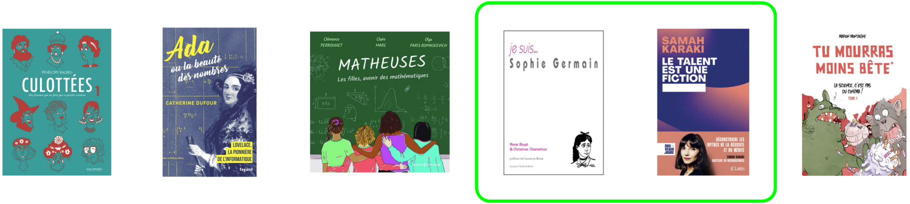
Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning



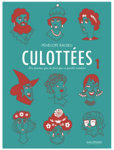
Sorting books : The one-handed librarian's sort aka. Bubble sort

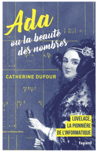
Algorithm

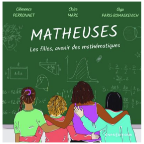
Going from left to right, the librarian

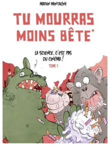
- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning









Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

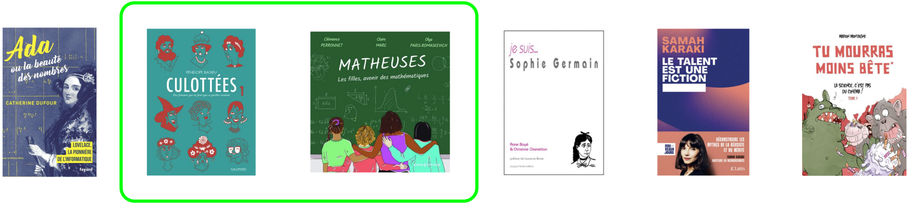
Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning



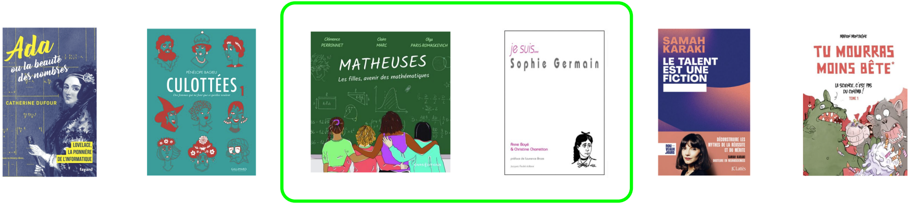
Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning



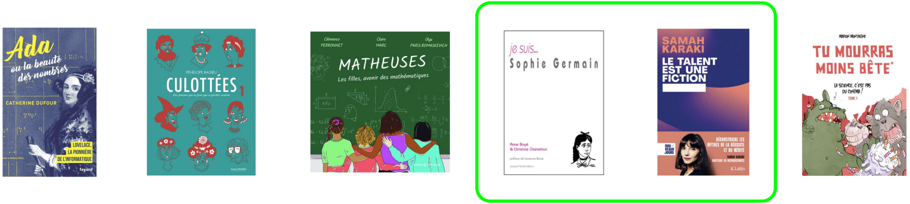
Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning



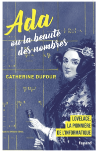
Sorting books : The one-handed librarian's sort aka. Bubble sort

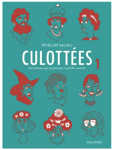
Algorithm

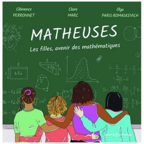
Going from left to right, the librarian

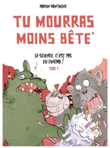
- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning









Sorting books : The one-handed librarian's sort aka. Bubble sort

Algorithm

Going from left to right, the librarian

- compares two books
- exchanges them if needed

At the end of the shelf, starts again from the beginning

→ Sorted !

Sorting permutations

Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

Sorting permutations

Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

21
|
12

Sorting permutations

Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

321

21

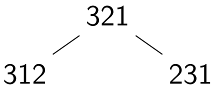
|

12

Sorting permutations

Questions

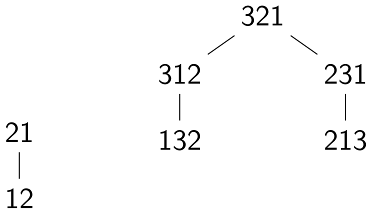
- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?



Sorting permutations

Questions

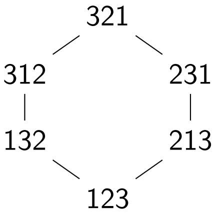
- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?



Sorting permutations

Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

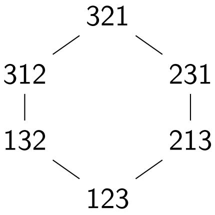


Sorting permutations

1234

Questions

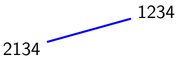
- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

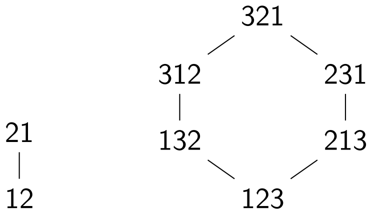


Sorting permutations

Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

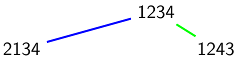


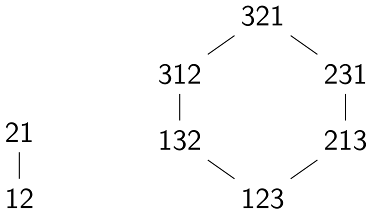


Sorting permutations

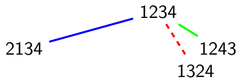
Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?



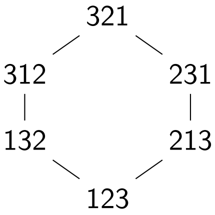


Sorting permutations



Questions

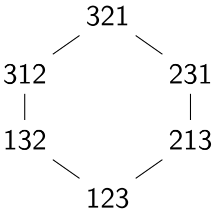
- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

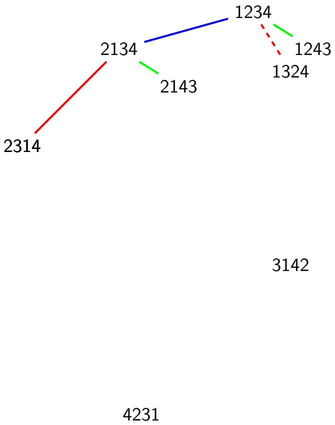


Sorting permutations

Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

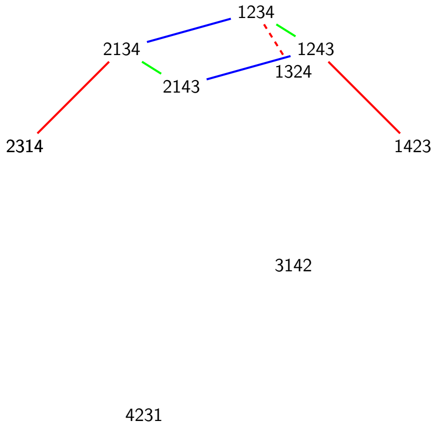


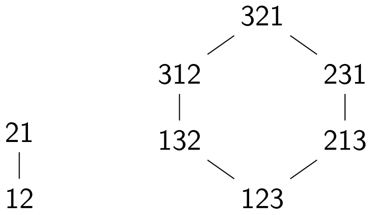


Sorting permutations

Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

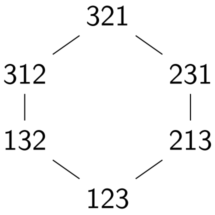


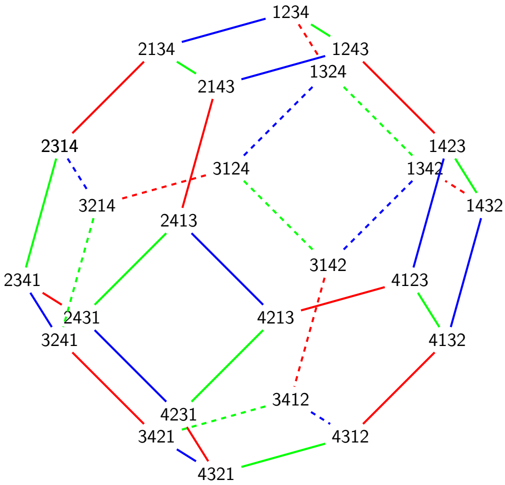


Sorting permutations

Questions

- Does the order in which we do the exchange (from left to right) matter ?
- How many exchanges must be done ?

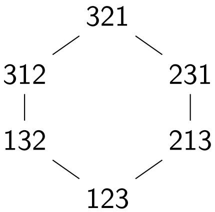


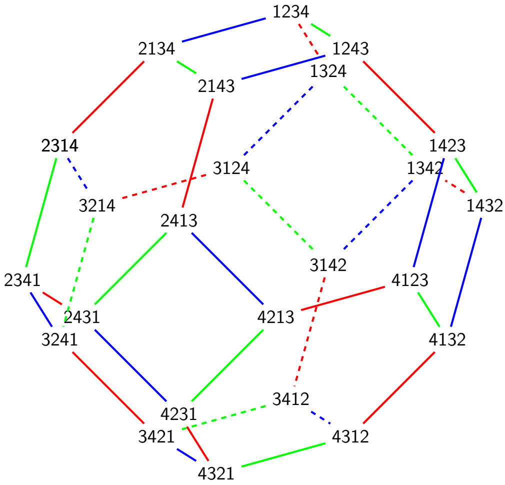


Sorting permutations

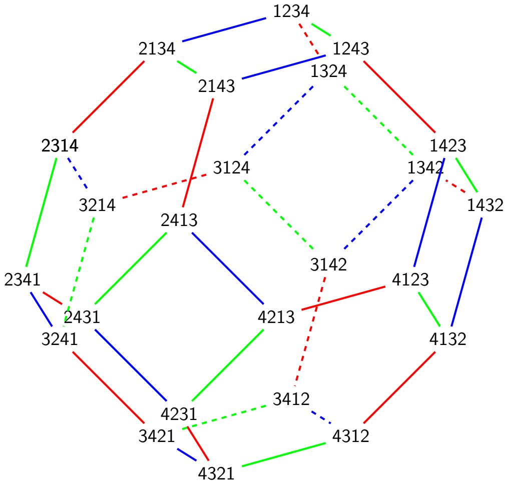
Answers [Friend, 1956]

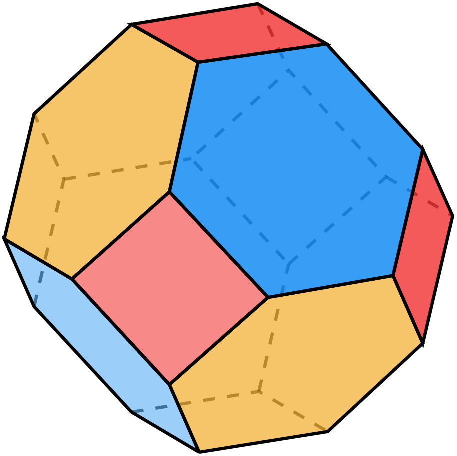
- Does the order in which we do the exchange (from left to right) matter ? No
- How many exchanges must be done ? At most $\frac{n(n-1)}{2}$





Weak Bruhat order and permutohedron





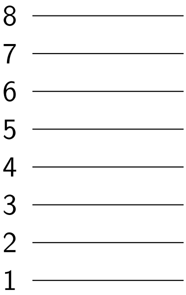
Shuffling cards

Outline

1. Tidying books
2. Shuffling cards
3. A touch of magic

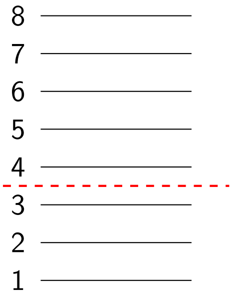
Why shuffling cards ?

A naive (but useful) shuffle



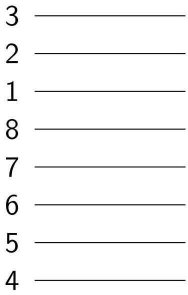
1. Cut the deck
2. Put the bottom part above

A naive (but useful) shuffle



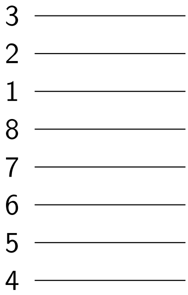
1. Cut the deck
2. Put the bottom part above

A naive (but useful) shuffle



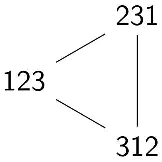
1. Cut the deck
2. Put the bottom part above

A naive (but useful) shuffle



1. Cut the deck
2. Put the bottom part above

For $n = 3$



How many shuffles ?

n

How do you shuffle cards ?

More we shuffle cards, the better they are shuffled ?

https://www.youtube.com/shorts/pwW6-YpWUuo

Shuffles (in math)

Let $\mathcal{A}$ be a finite alphabet and $\mathcal{A}^*$ its set of (possibly empty) words (finite sequences of letters).
For any $a, b \in \mathcal{A}$ and $m, n \in \mathcal{A}^*$ the shuffle product is:

$$a.m \sqcup\!\sqcup\, b.n = a.(m \sqcup\!\sqcup\, b.n) + b.(a.m \sqcup\!\sqcup\, n),$$

with $\varepsilon \sqcup\!\sqcup\, m = m \sqcup\!\sqcup\, \varepsilon = m$, where ε is the empty word.

Example :

$ana \sqcup\!\sqcup\, bob =$

Shuffles (in math)

Let $\mathcal{A}$ be a finite alphabet and $\mathcal{A}^*$ its set of (possibly empty) words (finite sequences of letters).
For any $a, b \in \mathcal{A}$ and $m, n \in \mathcal{A}^*$ the shuffle product is:

$$a.m \sqcup\!\sqcup\, b.n = a.(m \sqcup\!\sqcup\, b.n) + b.(a.m \sqcup\!\sqcup\, n),$$

with $\varepsilon \sqcup\!\sqcup\, m = m \sqcup\!\sqcup\, \varepsilon = m$, where ε is the empty word.

Example :

$ana \sqcup\!\sqcup\, bob = anabob + anbaob + anboab + anboba + abnaob + abnoab + abnoba + abonab + abonba + abobna$

Shuffles (in math)

Let $\mathcal{A}$ be a finite alphabet and $\mathcal{A}^*$ its set of (possibly empty) words (finite sequences of letters).
For any $a, b \in \mathcal{A}$ and $m, n \in \mathcal{A}^*$ the shuffle product is:

$$a.m \sqcup\!\sqcup\, b.n = a.(m \sqcup\!\sqcup\, b.n) + b.(a.m \sqcup\!\sqcup\, n),$$

with $\varepsilon \sqcup\!\sqcup\, m = m \sqcup\!\sqcup\, \varepsilon = m$, where ε is the empty word.

Example :

$ana \sqcup\!\sqcup\, bob = anabob + anbaob + anboab + anboba + abnaob + abnoab + abnoba + abonab + abonba + abobna$
$+ banaob + banoab + banoba + baonab + baonba + baobna + boanab + boanba + boabna + bobana$

Dovetail shuffle and probabilities

- Mixing the deck "1 2 3 4", we can get the following decks :
  - 1 2 3 4,
  - 1 3 2 4,
  - 1 3 4 2,
  - 3 1 2 4,
  - 3 1 4 2
  - 3 4 1 2.

Dovetail shuffle and probabilities

- Mixing the deck ”1 2 3 4”, we can get the following decks :
  - 1 2 3 4,
  - 1 3 2 4,
  - 1 3 4 2,
  - 3 1 2 4,
  - 3 1 4 2
  - 3 4 1 2.
- Mixing two times this deck, we can reach all the possible decks

Dovetail shuffle and probabilities

- Mixing the deck "1 2 3 4", we can get the following decks :
  - 1 2 3 4,
  - 1 3 2 4,
  - 1 3 4 2,
  - 3 1 2 4,
  - 3 1 4 2
  - 3 4 1 2.
- Mixing two times this deck, we can reach all the possible decks

But

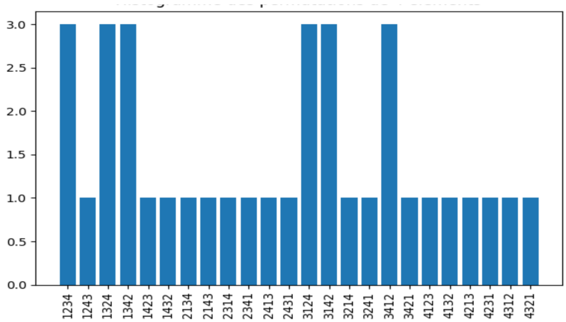
We didn't get them with the same probability.

Dovetail shuffle and probabilities

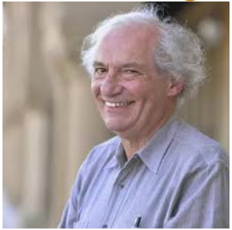
- Mixing the deck "1 2 3 4", we can get the following decks :
  - 1 2 3 4,
  - 1 3 2 4,
  - 1 3 4 2,
  - 3 1 2 4,
  - 3 1 4 2
  - 3 4 1 2.
- Mixing two times this deck, we can reach all the possible decks

But

We didn't get them with the same probability.

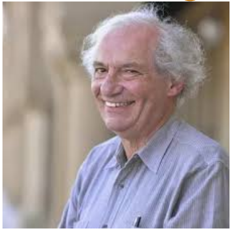


Persi Diaconis



- Magician, student of Dai Vernon
- At 29 : PhD in mathematics
- At 53 : Professor at Stanford

Persi Diaconis



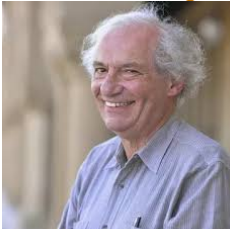
- Magician, student of Dai Vernon
- At 29 : PhD in mathematics
- At 53 : Professor at Stanford

Question :

How many shuffles to get a well-shuffled deck of 52 cards ?

Persi Diaconis

- Magician, student of Dai Vernon
- At 29 : PhD in mathematics
- At 53 : Professor at Stanford



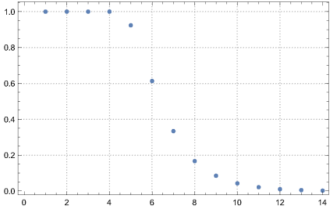
Question :

How many shuffles to get a well-shuffled deck of 52 cards ?

TABLE 3

Total variation distance for m shuffles of 25, 32, 52, 78, 104, 208 or 312 distinct cards

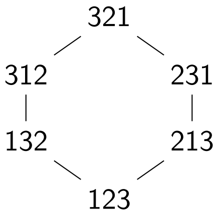
m	1	2	3	4	5	6	7	8	9	10
25	1.000	1.000	0.999	0.775	0.437	0.231	0.114	0.056	0.028	0.014
32	1.000	1.000	1.000	0.929	0.597	0.322	0.164	0.084	0.042	0.021
52	1.000	1.000	1.000	1.000	0.924	0.614	0.334	0.167	0.085	0.043
78	1.000	1.000	1.000	1.000	1.000	0.893	0.571	0.307	0.153	0.078
104	1.000	1.000	1.000	1.000	1.000	0.988	0.772	0.454	0.237	0.119
208	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.914	0.603	0.329
312	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999	0.883	0.565



A touch of magic

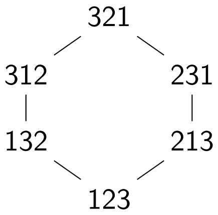
Outline

Link between shuffling cards and tidying books



$$\begin{cases} 1 \shuffle 23 & = 123 + 213 + 231 \\ 1 \shuffle 32 & = 132 + 312 + 321 \\ 12 \shuffle 3 & = 123 + 132 + 312 \\ 21 \shuffle 3 & = 213 + 231 + 321 \end{cases}$$

Link between shuffling cards and tidying books

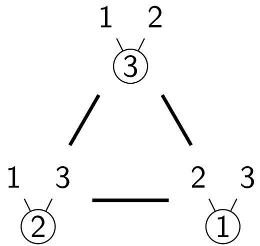


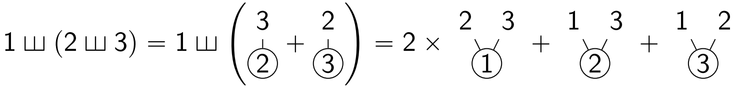
$$\begin{cases} 1 \shuffle 23 & = 123 + 213 + 231 \\ 1 \shuffle 32 & = 132 + 312 + 321 \\ 12 \shuffle 3 & = 123 + 132 + 312 \\ 21 \shuffle 3 & = 213 + 231 + 321 \end{cases}$$

Theorem [Loday-Ronco, 02]

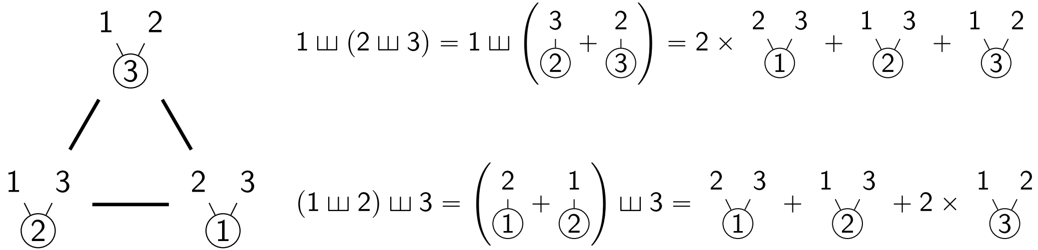
$$u \shuffle v = \sum_{(u \triangleleft v) \leqslant w \leqslant (u \triangleright v)} w$$

Could we shuffle something else ?

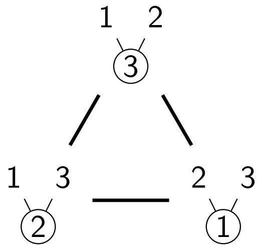


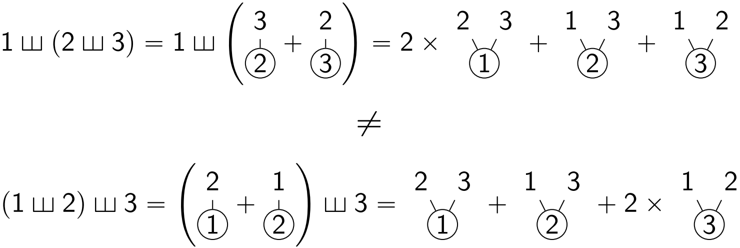


Could we shuffle something else ?

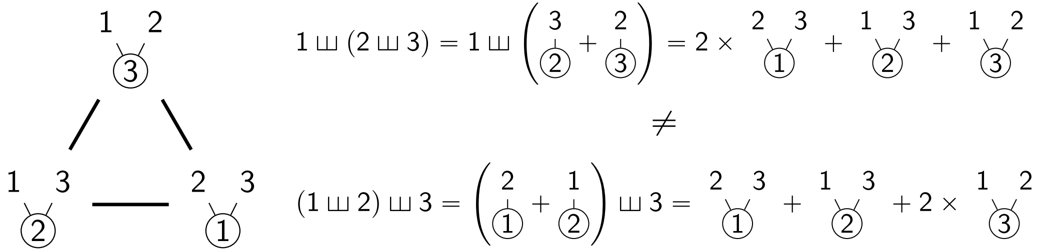


Could we shuffle something else ?





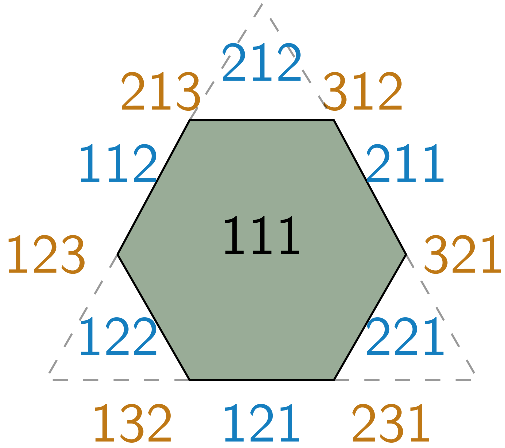
Could we shuffle something else ?



Idea :

- Shuffle faces (and not just vertices)
- Of "good" nestohedra

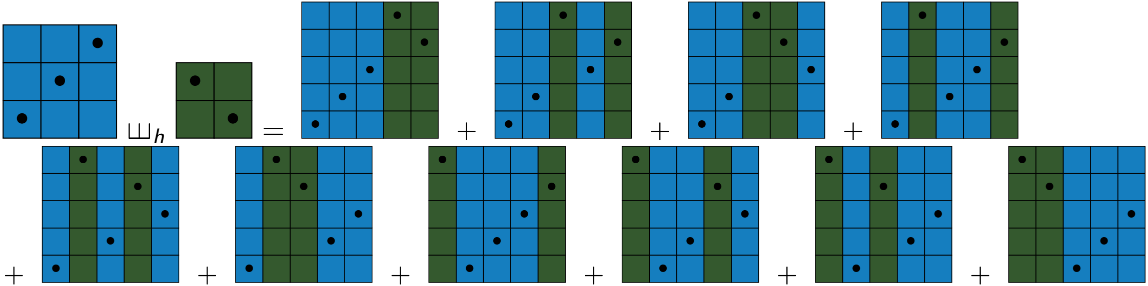
Let's shuffle faces !



Back to the previous shuffle on permutations : Horizontal shuffle

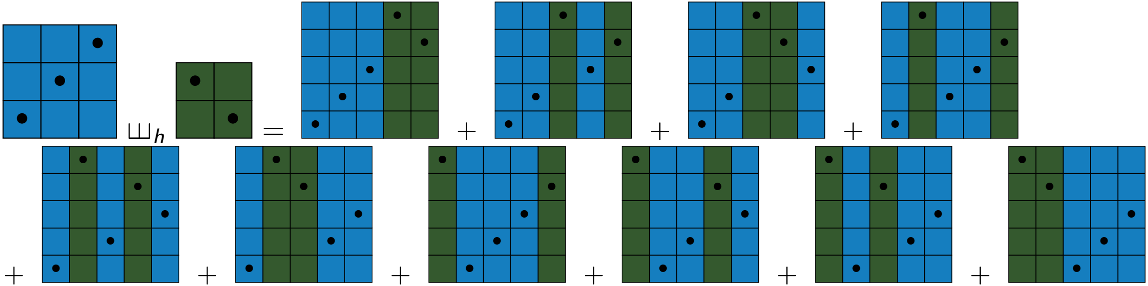
Example:

$123 \sqcup\!\sqcup_h 21 = 12354 + 12534 + 12543 + 15234 + 15243 + 15423 + 51234 + 51243 + 51423 + 54123$



Example:

$123 \sqcup\!\sqcup_h 21 = 12354 + 12534 + 12543 + 15234 + 15243 + 15423 + 51234 + 51243 + 51423 + 54123$



Question

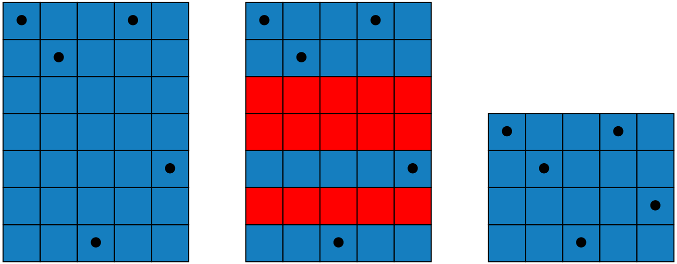
How could we get something like 12321 ?

Vertical shuffle

For σ and τ two surjections,

$$\sigma \sqcup\!\sqcup_v \tau = \sum_{\substack{\mathrm{std}(s)=\sigma \\ \mathrm{std}(t)=\tau}} s.t,$$

where std(76173)=43132



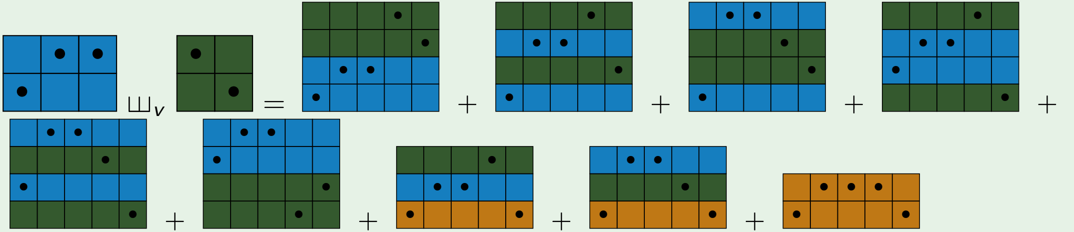
Vertical shuffle

For σ and τ two surjections,

$$\sigma \sqcup\!\sqcup_v \tau = \sum_{\substack{\mathrm{std}(s)=\sigma \\ \mathrm{std}(t)=\tau}} s.t,$$

Example:

$122 \sqcup\!\sqcup_v 21 = 12243 + 13342 + 14432 + 23341 + 24431 + 34421 + 12231 + 13321 + 12221$



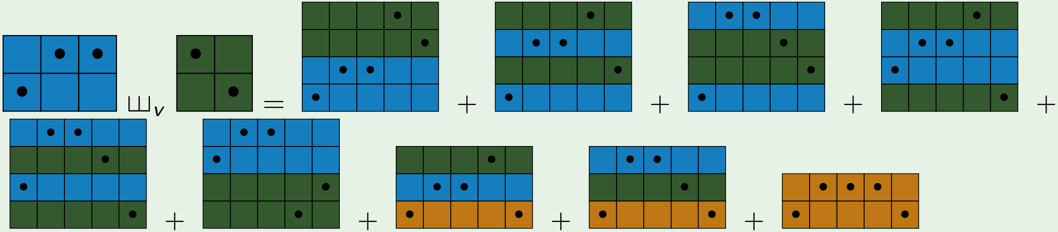
Vertical shuffle: tridendriform products

For σ and τ two surjections,

$$\sigma \prec_v \tau = \sum_{\substack{\mathrm{std}(s)=\sigma \\ \mathrm{std}(t)=\tau \\ \min(s.t)=\min(s)}} s.t, \quad \sigma \succ_v \tau = \sum_{\substack{\mathrm{std}(s)=\sigma \\ \mathrm{std}(t)=\tau \\ \min(s.t)=\min(t)}} s.t, \quad \sigma \cdot_v \tau = \sum_{\substack{\mathrm{std}(s)=\sigma \\ \mathrm{std}(t)=\tau \\ \min(s.t)=\min(s)=\min(t)}} s.t,$$

Example:

$122 ⧢_v 21 = 12243 + 13342 + 14432 + 23341 + 24431 + 34421 + 12231 + 13321 + 12221$



Tridendriform algebras

Definition (Loday, Ronco, 2004 ; Chapoton 2002)

A tridendriform algebra is a vector space A endowed with products $\prec: A \otimes A \to A$, $\cdot : A \otimes A \to A$ and $\succ: A \otimes A \to A$, such that:

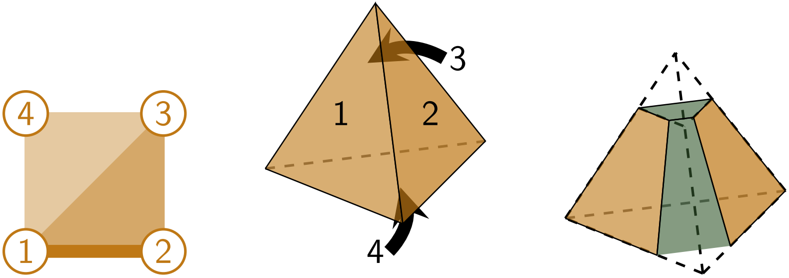
1. $(a \prec b) \prec c = a \prec (b * c)$,
2. $(a * b) \succ c = a \succ (b \succ c)$,
3. $(a \succ b) \prec c = a \succ (b \prec c)$,
4. $(a \cdot b) \cdot c = a \cdot (b \cdot c)$,
5. $(a \succ b) \cdot c = a \succ (b \cdot c)$,
6. $(a \prec b) \cdot c = a \cdot (b \succ c)$,
7. $(a \cdot b) \prec c = a \cdot (b \prec c)$,

with $* = \prec + \cdot + \succ$

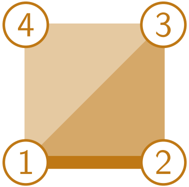
Theorem [Burgunder-Curien-Ronco, 15]

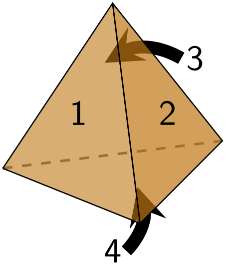
The tridendriform algebra of surjections is free.

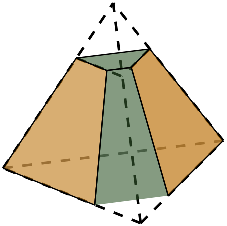
Nestohedra

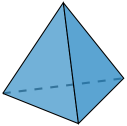


Nestohedra

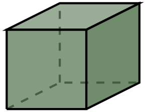




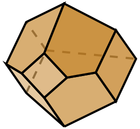




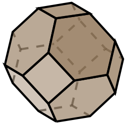
Simplices



Hypercubes



Associahedra



Permutohedra

Faces of the nestohedra

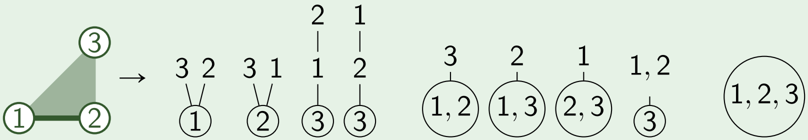
Constructs

A construct c of a hypergraph H is a rooted tree whose vertices form a partition of $V(H)$ and defined inductively by:

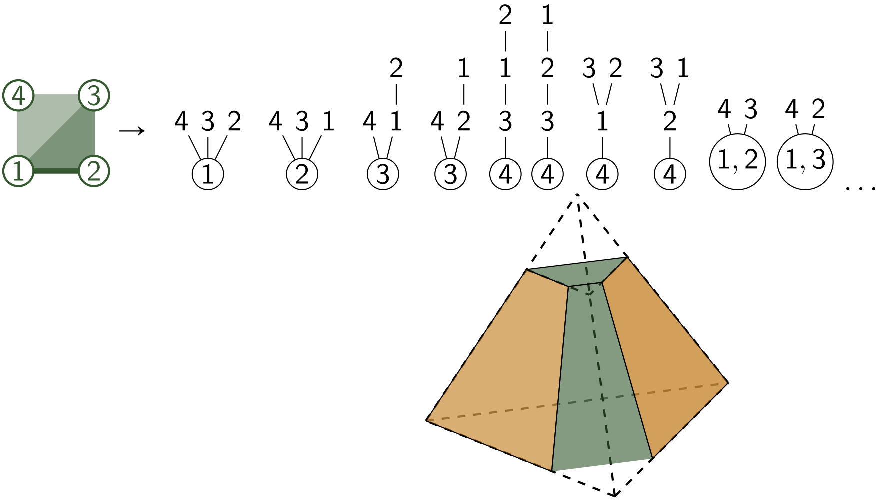
- c has only one node labelled by $V(H)$,
- or the root of c is $E \subseteq V(H)$
  and each of its children is a construct of a connected component of $H - E$.

The set of constructs of a given hypergraph labels faces of the associated polytope.

First example:



Second example geometrically



Heuristics for a tridendriform structure

Let $\mathbf{H}^{\mathcal{X}}$ be a family of hypergraph polytopes, indexed by some finite sets $\mathcal{X}$ (sets of vertices of the associated hypergraphs).
For $S = A(S_1, \ldots, S_m)$ and $T = B(T_1, \ldots, T_n)$ two constructs of $\mathbf{H}^{\mathcal{X}}$ and $\mathbf{H}^{\mathcal{Y}}$ respectively ($\mathcal{X}, \mathcal{Y}$ disjoint), we would like to define the following operations

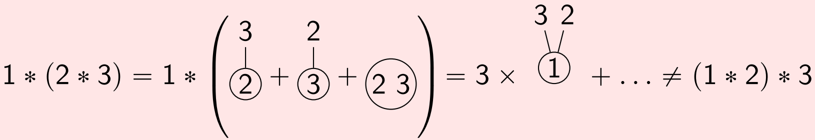
- $S \prec T$ as a sum of constructs of $\mathbf{H}^{\mathcal{X} \cup \mathcal{Y}}$ having **root** A**,**
- $S \succ T$ as a sum of constructs of $\mathbf{H}^{\mathcal{X} \cup \mathcal{Y}}$ having **root** B**,**
- $S \cdot T$ as a sum of constructs of $\mathbf{H}^{\mathcal{X} \cup \mathcal{Y}}$ having **root** $A \cup B$**.**

Heuristics for a tridendriform structure

Let $\mathbf{H}^{\mathcal{X}}$ be a family of hypergraph polytopes, indexed by some finite sets $\mathcal{X}$ (sets of vertices of the associated hypergraphs).
For $S = A(S_1, \dots, S_m)$ and $T = B(T_1, \dots, T_n)$ two constructs of $\mathbf{H}^{\mathcal{X}}$ and $\mathbf{H}^{\mathcal{Y}}$ respectively ($\mathcal{X}, \mathcal{Y}$ disjoint), we would like to define the following operations

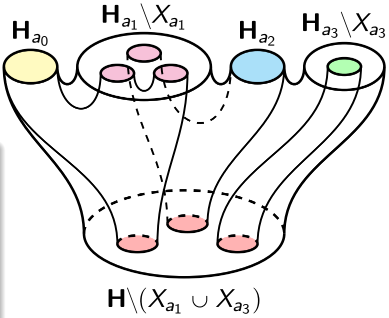
- $S \prec T$ as a sum of constructs of $\mathbf{H}^{\mathcal{X} \cup \mathcal{Y}}$ having **root** A,
- $S \succ T$ as a sum of constructs of $\mathbf{H}^{\mathcal{X} \cup \mathcal{Y}}$ having **root** B,
- $S \cdot T$ as a sum of constructs of $\mathbf{H}^{\mathcal{X} \cup \mathcal{Y}}$ having **root** $A \cup B$.

First problem in hypercubes



Strict and quasi-strict teams [Curien–D.O.–Obradovic, 25]

A team is a (resp. quasi-strict) strict team if each edge of a hypergraph $\mathbf{H}_a$ is connected in $\mathbf{H}$ (resp. or totally disconnected).



$(X_{a_0} = X_{a_2} = \varnothing)$

Examples:

- Strict teams : Associahedra, Permutohedra, Restrictohedra, . . .
- Quasi-strict teams : Simplices, Hypercubes, Erosohedra, . . .

Shuffle product

Considering a team E and denoting by δ a tuple of constructs of the team's participating hypergraphs, we inductively associate to δ a sum of constructs of the supporting hypergraph:

$$*(\delta) = \sum_{\emptyset \subset B \subseteq A} q^{|B|-1} *_B (\delta),$$

where

$$*_B(\delta) = (\bigcup_{b \in B} X_b)(*(\delta_1^B), \ldots, *(\delta_{n_B}^B)).$$

Proposition (Curien-D.O.-Obradovic, 25)

$$*_B(\delta) \quad = \quad \sum_{U:\mathbf{H},\ root(U)=X_B \text{ and } \forall a \in A, U_{|\mathbf{H}_a}=C_a} q^{\mu^\tau(U)-|B|+1} U.$$

Our result

Theorem [**BDO**–Curien–Obradović, 25 and 26+]

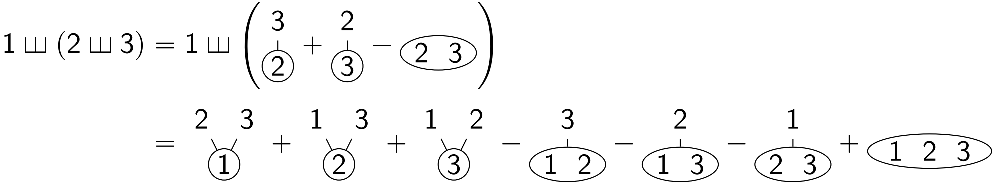
- q-tridendriform product on faces of "strict" nestohedra (including associahedra, permutohedra, restrictohedra, friesohedra...) + link with an order
- (-1)-tridendriform product on faces of "semi-strict" nestohedra (including simplices, hypercubes, erosohedra...)

Our result

Theorem [BDO–Curien–Obradović, 25 and 26+]

- q-tridendriform product on faces of "strict" nestohedra (including associahedra, permutohedra, restrictohedra, friesohedra...) + link with an order
- (-1)-tridendriform product on faces of "semi-strict" nestohedra (including simplices, hypercubes, erosohedra...)

Back to the simplex:

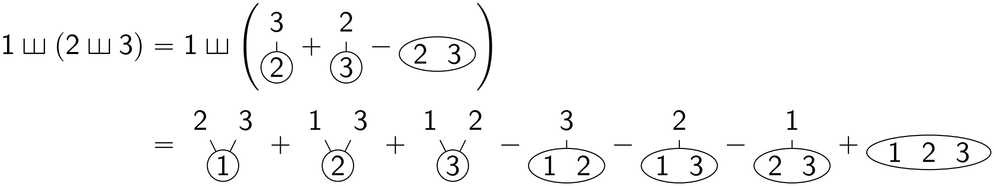


Our result

Theorem [BDO–Curien–Obradović, 25 and 26+]

- *q*-tridendriform product on faces of "strict" nestohedra (including associahedra, permutohedra, restrictohedra, friesohedra...) + link with an order
- (-1)-tridendriform product on faces of "semi-strict" nestohedra (including simplices, hypercubes, erosohedra...)

Back to the simplex:



Thank you !