Course IMSc Chennai, India

January-March 2017

Enumerative and algebraic combinatorics,
a bijective approach:

commutations and heaps of pieces

(with interactions in physics, mathematics and computer science)

Monday and Thursday 14h-15h30

www.xavierviennot.org/coursIMSc2017

IMSc

January-March 2017

Xavier Viennot

CNRS, LaBRI, Bordeaux

www.xavierviennot.org

Chapter 1
Commutation monoids
and
heaps of pieces:

basic definitions
(1)

IMSc, Chennai
5 January 2017

§1 Commutation monoids

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$(a+b)^2 = \cancel{a^2 + 2ab + b^2}$$

$$\text{if} \quad ab \neq ba$$

$$= a^2 + ab + ba + b^2$$

$a, b, c, d, \ldots$

letters
formal variables

$$ad = da \qquad ab \neq ba$$
$$cd = dc \qquad ac \neq ca$$
$$bc = cb \qquad bd \neq db$$

$a, b, c, d, \ldots$

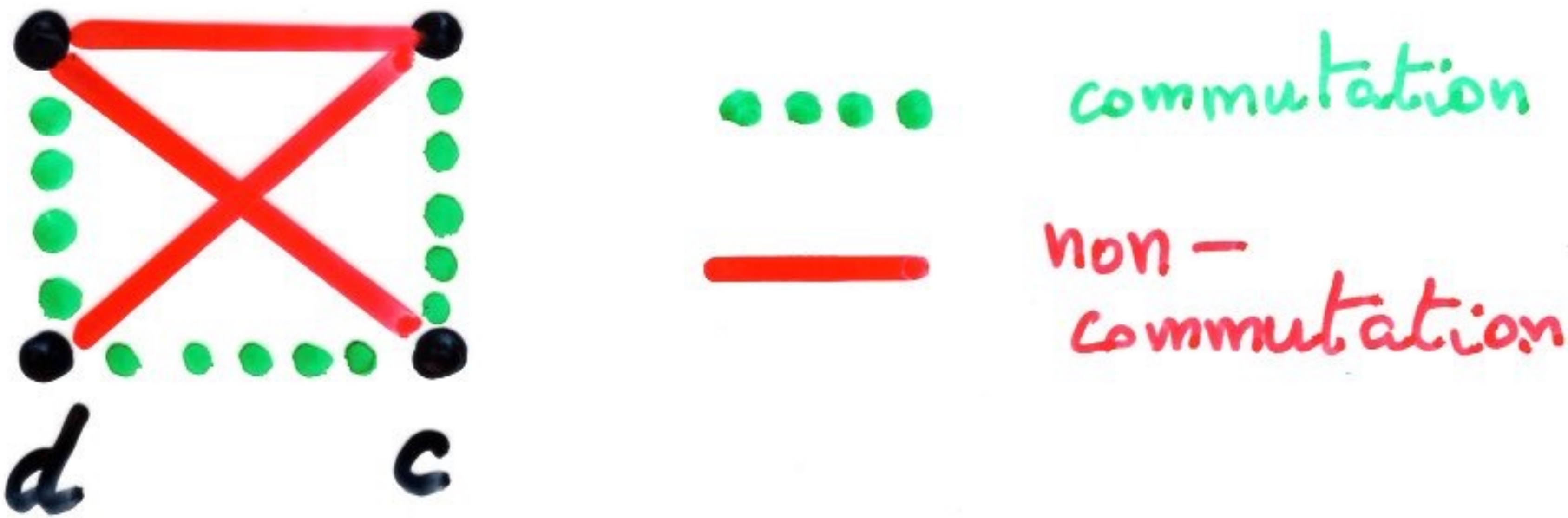
letters

formal variables

$ad = da \qquad ab \neq ba$

$cd = dc \qquad ac \neq ca$

$bc = cb \qquad bd \neq db$



$abcad$

word
monomial

$w = a\underline{bc}ad$

$ad = da$

$cd = dc$

$bc = cb$

abcad

word
monomial

$w = abcad$

$acbad$

$ad = da$

$cd = dc$

$bc = cb$

abcad

word
monomial

$$\begin{tikzcd} w = abcad \arrow[r, dash] \arrow[d, dash] & ab\underline{cd}a \\ acbad & \end{tikzcd}$$

$ad = da$

$cd = dc$

$bc = cb$

abcad

word
monomial

$$\begin{tikzcd}
w = abcad \arrow[r, dash] \arrow[d, dash] & abcda \arrow[dr, dash] & \\
acb\underline{ad} & & abdca
\end{tikzcd}$$

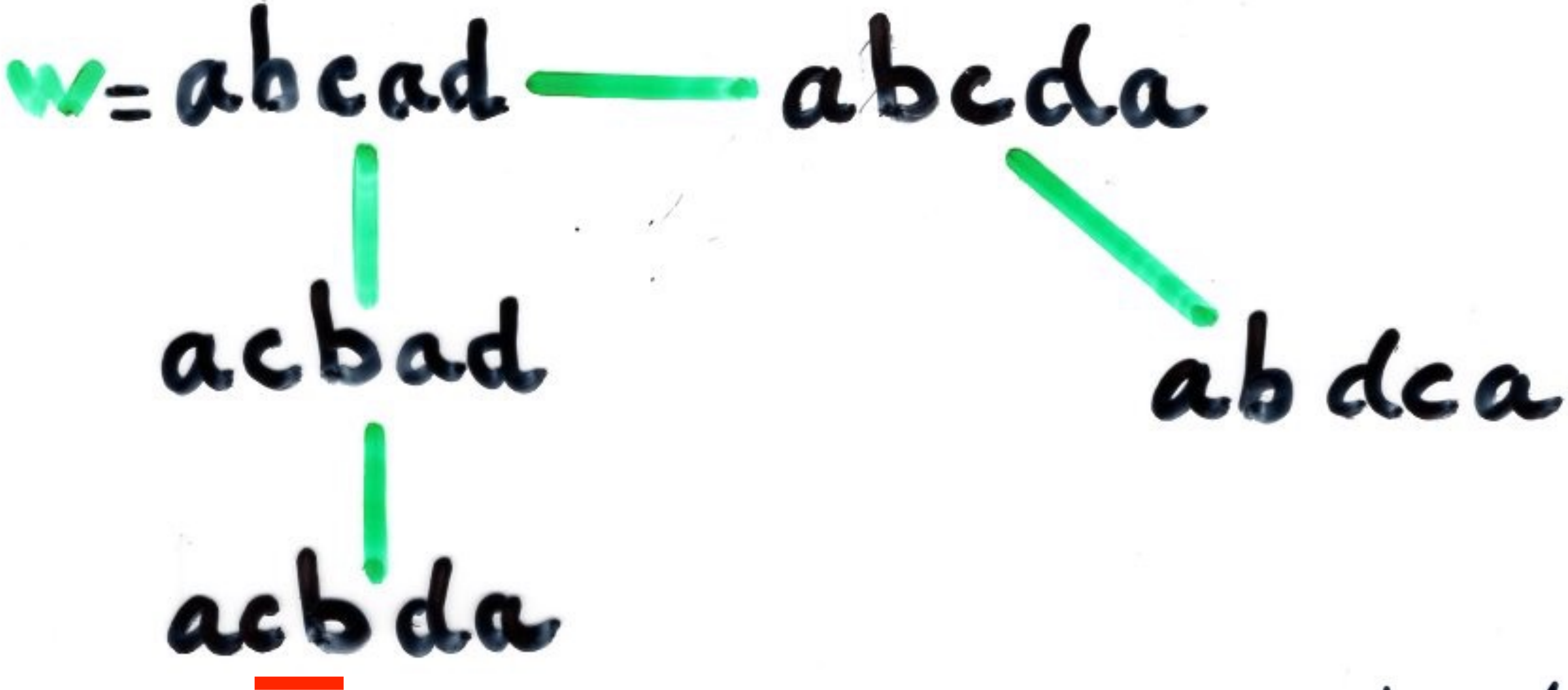
$ad = da$

$cd = dc$

$bc = cb$

abcad

word
monomial



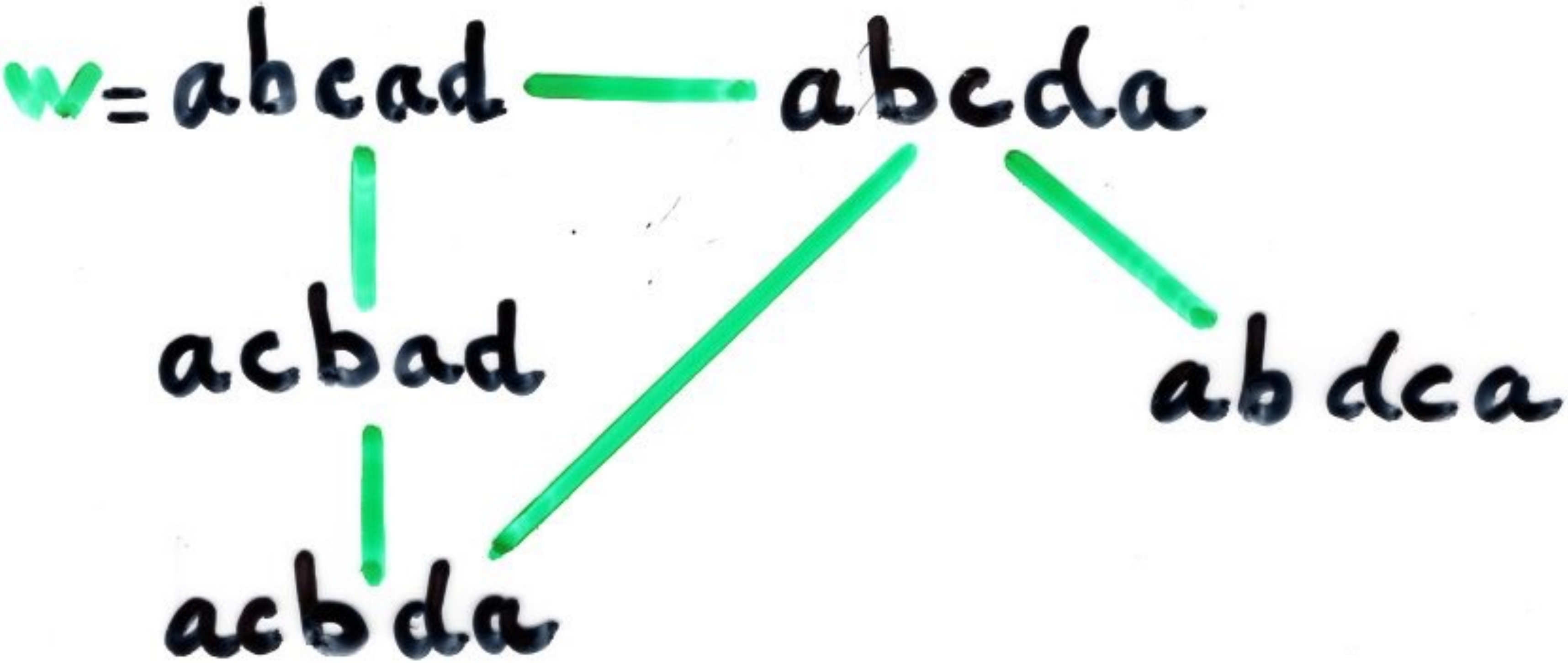
$ad = da$

$cd = dc$

$bc = cb$

abcad

word
monomial



$ad = da$

$cd = dc$

$bc = cb$

ex: $A = \{a, b, c, d\}$

$C \quad \begin{cases} ad = da \\ bc = cb \\ cd = dc \end{cases}$

equivalence class

$$
\begin{tikzcd}
w = abcad \arrow[rr, dash] \arrow[d, dash] & & abcda \arrow[dr, dash] & \\
acbad \arrow[d, dash] & & & abdca \\
acbda \arrow[uurr, dash] & & &
\end{tikzcd}
$$

Cartier-Foata monography
in SLC Séminaire Lotharingien
de Combinatoire
(2006)

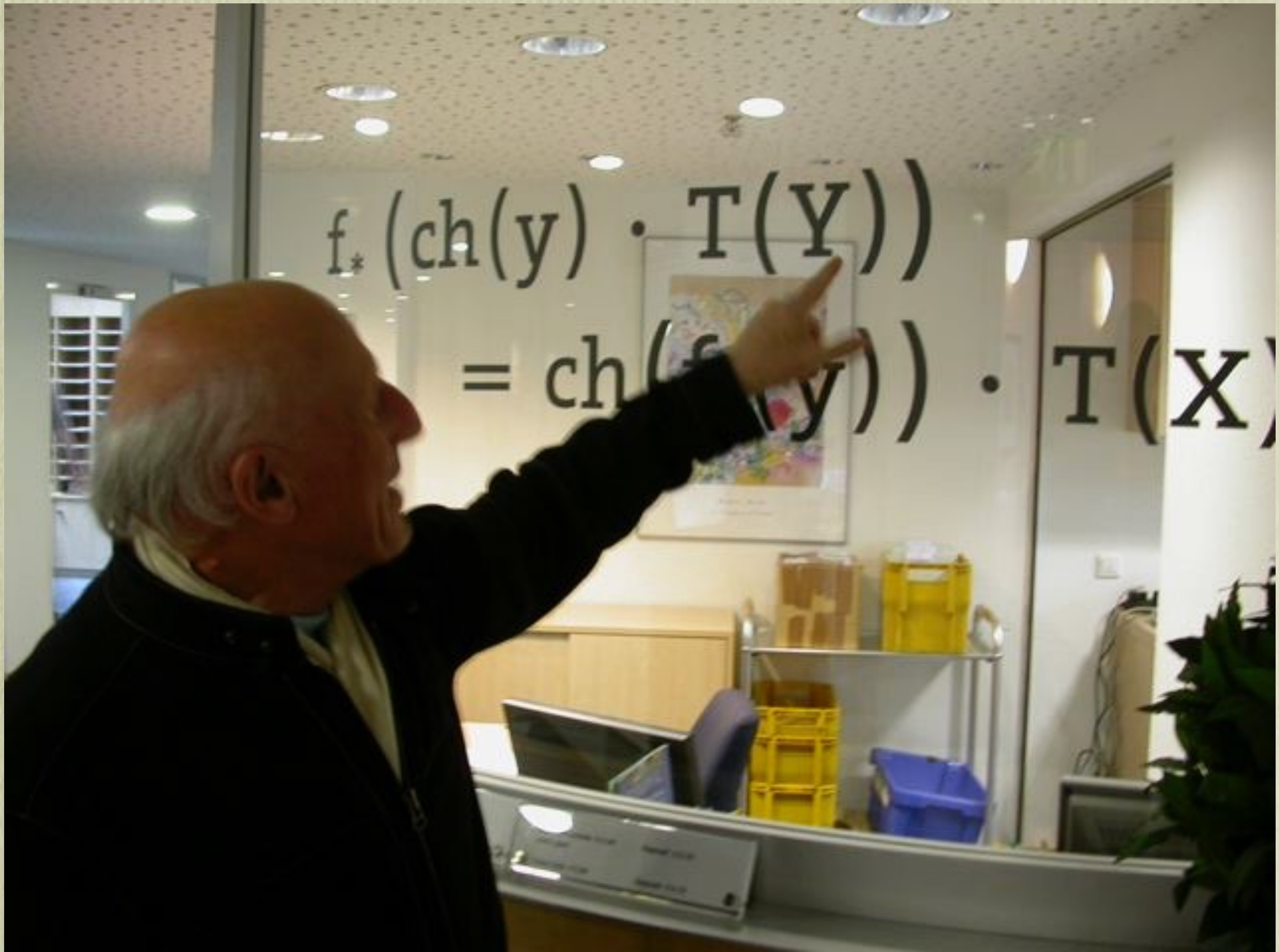
http://www.mat.univie.ac.at/~slc/

with an appendix
by C. Krattenthaler

Cartier-Foata **commutation** **monoid**

Lecture Note in Maths n° 85 (1969)

"Problèmes combinatoires de
commutation et réarrangements"


$f_*(ch(y) \cdot T(Y))$

A, C)
such
form
mutatio of ∞, 1, 2, . . . , n
, an ele
begi
Pro g d is a bijection of ∞S
ont

monoid $\quad M \quad (u,v) \to u \bullet v$

- associativity $\quad (u \bullet v) \bullet w = u \bullet (v \bullet w)$
- neutral element $\quad u \bullet e = e \bullet u$

examples - $\mathbb{N} \quad + \, , \, 0 \quad$ addition

- $\mathbb{N} \quad \times \, , \, 1 \quad$ product

alphabet $\quad A$

free monoid $\quad A^*$

words $\quad w = a_1 a_2 \cdots a_p$

product : concatenation

$$\left.\begin{array}{l} u = a_1 \cdots a_p \\ v = b_1 \cdots b_q \end{array}\right\} uv = a_1 \cdots a_p b_1 \cdots b_q$$

empty word

- $a\,C\,b \Leftrightarrow b\,C\,a$
- $\cancel{a\,C\,a}$

commutation relation C antireflexive symmetric

$\equiv_C$ congruence of A^* generated by the commutations

$$ab \equiv ba \quad \text{iff} \quad a\,C\,b$$

commutation monoid $A^*/\equiv_C$

$[w]$ equivalence class of the word $w \in A^*$

$A^*/\equiv_C$ • product in the commutation monoid

$$[u]\cdot[v] = [uv]$$

independant of the choices of representants u and v

commutation monoid $A^*/\equiv_C$

free monoid A^*

word w

free abelian monoid $Ab(A)$

$x_1^{\alpha_1} \cdots x_k^{\alpha_k}$

commutation monoids

= free partially commutative monoids

Trace monoids

Computer Science

model for parallelism

concurrency access to data structures

Trace

Mazurkiewicz (1977)

model of the logical behavior of safe Petri nets

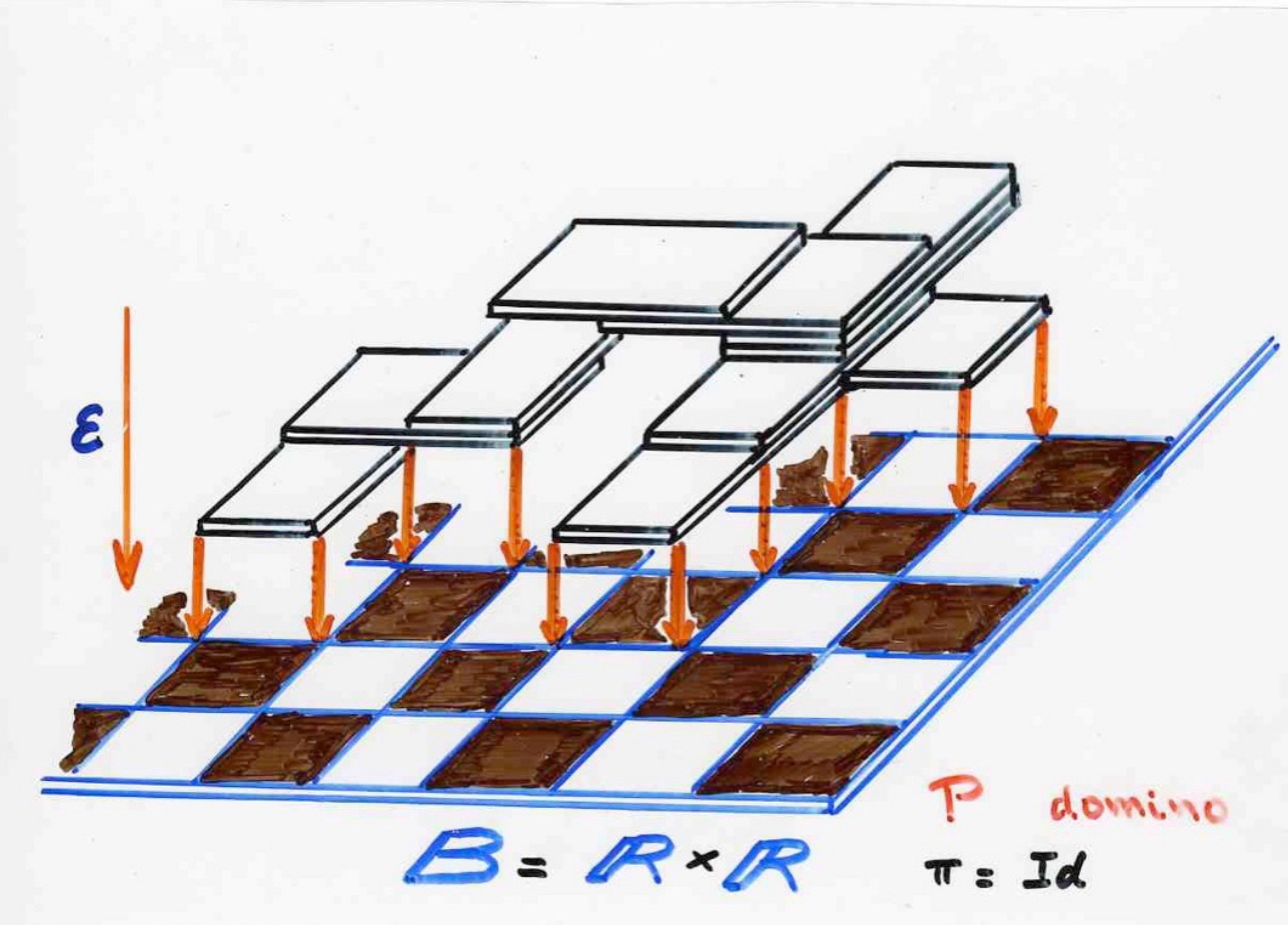
Diekert, Rosenberg ed. (1995)

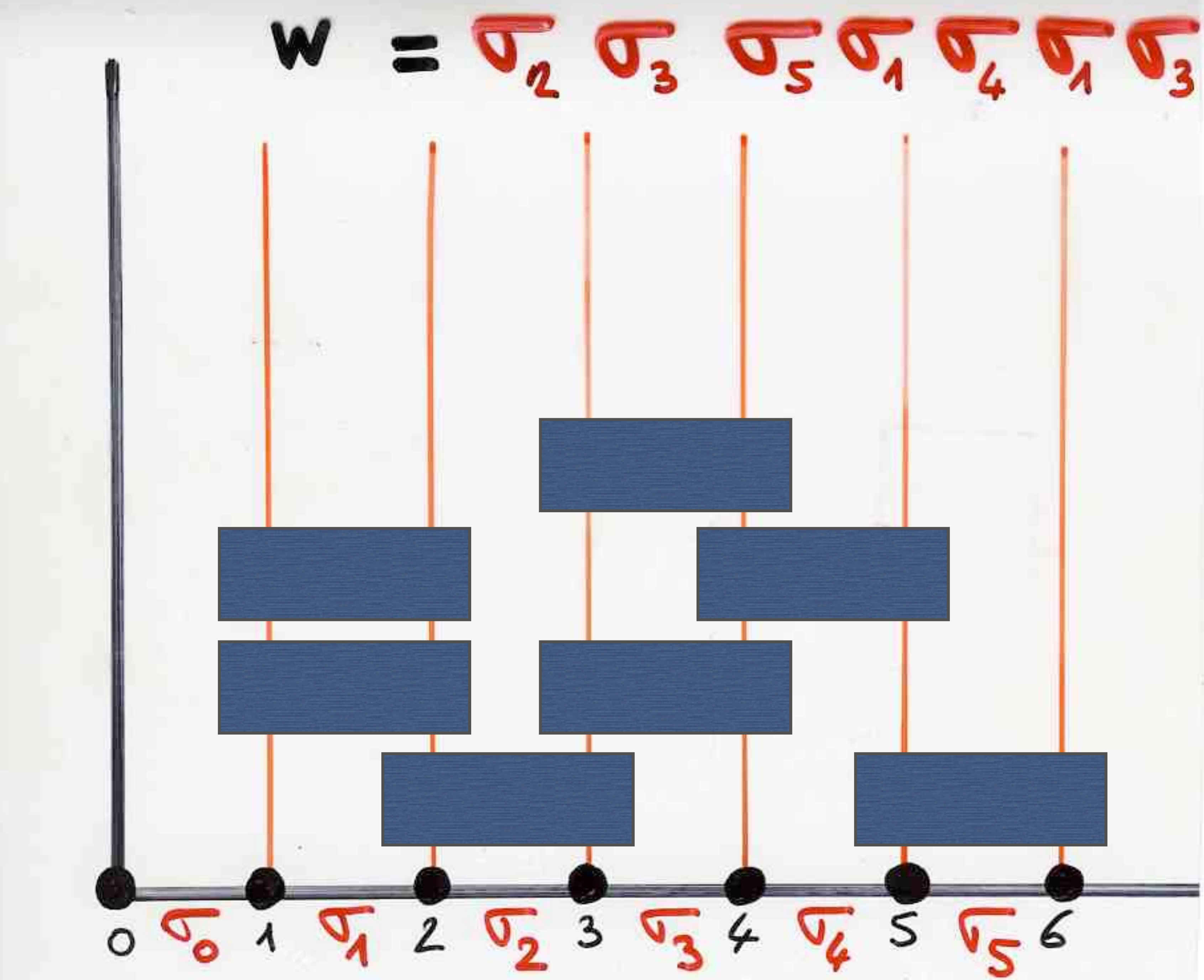
The book of traces

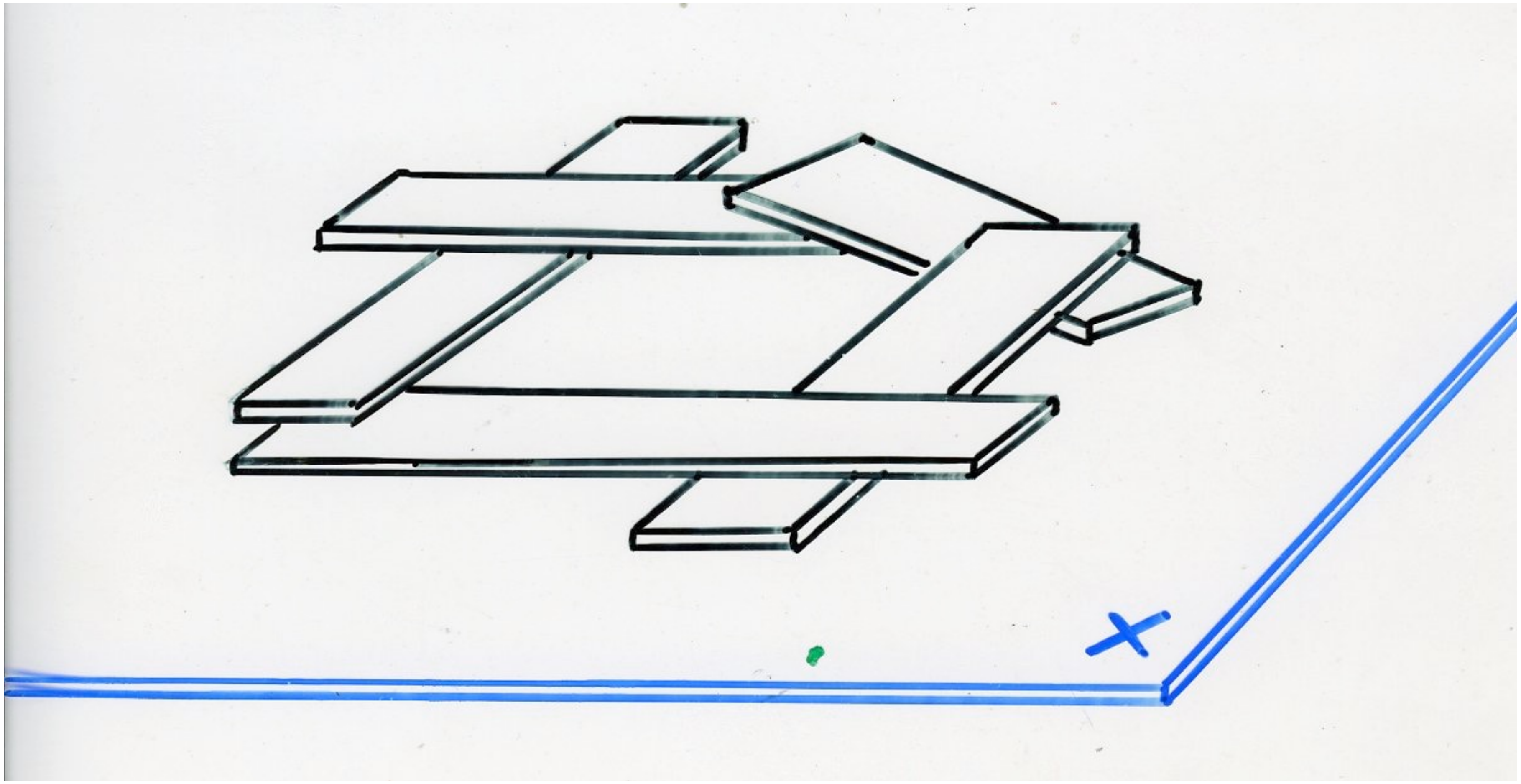
§2 Heaps of pieces
definition, examples

(X.V. 1985)

Introduction
Heaps




W = σ2 σ3 σ5 σ1 σ4 σ1 σ3
0 σ0 1 σ1 2 σ2 3 σ3 4 σ4 5 σ5 6



heap definition

- $\mathcal{P}$ set (of basic pieces)
- $\mathcal{C}$ binary relation on $\mathcal{P}$ $\begin{cases} \text{symmetric} \\ \text{reflexive} \end{cases}$

  (dependency relation)
- heap E, finite set of pairs

  (α, i) $\alpha \in \mathcal{P}$, $i \in \mathbb{N}$ (called pieces)

  projection level

(i)

(ii)

heap definition

- $\mathbf{P}$ set (of basic pieces)
- $\mathcal{C}$ binary relation on $\mathbf{P}$ $\begin{cases}\text{symmetric}\\ \text{reflexive}\end{cases}$

  (dependency relation)
- heap E, finite set of pairs

  (α, i) $\alpha \in P$, $i \in \mathbb{N}$ (called pieces)

  projection level

(i) $(\alpha, i), (\beta, j) \in E$, $\alpha \, \mathcal{C} \, \beta \implies i \neq j$

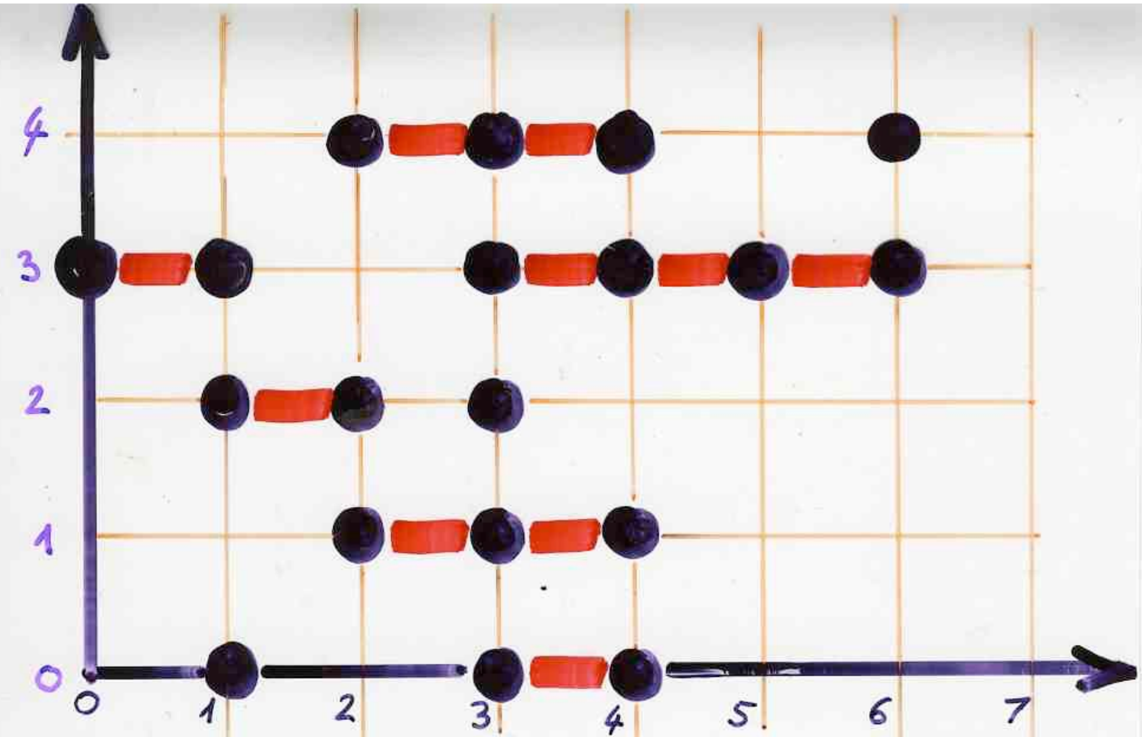
(ii) $(\alpha, i) \in E$, $i > 0 \implies \exists \beta \in P$, $\alpha \, \mathcal{C} \, \beta$,

$(\beta, i-1) \in E$

ex: heap of segments over $\mathbb{N}$

$P = \{ [a,b] = \{a, a+1, \dots, b\} \;, 0 \leq a \leq b \}$

$\mathcal{C} \qquad [a,b] \,\mathcal{C}\, [c,d] \iff [a,b] \cap [c,d] \neq \emptyset$

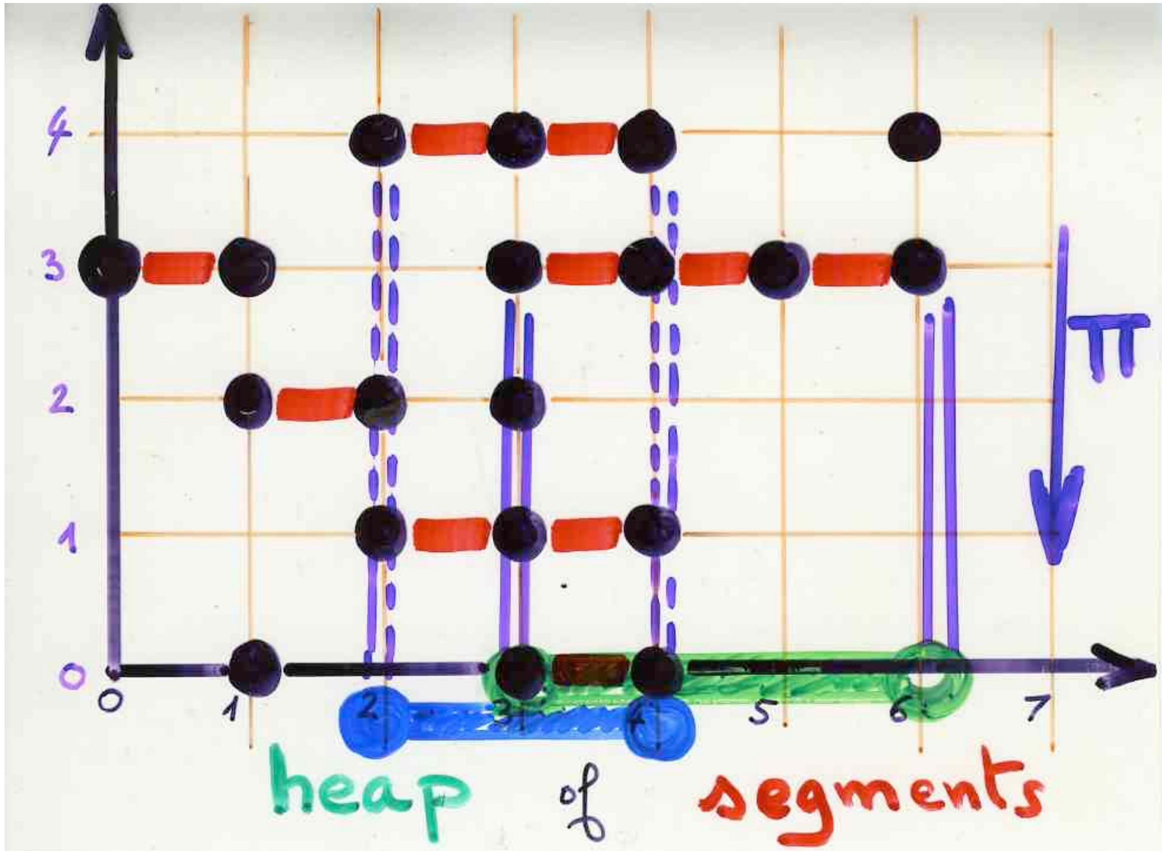


heap of segments

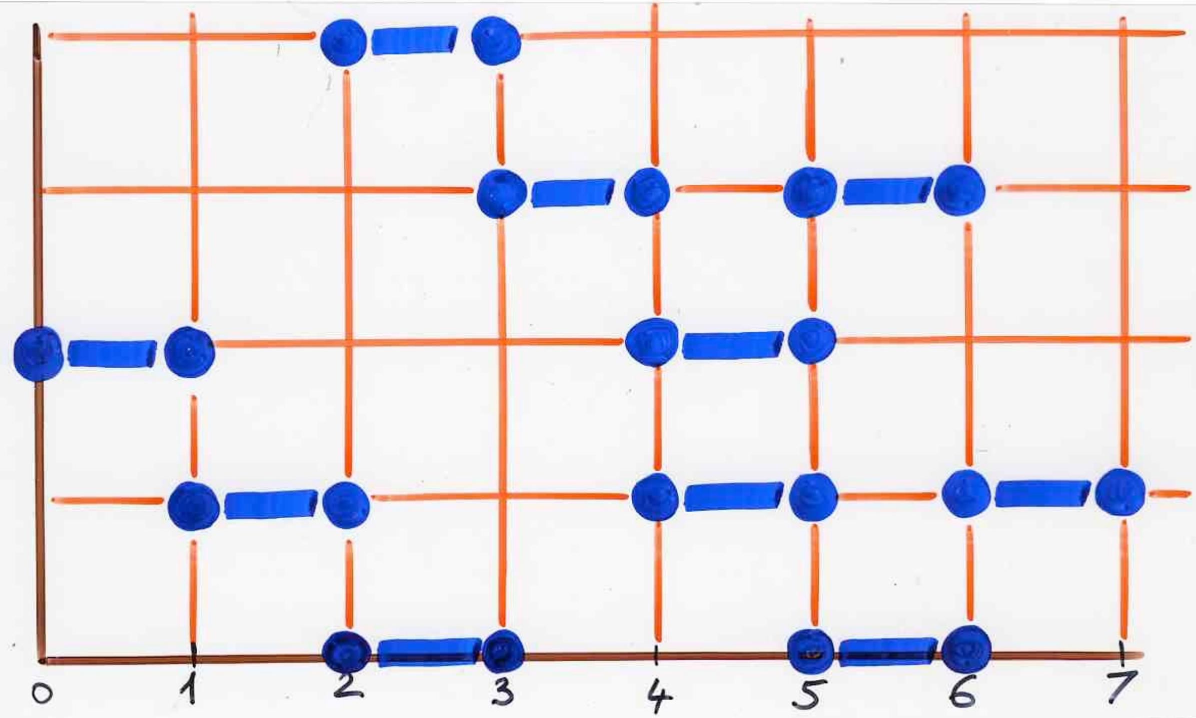
ex: heap of segments over $\mathbb{N}$

$$P = \{ [a,b] = \{a, a+1, \dots, b\} \ , 0 \le a \le b \}$$

$$\mathcal{C} \qquad [a,b] \ \mathcal{C} \ [c,d] \iff [a,b] \cap [c,d] \neq \emptyset$$



Heap of dimers over [1, n]



ex: non-empty subsets of a set X

- $\mathcal{P}$ set of non-empty subsets of X

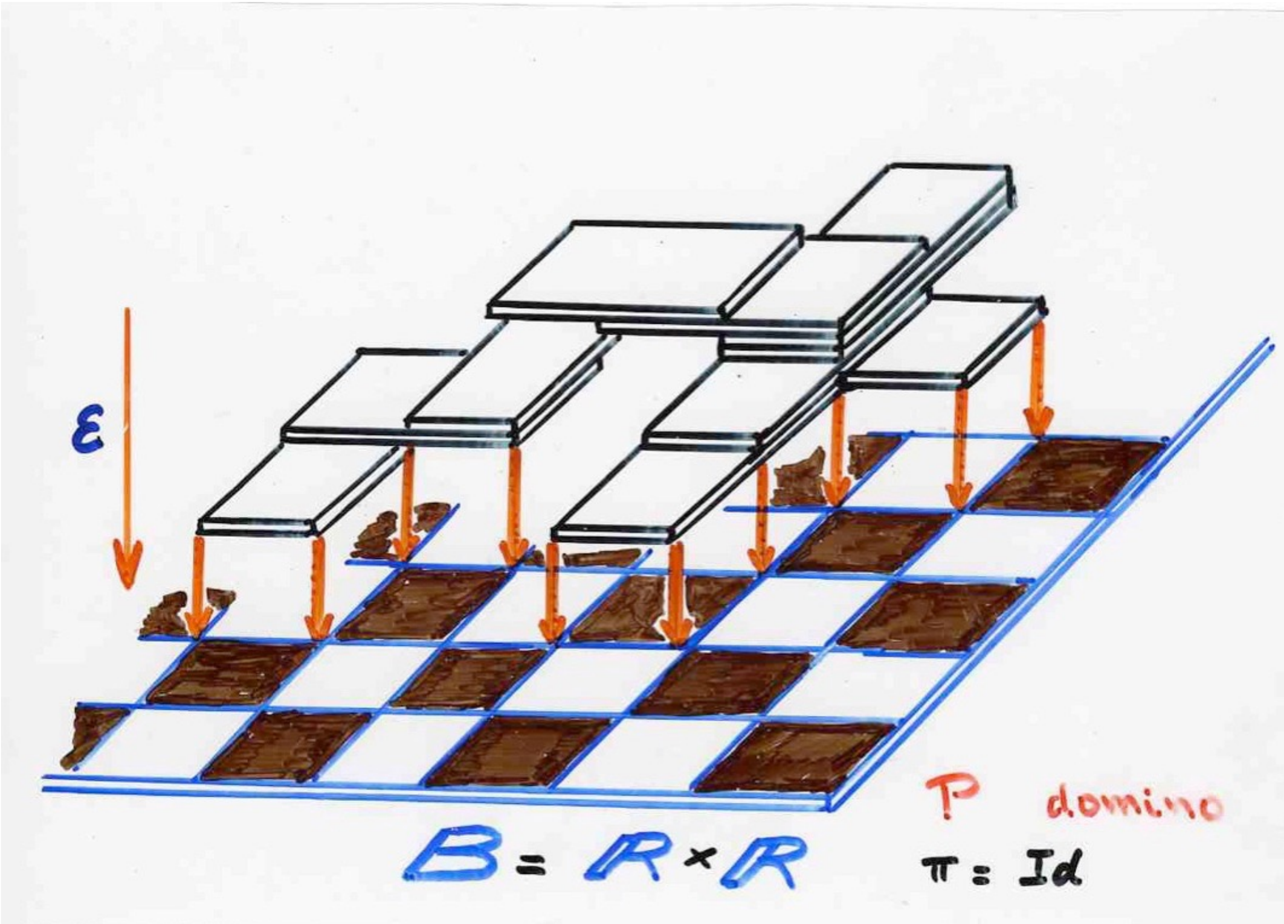
  basic pieces

  $$\mathcal{P} \subseteq \mathcal{P}(X)$$

- $\mathcal{C}$ dependency relation

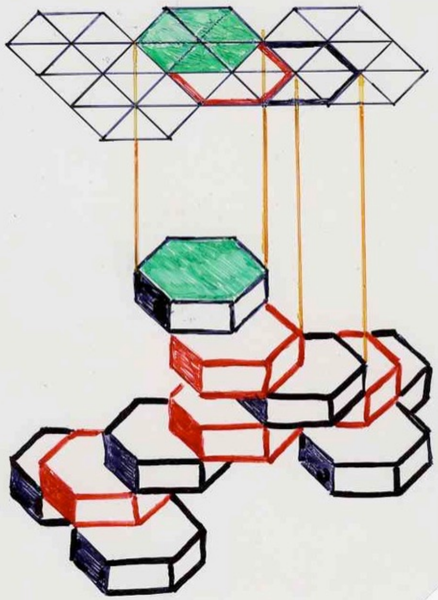
  $$A, B \in \mathcal{P}, \quad A \,\mathcal{C}\, B \Leftrightarrow A \cap B \neq \emptyset$$

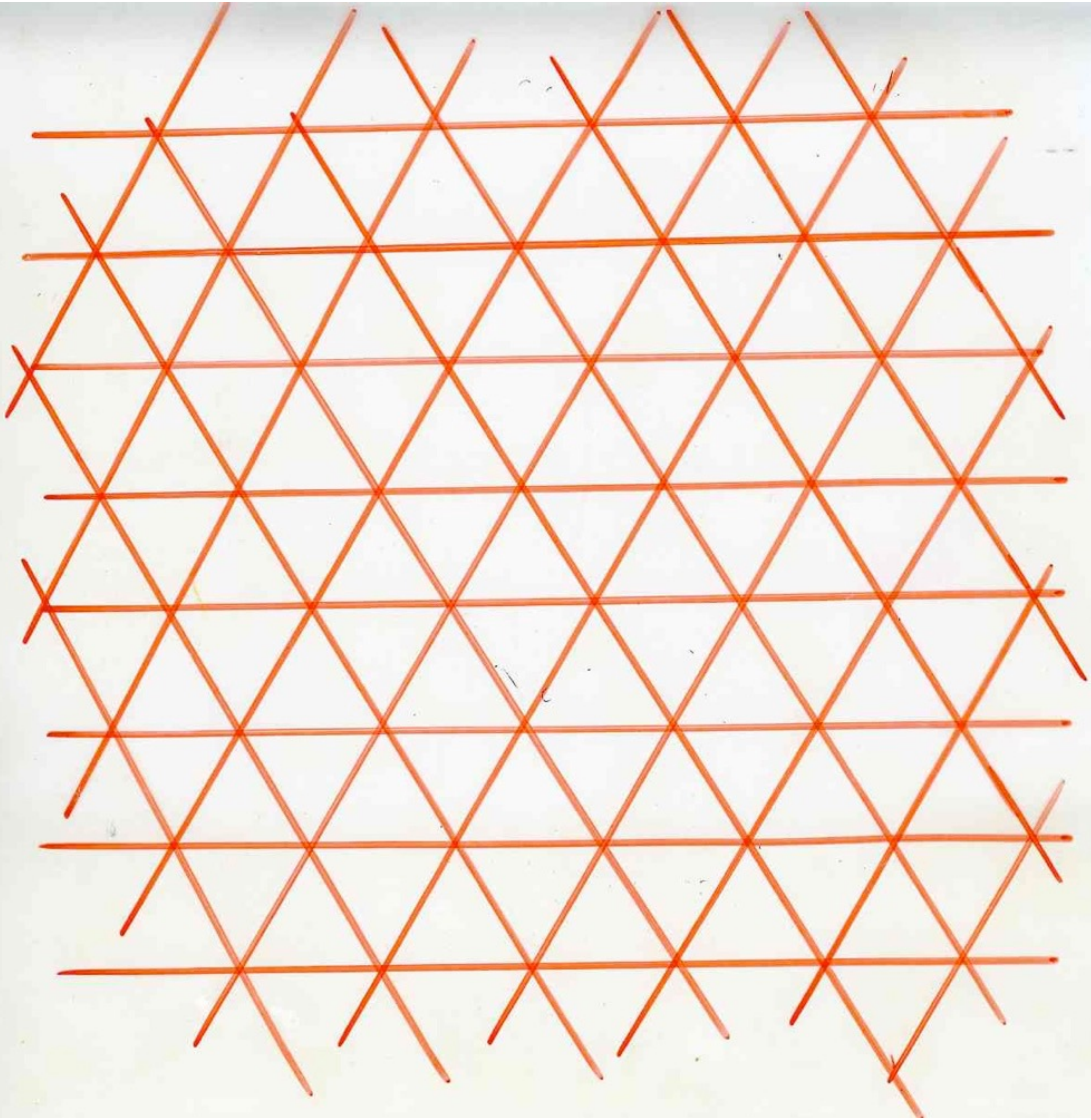
Heaps of "hard dimers"
on a chessboard

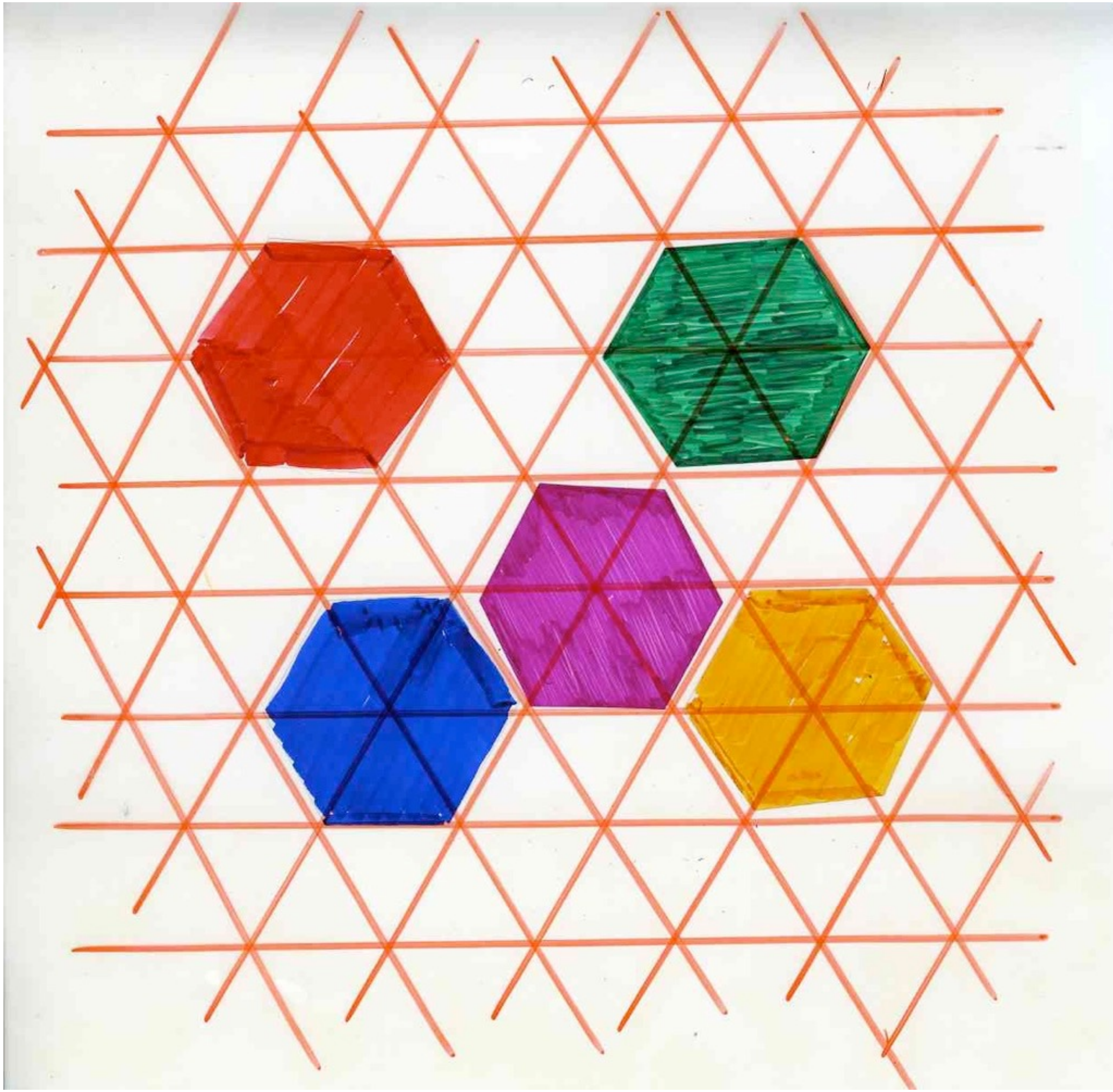


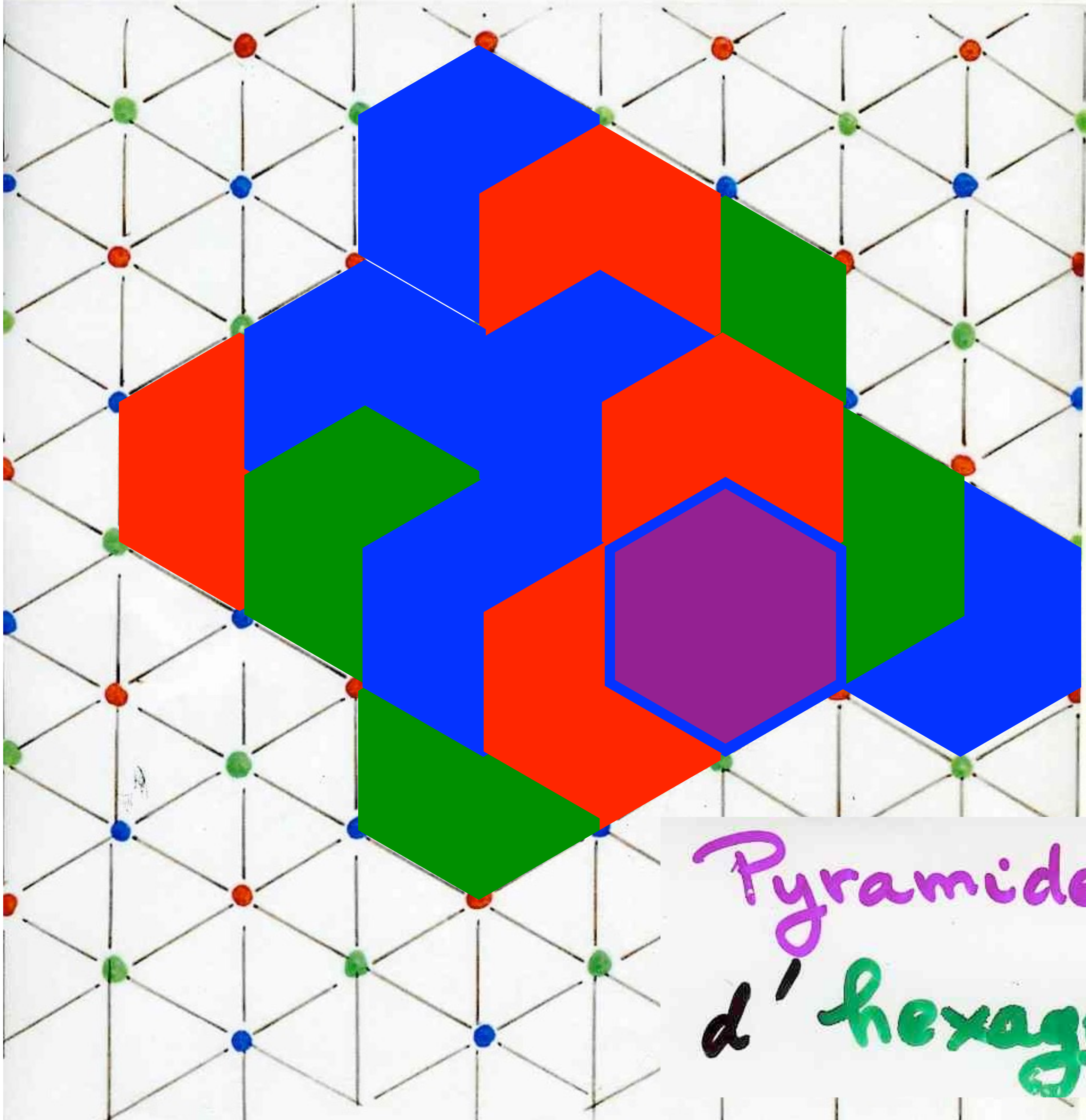
Heaps of "hard hexagons"

$$-\rho(-t) = y$$



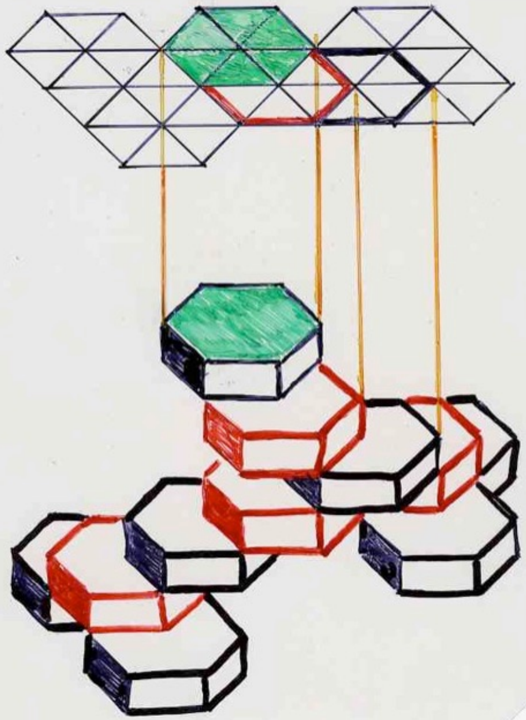




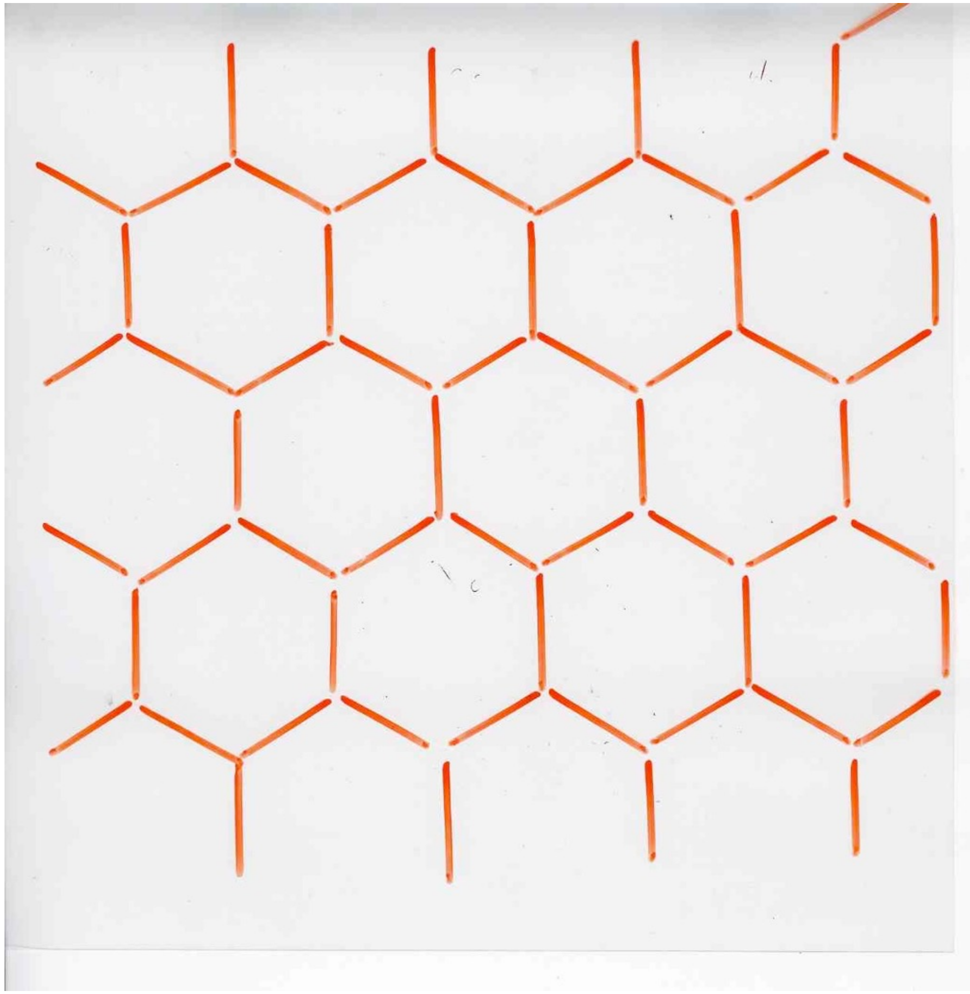


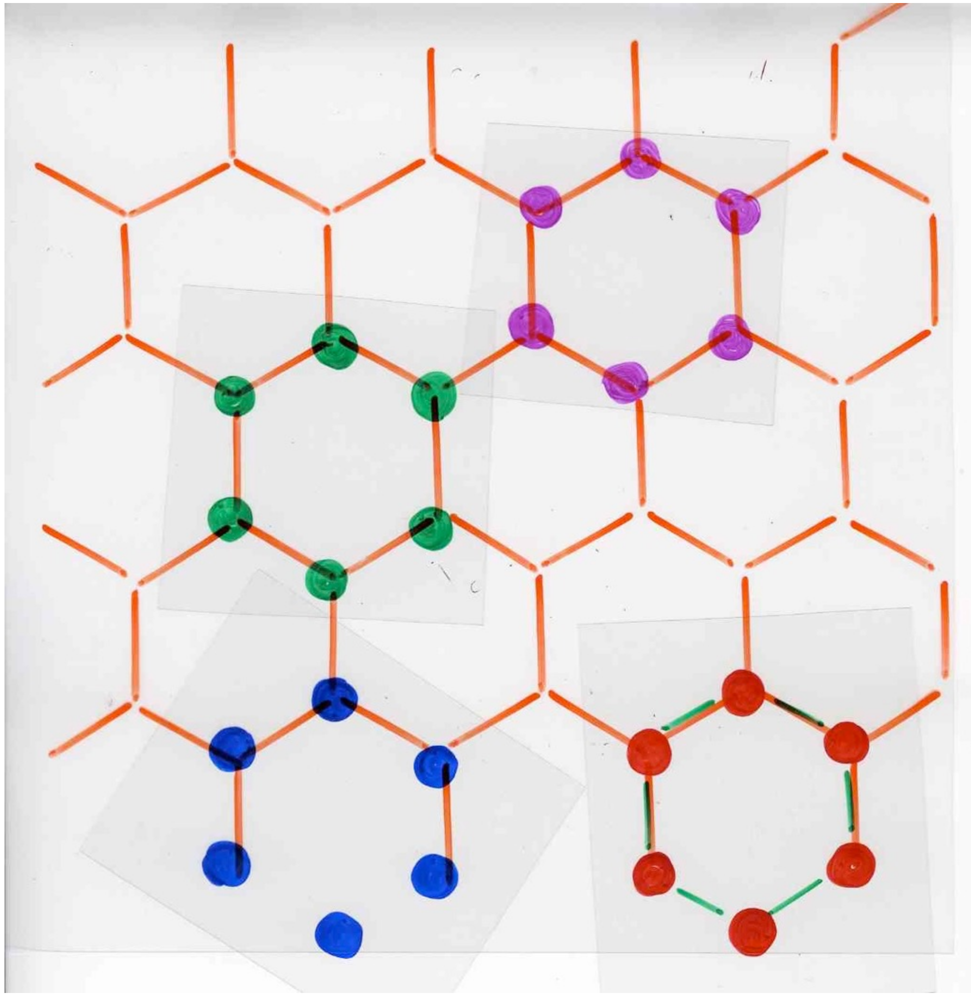
Pyramide d'hexagones

$$-\rho(-t) = y$$

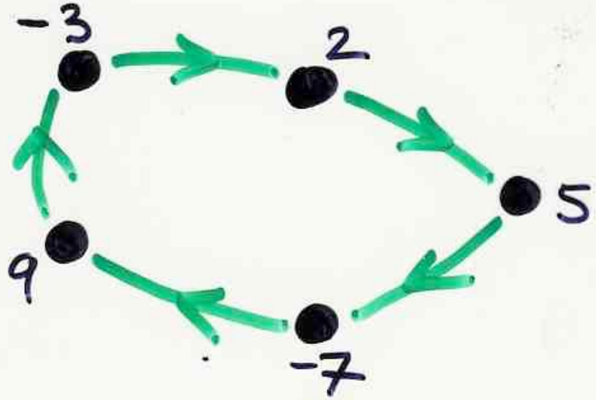


HC.





basic pieces $P = \{$ cycles on $\mathbb{Z} \}$

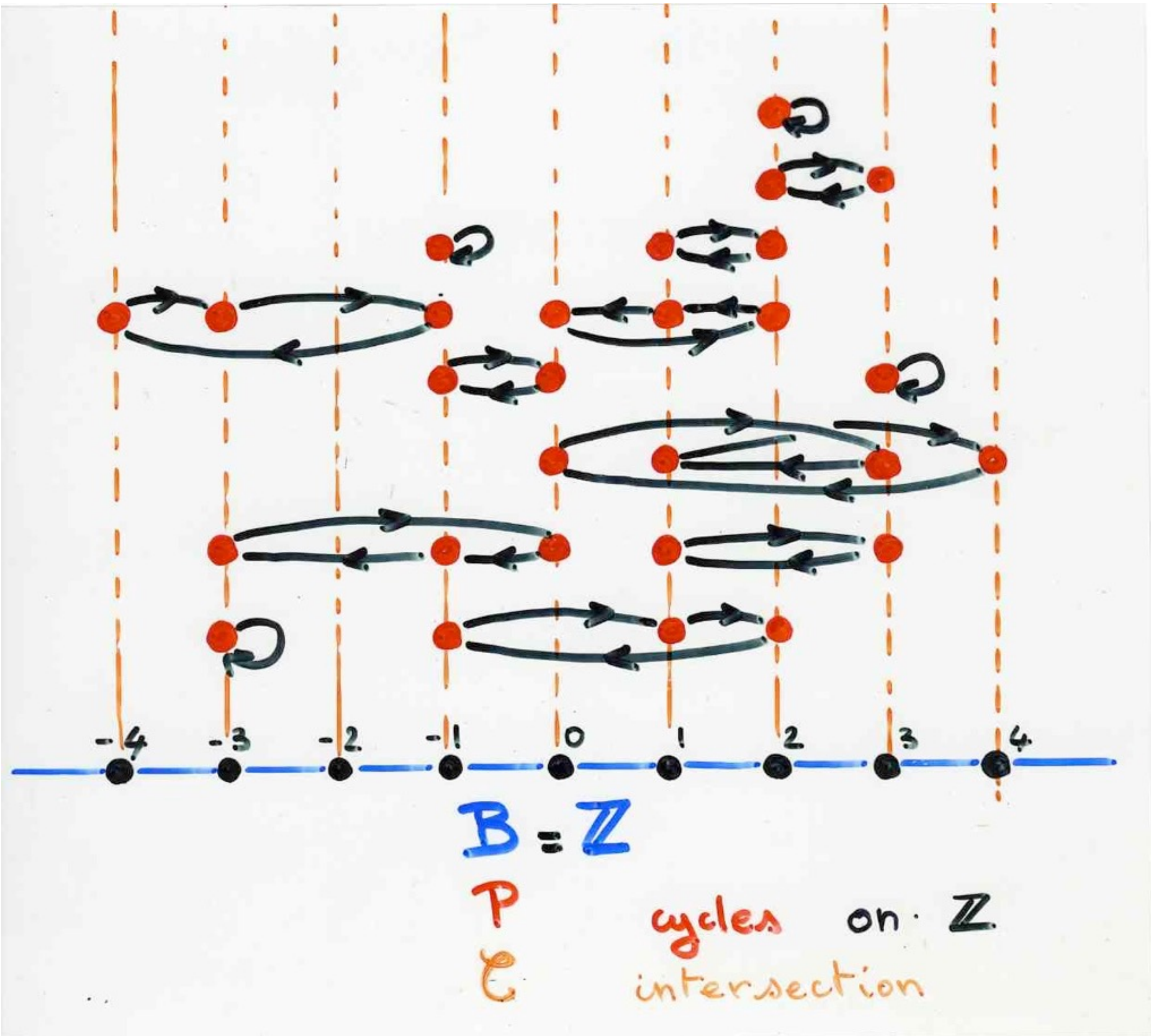
$\gamma =$ 

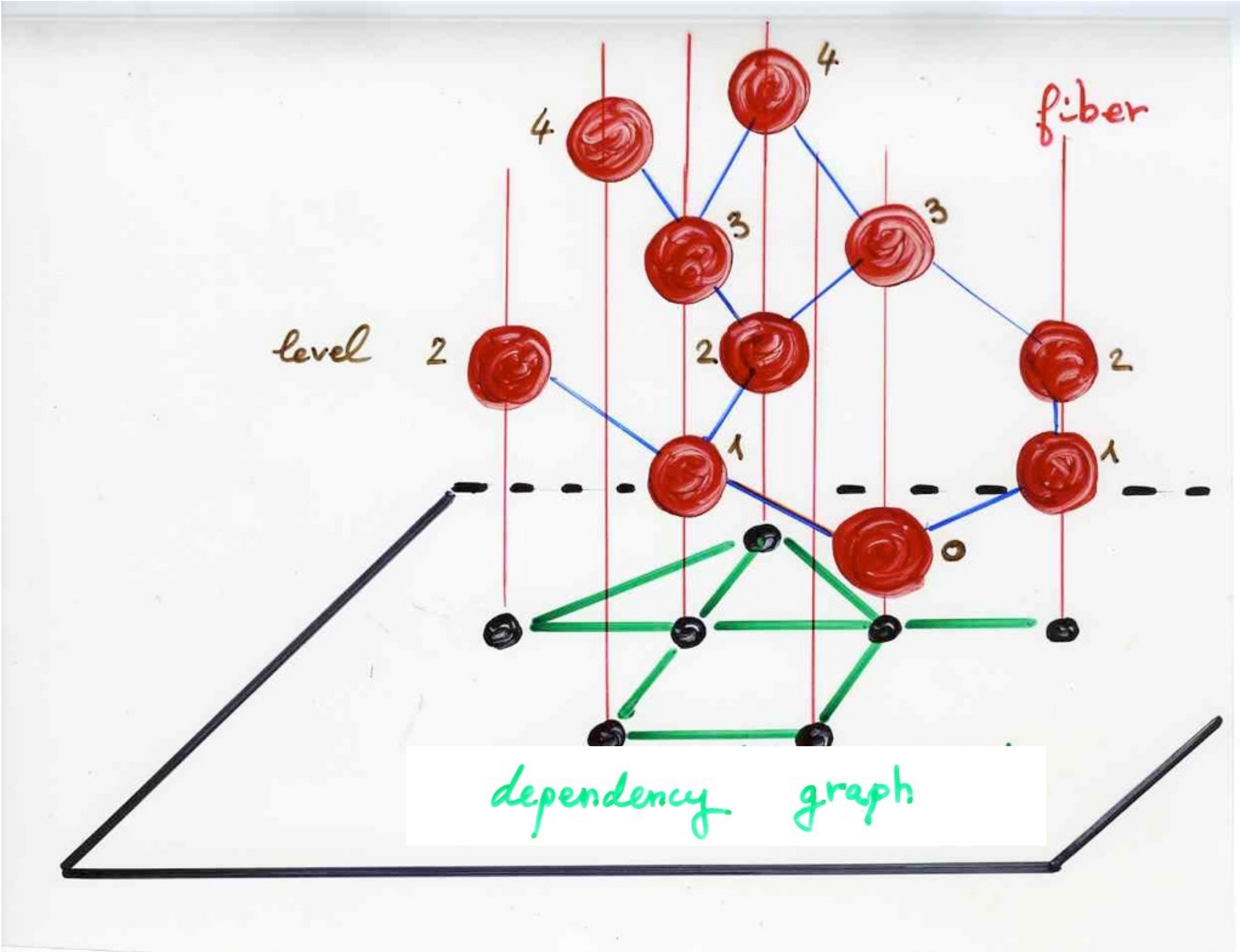
$\mathrm{Supp}(\gamma)$
$= \{-7, -3, 2, 5, 9\}$

Support

$\mathcal{C}$ dependency relation

$\gamma \,\mathcal{C}\, \delta \iff \mathrm{Supp}(\gamma) \cap \mathrm{Supp}(\delta) \neq \emptyset$


-4
-3
-2
-1
0
1
2
3
4
B = ℤ
P cycles on ℤ
C intersection



4
4
fiber
3
3
level
2
2
2
1
1
0
dependency graph

§3 Heaps monoids

Déf. pre-heap E

P $\quad$ $\mathcal{C}$

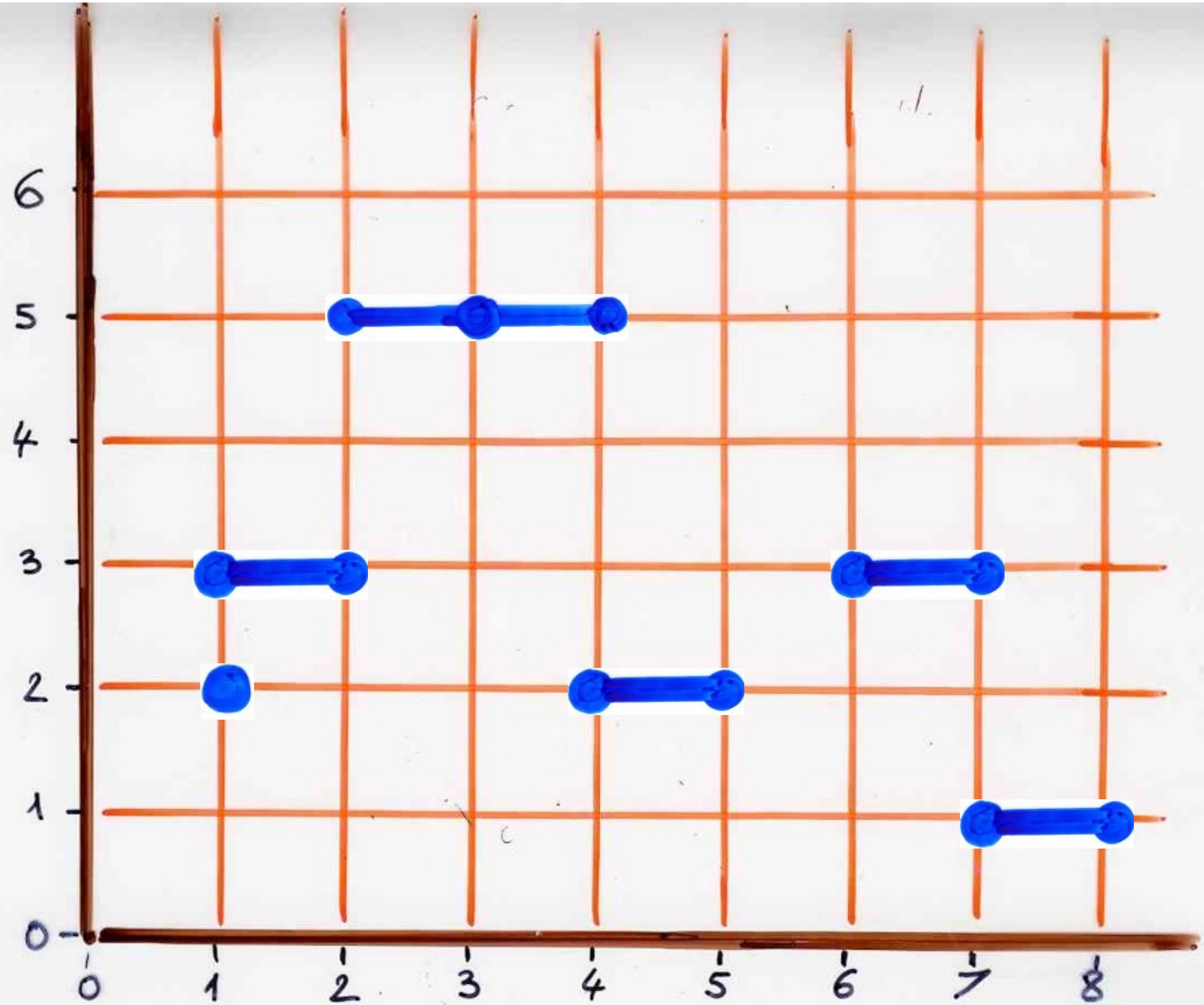
E

(α, i) $\quad$ $\alpha \in P$

$i \in \mathbb{N}$

(i) $(\alpha, i), (\beta, j) \in E$

$\alpha \,\mathcal{C}\, \beta \Rightarrow i \neq j$

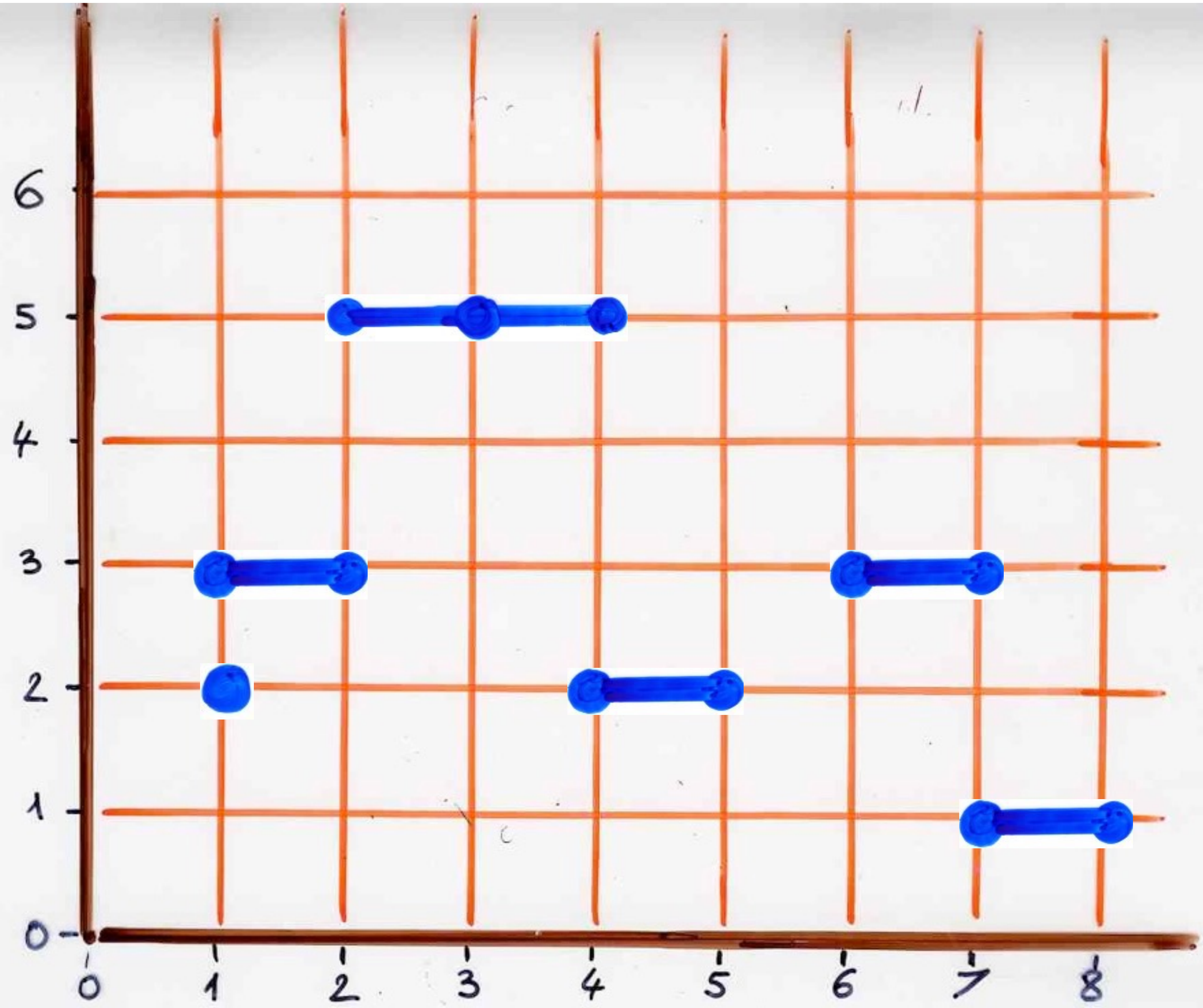


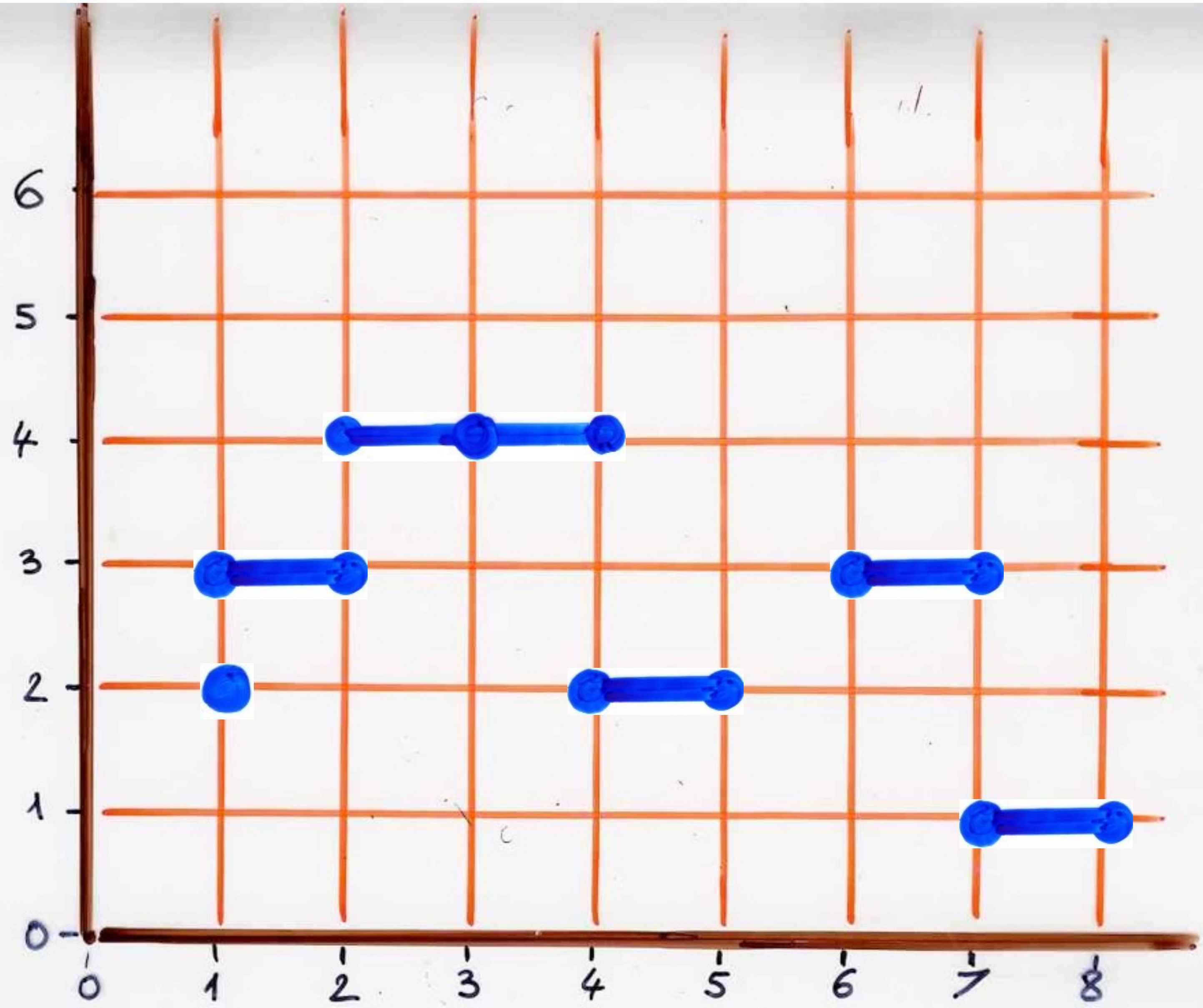
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8

Def. <u>elementary move</u>

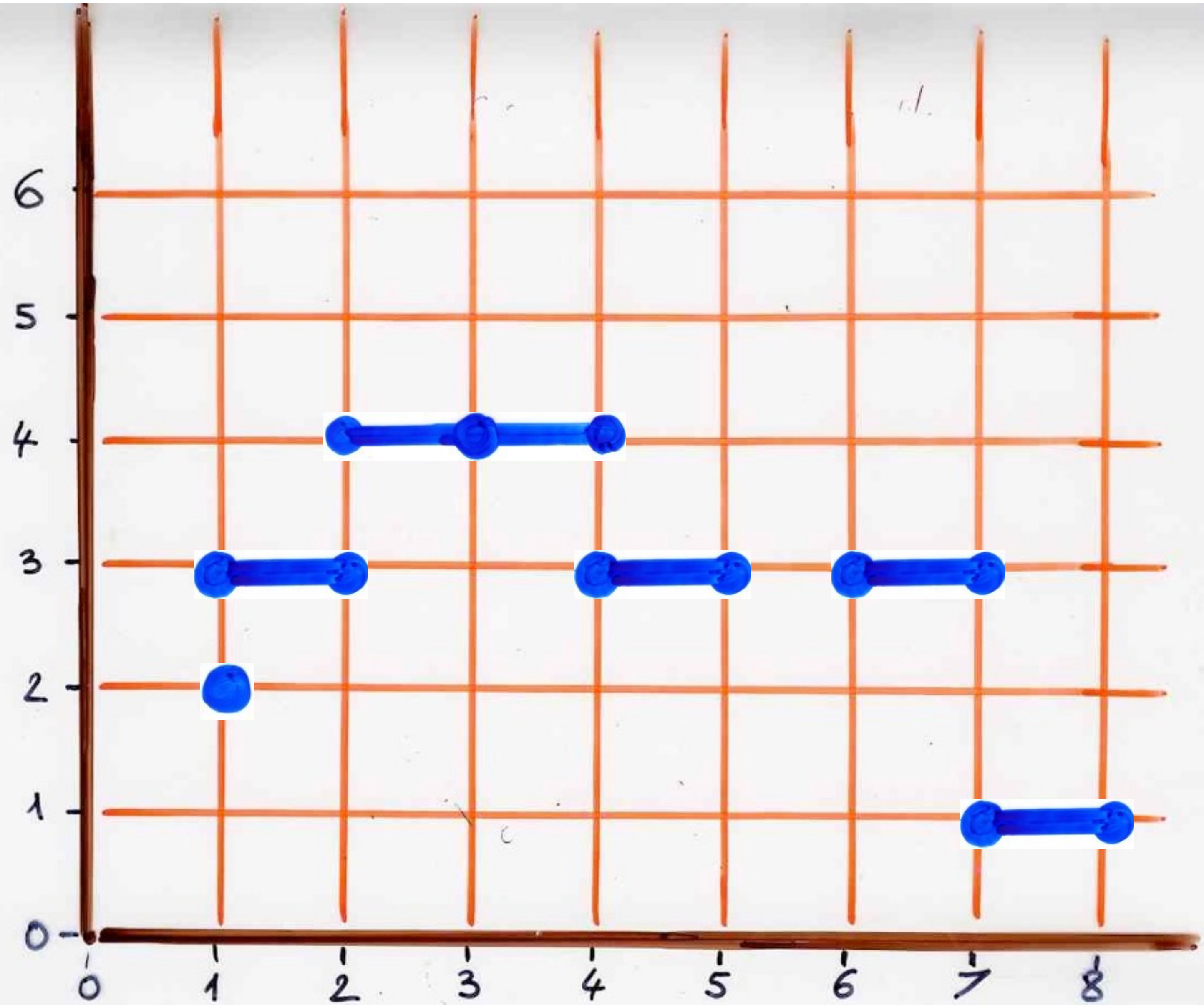
on a pre-heap E

$$(\alpha, i) \longrightarrow (\alpha, i-1) \text{ or } (\alpha, i+1) \qquad \text{(if possible)}$$


6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8



6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8



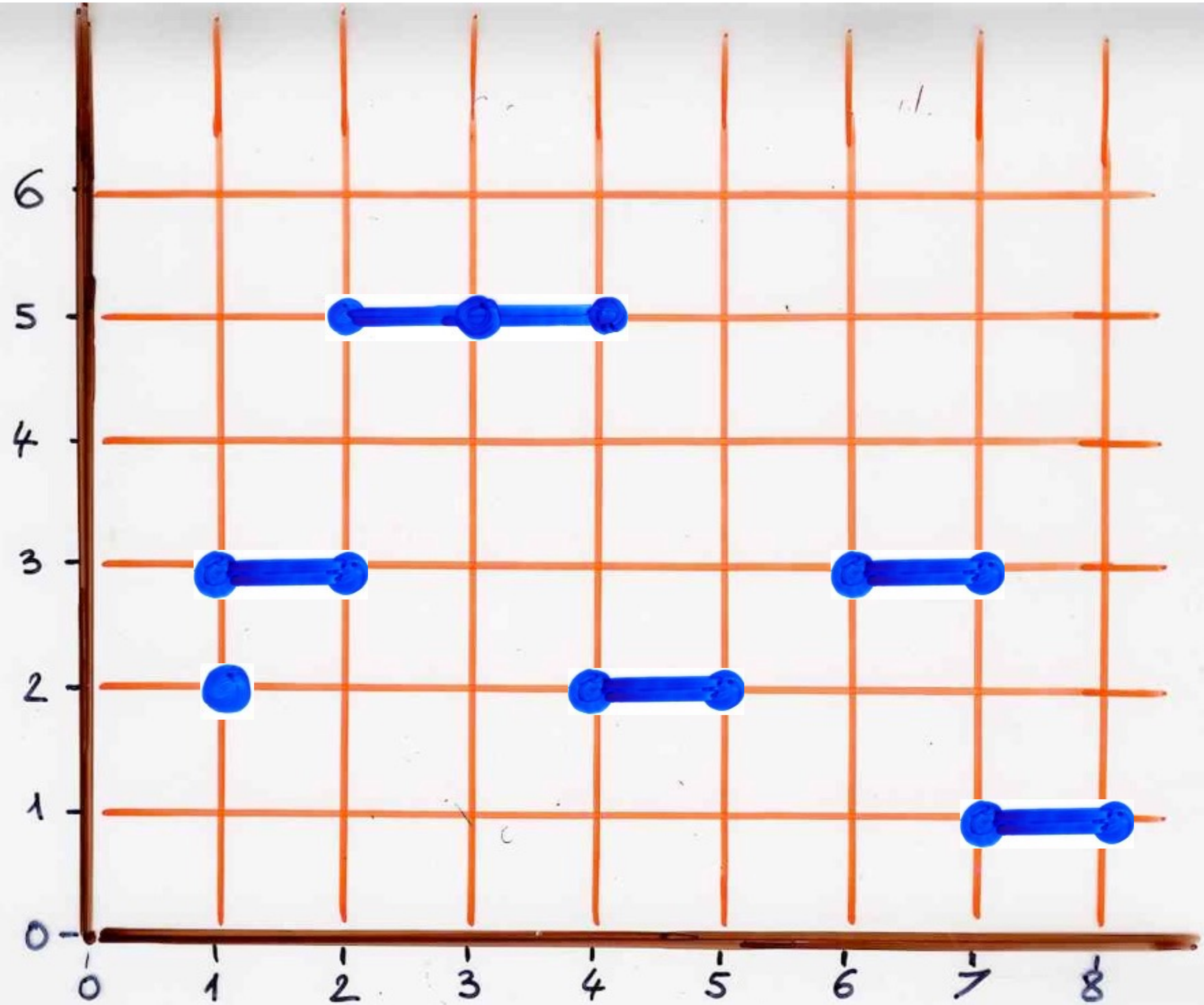
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8

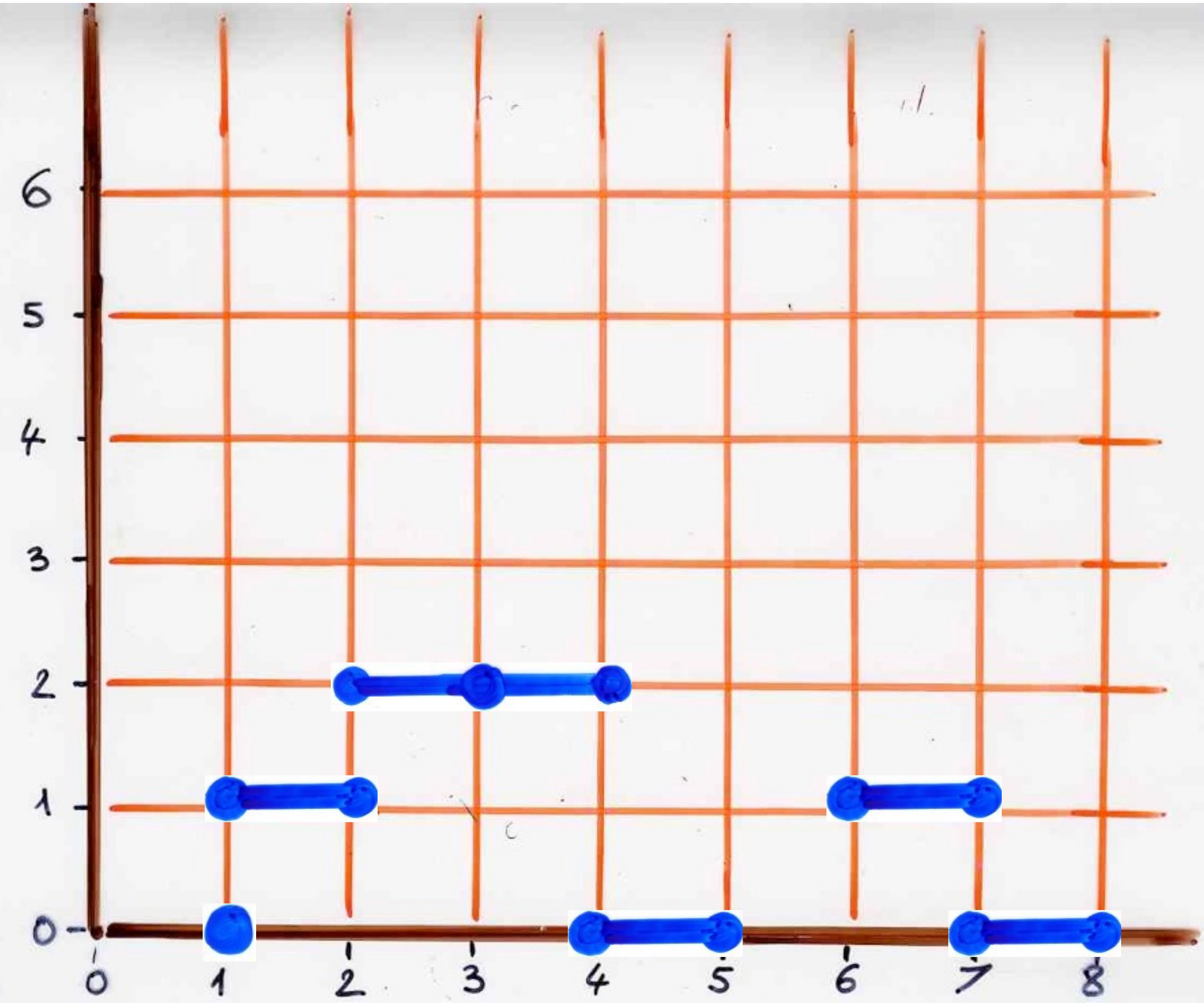
heap :

pre-heap up to the equivalence $\sim$

- in each equivalence class for $\sim$ there exist a unique **heap**

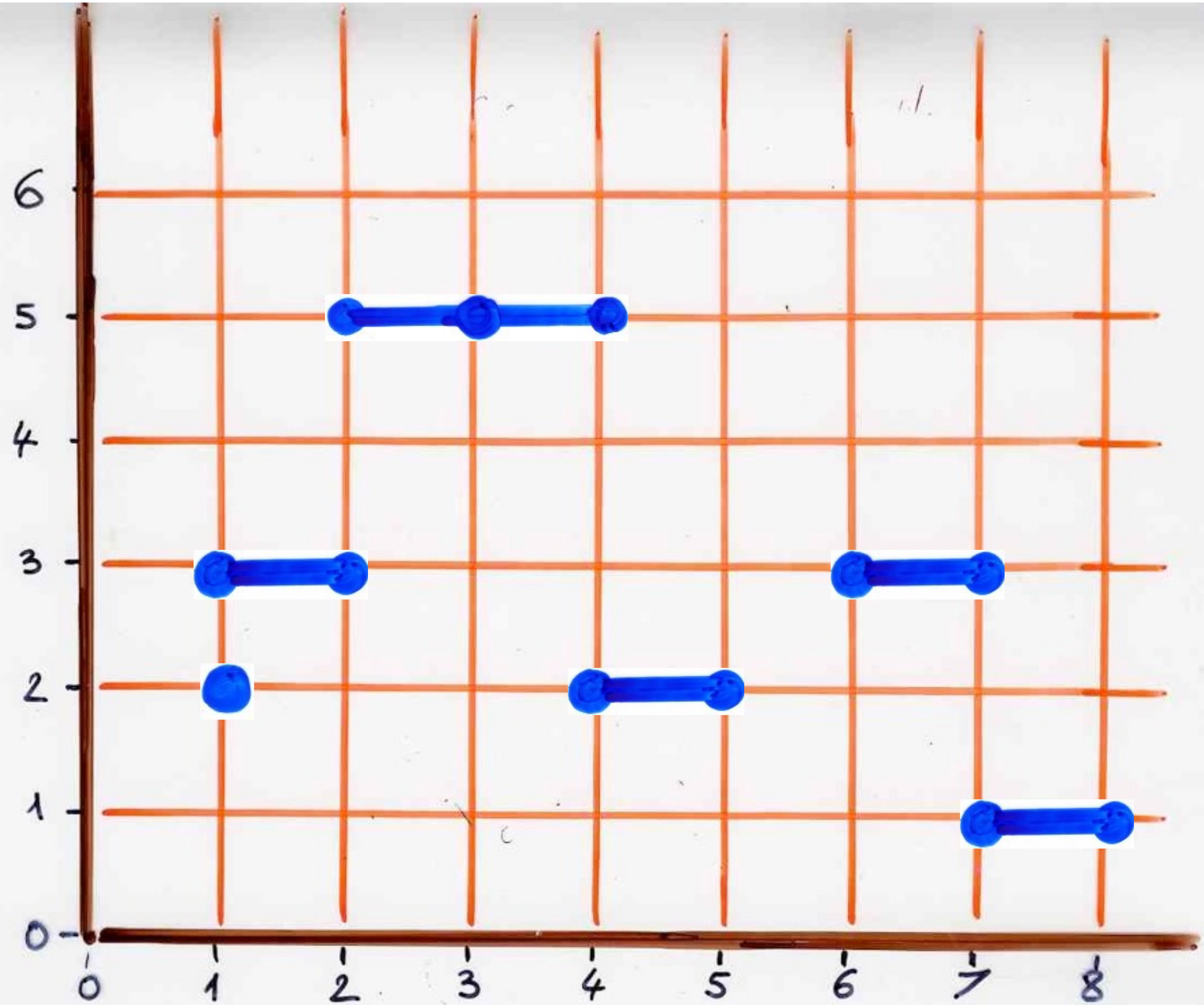
PE pre-heap $\longrightarrow$ $\underset{\text{heap}}{E_{\sim}(PE)}$

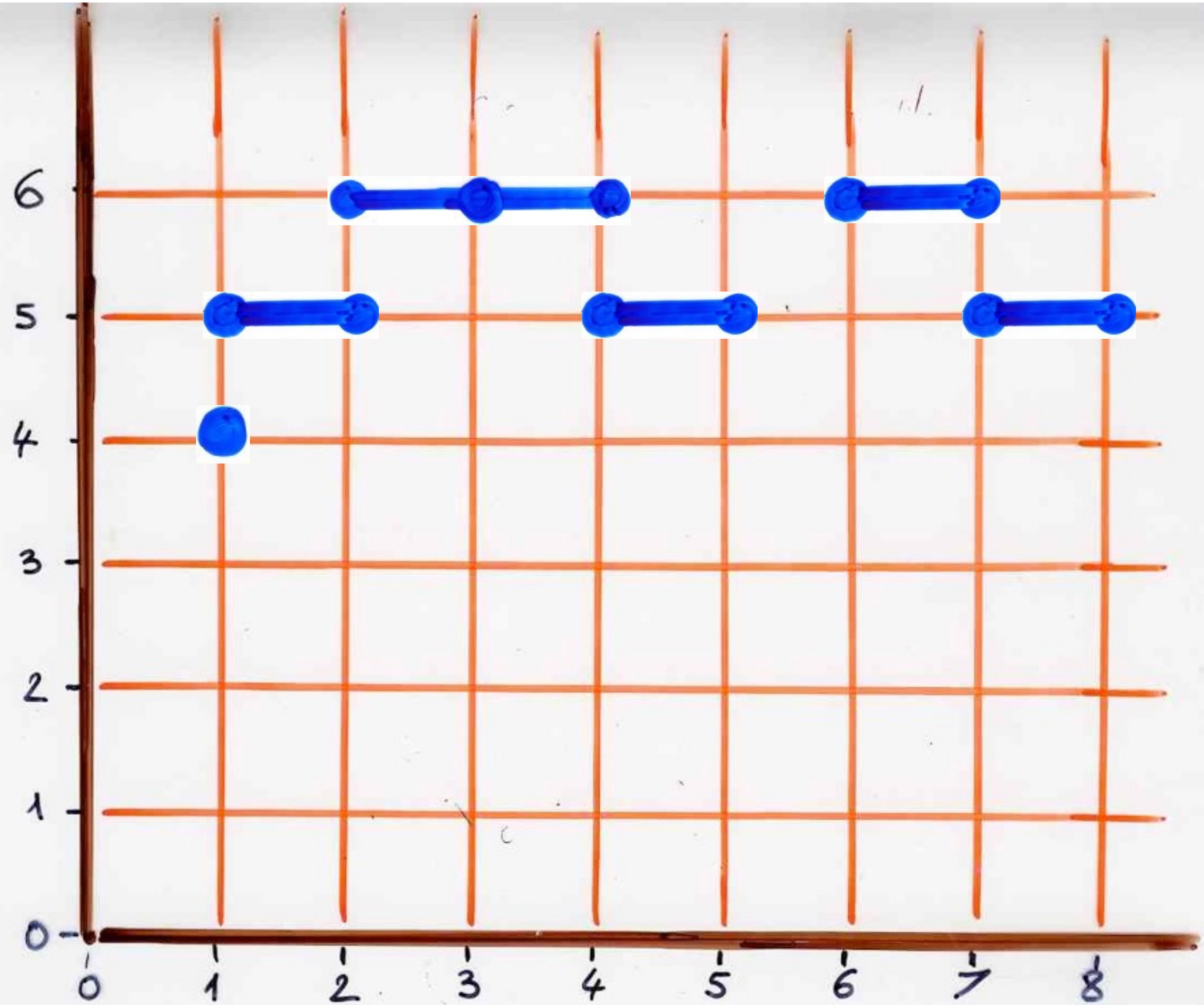

6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8



6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8

- heap → "anti-heap" (helium)


0
1
2
3
4
5
6
7
8

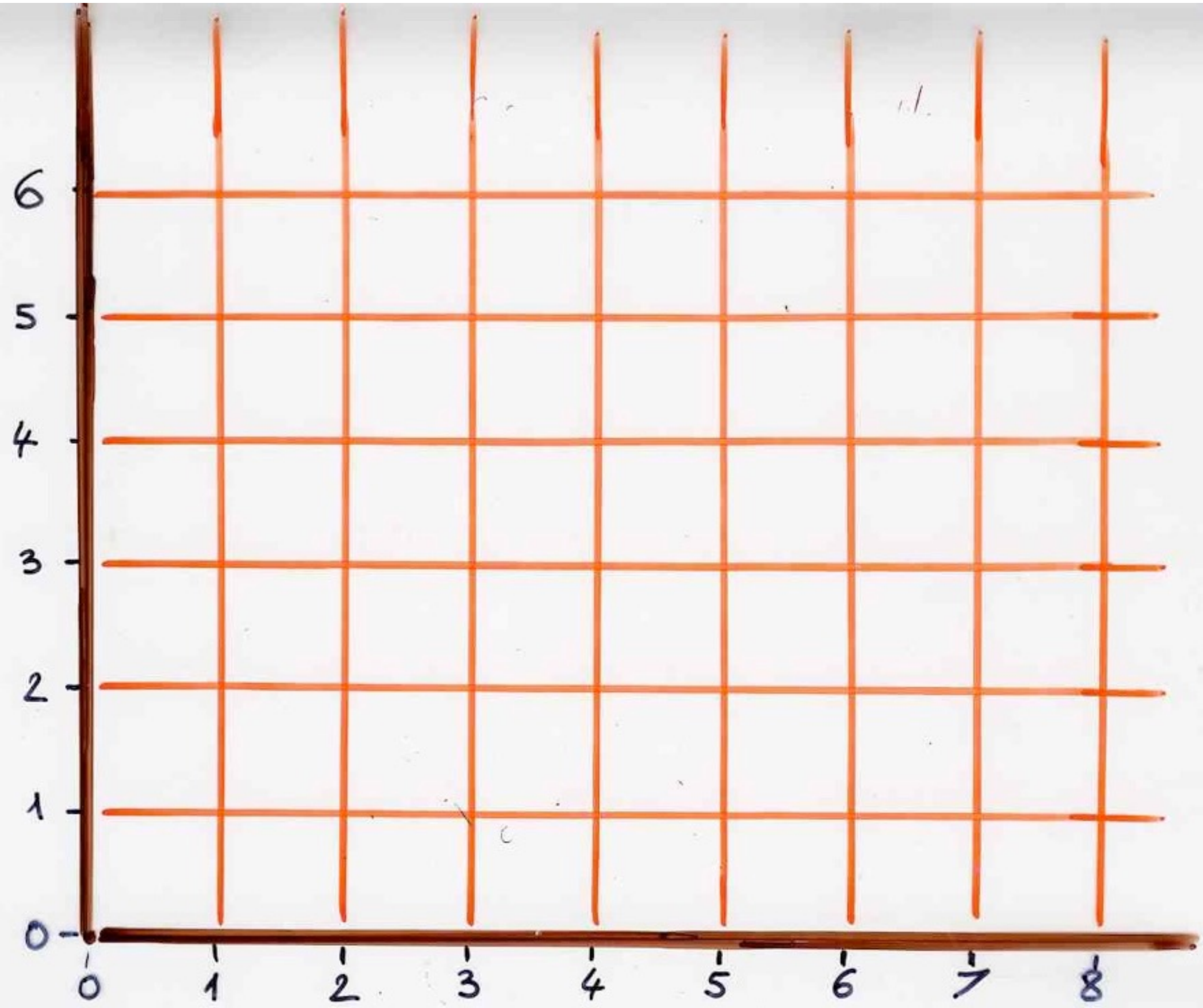

0
1
2
3
4
5
6
7
8
0
1
2
3
4
5
6

Heaps monoid

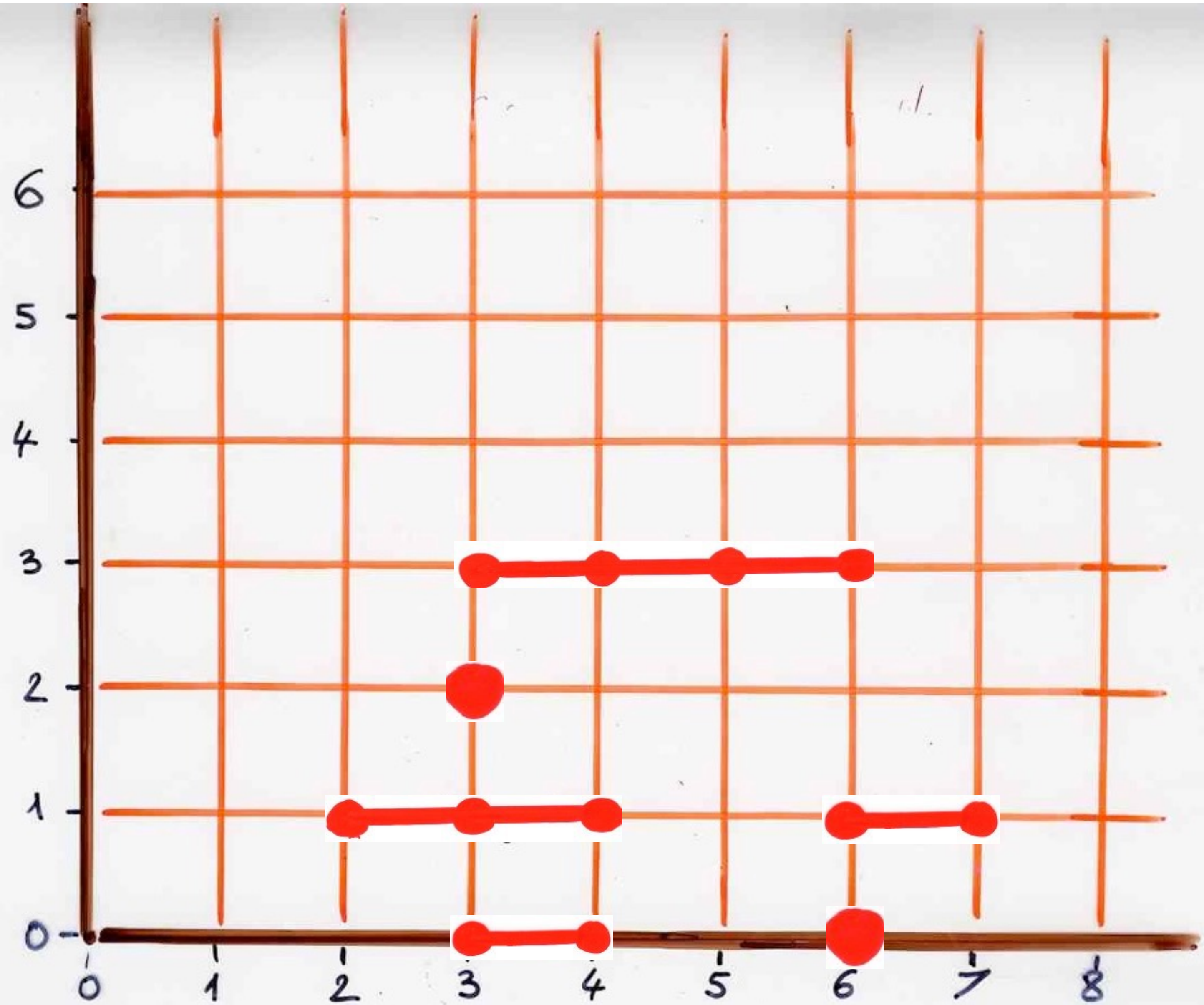
$H(P, \mathcal{C})$

product of two heaps

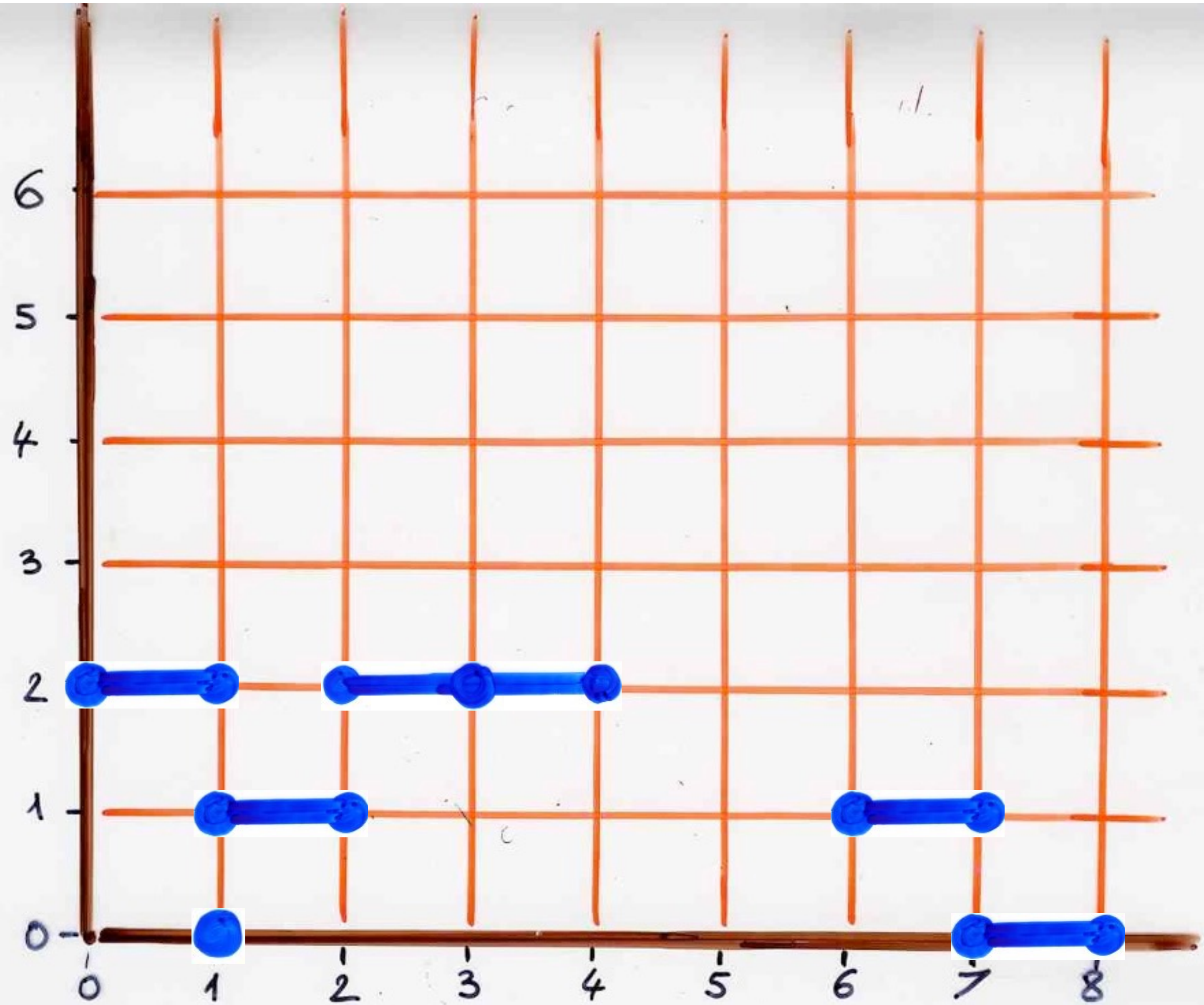
$E \circ F$



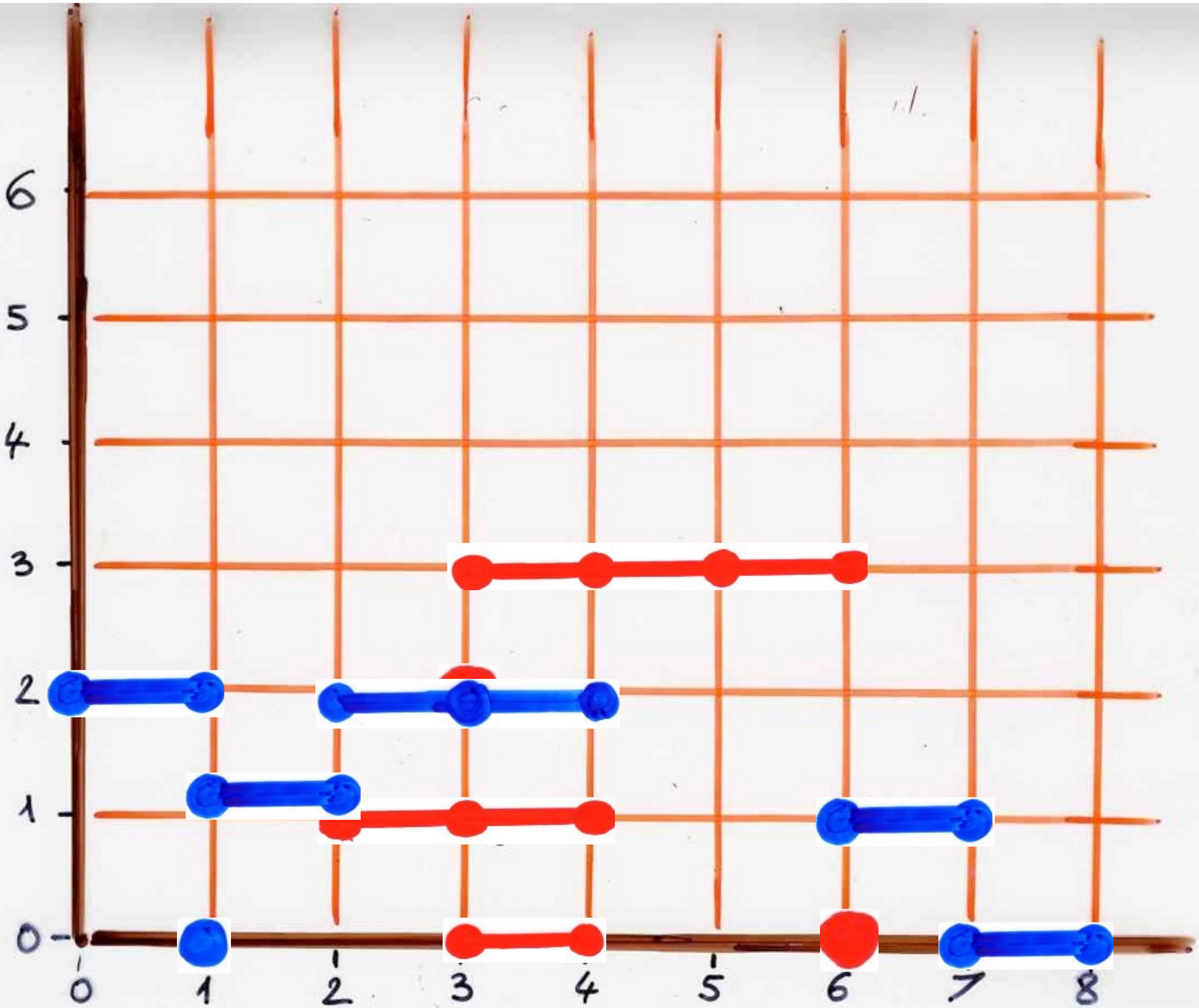
0
1
2
3
4
5
6
7
8

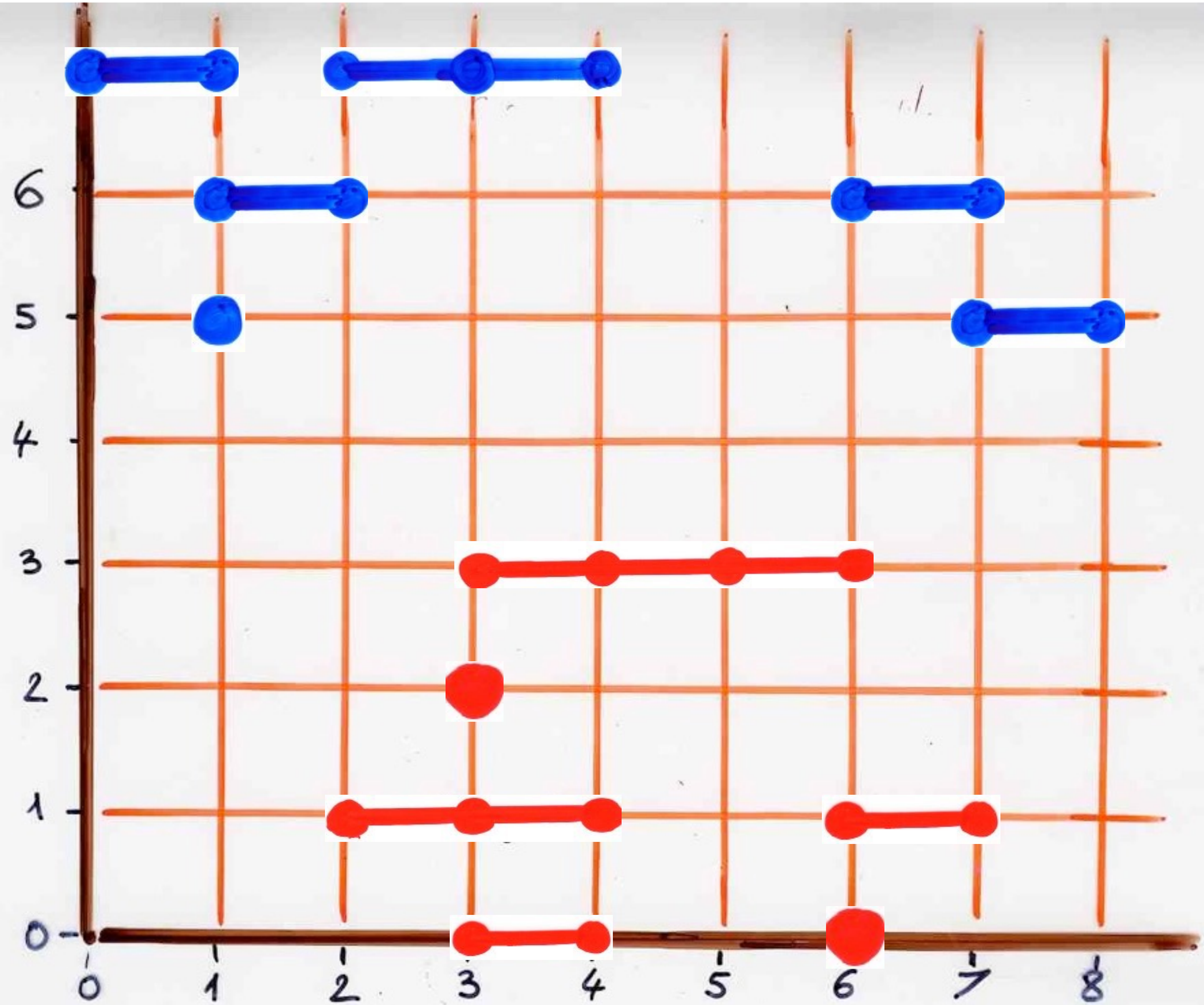


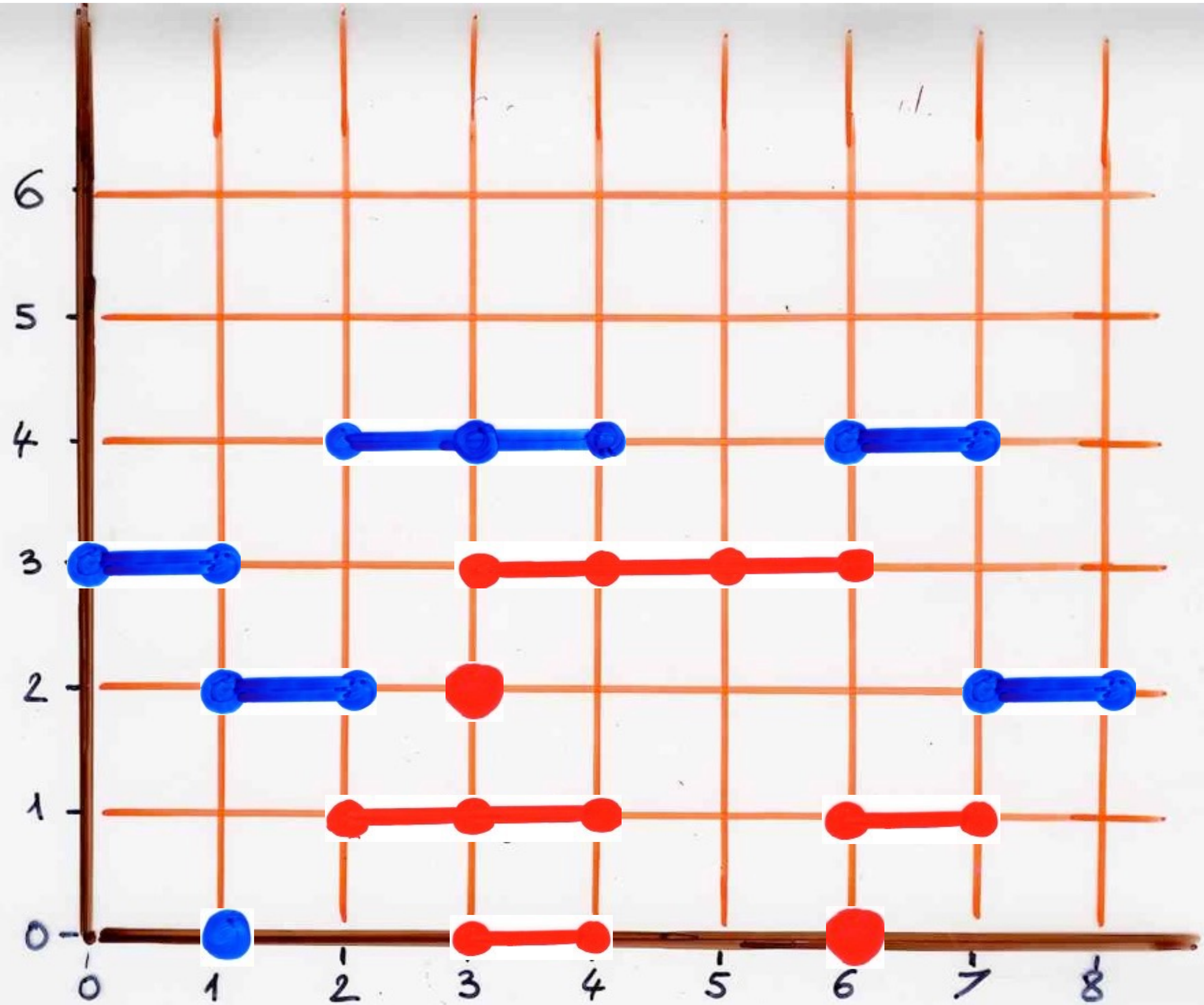
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8



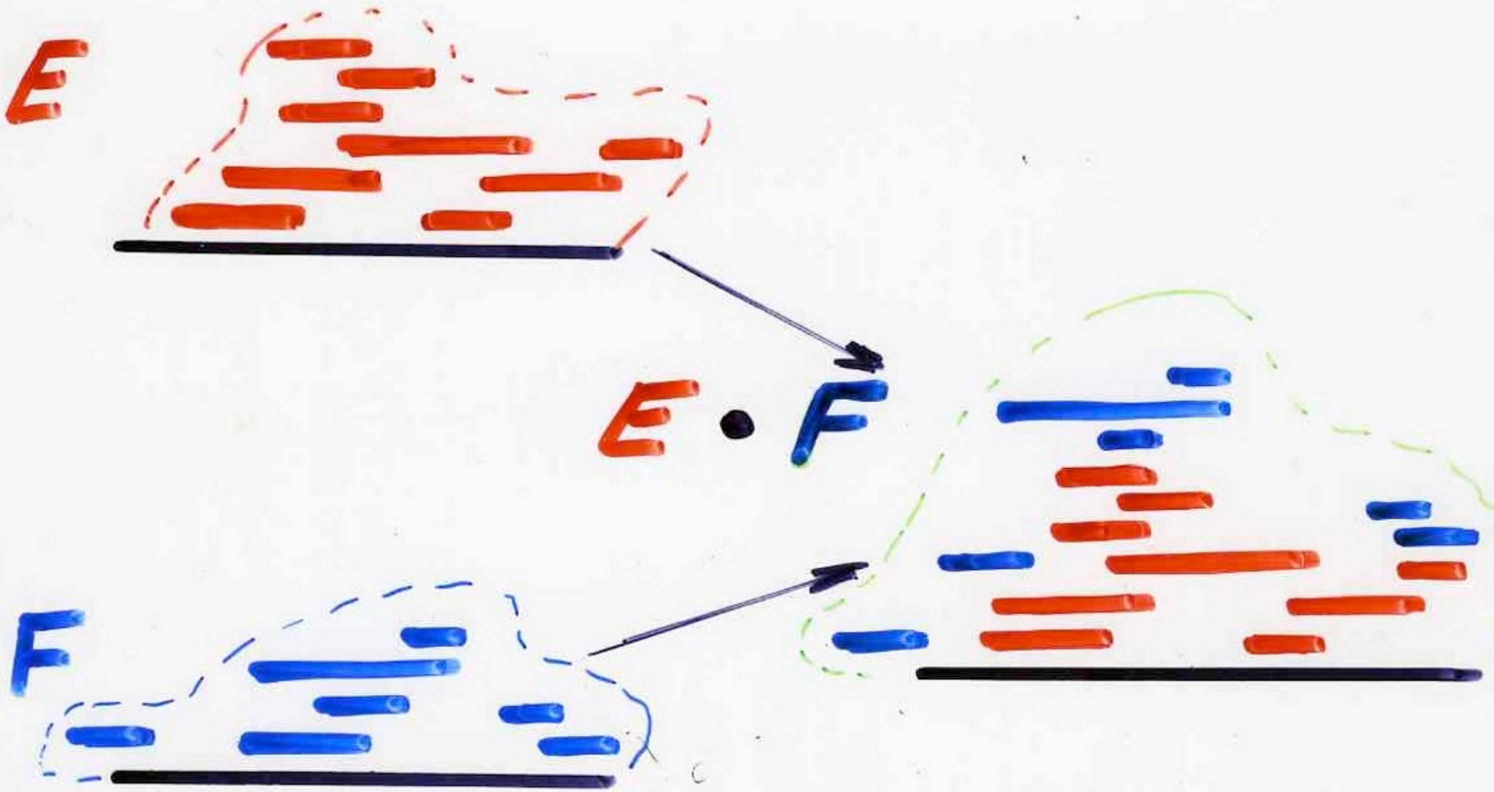
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8


6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8


0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8



6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8



$$E \cdot F = E_{\sim}(E \cup T(F))$$

§4 Equivalence commutation monoids and heaps monoids

$\mathrm{Heap}(P, \mathcal{C}) \simeq$ commutation monoid

heaps monoid

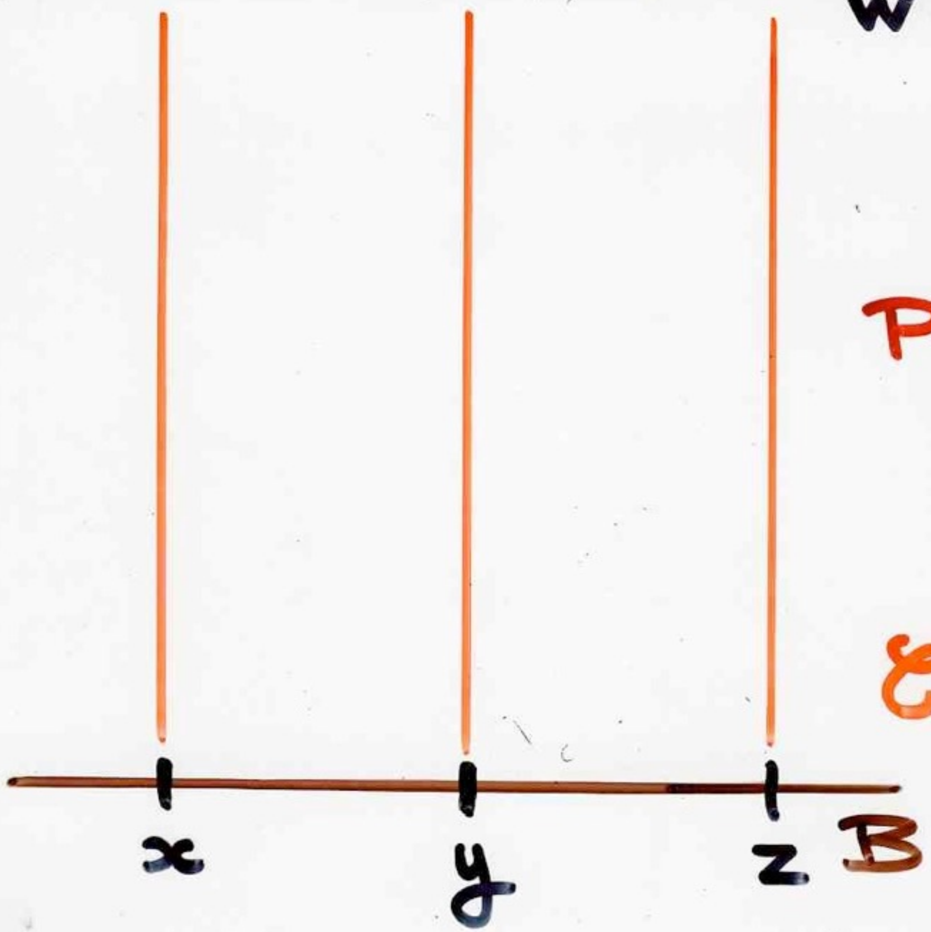
$P \subseteq \mathrm{Heap}(P, \mathcal{C})$

$\alpha \longleftrightarrow \{(\alpha, 0)\}$

$\varphi : P^* \longrightarrow \mathrm{Heap}(P, \mathcal{C})$

$w = \alpha_1 \alpha_2 \cdots \alpha_n \longrightarrow \alpha_1 \odot \alpha_2 \odot \cdots \odot \alpha_n$

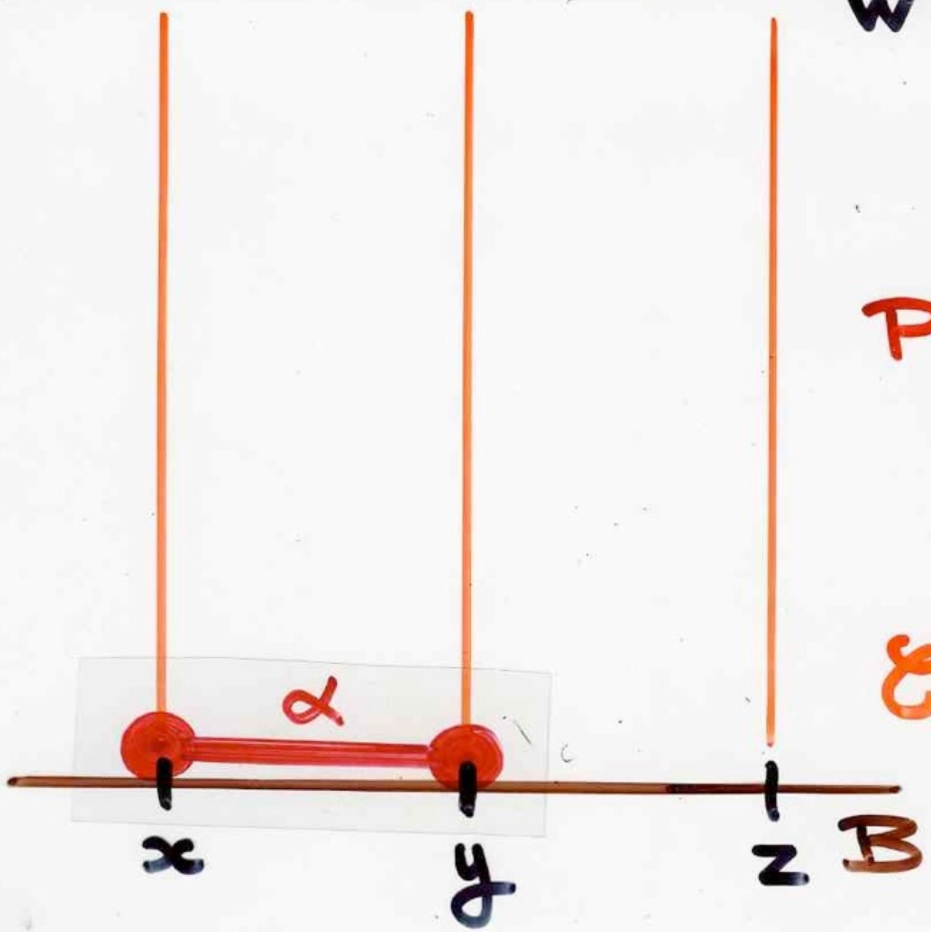
word heap



$W = \alpha\beta\gamma\alpha\delta$

$$P \begin{cases} \alpha = \{x, y\} \\ \beta = \{y, z\} \\ \gamma = \{x\} \\ \delta = \{z\} \end{cases}$$

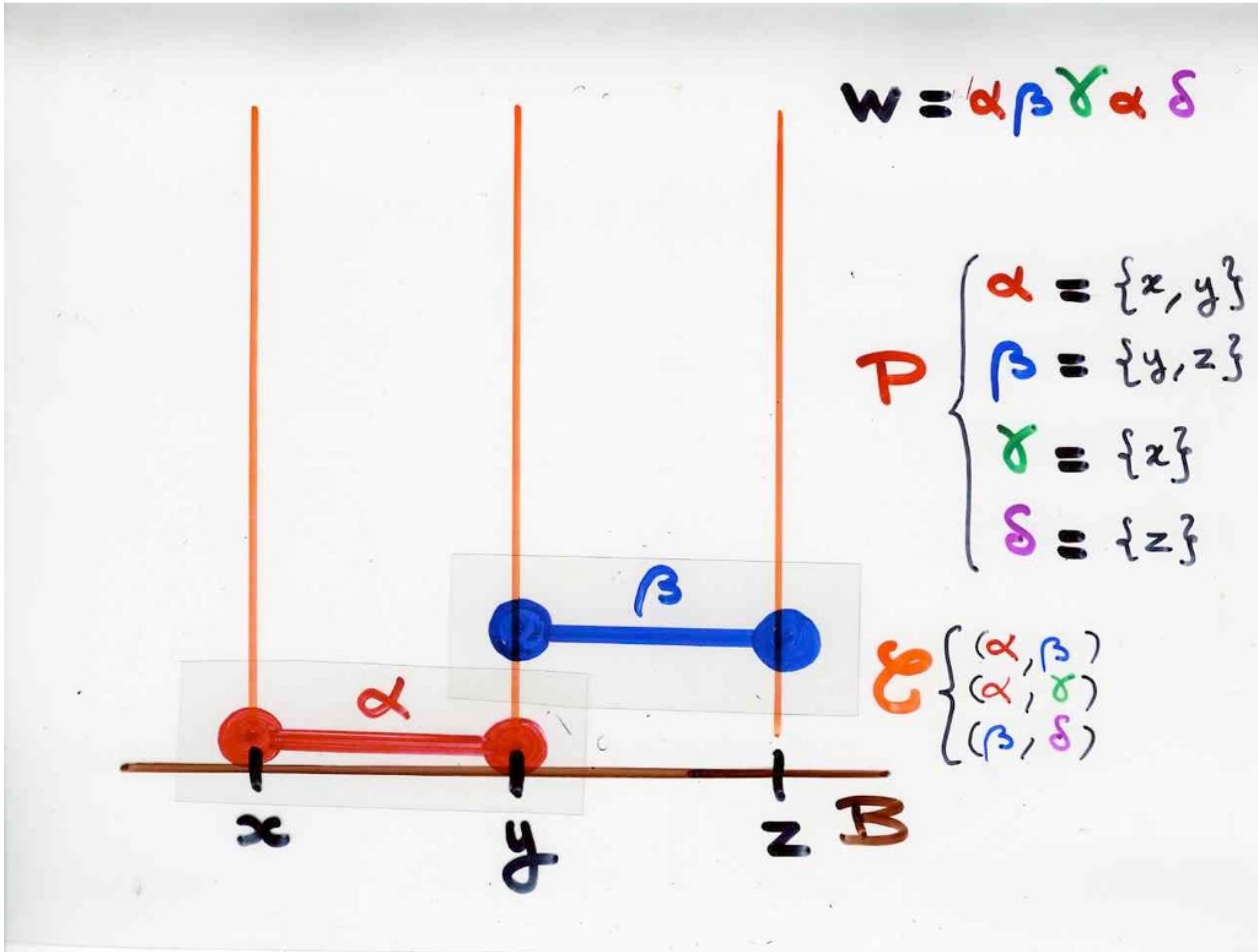
$$\mathcal{C} \begin{cases} (\alpha, \beta) \\ (\alpha, \gamma) \\ (\beta, \delta) \end{cases}$$



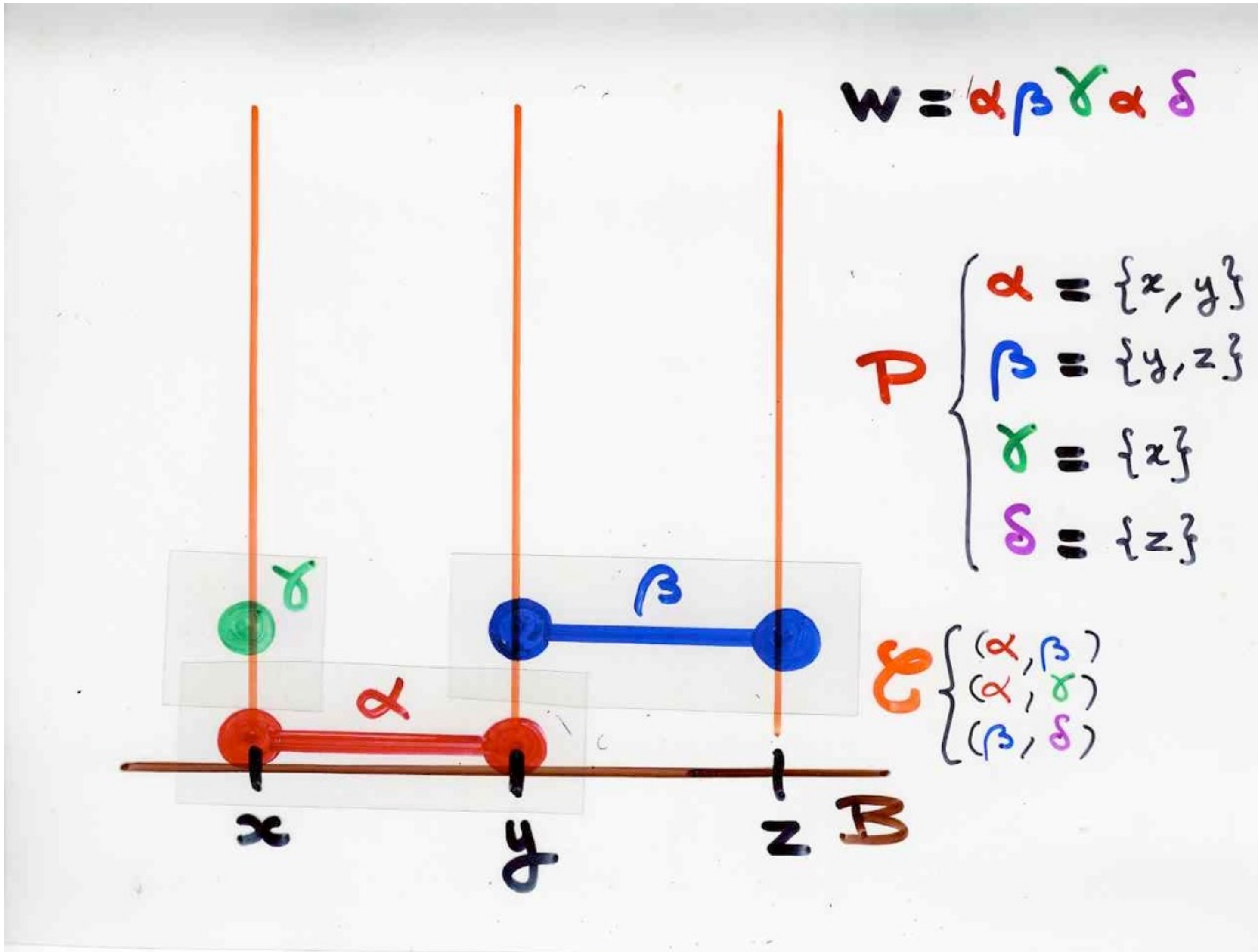
$W = \alpha\beta\gamma\alpha\delta$

$$P \begin{cases} \alpha = \{x, y\} \\ \beta = \{y, z\} \\ \gamma = \{x\} \\ \delta = \{z\} \end{cases}$$

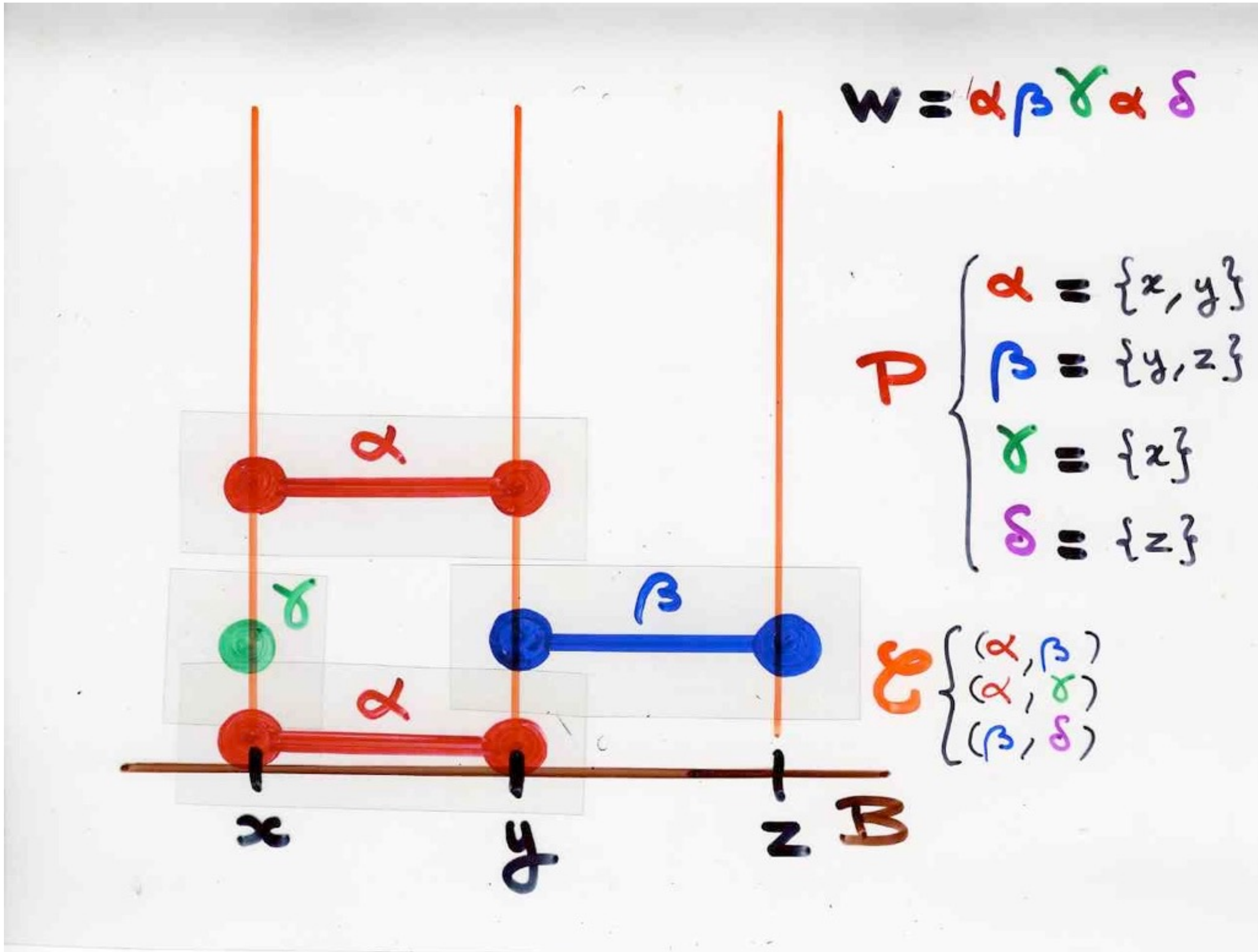
$$\mathcal{E} \begin{cases} (\alpha, \beta) \\ (\alpha, \gamma) \\ (\beta, \delta) \end{cases}$$



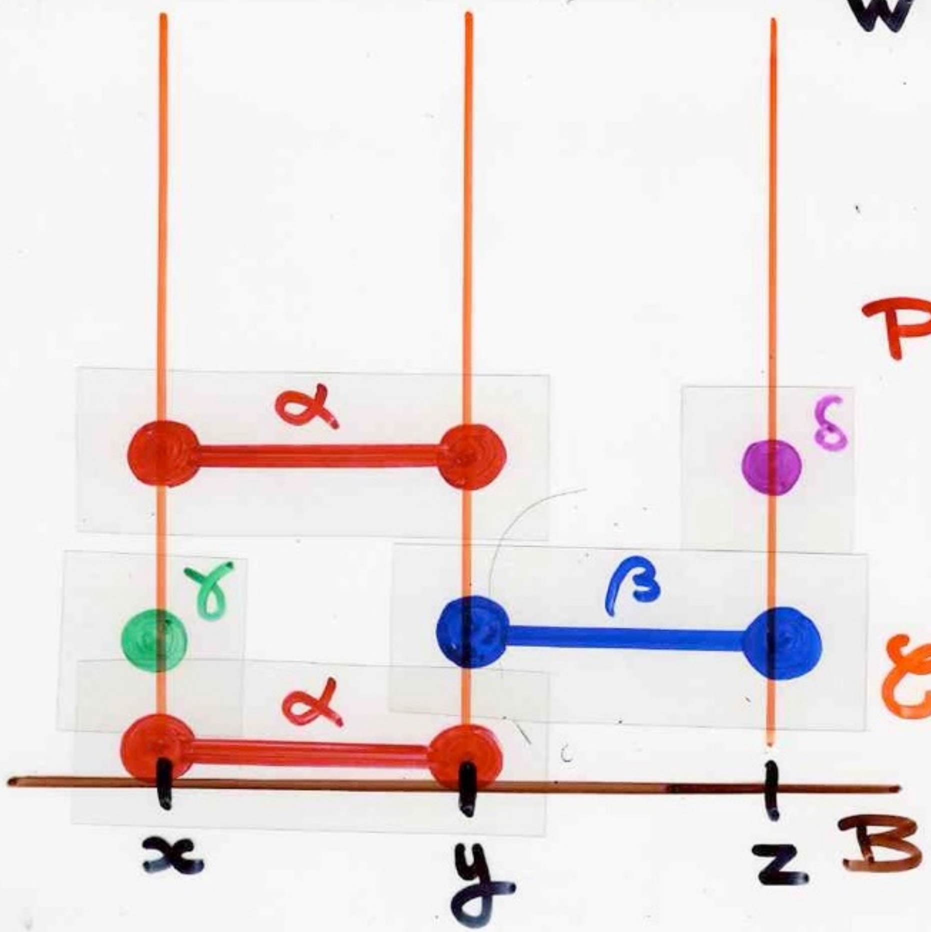
w = αβγαδ
P
α = {x, y}
β = {y, z}
γ = {x}
δ = {z}
ℰ
(α, β)
(α, γ)
(β, δ)
β
α
x
y
z
B



W = α β γ α δ
P
α = {x, y}
β = {y, z}
γ = {x}
δ = {z}
C
(α, β)
(α, γ)
(β, δ)
γ
β
α
x
y
z
B

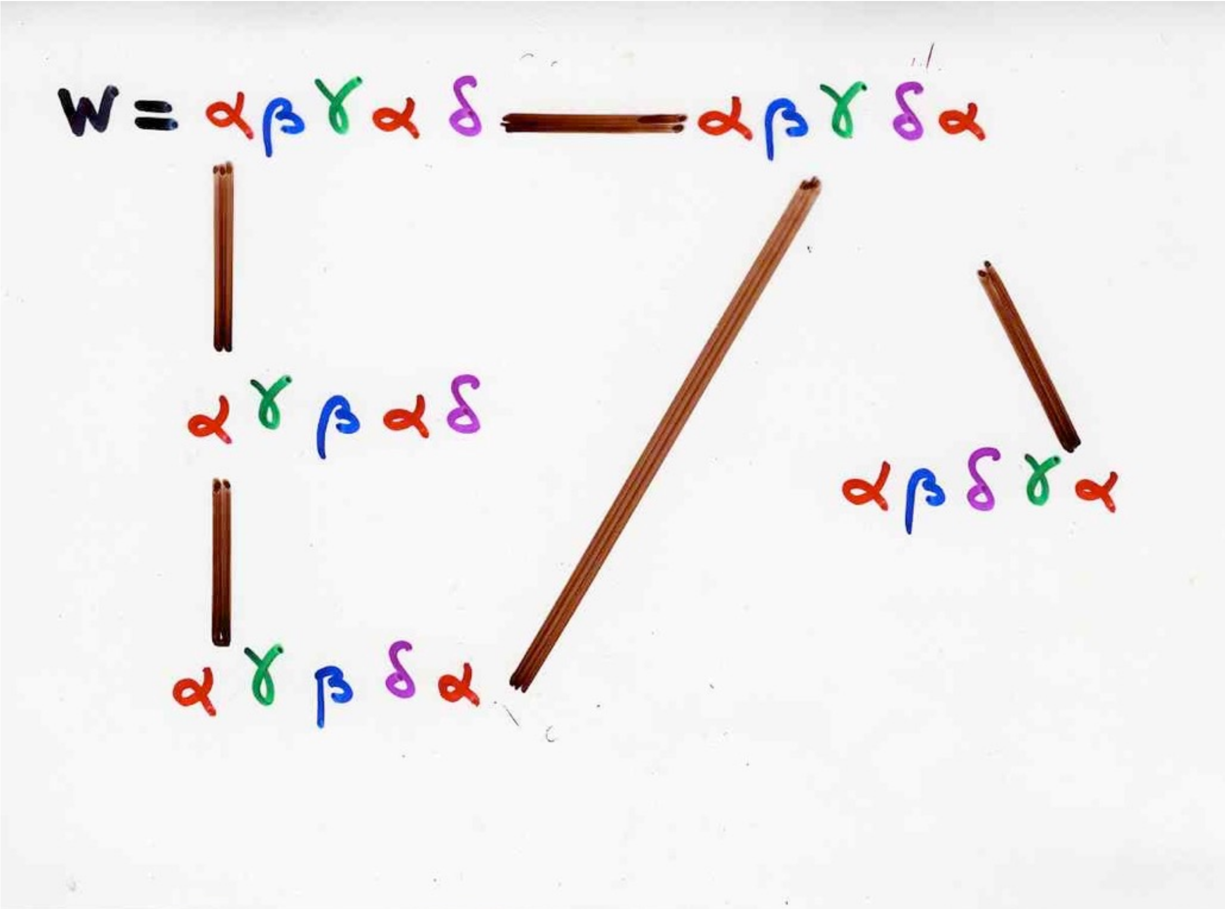

w = αβγαδ
P
α = {x, y}
β = {y, z}
γ = {x}
δ = {z}
(α, β)
(α, γ)
(β, δ)
α
γ
β
α
x
y
z
B

$w = \alpha\beta\gamma\alpha\delta$



$$\mathcal{P}\begin{cases}\alpha = \{x, y\}\\ \beta = \{y, z\}\\ \gamma = \{x\}\\ \delta = \{z\}\end{cases}$$

$$\mathcal{C}\begin{cases}(\alpha, \beta)\\ (\alpha, \gamma)\\ (\beta, \delta)\end{cases}$$



commutations

$c = \overline{\tau}$

$$C \begin{cases} (\alpha, \delta) \\ (\beta, \gamma) \\ (\gamma, \delta) \end{cases}$$

ex: $A = \{a, b, c, d\}$

$C \begin{cases} ad = da \\ bc = cb \\ cd = dc \end{cases}$

equivalence class

w= abcad — abcda

acbad

acbda

abdca

commutations

$C = \overline{C}$

$C \begin{cases} (\alpha, \delta) \\ (\beta, \gamma) \\ (\gamma, \delta) \end{cases}$

$$P \subseteq \mathrm{Heap}(P, \mathcal{C})$$

$$\alpha \longleftrightarrow \{(\alpha, 0)\}$$

$$\varphi : P^* \longrightarrow \mathrm{Heap}(P, \mathcal{C})$$

$$w = \alpha_1 \alpha_2 \cdots \alpha_n \longrightarrow \alpha_1 \odot \alpha_2 \odot \cdots \odot \alpha_n$$

word heap

$$C = \overline{\mathcal{C}}$$

C: commutation relation

$\overline{\mathcal{C}}$: complementary of the dependency relation

Lemma 1

$$u \equiv_C v \implies \varphi(u) = \varphi(v)$$

Lemma 2

$$\varphi(u) = \varphi(v) \implies u \equiv_C v$$

Proof in the next §

<u>Definition</u> $\overline{\varphi}([u]) = \varphi(u)$

$$
\begin{tikzcd}
\mathcal{P}^* \arrow[r, "\varphi"] \arrow[d] & \mathcal{H}(\mathcal{P}, \mathcal{E}) \\
\mathcal{P}^* / \equiv_{\mathcal{C}} \arrow[ur, dashed, "\overline{\varphi}"'] &
\end{tikzcd}
$$

Proposition $\overline{\varphi}$ is an isomorphism of monoids

$$\mathrm{Heap}(P, \mathcal{E}) \simeq P^* / \equiv_C$$

heaps monoid

commutation monoid

$$C = \overline{\mathcal{E}}$$

complementary relation

another example:
heaps of dimers

ex: heaps of dimers on $\mathbb{N}$

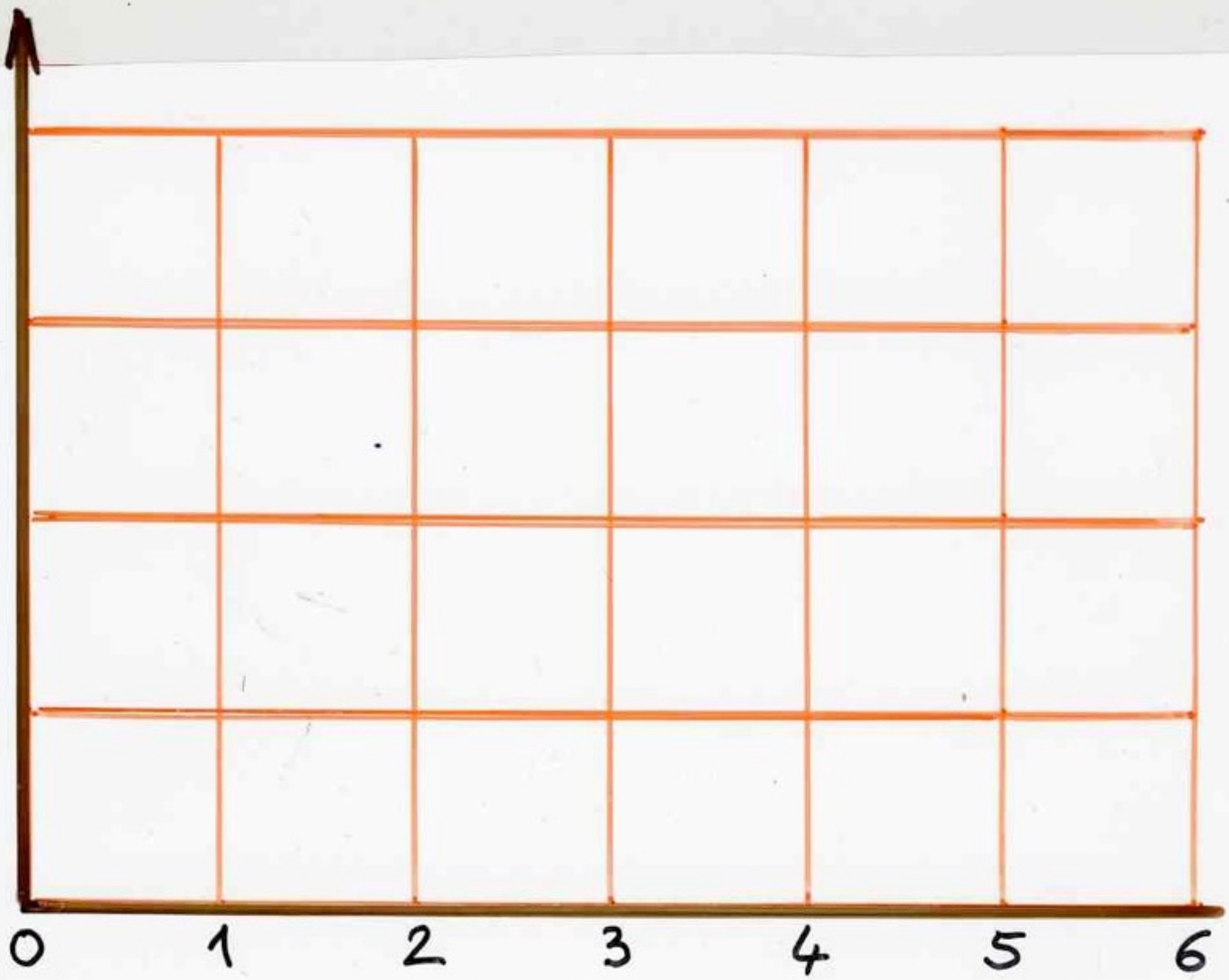
$$P = \{ [i, i+1] = \sigma_i , i \geq 0 \}$$

$\mathcal{C}$

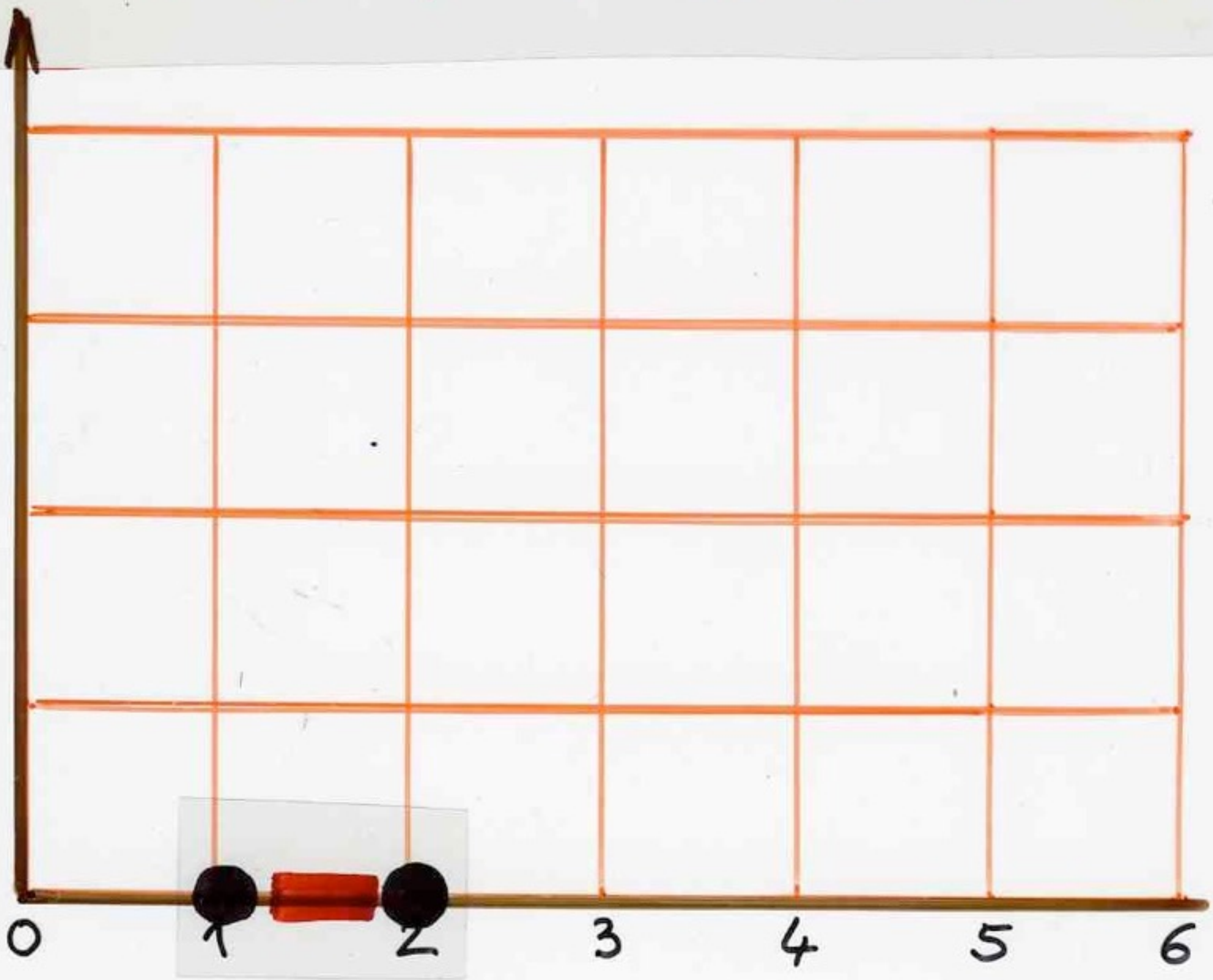
C commutations

$$\sigma_i \sigma_j = \sigma_j \sigma_i \text{ iff } |i - j| \geq 2$$

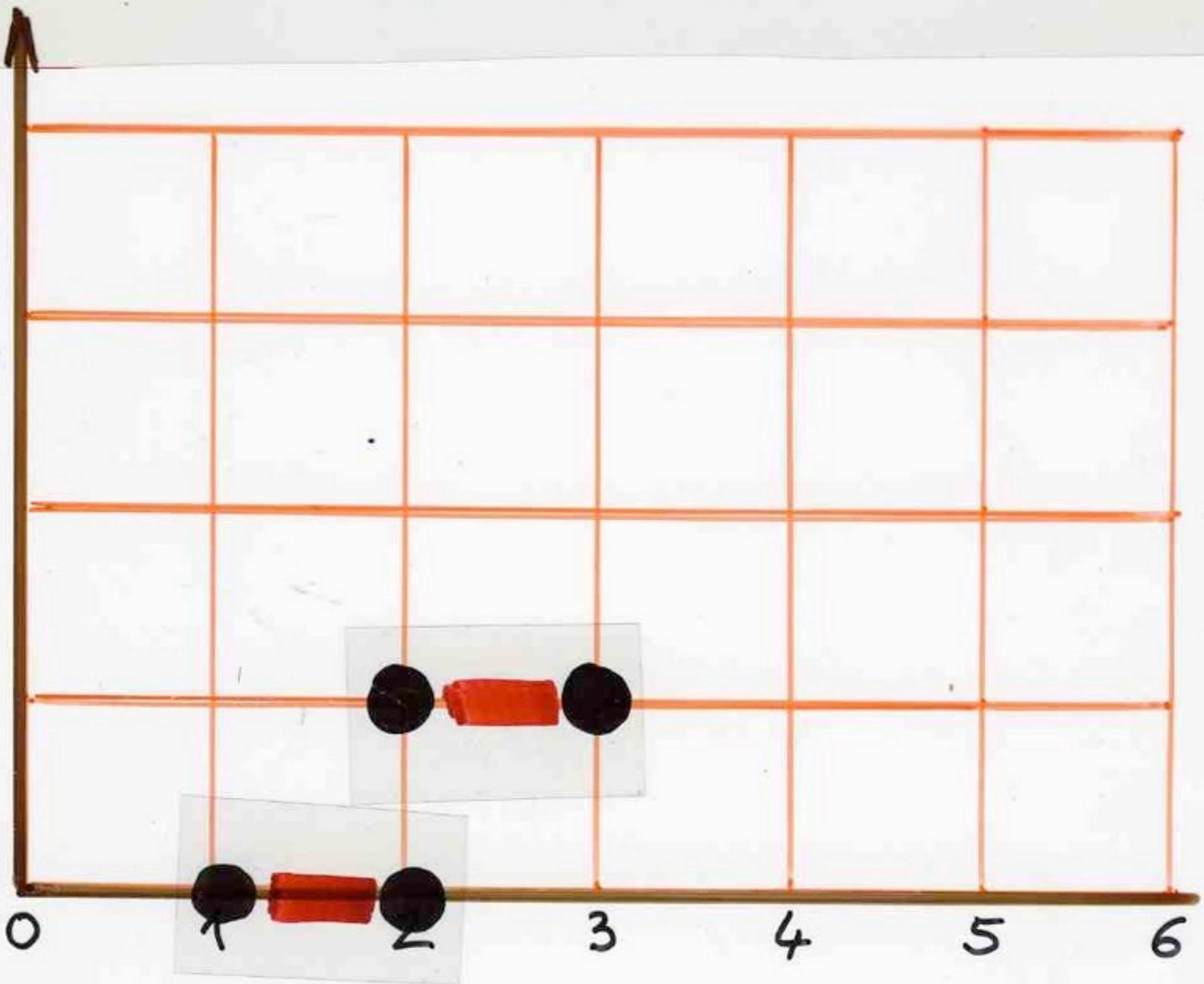
$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$



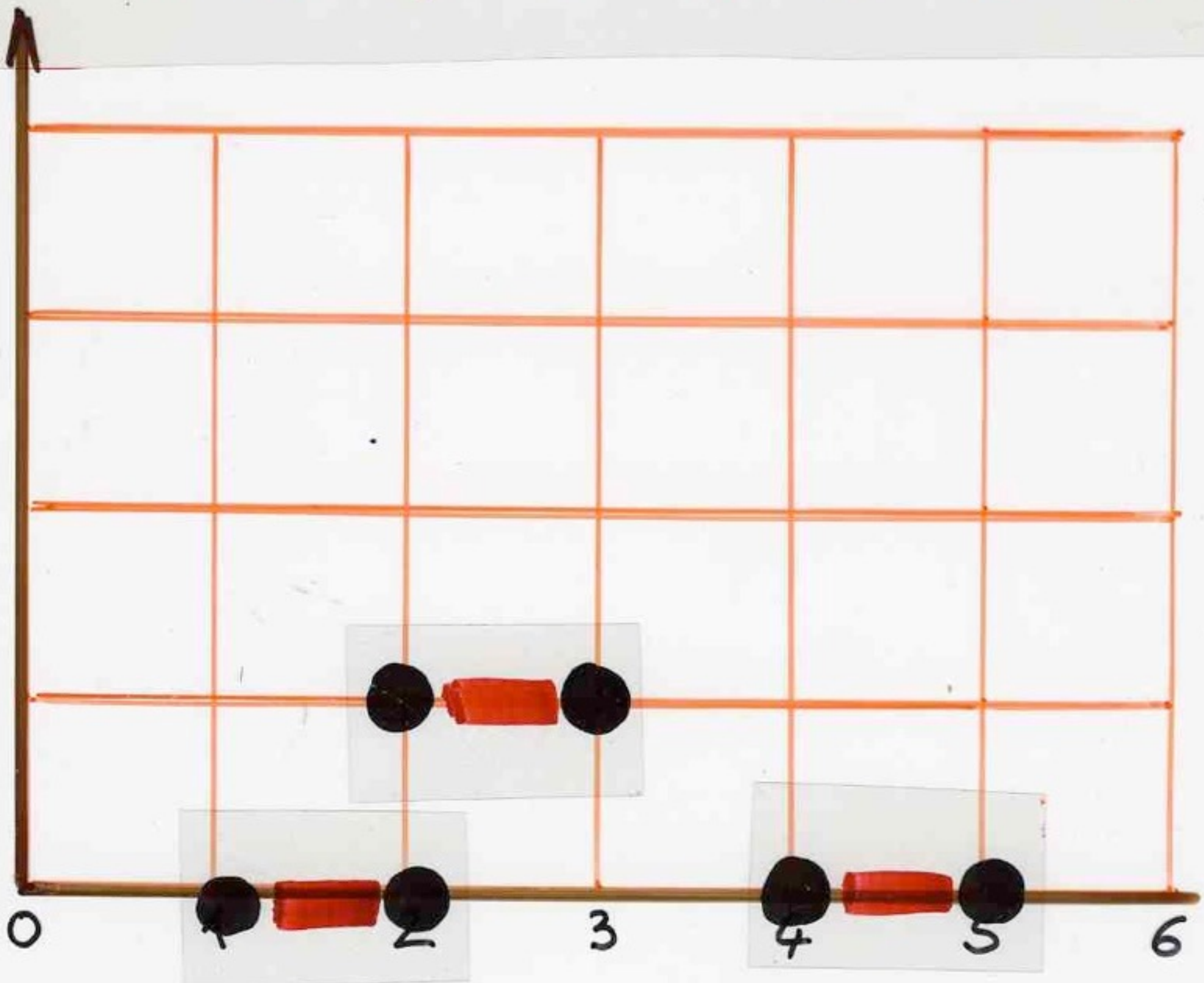
$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$



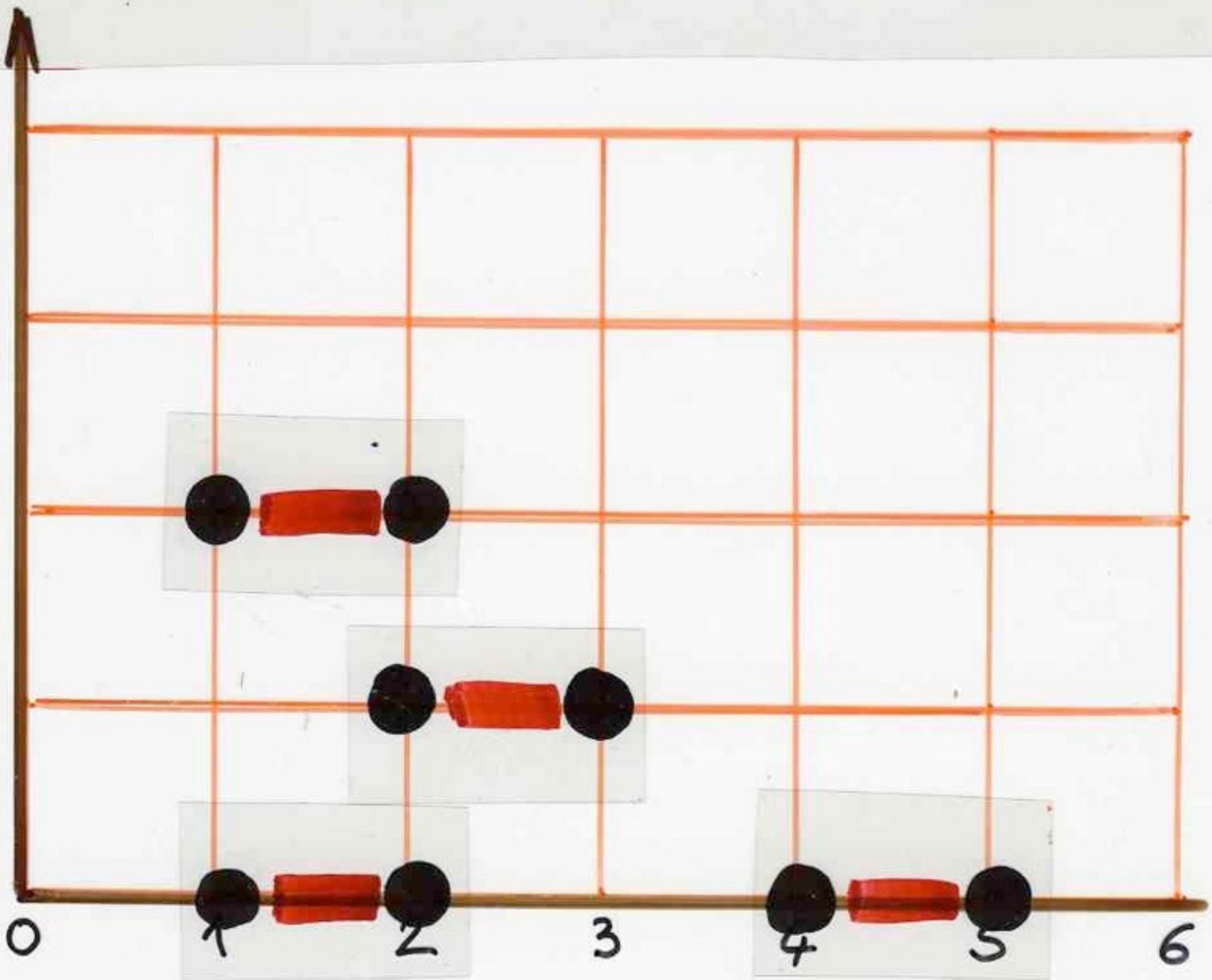
$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$



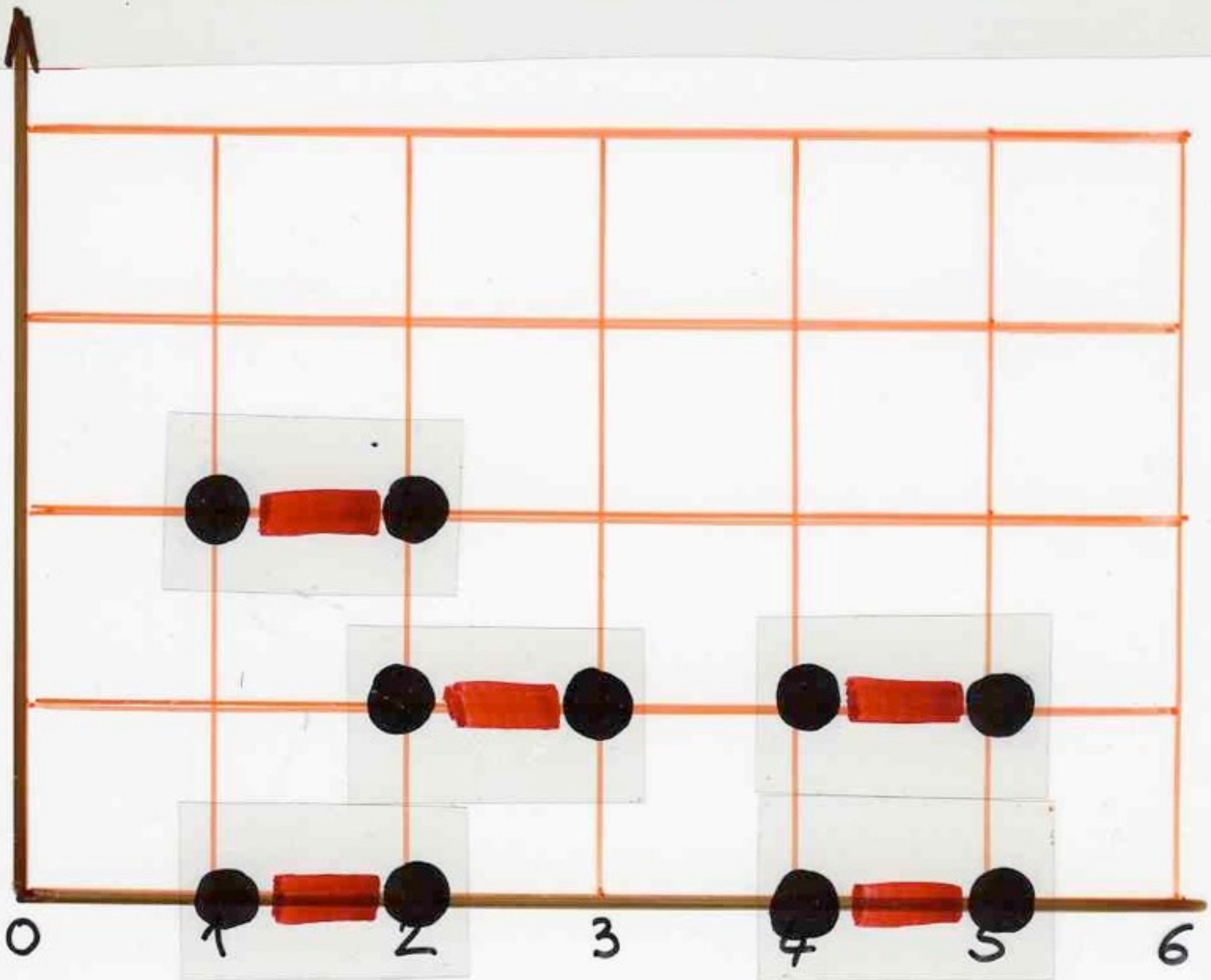
$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$



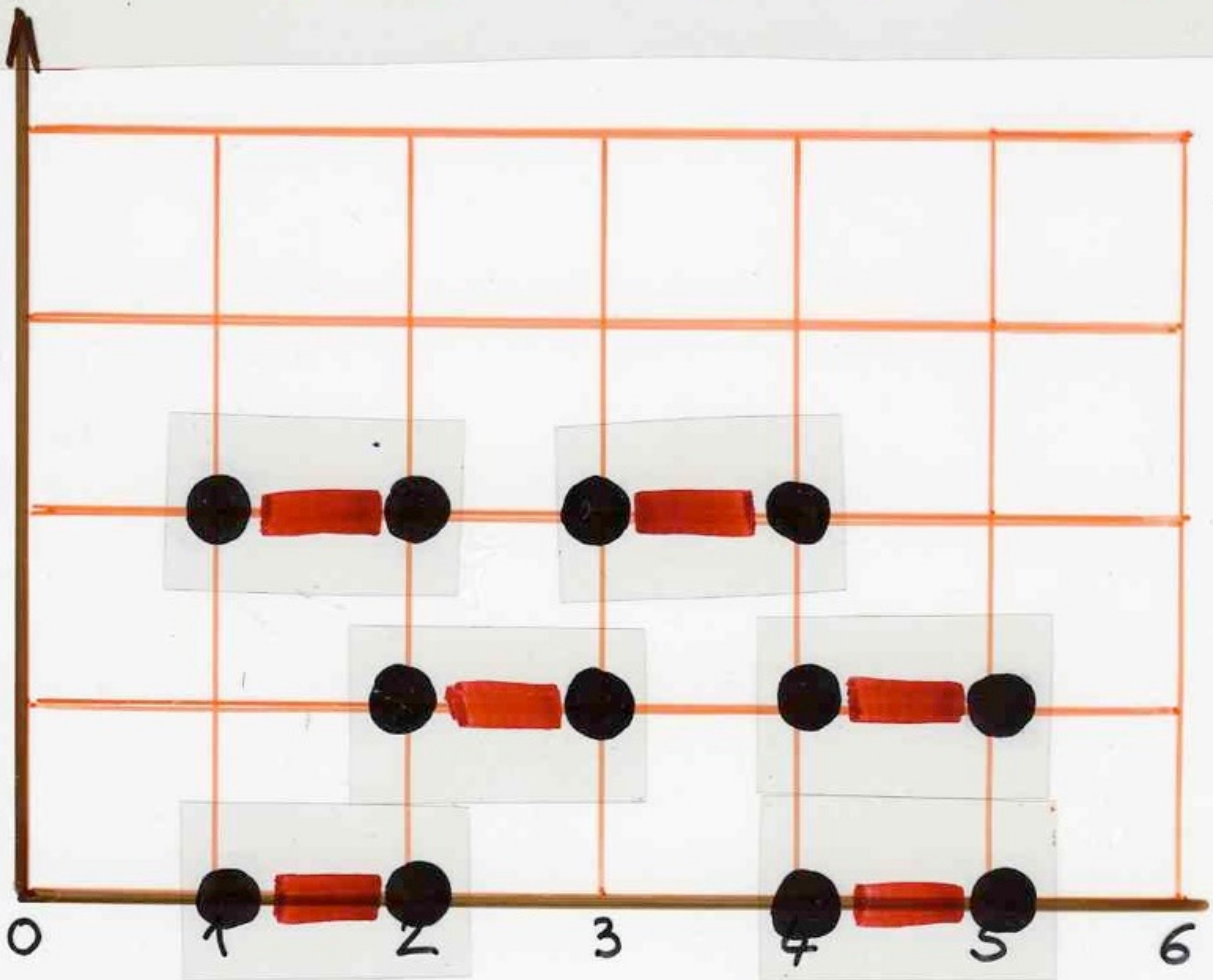
$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$



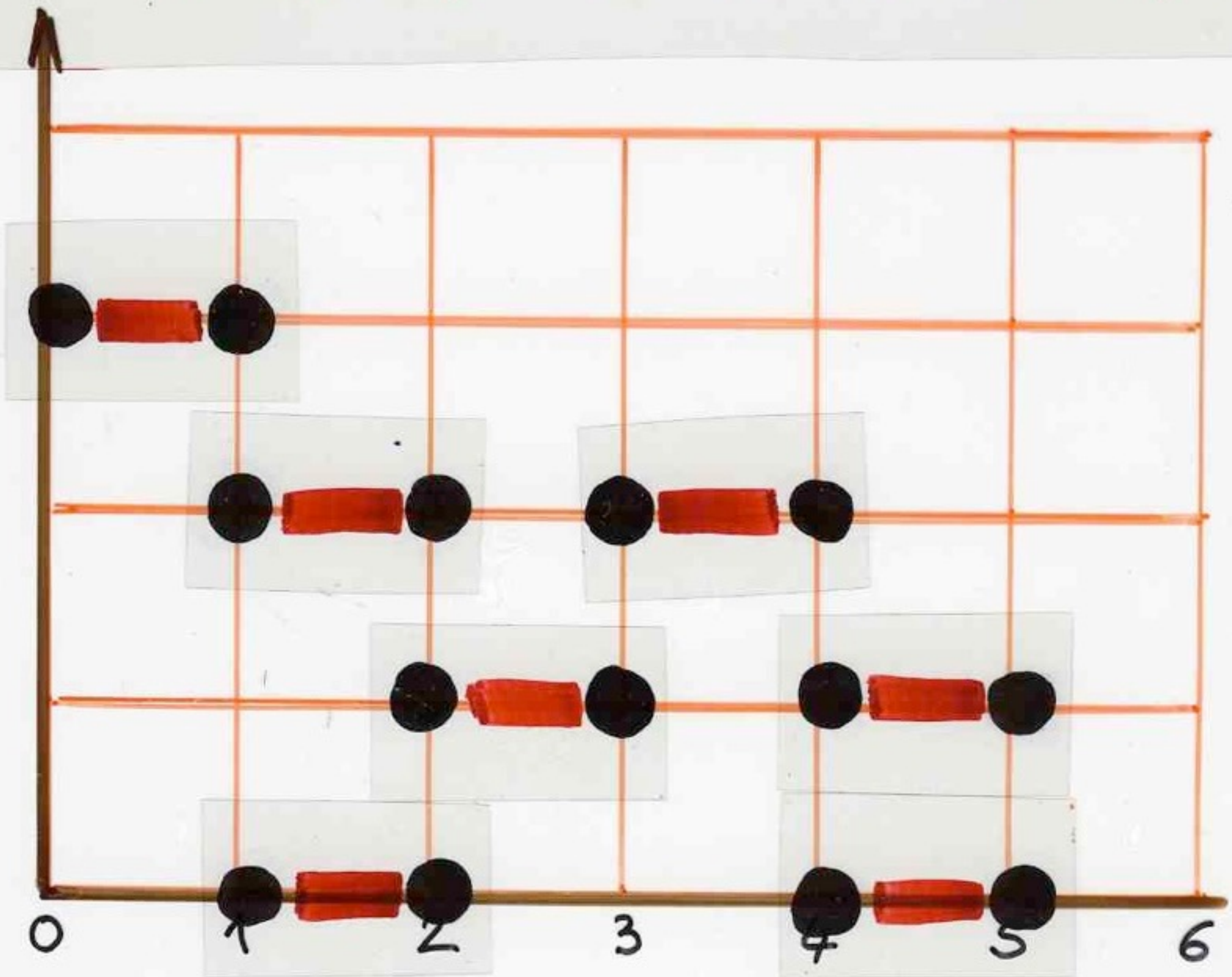
$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$



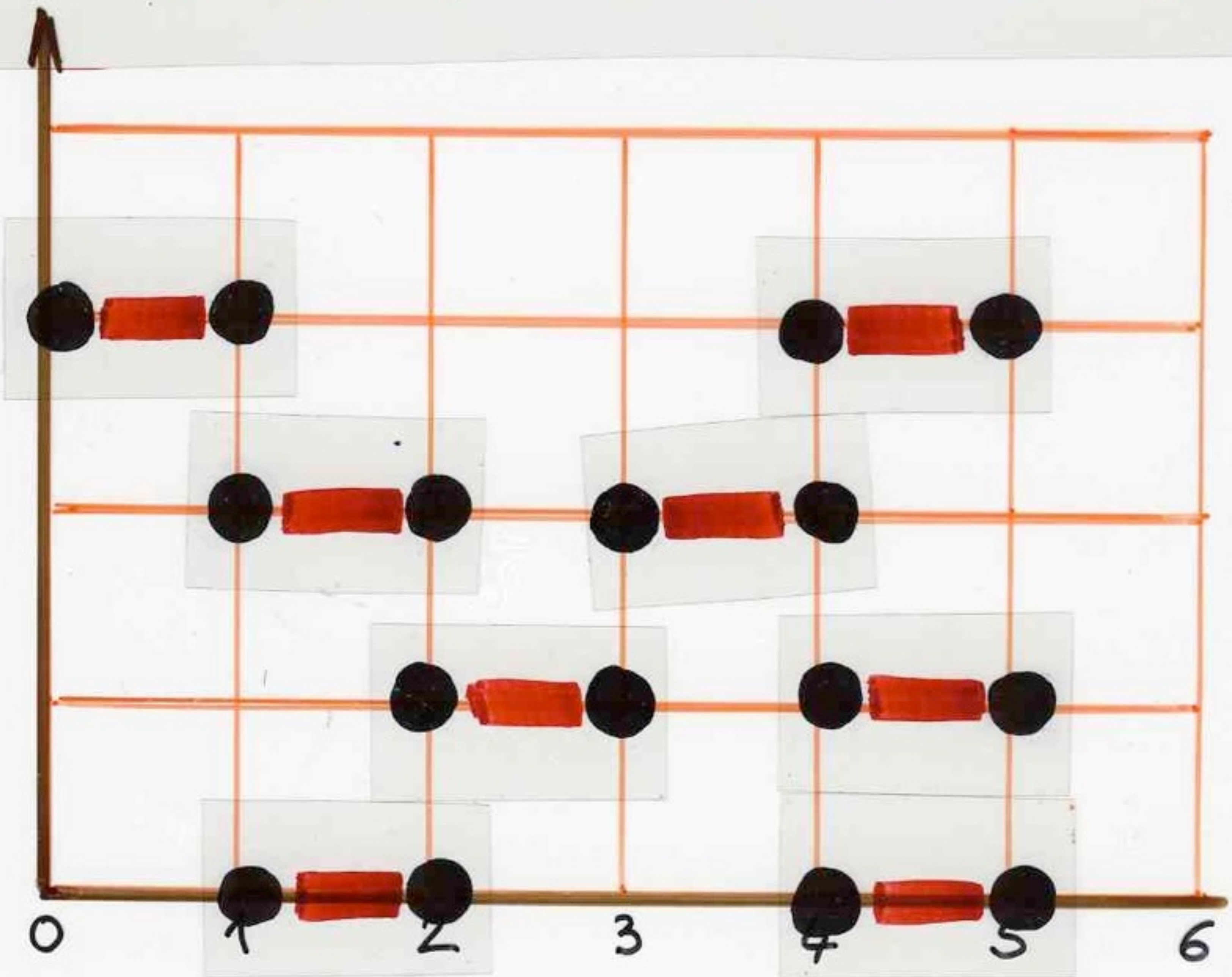
$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$



$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$

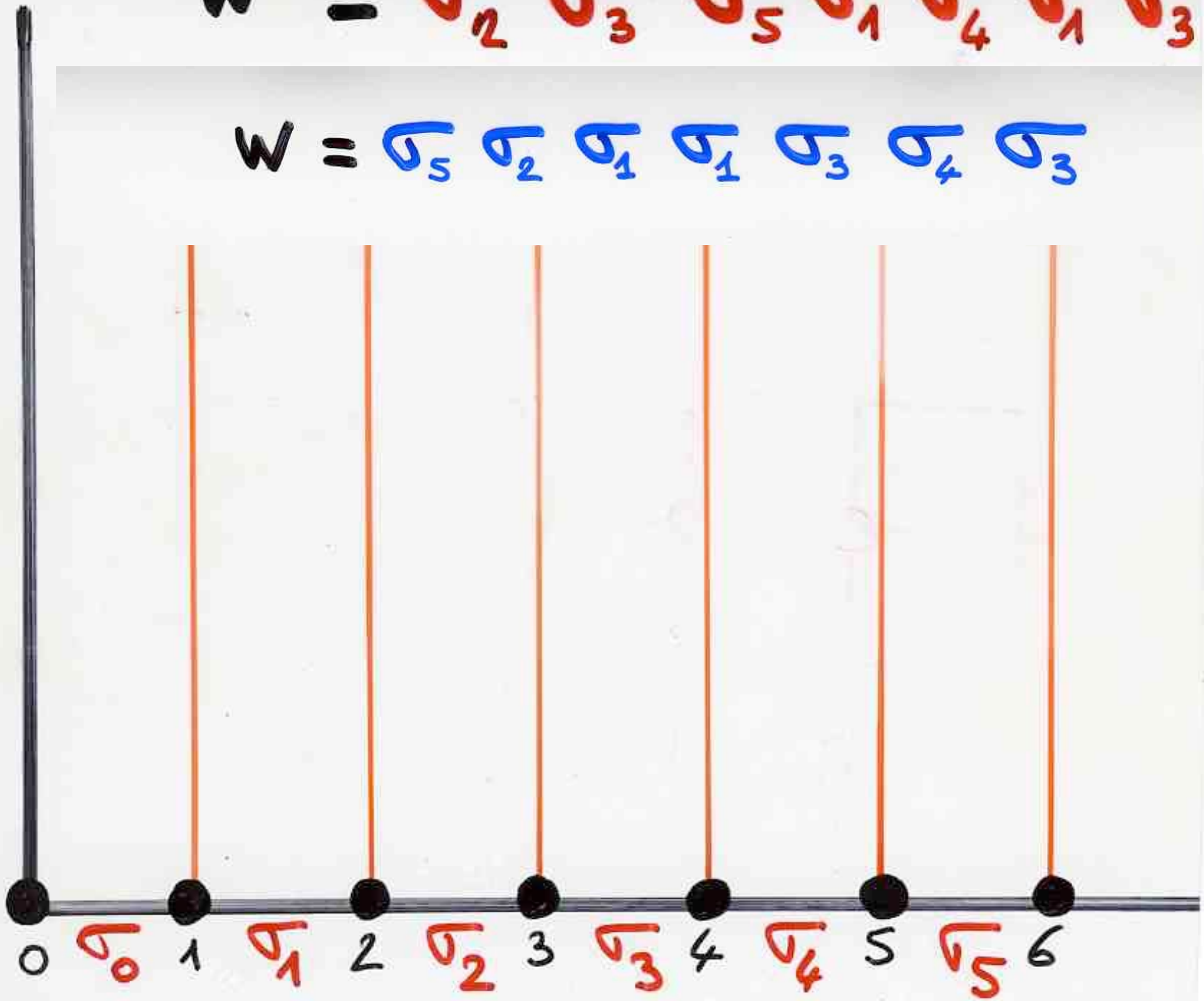


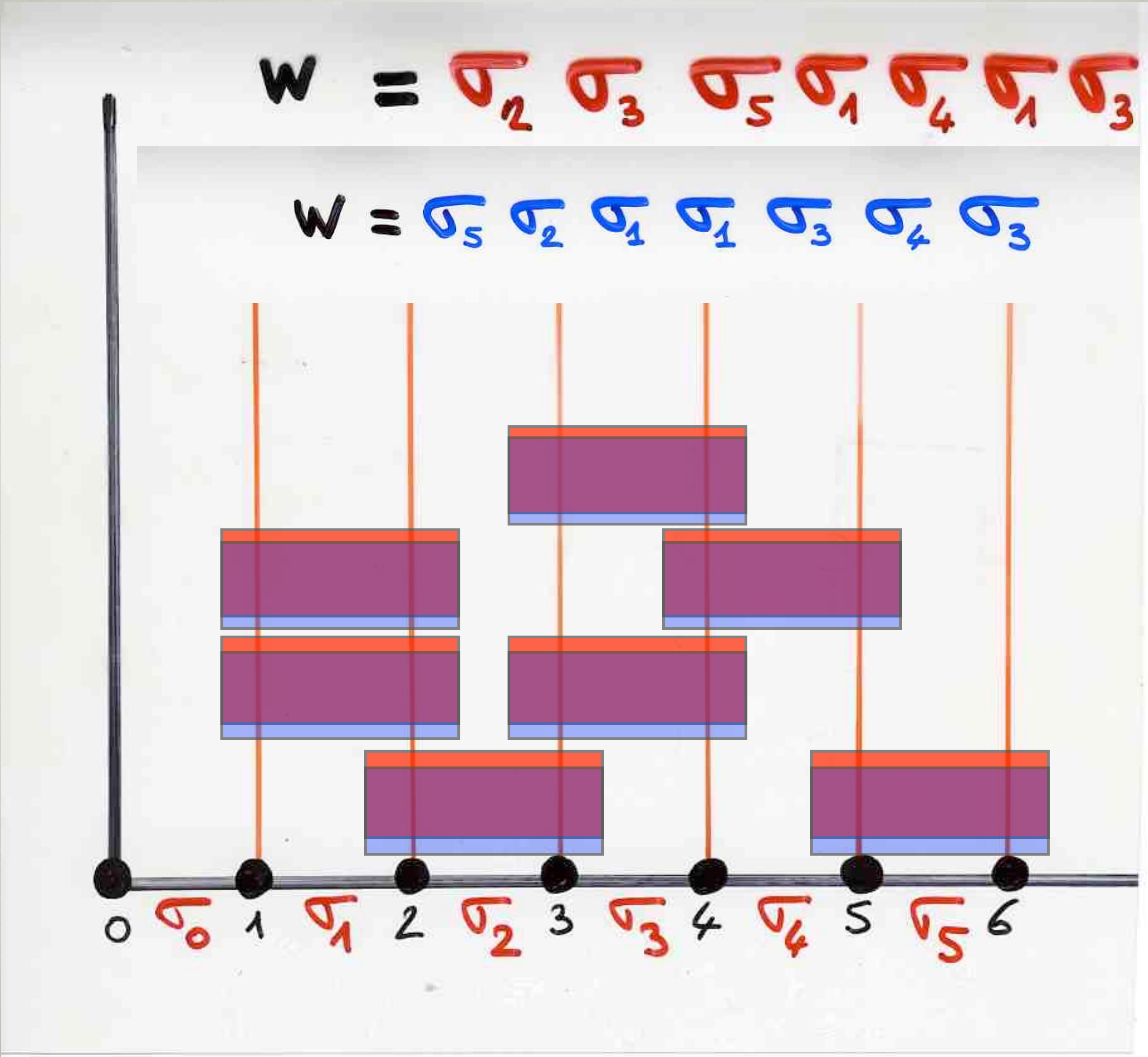
$$w = \sigma_1 \sigma_2 \sigma_4 \sigma_1 \sigma_4 \sigma_3 \sigma_0 \sigma_4$$



$$w = \sigma_2 \sigma_3 \sigma_5 \sigma_1 \sigma_4 \sigma_1 \sigma_3$$

$$w = \sigma_5 \sigma_2 \sigma_1 \sigma_1 \sigma_3 \sigma_4 \sigma_3$$





$w = \sigma_2 \sigma_3 \sigma_5 \sigma_1 \sigma_4 \sigma_1 \sigma_3$
$w = \sigma_5 \sigma_2 \sigma_1 \sigma_1 \sigma_3 \sigma_4 \sigma_3$
0 σ_0 1 σ_1 2 σ_2 3 σ_3 4 σ_4 5 σ_5 6

braid group B_n

symmetric group S_n

Temperley-Lieb algebra $A_n(\tau)$

$$\begin{cases} \sigma_i\sigma_j = \sigma_j\sigma_i & |i-j|\geq 2 \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} \end{cases}
\qquad
\begin{cases} \sigma_i^2 = 1 \\ \sigma_i\sigma_j = \sigma_j\sigma_i & |i-j|\geq 2 \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} \end{cases}
\qquad
\begin{cases} e_i^2 = e_i \\ e_ie_j = e_j\sigma_i & |i-j|\geq 2 \\ e_ie_{i\pm 1}e_i = \tau e_i \end{cases}$$

ex: heaps of dimers on $\mathbb{N}$

$P = \{ [i, i+1] = \sigma_i \, , \, i \geq 0 \}$

$\mathcal{C}$

C commutations

$\sigma_i \sigma_j = \sigma_j \sigma_i$ iff $|i-j| \geq 2$

Course IMSc

January-March 2016

An introduction to

enumerative
algebraic combinatorics
bijective

coursimsc2016.xavierviennot.org

content of the course

3 basic lemma

- generating functions for **heaps** $\qquad \dfrac{N}{D}$ "trivial" heaps
- log (heaps) = pyramids
- path = heap

Basic definitions and theorems

- commutation monoids and heaps of pieces: basic definitions
- generating functions for heaps
  - $\dfrac{1}{D}$, $\dfrac{N}{D}$, inversion lemma
  - logarithmic lemma
- Heaps and paths, flow monoid, rearrangements

path = heap

rearrangement = $\text{heap}_{\text{cycles}}$

Some applications to classical mathematics

- heaps and linear algebra:
  bijective proofs of classical theorems
- heaps and combinatorial theory of
  orthogonal polynomials and continued fraction
- heaps and algebraic graph theory

Some applications in theoretical physics

- directed animals and gas model in statistical physics
- Lorentzian triangulations in 2D quantum gravity
- q-Bessel functions in physics: polyominoes and SOS model

Applications to more advanced mathematics

- fully commutative class of words
  in Coxeter groups

  → representation theory of Lie algebras
  with operators on heaps

  Temperley-Lieb algebra

Complementary topics

- zeta function on graph and number theory
  (Giscard, Rochet)
- minuscule representations of Lie algebra
  (R. Green and students) book
- basis of free partially commutative Lie algebra (Lalonde, Duchamp-Krob, ...)
  - computer science:
    the SAT problem revisited with heaps
    (D. Knuth, vol 4, Fascicle 6)
  - computer science:
    Petri nets, asynchronous automata,
    Zielonka theorem
- statistical physics: (T. Helmuth)
  Ising model revisited
- string theory and heaps
  gauge theory, quivers
  (Ramgoolam)

3 basic lemma

- generating functions for **heaps** $\dfrac{N}{D}$ ← "trivial" ← heaps
- log (heaps) = pyramids
- path = heap

Some applications to classical mathematics

- heaps and linear algebra :
  bijective proofs of classical theorems
- heaps and combinatorial theory of
  orthogonal polynomials and continued fraction
- heaps and algebraic graph theory

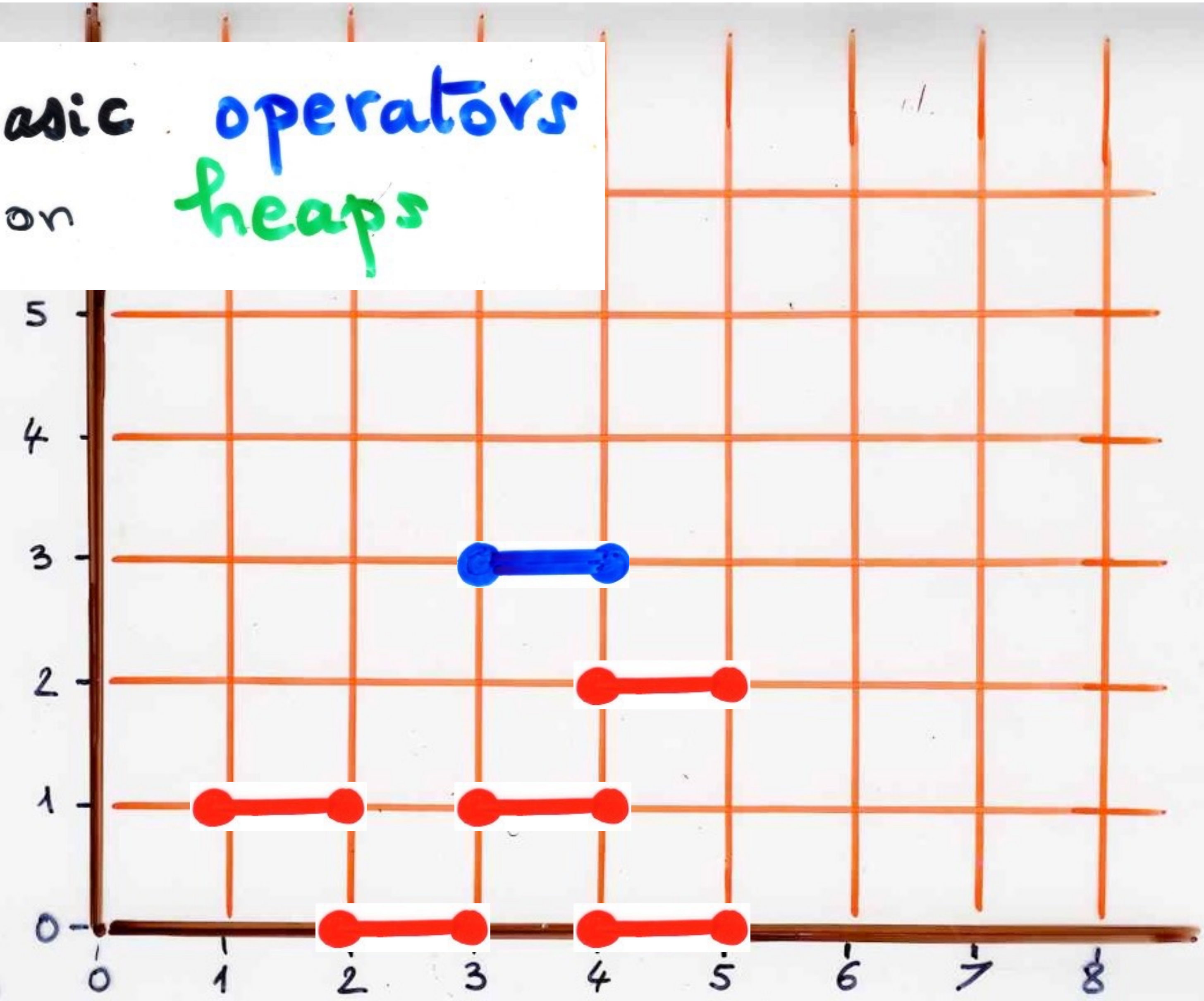
Some applications in theoretical physics

- directed animals and gas model
  in statistical physics
- Lorentzian triangulations in 2D
  quantum gravity
- q-Bessel functions in physics :
  polyominoes and SOS model

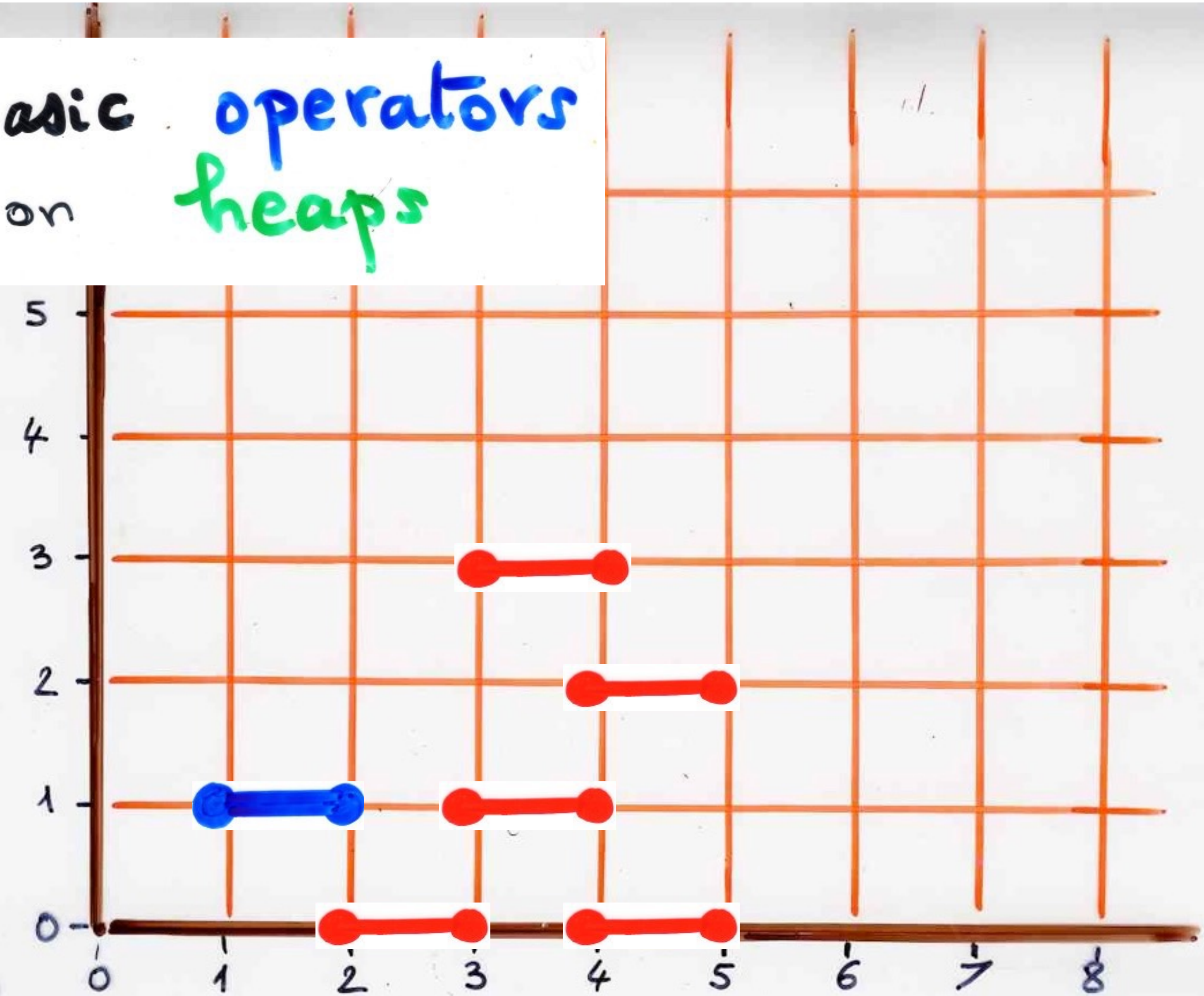
Complementary topics

- zeta function on graph and number theory
  (Giscard, Rochet)
- minuscule representations of Lie algebra
  (R. Green and students) book
- computer science:
  the SAT problem revisited with heaps
  (D. Knuth, vol 4, Fascicle 6)
- computer science:
  Petri nets, asynchronous automata,
  Zielonka theorem
- statistical physics: (T. Helmuth)
  Ising model revisited
- string theory and heaps
  gauge theory, quivers
  (Ramgoolam)

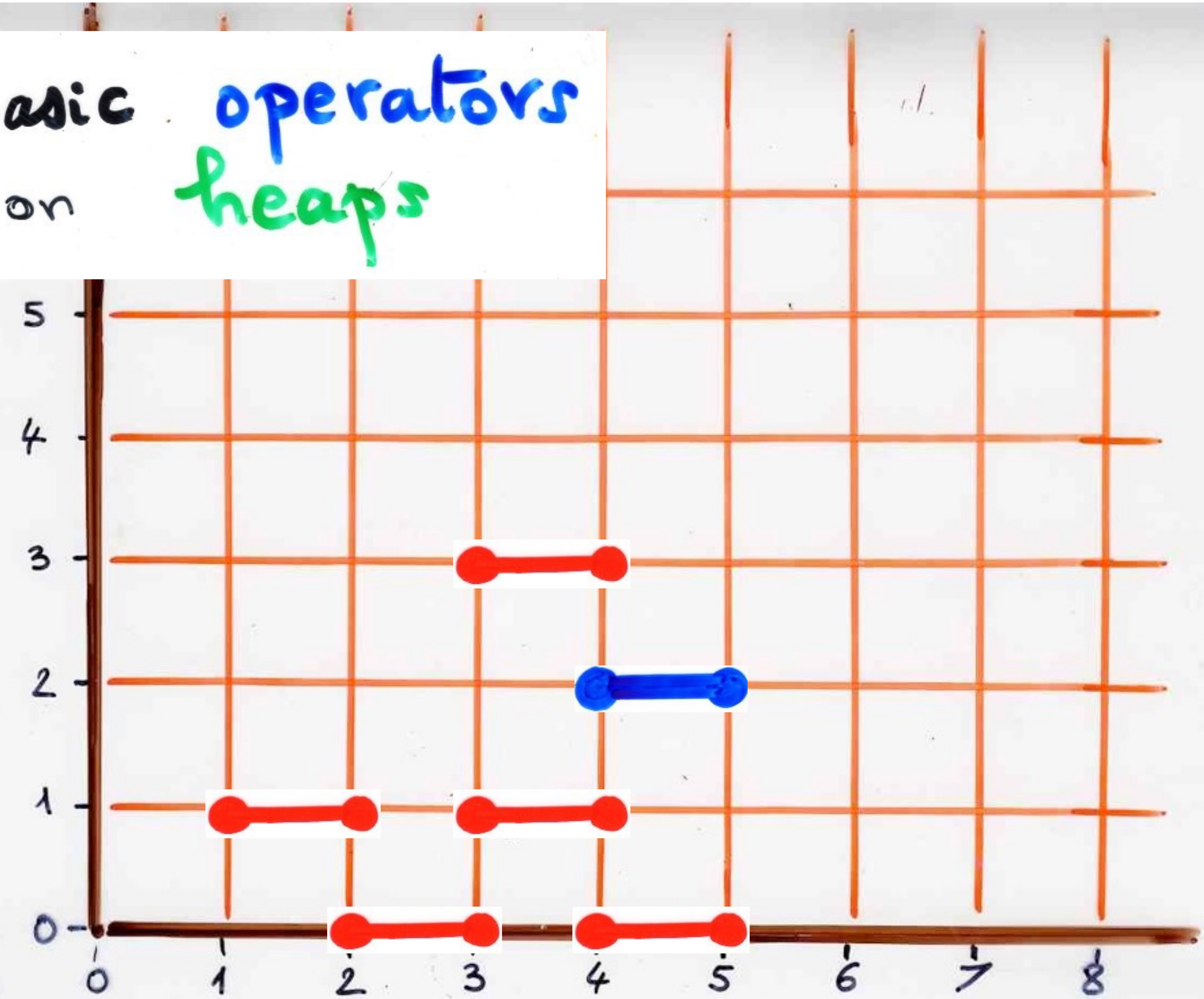
basic operators on heaps



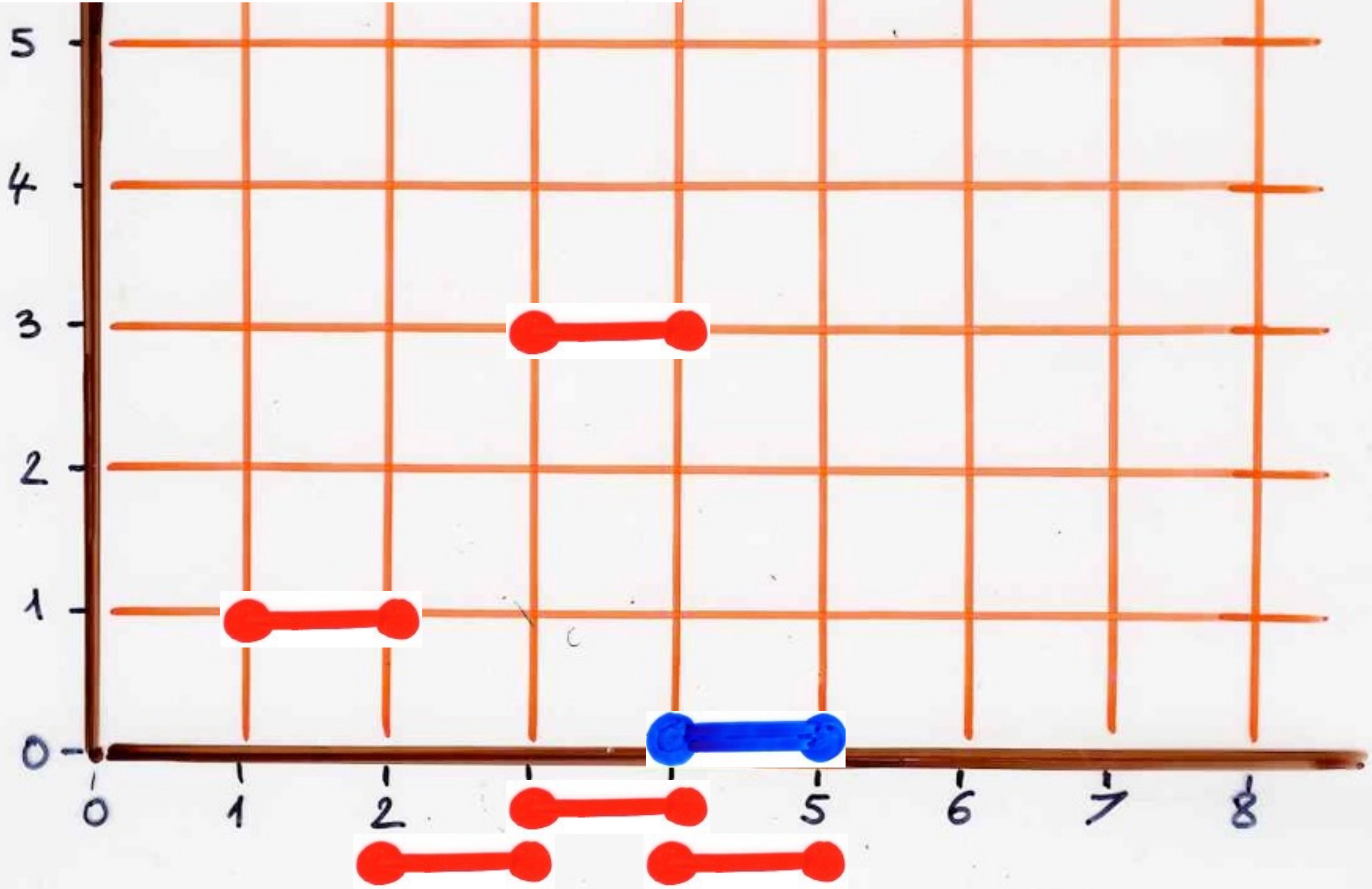
basic operators on heaps



basic operators on heaps



basic operators on heaps



www.xavierviennot.org/coursIMSc2017

or

coursimsc2017.xavierviennot.org/contents.html