A Space and Time Efficient Implementation for Computing Persistent Homology

Clément Maria

December 14, 2012

1 Introduction

Background

Abstract Simplicial Complex.

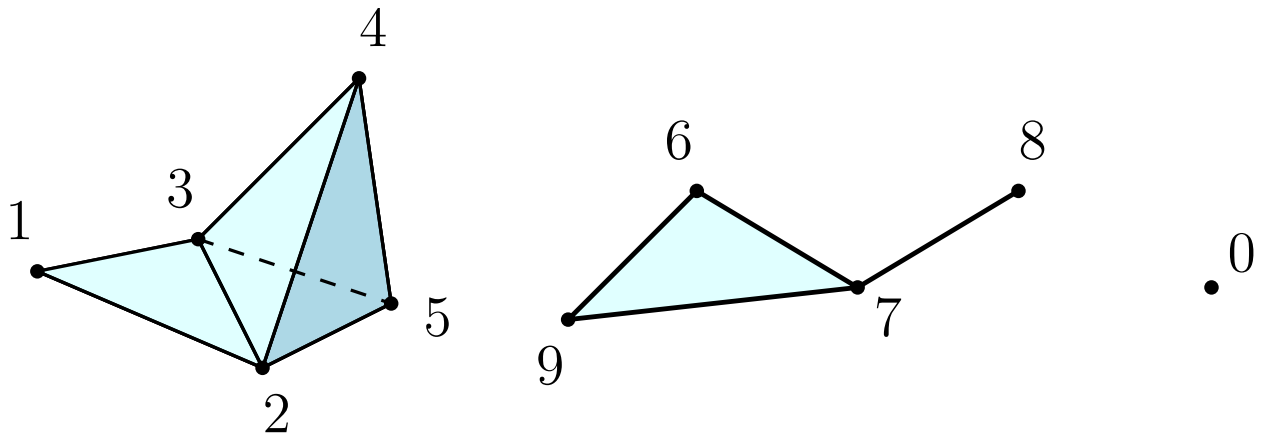
Given a set $V = \{1 \cdots n\}$ of vertices, an **abstract simplicial complex** $\mathcal{K}$ on V is a family of subsets of vertices s.t.:

$$\forall \sigma \in \mathcal{K} : \tau \subseteq \sigma \Rightarrow \tau \in \mathcal{K}$$

Such σ is called a simplex

Dimension of a simplex σ: $\dim(\sigma) = |\sigma| - 1$

Dimension of the simplicial complex $\mathcal{K}$: $\dim(\mathcal{K}) = \max_{\sigma \in \mathcal{K}} \dim(\sigma)$.



Really Short Introduction to Persistent Homology with $\mathbb{Z}_2$ Coefficients

Given a simplicial complex $\mathcal{K}$, $\mathcal{K}^p$ the p-dimensional simplices:

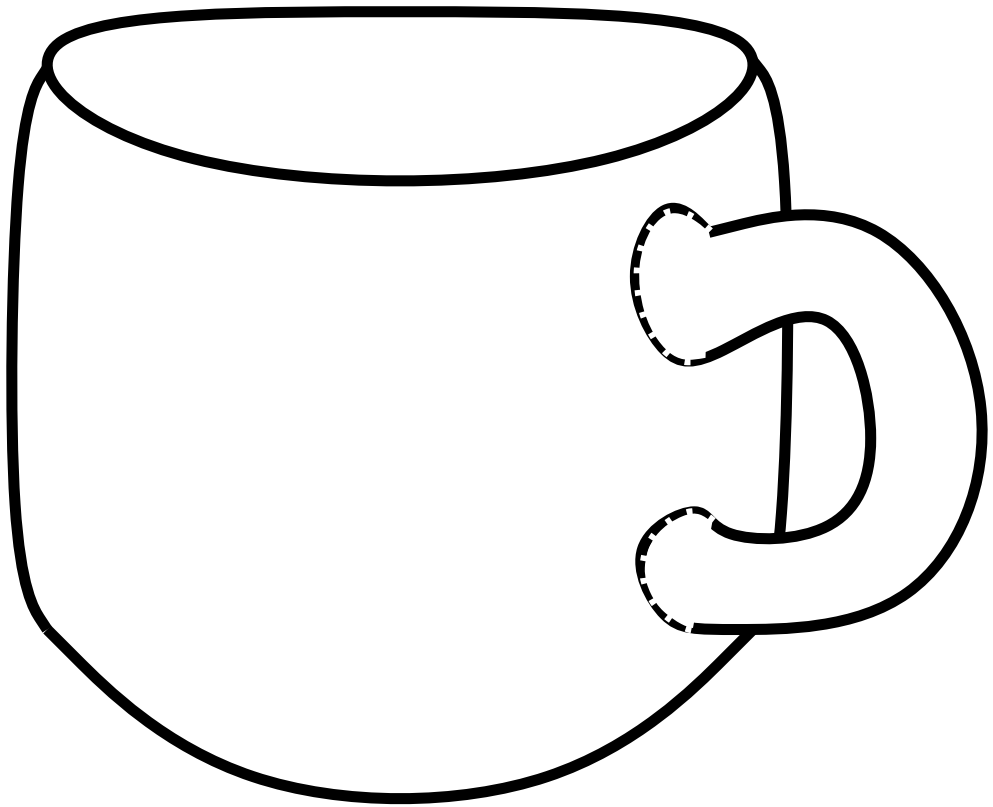
$C_p = 2^{\mathcal{K}^p}$

$$\begin{array}{lccl} \partial_p : & C_p & \rightarrow & C_{p-1} \\ & \sigma = \{v_0, \cdots, v_p\} & \rightarrow & \partial\sigma = \sum_{i=0\cdots p} \{v_0, \cdots, \widehat{v_i}, \cdots, v_p\} \\ & \partial_p \partial_{p+1} = 0 & & \end{array}$$

$$\left\{ \begin{array}{rcl} Ker(\partial_p) & = & Z_p \\ Im(\partial_{p+1}) & = & B_p \end{array} \right. \qquad \begin{array}{l} \partial_p \partial_{p+1} = 0 \Rightarrow B_p \subseteq Z_p \\ H_p = Z_p / B_p \end{array}$$

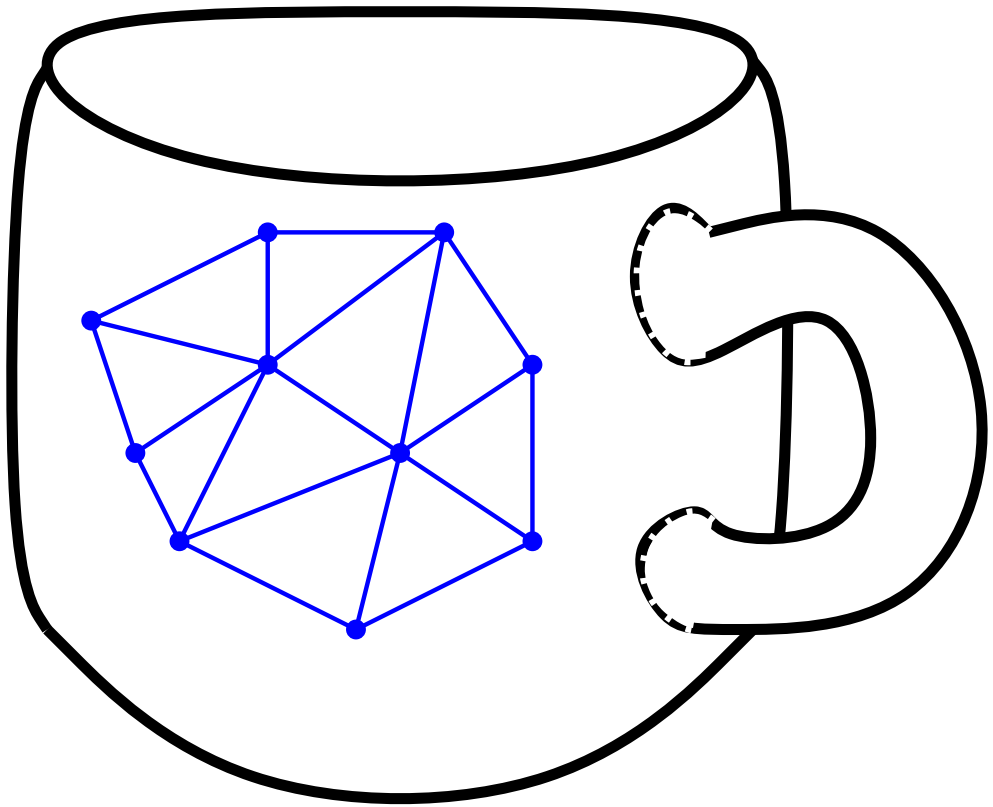
Really Short Introduction to Persistent Homology with $\mathbb{Z}_2$ Coefficients

$$\begin{cases} Ker(\partial_p) = Z_p \\ Im(\partial_{p+1}) = B_p \end{cases} \qquad \partial_p \partial_{p+1} = 0 \Rightarrow B_p \subseteq Z_p \qquad H_p = Z_p / B_p$$



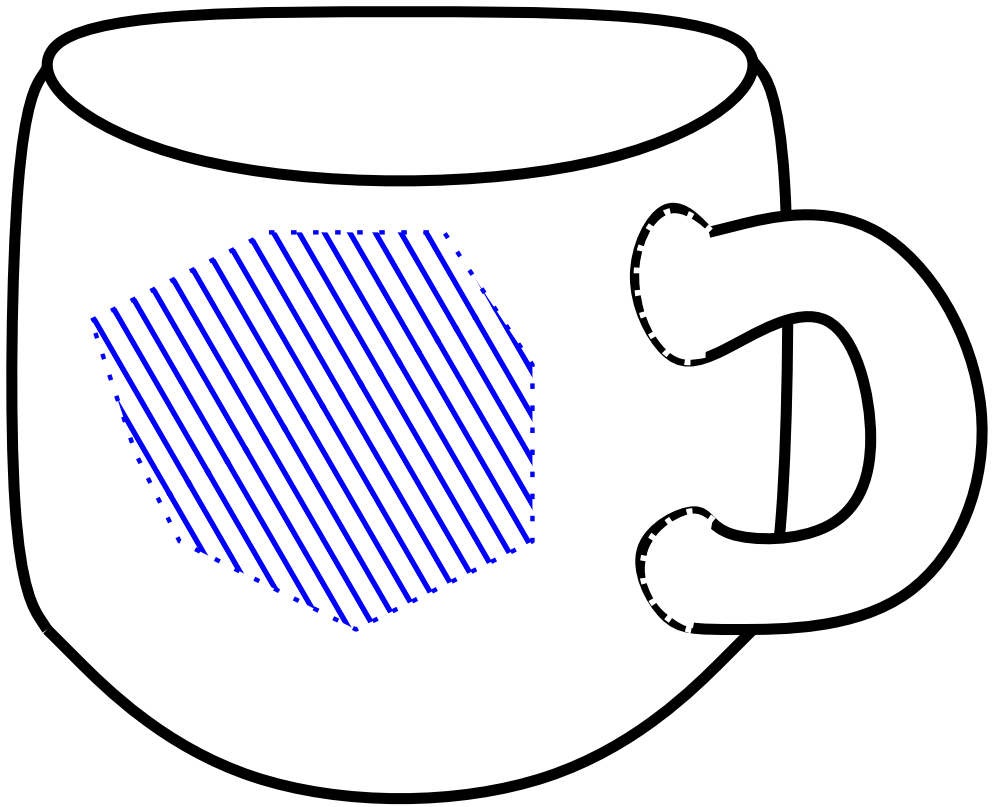
Really Short Introduction to Persistent Homology with $\mathbb{Z}_2$ Coefficients

$$\begin{cases} Ker(\partial_p) = Z_p \\ Im(\partial_{p+1}) = B_p \end{cases} \qquad \partial_p \partial_{p+1} = 0 \Rightarrow B_p \subseteq Z_p \qquad H_p = Z_p / B_p$$



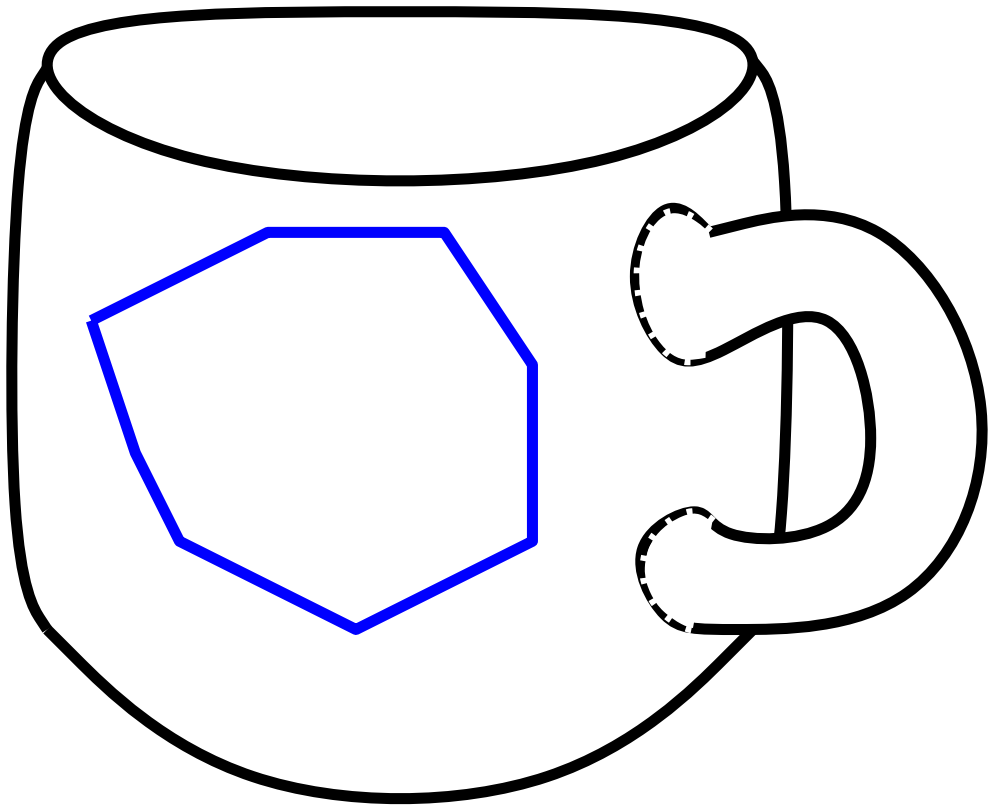
Really Short Introduction to Persistent Homology with $\mathbb{Z}_2$ Coefficients

$$\left\{ \begin{array}{rcl} Ker(\partial_p) & = & Z_p \\ Im(\partial_{p+1}) & = & B_p \end{array} \right. \qquad \partial_p \partial_{p+1} = 0 \quad \Rightarrow \quad B_p \subseteq Z_p \qquad H_p = Z_p / B_p$$



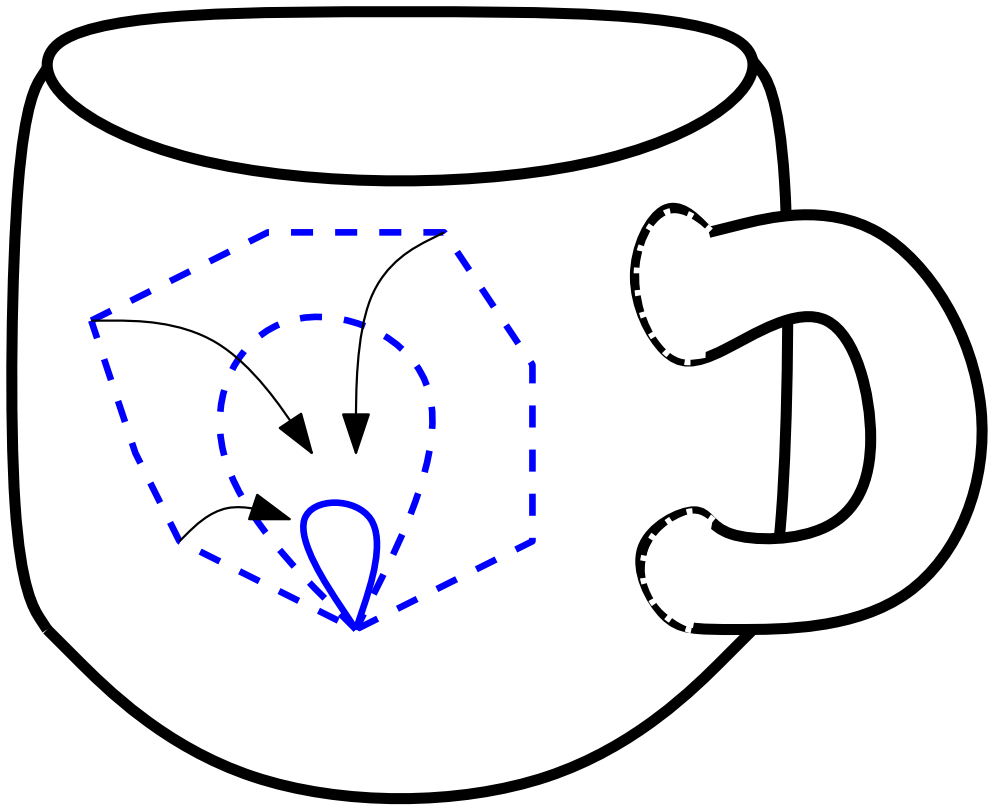
Really Short Introduction to Persistent Homology with $\mathbb{Z}_2$ Coefficients

$$\begin{cases} Ker(\partial_p) = Z_p \\ Im(\partial_{p+1}) = B_p \end{cases} \qquad \partial_p \partial_{p+1} = 0 \Rightarrow B_p \subseteq Z_p \qquad H_p = Z_p / B_p$$



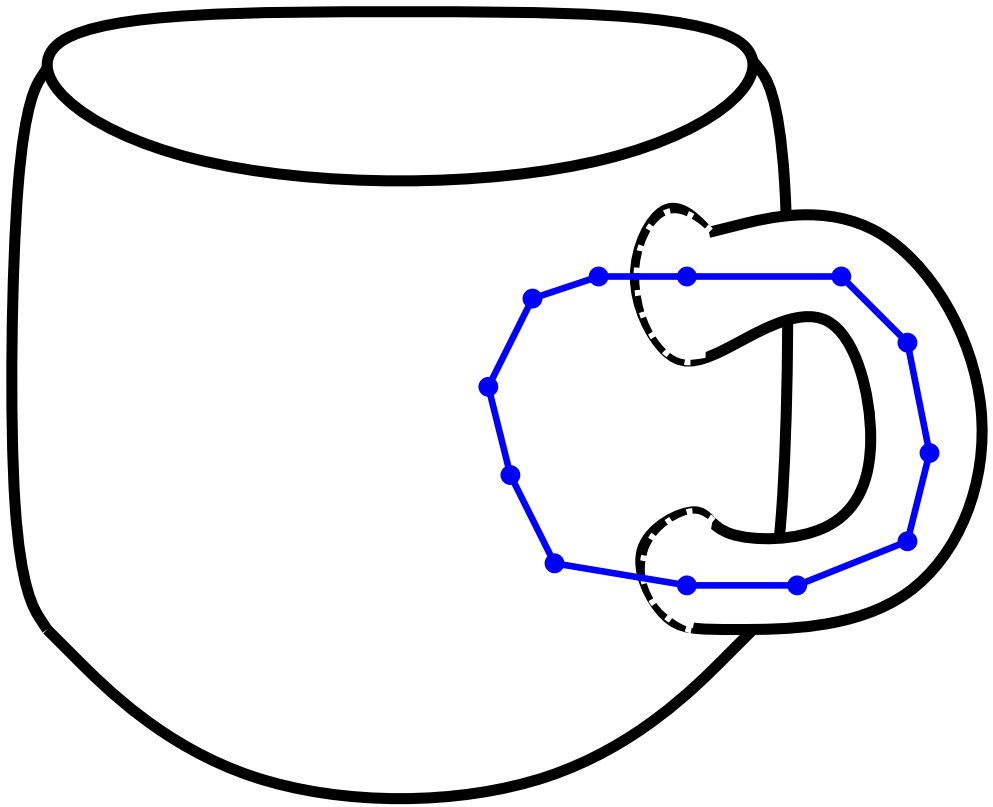
Really Short Introduction to Persistent Homology with $\mathbb{Z}_2$ Coefficients

$$\begin{cases} Ker(\partial_p) = Z_p \\ Im(\partial_{p+1}) = B_p \end{cases} \qquad \partial_p \partial_{p+1} = 0 \Rightarrow B_p \subseteq Z_p \qquad H_p = Z_p / B_p$$



Really Short Introduction to Persistent Homology with $\mathbb{Z}_2$ Coefficients

$$\begin{cases} Ker(\partial_p) = Z_p \\ Im(\partial_{p+1}) = B_p \end{cases} \qquad \partial_p \partial_{p+1} = 0 \Rightarrow B_p \subseteq Z_p \qquad H_p = Z_p / B_p$$

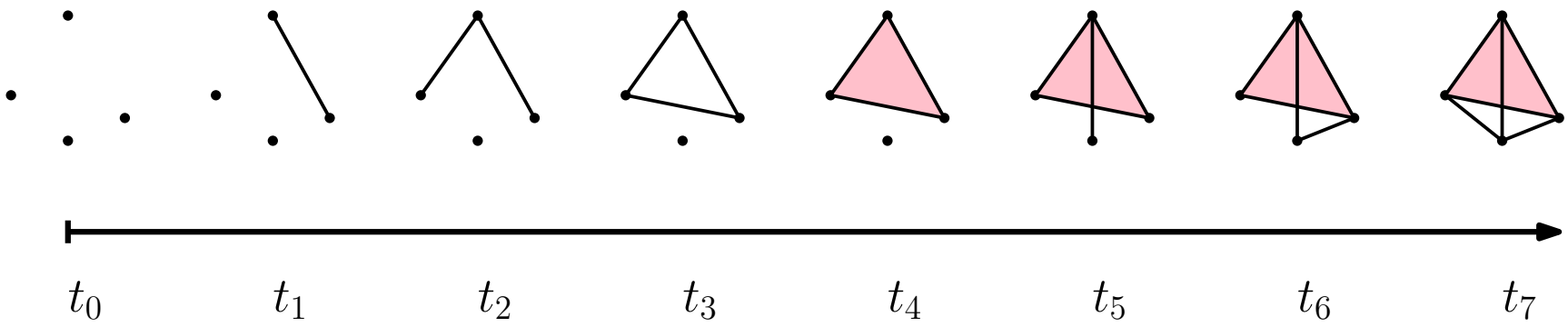


Filtration of a Simplicial Complex

A *filtration* of a simplicial complex $\mathcal{K}$ is defined by an indexed family of simplicial complexes $(\mathcal{K}_{t_i})_{i=0\cdots n}$ such that:

$$\emptyset = \mathcal{K}_{t_0} \subset \mathcal{K}_{t_1} \subset \cdots \subset \mathcal{K}_{t_{n-1}} \subset \mathcal{K}_{t_n} = \mathcal{K}$$

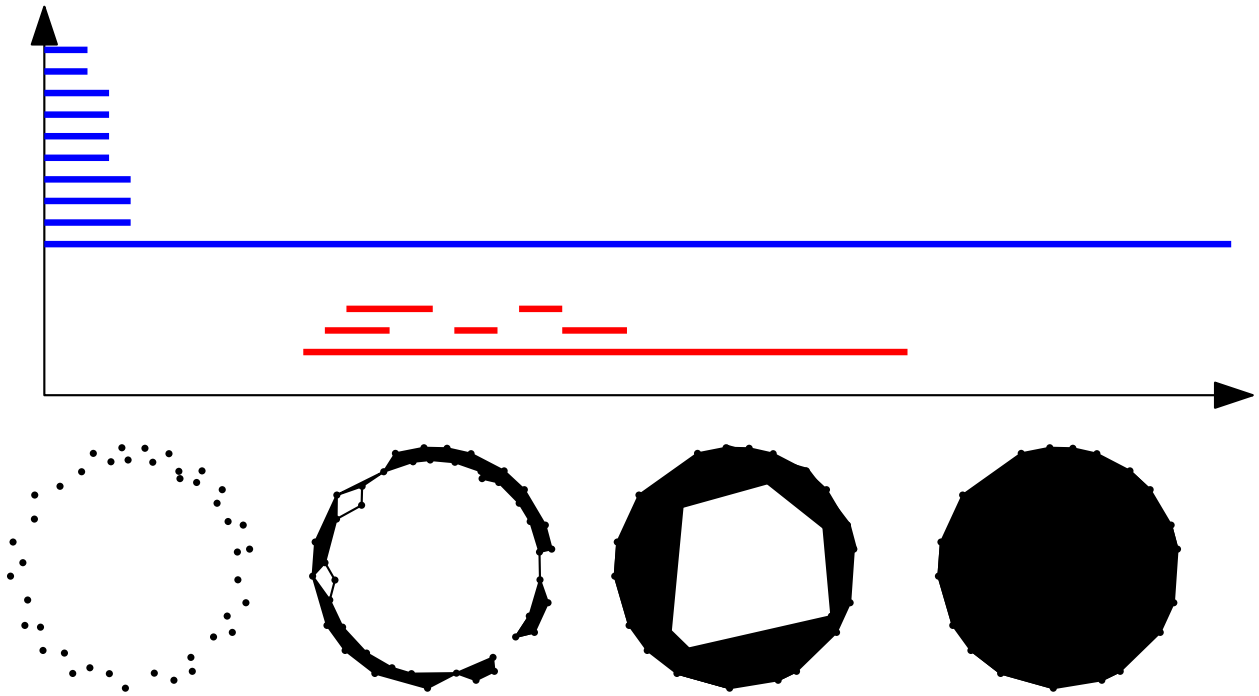
We can attach to each face of $\mathcal{K}$ a *time of appearance* t_i.



Incremental Algorithm: $\sigma = \mathcal{K}_i \setminus \mathcal{K}_{i-1}, \dim(\sigma) = p$
If σ belongs to a p-cycle **then** $\beta_p + +$
else $\beta_{p-1} - -$

Persistence Diagram

Incremental Algorithm: $\sigma = \mathcal{K}_i \setminus \mathcal{K}_{i-1}, \dim(\sigma) = p$
If σ belongs to a p-cycle **then** $\beta_p ++$
else $\beta_{p-1} --$



Implementation for Computing Persistent Homology

Focus on simplices:

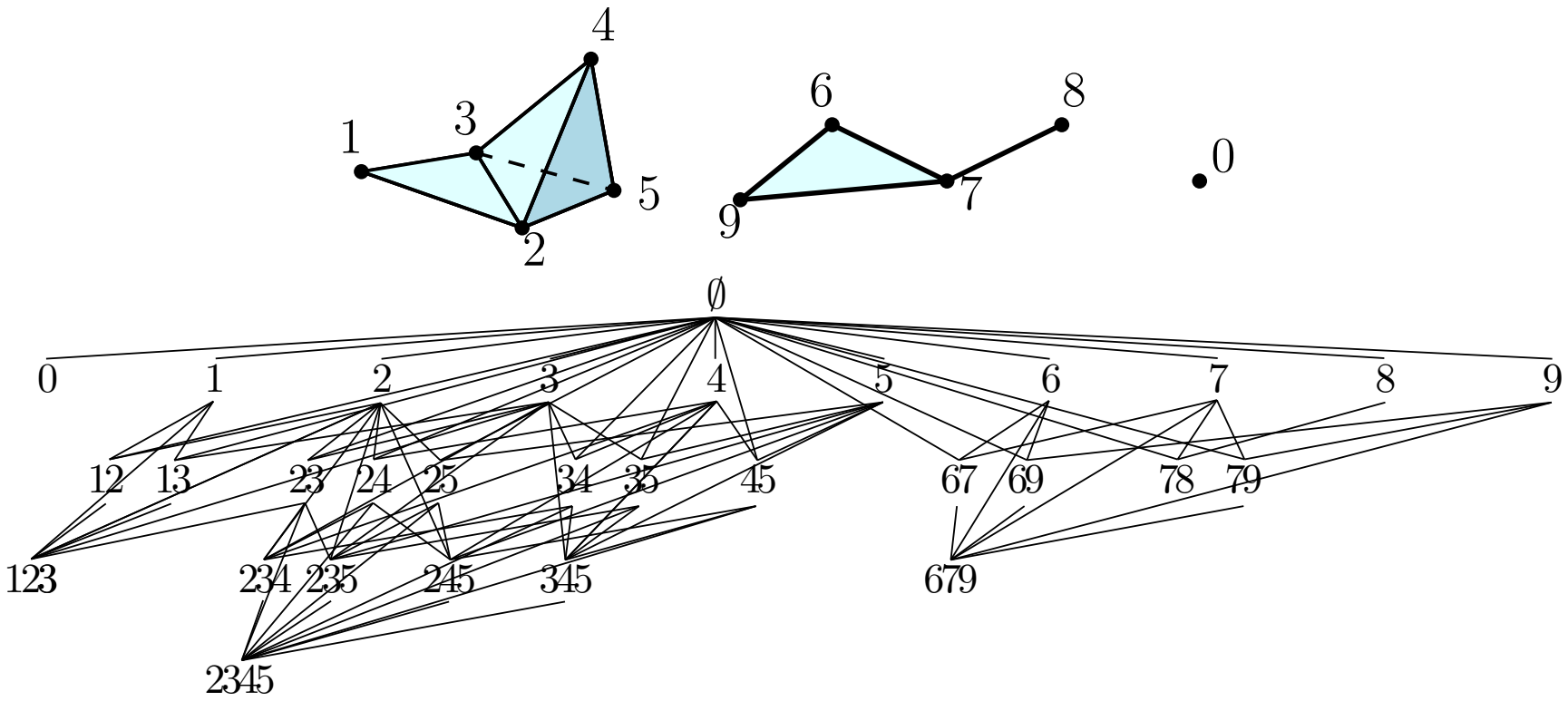
1. Represent the filtered simplicial complex:
   - Represent the simplices and their time of appearance
   - Retrieve incidence relation (compute $\partial\sigma$)
2. Attach topological information to simplices
   - Check if a new simplex is in a cycle or not
   - Update the topological information under insertion of a simplex
3. Interface the two structures

2
The Simplex Tree
[Boissonnat, M. '12]

an Efficient Data Structure for General Simplicial Complexes

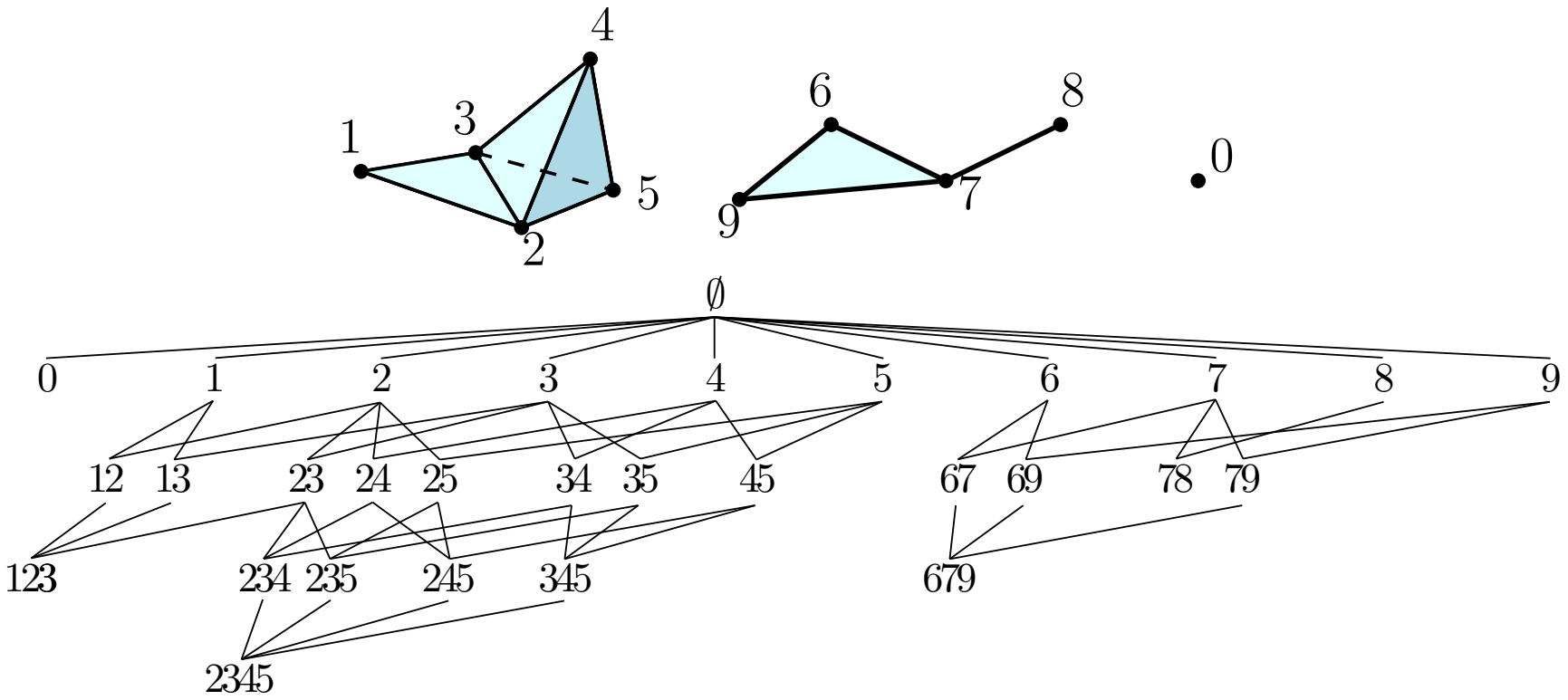
The Simplex Tree

A simplicial complex is a partially ordered set with regard to inclusion. All the incidences are represented in the following:



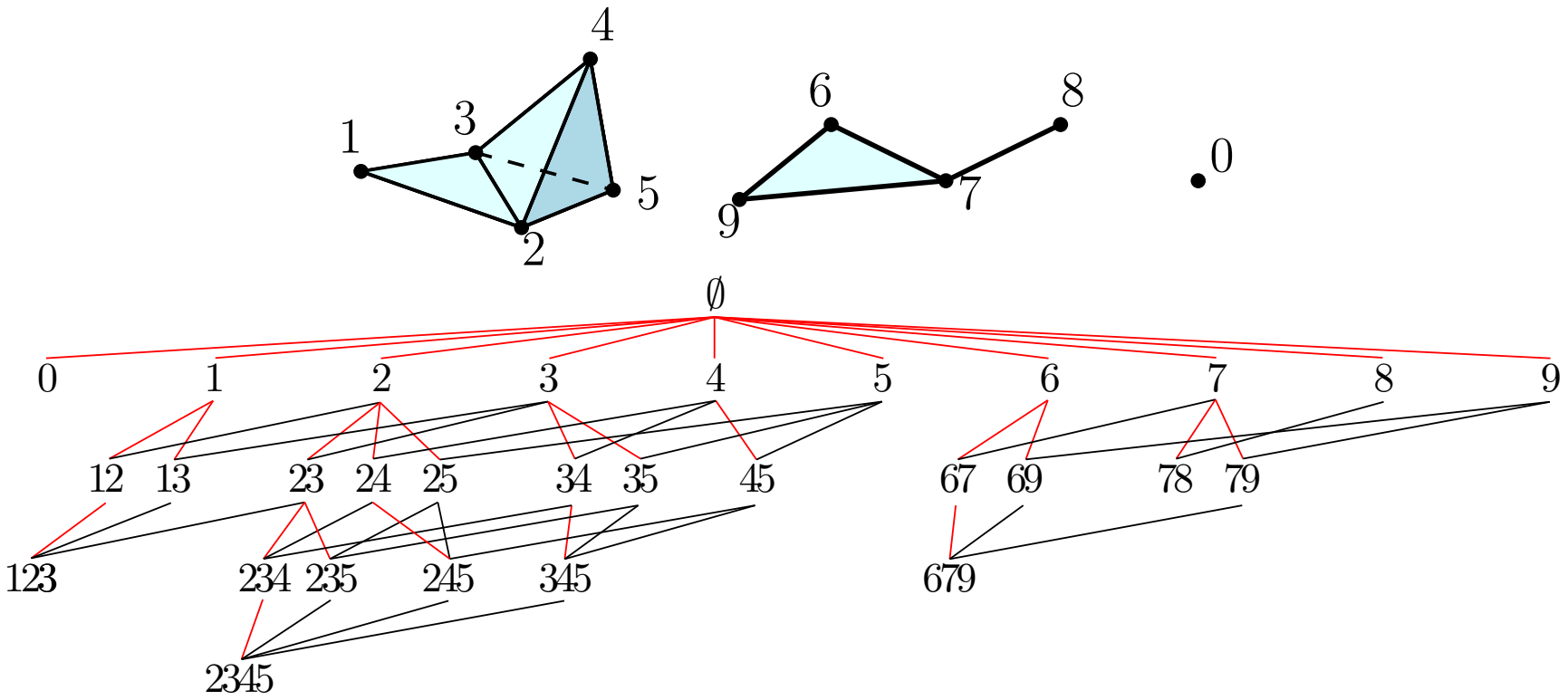
The Simplex Tree

Only incidences from an element to the smallest bigger elements are represented in the Hasse diagram:



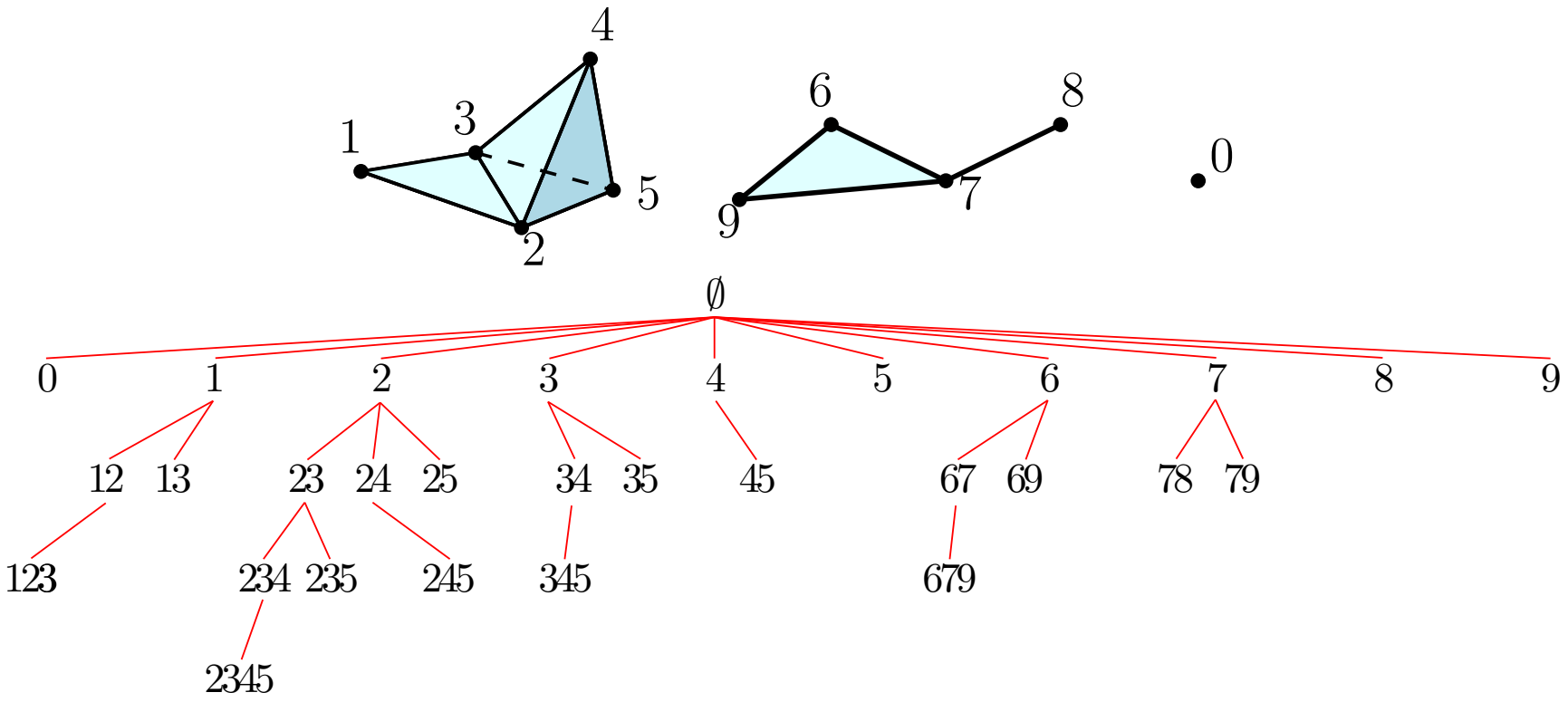
The Simplex Tree

We select a specific spanning tree of the Hasse diagram. The choosen incidences respect also the lexicographic order:



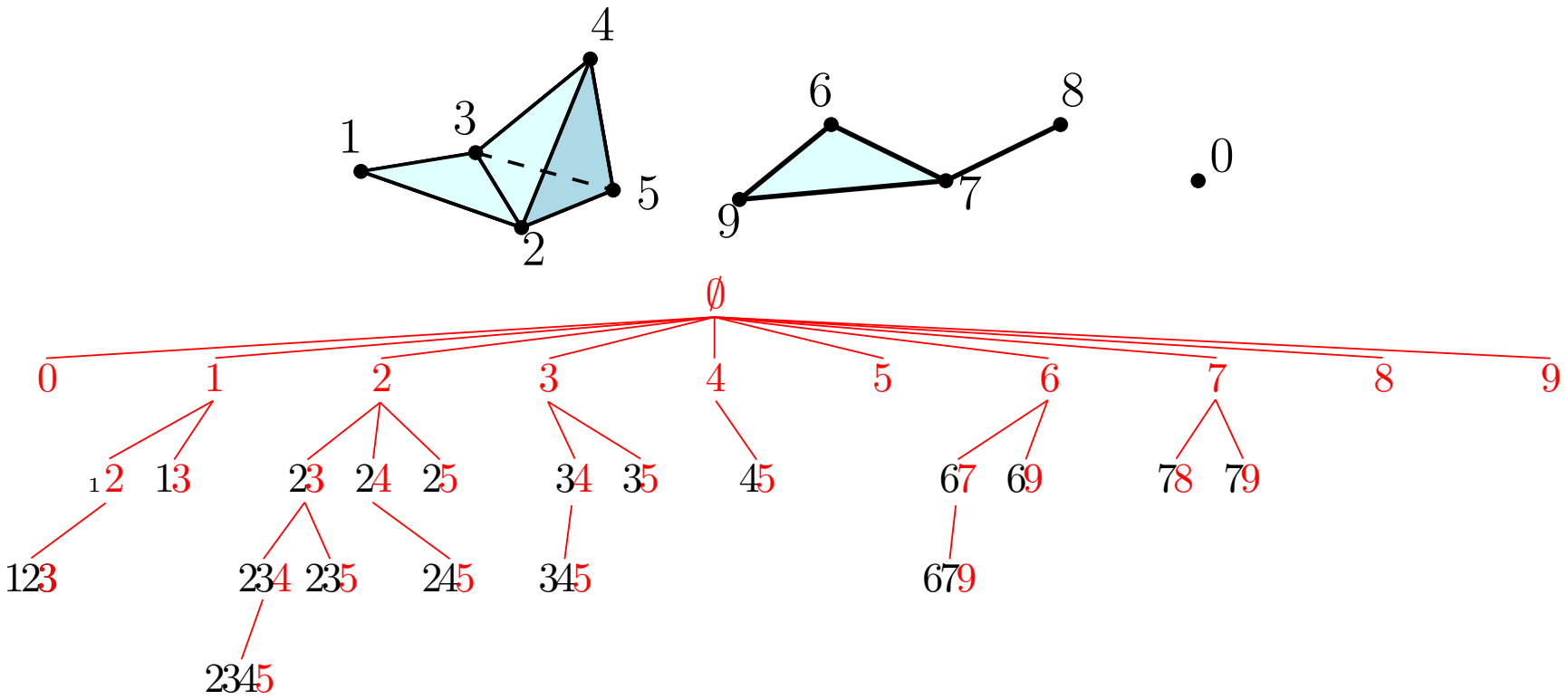
The Simplex Tree

We select a specific spanning tree of the Hasse diagram. The choosen incidences respect also the lexicographic order:



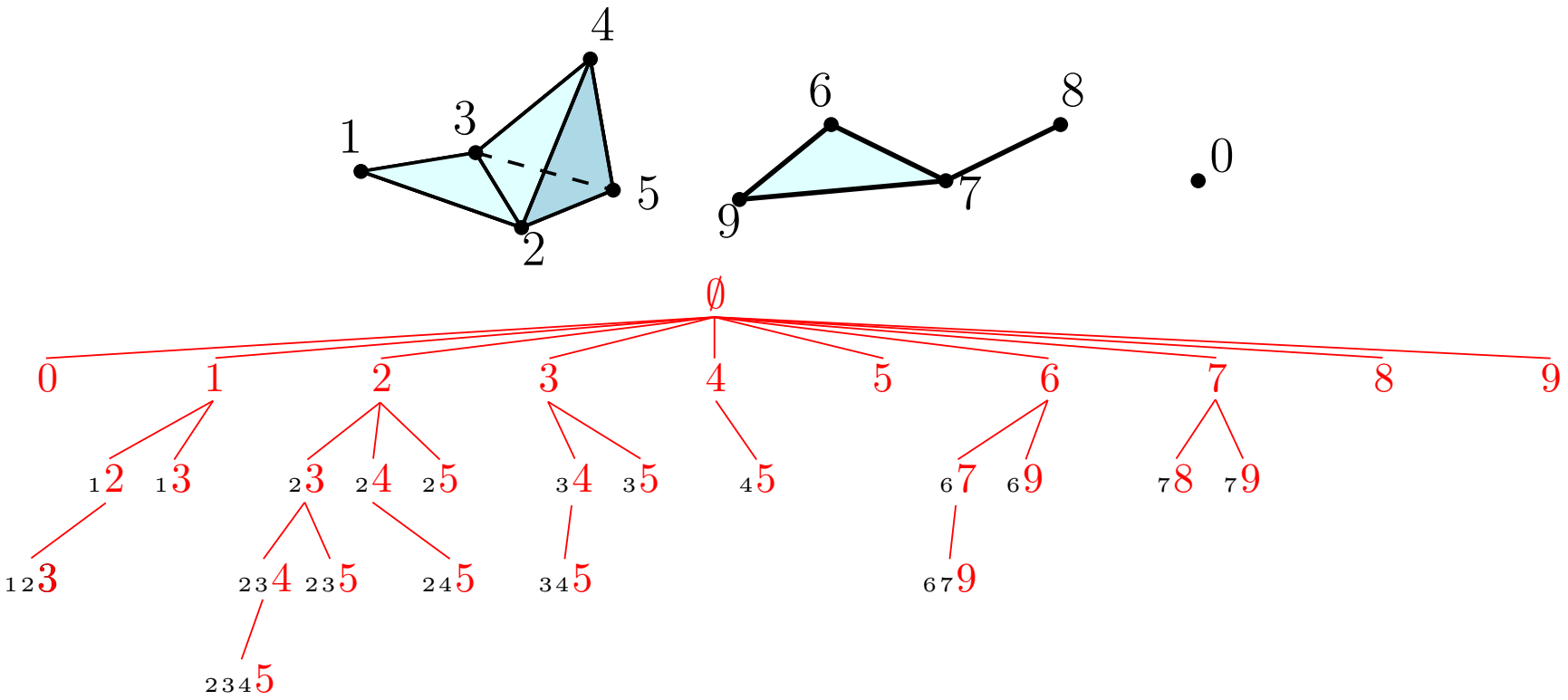
The Simplex Tree

We keep only the biggest vertex in each simplex. The vertices of a simplex are encountered in the path from the root to its node:



The Simplex Tree

We keep only the biggest vertex in each simplex. The vertices of a simplex are encountered in the path from the root to its node:

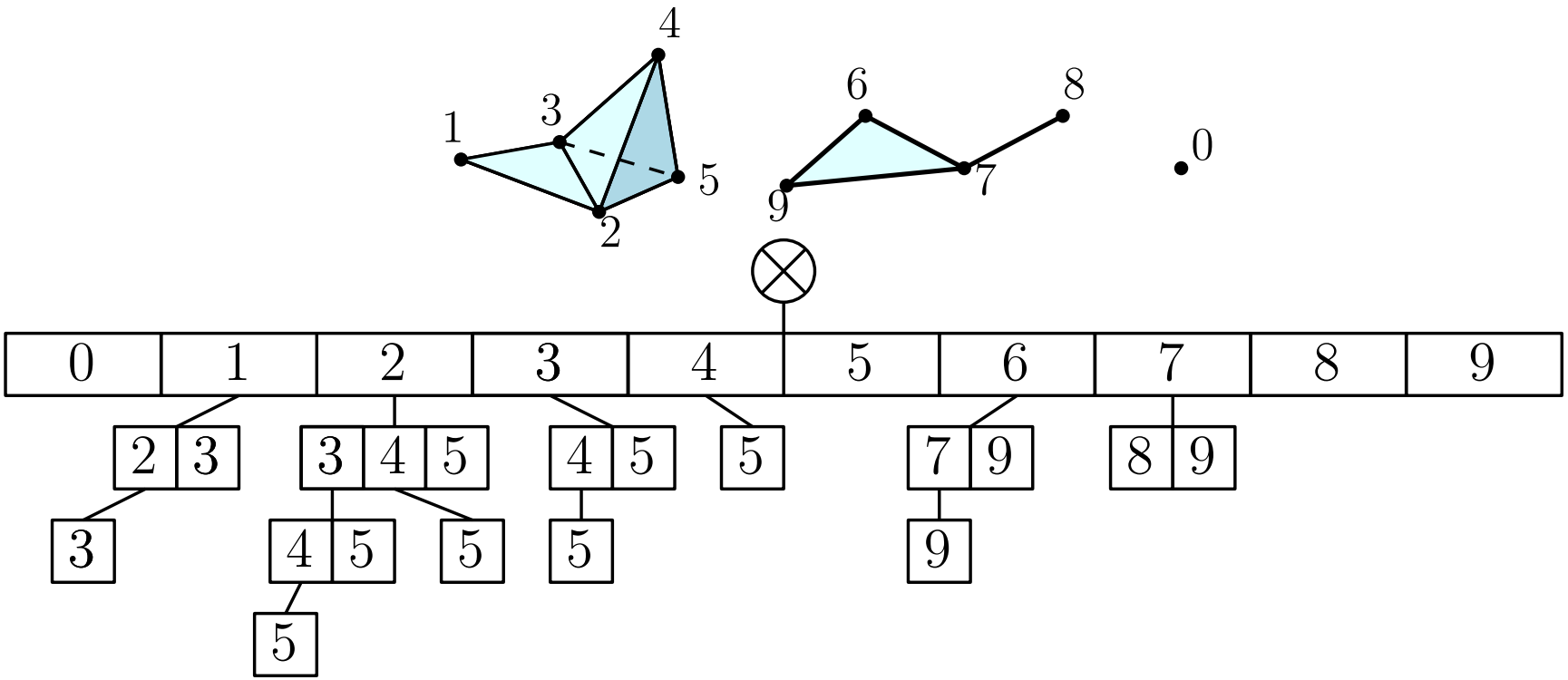


The Simplex Tree

[Boissonnat, M. '12]

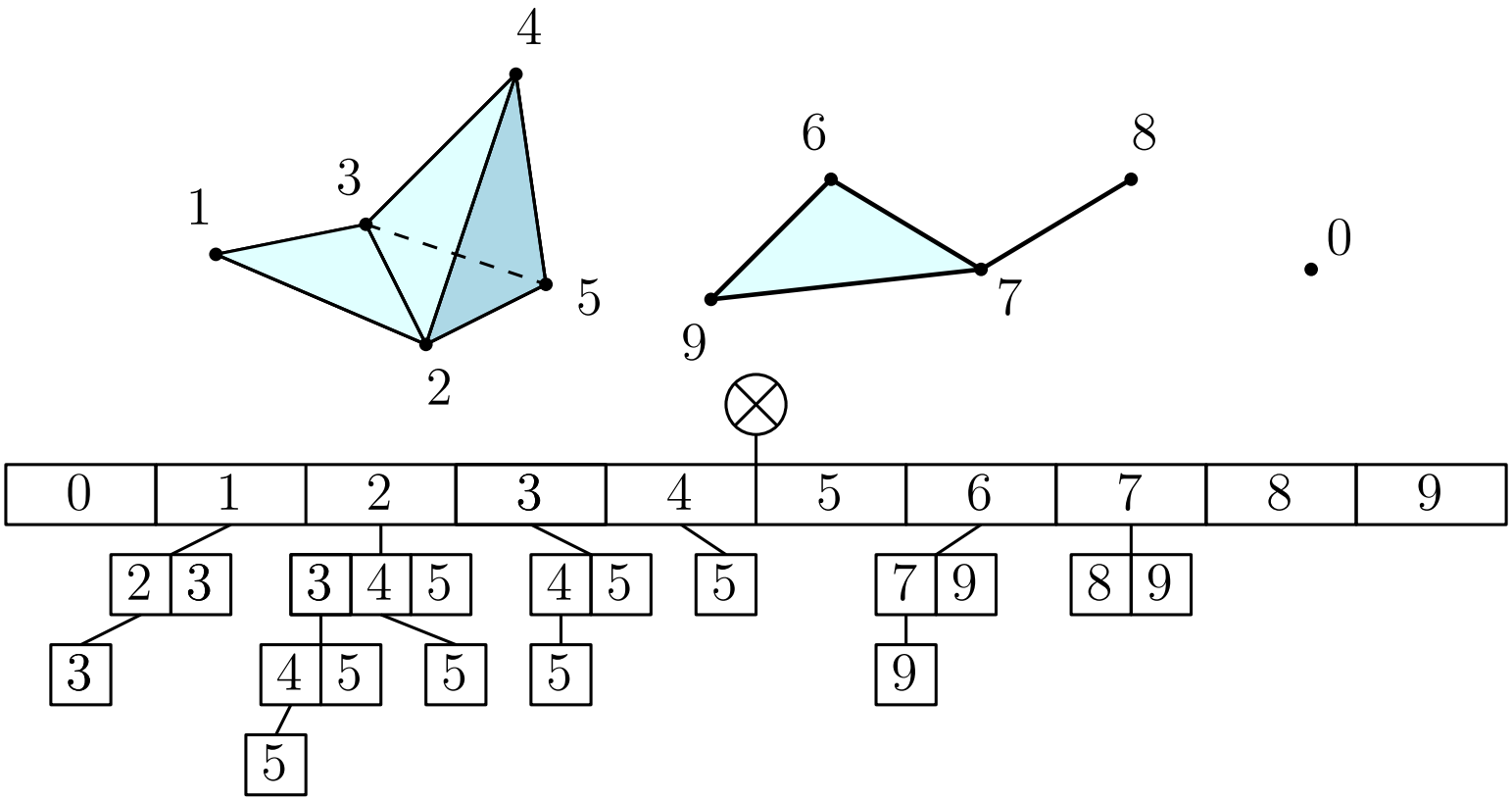
We call this structure a *Simplex Tree*.

Memory complexity: $O(1)$ per simplex.



The Simplex Tree

EQUIVALENTLY: to each simplex $\sigma \in \mathcal{K}$ we associate the ordered list of its vertices, and store it in a *trie*.

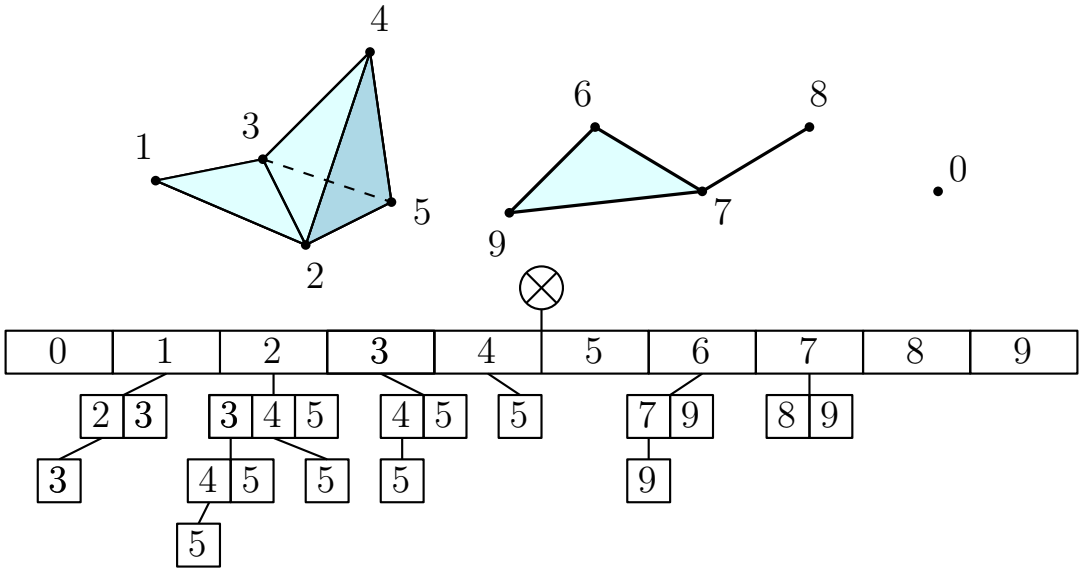


The Simplex Tree

Memory complexity: $O(1)$ per simplex.

Some properties:

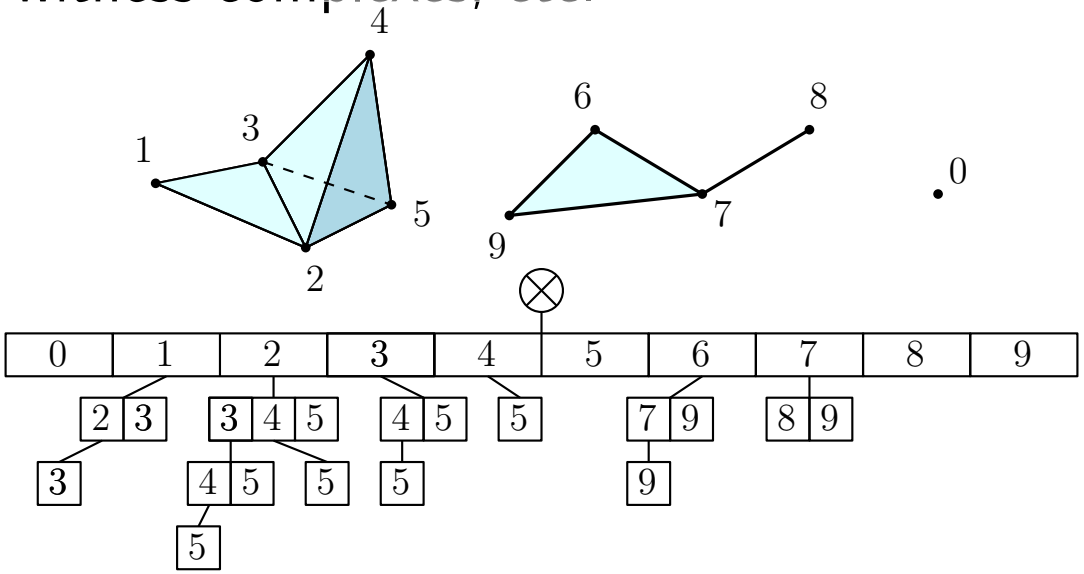
Simplex Tree		Simplicial Complex
#nodes	$=$	$\#\mathcal{K}$
depth	$=$	$\dim(\mathcal{K})+1$
#children(σ)	$\leq$	$\deg(\sigma)$



Simplex tree: Atomic operations

The Simplex Tree allows to compute efficiently:

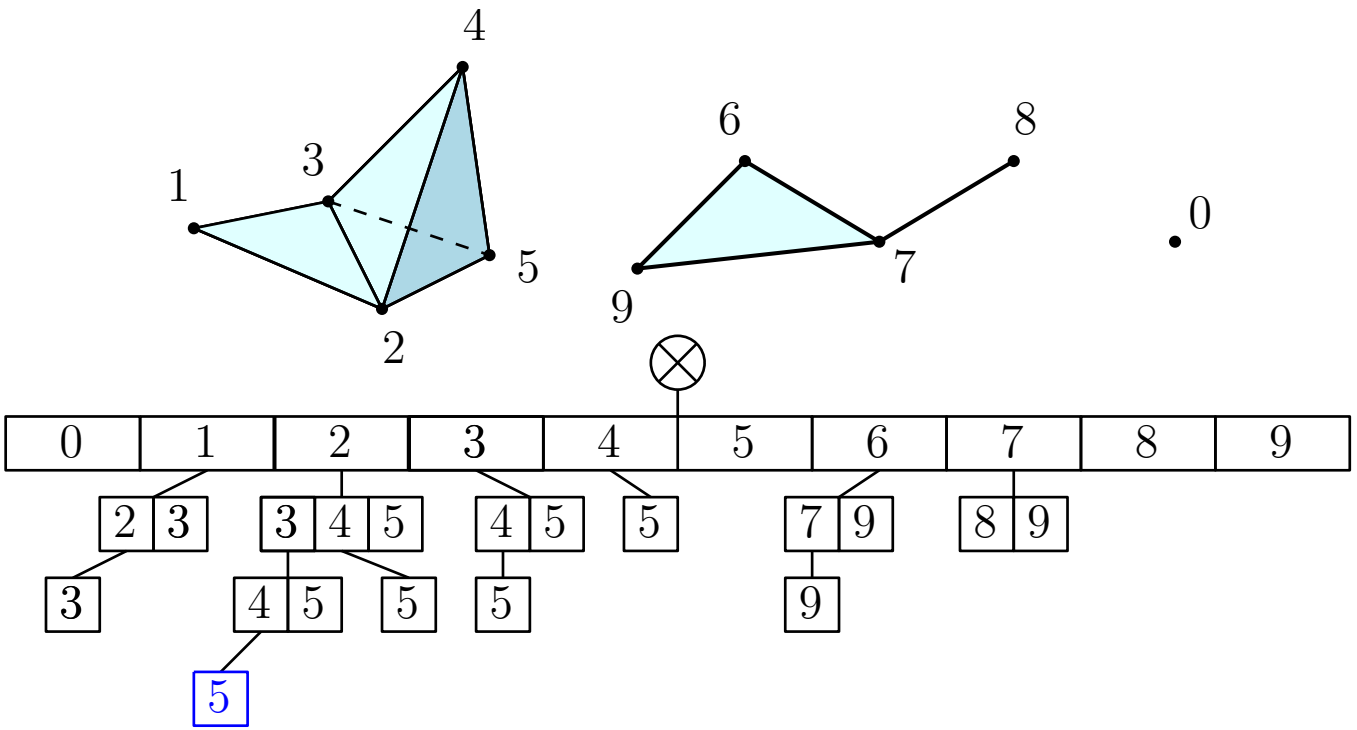
- **Look-up/Insertion/Deletion** of a simplex
- The **facets** and **subfaces** of a simplex
- The **cofaces** of a simplex
- **Edge contractions**
- **Elementary collapses**
- **Fast construction** of Flag complexes, witness complexes, relaxed witness complexes, etc.



Facets/Boundary Computation

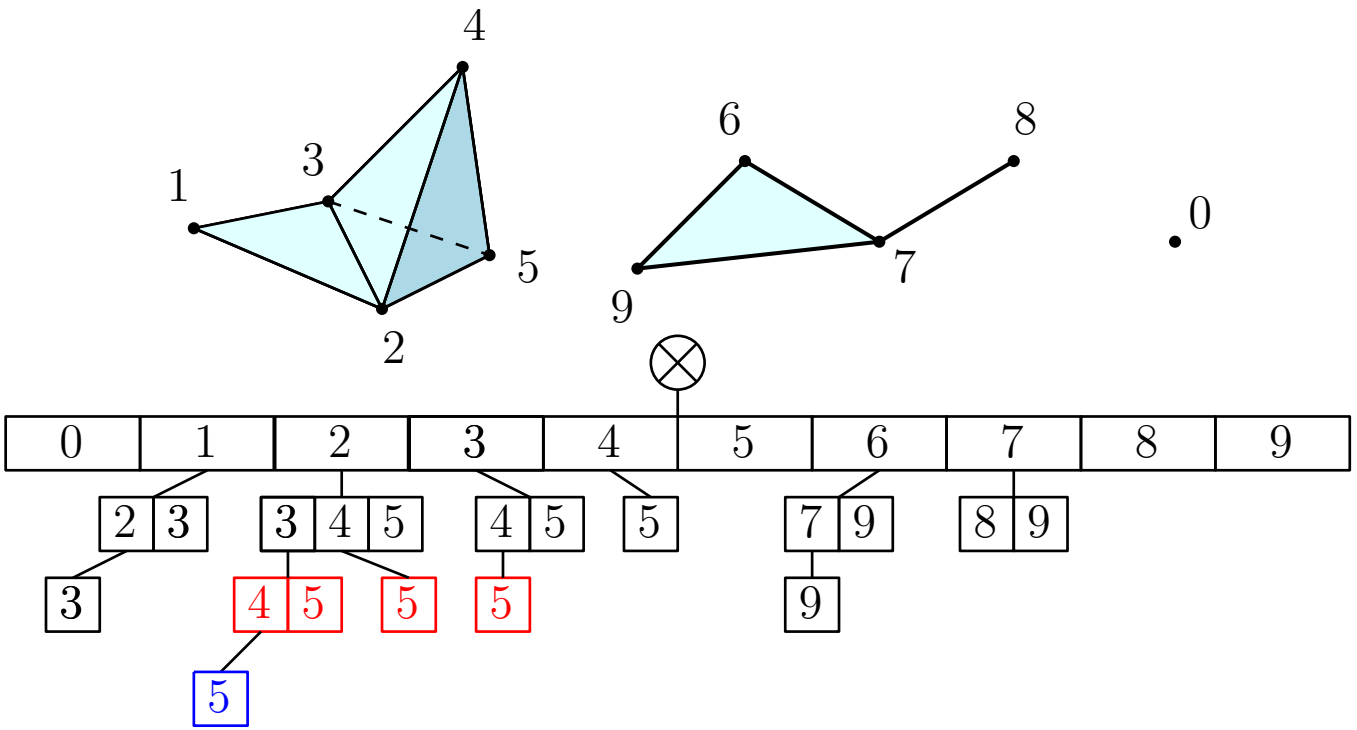
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



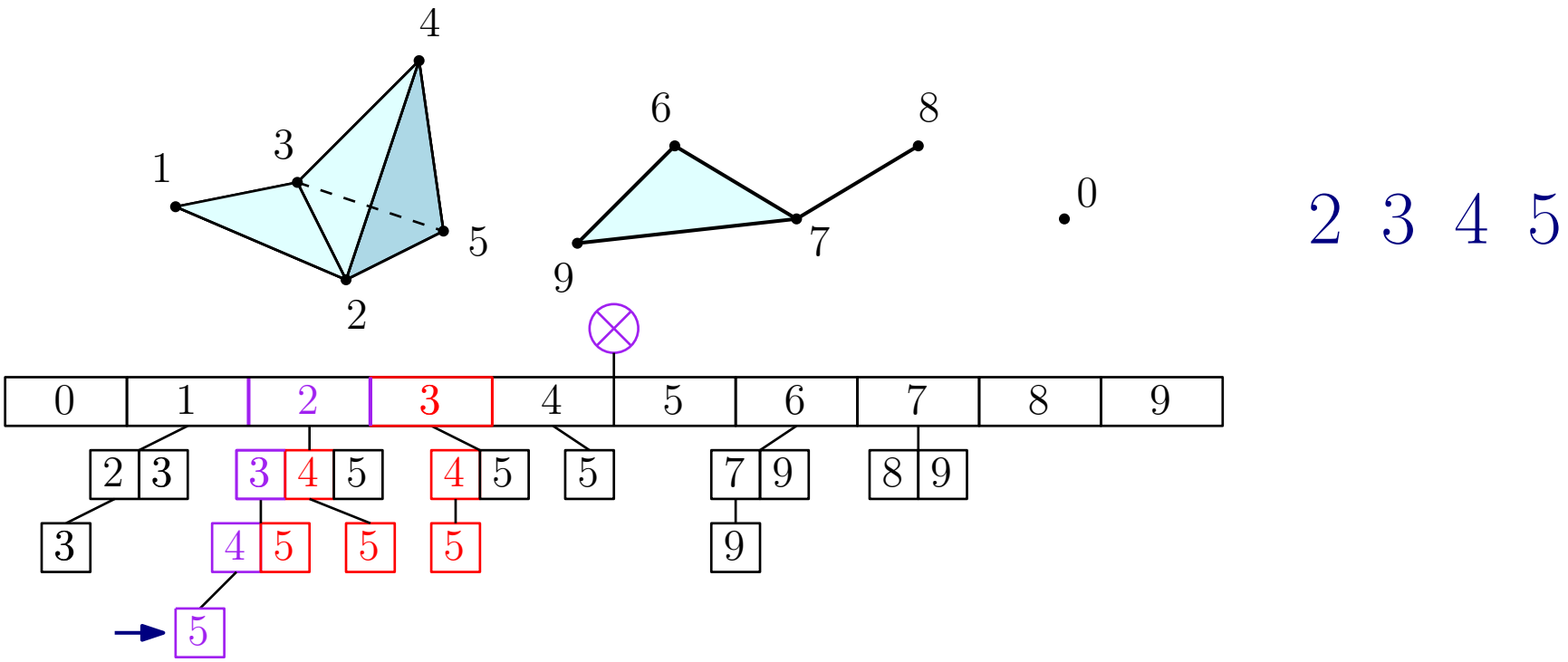
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



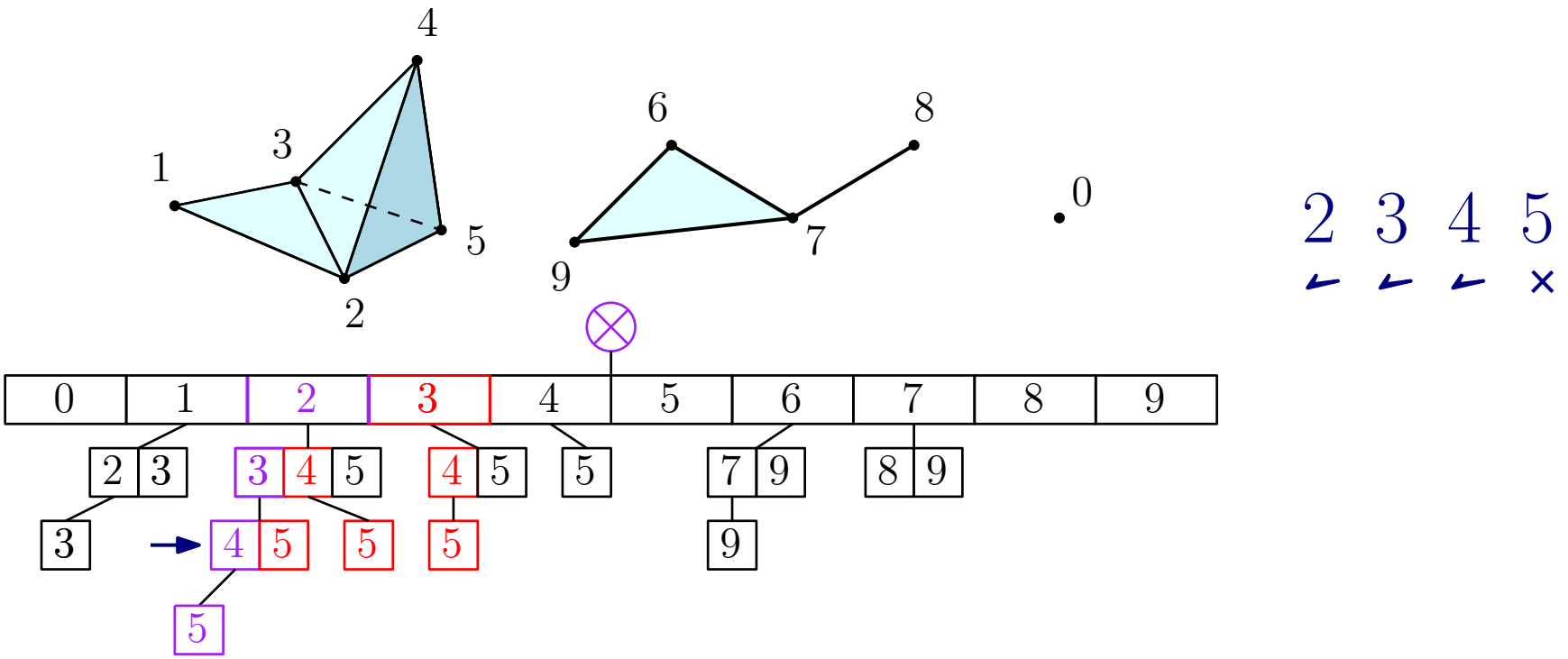
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



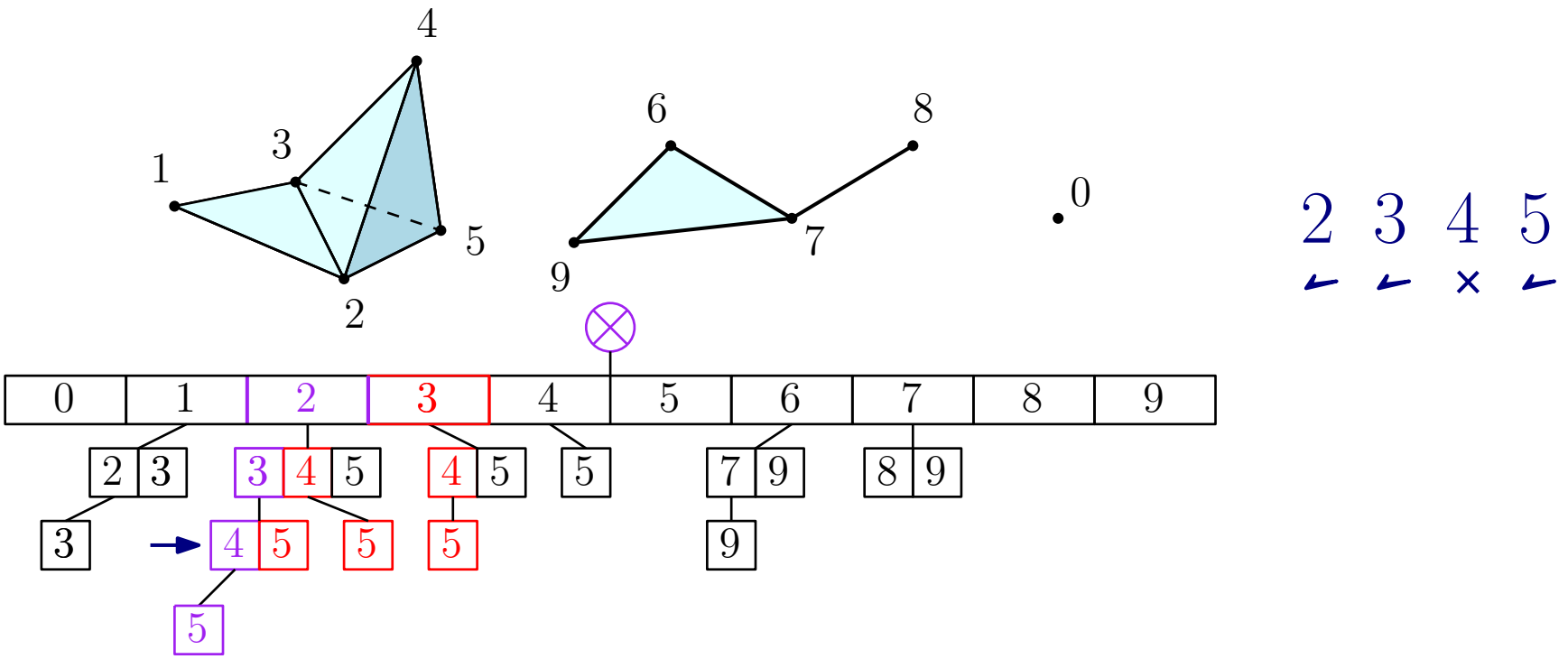
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



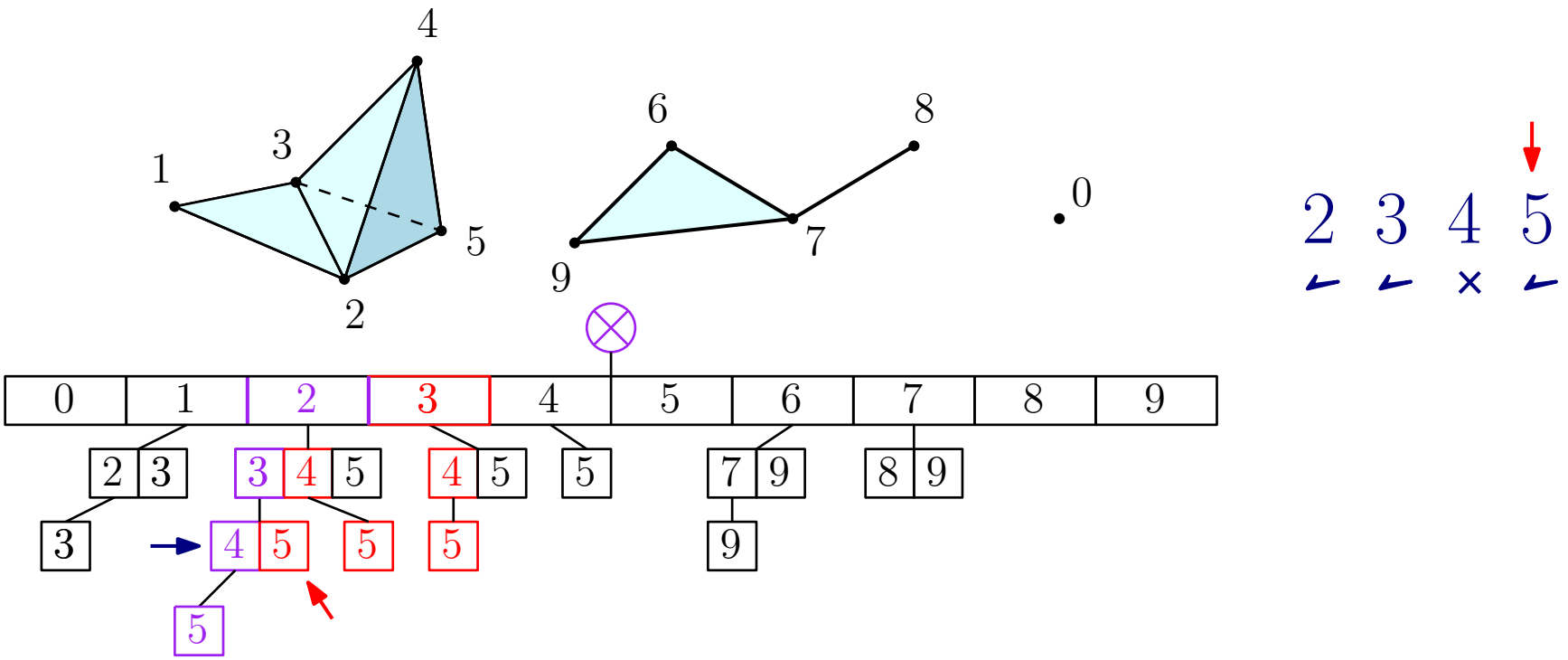
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



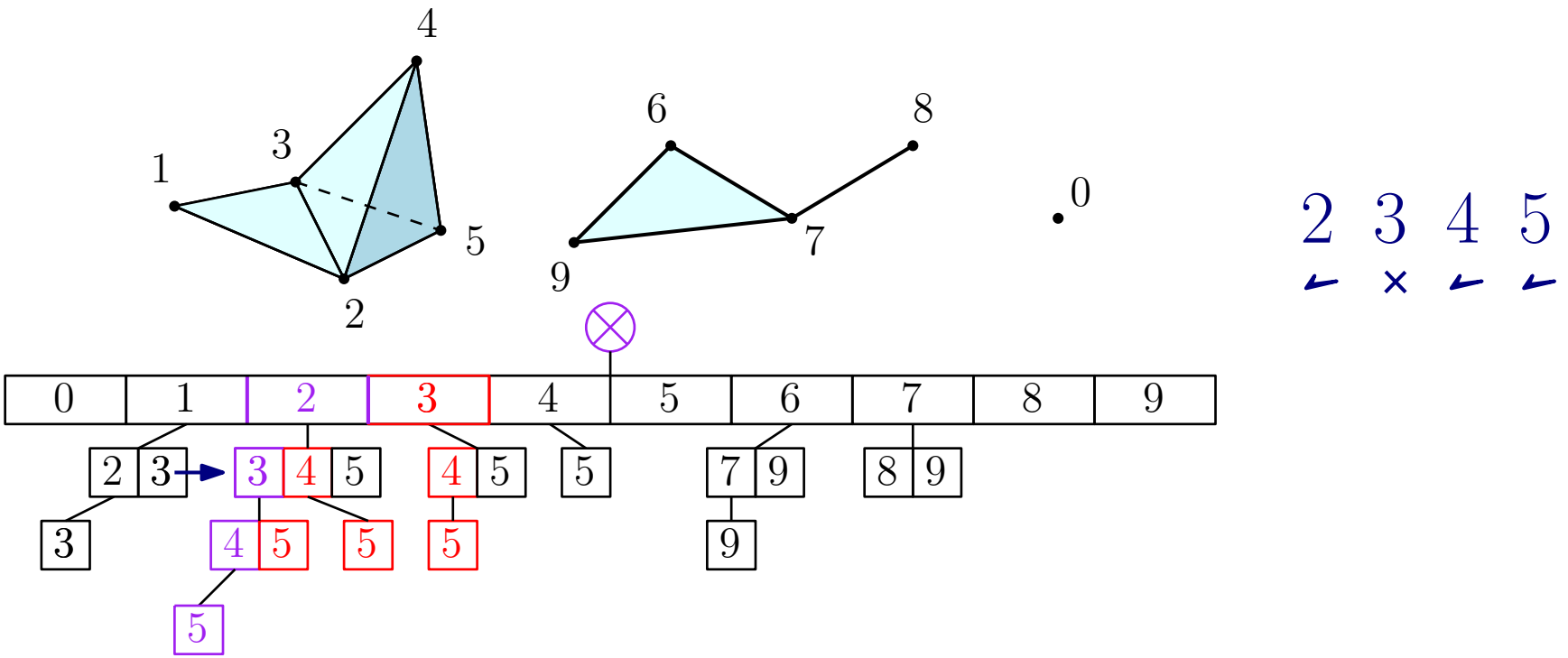
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



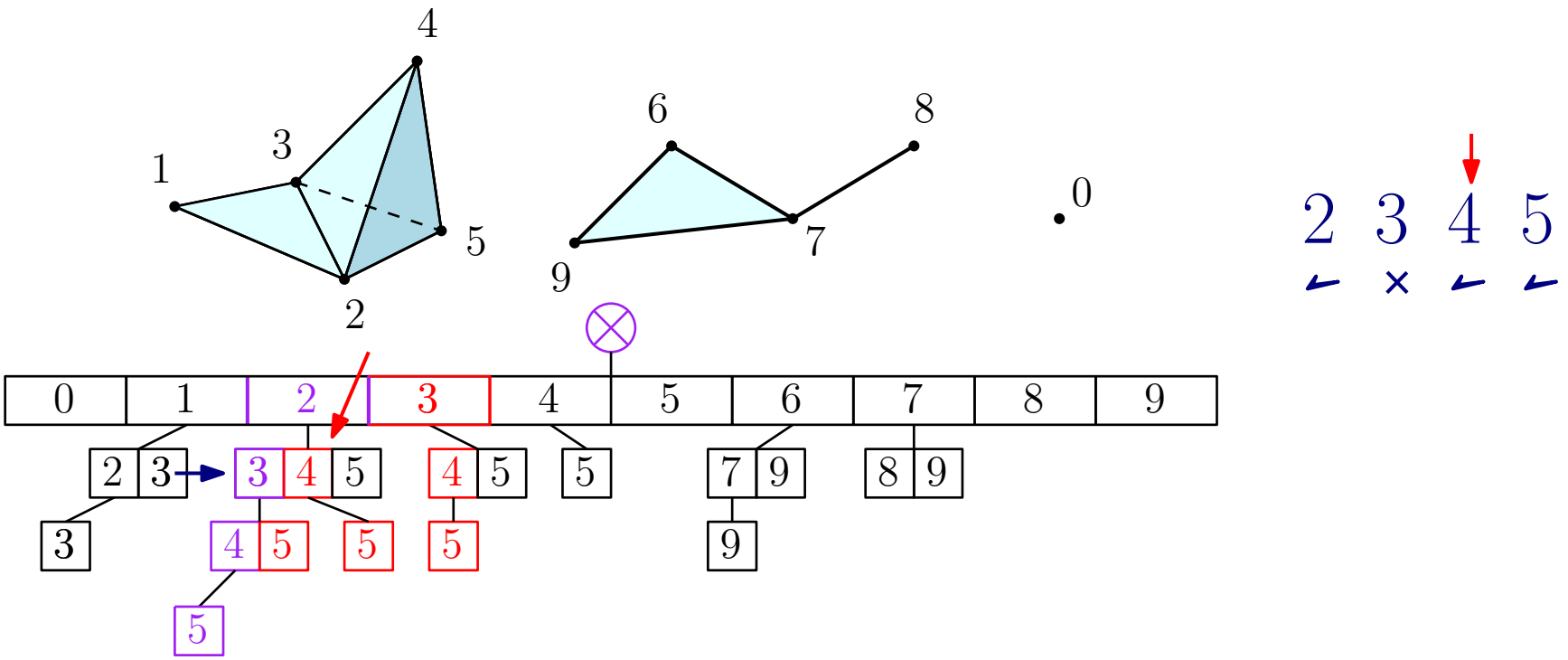
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



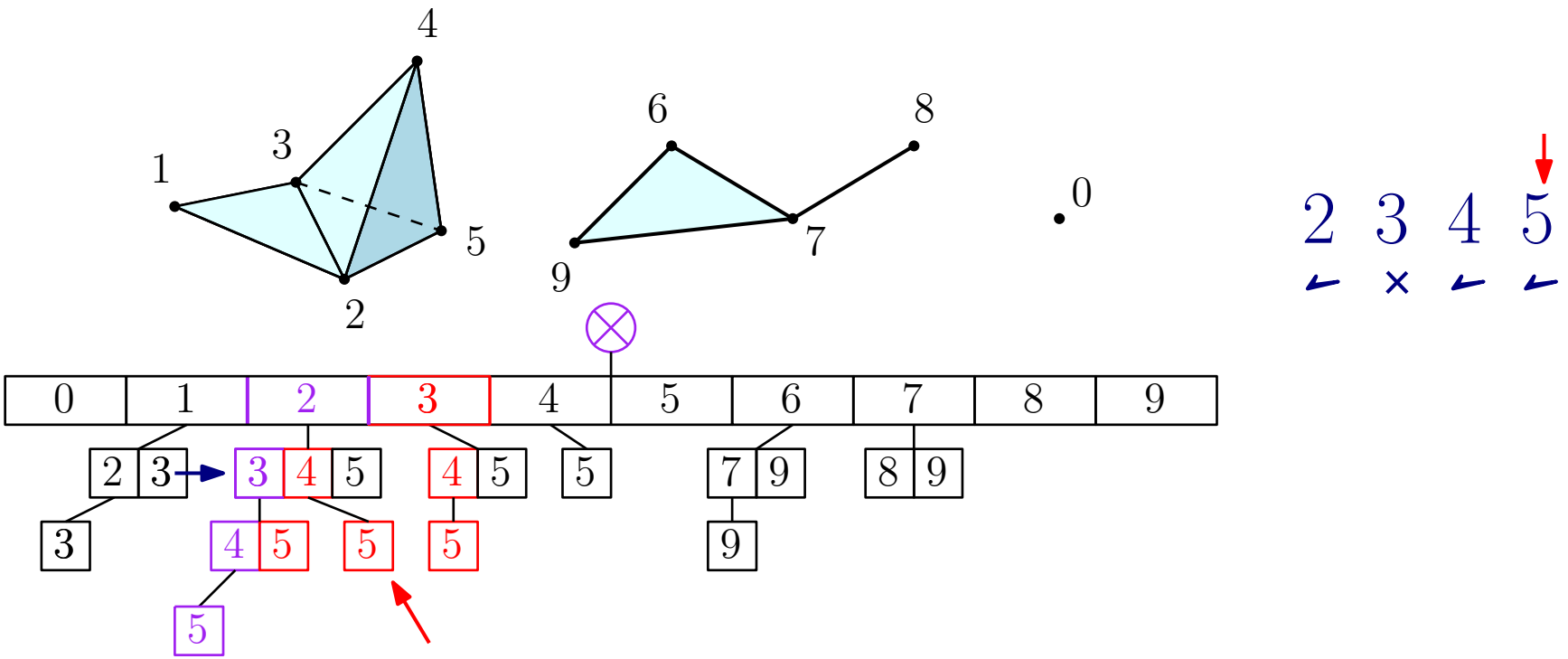
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



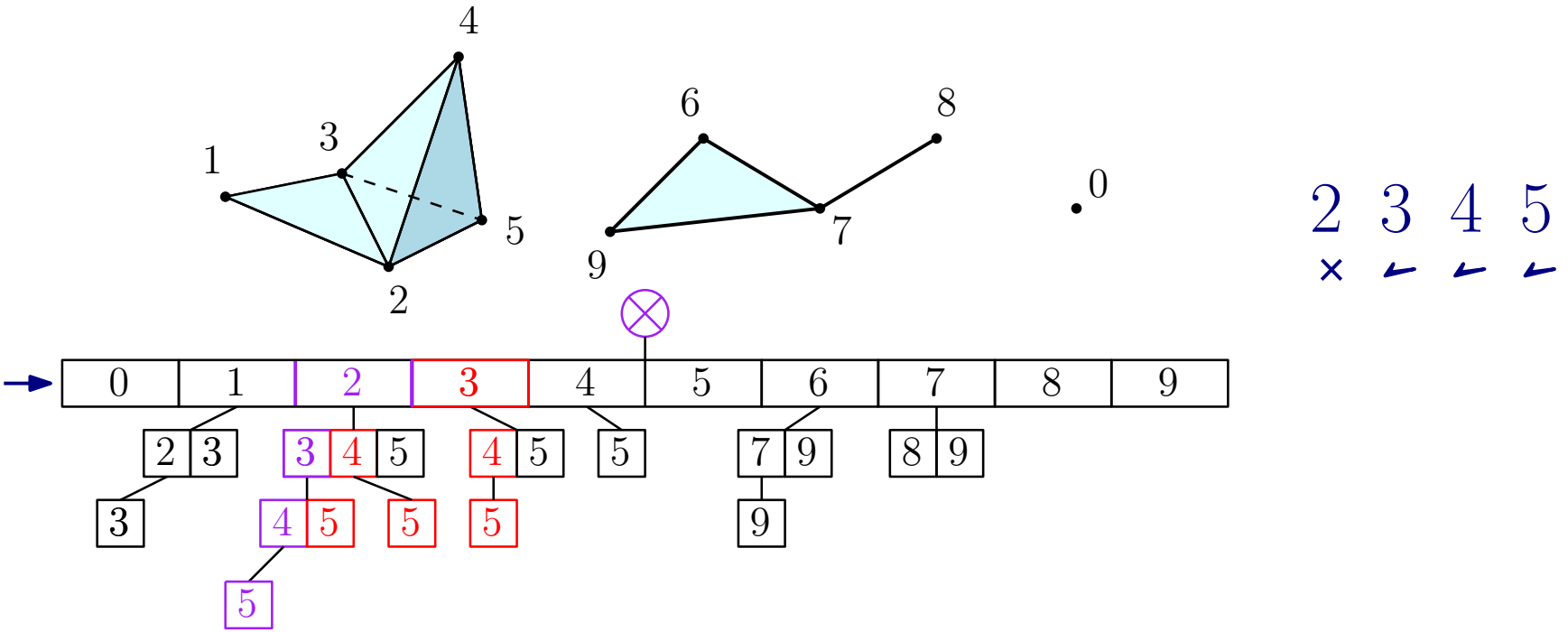
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



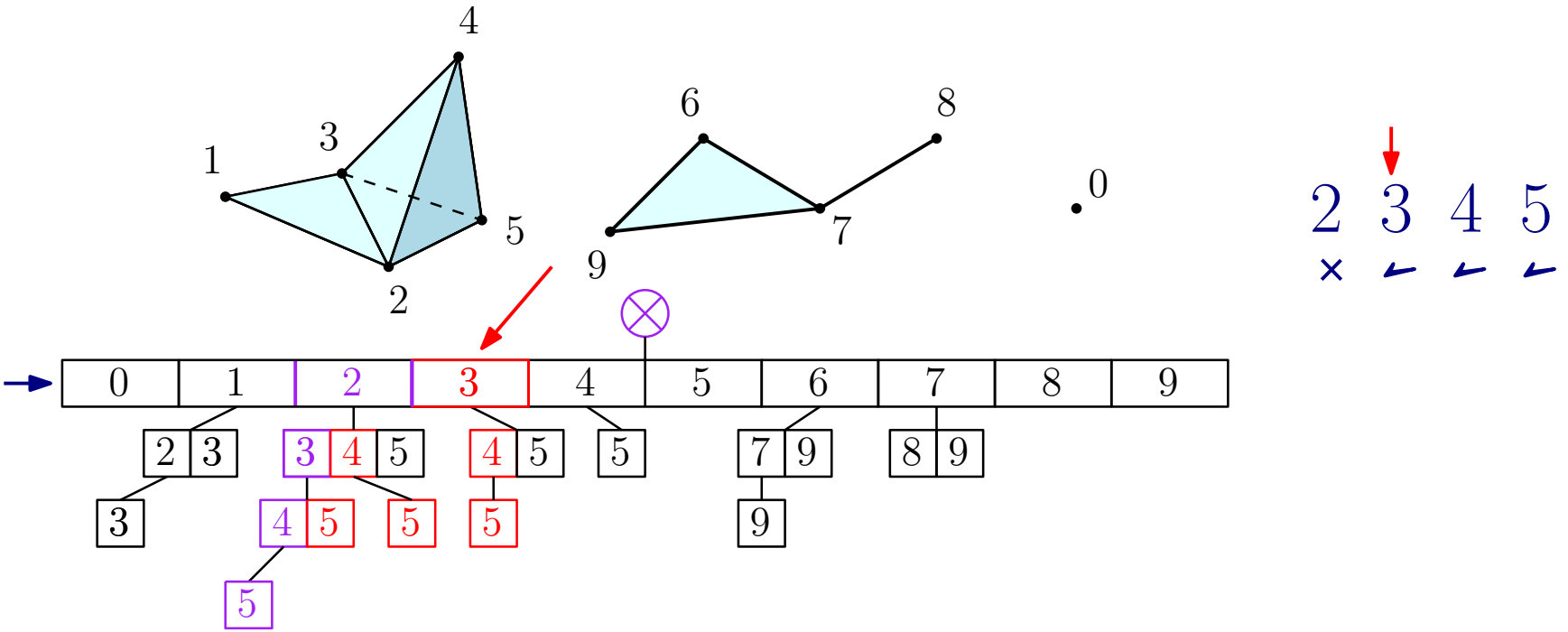
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



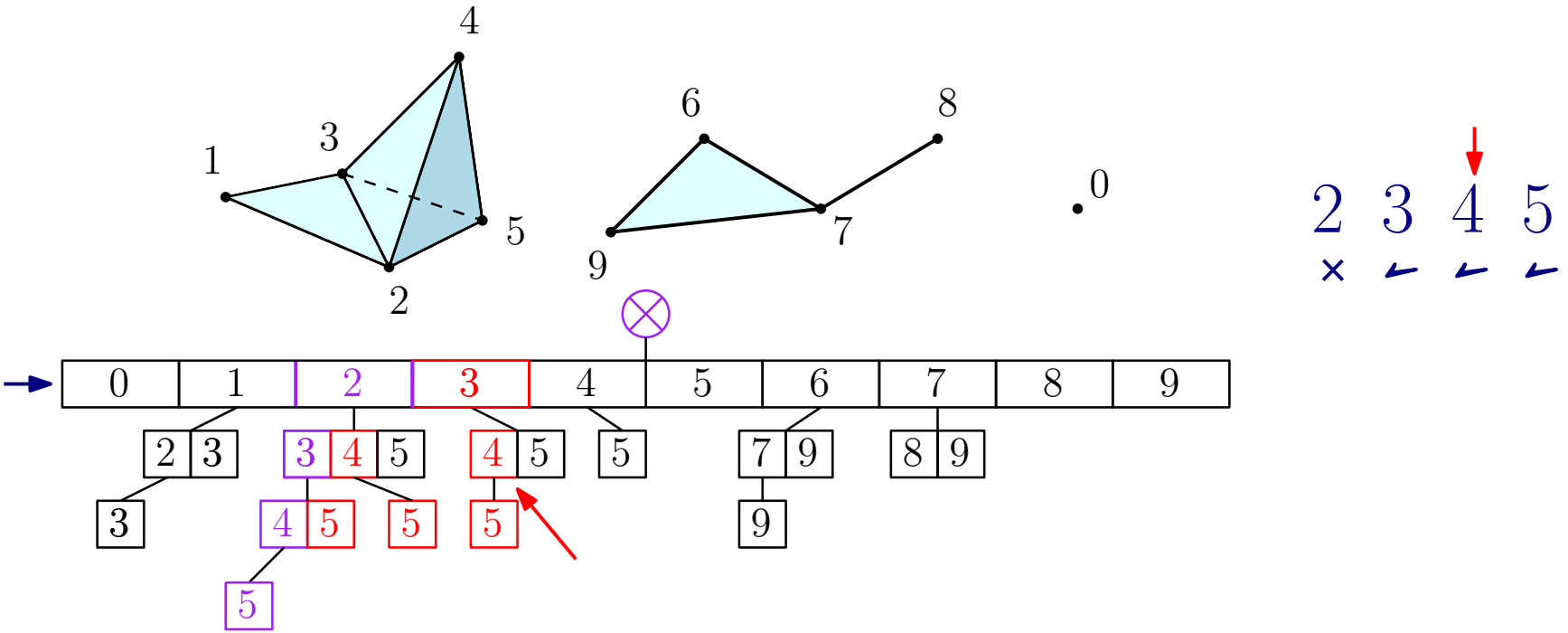
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



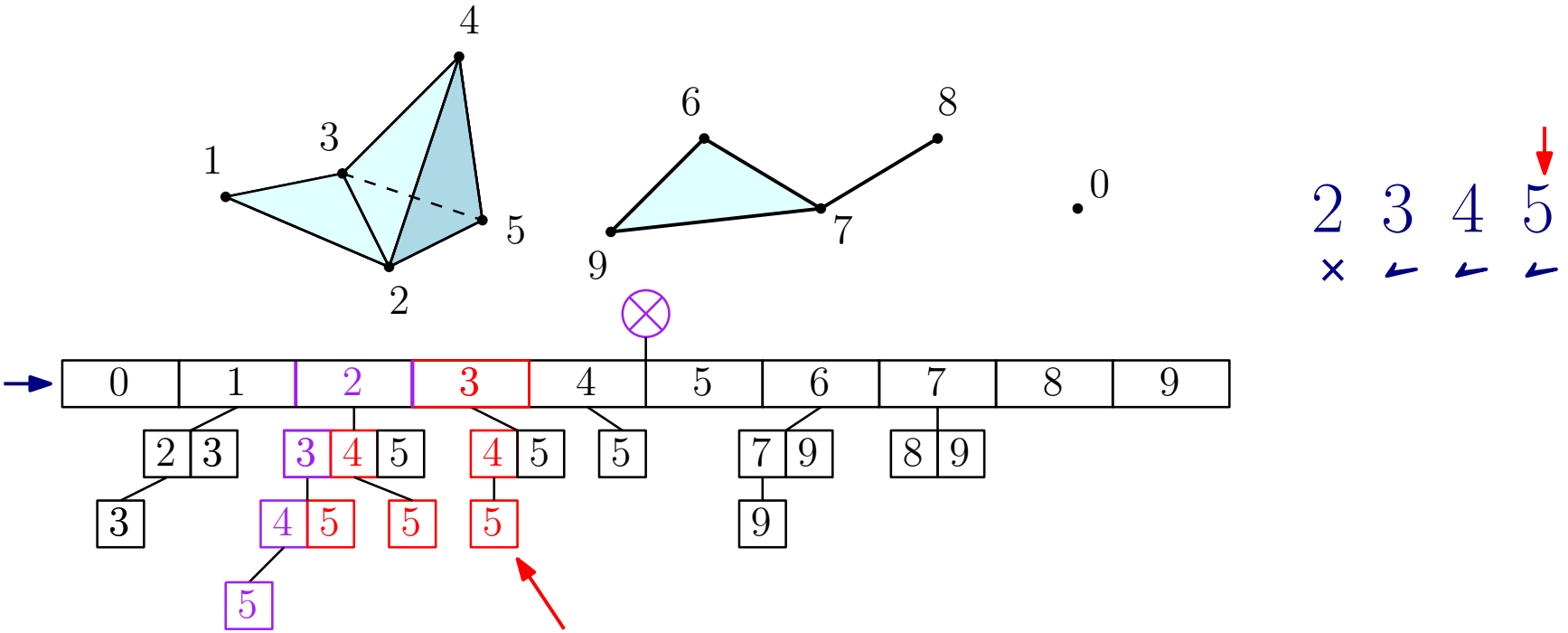
Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



Facets/Boundary Computation

Given a simplex σ we find its $|\sigma|$ codimension 1 subfaces.



3
Persistent Cohomology and Annotations

Annotations

[Dey,Fan,Wang]

Given a simplicial complex $\mathcal{K}$, let $\mathcal{K}^p$ denote the set of p-simplices in $\mathcal{K}$.

Definition

An *annotation* for $\mathcal{K}^p$ is an assignement $\mathrm{a}^p : \mathcal{K}^p \to \mathbb{Z}_2^g$ of a binary vector $\mathrm{a}_\sigma = \mathrm{a}^p(\sigma)$ of length g. Linear extension to C_p.

Definition

An annotation $\mathrm{a} : \mathcal{K}^p \to \mathbb{Z}_2^g$ is *valid* if:

1. $g = \operatorname{rank} H_p(K)$, and
2. two p-cycles z_1 and z_2 have $\mathrm{a}_{z_1} = \mathrm{a}_{z_2}$ iff their homology classes $[z_1]$ and $[z_2]$ are identical.

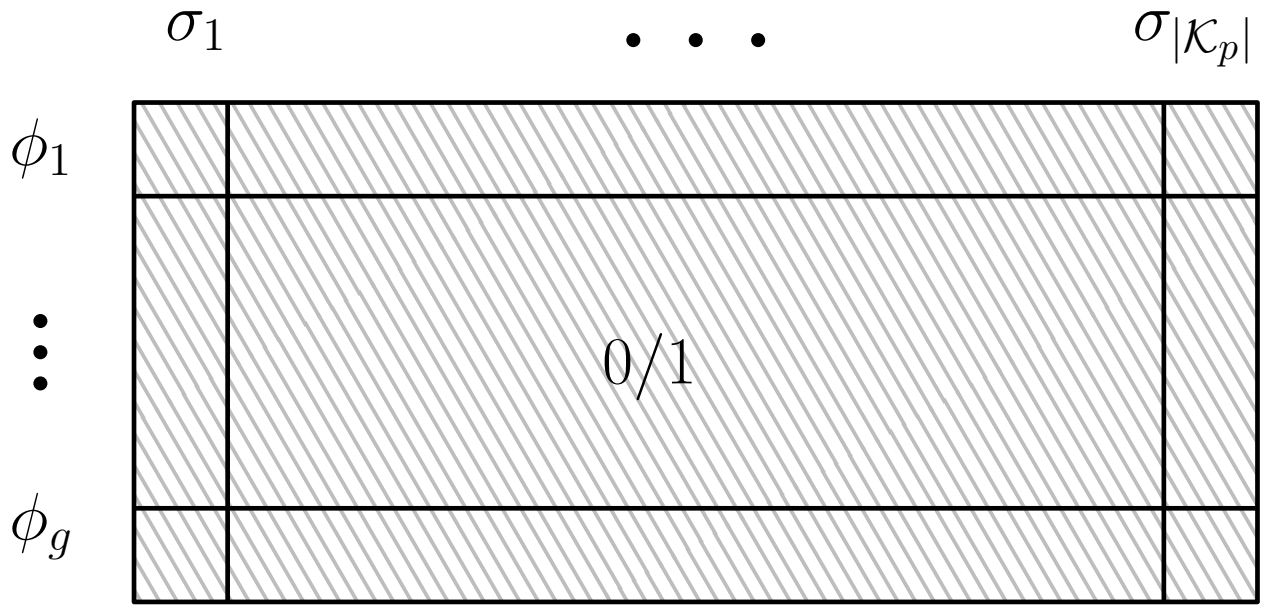
Annotations

[Dey,Fan,Wang]

Theorem

The following two statements are equivalent

1. *An annotation* $\mathsf{a} : \mathcal{K}^p \to \mathbb{Z}_2^g$ *is valid*
2. *The cochains* $\{\phi_j\}_{j=1\cdots g}$ *given by* $\phi_j(\sigma) = \mathsf{a}_\sigma[j]$ *for all* $\sigma \in \mathcal{K}^p$ *are cocycles whose cohomology classes* $\{[\phi_j]\}_{j=1\cdots g}$ *constitute a basis of* $H^p(\mathcal{K})$.

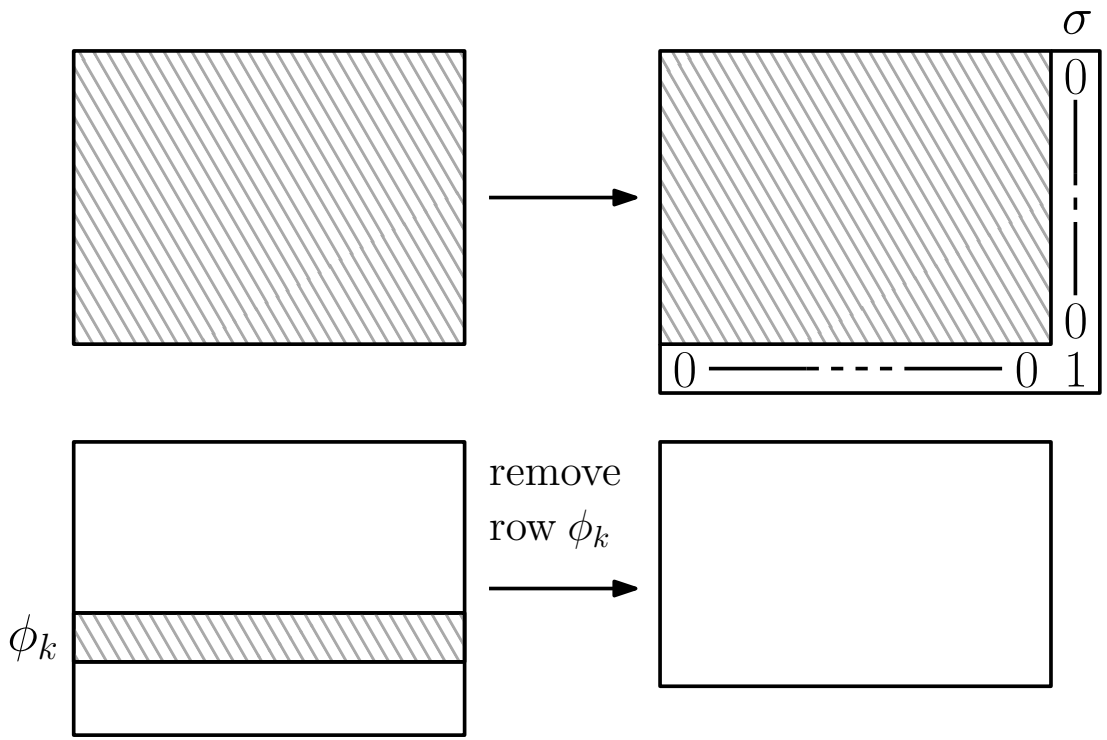


Persistent coHomology Algorithm

[de Silva, Morozov, Vejdemo-Johansson '11]; [Dey,Fan,Wang]

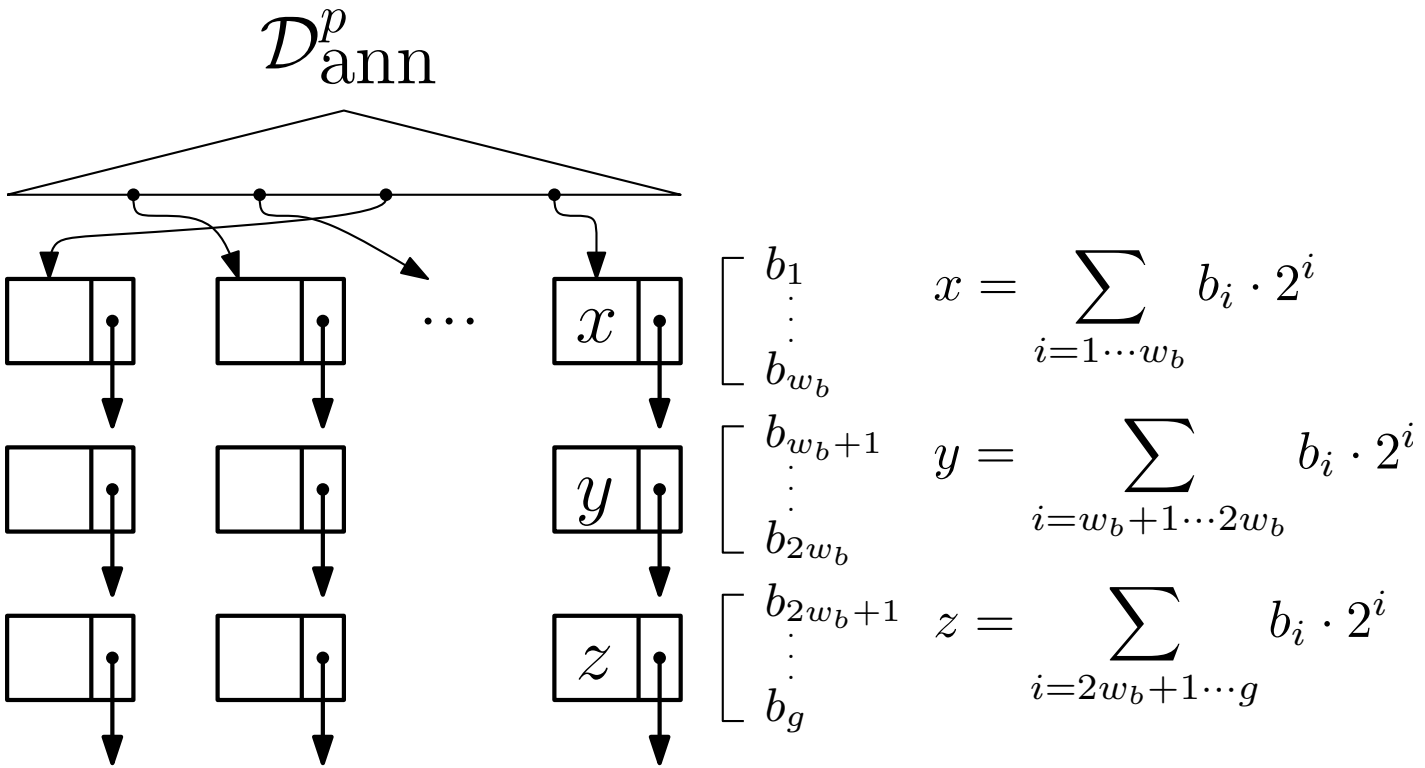
Insert $\sigma = \mathcal{K}_{i+1} \setminus \mathcal{K}_i$; compute $a_{\partial\sigma}$

- if $a_{\partial\sigma} = 0$, create a class
- else kills a class ϕ_k by adding $a_{\partial\sigma}$ to all columns with a 1 at index k

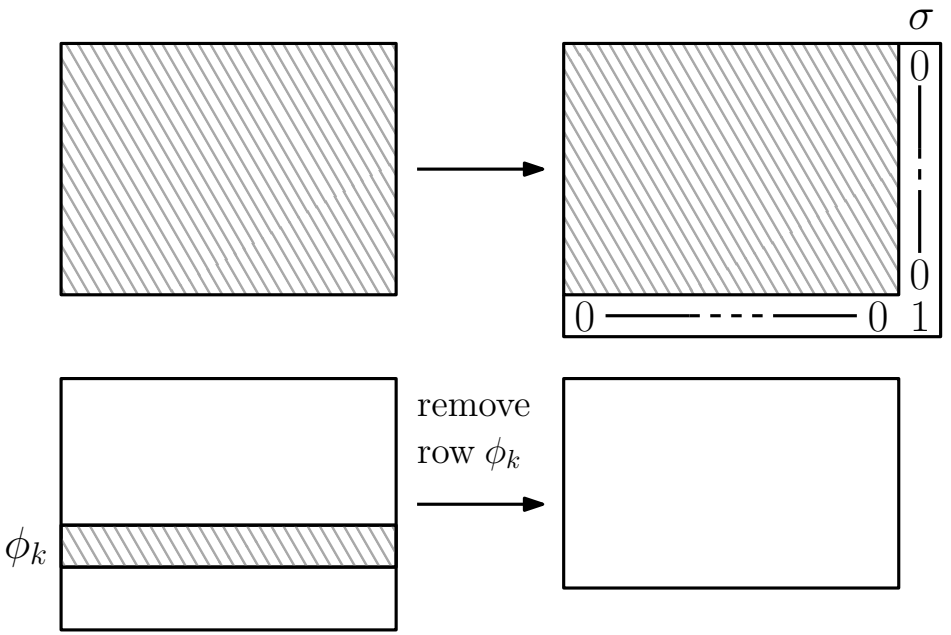


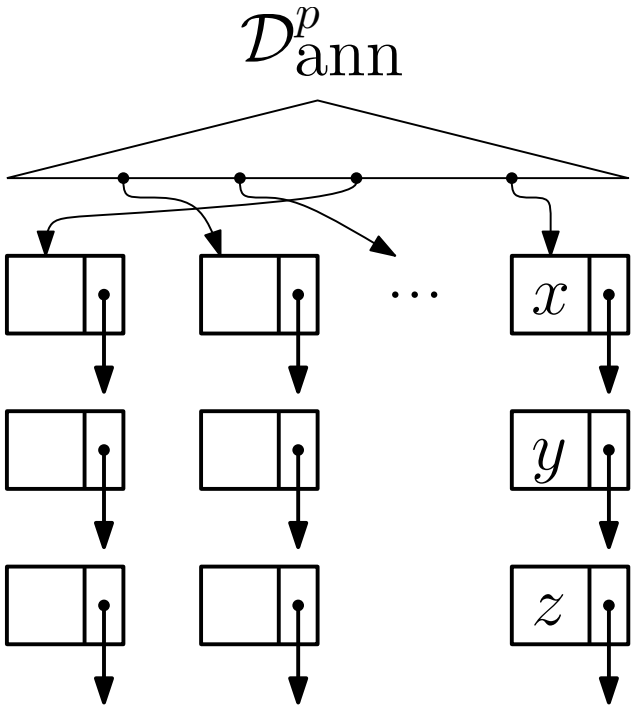
Representation of the Annotation [Boissonnat, Dey, M.]

- No duplicate vector
- Bits are "packed" into integers



Representation of the Annotation [Boissonnat, Dey, M.]



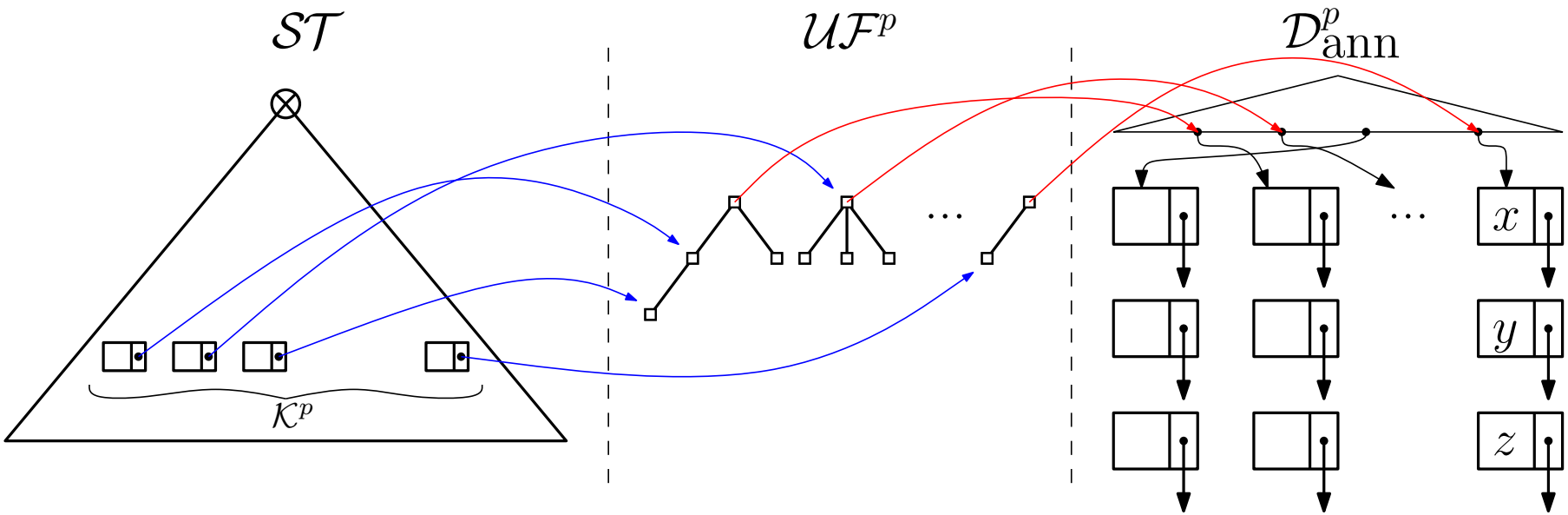


4
The Implementation

[Boissonnat, Dey, M.]

The Data Structure

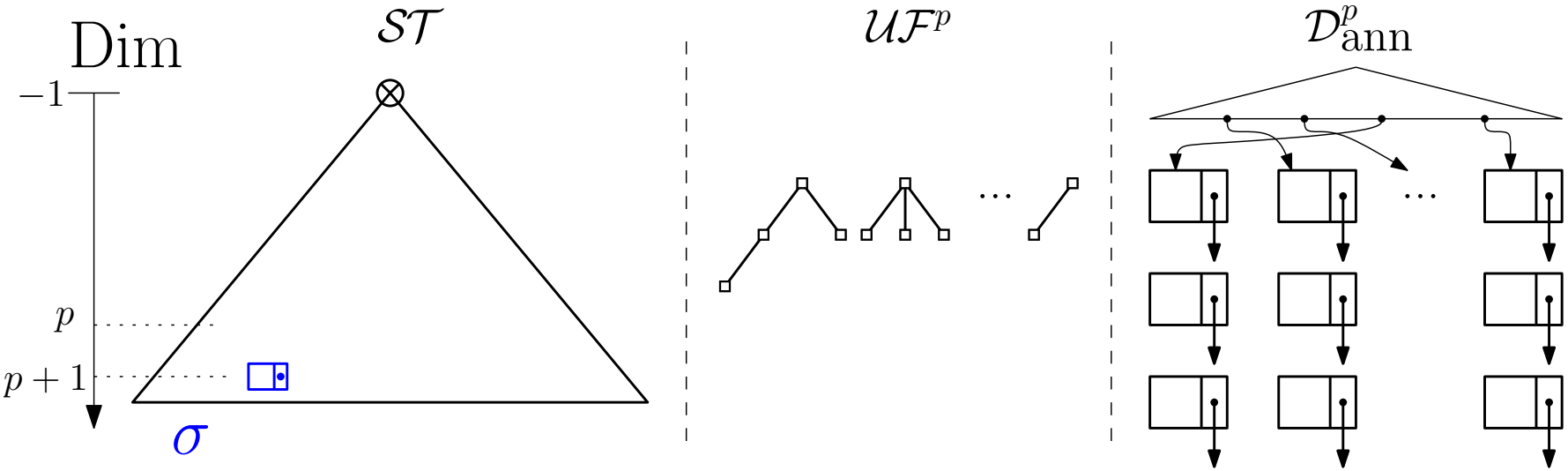
We connect simplicial complex and annotations with a union-find:



Implementation of the Algorithm

Insert $\sigma = \mathcal{K}_{i+1} \setminus \mathcal{K}_i$; compute $a_{\partial\sigma}$

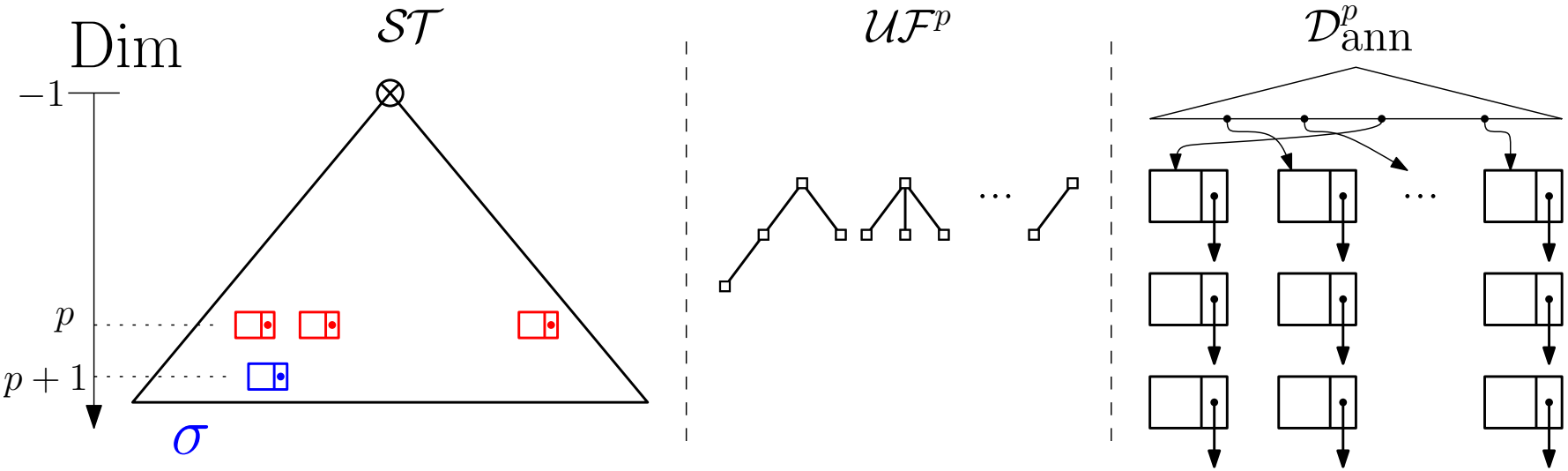
- if $a_{\partial\sigma} = 0$, create a class
- else kills a class ϕ_k by adding $a_{\partial\sigma}$ to all columns with a 1 at index k



Implementation of the Algorithm

Insert $\sigma = \mathcal{K}_{i+1} \setminus \mathcal{K}_i$; compute $a_{\partial\sigma}$

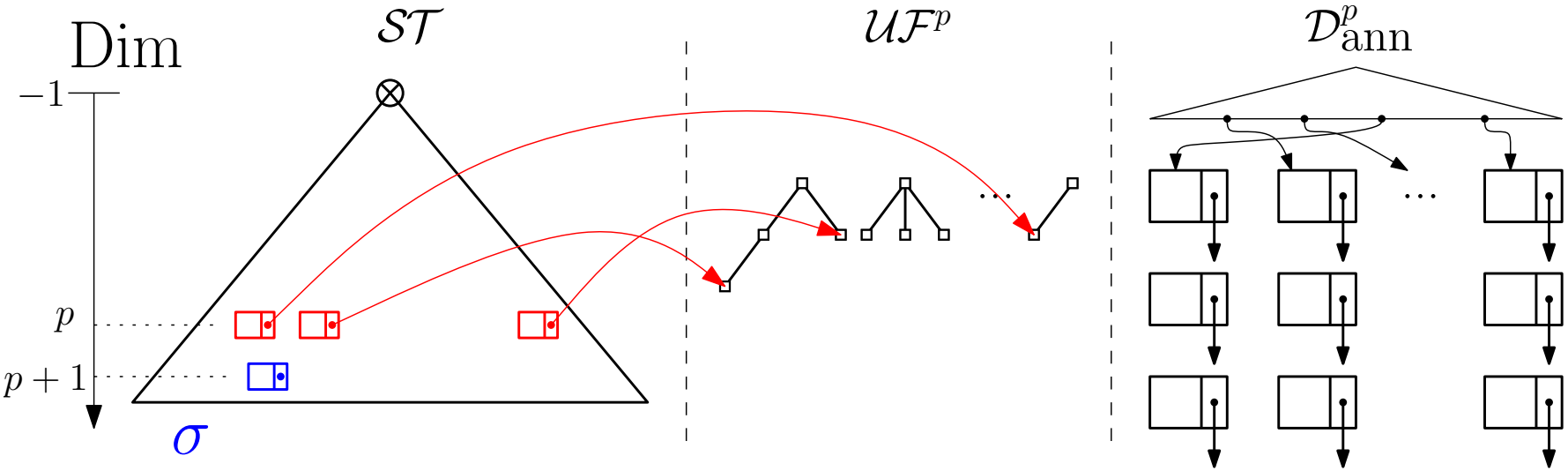
- if $a_{\partial\sigma} = 0$, create a class
- else kills a class ϕ_k by adding $a_{\partial\sigma}$ to all columns with a 1 at index k



Implementation of the Algorithm

Insert $\sigma = \mathcal{K}_{i+1} \setminus \mathcal{K}_i$; compute $a_{\partial\sigma}$

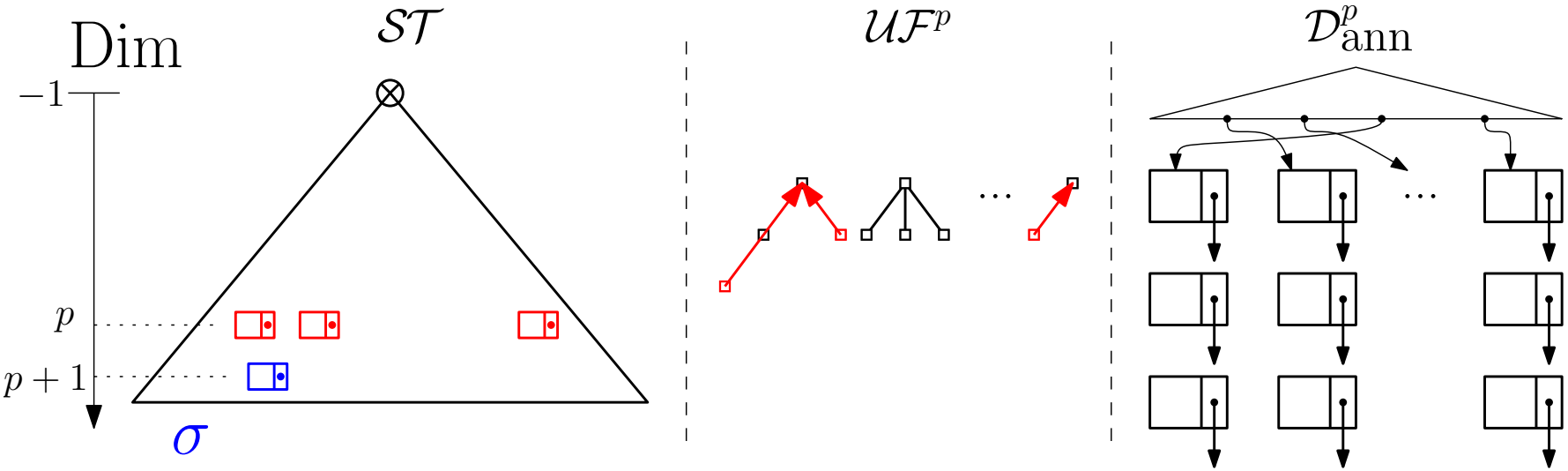
- if $a_{\partial\sigma} = 0$, create a class
- else kills a class ϕ_k by adding $a_{\partial\sigma}$ to all columns with a 1 at index k



Implementation of the Algorithm

Insert $\sigma = \mathcal{K}_{i+1} \setminus \mathcal{K}_i$; compute $a_{\partial\sigma}$

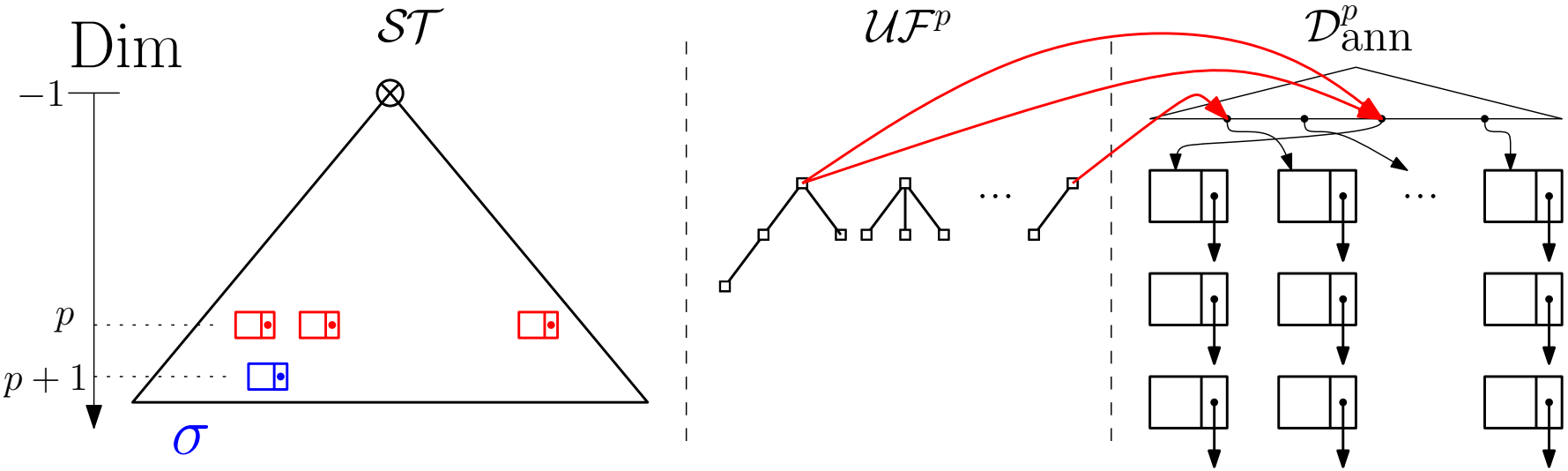
- if $a_{\partial\sigma} = 0$, create a class
- else kills a class ϕ_k by adding $a_{\partial\sigma}$ to all columns with a 1 at index k



Implementation of the Algorithm

Insert $\sigma = \mathcal{K}_{i+1} \setminus \mathcal{K}_i$; compute $a_{\partial\sigma}$

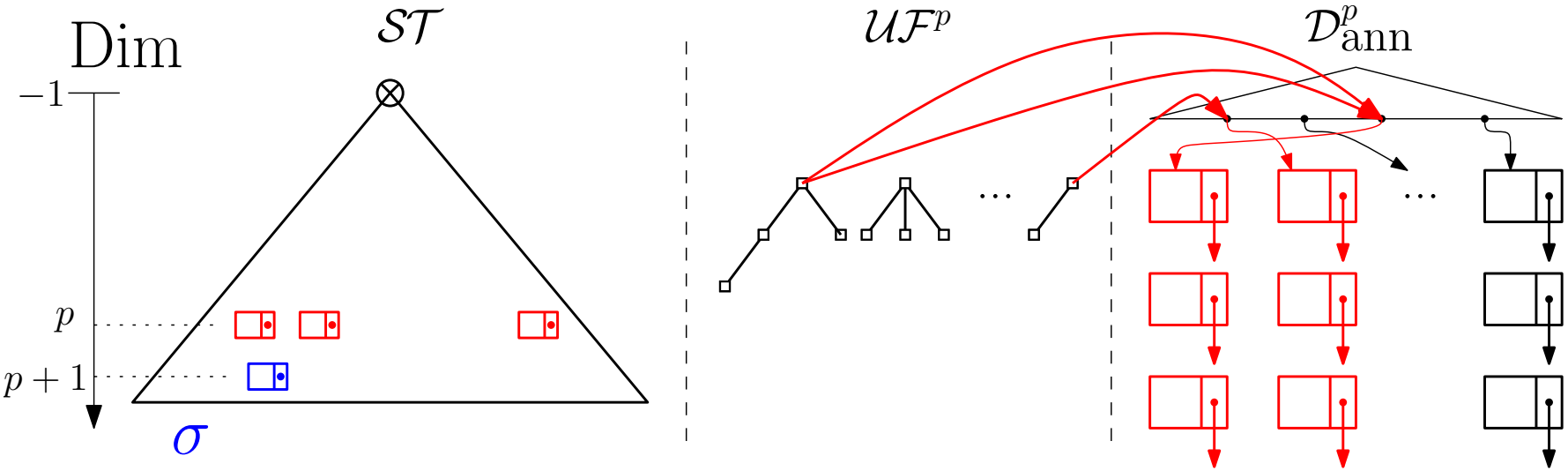
- if $a_{\partial\sigma} = 0$, create a class
- else kills a class ϕ_k by adding $a_{\partial\sigma}$ to all columns with a 1 at index k



Implementation of the Algorithm

Insert $\sigma = \mathcal{K}_{i+1} \setminus \mathcal{K}_i$; compute $a_{\partial\sigma}$

- if $a_{\partial\sigma} = 0$, create a class
- else kills a class ϕ_k by adding $a_{\partial\sigma}$ to all columns with a 1 at index k



Complexity

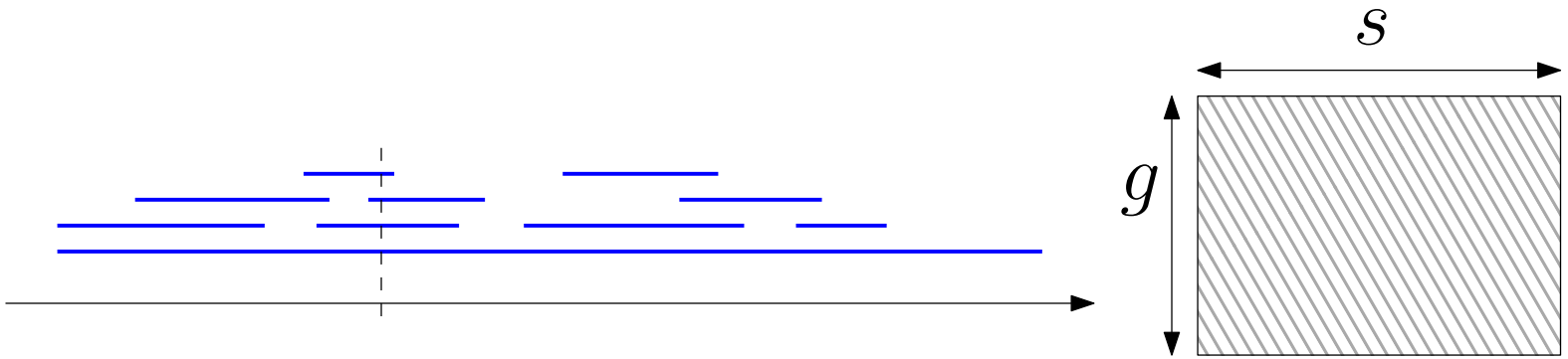
Space:

$$O\left(|\mathcal{K}| + k\frac{g_m s_m}{w_b}\right)$$

Time:

$$O\left(|\mathcal{K}|\left[k\left(\alpha(|\mathcal{K}|) + \left\lceil\frac{g_m}{w_b}\right\rceil + kD_m\right) + s_m\left(\left\lceil\frac{g_m}{w_b}\right\rceil \log(s_m) + \alpha(|\mathcal{K}|)\right)\right]\right)$$

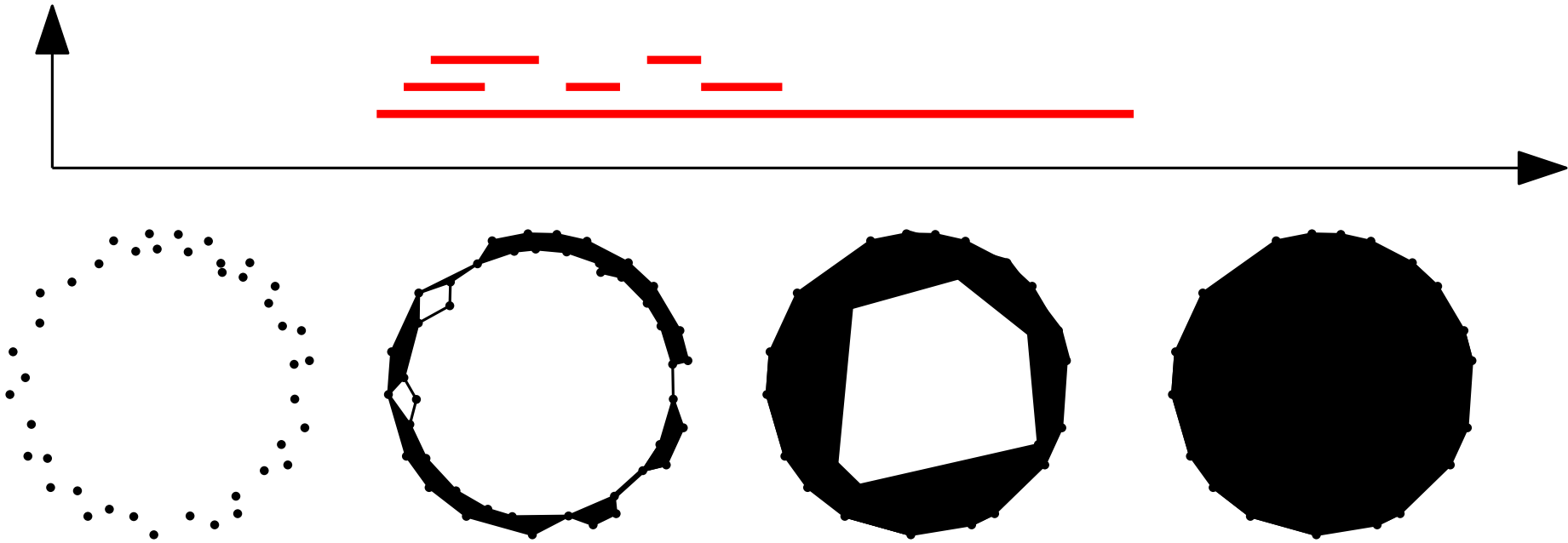
intuitively...



Lazy Evalutation of Simplices

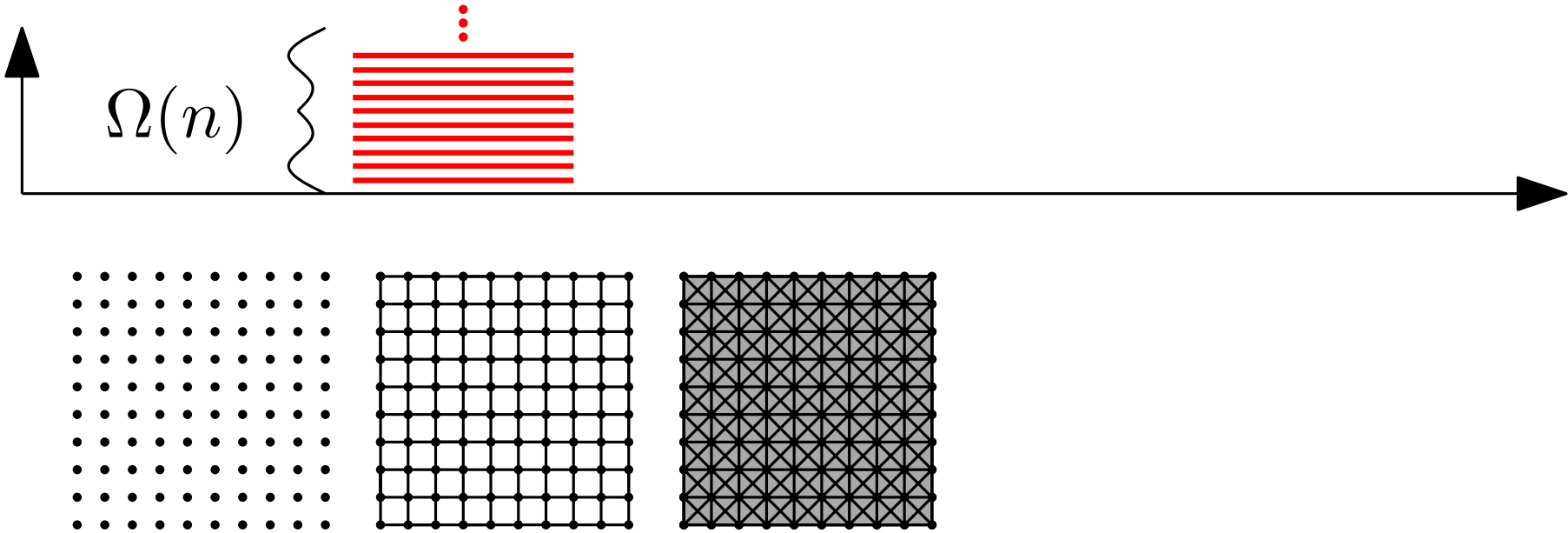
Lazy Evalutation of Simplices

We minimize the number of “alive bars” living in parallel.



Lazy Evalutation of Simplices

We minimize the number of "alive bars" living in parallel.

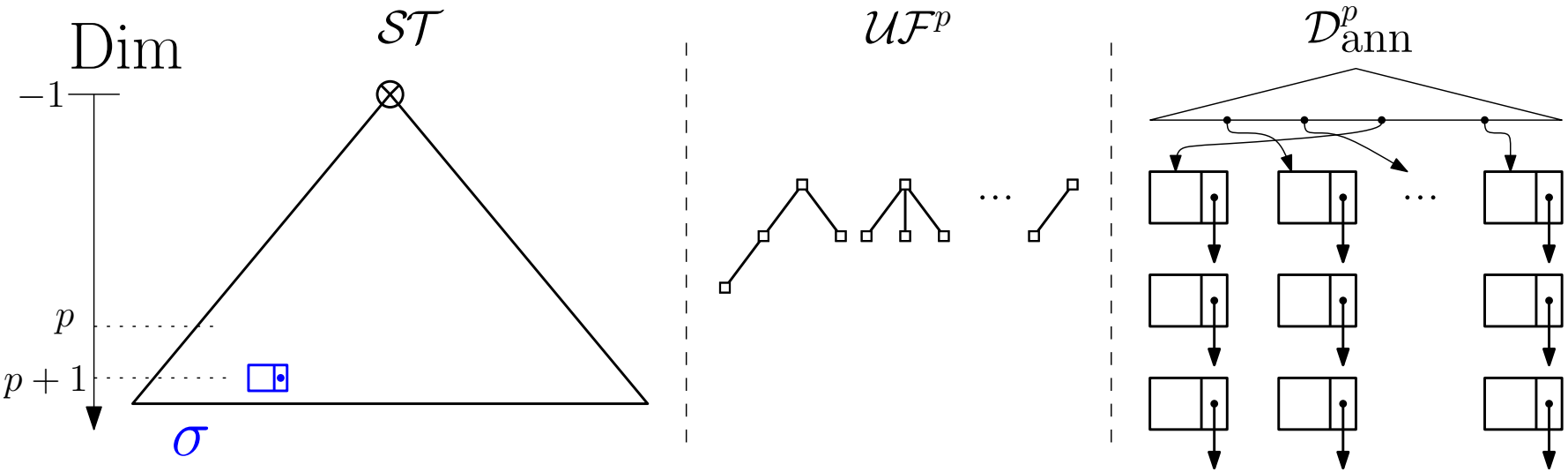


Lazy Evalutation of Simplices

$$\emptyset = \mathcal{K}_{\rho_0} \subset \mathcal{K}_{\rho_1} \subset \cdots \subset \mathcal{K}_{\rho_{m-1}} \subset \mathcal{K}_{\rho_m} = \mathcal{K}$$
$$\sigma_0 \quad \sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_{m-2} \quad \sigma_{m-1} \quad \sigma_m$$

- we insert killers as soon as we encounter them
- we postpone the insertion of a creator σ until we encounter one of its cofaces

We mark the simplices whose insertion has been postponed. When considering σ in the filtration:

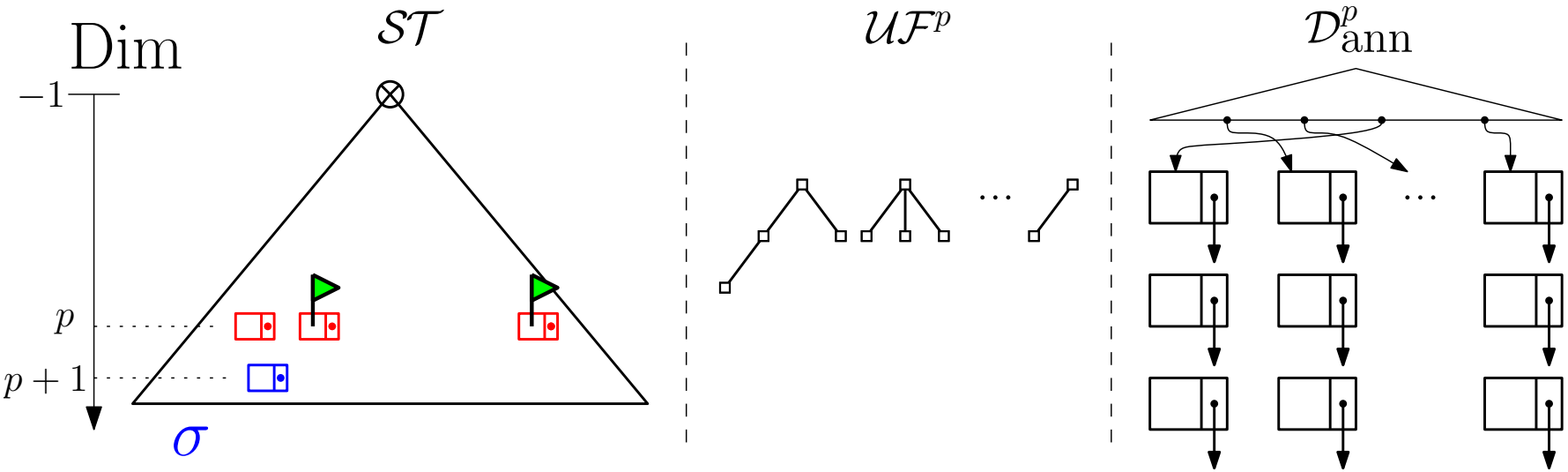


Lazy Evalutation of Simplices

$$\emptyset = \mathcal{K}_{\rho_0} \subset \mathcal{K}_{\rho_1} \subset \cdots \subset \mathcal{K}_{\rho_{m-1}} \subset \mathcal{K}_{\rho_m} = \mathcal{K}$$
$$\sigma_0 \quad \sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_{m-2} \quad \sigma_{m-1} \quad \sigma_m$$

- we insert killers as soon as we encounter them
- we postpone the insertion of a creator σ until we encounter one of its cofaces

We mark the simplices whose insertion has been postponed. When considering σ in the filtration:

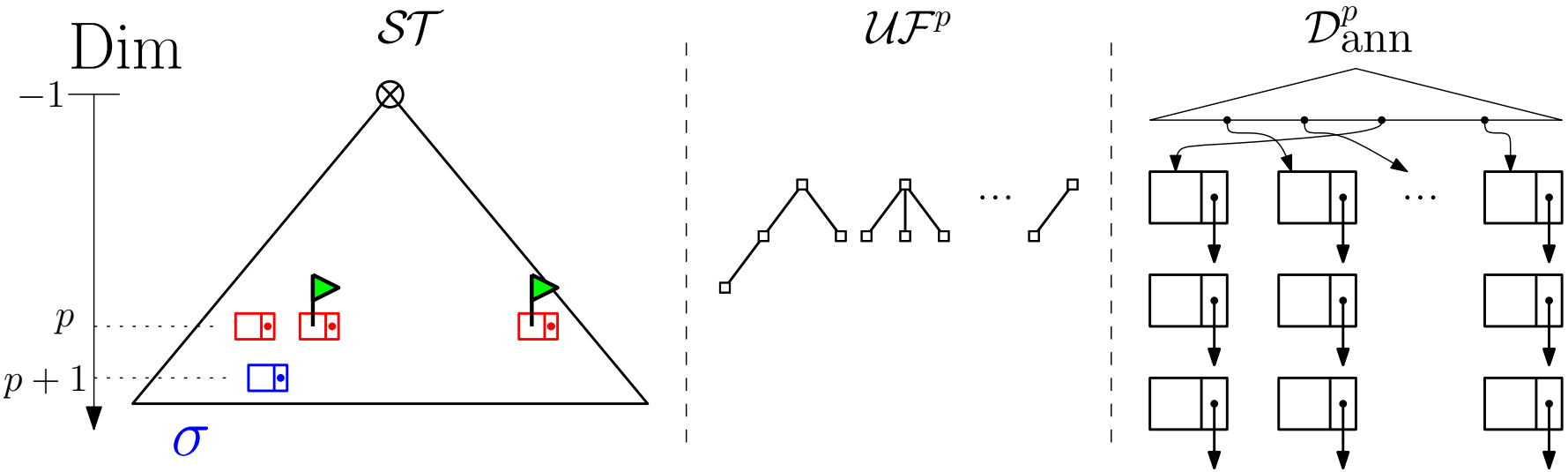


Lazy Evalutation of Simplices

$$\emptyset = \mathcal{K}_{\rho_0} \subset \mathcal{K}_{\rho_1} \subset \cdots \subset \mathcal{K}_{\rho_{m-1}} \subset \mathcal{K}_{\rho_m} = \mathcal{K}$$
$$\sigma_0 \quad \sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_{m-2} \quad \sigma_{m-1} \quad \sigma_m$$

- we insert killers as soon as we encounter them
- we postpone the insertion of a creator σ until we encounter one of its cofaces

No additional complexity!
Same persistence diagram!



5
Experiments

Performance

Data	$\lvert\mathcal{P}\rvert$	D	d	r	k	$\lvert\mathcal{K}\rvert$	T_{ann}	$T_{\text{ann}}/\lvert\mathcal{K}\rvert$(s.)
S4	507	5	4	0.75	5	$191 \cdot 10^6$	$\sim$ 32min	$1.0 \cdot 10^{-5}$
Bud	1000	3	2	2.4	3	$283 \cdot 10^6$	$\sim$ 10min	$2.2 \cdot 10^{-6}$
Bro	500	25	?	0.8	16	$38 \cdot 10^6$	$\sim$ 181min	$2.8 \cdot 10^{-4}$
Cy8	500	24	2	1.3	17	$18 \cdot 10^6$	$\sim$ 9min	$3.1 \cdot 10^{-5}$
KI	4900	5	2	0.415	5	$218 \cdot 10^6$	$\sim$ 29min	$7.9 \cdot 10^{-6}$

Figure: Data, timings and statistics

$$O\left(|\mathcal{K}| + k\frac{g_{\text{m}}s_{\text{m}}}{w_b}\right)$$

$$O\left(|\mathcal{K}|\left[k\left(\alpha(|\mathcal{K}|) + \left\lceil\frac{g_{\text{m}}}{w_{\text{b}}}\right\rceil + kD_{\text{m}}\right) + s_{\text{m}}\left(\left\lceil\frac{g_{\text{m}}}{w_{\text{b}}}\right\rceil \log(s_{\text{m}}) + \alpha(|\mathcal{K}|)\right)\right]\right)$$

Homology vs coHomology

Comparison with JPLEX and DIONYSUS: Persistent Homology vs Persistent coHomology

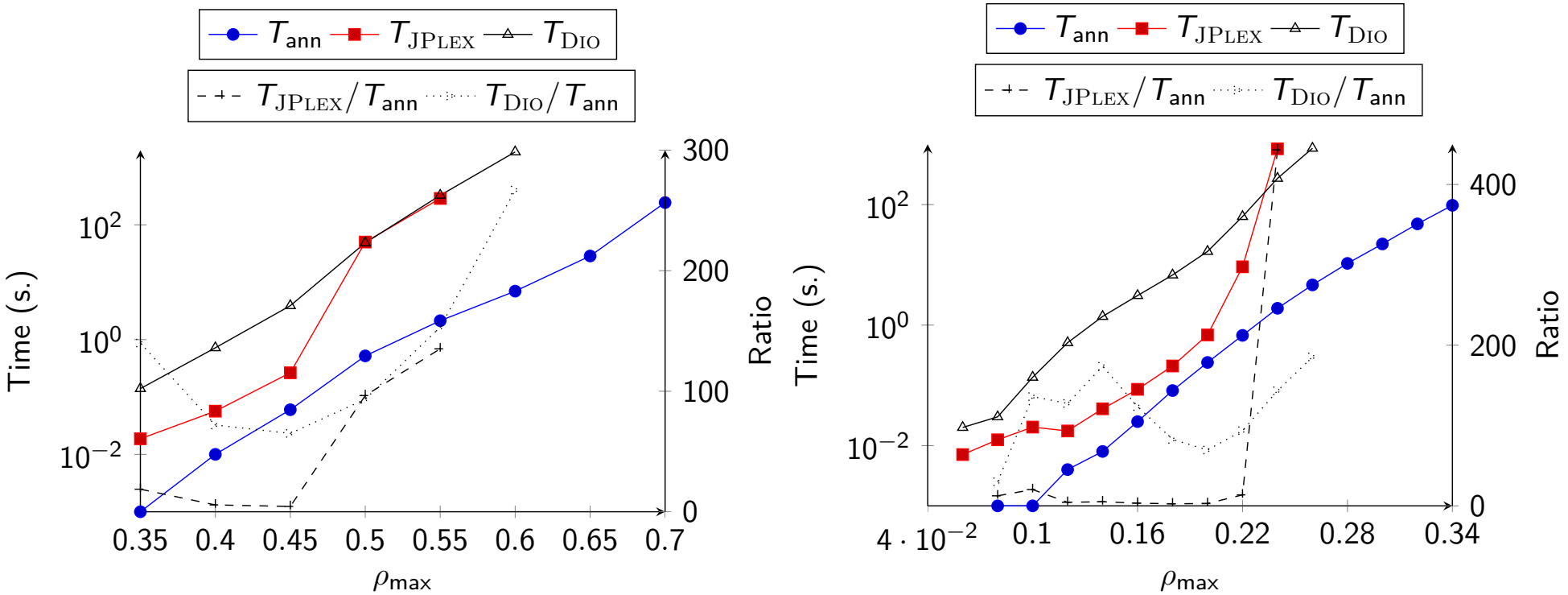
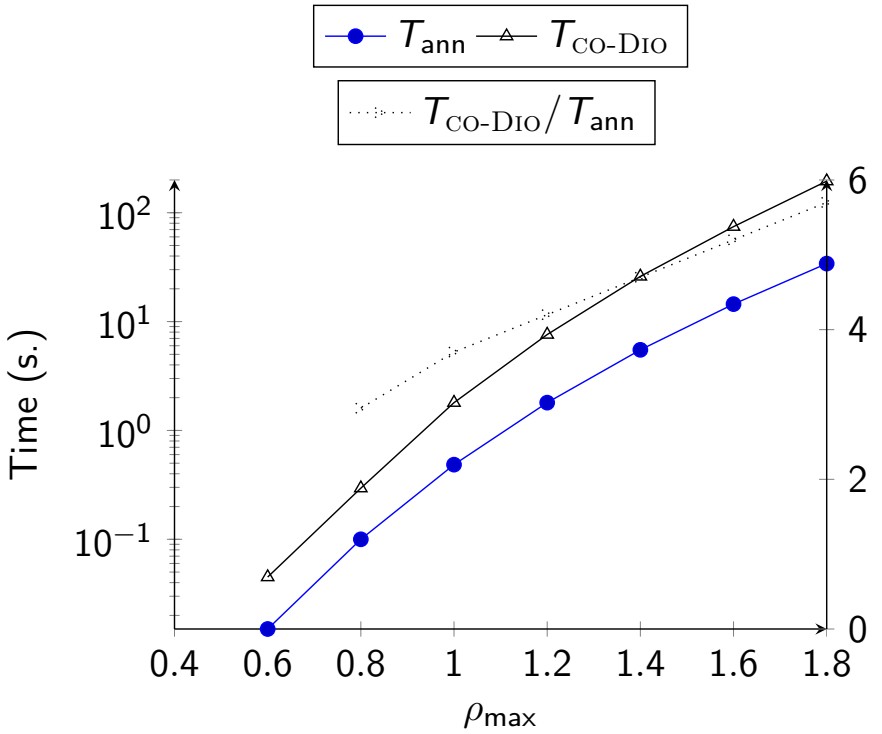


Figure: **S4** and **KI**

coHomology vs coHomology

Comparison with DIONYSUS: coHomology vs coHomology



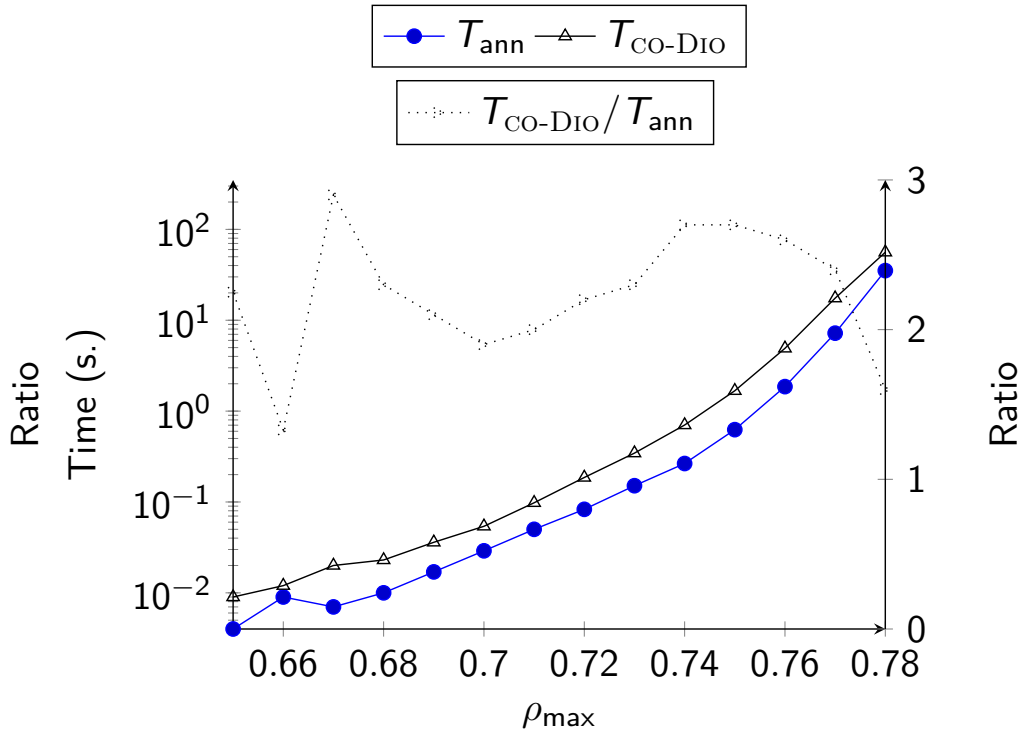
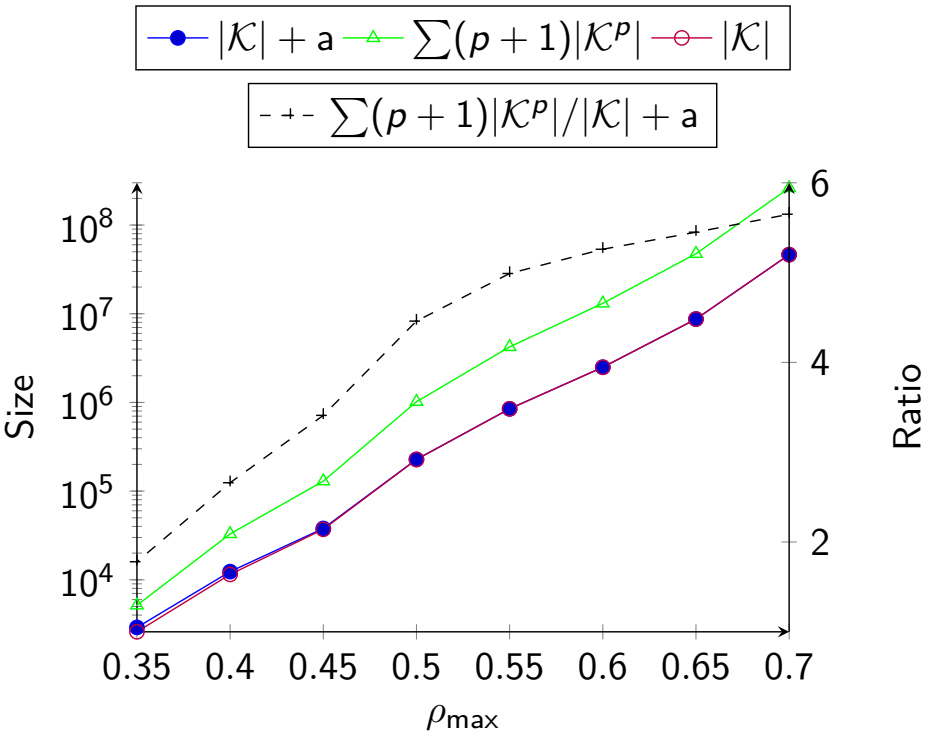


Figure: **Bud** and **Bro**

Memory



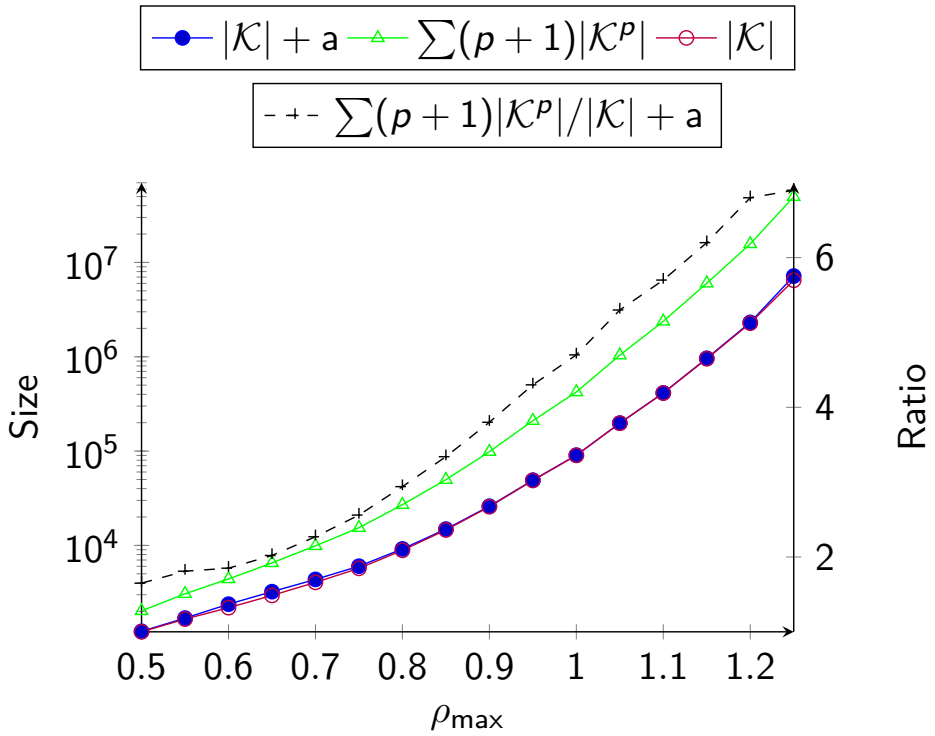


Figure: **S4** and **Cy8**

Numbers!

Dim:	**S4**: 1	2	3	4	**Bud**: 1	2	**Kl**: 1	2	3	4
G_m/g_m	4.9	1.2	6.2	6.1	2.1	8.4	11.8	10.3	10.3	8.9
$\lvert\mathcal{K}^p\rvert/s_m$	10^2	10^3	10^4	10^3	10^3	10^5	10^3	10^4	10^4	10^4
$\lceil g_m/w_b \rceil$	5	4	10	86	5	2	4	1	11	87

Dim:	**Bro**: 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G_m/g_m	2.1	1.8	5.4	5.7	5.2	5.6	4.0	4.2	3.0	3.7	3.7	2.9	3.0	3.7	3.0
$\lvert\mathcal{K}^p\rvert/s_m$	10^1	10^3	10^3	10^3	10^2	10^2	10^2	10^2	10^2	10^2	10^2	10^2	10^2	10^2	10^2
$\lceil g_m/w_b \rceil$	6	6	6	36	150	330	703	694	726	343	133	46	8	228	342

Dim:	**Cy8**: 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
G_m/g_m	2.3	1.7	3.9	3.8	4.1	4.1	3.5	3.5	3.9	2.7	2.6	3.0	2.8	1.9	2.7	1
$\lvert\mathcal{K}^p\rvert/s_m$	10^3	10^3	10^3	10^3	10^3	10^3	10^2	10^2	10^2	10^2	10^2	10^2	10^2	10^2	10^2	10^2
$\lceil g_m/w_b \rceil$	3	1	3	12	33	74	147	186	160	168	93	33	10	3	1	1

Figure: Statistics about the evolution of the dimension of the homology groups we maintain during the computation.

Conclusion

Future Work

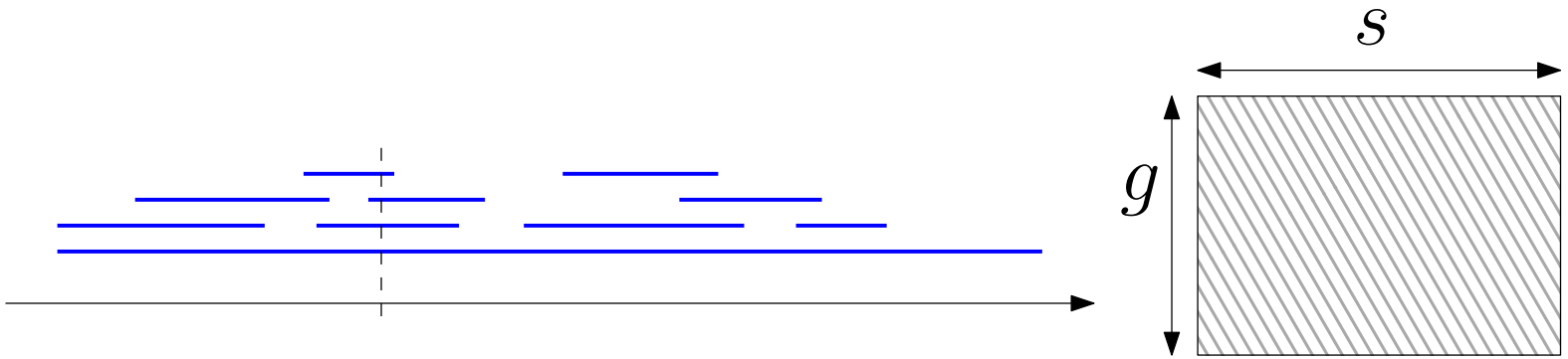
Space:

$$O\left(|\mathcal{K}| + k\frac{g_m s_m}{w_b}\right)$$

Time:

$$O\left(|\mathcal{K}|\left[k\left(\alpha(|\mathcal{K}|) + \left\lceil\frac{g_m}{w_b}\right\rceil + kD_m\right) + s_m\left(\left\lceil\frac{g_m}{w_b}\right\rceil \log(s_m) + \alpha(|\mathcal{K}|)\right)\right]\right)$$

intuitively...



Thank you!

Question?