A Topological Tree of Shapes

Nicolas Passat[(1)] & Yukiko Kenmochi[(2)]

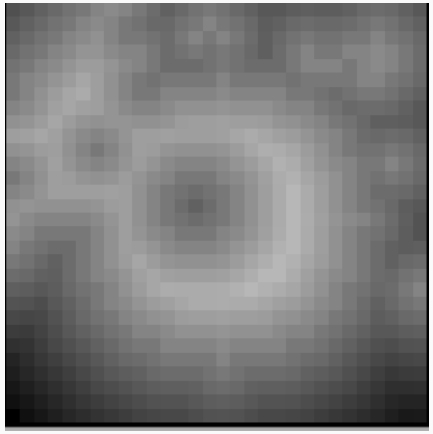
(1) Université de Reims Champagne-Ardenne, CReSTIC, Reims, France
(2) Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France

DGMM 2022, Strasbourg, 26 October 2022

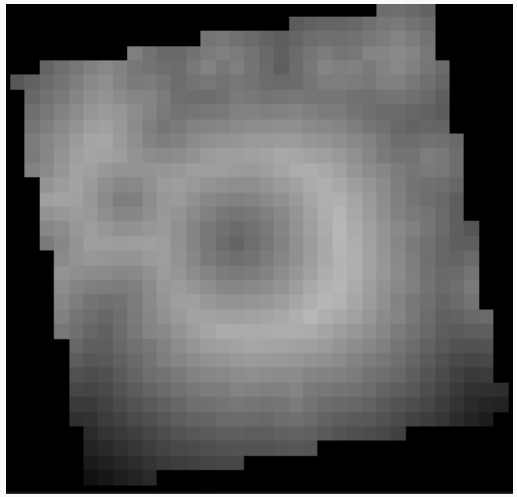
Introduction

Foreword

How to visualize/quantify topological differences between

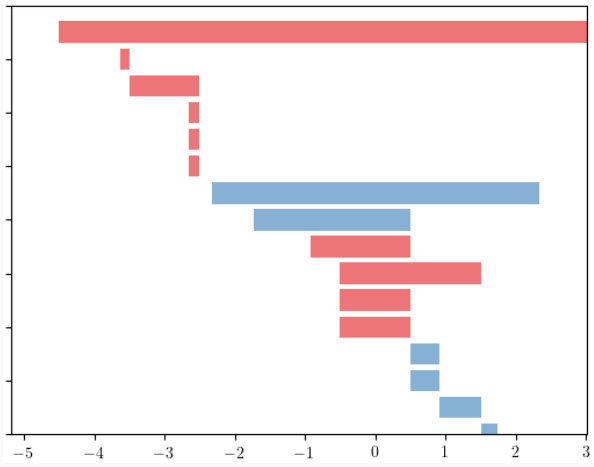


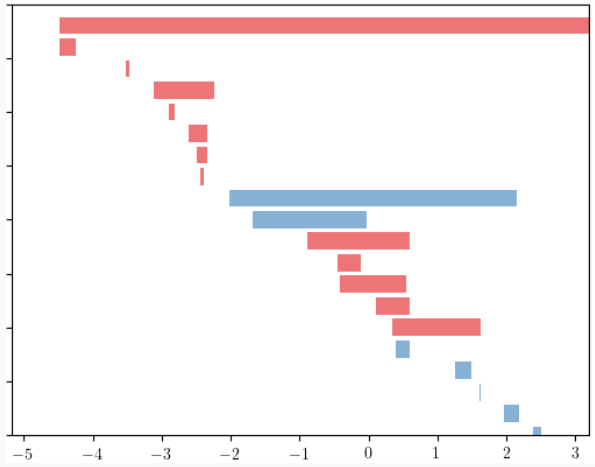
and



?

For example, we can use tools of persistent homology:





Foreword

Context

- Grey-level imaging
- Morphological hierarchies
- Topological descriptors

Foreword

Context

- Grey-level imaging
- Morphological hierarchies
- Topological descriptors

Motivations

- Morphological hierarchies as topological descriptors
  → “Beyond persistent homology”

Foreword

Context

- Grey-level imaging
- Morphological hierarchies
- Topological descriptors

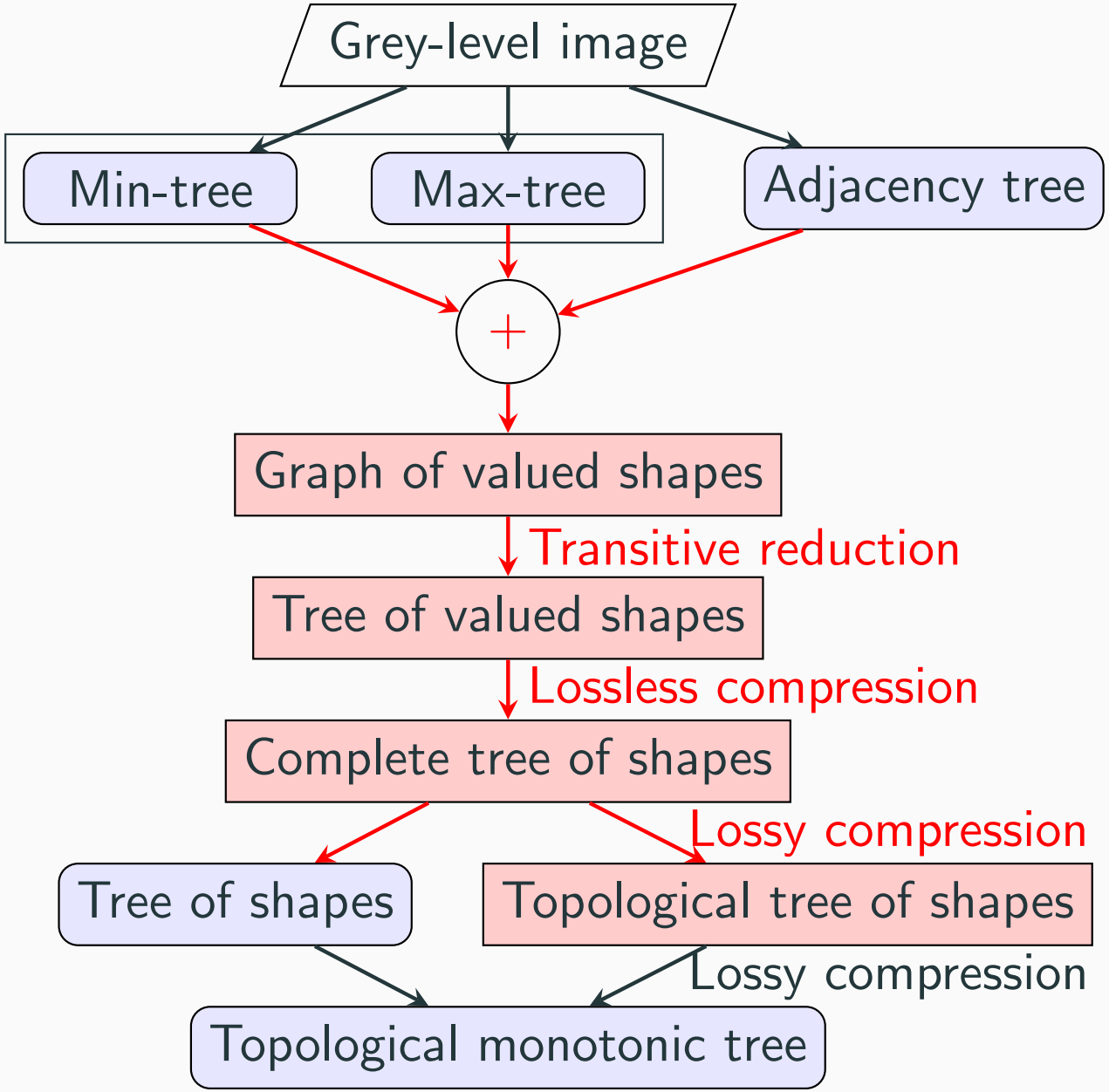
Motivations

- Morphological hierarchies as topological descriptors
  → "Beyond persistent homology"

Contributions

- New hierarchical models
- Unification of well-known hierarchical models
- Theoretical study

Graphical abstract (for those who are too tired...)



Overview

Component-tree

Grey-level images

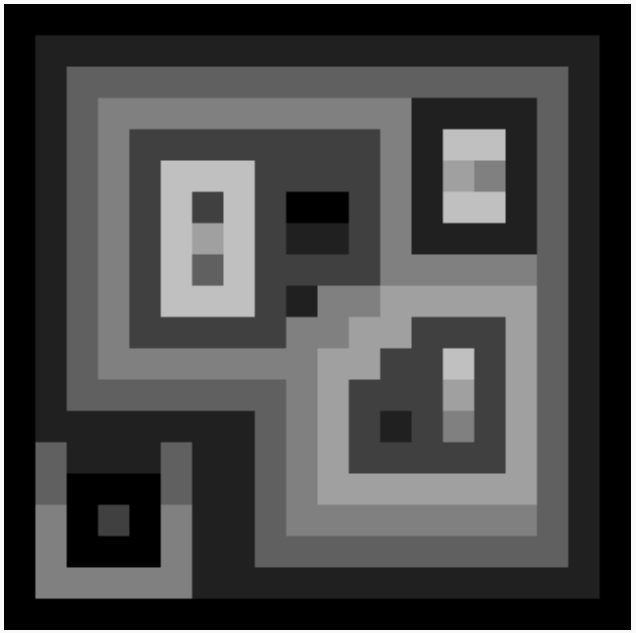
Space, values, image

- Space $\mathbb{U} = \mathbb{Z}^n$ ($n \geq 2$) endowed with digital topology (but other tesselations of $\mathbb{R}^n$ would work...)
- Values $\mathbb{V} \simeq \mathbb{Z}$ (i.e. endowed with a total order $\leqslant_{\mathbb{V}}$)
- Image = function $\mathcal{F} : \mathbb{U} \to \mathbb{V}$

Grey-level images

Space, values, image

- Space $\mathbb{U} = \mathbb{Z}^n$ $(n \geq 2)$ endowed with digital topology (but other tesselations of $\mathbb{R}^n$ would work. . .)
- Values $\mathbb{V} \simeq \mathbb{Z}$ (i.e. endowed with a total order $\leqslant_{\mathbb{V}}$)
- Image = function $\mathcal{F} : \mathbb{U} \to \mathbb{V}$



Thresholding(s), binary images

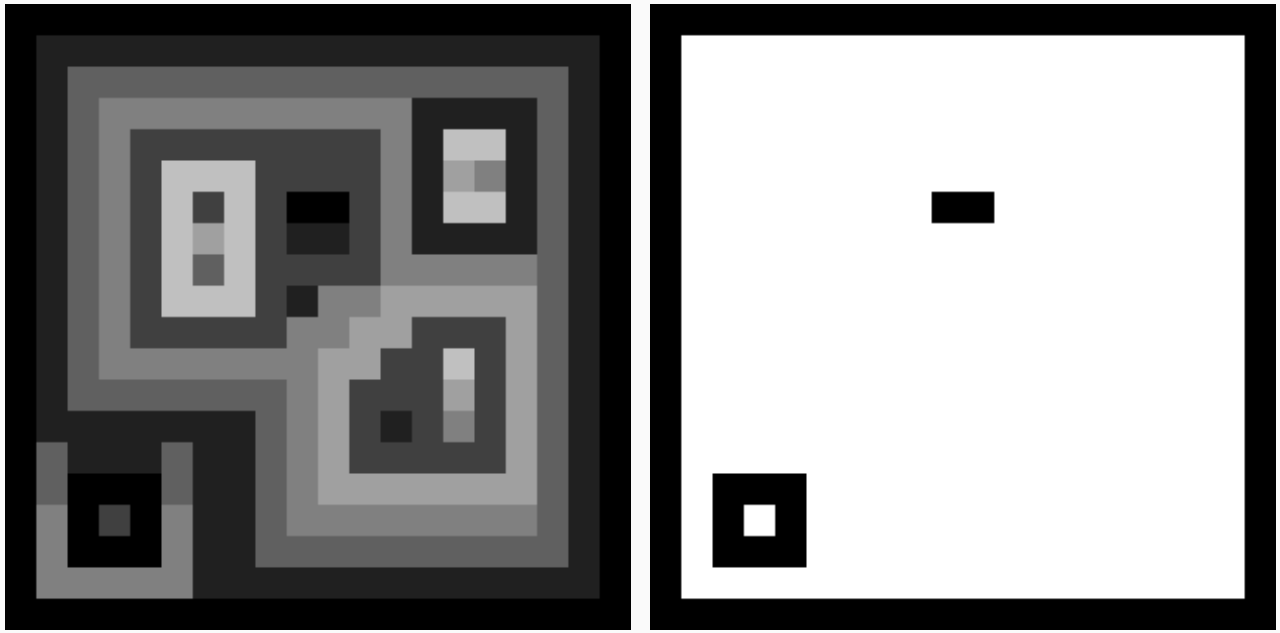
Thresholding(s)

- $\Lambda_v^\circ(\mathcal{F}) \subseteq \mathbb{U}$: upper threshold set of $\mathcal{F}$ at value v (in white)
- $\Lambda_v^\bullet(\mathcal{F}) \subseteq \mathbb{U}$: lower threshold set of $\mathcal{F}$ at value v (in black)

Thresholding(s), binary images

Thresholding(s)

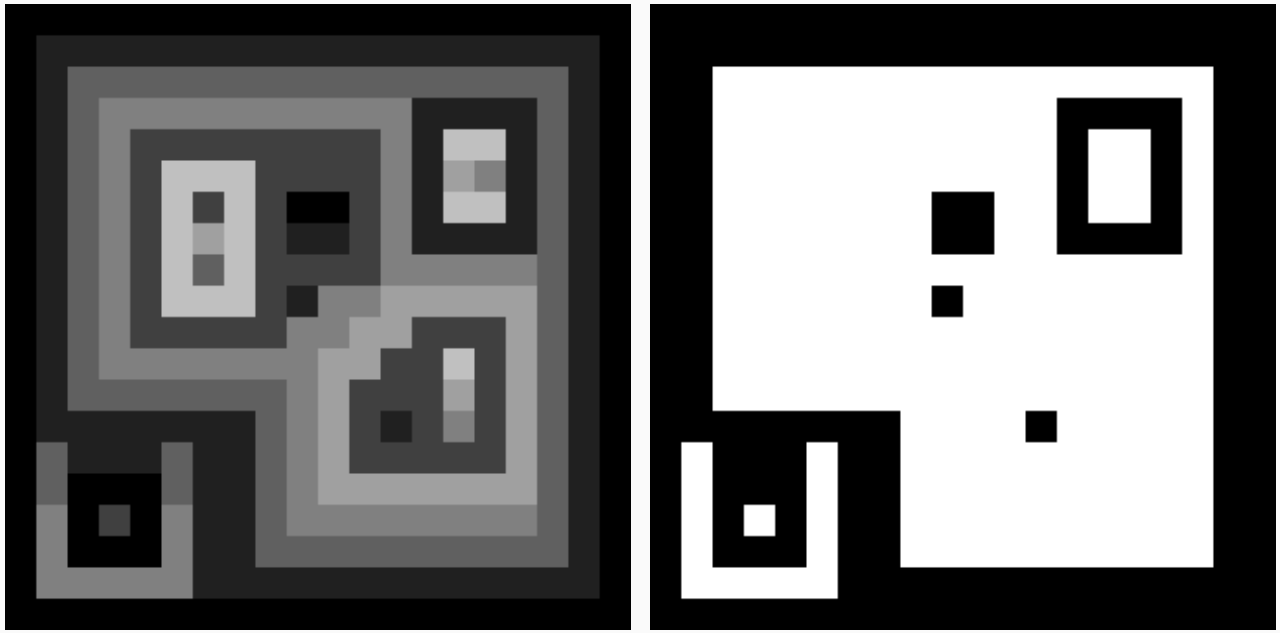
- $\Lambda_v^\circ(\mathcal{F}) \subseteq \mathbb{U}$: upper threshold set of $\mathcal{F}$ at value v (in white)
- $\Lambda_v^\bullet(\mathcal{F}) \subseteq \mathbb{U}$: lower threshold set of $\mathcal{F}$ at value v (in black)



Thresholding(s), binary images

Thresholding(s)

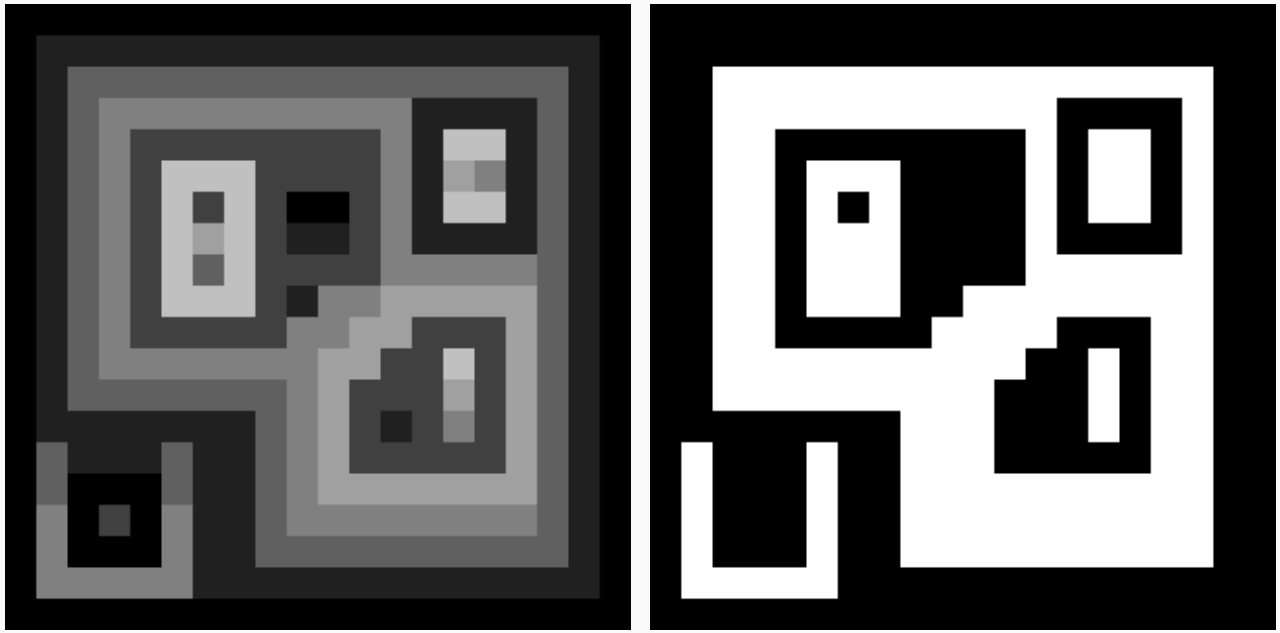
- $\Lambda_v^\circ(\mathcal{F}) \subseteq \mathbb{U}$: upper threshold set of $\mathcal{F}$ at value v (in white)
- $\Lambda_v^\bullet(\mathcal{F}) \subseteq \mathbb{U}$: lower threshold set of $\mathcal{F}$ at value v (in black)



Thresholding(s), binary images

Thresholding(s)

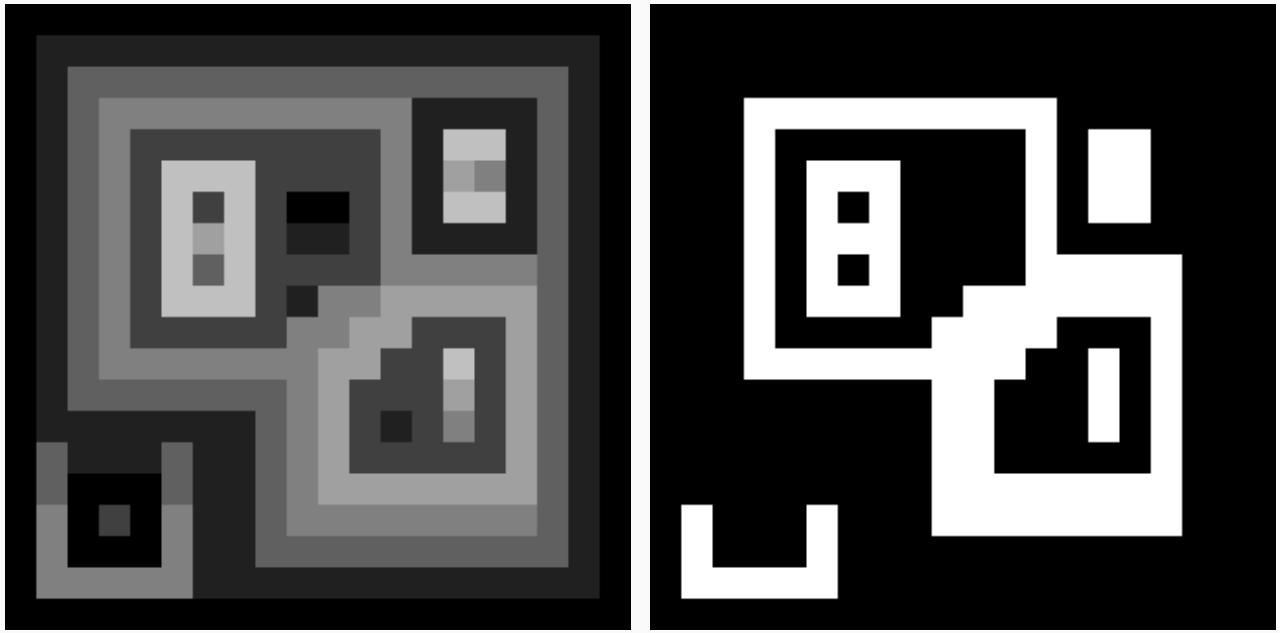
- $\Lambda_v^\circ(\mathcal{F}) \subseteq \mathbb{U}$: upper threshold set of $\mathcal{F}$ at value v (in white)
- $\Lambda_v^\bullet(\mathcal{F}) \subseteq \mathbb{U}$: lower threshold set of $\mathcal{F}$ at value v (in black)



Thresholding(s), binary images

Thresholding(s)

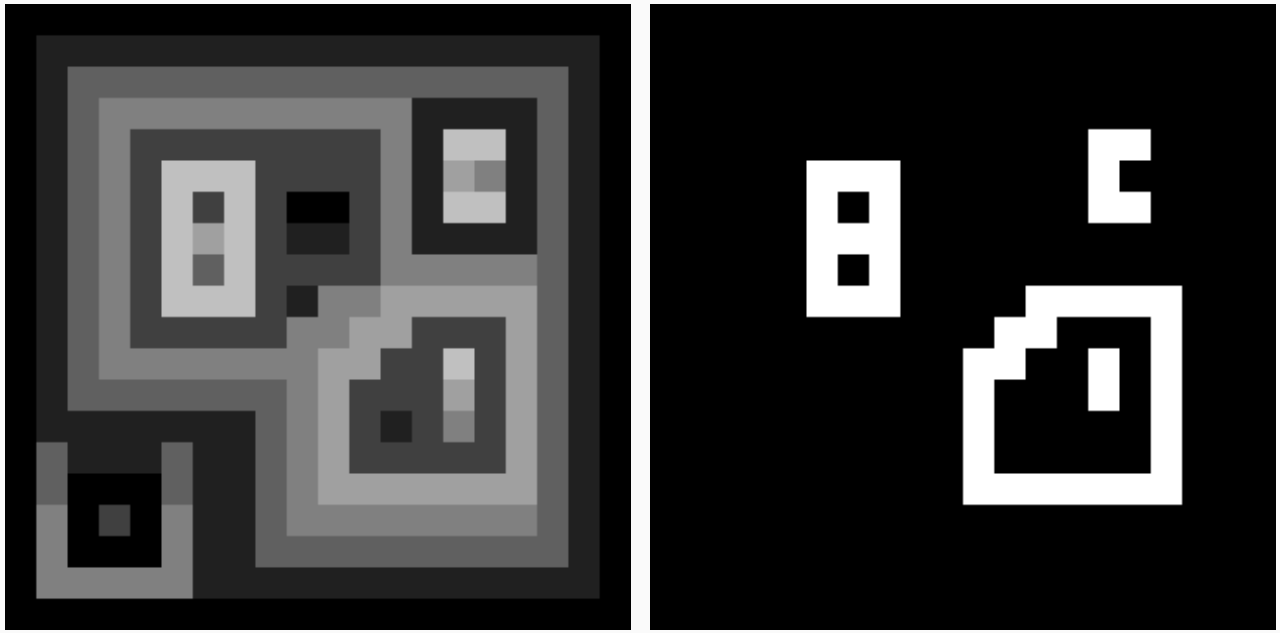
- $\Lambda_v^\circ(\mathcal{F}) \subseteq \mathbb{U}$: upper threshold set of $\mathcal{F}$ at value v (in white)
- $\Lambda_v^\bullet(\mathcal{F}) \subseteq \mathbb{U}$: lower threshold set of $\mathcal{F}$ at value v (in black)



Thresholding(s), binary images

Thresholding(s)

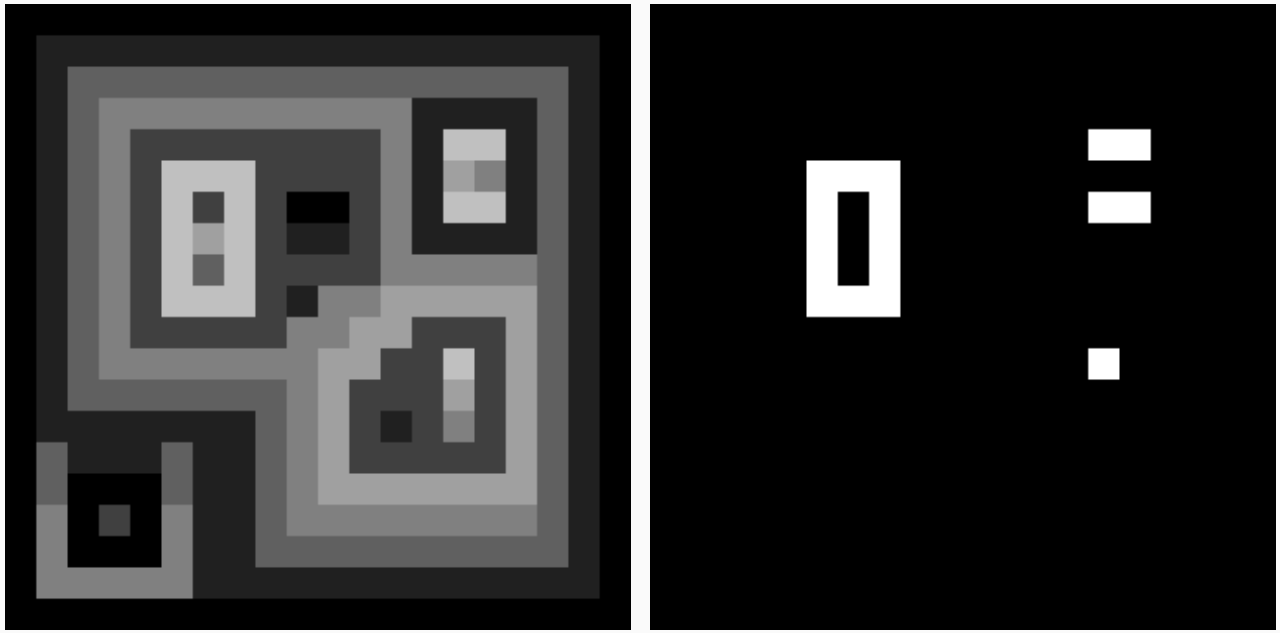
- $\Lambda_v^\circ(\mathcal{F}) \subseteq \mathbb{U}$: upper threshold set of $\mathcal{F}$ at value v (in white)
- $\Lambda_v^\bullet(\mathcal{F}) \subseteq \mathbb{U}$: lower threshold set of $\mathcal{F}$ at value v (in black)



Thresholding(s), binary images

Thresholding(s)

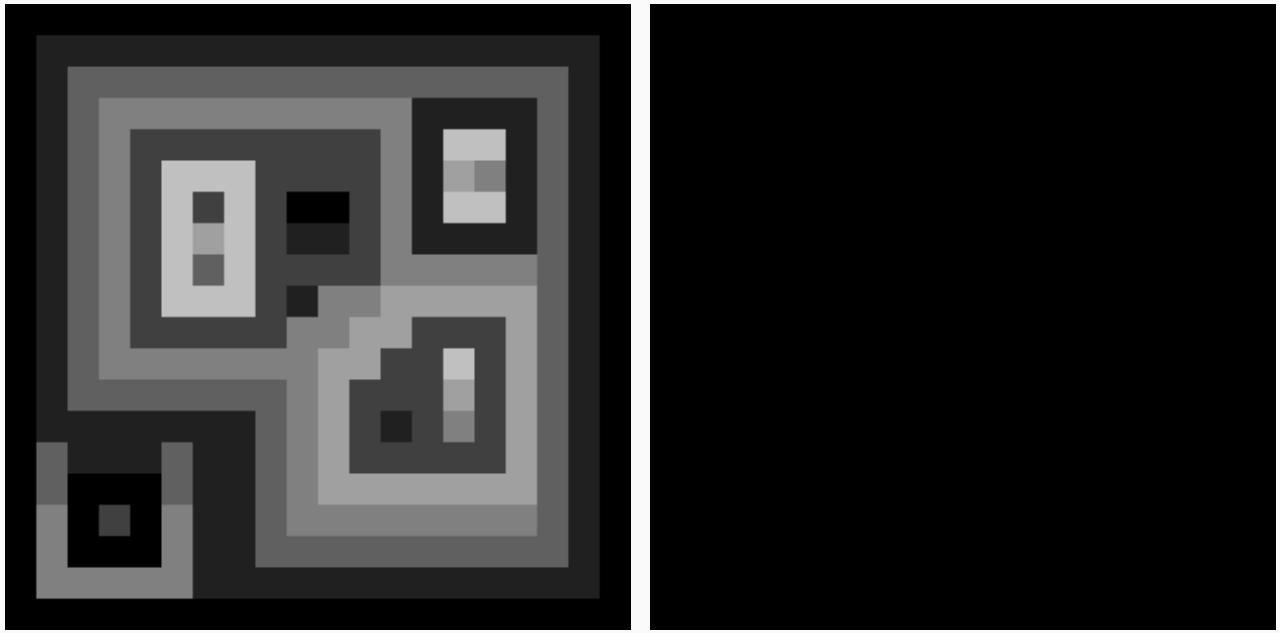
- $\Lambda_v^\circ(\mathcal{F}) \subseteq \mathbb{U}$: upper threshold set of $\mathcal{F}$ at value v (in white)
- $\Lambda_v^\bullet(\mathcal{F}) \subseteq \mathbb{U}$: lower threshold set of $\mathcal{F}$ at value v (in black)



Thresholding(s), binary images

Thresholding(s)

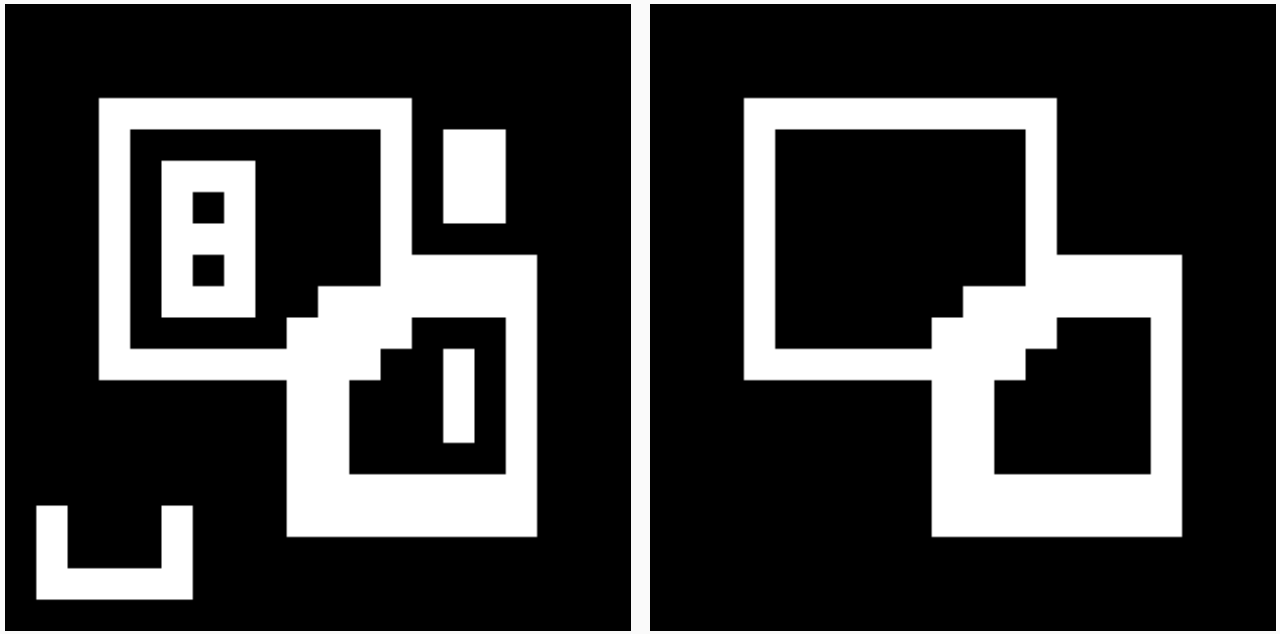
- $\Lambda_v^\circ(\mathcal{F}) \subseteq \mathbb{U}$: upper threshold set of $\mathcal{F}$ at value v (in white)
- $\Lambda_v^\bullet(\mathcal{F}) \subseteq \mathbb{U}$: lower threshold set of $\mathcal{F}$ at value v (in black)



Connected components (CCs)

Connected components of a threshold set

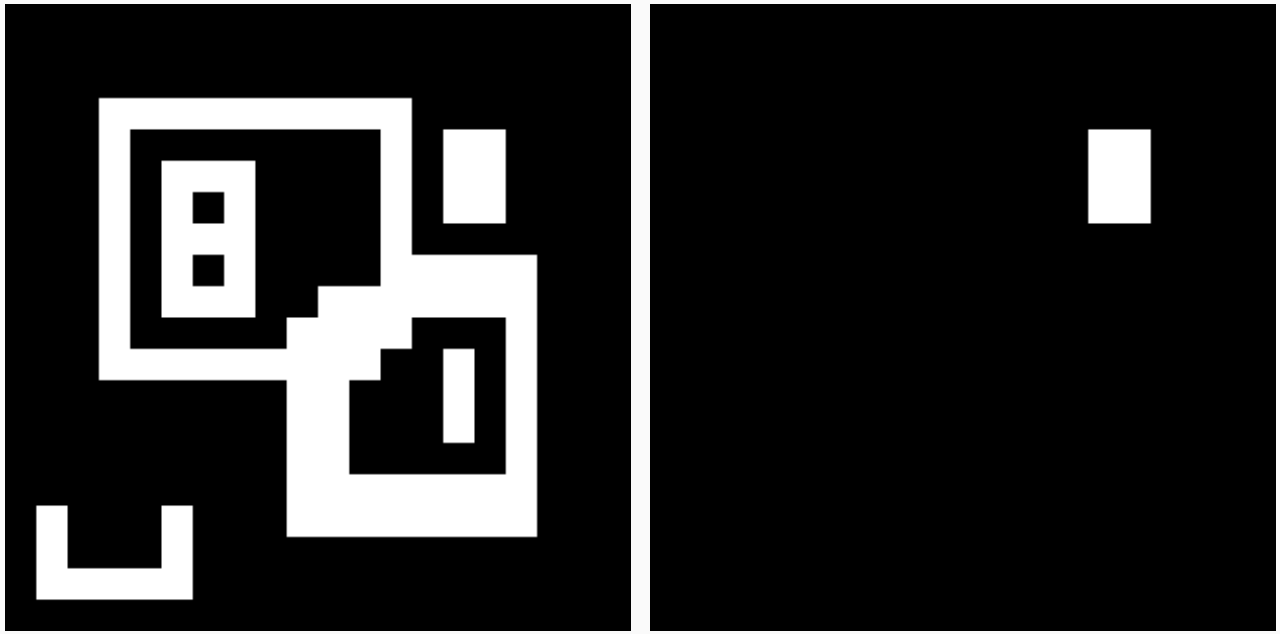
- $X \in 2^{\mathbb{U}}$: a CC of $\mathcal{F}$ at value v
- $\Theta \subset 2^{\mathbb{U}}$: the set of all the CCs of $\mathcal{F}$ at all values v



Connected components (CCs)

Connected components of a threshold set

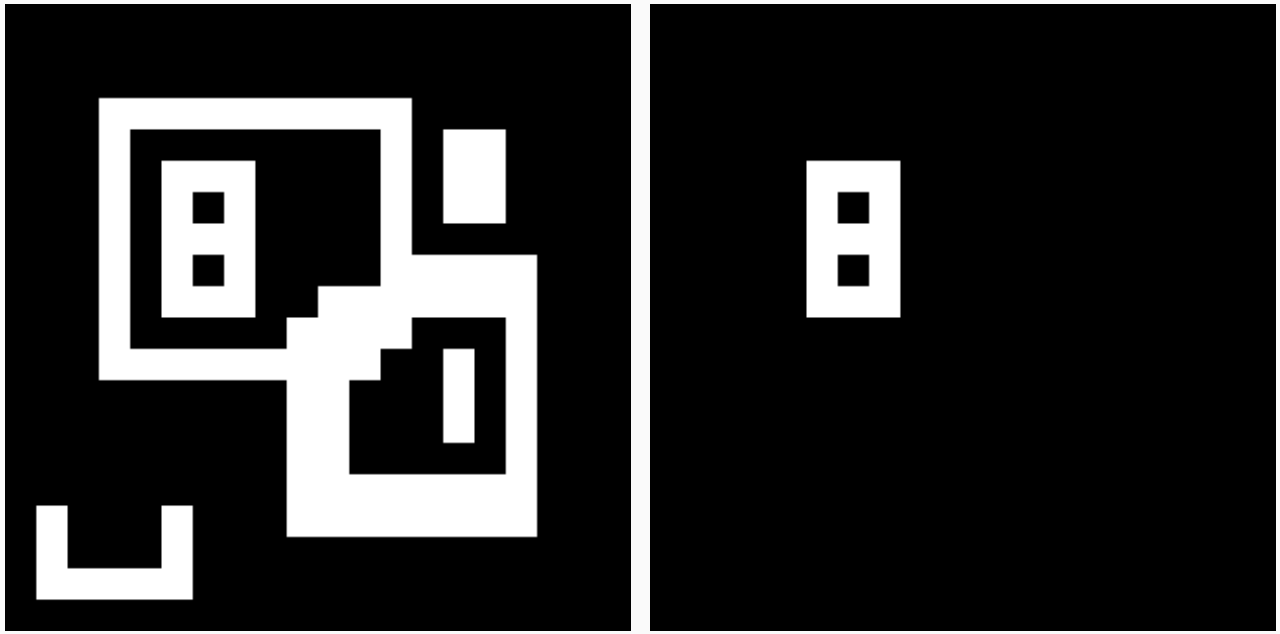
- $X \in 2^{\mathbb{U}}$: a CC of $\mathcal{F}$ at value v
- $\Theta \subset 2^{\mathbb{U}}$: the set of all the CCs of $\mathcal{F}$ at all values v



Connected components (CCs)

Connected components of a threshold set

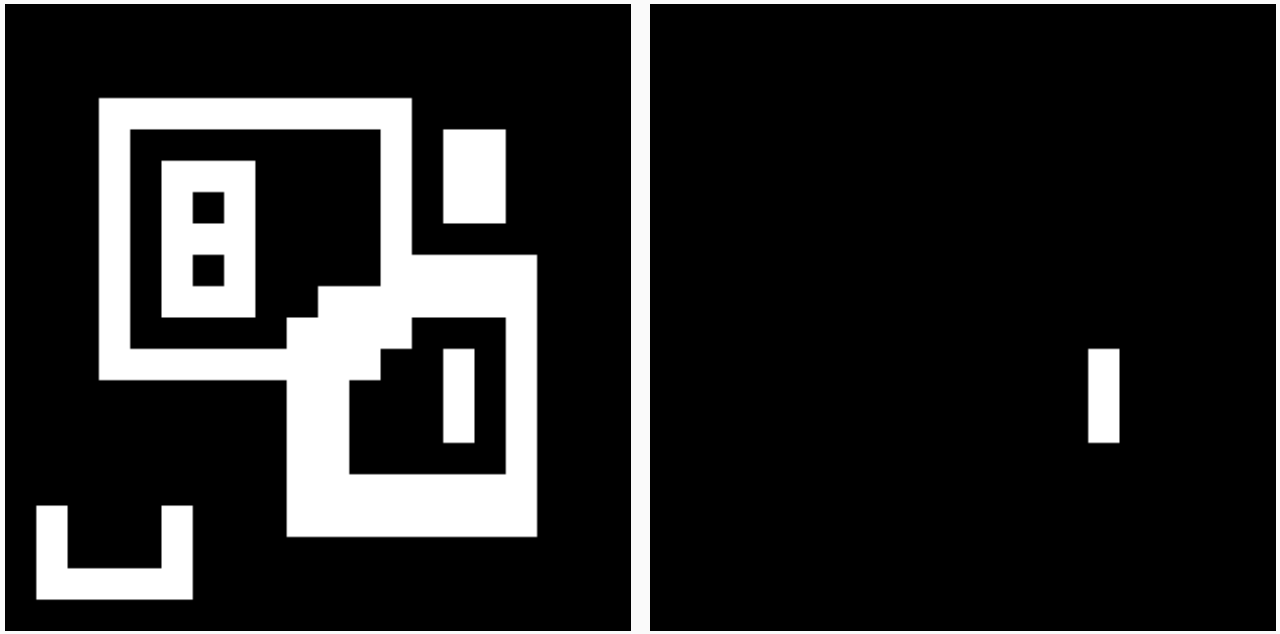
- $X \in 2^{\mathbb{U}}$: a CC of $\mathcal{F}$ at value v
- $\Theta \subset 2^{\mathbb{U}}$: the set of all the CCs of $\mathcal{F}$ at all values v



Connected components (CCs)

Connected components of a threshold set

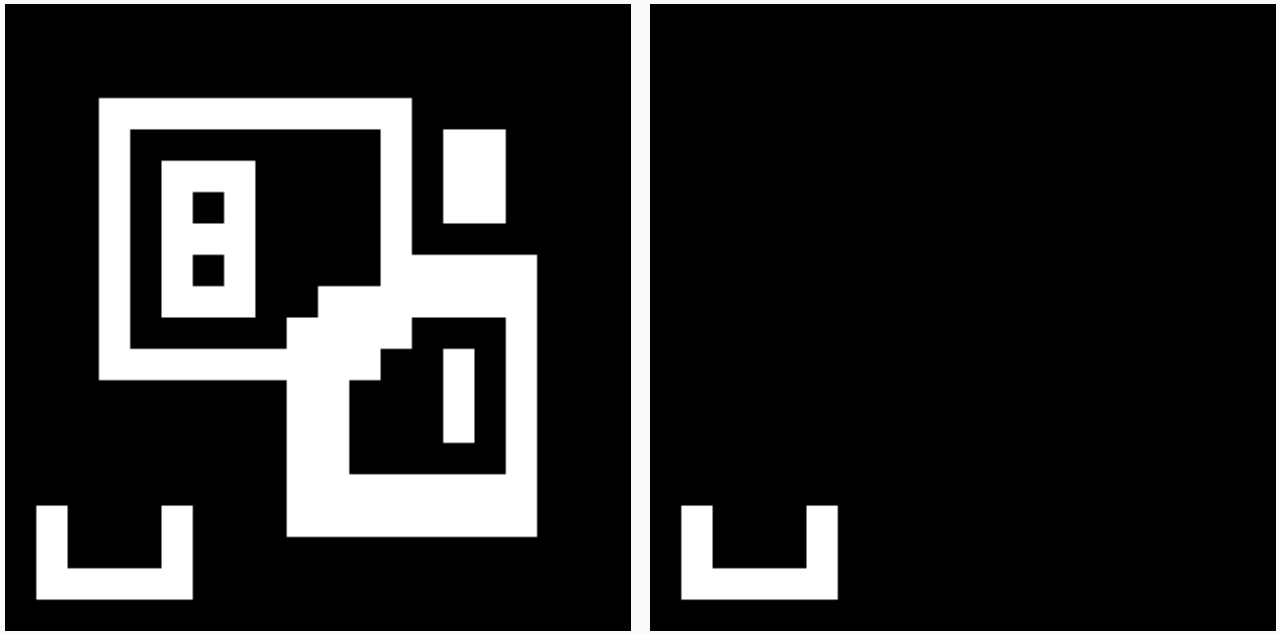
- $X \in 2^{\mathbb{U}}$: a CC of $\mathcal{F}$ at value v
- $\Theta \subset 2^{\mathbb{U}}$: the set of all the CCs of $\mathcal{F}$ at all values v



Connected components (CCs)

Connected components of a threshold set

- $X \in 2^{\mathbb{U}}$: a CC of $\mathcal{F}$ at value v
- $\Theta \subset 2^{\mathbb{U}}$: the set of all the CCs of $\mathcal{F}$ at all values v



Component-tree

Ordering on Θ

Two CCs $X, Y \in \Theta$ are either:

- non-intersecting: $X \cap Y = \emptyset$; or
- included one in the other: $X \subseteq Y$

$\Longrightarrow$ The partial ordering $\subseteq$ has a specific structure

$\longrightarrow$ Its Hasse diagram is a **tree**.

Component-tree

Ordering on Θ

Two CCs $X, Y \in \Theta$ are either:

- non-intersecting: $X \cap Y = \emptyset$; or
- included one in the other: $X \subseteq Y$

$\Longrightarrow$ The partial ordering $\subseteq$ has a specific structure

$\longrightarrow$ Its Hasse diagram is a **tree**.

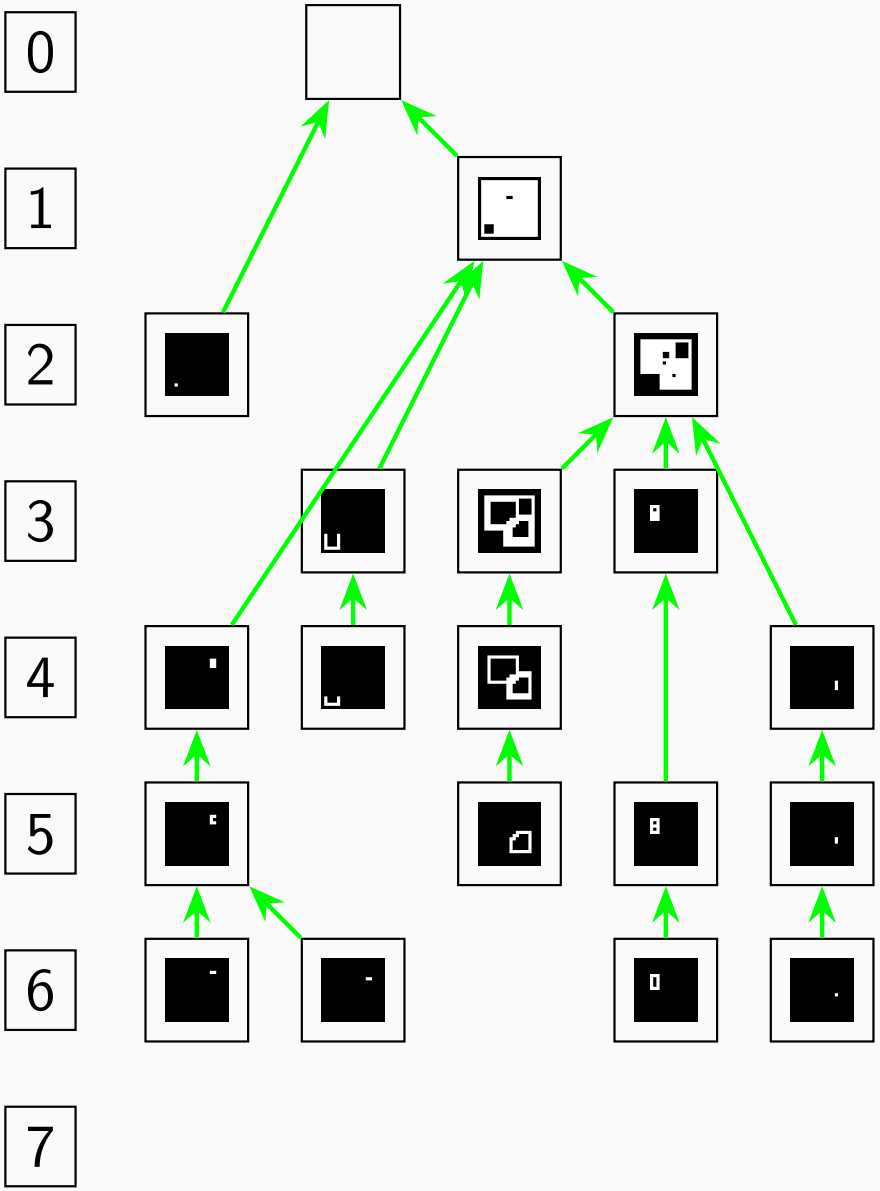
Component-tree

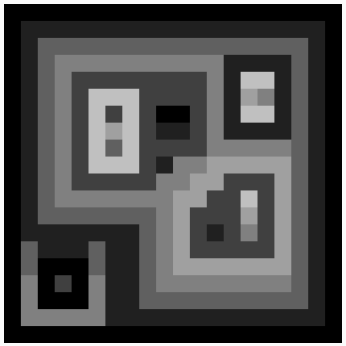
Component-tree = Hasse diagram of $(\Theta, \subseteq)$

If Θ is induced by:

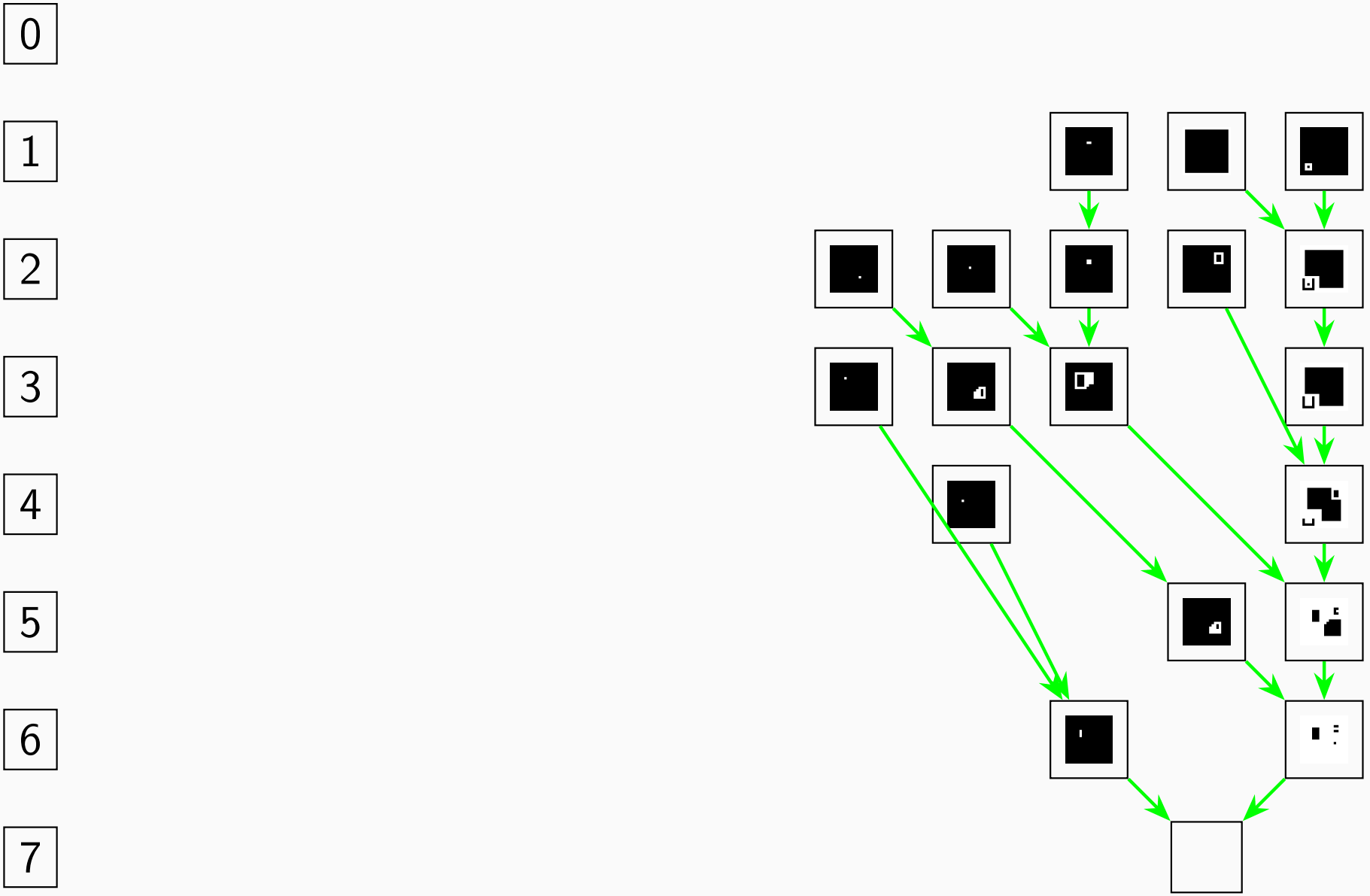
- upper thresholding $\rightarrow$ max-tree
- lower thresholding $\rightarrow$ min-tree

Component-tree: max-tree

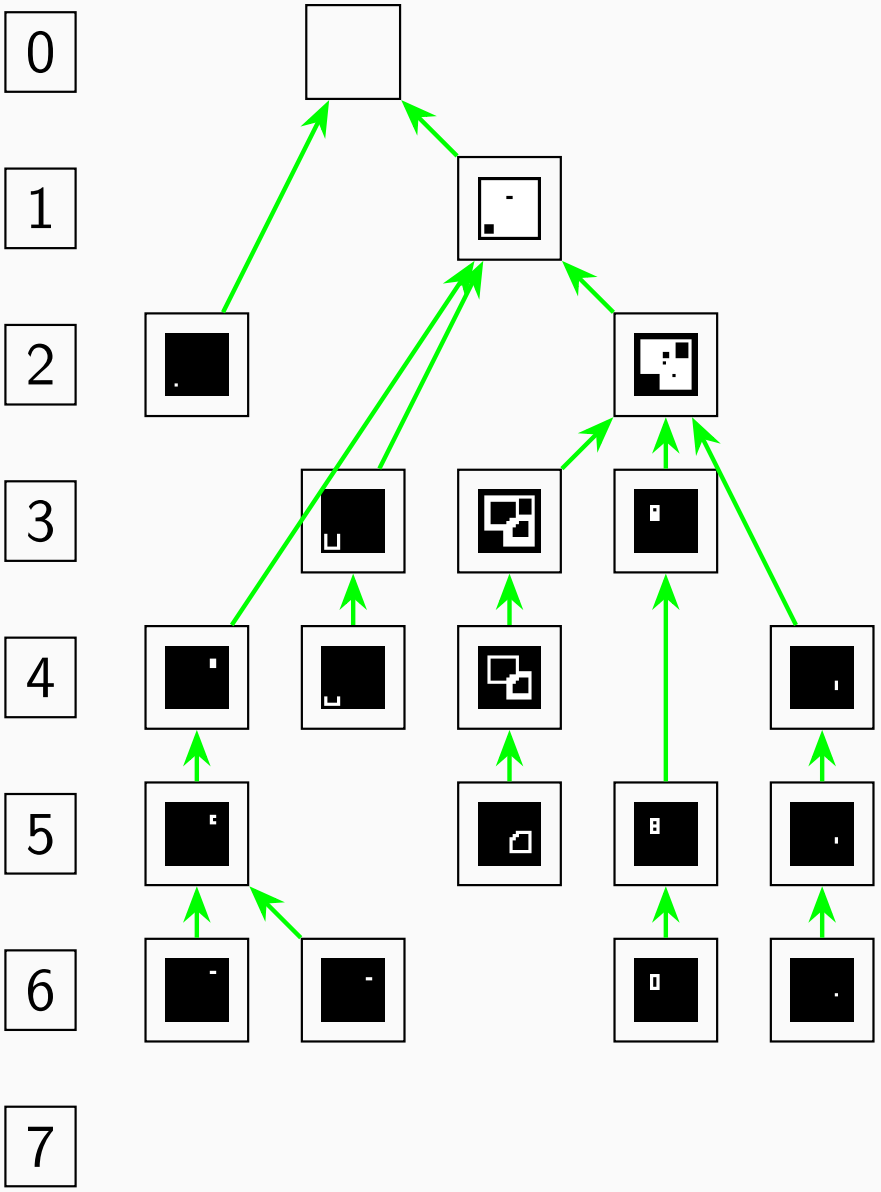


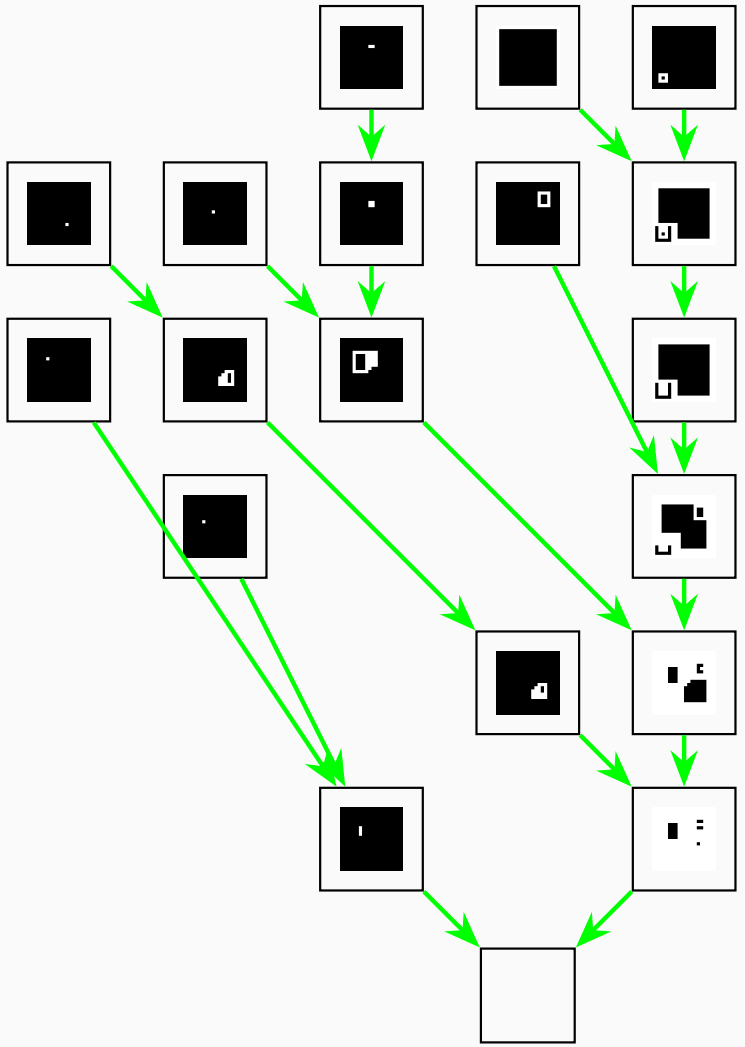


Component-tree: min-tree



Component-tree: max-tree and min-tree





Valued connected components

Connected components of a threshold set

- $X \in 2^{\mathbb{U}}$: a CC of $\mathcal{F}$ at value v
- $\Theta \subset 2^{\mathbb{U}}$: the set of all the CCs of $\mathcal{F}$ at all values v

We can enrich the CCs with the values at which they are defined.

Valued connected components

Connected components of a threshold set

- $X \in 2^{\mathbb{U}}$: a CC of $\mathcal{F}$ at value v
- $\Theta \subset 2^{\mathbb{U}}$: the set of all the CCs of $\mathcal{F}$ at all values v

We can enrich the CCs with the values at which they are defined.

Valued connected components of a threshold set

- $(X, v) \in 2^{\mathbb{U}} \times \mathbb{V}$: a valued CC of $\mathcal{F}$ at value v
- $\Xi \subset 2^{\mathbb{U}} \times \mathbb{V}$: the set of all the valued CCs of $\mathcal{F}$ at all values v

Ordering on valued connected components

Ordering on CCs

- CC: $X \in 2^{\mathbb{U}}$
  $\rightarrow$ Set of CCs: $\Theta \subseteq 2^{\mathbb{U}}$
  $\rightarrow$ Ordering on Θ : $X \subseteq Y$

Ordering on valued connected components

Ordering on CCs

- CC: $X \in 2^{\mathbb{U}}$
  $\rightarrow$ Set of CCs: $\Theta \subseteq 2^{\mathbb{U}}$
  $\rightarrow$ Ordering on Θ : $X \subseteq Y$

Ordering on valued CCs

- Valued CC: $(X, v) \in 2^{\mathbb{U}} \times \mathbb{V}$
  $\rightarrow$ Set of CCs: $\Xi \subseteq 2^{\mathbb{U}} \times \mathbb{V}$
  $\rightarrow$ Ordering on Ξ :
  $((X, v) \sqsubseteq (Y, w)) \Leftrightarrow (X \subseteq Y \wedge w \leqslant v)$ (max-tree)
  $((X, v) \sqsubseteq (Y, w)) \Leftrightarrow (X \subseteq Y \wedge w \geqslant v)$ (min-tree)

Valued component-tree

Component-tree

Component-tree = Hasse diagram of $(\Theta, \subseteq)$

If Θ is generated by:

- upper thresholding $\rightarrow$ max-tree
- lower thresholding $\rightarrow$ min-tree

Valued component-tree

Component-tree

Component-tree = Hasse diagram of $(\Theta, \subseteq)$

If Θ is generated by:

- upper thresholding → max-tree
- lower thresholding → min-tree

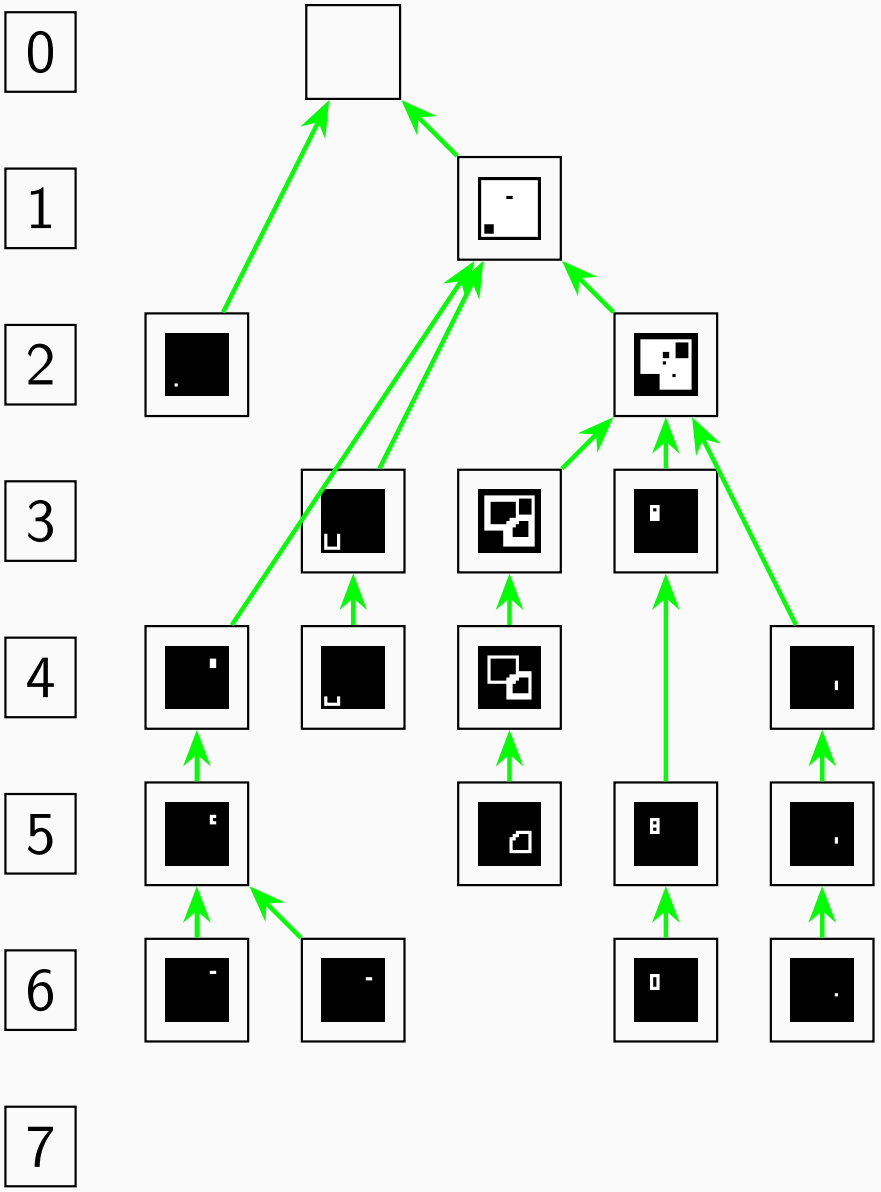
Valued component-tree

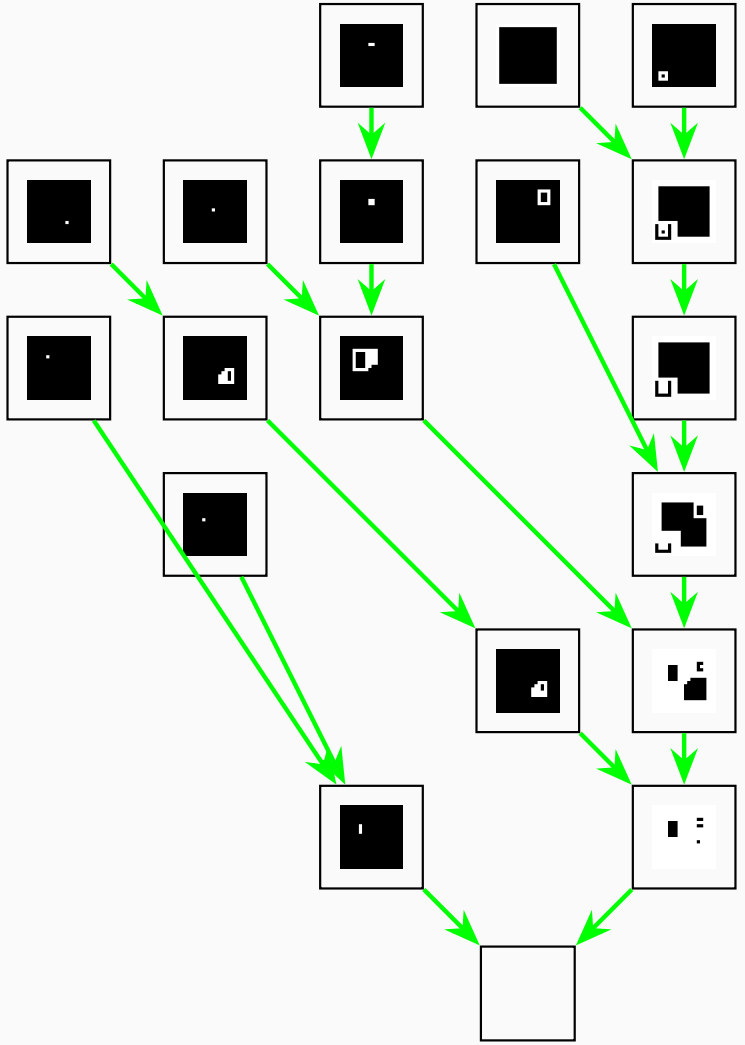
Valued component-tree = Hasse diagram of $(\Xi, \sqsubseteq)$

If Ξ is generated by:

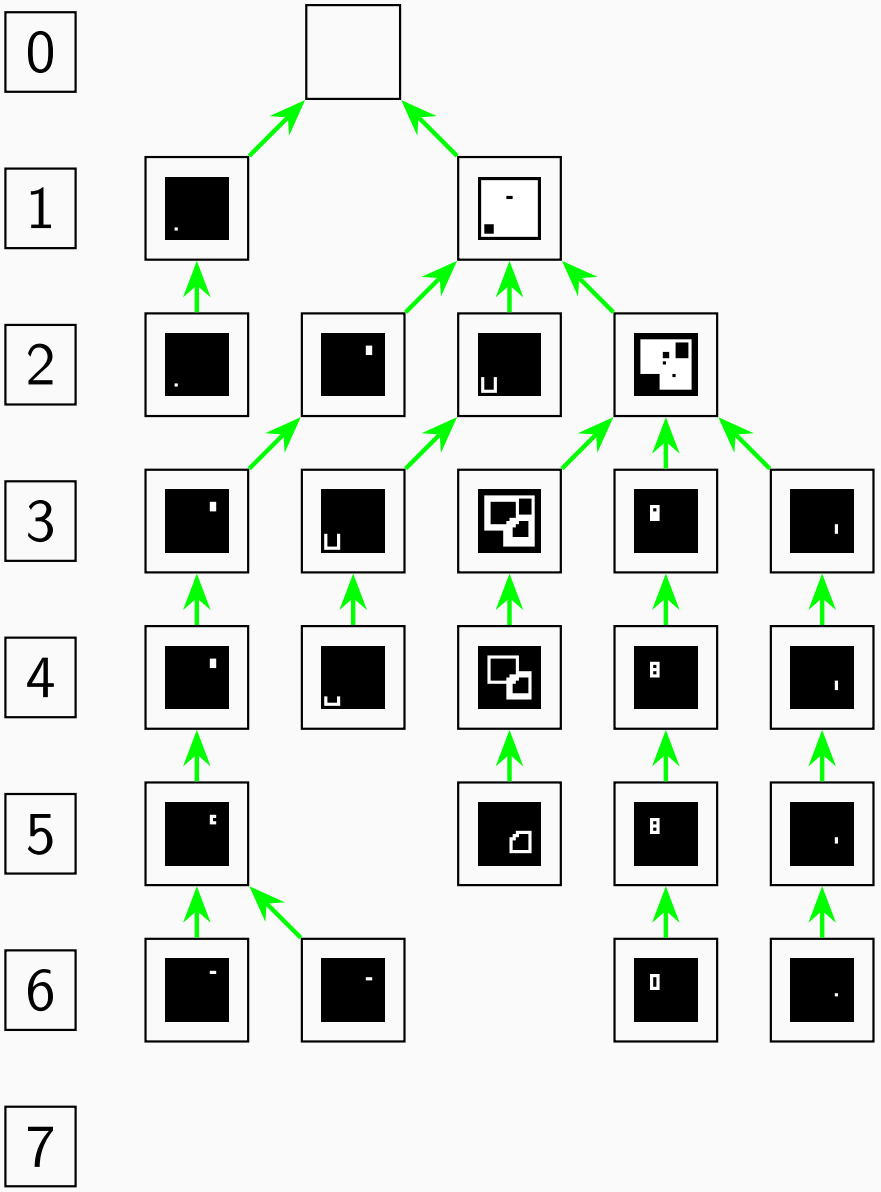
- upper thresholding → valued max-tree
- lower thresholding → valued min-tree

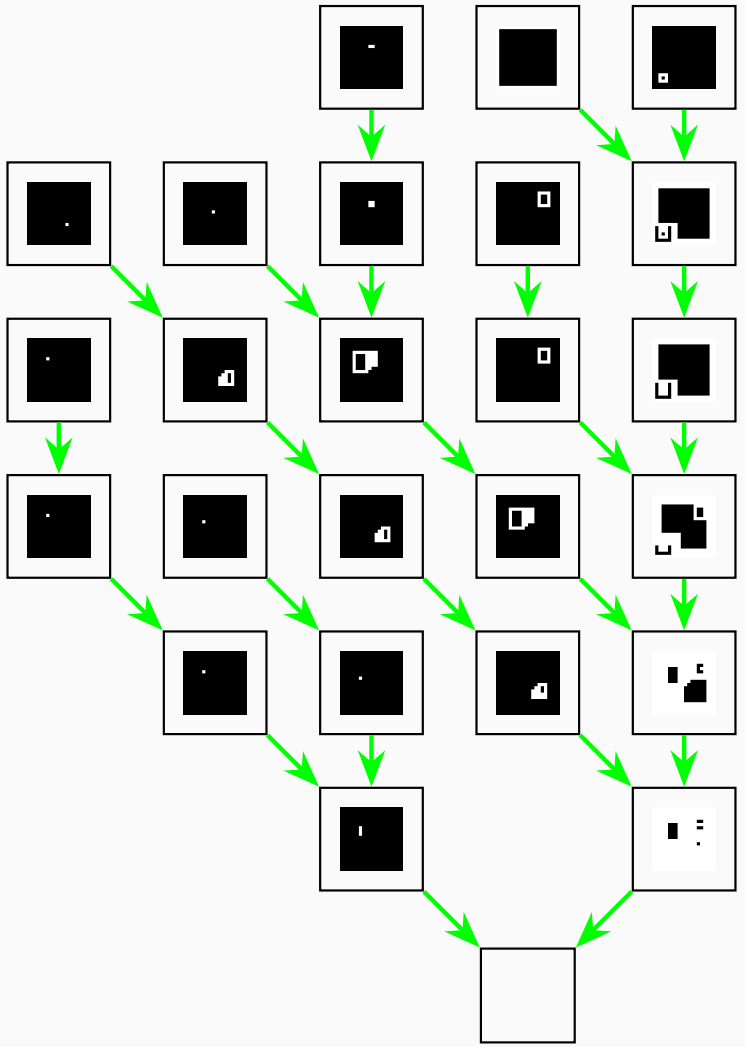
Component-tree: max-tree and min-tree





Valued component-tree: valued max-tree and valued min-tree





Component-tree as a topological descriptor

Topological description of images

Binary images

- Low-level: Euler characteristics, Betti numbers...
- High-level: homology groups, homotopy...
- Hierarchical: adjacency tree

Topological description of images

Binary images

- Low-level: Euler characteristics, Betti numbers...
- High-level: homology groups, homotopy...
- Hierarchical: adjacency tree

Grey-level images

- High-level: persistent homology
- Hierarchical: component-tree, tree of shapes

Persistent homology vs. Component-trees

Both compute binary images by thresholding, $\forall v \in \mathbb{V}$ ("timeline").
Then, for each binary image,

- **persistent homology** computes **homology groups**
- **component-trees** compute **CCs**

	Persistent homology	Component-trees
Foreground/background CCs	+	+
3D handles (tunnels)	+	−
Component creation/deletion	+	+
Component **merging**/**splitting**	−	+

Persistent homology vs. Component-trees

Both compute binary images by thresholding, $\forall v \in \mathbb{V}$ ("timeline").
Then, for each binary image,

- **persistent homology** computes **homology groups**
- **component-trees** compute **CCs**

	Persistent homology	Component-trees
Foreground/background CCs	+	+
3D handles (tunnels)	+	−
Component creation/deletion	+	+
Component **merging**/**splitting**	−	+

☺ **max-tree + min tree**: richer descriptor for grey-level images.

☹ No topological links between min- and max-trees.

Persistent homology vs. Component-trees

Both compute binary images by thresholding, $\forall v \in \mathbb{V}$ ("timeline").
Then, for each binary image,

- **persistent homology** computes **homology groups**
- **component-trees** compute **CCs**

	Persistent homology	Component-trees
Foreground/background CCs	+	+
3D handles (tunnels)	+	−
Component creation/deletion	+	+
Component **merging**/**splitting**	−	+

☺ **max-tree + min tree**: richer descriptor for grey-level images.

☹ No topological links between min- and max-trees.

Component-trees as topological descriptor: a step forward

Add topological links between valued min- and max-trees thanks to the notion of **adjacency-tree**.

Step 1 – Enrichment: Graph of valued shapes

Adjacency tree: a topological descriptor for binary images

Adjacency tree

- $B \subseteq \mathbb{U}$ is a binary image foreground and $\overline{B} = \mathbb{U} \setminus B$ its complement

Adjacency tree: a topological descriptor for binary images

Adjacency tree

- $B \subseteq \mathbb{U}$ is a binary image foreground and $\overline{B} = \mathbb{U} \setminus B$ its complement
- Θ_B the CCs of B and of $\overline{B}$

Adjacency tree: a topological descriptor for binary images

Adjacency tree

- $B \subseteq \mathbb{U}$ is a binary image foreground and $\overline{B} = \mathbb{U} \setminus B$ its complement
- Θ_B the CCs of B and of $\overline{B}$
- $\sqsubseteq^{\psi}$ the order relation "nested" on Θ_B
  $X \sqsubseteq^{\psi} Y \Leftrightarrow \tau(X) \subseteq \tau(Y)$ with τ the "hole closing" operator

Adjacency tree: a topological descriptor for binary images

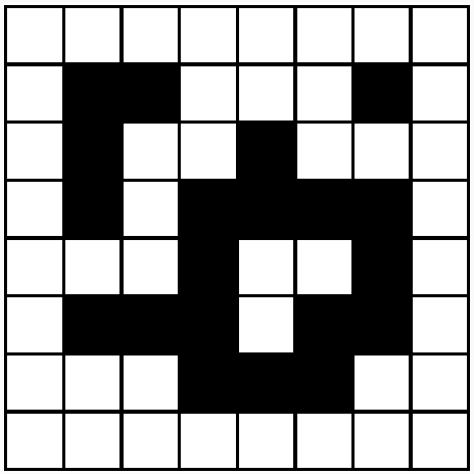
Adjacency tree

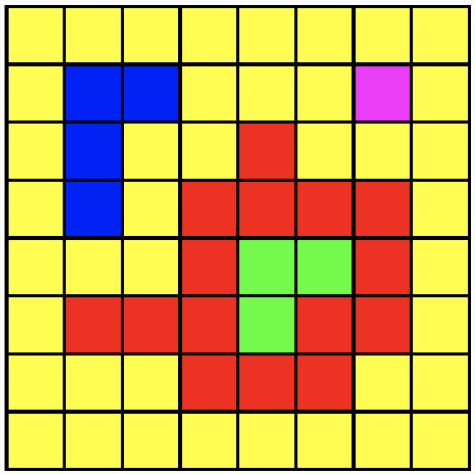
- $B \subseteq \mathbb{U}$ is a binary image foreground and $\overline{B} = \mathbb{U} \setminus B$ its complement
- Θ_B the CCs of B and of $\overline{B}$
- $\sqsubseteq^{\psi}$ the order relation "nested" on Θ_B
  $X \sqsubseteq^{\psi} Y \Leftrightarrow \tau(X) \subseteq \tau(Y)$ with τ the "hole closing" operator
- Adjacency tree of B = Hasse diagram of $(\Theta_B, \sqsubseteq^{\psi})$

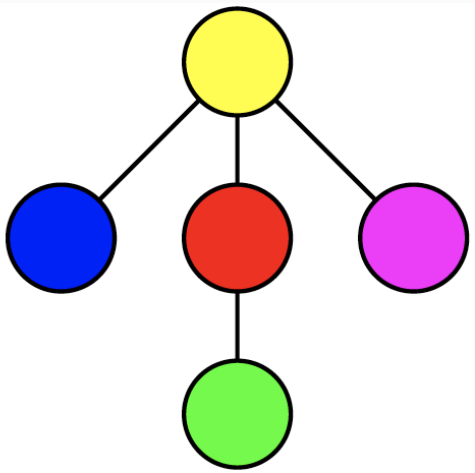
Adjacency tree: a topological descriptor for binary images

Adjacency tree

- $B \subseteq \mathbb{U}$ is a binary image foreground and $\overline{B} = \mathbb{U} \setminus B$ its complement
- Θ_B the CCs of B and of $\overline{B}$
- $\sqsubseteq^{\psi}$ the order relation "nested" on Θ_B
  $X \sqsubseteq^{\psi} Y \Leftrightarrow \tau(X) \subseteq \tau(Y)$ with τ the "hole closing" operator
- Adjacency tree of B = Hasse diagram of $(\Theta_B, \sqsubseteq^{\psi})$







Graph of valued shapes: General idea

Idea: enriching the information of an image $\mathcal{F}$ carried by

- the valued min-tree
- the valued max-tree

Graph of valued shapes: General idea

Idea: enriching the information of an image $\mathcal{F}$ carried by

- the valued min-tree
- the valued max-tree

. . . with additional information

- carried by the adjacency tree
- for each binary image
- at each threshold set of $\mathcal{F}$

Graph of valued shapes: General idea

Idea: enriching the information of an image $\mathcal{F}$ carried by

- the valued min-tree
- the valued max-tree

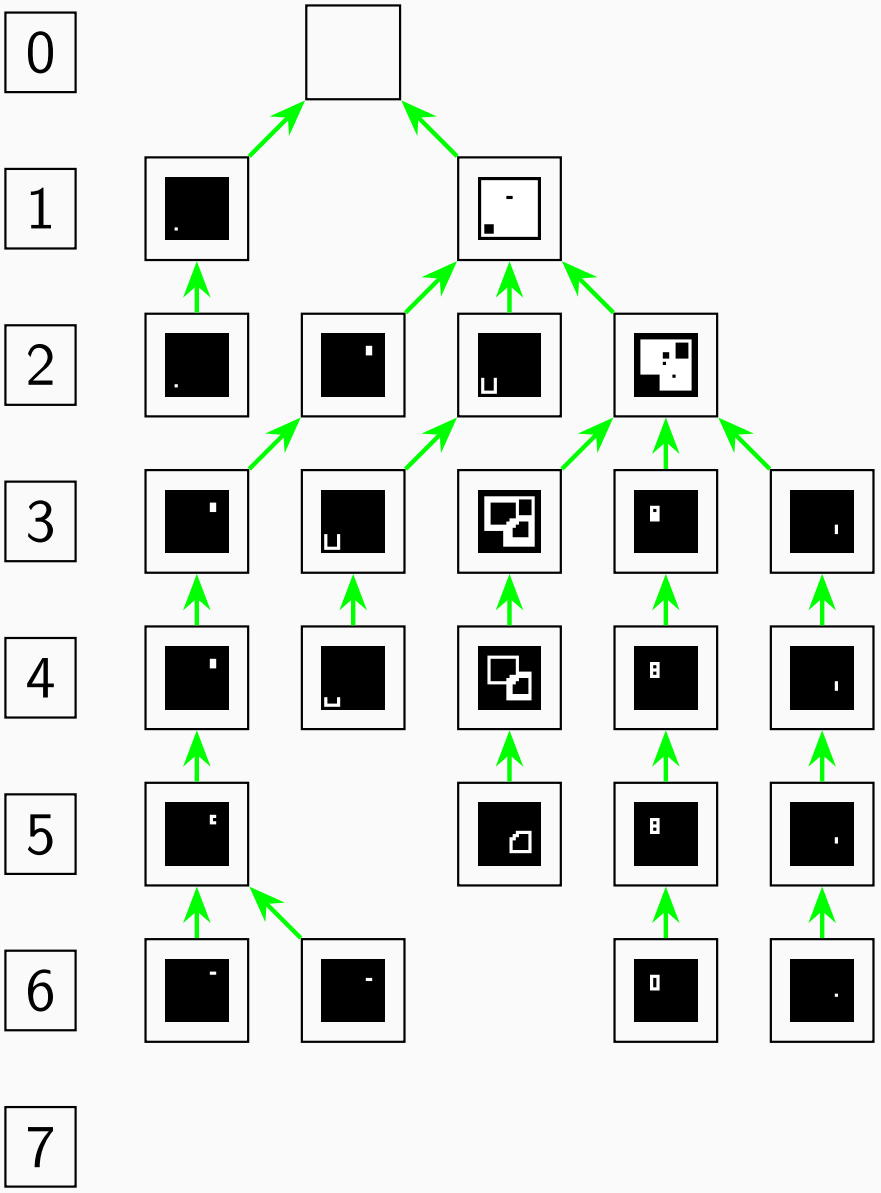
. . . with additional information

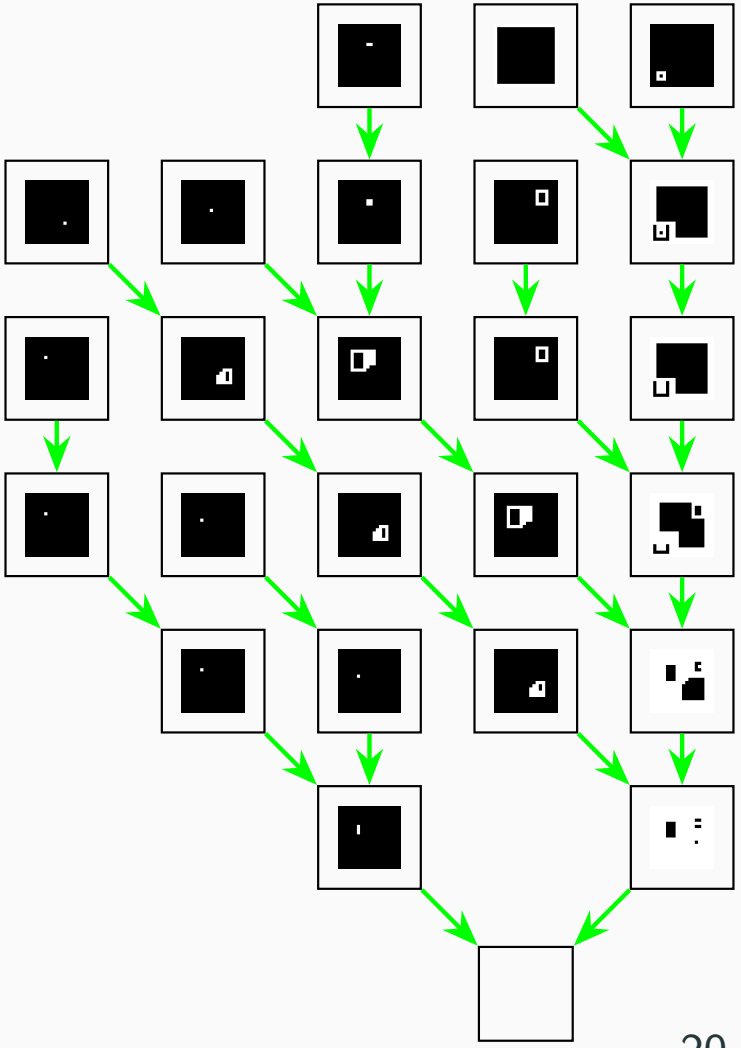
- carried by the adjacency tree
- for each binary image
- at each threshold set of $\mathcal{F}$

In other words

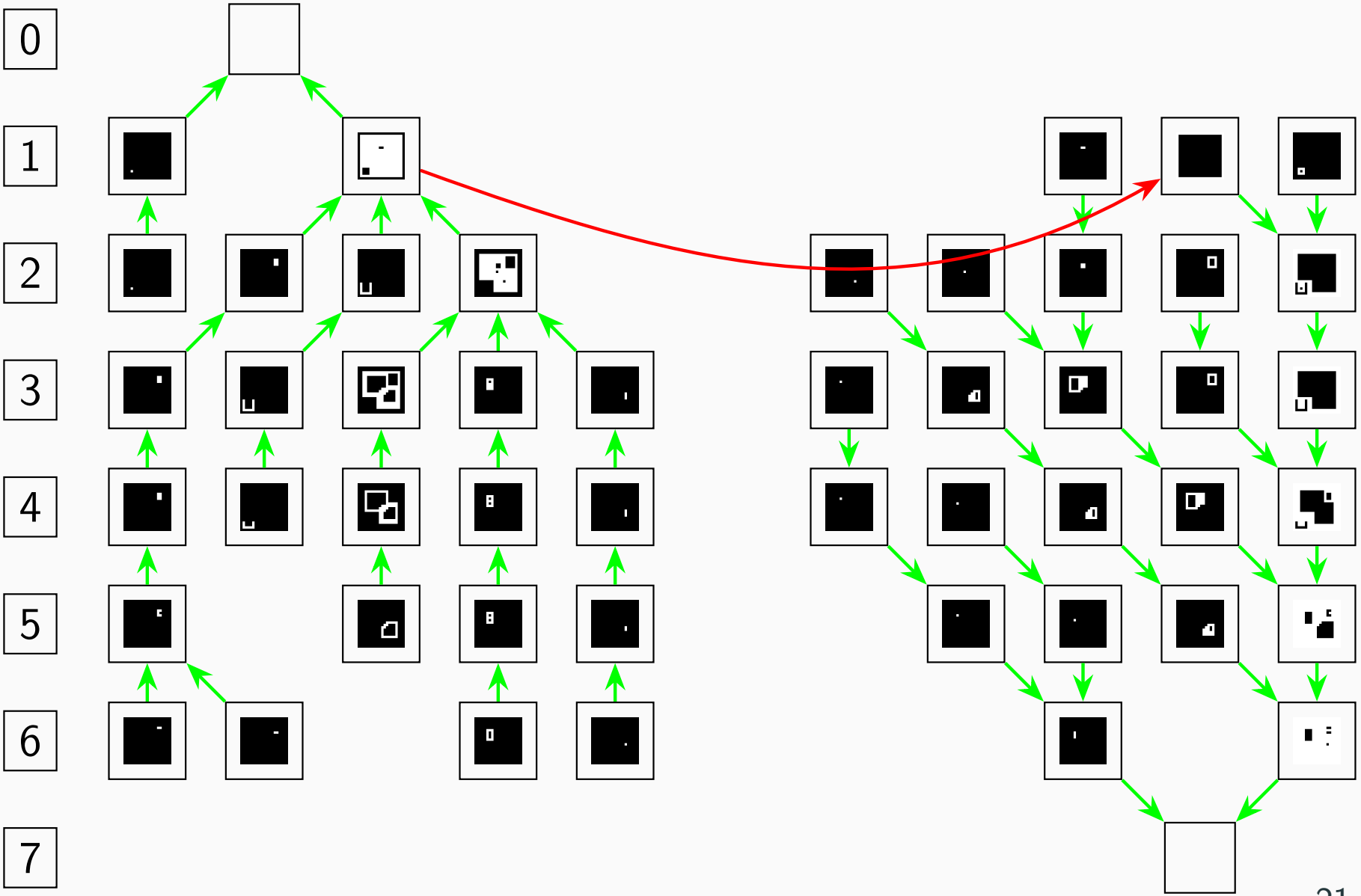
Modeling topological relations between the valued min- and max-trees by the adjacency trees

Valued min-tree + valued max-tree

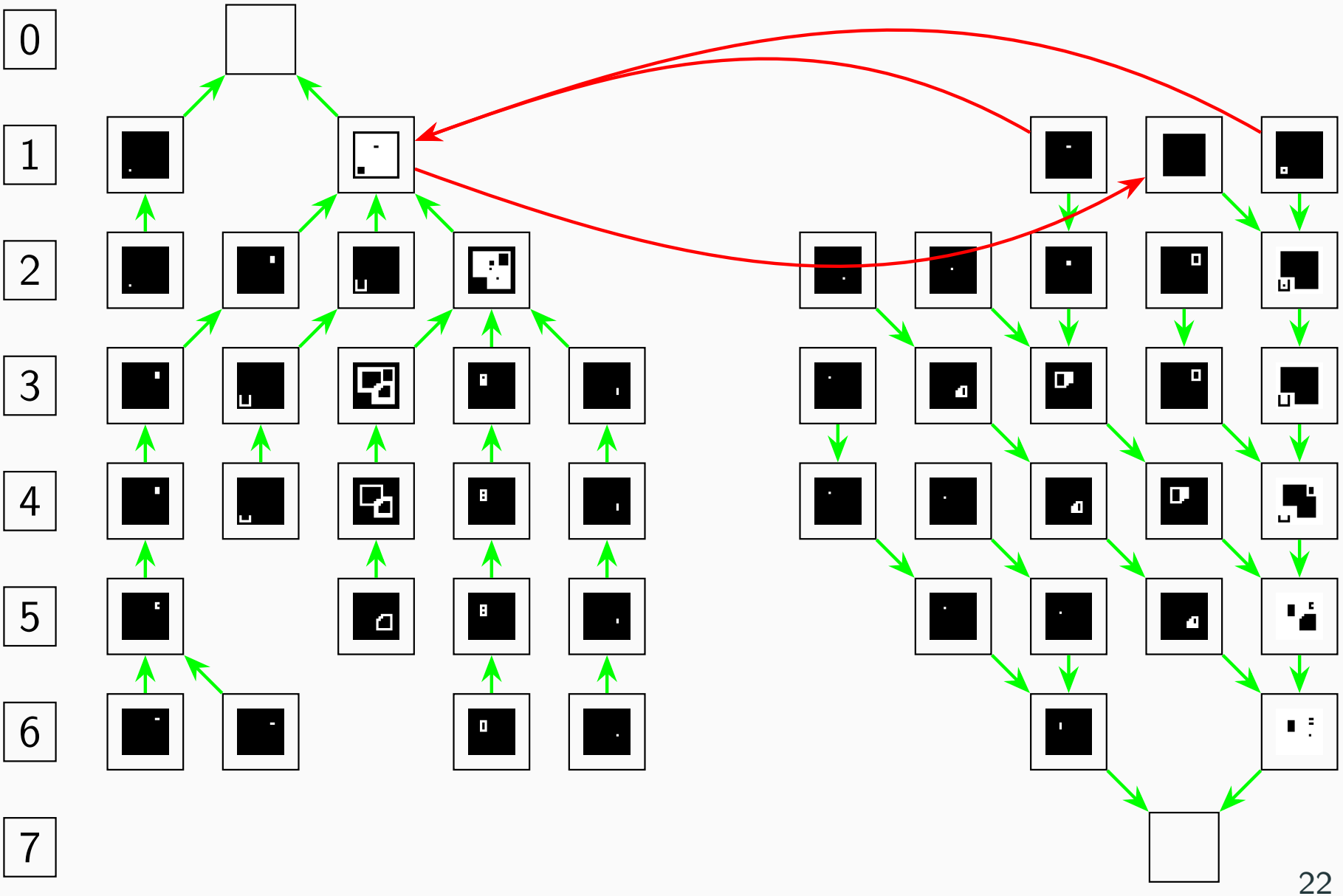




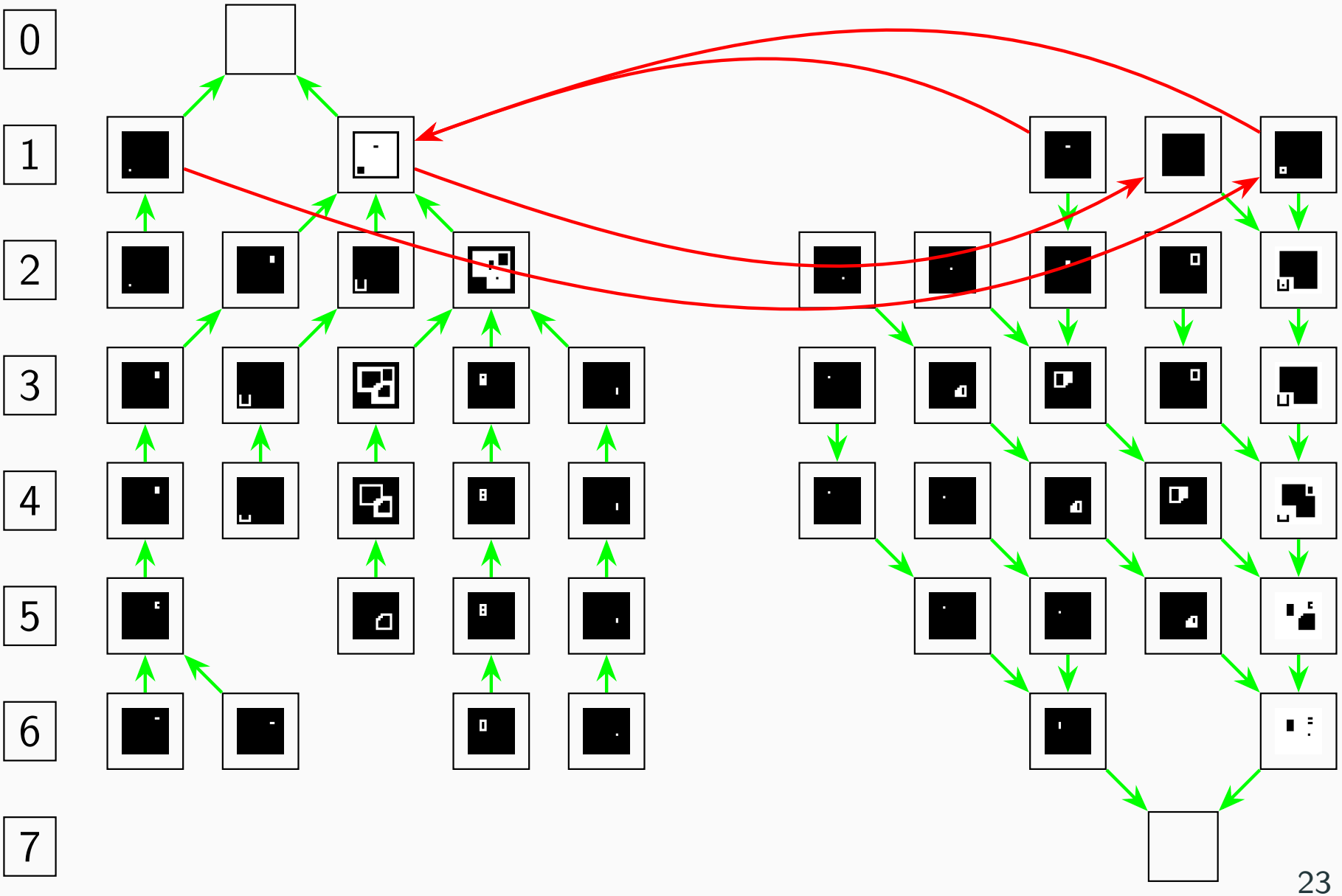
Valued min-tree + valued max-tree + 1 adjacency tree



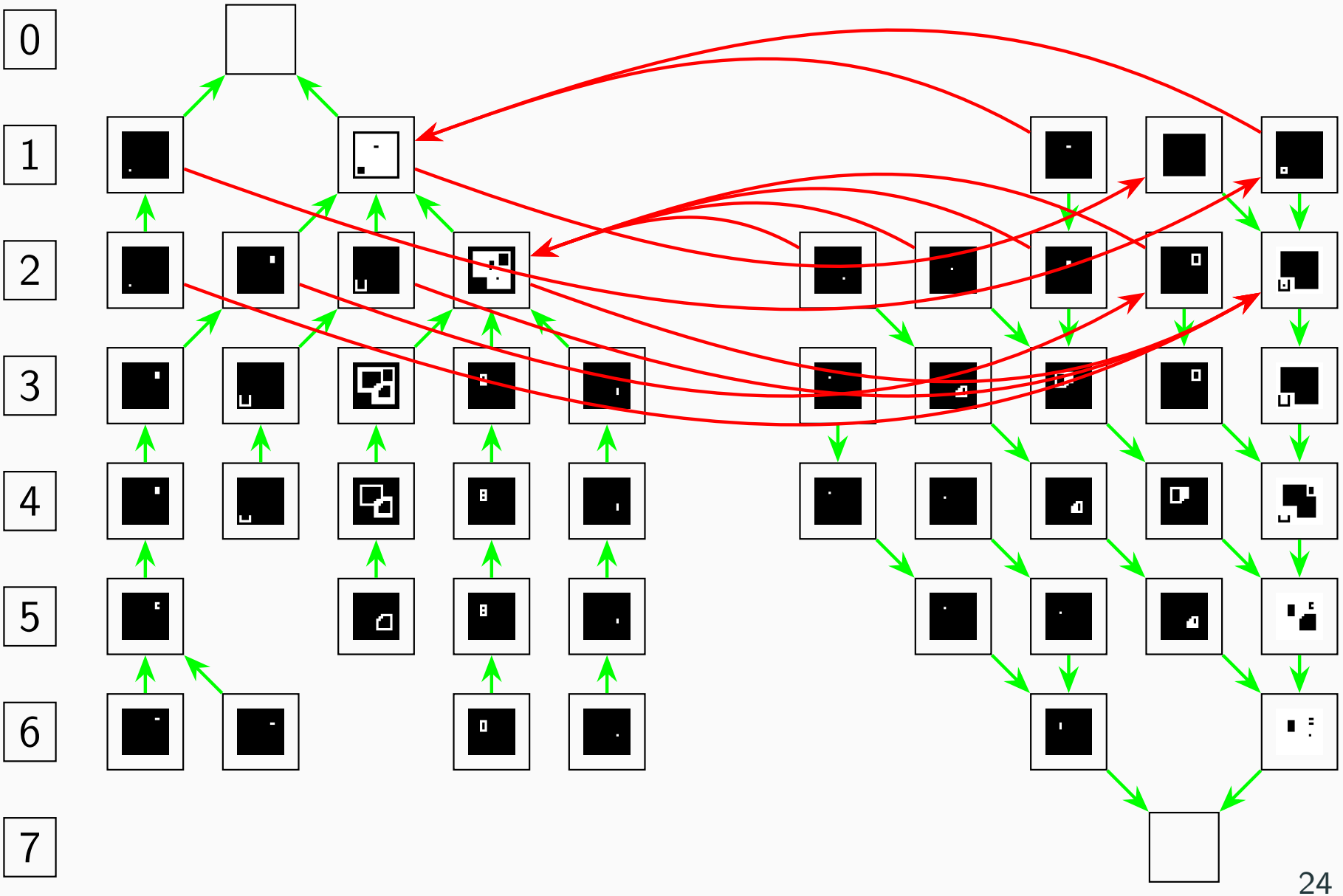
Valued min-tree + valued max-tree + 1 adjacency tree



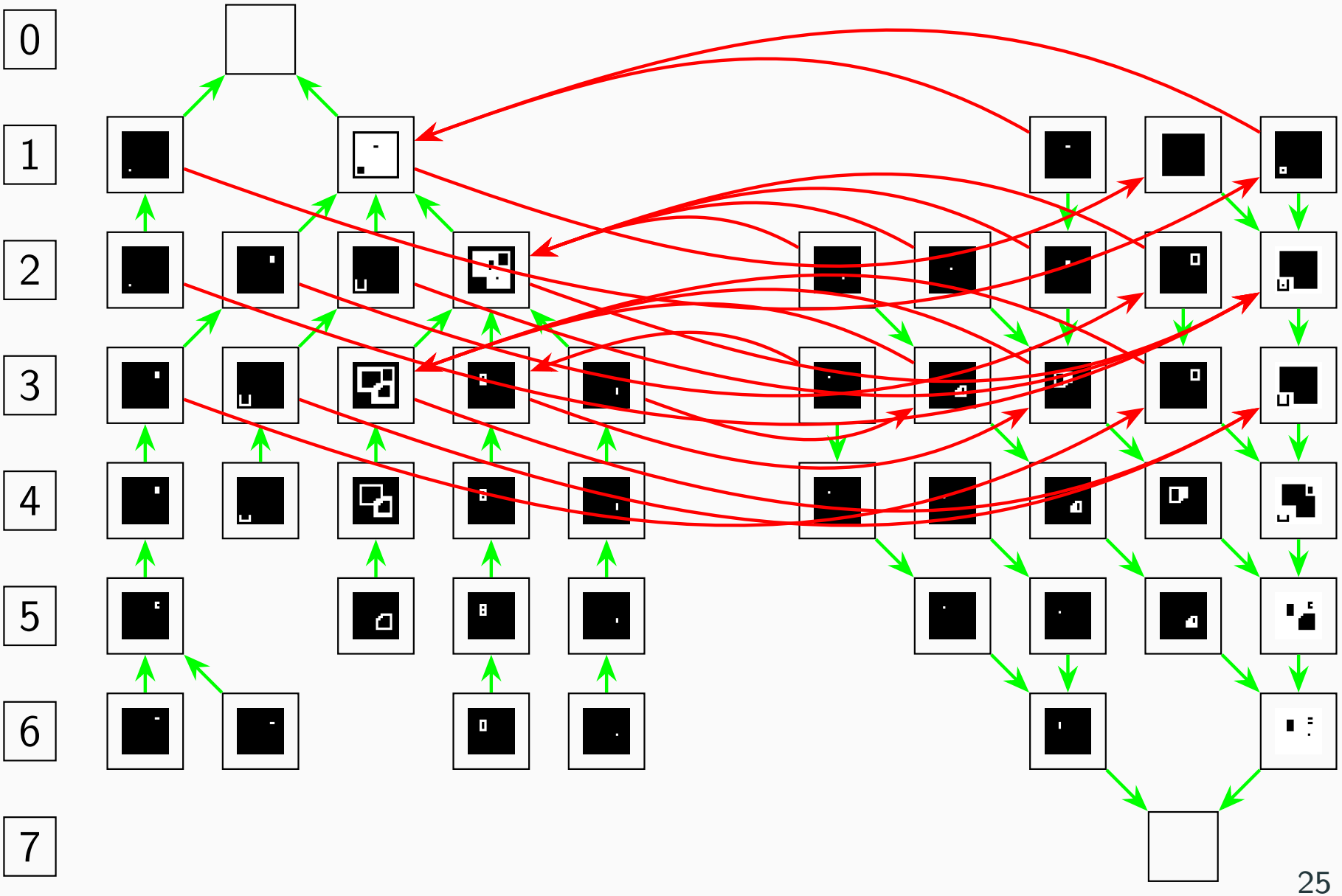
Valued min-tree + valued max-tree + 1 adjacency tree



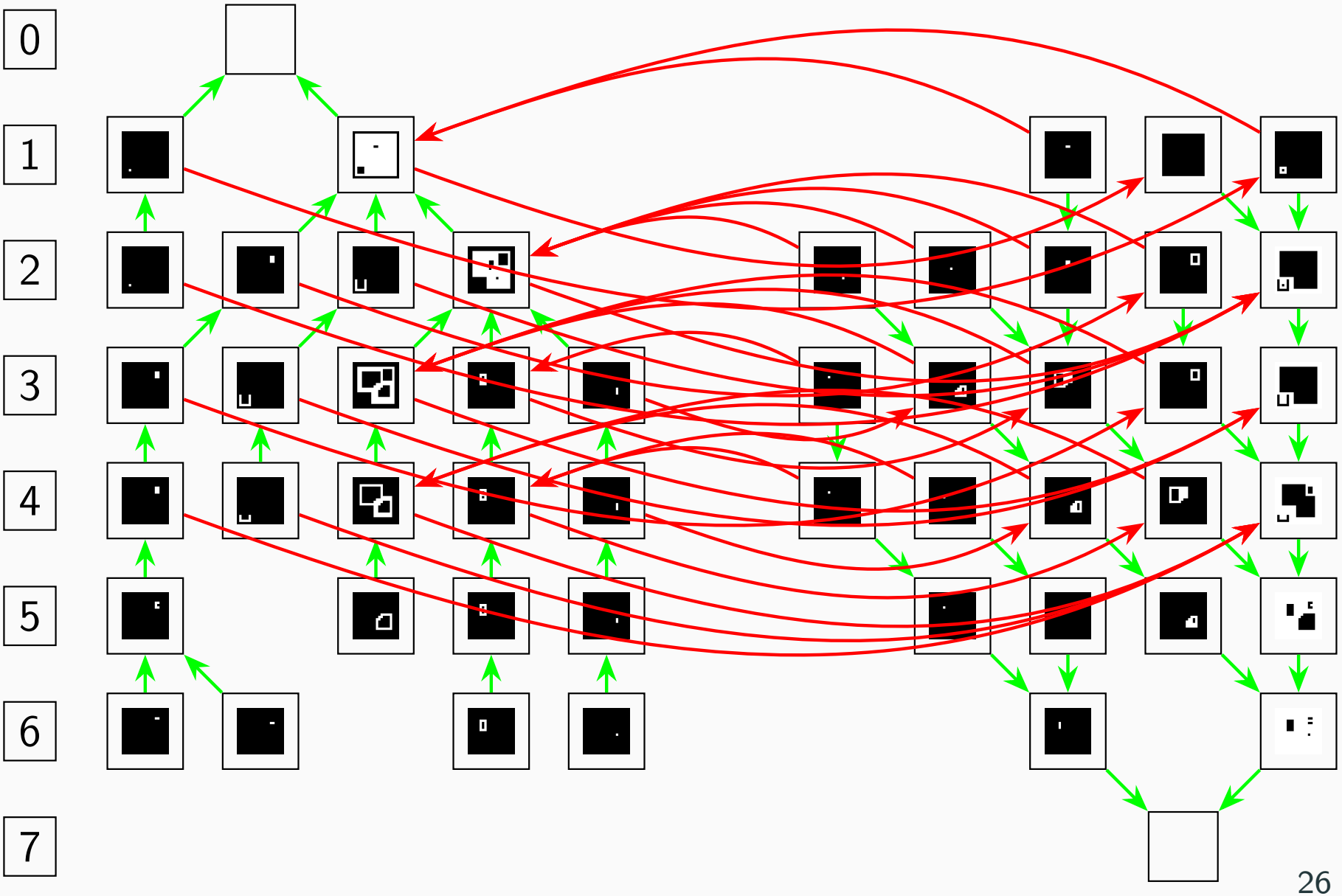
Valued min-tree + valued max-tree + 2 adjacency trees



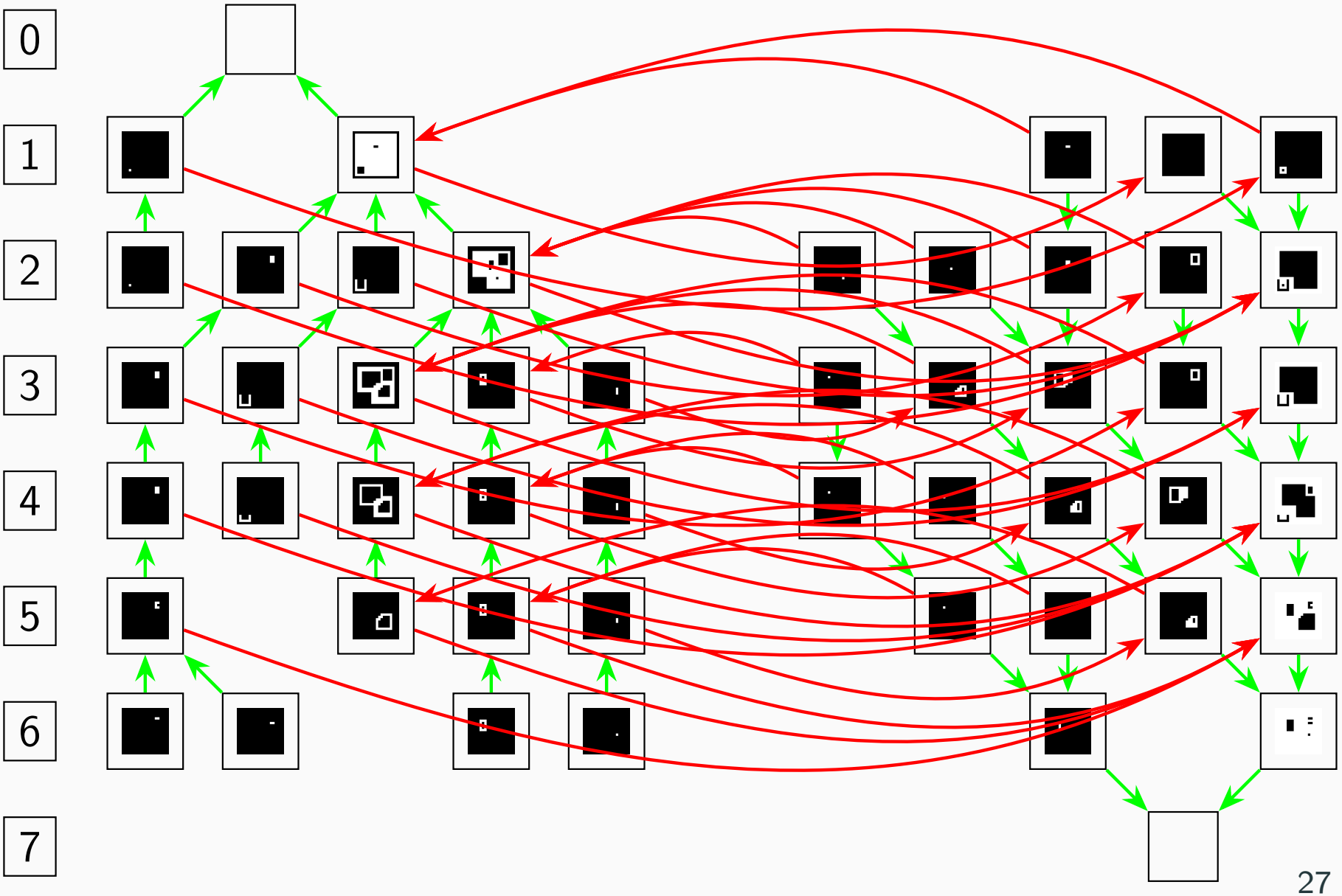
Valued min-tree + valued max-tree + 3 adjacency trees



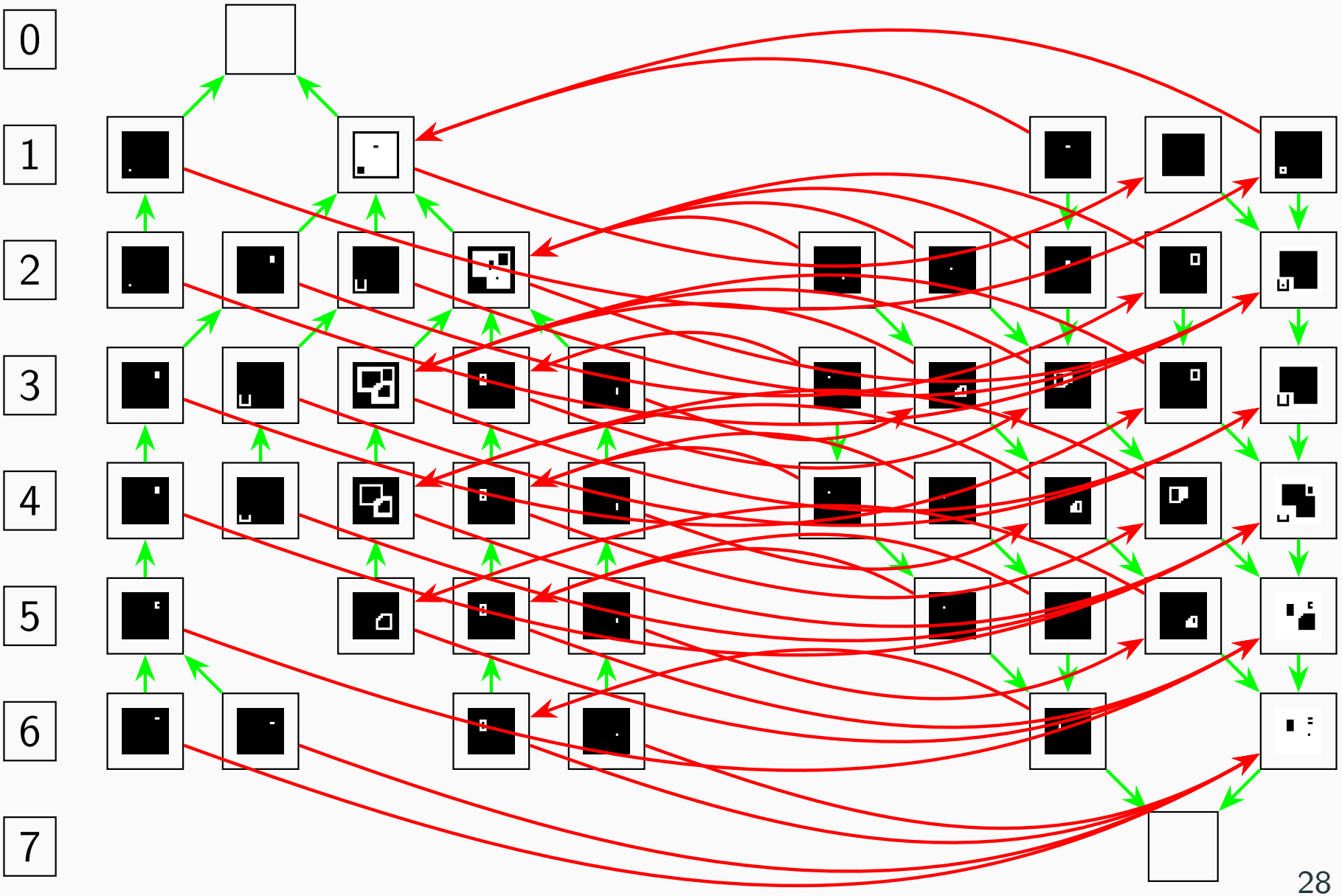
Valued min-tree + valued max-tree + 4 adjacency trees



Valued min-tree + valued max-tree + 5 adjacency trees



Valued min-tree + valued max-tree + $\sum$ adjacency trees



Graph of valued shapes: structure

The graph of valued shapes is modeled as $(\Xi, \lhd_\Xi)$ with

- Ξ = all the valued CCs of the valued min- and max-trees
- $\lhd^\varphi$ the edges of the min-tree and max-tree
- $\lhd^\psi$ the edges of all the the adjacency trees
- $\lhd_\Xi$ the union of $\lhd^\varphi$ and $\lhd^\psi$

Graph of valued shapes: structure

The graph of valued shapes is modeled as $(\Xi, \lhd_\Xi)$ with

- Ξ = all the valued CCs of the valued min- and max-trees
- $\lhd^\varphi$ the edges of the min-tree and max-tree
- $\lhd^\psi$ the edges of all the the adjacency trees
- $\lhd_\Xi$ the union of $\lhd^\varphi$ and $\lhd^\psi$

Bad news

The graph of valued shapes is rather complex!

Graph of valued shapes: structure

The graph of valued shapes is modeled as $(\Xi, \lhd_\Xi)$ with

- Ξ = all the valued CCs of the valued min- and max-trees
- $\lhd^\varphi$ the edges of the min-tree and max-tree
- $\lhd^\psi$ the edges of all the the adjacency trees
- $\lhd_\Xi$ the union of $\lhd^\varphi$ and $\lhd^\psi$

Good news

- The graph of valued shapes is a **directed acyclic graph (DAG)**
  $\Rightarrow$ It induces an order relation $\sqsubseteq_\Xi$
- We can **simplify** it...

Step 2 – Simplification: Tree of valued shapes

Structures of the graph of valued shapes

Structure around a node

Let

- $(\Xi, \lhd_\Xi)$ be a graph of valued shapes,
- $P = (X, v) \in \Xi$ be a node of the graph.

There exist:

Structures of the graph of valued shapes

Structure around a node

Let

- $(\Xi, \lhd_\Xi)$ be a graph of valued shapes,
- $P = (X, v) \in \Xi$ be a node of the graph.

There exist:

- **at most one** $Q \in \Xi$ such that $P \lhd^\varphi Q$
  $\longrightarrow$ "father" $Q = \varphi(P)$ of P in the valued min- or max-tree

Structures of the graph of valued shapes

Structure around a node

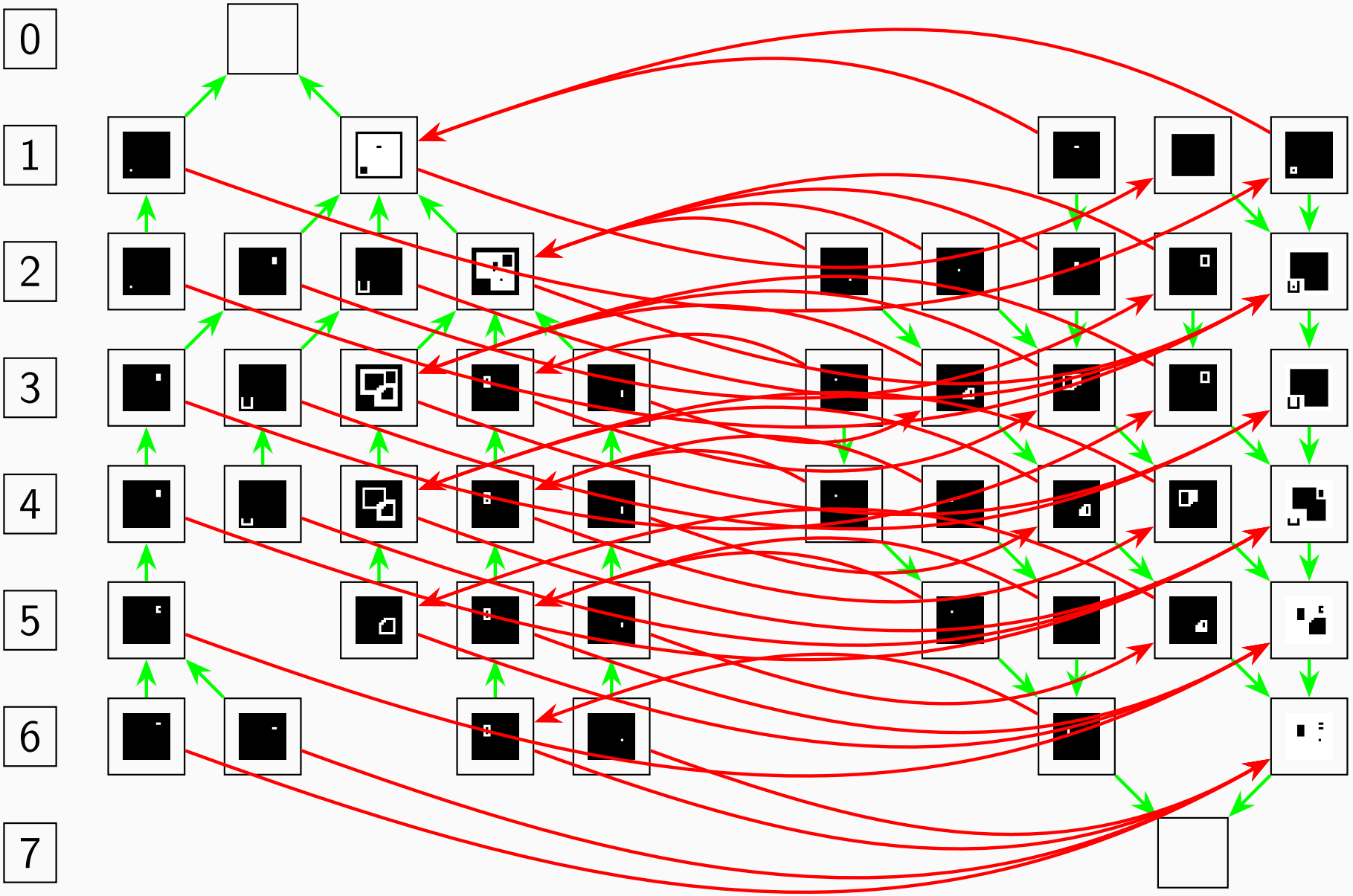
Let

- $(\Xi, \lhd_\Xi)$ be a graph of valued shapes,
- $P = (X, v) \in \Xi$ be a node of the graph.

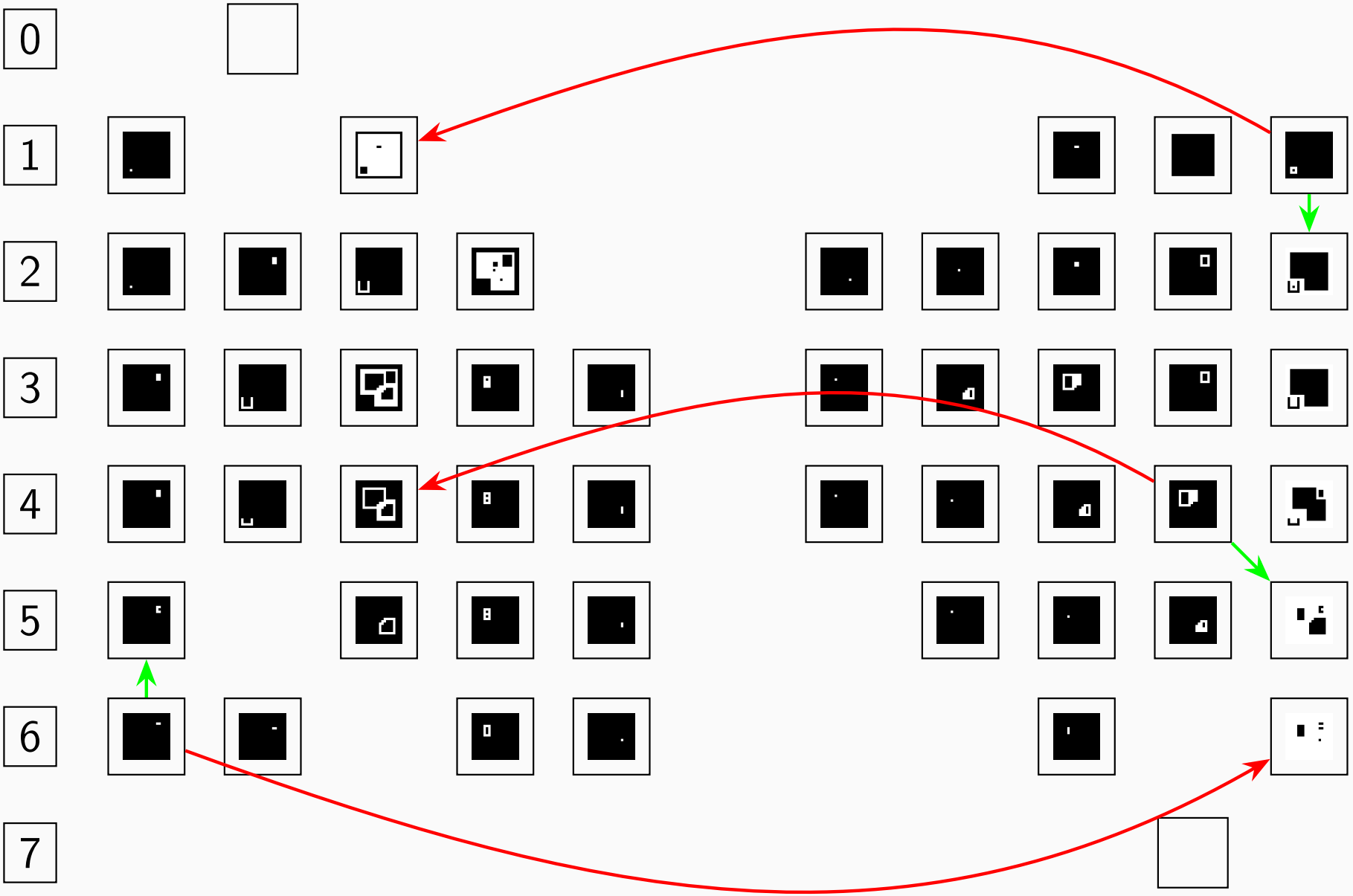
There exist:

- **at most one** $Q \in \Xi$ such that $P \lhd^\varphi Q$
  $\longrightarrow$ "father" $Q = \varphi(P)$ of P in the valued min- or max-tree
- **at most one** $R \in \Xi$ such that $P \lhd^\psi R$
  $\longrightarrow$ "father" $R = \psi(P)$ of P in the adjacency tree(s)

Examples: structures of the graph of valued shapes



Examples: structures of the graph of valued shapes



Transitive patterns: characterization

Redundant links

$$\psi(P) \longleftarrow P \longrightarrow \varphi(P)$$

If both edges exist, exactly one is redundant by transitivity

Transitive patterns: characterization

Redundant links

$$\psi(P) \longleftarrow P \longrightarrow \varphi(P)$$

If both edges exist, exactly one is redundant by transitivity

Three transitive patterns

1. if $\psi(P) = [\varphi \circ \psi \circ \varphi](P)$ then $\psi(P) \longleftarrow P$ is removed
2. if $\varphi(P) = [\varphi \circ \psi \circ \psi](P)$ then $P \longrightarrow \varphi(P)$ is removed
3. if $\varphi(P) = [\varphi^{|\mathbb{V}|-2} \circ \psi](P)$ then $P \longrightarrow \varphi(P)$ is removed

Transitive pattern 1

Redundant links

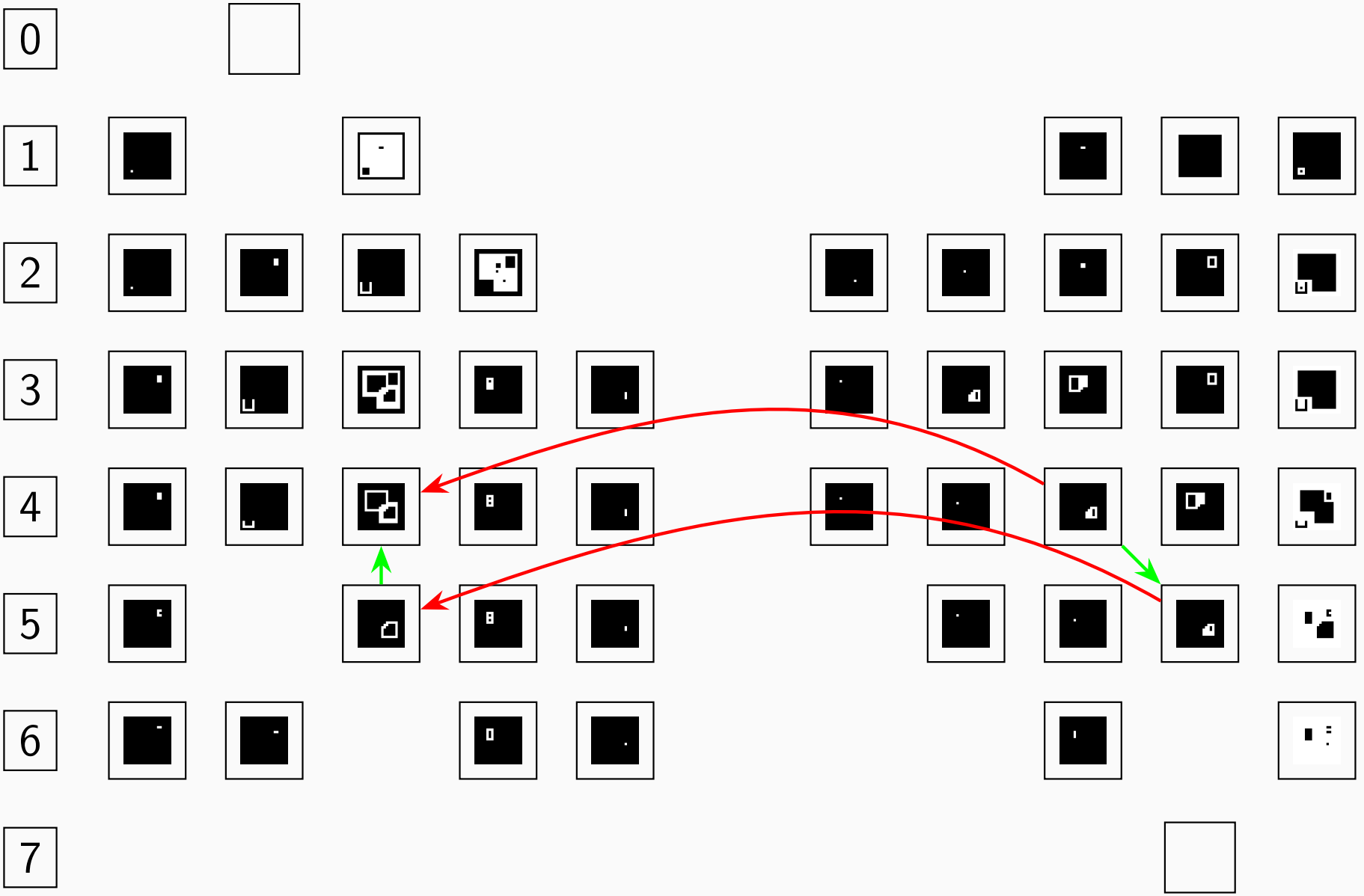
$$\psi(P) \longleftarrow P \longrightarrow \varphi(P)$$

If both edges exist, exactly one is redundant by transitivity

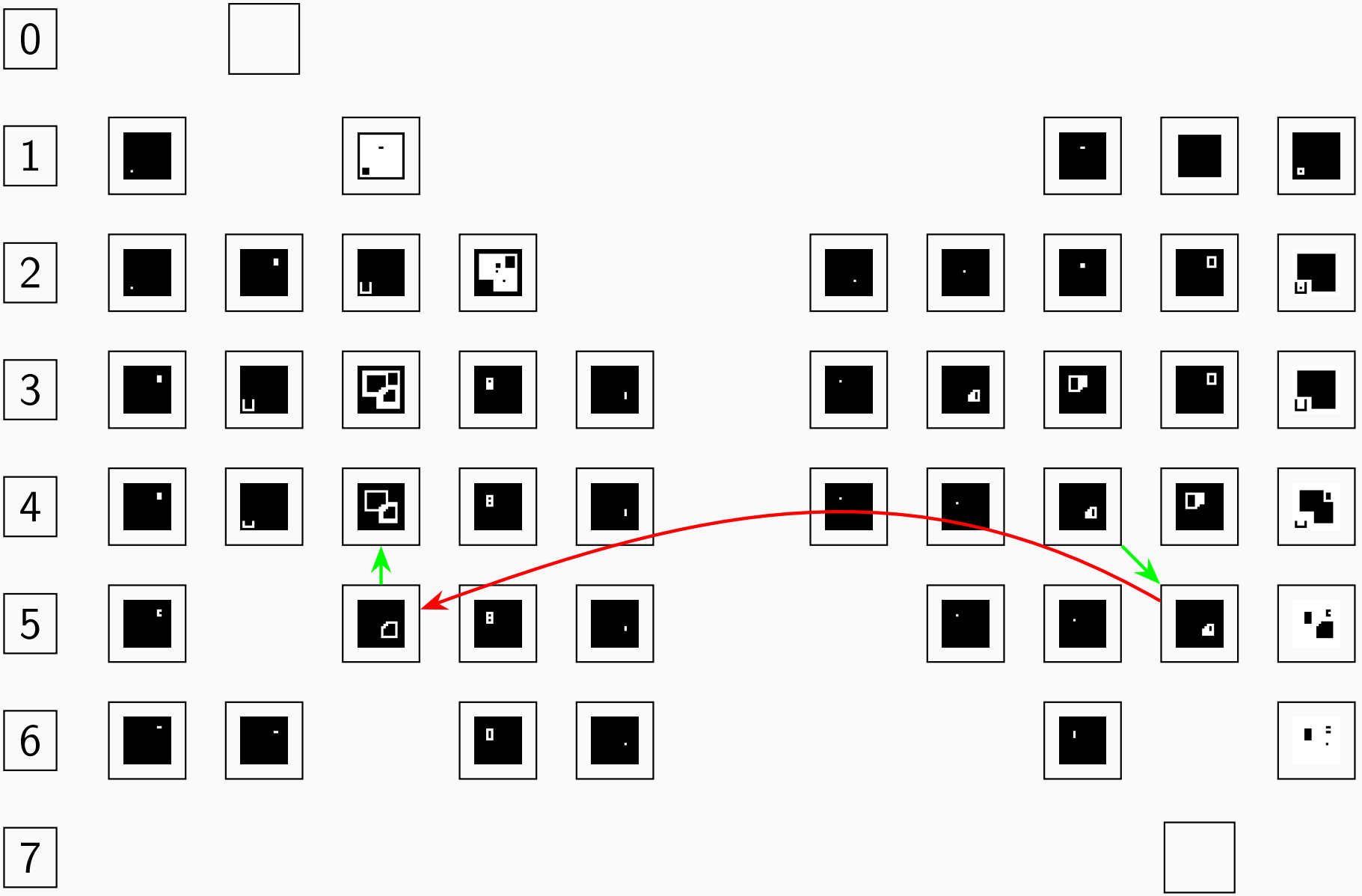
Three transitive patterns

1. if $\psi(P) = [\varphi \circ \psi \circ \varphi](P)$ then $\psi(P) \longleftarrow P$ is removed
2.
3.

Transitive pattern 1



Transitive pattern 1



Transitive pattern 2

Redundant links

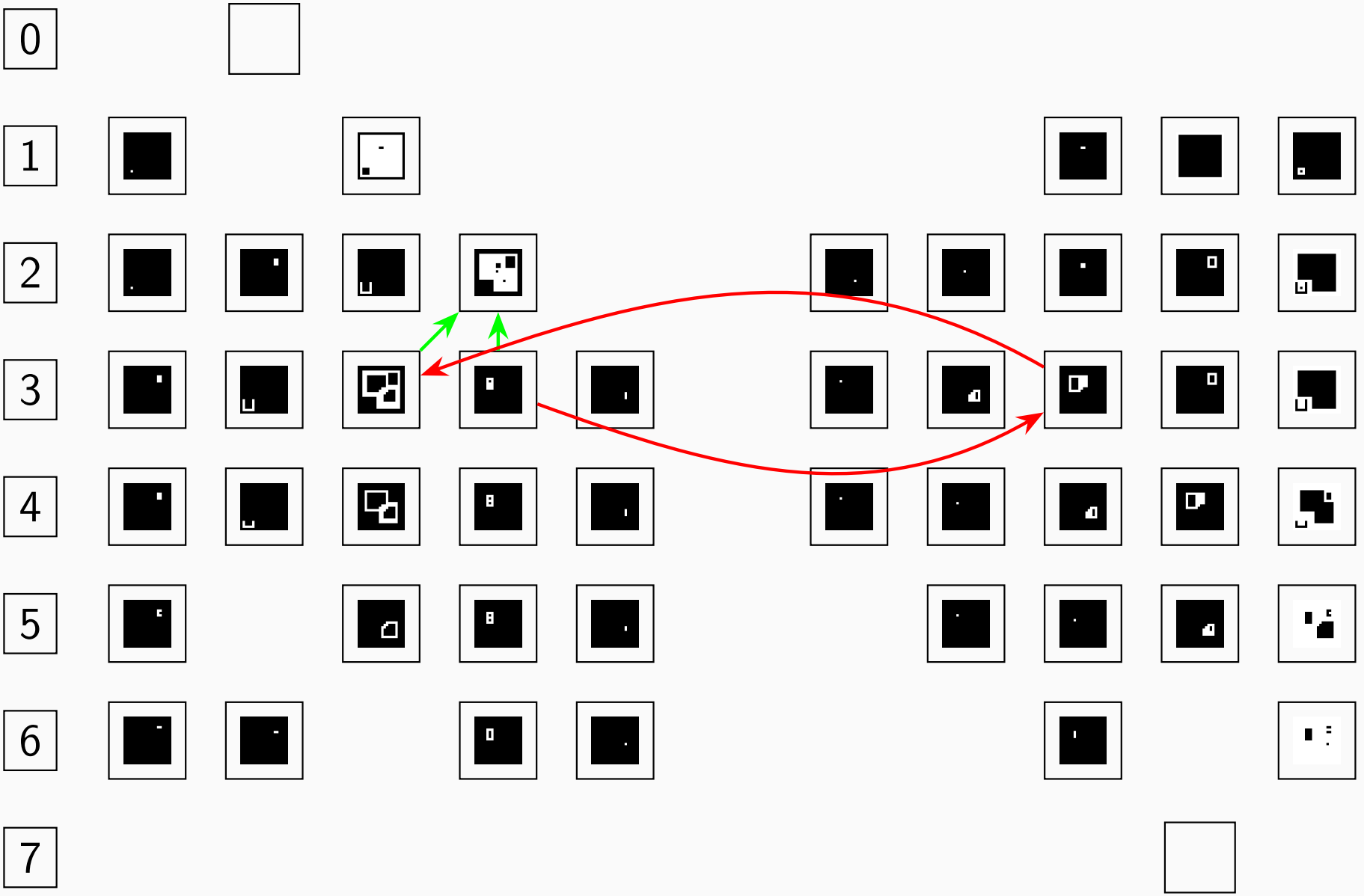
$$\psi(P) \longleftarrow P \longrightarrow \varphi(P)$$

If both edges exist, exactly one is redundant by transitivity

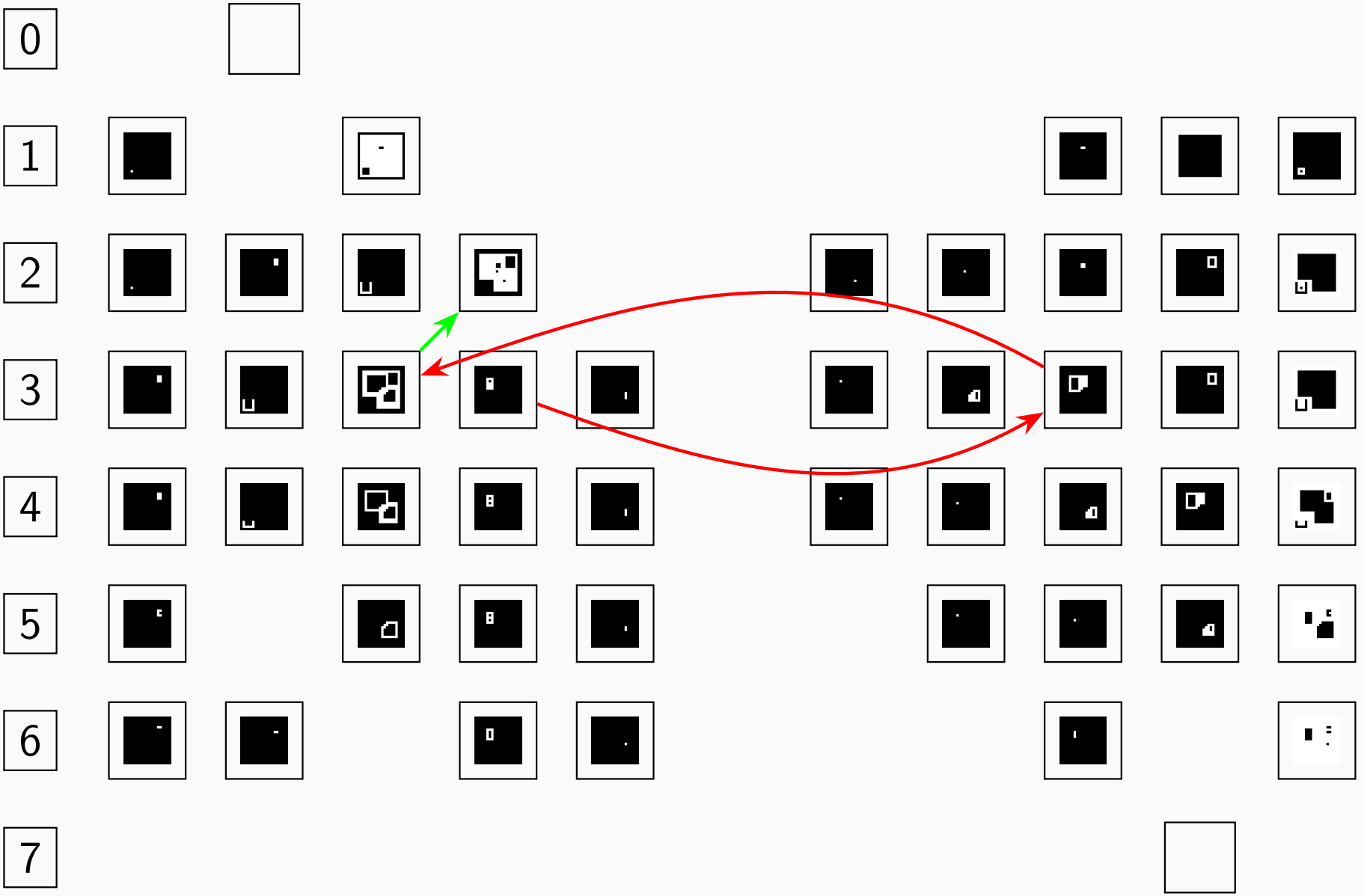
Three transitive patterns

-
- if $\varphi(P) = [\varphi \circ \psi \circ \psi](P)$ then $P \longrightarrow \varphi(P)$ is removed
-

Transitive pattern 2



Transitive pattern 2



Transitive pattern 3

Redundant links

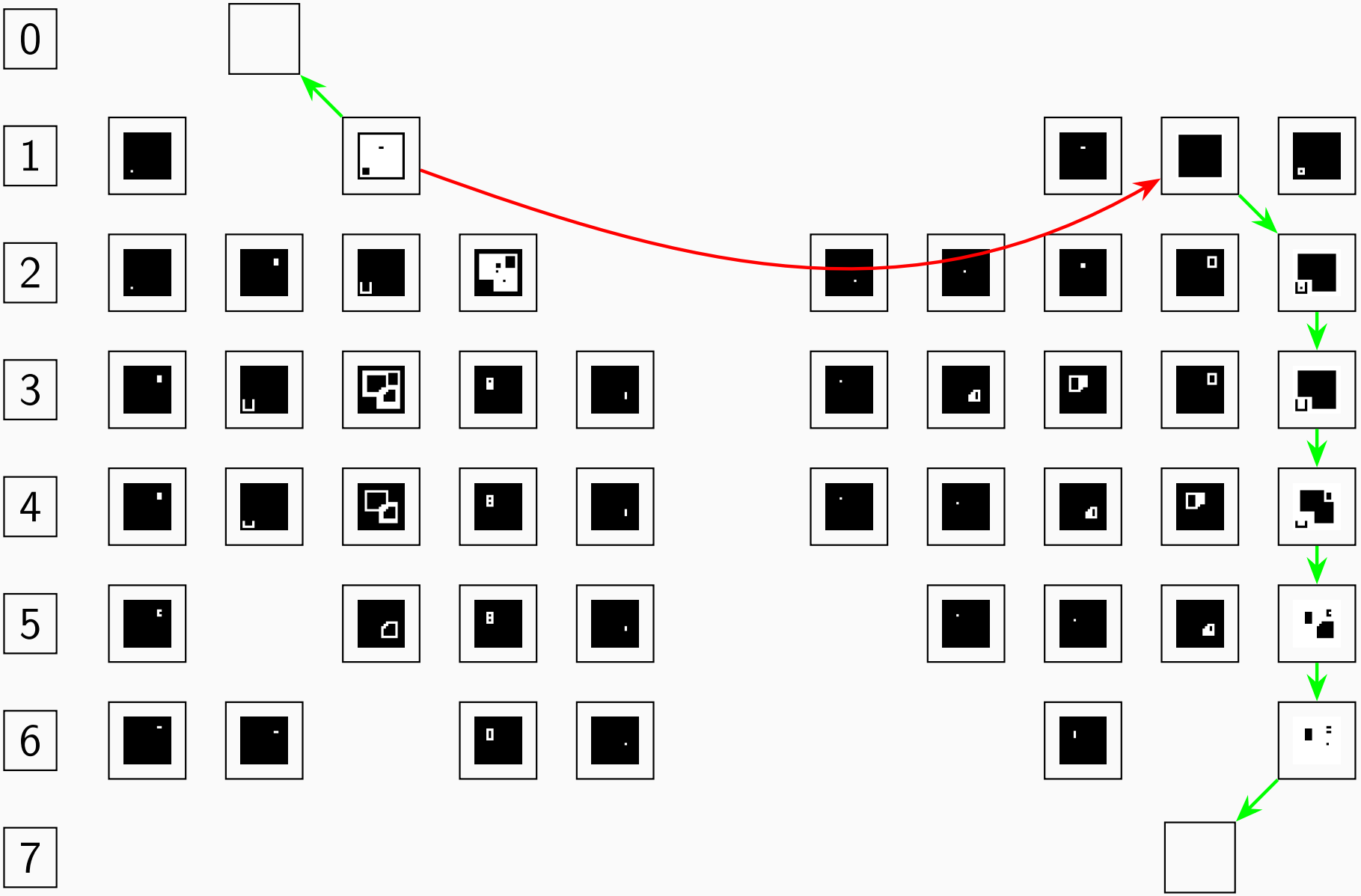
$$\psi(P) \longleftarrow P \longrightarrow \varphi(P)$$

If both edges exist, exactly one is redundant by transitivity

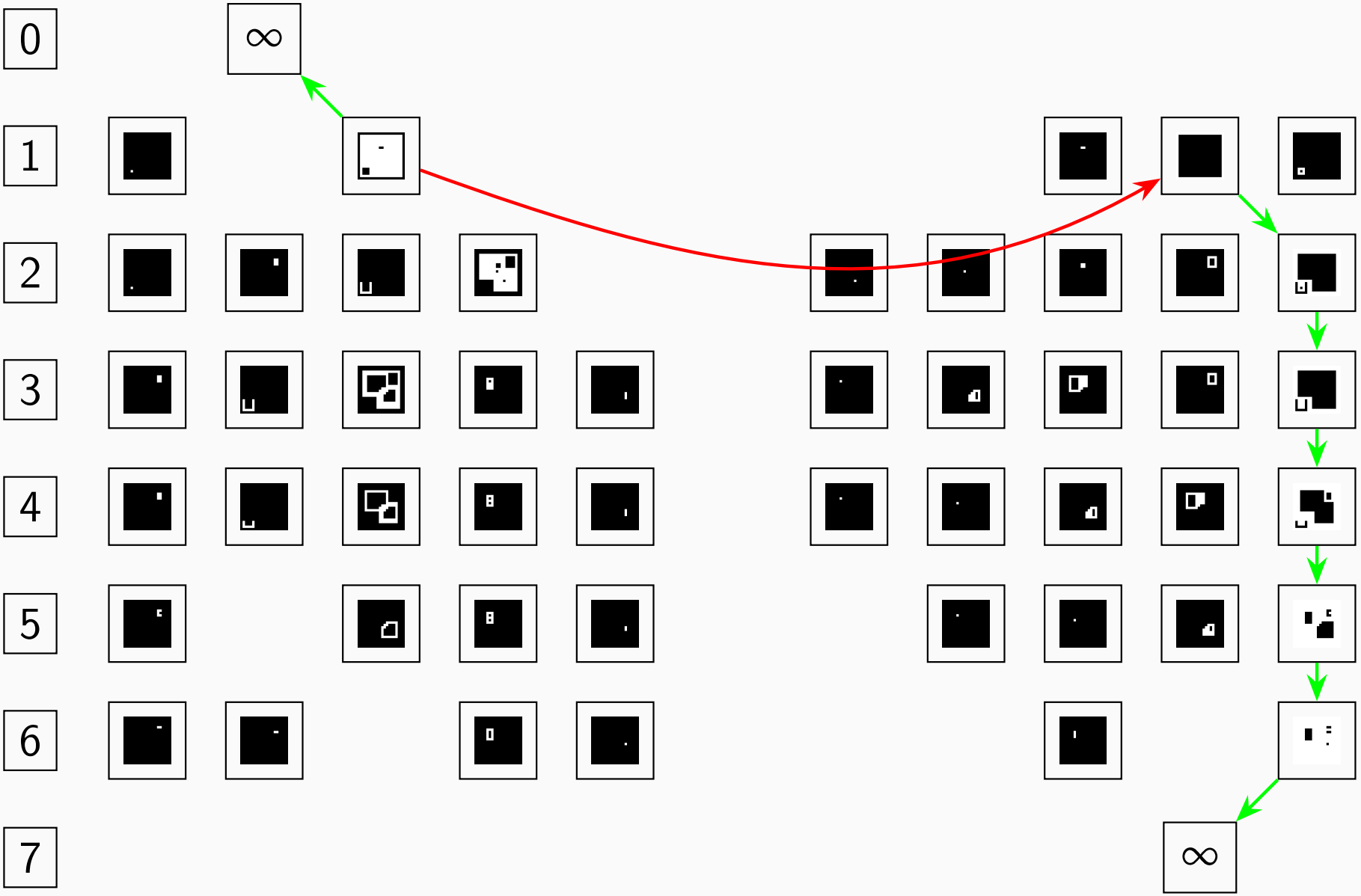
Three transitive patterns

1.
2.
3. if $\varphi(P) = [\varphi^{|\mathbb{V}|-2} \circ \psi](P)$ then $P \longrightarrow \varphi(P)$ is removed

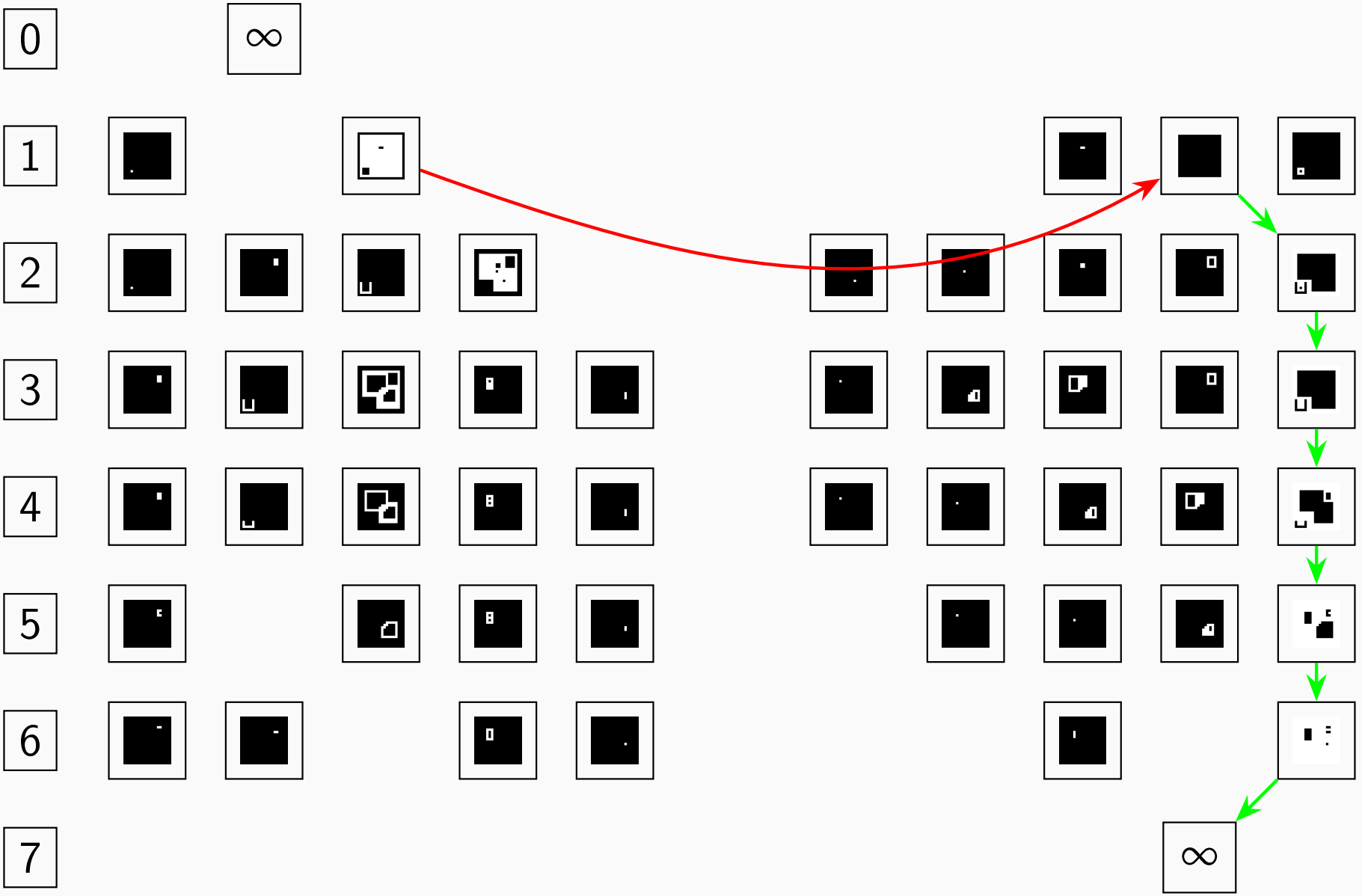
Transitive pattern 3



Transitive pattern 3



Transitive pattern 3



Transitive reduction

Transitive reduction

Removal from the graph of valued shapes $(\Xi, \lhd_{\equiv})$ of all the redundant (transitive) edges related to the transitive patterns:

1. $\psi(P) = [\varphi \circ \psi \circ \varphi](P)$
2. $\varphi(P) = [\varphi \circ \psi \circ \psi](P)$
3. $\varphi(P) = [\varphi^{|\mathbb{V}|-2} \circ \psi](P)$

Transitive reduction

Transitive reduction

Removal from the graph of valued shapes $(\Xi, \lhd_{\equiv})$ of all the redundant (transitive) edges related to the transitive patterns:

1. $\psi(P) = [\varphi \circ \psi \circ \varphi](P)$
2. $\varphi(P) = [\varphi \circ \psi \circ \psi](P)$
3. $\varphi(P) = [\varphi^{|\mathbb{V}|-2} \circ \psi](P)$

Before transitive reduction

For each node $P \in \Xi$, we may have either:

- $\psi(P) \longleftarrow P \longrightarrow \varphi(P)$
- $P \longrightarrow \varphi(P)$ (infinite "background" nodes)
- P (node(s) ∞)

Transitive reduction

Transitive reduction

Removal from the graph of valued shapes $(\Xi, \lhd_{\equiv})$ of all the redundant (transitive) edges related to the transitive patterns:

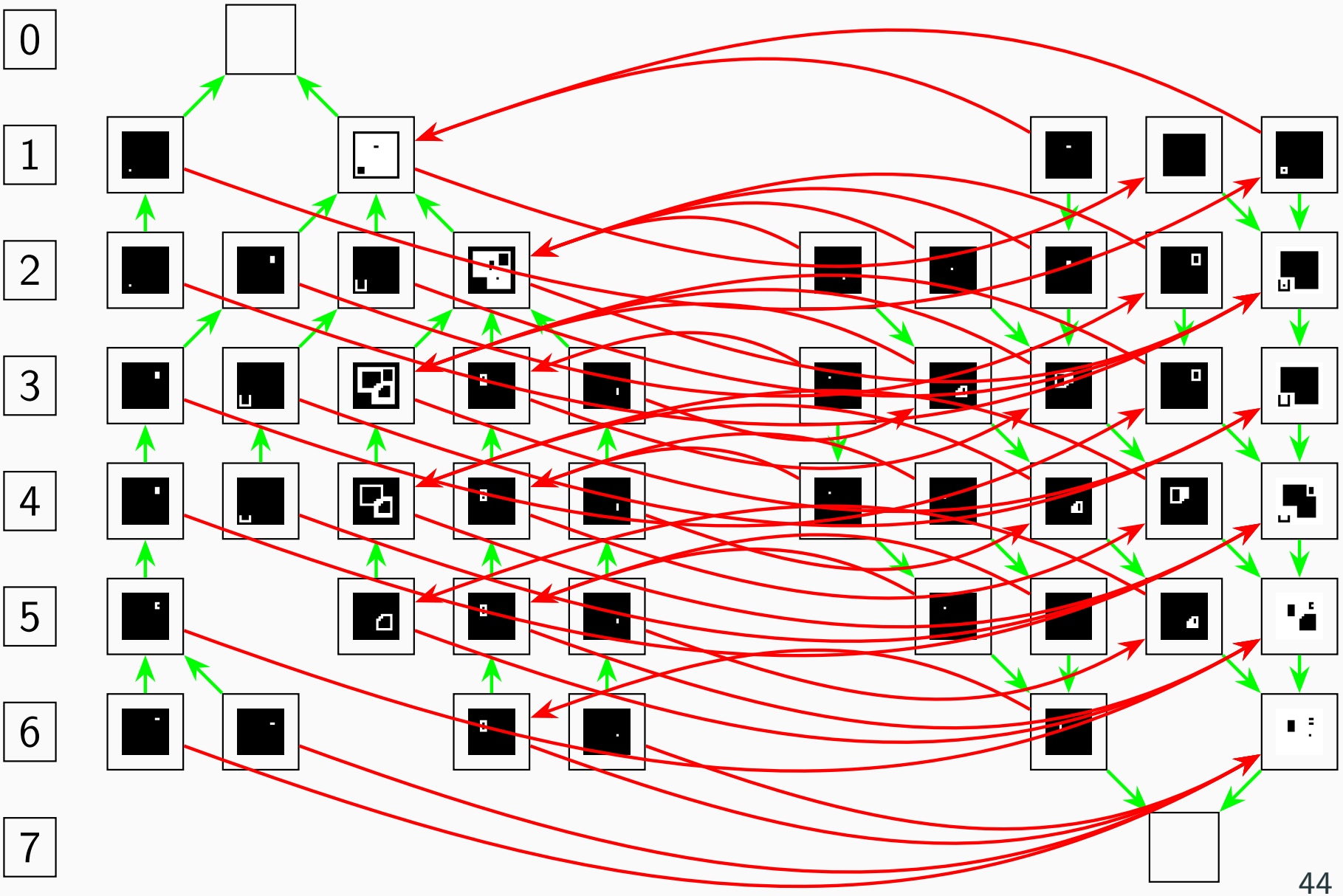
1. $\psi(P) = [\varphi \circ \psi \circ \varphi](P)$
2. $\varphi(P) = [\varphi \circ \psi \circ \psi](P)$
3. $\varphi(P) = [\varphi^{|\mathbb{V}|-2} \circ \psi](P)$

After transitive reduction

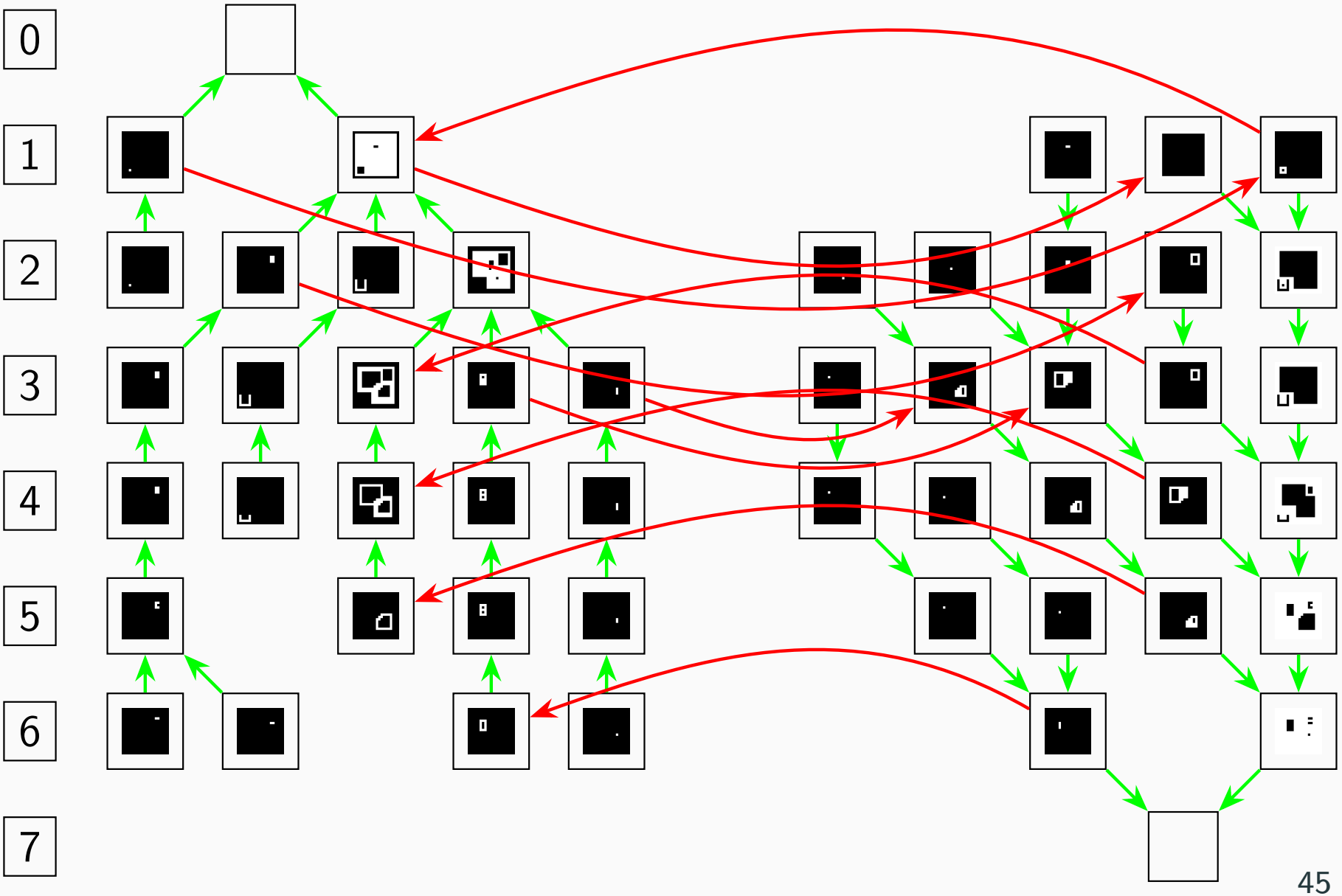
For each node $P \in \Xi$, we have either:

- $\psi(P) \longleftarrow P$
- $P \longrightarrow \varphi(P)$
- P (node(s) ∞)

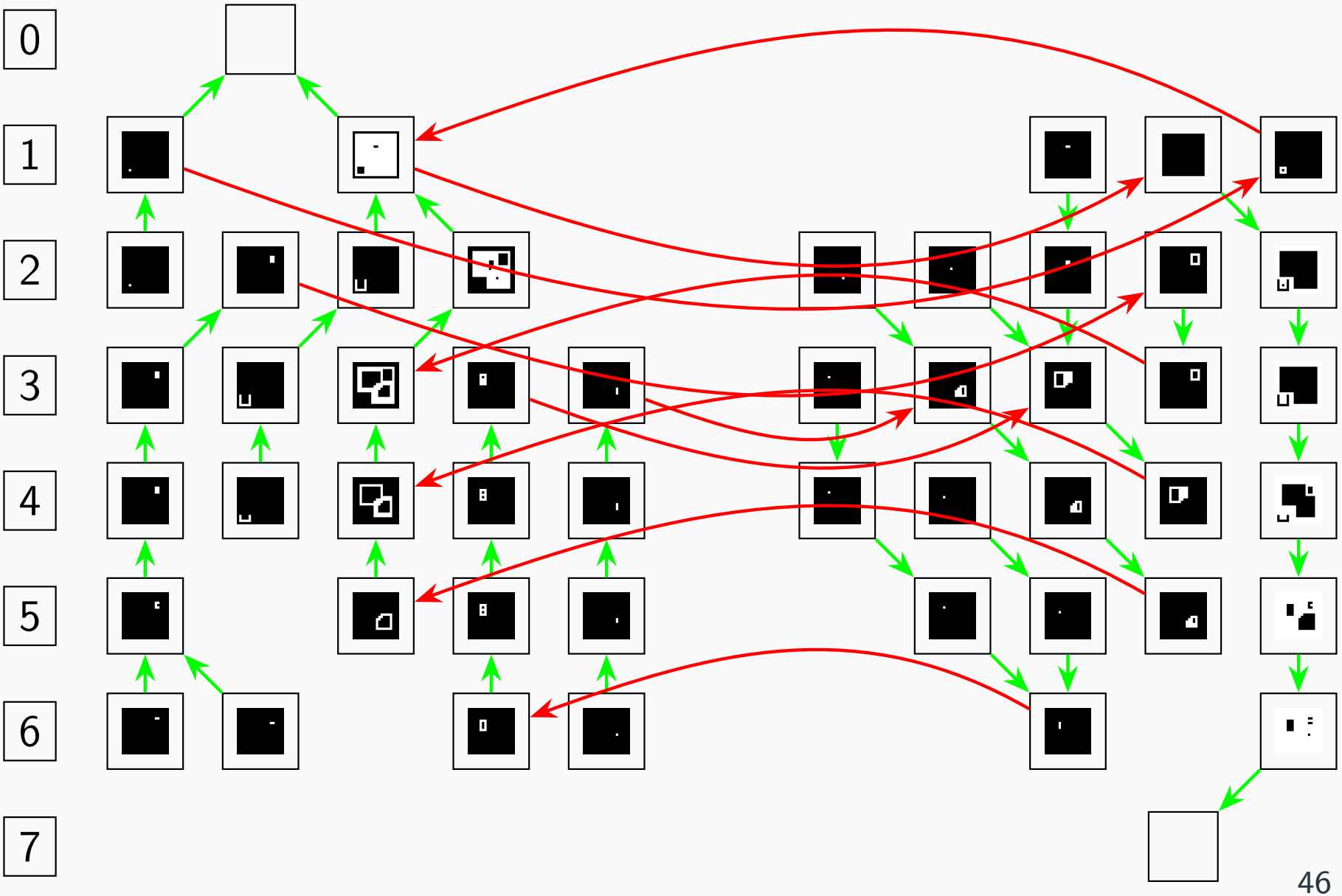
Transitive reduction: before



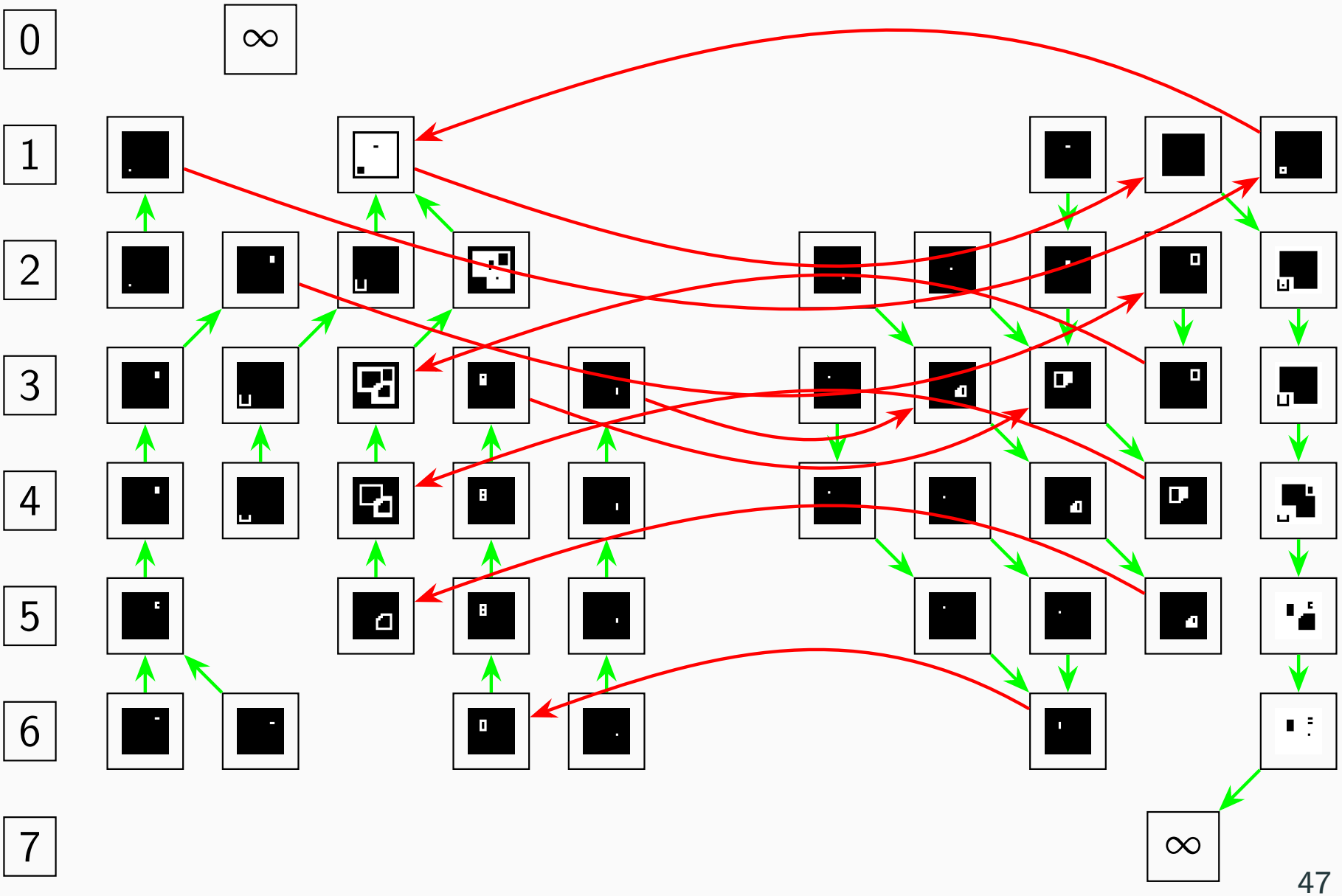
Transitive reduction: transitive pattern 1



Transitive reduction: transitive patterns 1 & 2



Transitive reduction: transitive patterns 1 & 2 & 3



Tree of valued shapes

Before transitive reduction: graph of valued shapes

Before transitive reduction:

- Graph of valued shapes
- $(\Xi, \lhd_{\Xi})$
- The graph of valued shapes is a DAG

Tree of valued shapes

Before transitive reduction: graph of valued shapes

Before transitive reduction:

- Graph of valued shapes
- $(\Xi, \lhd_\Xi)$
- The graph of valued shapes is a DAG

After transitive reduction: tree of valued shapes

After transitive reduction:

- Tree of valued shapes
- $(\Xi, \blacktriangleleft_\Xi)$ with $\blacktriangleleft_\Xi$ a subset of $\lhd_\Xi$
- **The tree of valued shapes is a tree (rooted in ∞)**

Step 3 – Compression: New trees of shapes

Spatial compression: Complete tree of shape

Nodes with similar supports

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ

If $P \neq Q$, 2 possible cases:

Spatial compression: Complete tree of shape

Nodes with similar supports

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ

If $P \neq Q$, 2 possible cases:

- $X \neq Y$

Spatial compression: Complete tree of shape

Nodes with similar supports

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ

If $P \neq Q$, 2 possible cases:

- $X \neq Y$
- $X = Y$ but $v \neq w$

Spatial compression: Complete tree of shape

Nodes with similar supports

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ

If $P \neq Q$, 2 possible cases:

- $X \neq Y$
- $X = Y$ but $v \neq w$

Equivalence classes of nodes

- Equivalence relation: $P \sim_\Theta Q \Longleftrightarrow X = Y$

Spatial compression: Complete tree of shape

Nodes with similar supports

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ

If $P \neq Q$, 2 possible cases:

- $X \neq Y$
- $X = Y$ but $v \neq w$

Equivalence classes of nodes

- Equivalence relation: $P \sim_\Theta Q \Longleftrightarrow X = Y$
- Bijection $\tilde{\pi}_\Theta$ between $\Xi/\sim_\Theta$ and Θ

Spatial compression: Complete tree of shape

Nodes with similar supports

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ

If $P \neq Q$, 2 possible cases:

- $X \neq Y$
- $X = Y$ but $v \neq w$

Equivalence classes of nodes

- Equivalence relation: $P \sim_\Theta Q \iff X = Y$
- Bijection $\tilde{\pi}_\Theta$ between $\Xi/\sim_\Theta$ and Θ
- $\tilde{\pi}_\Theta$ induces a homeomorphism between the tree of valued shapes $(\Xi, \blacktriangleleft_\Xi)$ and another tree $(\Theta, \blacktriangleleft_\Theta)$

Spatial compression: Complete tree of shape

Nodes with similar supports

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ

If $P \neq Q$, 2 possible cases:

- $X \neq Y$
- $X = Y$ but $v \neq w$

Equivalence classes of nodes

- Equivalence relation: $P \sim_\Theta Q \iff X = Y$
- Bijection $\tilde{\pi}_\Theta$ between $\Xi/\sim_\Theta$ and Θ
- $\tilde{\pi}_\Theta$ induces a homeomorphism between the tree of valued shapes $(\Xi, \blacktriangleleft_\Xi)$ and another tree $(\Theta, \blacktriangleleft_\Theta)$
- This new tree $(\Theta, \blacktriangleleft_\Theta)$ is called the complete tree of shapes.

Spatial compression: Complete tree of shape

Nodes with similar supports

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ

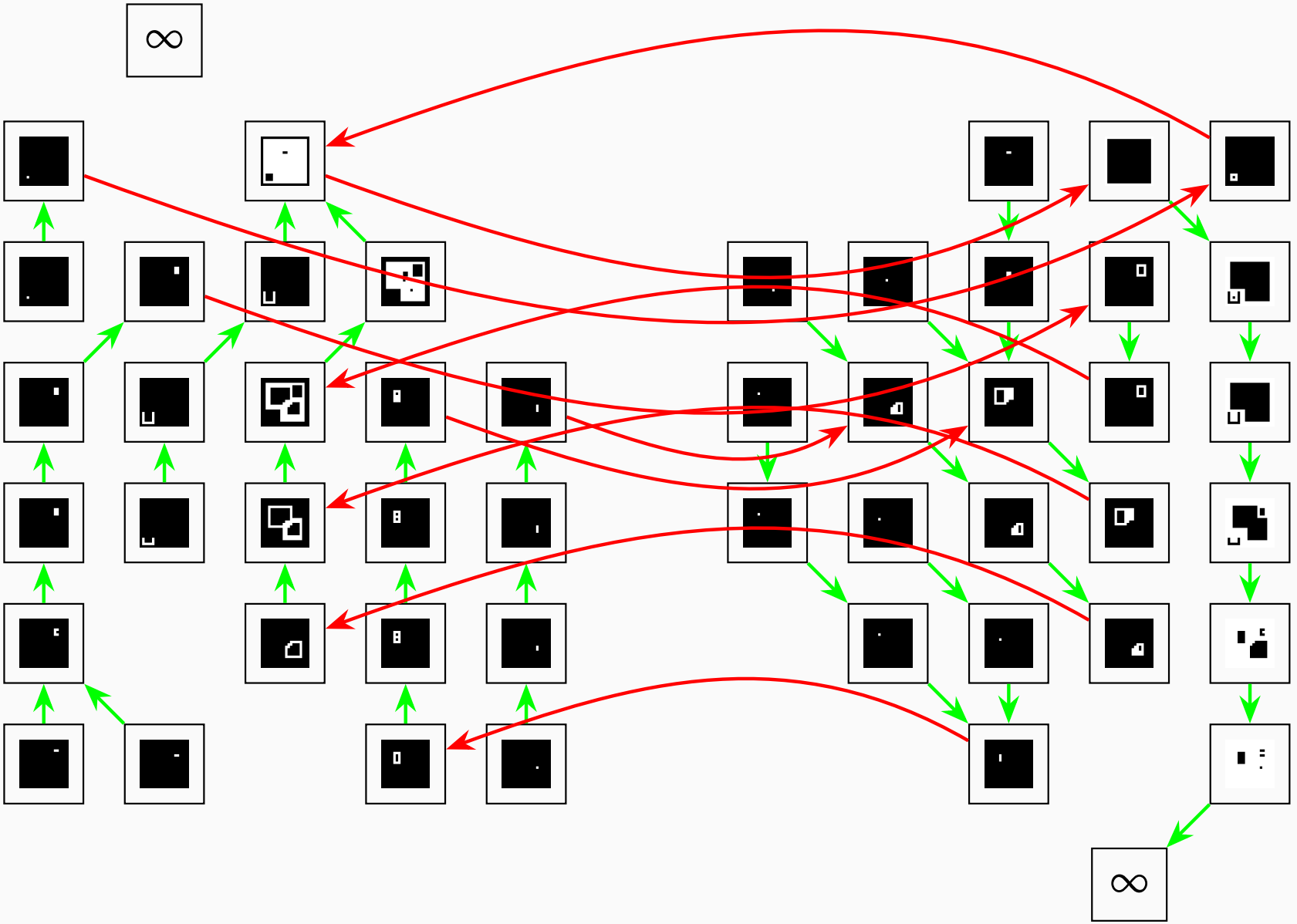
If $P \neq Q$, 2 possible cases:

- $X \neq Y$
- $X = Y$ but $v \neq w$

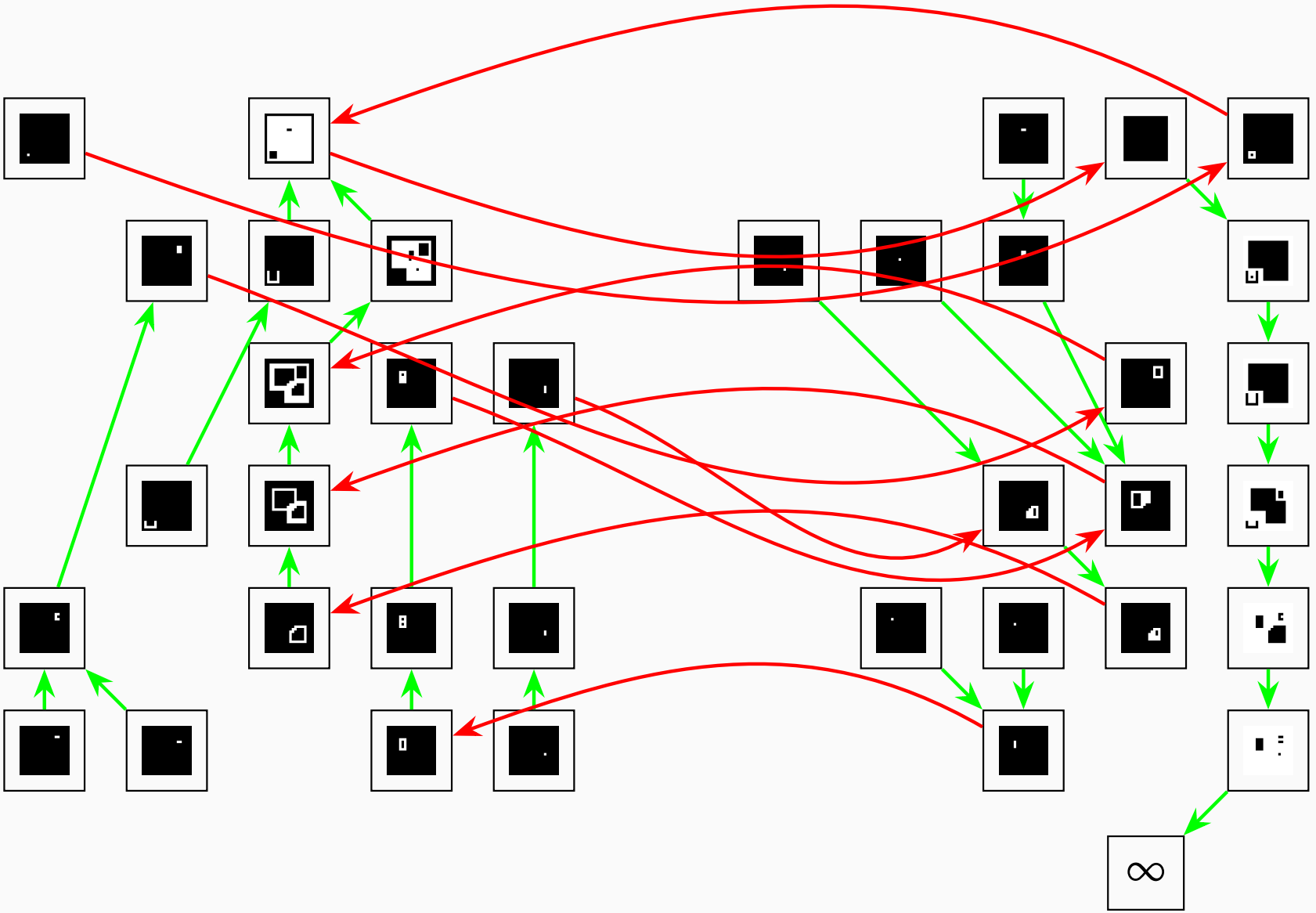
Equivalence classes of nodes

- Equivalence relation: $P \sim_\Theta Q \iff X = Y$
- Bijection $\tilde{\pi}_\Theta$ between $\Xi/\sim_\Theta$ and Θ
- $\tilde{\pi}_\Theta$ induces a homeomorphism between the tree of valued shapes $(\Xi, \blacktriangleleft_\Xi)$ and another tree $(\Theta, \blacktriangleleft_\Theta)$
- This new tree $(\Theta, \blacktriangleleft_\Theta)$ is called the complete tree of shapes.
- $(\Theta, \blacktriangleleft_\Theta)$ is a lossless compression of $(\Xi, \blacktriangleleft_\Xi)$

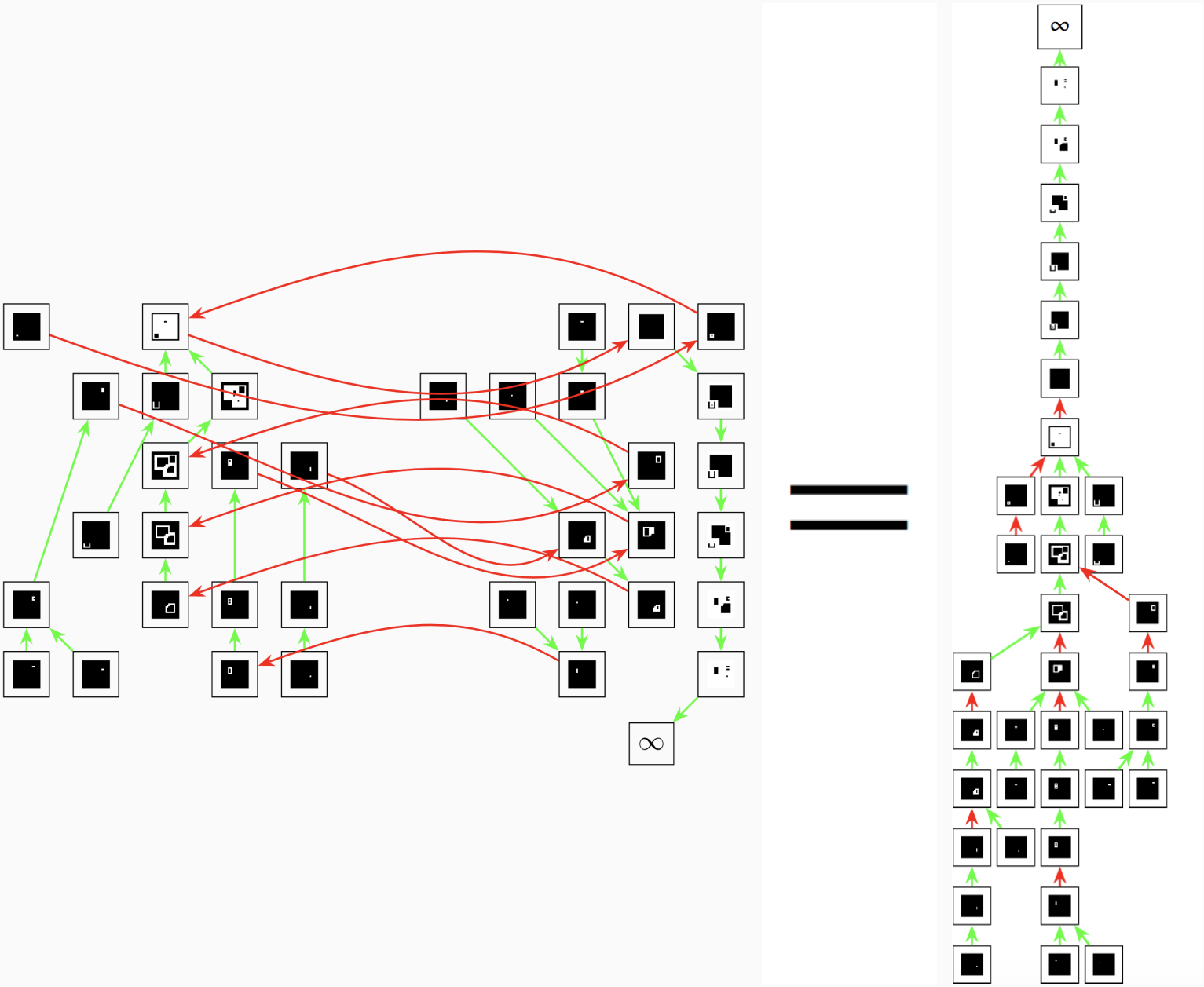
From the tree of valued shapes to the complete tree of shapes



From the tree of valued shapes to the complete tree of shapes



Complete tree of shapes



Topological compression: strong deletability

Strong deletability (Ronse 1986)

$X, D \subseteq \mathbb{U}$. $Y = X \setminus D \subseteq \mathbb{U}$.

D is strongly deletable (from X) if $\subseteq$ induces:

- a bijection between the CCs of Y and the CCs of X
- a bijection between the CCs of $\mathbb{U} \setminus X$ and the CCs if $\mathbb{U} \setminus Y$

Topological compression: strong deletability

Strong deletability (Ronse 1986)

$X, D \subseteq \mathbb{U}$. $Y = X \setminus D \subseteq \mathbb{U}$.

D is strongly deletable (from X) if $\subseteq$ induces:

- a bijection between the CCs of Y and the CCs of X
- a bijection between the CCs of $\mathbb{U} \setminus X$ and the CCs if $\mathbb{U} \setminus Y$

Strong deletability vs. homotopy

- $\mathbb{U} = \mathbb{Z}^2$: strong deletability $\Leftrightarrow$ decreasing homotopy
- $\mathbb{U} = \mathbb{Z}^3$: strong deletability $\Leftarrow$ decreasing homotopy

Topological compression: strong deletability

Strong deletability (Ronse 1986)

$X, D \subseteq \mathbb{U}$. $Y = X \setminus D \subseteq \mathbb{U}$.

D is strongly deletable (from X) if $\subseteq$ induces:

- a bijection between the CCs of Y and the CCs of X
- a bijection between the CCs of $\mathbb{U} \setminus X$ and the CCs if $\mathbb{U} \setminus Y$

Strong deletability vs. homotopy

- $\mathbb{U} = \mathbb{Z}^2$: strong deletability $\Leftrightarrow$ decreasing homotopy
- $\mathbb{U} = \mathbb{Z}^3$: strong deletability $\Leftarrow$ decreasing homotopy

Remarks on strong deletability

- ☺ Good topological invariant (pretty good in 2D)
- ☺ Easy to manipulate
- ☹ Does not deal with tunnels/handles in 3D

Topological compression: Topological equivalent relation

Successive nodes

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ such that $P \longrightarrow Q$, i.e.

- $P \xrightarrow{\text{red}} Q$ or
- $P \xrightarrow{\text{green}} Q$

Topological compression: Topological equivalent relation

Successive nodes

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ such that $P \longrightarrow Q$, i.e.

- $P \longrightarrow Q$ or
- $P \longrightarrow Q$

Relation based on strong deletability

$Q \searrow P$ if

- $P \longrightarrow Q$

Topological compression: Topological equivalent relation

Successive nodes

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ such that $P \longrightarrow Q$, i.e.

- $P \longrightarrow Q$ or
- $P \longrightarrow Q$

Relation based on strong deletability

$Q \searrow P$ if

- $P \longrightarrow Q$
- P is unique (coded in the graph of valued shapes)

Topological compression: Topological equivalent relation

Successive nodes

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ such that $P \longrightarrow Q$, i.e.

- $P \longrightarrow Q$ or
- $P \longrightarrow Q$

Relation based on strong deletability

$Q \searrow P$ if

- $P \longrightarrow Q$
- P is unique (coded in the graph of valued shapes)
- $Q \setminus P$ is strongly deletable (idem)

Topological compression: Topological equivalent relation

Successive nodes

$P = (X, v)$, $Q = (Y, w)$ two nodes of Ξ such that $P \longrightarrow Q$, i.e.

- $P \xrightarrow{\quad} Q$ or
- $P \longrightarrow Q$

Relation based on strong deletability

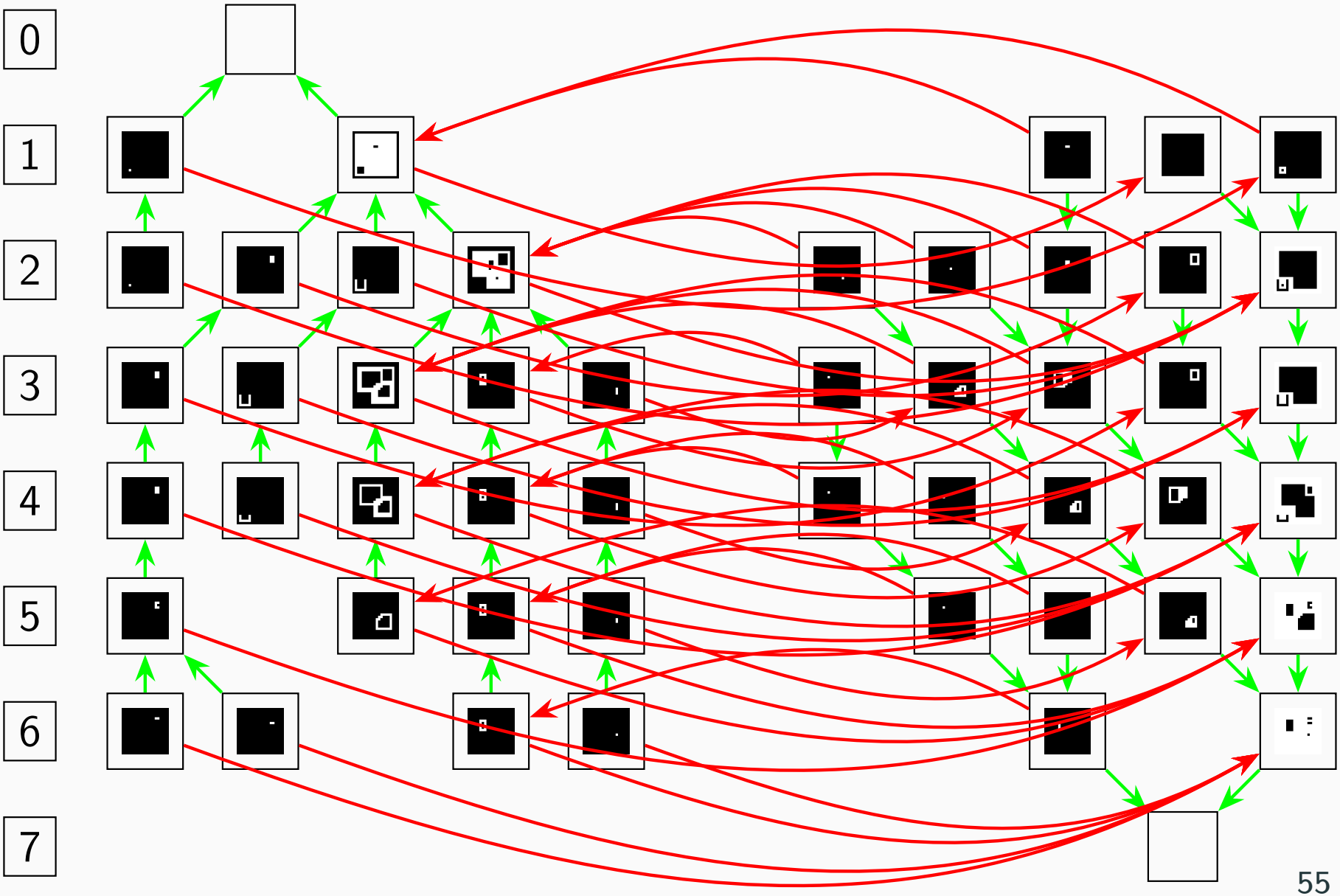
$Q \searrow P$ if

- $P \longrightarrow Q$
- P is unique (coded in the graph of valued shapes)
- $Q \setminus P$ is strongly deletable (idem)

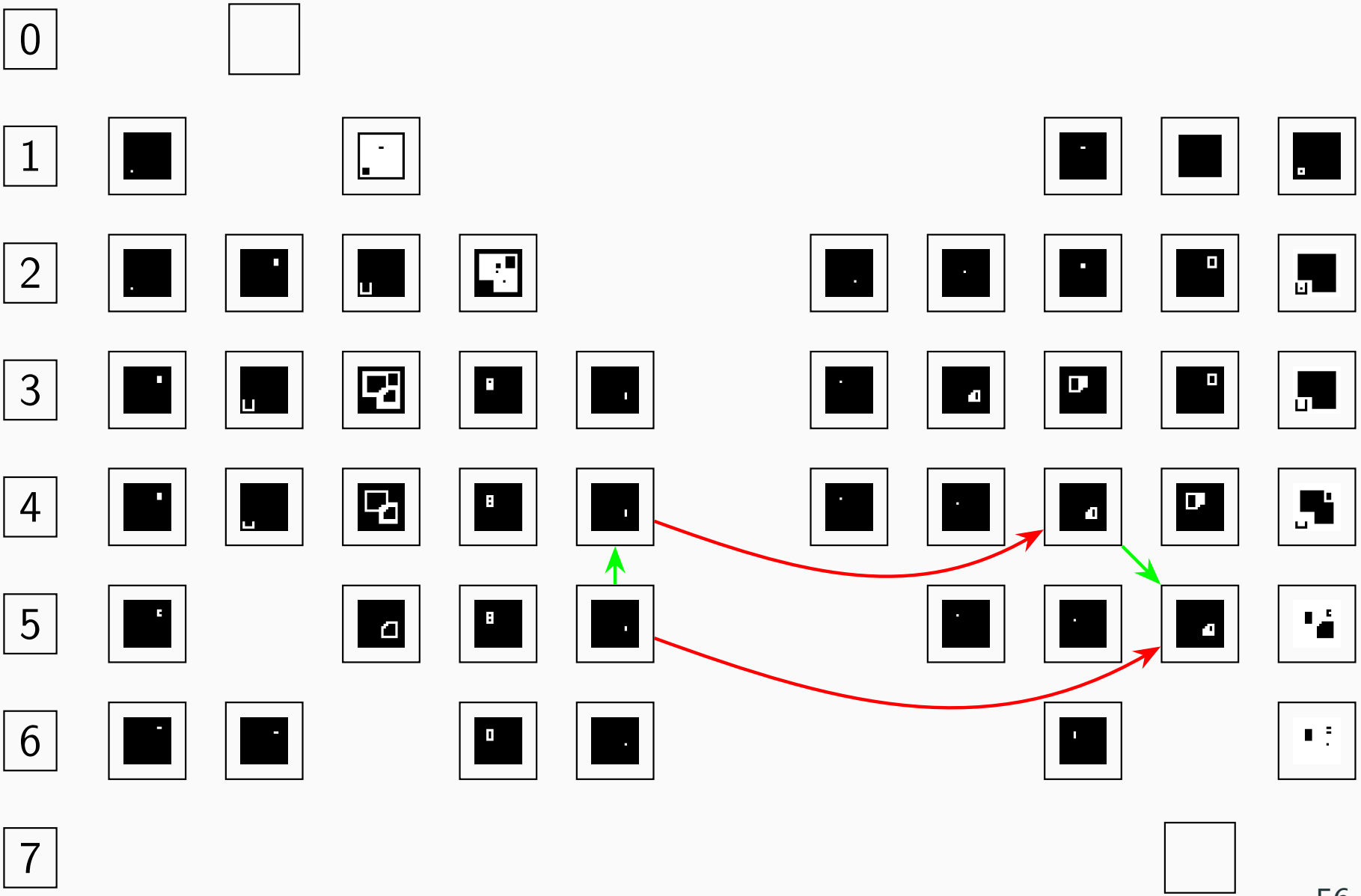
Equivalence relation

We define the topological equivalence relation $\sim_H$ on Ξ as the **reflexive-transitive-symmetric closure of** $\searrow$.

Topological compression: Topological tree of shape



Topological compression: Topological tree of shape



Topological compression: Topological tree of shape

Topological tree of shapes

- $H = \Xi/\sim_H$

Topological compression: Topological tree of shape

Topological tree of shapes

- $H = \Xi/{\sim_H}$
- $\blacktriangleleft_H$: the relation on H induced by $\blacktriangleleft_\Xi$ on Ξ wrt $\sim_H$

Topological compression: Topological tree of shape

Topological tree of shapes

- $H = \Xi/\sim_H$
- $\blacktriangleleft_H$: the relation on H induced by $\blacktriangleleft_\Xi$ on Ξ wrt $\sim_H$
- $(H, \blacktriangleleft_H)$ is a new tree called the topological tree of shapes

Topological compression: Topological tree of shape

Topological tree of shapes

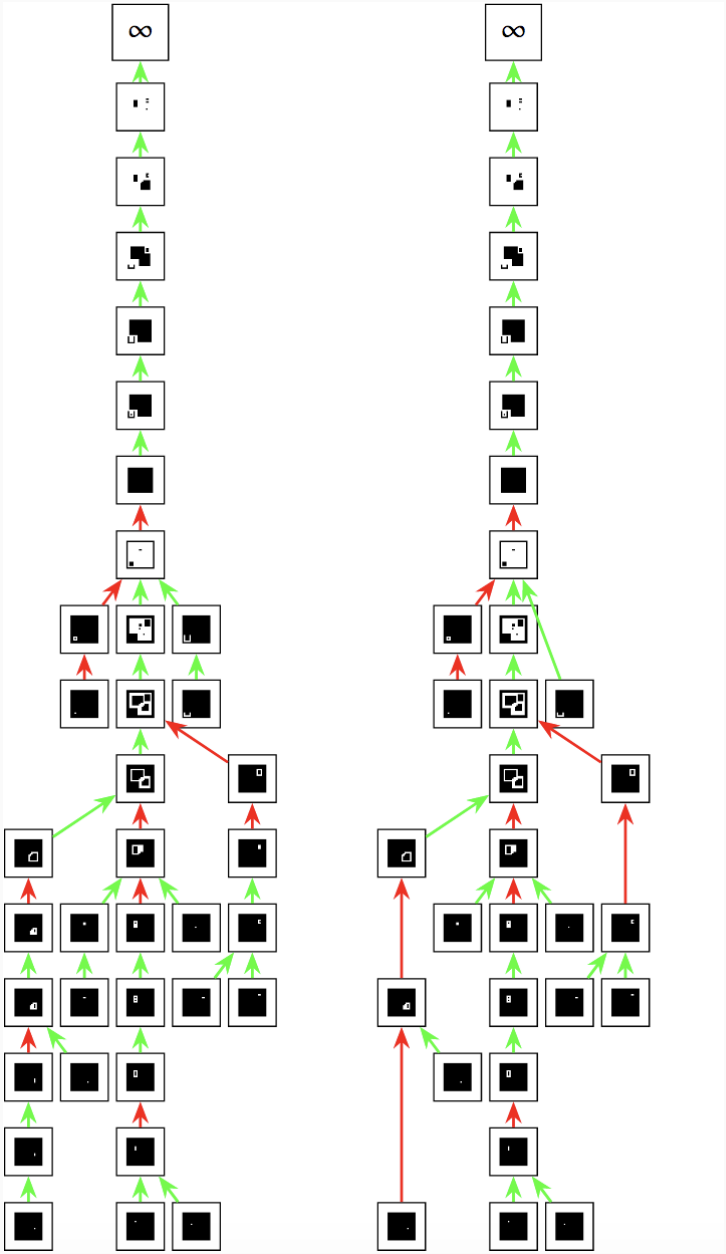
- $H = \Xi/\sim_H$
- $\blacktriangleleft_H$: the relation on H induced by $\blacktriangleleft_\Xi$ on Ξ wrt $\sim_H$
- $(H, \blacktriangleleft_H)$ is a new tree called the topological tree of shapes
- Decreasing homeomorphism from $(\Xi, \blacktriangleleft_\Xi)$ to $(H, \blacktriangleleft_H)$

Topological compression: Topological tree of shape

Topological tree of shapes

- $H = \Xi/\sim_H$
- $\blacktriangleleft_H$: the relation on H induced by $\blacktriangleleft_\Xi$ on Ξ wrt $\sim_H$
- $(H, \blacktriangleleft_H)$ is a new tree called the topological tree of shapes
- Decreasing homeomorphism from $(\Xi, \blacktriangleleft_\Xi)$ to $(H, \blacktriangleleft_H)$
- $(H, \blacktriangleleft_\Theta)$ is a lossy compression of $(\Xi, \blacktriangleleft_\Xi)$

Topological tree of shapes



Links between hierarchical structures

Graph of valued shapes vs. morphological trees

The graph of valued shapes $(\Xi, \lhd_\Xi)$ includes

- the valued max-tree (homeomorphic to the max-tree)

Graph of valued shapes vs. morphological trees

The graph of valued shapes $(\Xi, \lhd_\Xi)$ includes

- the valued max-tree (homeomorphic to the max-tree)
- the valued min-tree (homeomorphic to the min-tree)

Graph of valued shapes vs. morphological trees

The graph of valued shapes $(\Xi, \lhd_\Xi)$ includes

- the valued max-tree (homeomorphic to the max-tree)
- the valued min-tree (homeomorphic to the min-tree)
- all the adjacency trees at each threshold set

Graph of valued shapes vs. morphological trees

The graph of valued shapes $(\Xi, \lhd_\Xi)$ includes

- the valued max-tree (homeomorphic to the max-tree)
- the valued min-tree (homeomorphic to the min-tree)
- all the adjacency trees at each threshold set

The graph of valued shapes $(\Xi, \lhd_\Xi)$ includes

- the tree of valued shapes $(\Xi, \blacktriangleleft_\Xi)$
  and the transformation between both is reversible

Tree of valued shapes vs. morphological trees

The tree of valued shapes $(\Xi, \blacktriangleleft_\Xi)$ is homeomorphic to

- the complete topological tree of shapes

Tree of valued shapes vs. morphological trees

The tree of valued shapes $(\Xi, \blacktriangleleft_\Xi)$ is homeomorphic to

- the complete topological tree of shapes

which is homeomorphic to

- the topological tree of shapes

Tree of valued shapes vs. morphological trees

The tree of valued shapes $(\Xi, \blacktriangleleft_\Xi)$ is homeomorphic to

- the complete topological tree of shapes

which is homeomorphic to

- the topological tree of shapes
- the tree of shapes (Monasse et al. 2000)

Tree of valued shapes vs. morphological trees

The tree of valued shapes $(\Xi, \blacktriangleleft_\Xi)$ is homeomorphic to

- the complete topological tree of shapes

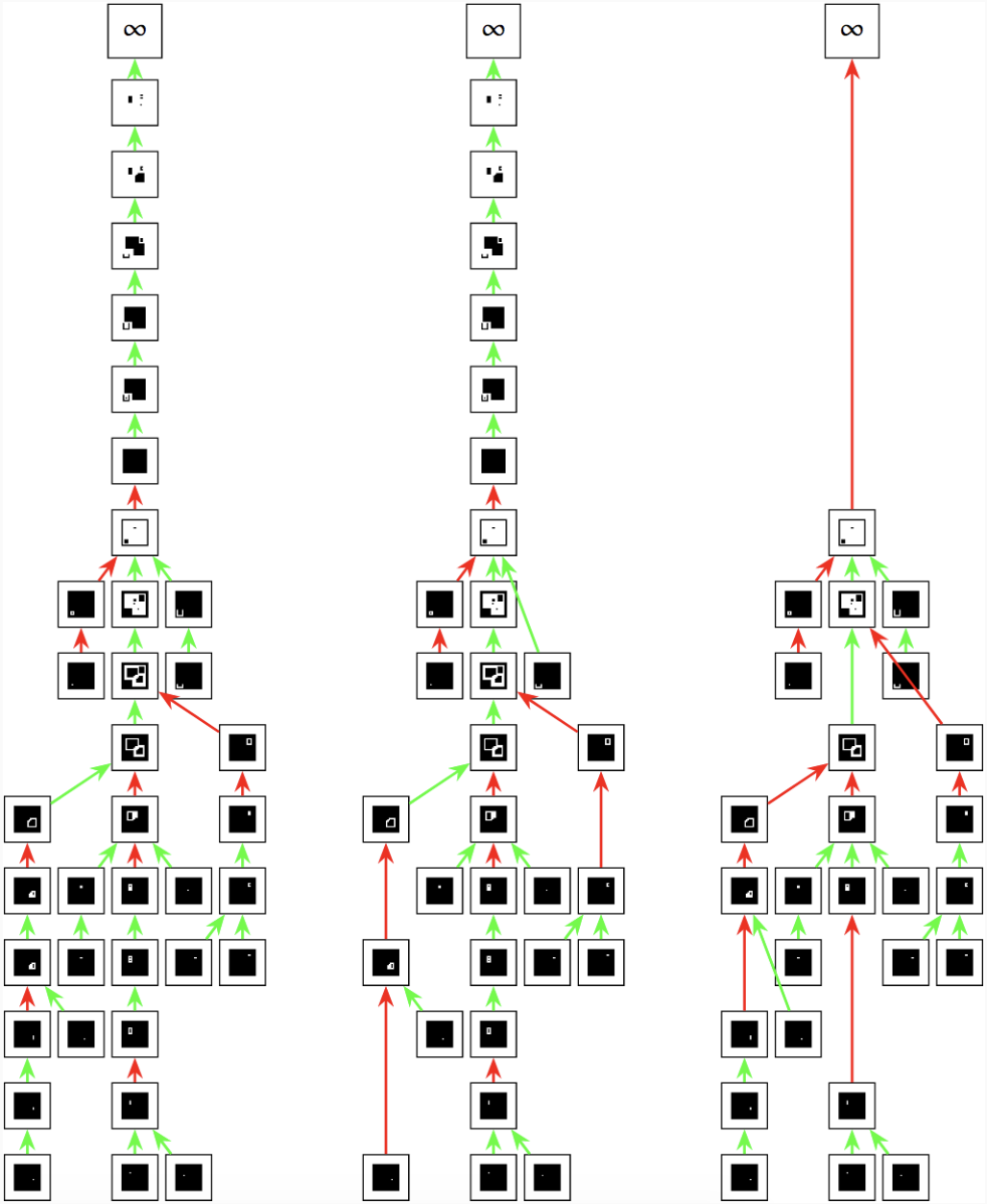
which is homeomorphic to

- the topological tree of shapes
- the tree of shapes (Monasse et al. 2000)

which are both homeomorphic to

- the monotonic tree (Song et al. 2002)

Complete, topological and “standard” trees of shapes



Conclusion

Contributions

New hierarchical structures

- (DAG) graph of valued shapes

Contributions

New hierarchical structures

- (DAG) graph of valued shapes
- (Tree) tree of valued shapes

Contributions

New hierarchical structures

- (DAG) graph of valued shapes
- (Tree) tree of valued shapes
- (Tree) Complete tree of shapes

Contributions

New hierarchical structures

- (DAG) graph of valued shapes
- (Tree) tree of valued shapes
- (Tree) Complete tree of shapes
- (Tree) Topological tree of shapes

Contributions

New hierarchical structures

- (DAG) graph of valued shapes
- (Tree) tree of valued shapes
- (Tree) Complete tree of shapes
- (Tree) Topological tree of shapes

Unification of morphological trees

A new vision about the links between:

- component-trees
- trees of shapes

Contributions

New hierarchical structures

- (DAG) graph of valued shapes
- (Tree) tree of valued shapes
- (Tree) Complete tree of shapes
- (Tree) Topological tree of shapes

Unification of morphological trees

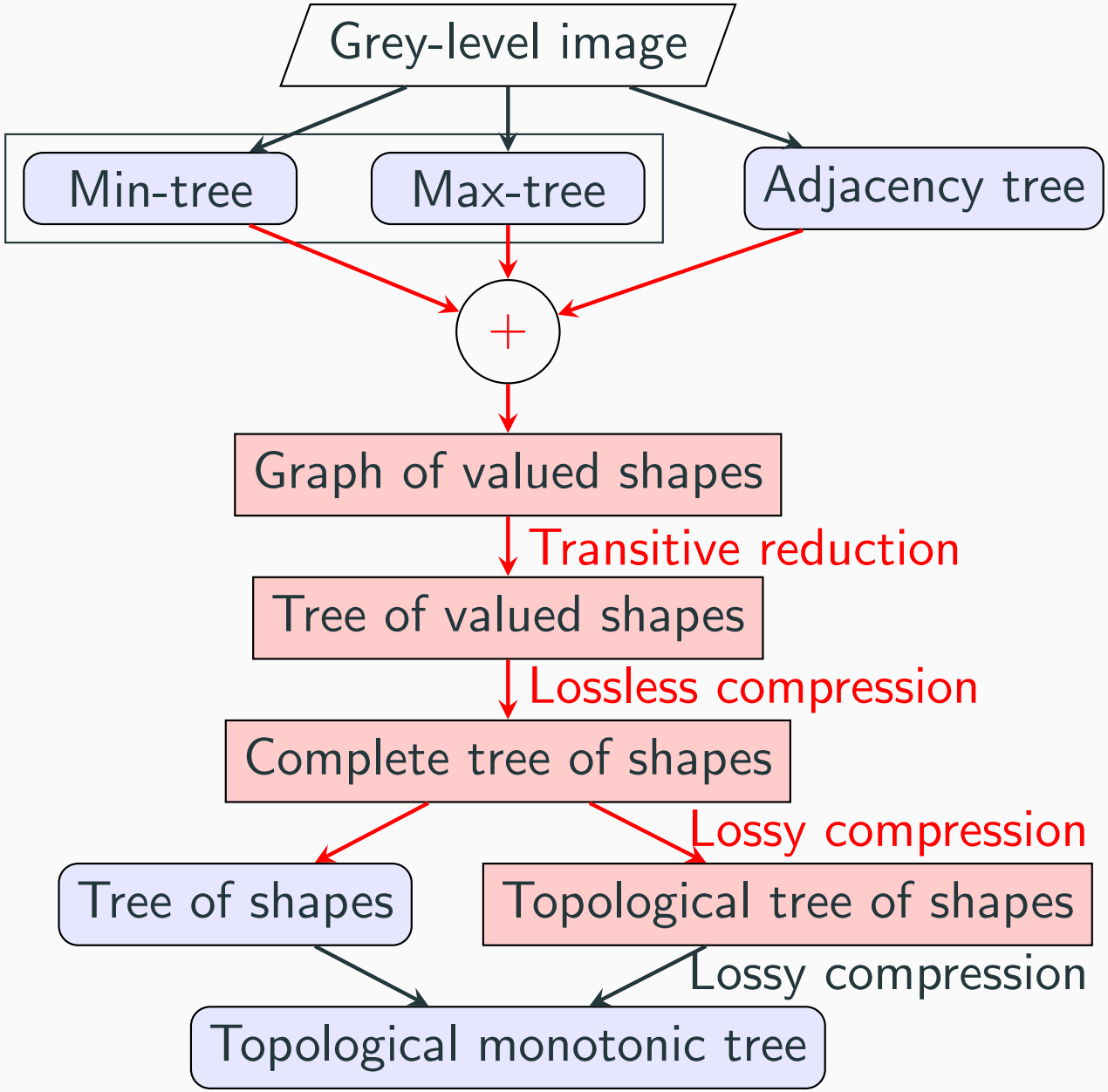
A new vision about the links between:

- component-trees
- trees of shapes

Embedding topology in morphological trees

- Adjacency-trees / component-trees
- Strong deletability / trees of shapes

Contributions



Perspectives

Theory

- Deeper exploration of the links between trees
- Topological handling in higher dimensions

Perspectives

Theory

- Deeper exploration of the links between trees
- Topological handling in higher dimensions

Algorithmics

- Construction of these new trees / DAGs
- Optimal bounds

Perspectives

Theory

- Deeper exploration of the links between trees
- Topological handling in higher dimensions

Algorithmics

- Construction of these new trees / DAGs
- Optimal bounds

Applications

- Grey-scale / fuzzy topology
- Topological data analysis
- Homotopic morphology operators
- Topological image compression

Thank you!