Weak Symmetry Breaking and Simplex Path Demonochromatizing

Colloquium

Jan-Philipp Litza

20.03.2015

1 Distributed Computing

2 Weak Symmetry Breaking

3 Simplex Path Demonochromatizing
- Algorithm by Kozlov
- Idea: Global Approach

Model

Processes $p_0, p_1, \ldots, p_n$ (e.g. computers, processors, humans)

- **communicate** to solve a common **task**
- have **process IDs/names** $0, \ldots, n \in \Pi$ (or {●, ●, ●}),
  input values $v_0, \ldots, v_n \in V^{in}$ and
  output values $o_0, \ldots, o_n \in V^{out}$.

Assumptions

Asynchronous	Every process acts as fast as it can or wants
Wait-Free	No process is allowed to wait for another one
Rank-Symmetric	Process IDs are only compared to each other, not used as absolute values

Model

Processes $p_0, p_1, \ldots, p_n$ (e.g. computers, processors, humans)

- **communicate** to solve a common **task**
- have **process IDs/names** $0, \ldots, n \in \Pi$ (or $\{\bullet, \bullet, \bullet\}$),
  input values $v_0, \ldots, v_n \in V^{in}$ and
  output values $o_0, \ldots, o_n \in V^{out}$.

Assumptions

Asynchronous	Every process acts as fast as it can or wants
Wait-Free	No process is allowed to wait for another one
$\Longrightarrow$	n processes can silently crash
Rank-Symmetric	Process IDs are only compared to each other, not used as absolute values

Model

Processes $p_0, p_1, \ldots, p_n$ (e.g. computers, processors, humans)

- **communicate** to solve a common **task**
- have **process IDs/names** $0, \ldots, n \in \Pi$ (or $\{\bullet, \bullet, \bullet\}$),
  input values $v_0, \ldots, v_n \in V^{in}$ and
  output values $o_0, \ldots, o_n \in V^{out}$.

Assumptions

Asynchronous	Every process acts as fast as it can or wants
Wait-Free	No process is allowed to wait for another one
$\Longrightarrow$	n processes can silently crash
Rank-Symmetric	Process IDs are only compared to each other, not used as absolute values

Input/Output Complexes

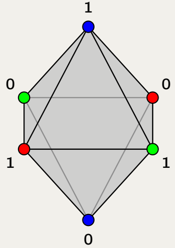
Configuration: Assignment of values/states to processes

- Not all input configurations might be valid
- Processes might crash even before starting
  $\Rightarrow$ every subset of a valid input configuration is valid again

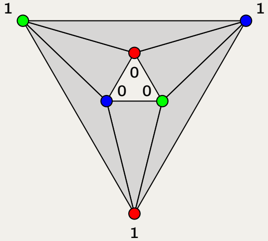
$\Rightarrow$ Model input configurations as pure simp. comp. $\mathcal{I} \subseteq 2^{\Pi \times V^{in}}$

Example

- 3 processes
- boolean input ($V^{in} = [1]$)



No restrictions



Not all the same value

Input/Output Complexes

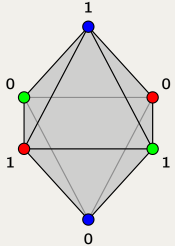
Configuration: Assignment of values/states to processes

- Not all input configurations might be valid
- Processes might crash even before starting
  $\Rightarrow$ every subset of a valid input configuration is valid again

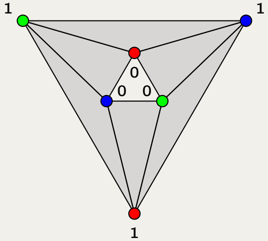
$\Rightarrow$ Model input configurations as pure simp. comp. $\mathcal{I} \subseteq 2^{\Pi \times V^{in}}$

Example

- 3 processes
- boolean input ($V^{in} = [1]$)



No restrictions



Not all the same value

Input/Output Complexes

Configuration: Assignment of values/states to processes

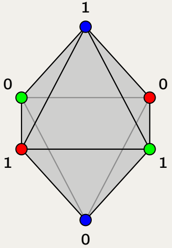
- Not all input configurations might be valid
- Processes might crash even before starting

  $\Rightarrow$ every subset of a valid input configuration is valid again

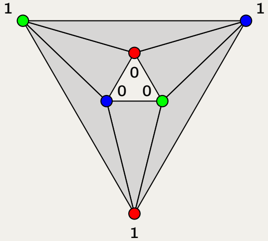
$\Rightarrow$ Model input configurations as pure simp. comp. $\mathcal{I} \subseteq 2^{\Pi \times V^{in}}$

Example

- 3 processes
- boolean input ($V^{in} = [1]$)



No restrictions



Not all the same value

Analogously:
Output complex $\mathcal{O} \subseteq 2^{\Pi \times V^{out}}$

Tasks

What output configurations may result from an input configuration?

$$\Delta \colon \mathcal{I} \to 2^{\mathcal{O}}$$

rigid $\Delta(\sigma)$ is pure of dimension $\dim \sigma$

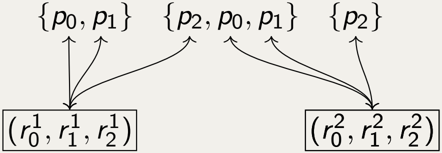
carrier map $\tau \subseteq \sigma \in \mathcal{I} \Rightarrow \Delta(\tau) \subseteq \Delta(\sigma) \subseteq \mathcal{O}$

name-preserving $\mathrm{pr}_{\Pi}(\sigma) = \bigcup_{\tau \in \Delta(\sigma)} \mathrm{pr}_{\Pi}(\tau)$

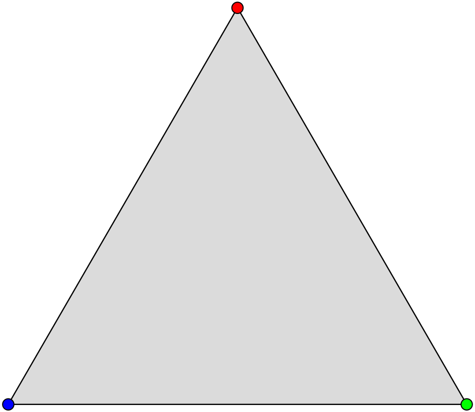
Protocol
Immediate Snapshots

- Communication happens in a predetermined number of **layers**/rounds
- Each layer has its own set of shared memory **storage registers**, one for each process
- A process executes a round by atomically writing to its own and reading all registers of its current round

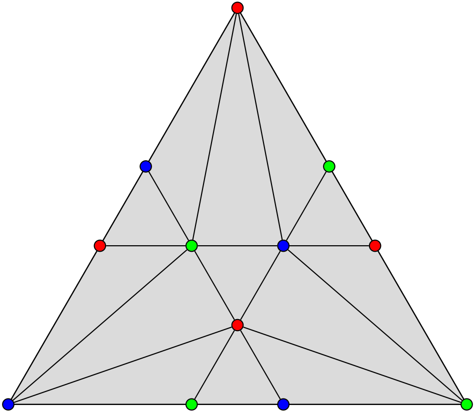
Example: 2 rounds, 3 processes



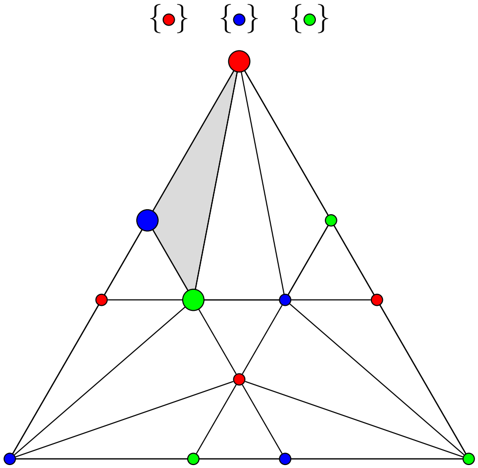
Standard Chromatic Subdivision χ



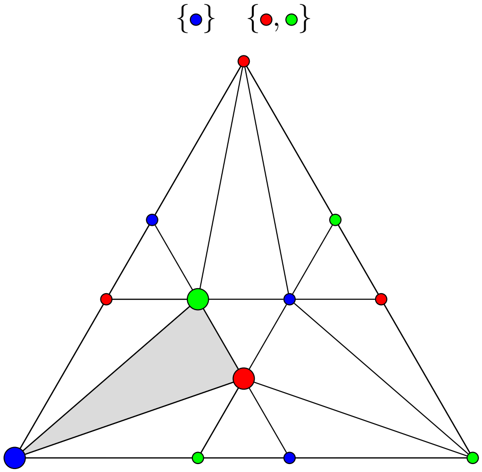
Standard Chromatic Subdivision χ



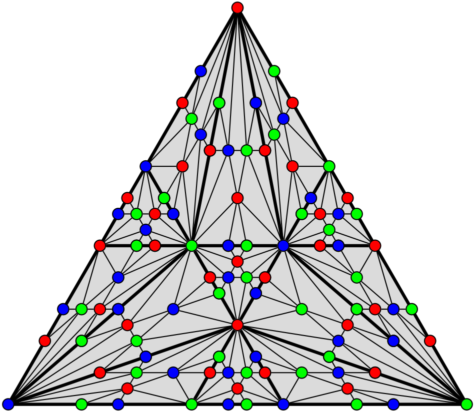
Standard Chromatic Subdivision χ



Standard Chromatic Subdivision χ



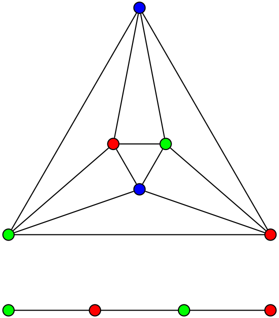
Standard Chromatic Subdivision χ



Basic Chromatic Subdivision

$n = 2$

$n = 1$

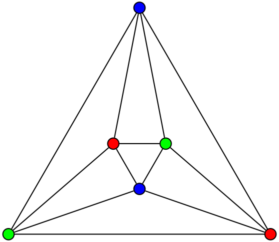


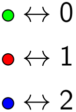
Basic chromatic subdivision

(Schlegel diagram of dual $(n+1)$-cube)

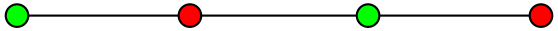
Basic Chromatic Subdivision

$n = 2$





$n = 1$

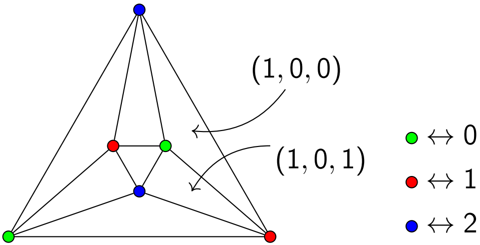


Basic chromatic subdivision

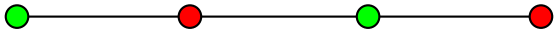
(Schlegel diagram of dual $(n+1)$-cube)

Basic Chromatic Subdivision

$n = 2$



$n = 1$



Basic chromatic subdivision

(Schlegel diagram of dual $(n+1)$-cube)

Computability

Theorem (Anonymous Computability, Herlihy and Shavit 1999)

A rank-symmetric decision task $(\mathcal{I}, \mathcal{O}, \Delta)$ *has a wait-free rank-symmetric protocol using immediate snapshots if and only if there exists an integer* K *and a color-preserving simplicial map*

$$\delta \colon \chi^K(\mathcal{I}) \to \mathcal{O}$$

such that $\delta \circ \chi^K$ *is carried by* Δ*.*

Theorem (Herlihy and Shavit 1999)

If $\mathcal{B}$ *is a chromatic subdivision of a complex* $\mathcal{A}$*, then there exists* $K \geqslant 0$ *and a color- and carrier-preserving simplicial map* $\chi^K(\mathcal{A}) \to \mathcal{B}$*.*

Computability

Theorem (Anonymous Computability, Herlihy and Shavit 1999)

A rank-symmetric decision task $(\mathcal{I}, \mathcal{O}, \Delta)$ has a wait-free rank-symmetric protocol using immediate snapshots if and only if there exists a subdivision $\Psi(\mathcal{I})$ and a color-preserving simplicial map

$$\delta\colon \Psi(\mathcal{I}) \to \mathcal{O}$$

such that $\delta \circ \Psi$ is carried by Δ.

Theorem (Herlihy and Shavit 1999)

If $\mathcal{B}$ is a chromatic subdivision of a complex $\mathcal{A}$, then there exists $K \geqslant 0$ and a color- and carrier-preserving simplicial map $\chi^K(\mathcal{A}) \to \mathcal{B}$.

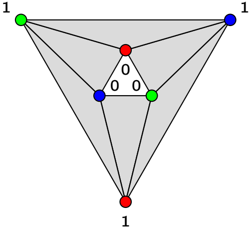
Weak Symmetry Breaking

Each of $n+1$ processes is assigned a unique process ID and has to decide on a boolean output value just by comparing its process ID with the others, such that if all processes participate, each value is output by at least one process.

$$\begin{aligned}
\Pi &= [n] \\
V^{in} &= \{\bot\} \\
\mathcal{I} &= 2^{\Pi \times V^{in}} = \sigma^{(n)} \\
V^{out} &= [1] \\
\mathcal{O} &= \{\tau \in 2^{\Pi \times V^{out}} : (|\operatorname{pr}_\Pi| \leqslant n) \vee (\operatorname{pr}_{V^{out}} = [1])\} \\
\Delta & \text{ „maximal"}
\end{aligned}$$

Weak Symmetry Breaking

Each of $n+1$ processes is assigned a unique process ID and has to decide on a boolean output value just by comparing its process ID with the others, such that if all processes participate, each value is output by at least one process.

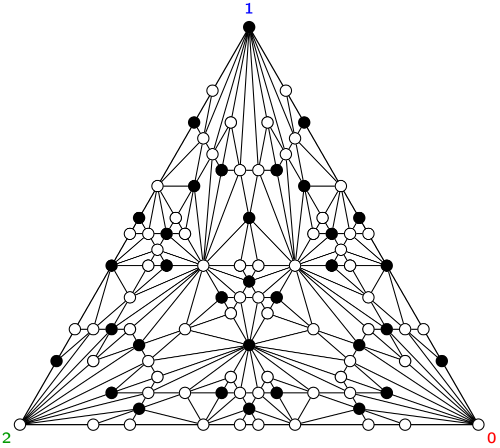


$$\Pi = [n]$$
$$V^{in} = \{\perp\}$$
$$\mathcal{I} = 2^{\Pi \times V^{in}} = \sigma^{(n)}$$
$$V^{out} = [1]$$
$$\mathcal{O} = \{\tau \in 2^{\Pi \times V^{out}} : (|\operatorname{pr}_{\Pi}| \leqslant n) \vee (\operatorname{pr}_{V^{out}} = [1])\}$$

Δ „maximal"

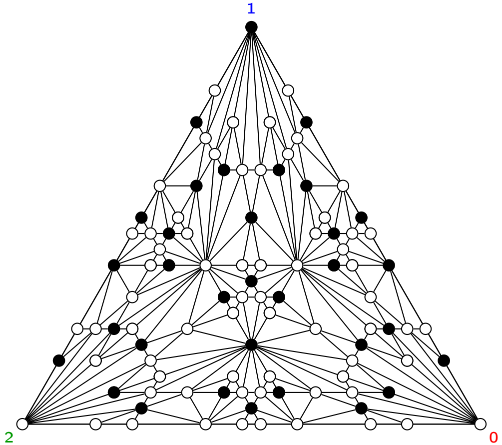
Binary Labeling

$$B\colon \Psi(\mathcal{I}) \to [1]$$



Binary Labeling

$$B\colon \Psi(\mathcal{I}) \to [1]$$



Non-monochromatic subdivision, but not rank-symmetric!

Roadmap

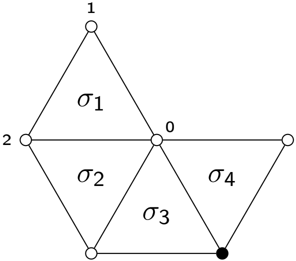
1. Generate equally many positively and negatively oriented 0-monochromatic n-simplices by subdividing rank-symmetrically
2. Pick two 0-monochromatic n-simplices σ and σ' of opposite orientation
3. Find a sequence $\sigma = \sigma_1, \dots, \sigma_\ell = \sigma'$ (**simplex path**) of n-simplices connecting them, where
   - $\sigma_{i,i+1} \coloneqq \sigma_i \cap \sigma_{i+1}$ is an $(n-1)$-face of both, and
   - only σ_1 and σ_ℓ are monochromatic
4. Demonochromatize this simplex path without changing its boundary

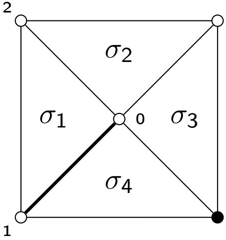
Roadmap

1. Generate equally many positively and negatively oriented 0-monochromatic n-simplices by subdividing rank-symmetrically
2. Pick two 0-monochromatic n-simplices σ and σ' of opposite orientation
3. Find a sequence $\sigma = \sigma_1, \ldots, \sigma_\ell = \sigma'$ (**simplex path**) of n-simplices connecting them, where
   - $\sigma_{i,i+1} \coloneqq \sigma_i \cap \sigma_{i+1}$ is an $(n-1)$-face of both, and
   - only σ_1 and σ_ℓ are monochromatic
4. Demonochromatize this simplex path without changing its boundary

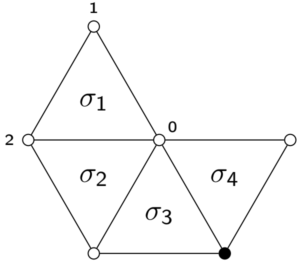
Problem: Not possible in parallel!

Simplex Path

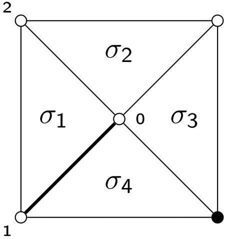




Simplex Path



$$I = (0,0,0)$$
$$C = (1,2,1)$$
$$V = (0,1,0)$$

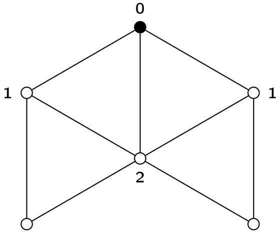


$C \in [n]^{\ell-1}$ Which vertices are "flipped"?

$V \in [1]^{\ell-1}$ What label does the flipped vertex get assigned?

$I \in [1]^{[n]}$ What labels does the first simplex get assigned?

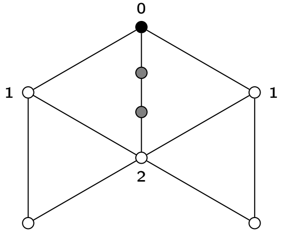
Edge Expansion



Example:

Edge Expansion

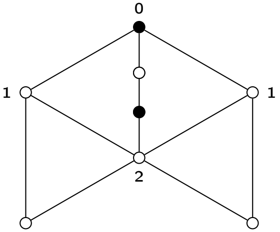
1. Subdivide $\sigma_{m,m+1}$ using basic chromatic subdivision



Example: $m = 2$,

Edge Expansion

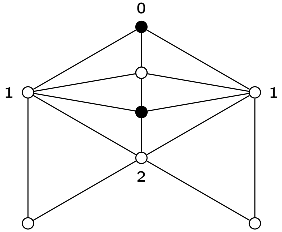
1. Subdivide $\sigma_{m,m+1}$ using basic chromatic subdivision
2. Assign boolean labels to new vertices according to
   $D = (d_0, \ldots, d_{C_m-1}, -, d_{C_m+1}, \ldots, d_n)$



Example: $m = 2$, $D = (1, -, 0)$,

Edge Expansion

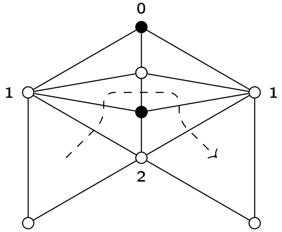
1. Subdivide $\sigma_{m,m+1}$ using basic chromatic subdivision
2. Assign boolean labels to new vertices according to $D = (d_0, \ldots, d_{C_m-1}, -, d_{C_m+1}, \ldots, d_n)$
3. Cone to $\sigma_m \triangle \sigma_{m+1}$



Example: $m = 2$, $D = (1, -, 0)$,

Edge Expansion

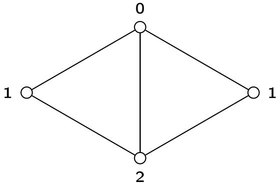
1. Subdivide $\sigma_{m,m+1}$ using basic chromatic subdivision
2. Assign boolean labels to new vertices according to $D = (d_0, \ldots, d_{C_m - 1}, -, d_{C_m + 1}, \ldots, d_n)$
3. Cone to $\sigma_m \triangle \sigma_{m+1}$
4. Re-route path according to n-cube-path $Q = (q_1, \ldots, q_t)$



Example: $m = 2,\ D = (1, -, 0),\ \ Q = (0, 2, 1, 2, 0)$

Edge Expansion

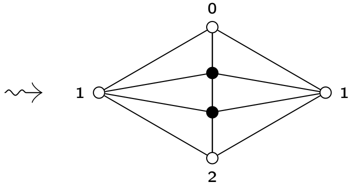
1. Subdivide $\sigma_{m,m+1}$ using basic chromatic subdivision
2. Assign boolean labels to new vertices according to $D = (d_0, \ldots, d_{C_m-1}, -, d_{C_m+1}, \ldots, d_n)$
3. Cone to $\sigma_m \triangle \sigma_{m+1}$
4. Re-route path according to n-cube-path $Q = (q_1, \ldots, q_t)$

$\rightsquigarrow$ 

Example: $m = 1$

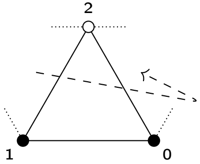
Edge Expansion

1. Subdivide $\sigma_{m,m+1}$ using basic chromatic subdivision
2. Assign boolean labels to new vertices according to $D = (d_0, \ldots, d_{C_m-1}, -, d_{C_m+1}, \ldots, d_n)$
3. Cone to $\sigma_m \triangle \sigma_{m+1}$
4. Re-route path according to n-cube-path $Q = (q_1, \ldots, q_t)$



Example: $m = 1$, $D = (1, \ldots, 1)$

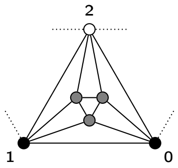
Vertex Expansion



Example:

Vertex Expansion

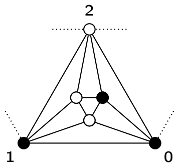
- Subdivide σ_m using basic chromatic subdivision



Example:

Vertex Expansion

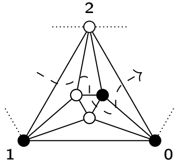
- Subdivide σ_m using basic chromatic subdivision
- Assign boolean labels to new vertices according to $D = (d_0, \ldots, d_n)$



Example: $D = (0, 1, 0)$,

Vertex Expansion

- Subdivide σ_m using basic chromatic subdivision
- Assign boolean labels to new vertices according to $D = (d_0, \ldots, d_n)$
- Re-route path according to n-cube-loop $Q = (q_1, \ldots, q_t)$



Example: $D = (0, 1, 0), \quad Q = (0, 1, 2, 0, 1, 2)$

Height Graph

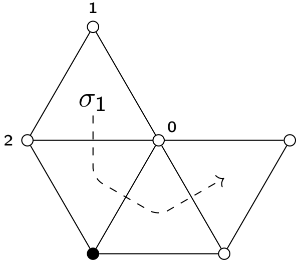
$h_i := \#(1, B(\sigma_i))$

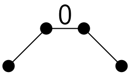
- Analogously: $h_{i,i+1} := \#(1, B(\sigma_{i,i+1}))$

Vertices (i, h_i) for $i = 1, \ldots, \ell$

Edges $\{(i, h_i), (i+, h_{i+1})\}$ for $i = 1, \ldots, \ell - 1$

Label edge $\{(i, h_i), (i+1, h_{i+1})\}$ with V_i if $h_i = h_{i+1}$

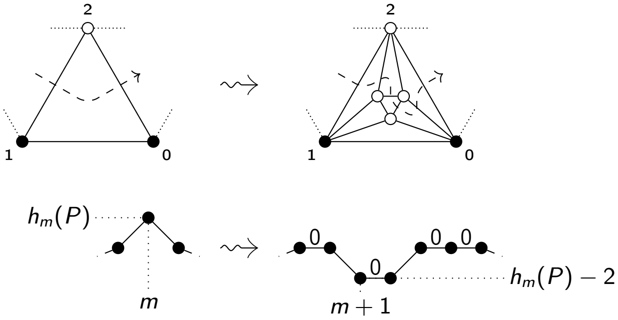




Summit Move

- Choose (odd) m such that $h_{m-1} < h_m > h_{m+1}$
- $B(\sigma_m) = (1, 1, 0, e_3, \ldots, e_n)$ (up to $S_{[n]}$-action)
- Vertex expansion of σ_m with $D := (0, 0, 0, \overline{e_3}, \ldots, \overline{e_n})$

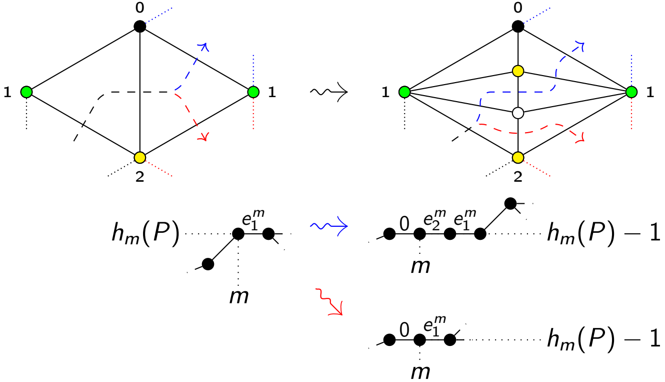
$$Q := (0, 1, 2, 0, 2, 1)$$



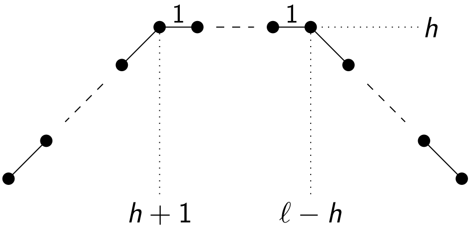
Plateau Move

- Choose m such that $h_{m-1} < h_m = h_{m+1}$
- $(C_{m-1}, C_m, C_{m+1}) = (0, 1, 2)$ or $(0, 1, 0)$ (up to $S_{[n]}$-action)
- Edge expansion of $\sigma_{m,m+1}$ with $D \coloneqq (0, -, e_2, \ldots, e_n)$

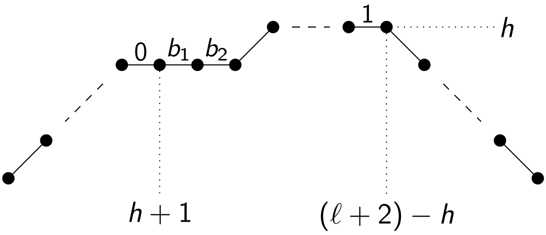
$Q \coloneqq (0, 2, 1, 0, 2)$ or $(0, 1, 0)$



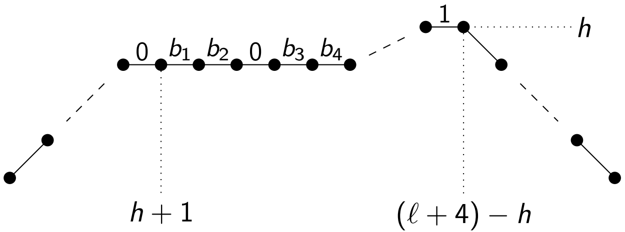
Repeat



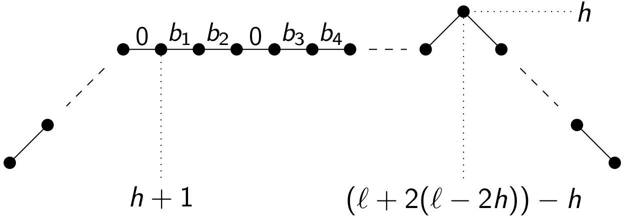
Repeat



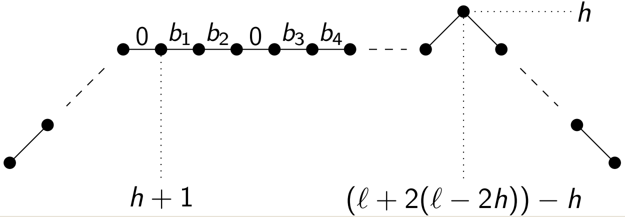
Repeat



Repeat



Repeat

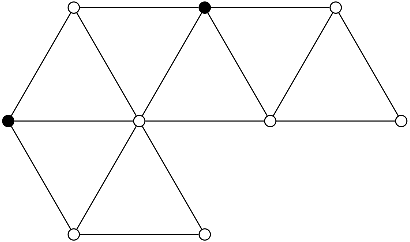


Complexity

Kozlov	$\mathcal{O}(\ell^{n-1})$ (without preprocessing) [L.]
Castañeda and Rajsbaum	$\mathcal{O}(n)$ (with preprocessing) [Attiya et al. 2013]

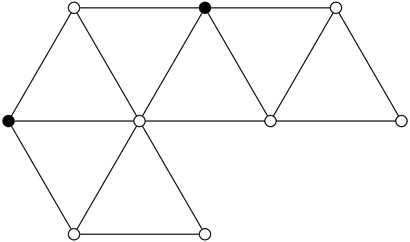
Motivation

- Brute-force: Why not simply apply standard chromatic subdivision everywhere?



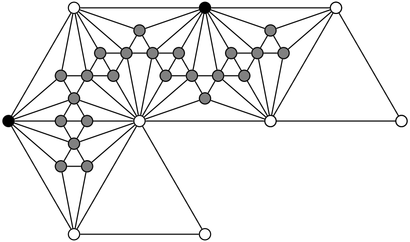
Motivation

- Brute-force: Why not simply apply standard chromatic subdivision everywhere?
  - ↯ Boundary must stay unmodified!



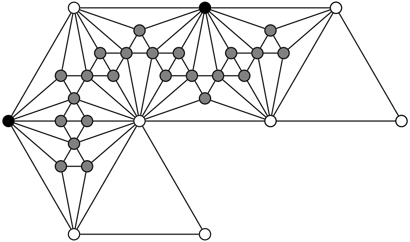
Motivation

- Brute-force: Why not simply apply standard chromatic subdivision everywhere?
  - ↯ Boundary must stay unmodified!



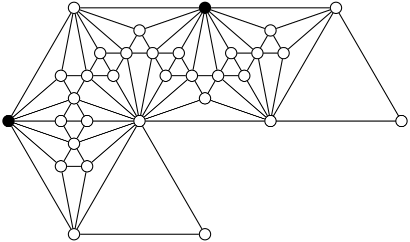
Motivation

- Brute-force: Why not simply apply standard chromatic subdivision everywhere?
  - ↯ Boundary must stay unmodified!
- Labeling?

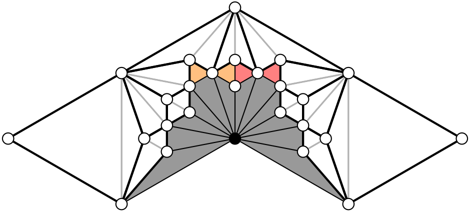


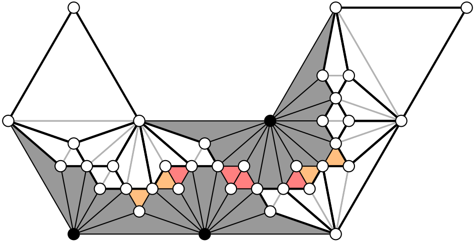
Motivation

- Brute-force: Why not simply apply standard chromatic subdivision everywhere?
  - ↯ Boundary must stay unmodified!
- Labeling? All 0 to maximize pairable simplices!

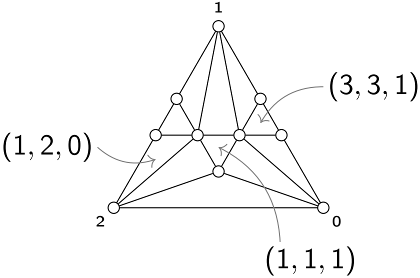


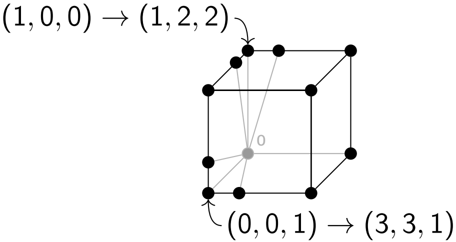
Problems





Possible Directions





- Adapt "exhaustive expansion" technique from Kozlov 2015
- Search for graph matchings

References

Herlihy, M. and N. Shavit (Nov. 1999). "The Topological Structure of Asynchronous Computability." In: **J. ACM** 46.6, pp. 858–923.

Attiya, H. et al. (2013). "Upper Bound on the Complexity of Solving Hard Renaming." In: **Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing**. PODC '13. Montréal, Québec, Canada: ACM, pp. 190–199.

Kozlov, D. N. (Feb. 2015). "Weak symmetry breaking and abstract simplex paths." In: **Mathematical Structures in Computer Science** FirstView, pp. 1–31.

Castañeda, A. and S. Rajsbaum (Mar. 2012). "New Combinatorial Topology Bounds for Renaming: The Upper Bound." In: **J. ACM** 59.1, 3:1–3:49.

Tasks

Example

$\Pi = [2]$, $V^{in} = V^{out} = [1]$, $\mathcal{I} = \Pi \times V^{in}$, $\mathcal{O} = \Pi \times V^{out}$

Task: Output the input value of any process

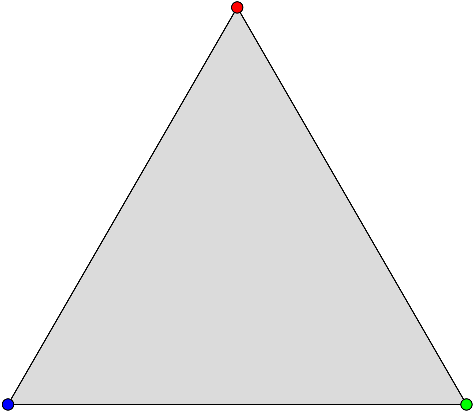
$$\Delta(\{0 \mapsto x, 1 \mapsto x, 2 \mapsto x\}) = 2^{\{0 \mapsto x, 1 \mapsto x, 2 \mapsto x\}}$$

$$\Delta(\{a \mapsto x, b \mapsto x\}) = 2^{\{a \mapsto x, b \mapsto x\}}$$

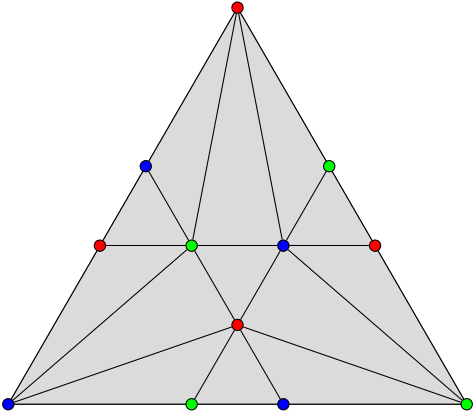
$$\Delta(\{a \mapsto x\}) = \{a \mapsto x\}$$

$$\text{for all other } \sigma \in \mathcal{I}: \quad \Delta(\sigma) = \{O \subseteq \mathcal{O} \mid \mathrm{pr}_\Pi O = \mathrm{pr}_\Pi \sigma\}$$

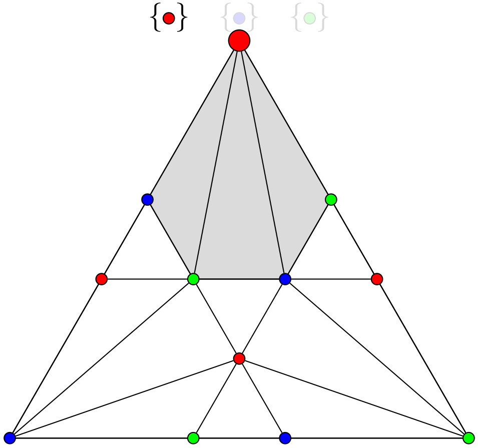
Standard Chromatic Subdivision χ



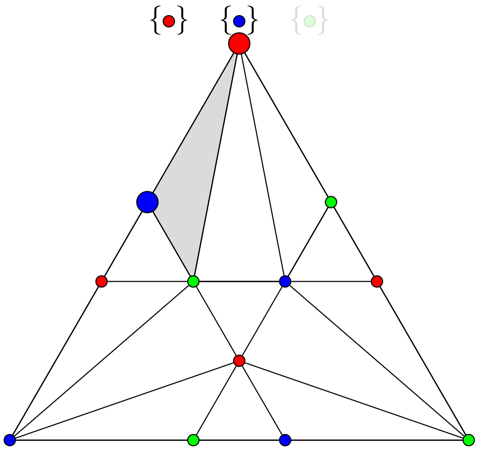
Standard Chromatic Subdivision χ



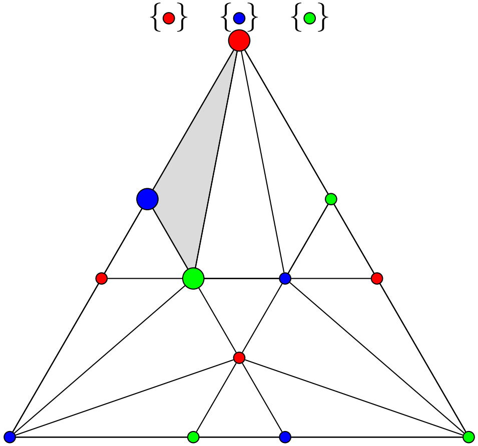
Standard Chromatic Subdivision χ



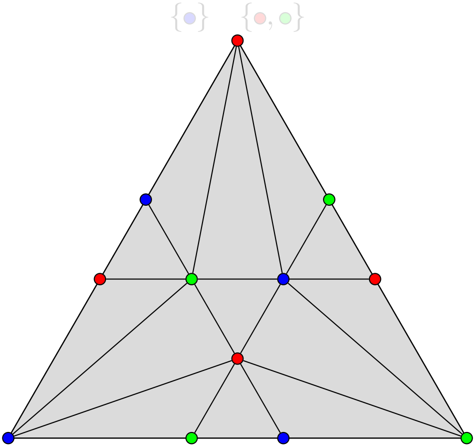
Standard Chromatic Subdivision χ



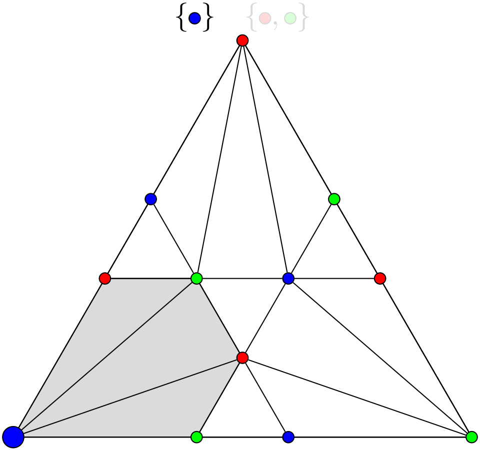
Standard Chromatic Subdivision χ



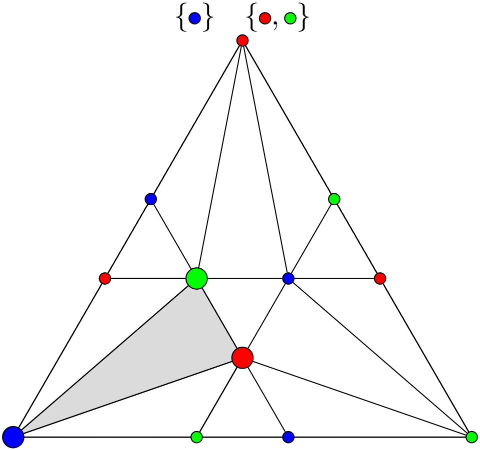
Standard Chromatic Subdivision χ



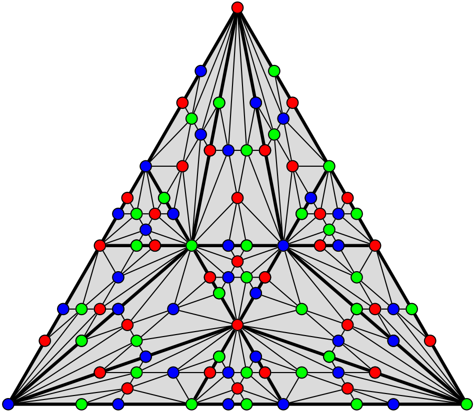
Standard Chromatic Subdivision χ



Standard Chromatic Subdivision χ

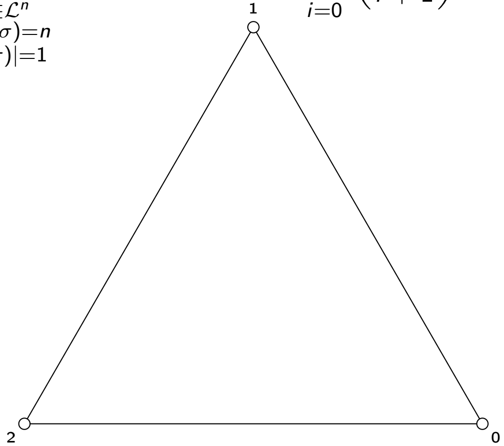


Standard Chromatic Subdivision χ



First Subdivision Step

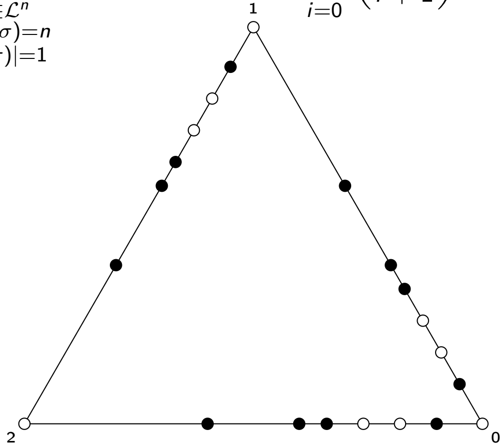
$$\sum_{\substack{\sigma\in\mathcal{L}^n \\ \dim(\sigma)=n \\ |B(\sigma)|=1}} (-1)^{B(\sigma)\cdot n} D(\sigma) \stackrel{!}{=} 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i \stackrel{!}{=} 0$$



$$k_0 = -1,\ k_1 = 1,\ k_2 = 1,\ k_3 = -2,\ k_4 = 0$$

First Subdivision Step

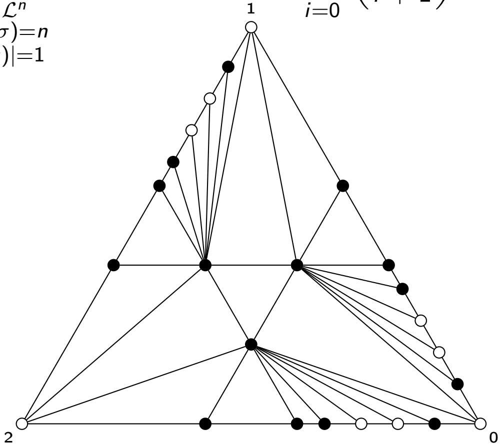
$$\sum_{\substack{\sigma \in \mathcal{L}^n \\ \dim(\sigma)=n \\ |B(\sigma)|=1}} (-1)^{B(\sigma)\cdot n} D(\sigma) \overset{!}{=} 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i \overset{!}{=} 0$$



$$k_0 = -1,\ k_1 = 1,\ k_2 = 1,\ k_3 = -2,\ k_4 = 0$$

First Subdivision Step

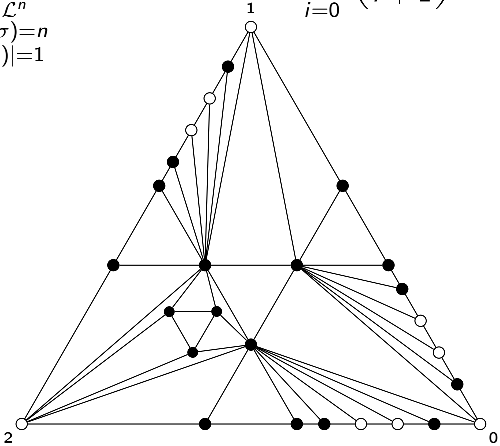
$$\sum_{\substack{\sigma \in \mathcal{L}^n \\ \dim(\sigma)=n \\ |B(\sigma)|=1}} (-1)^{B(\sigma)\cdot n} D(\sigma) \stackrel{!}{=} 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i \stackrel{!}{=} 0$$



$$k_0 = -1,\ k_1 = 1,\ k_2 = 1,\ k_3 = -2,\ k_4 = 0$$

First Subdivision Step

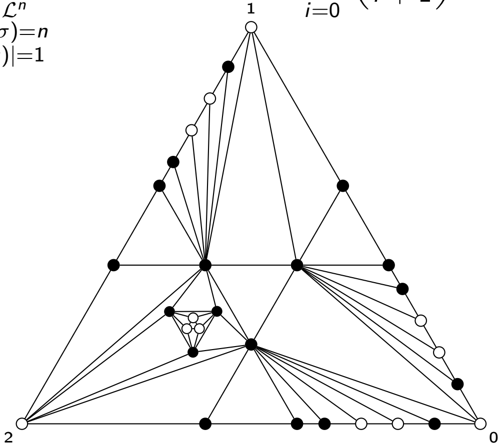
$$\sum_{\substack{\sigma\in\mathcal{L}^n \\ \dim(\sigma)=n \\ |B(\sigma)|=1}} (-1)^{B(\sigma)\cdot n} D(\sigma) \stackrel{!}{=} 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i \stackrel{!}{=} 0$$



$$k_0 = -1,\ k_1 = 1,\ k_2 = 1,\ k_3 = -2,\ k_4 = 0$$

First Subdivision Step

$$\sum_{\substack{\sigma\in\mathcal{L}^n\\ \dim(\sigma)=n\\ |B(\sigma)|=1}} (-1)^{B(\sigma)\cdot n} D(\sigma) \overset{!}{=} 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i \overset{!}{=} 0$$



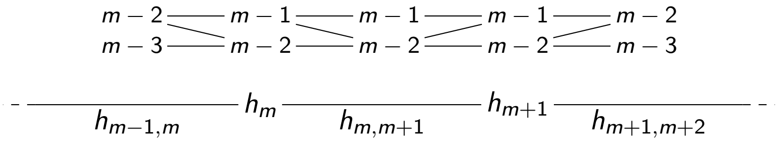
$$k_0 = -1,\ k_1 = 1,\ k_2 = 1,\ k_3 = -2,\ k_4 = 0$$

Subdivision Point

Choose m minimal such that

$$h_{m+1,m+2} \leqslant m-2$$

Then $m \leqslant \frac{\ell}{2}$ and

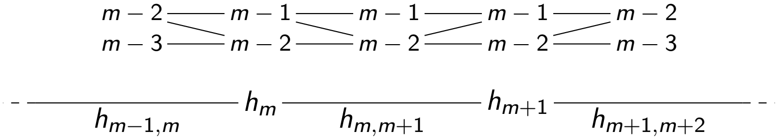


Case Analysis

Case 1 $h_m \neq h_{m+1}$ (Asymmetric)

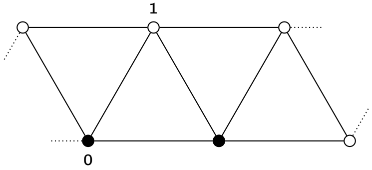
Case 2 $h_m = h_{m+1} = h_{m,m+1}$ (Symmetric 0)

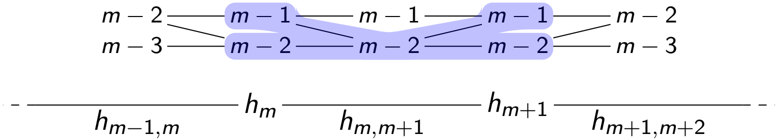
Case 3 $h_m = h_{m+1} \neq h_{m,m+1}$ (Symmetric 1)



Case 1: Asymmetric

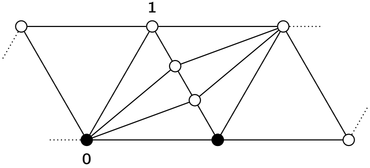
$$h_m \neq h_{m+1}$$

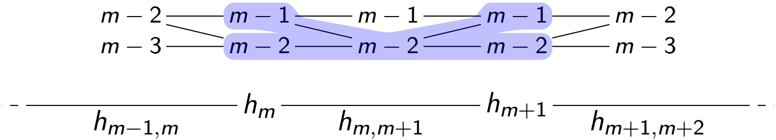




Case 1: Asymmetric

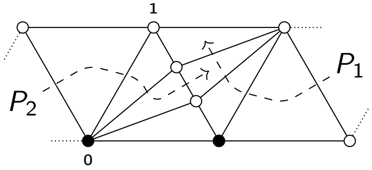
$$h_m \neq h_{m+1}$$

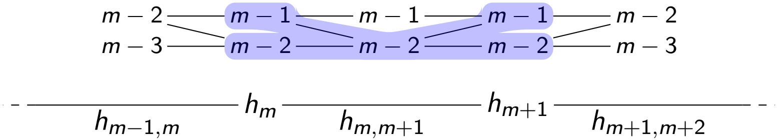




Case 1: Asymmetric

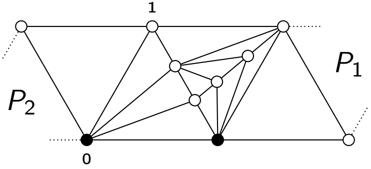
$$h_m \neq h_{m+1}$$

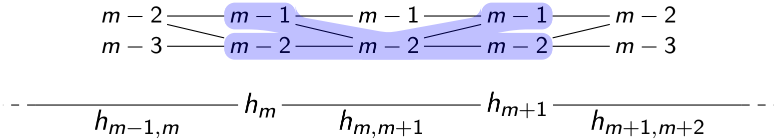




Case 1: Asymmetric

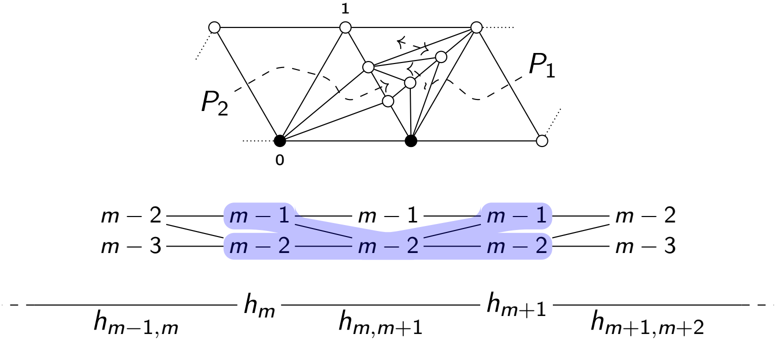
$$h_m \neq h_{m+1}$$





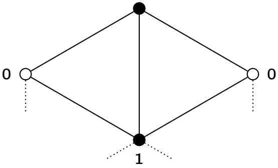
Case 1: Asymmetric

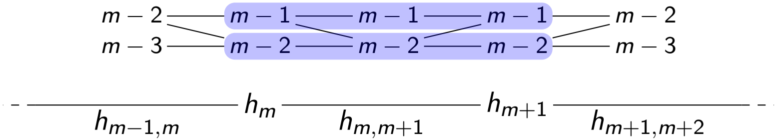
$$h_m \neq h_{m+1}$$



Case 2: Symmetric 0

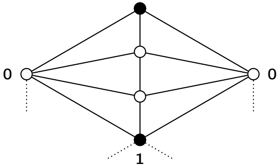
$$h_m = h_{m+1} = h_{m,m+1} \in \{m-2, m-1\}$$

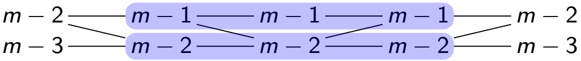


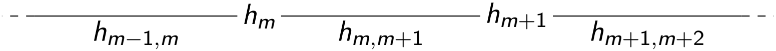


Case 2: Symmetric 0

$$h_m = h_{m+1} = h_{m,m+1} \in \{m-2, m-1\}$$

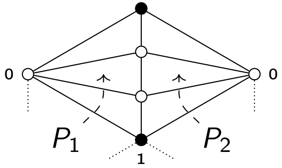


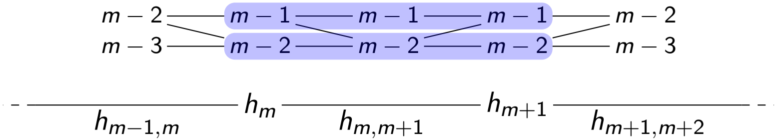




Case 2: Symmetric 0

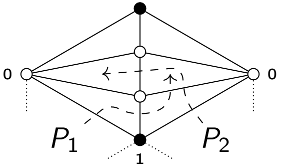
$$h_m = h_{m+1} = h_{m,m+1} \in \{m-2, m-1\}$$

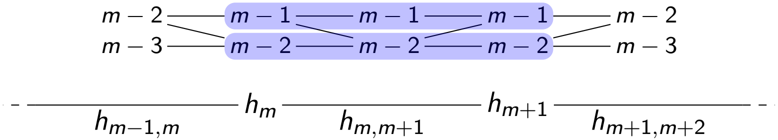




Case 2: Symmetric 0

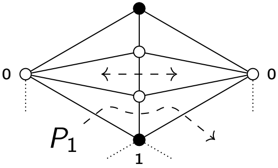
$$h_m = h_{m+1} = h_{m,m+1} \in \{m-2, m-1\}$$

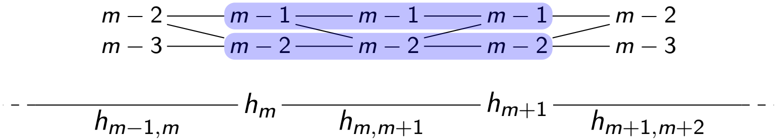




Case 2: Symmetric 0

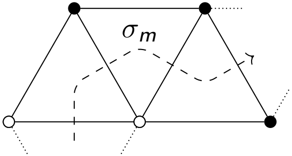
$$h_m = h_{m+1} = h_{m,m+1} \in \{m-2, m-1\}$$

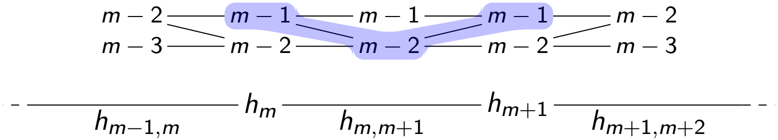




Case 3: Symmetric 1

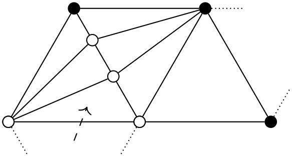
$$h_m = h_{m+1} \neq h_{m,m+1}$$

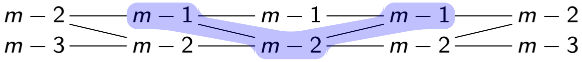


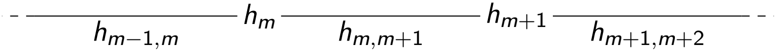


Case 3: Symmetric 1

$$h_m = h_{m+1} \neq h_{m,m+1}$$

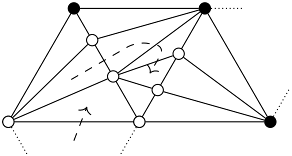


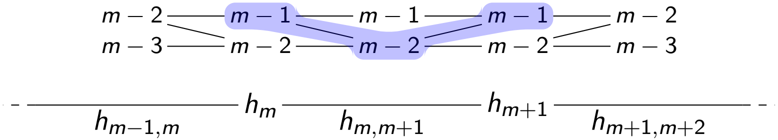




Case 3: Symmetric 1

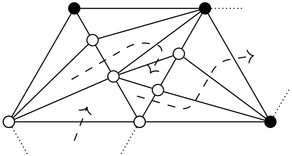
$$h_m = h_{m+1} \neq h_{m,m+1}$$

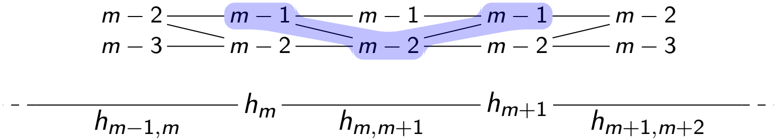




Case 3: Symmetric 1

$$h_m = h_{m+1} \neq h_{m,m+1}$$

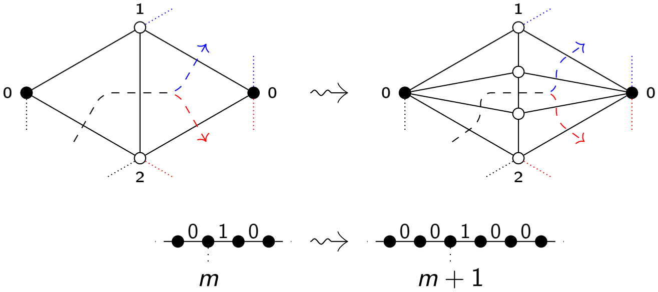




Flatten a unit

- Low admissible path
- Choose even m such that $V_m := B(\sigma_{m+1})_{C_m} = 1$
- $(C_{m-1}, C_m, C_{m+1}) = (1, 0, 1)$ or $(1, 0, 2)$ (up to $S_{[n]}$-action)
- Edge exp. of $\sigma_{m,m+1}$ w/ $D := (-, 0, \dots, 0)$

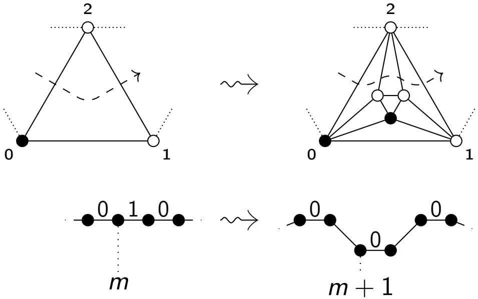
$$Q := (1, 2, 0, 2, 1) \text{ or } (1, 2, 0, 1, 2)$$



Eliminate a unit

- Low admissible path
- Choose (odd) m such that $V_m := B(\sigma_{m+1})_{C_m} = 1$
- $(C_{m-1}, C_m) = (1, 0)$ (up to $S_{[n]}$-action)
- Vertex expansion of σ_m with $D := (0, 0, 1, \ldots, 1)$

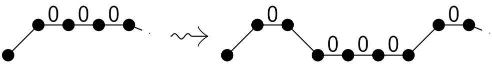
$$Q := (1, 0, 1, 0)$$



Shorten generic zeros

- Low admissible path
- Assume $V = (1, 0, 0, 0, V_5, \dots$
- $C = (0, 1, 2, 3, \dots)$ (up to $S_{[n]}$-action)
- Edge expansion on σ_3 with $D \coloneqq (0, 0, -, 0, 1, \dots, 1)$

$$Q \coloneqq (1, 0, 3, 2, 1, 0, 3)$$



Shorten special zeros

- Low admissible path
- Assume $V = (1, 0, 0, 0, V_5, \ldots$
- $C = (0, 1, 2, 1, \ldots)$ (up to $S_{[n]}$-action)
- Edge expansion on σ_3 with $D := (0, 0, -, 1, 1, \ldots, 1)$

$$Q := (1, 0, 2, 0, 1)$$