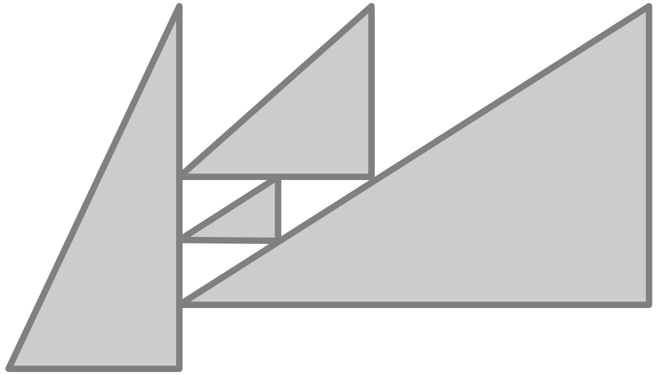
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

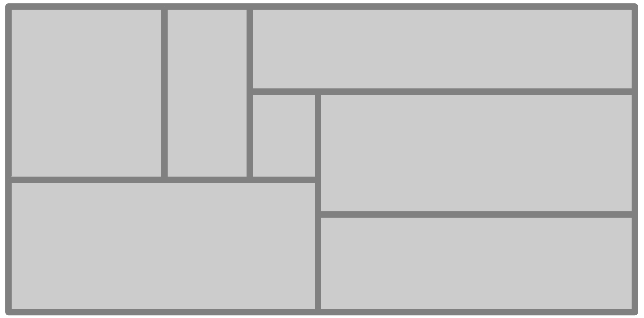
Visualization of graphs

Contact representations of planar graphs

Triangle contacts and rectangular duals

Jonathan Klawitter · Summer semester 2020





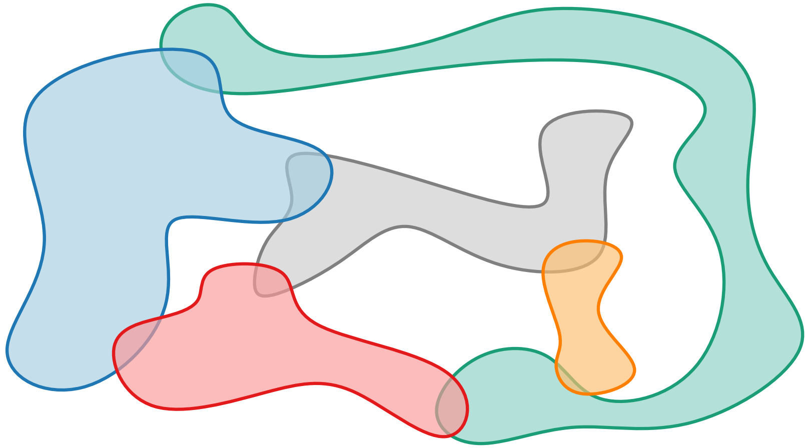
Intersection representation of graphs

Definitions.

In an **intersection representation** of a graph each vertex is represented as a set such that two sets intersect if and only if the corresponding vertices are adjacent.

Intersection representation of graphs

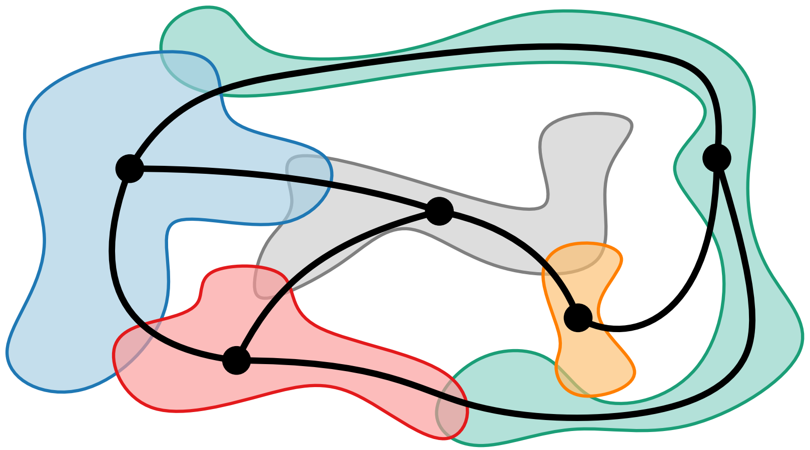
Definitions.
In an **intersection representation** of a graph each vertex is represented as a set such that two sets intersect if and only if the corresponding vertices are adjacent.



Intersection representation of graphs

Definitions.

In an **intersection representation** of a graph each vertex is represented as a set such that two sets intersect if and only if the corresponding vertices are adjacent.

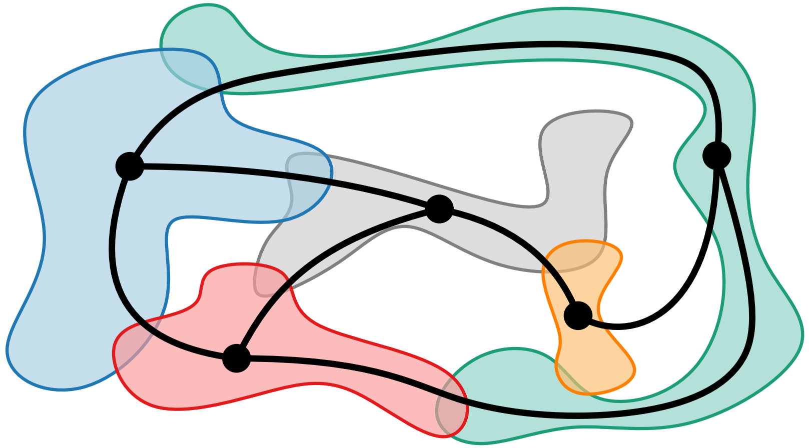


Intersection representation of graphs

Definitions.

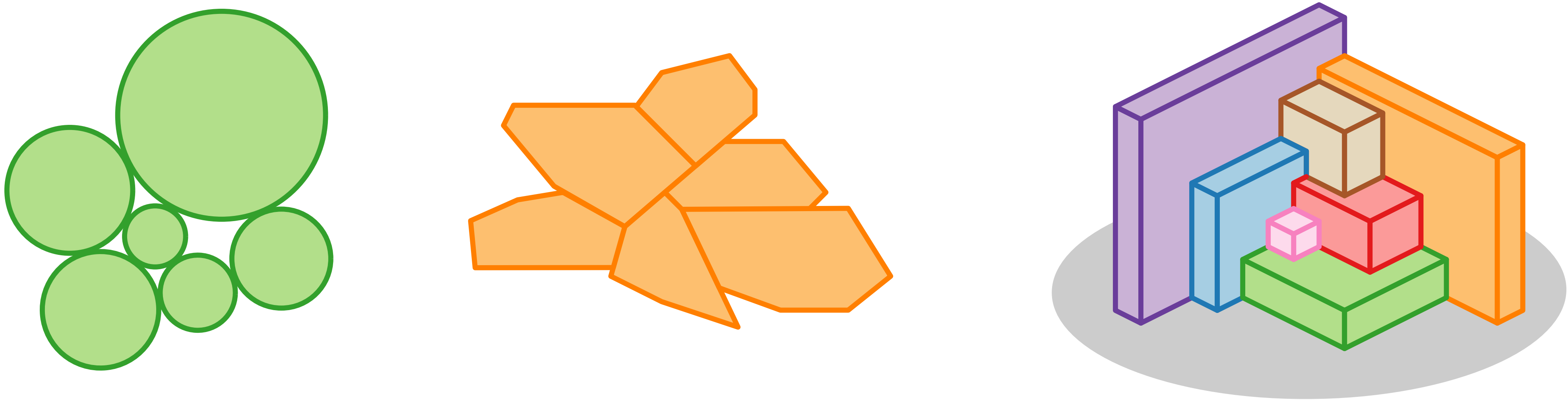
In an **intersection representation** of a graph each vertex is represented as a set such that two sets intersect if and only if the corresponding vertices are adjacent.

For a collection $\mathcal{S}$ of sets $S_1, \ldots, S_n$, the **intersection graph** $G(\mathcal{S})$ of $\mathcal{S}$ has vertex set $\mathcal{S}$ and edge set $\{S_i S_j \colon i, j \in \{1, \ldots, n\}, i \neq j, \text{ and } S_i \cap S_j \neq \emptyset\}$.



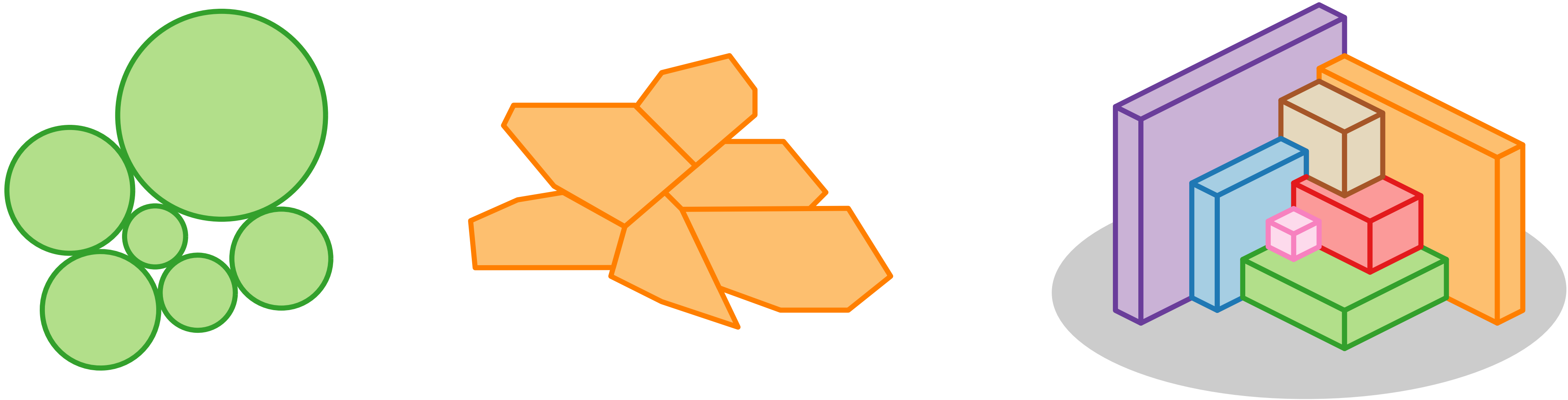
Contact representation of graphs

Definitions.
A collection of interiorly disjoint **objects** $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a called a **contact representation** of its interesction graph $G(\mathcal{S})$.



Contact representation of graphs

Definitions.
A collection of interiorly disjoint **objects** $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a called a **contact representation** of its interesction graph $G(\mathcal{S})$.



- Objects could be circles, line segments, triangles, boxes, ...
- Domain could be 2D, 3D, ...

Contact representation of planar graphs

- Is the intersection graph of a contact representation always planar?

Contact representation of planar graphs

- Is the intersection graph of a contact representation always planar?
  - No, not even for planar object types.

Contact representation of planar graphs

- Is the intersection graph of a contact representation always planar?
  - No, not even for planar object types.
- Which object types can be used to represent all planar graphs?
  - Contact of disks [Koebe '36]
  - Corner contact of triangles and T-shapes [de Fraysseix et al. '94]
  - Side contacts of 3D Boxes [Thomassen '86]
  - ...

Contact representation of planar graphs

- Is the intersection graph of a contact representation always planar?
  - No, not even for planar object types.
- Which object types can be used to represent all planar graphs?
  - Contact of disks [Koebe '36]
  - Corner contact of triangles and T-shapes [de Fraysseix et al. '94]
  - Side contacts of 3D Boxes [Thomassen '86]
  - . . .
- Some object types are used to represent special classes of planar graphs:
  - Line segment contact on grids for bipartite planar graphs [Hartman et al. '91, de Fraysseix et al. '94]
  - Rectangle dissections for so-called properly triangulated planar graphs [Kant, He '97]
  - L-shapes, k-bend path, . . .

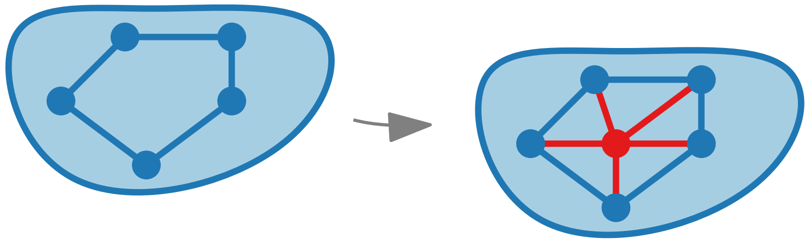
General approach

How to compute a contact representation of a given graph G?

General approach

How to compute a contact representation of a given graph G?

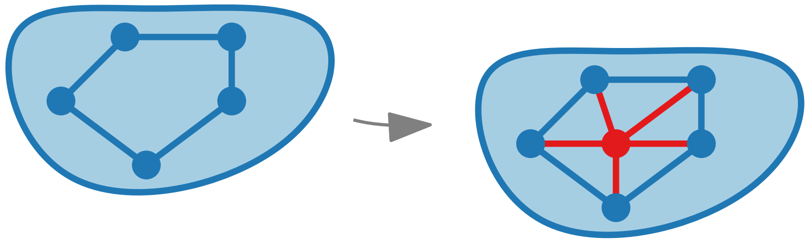
- Consider only inner triangulations (or maximally bipartite graphs, etc)
  - Triangulate by adding vertices, not by adding edges



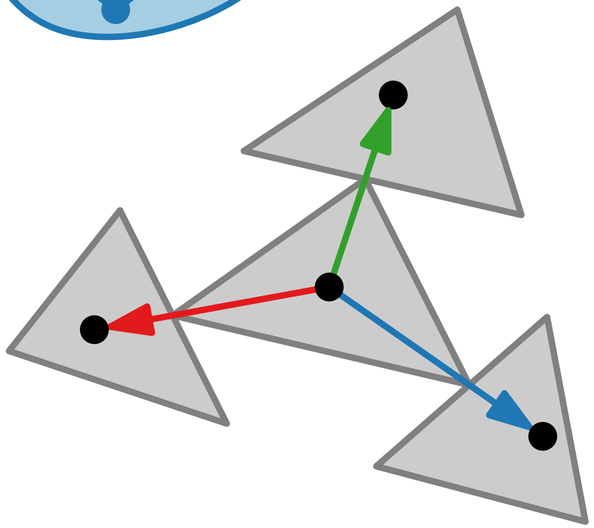
General approach

How to compute a contact representation of a given graph G?

- Consider only inner triangulations (or maximally bipartite graphs, etc)
  - Triangulate by adding vertices, not by adding edges



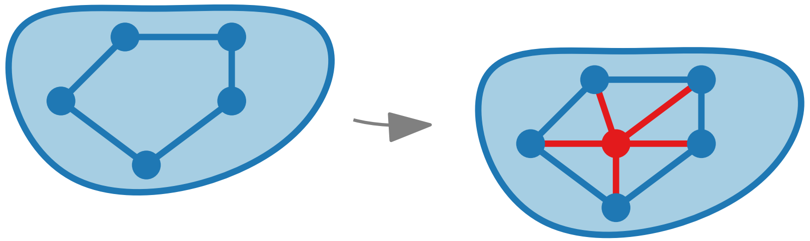
- Describe contact representation combinatorically.
  - Which objects contact each other in which way?



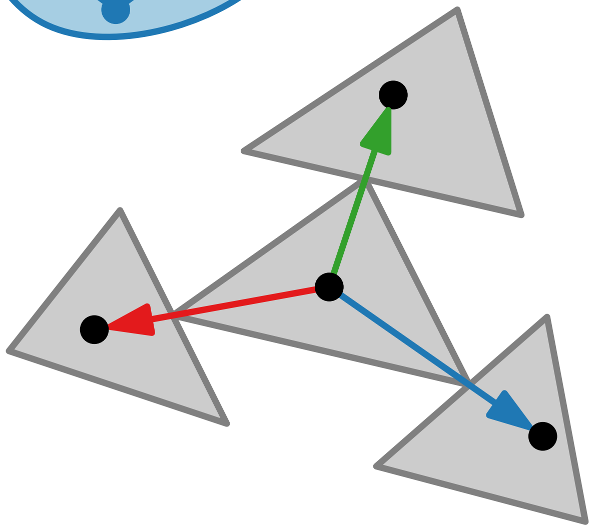
General approach

How to compute a contact representation of a given graph G?

- Consider only inner triangulations (or maximally bipartite graphs, etc)
  - Triangulate by adding vertices, not by adding edges



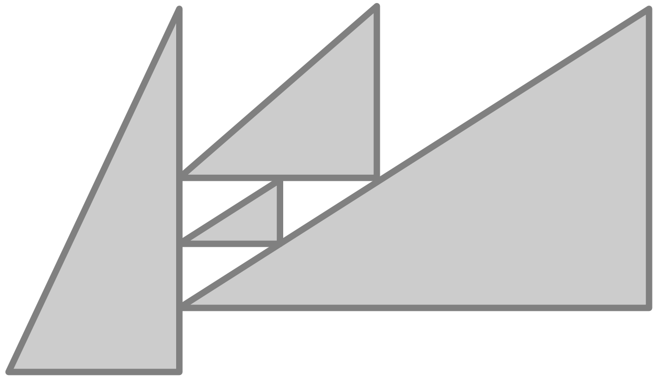
- Describe contact representation combinatorically.
  - Which objects contact each other in which way?



- Compute combinatorical description.
- Show that combinatorical description can be used to construct drawing.

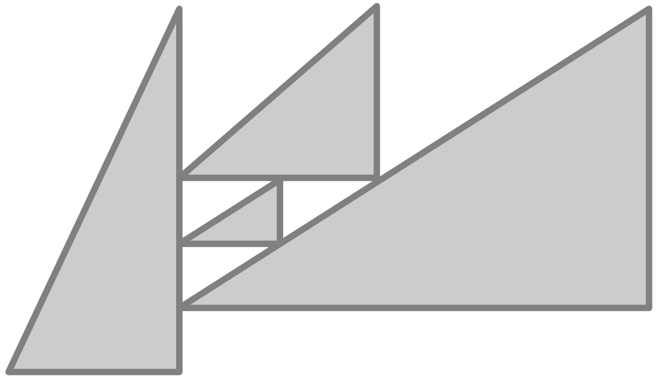
In this lecture

- Representations with right-triangles and corner contact
  - Use Schnyder realizer to describe contacts between triangles
  - Use canonical order to calculate drawing

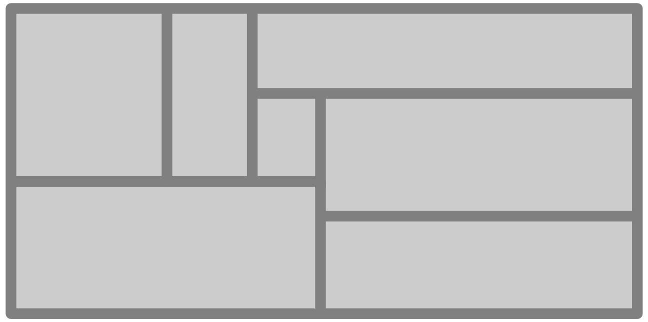


In this lecture

- Representations with right-triangles and corner contact
  - Use Schnyder realizer to describe contacts between triangles
  - Use canonical order to calculate drawing



- Representation with dissection of a rectangle, called **rectangular dual**
  - Find similar description like Schnyder realizer for rectangles
  - Construct drawing via st-digraphs, duals, and topological sorting.

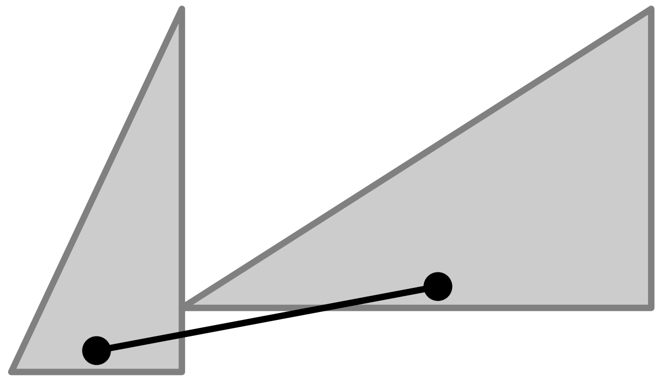


Triangle corner contact representation

Idea.
Use canonical order and Schnyder forest to find coordinates for triangles.

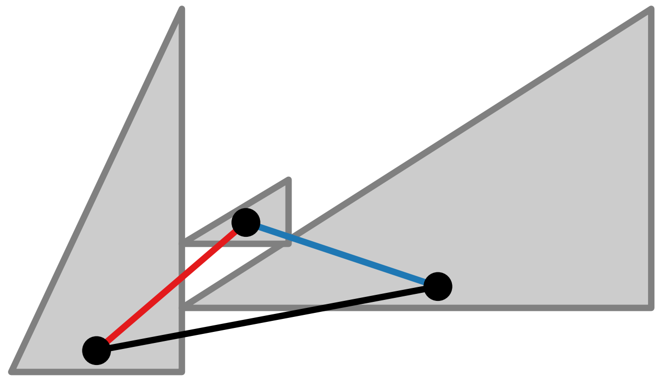
Triangle corner contact representation

Idea.
Use canonical order and Schnyder forest to find coordinates for triangles.



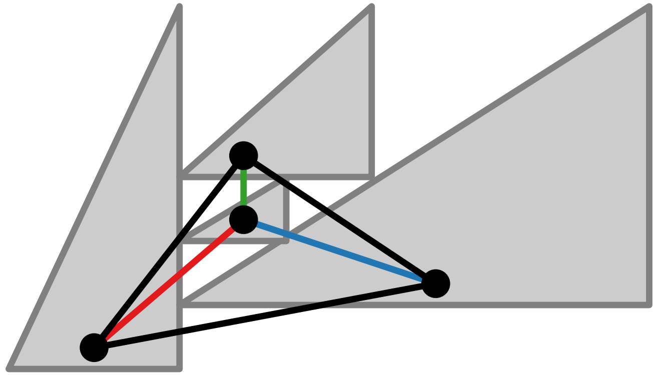
Triangle corner contact representation

Idea.
Use canonical order and Schnyder forest to find coordinates for triangles.



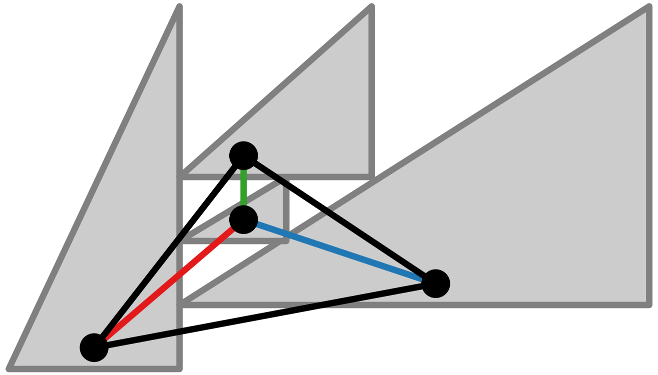
Triangle corner contact representation

Idea.
Use canonical order and Schnyder forest to find coordinates for triangles.



Triangle corner contact representation

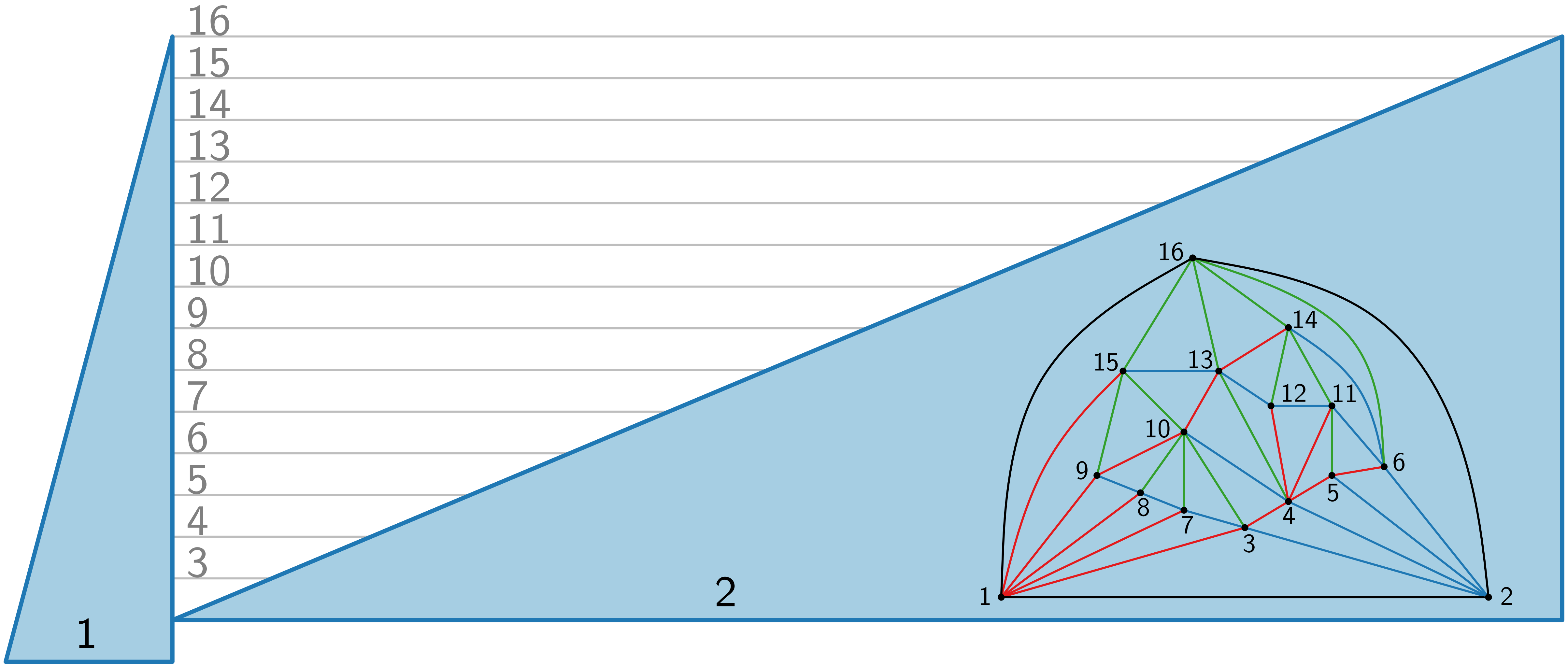
Idea.
Use canonical order and Schnyder forest to find coordinates for triangles.



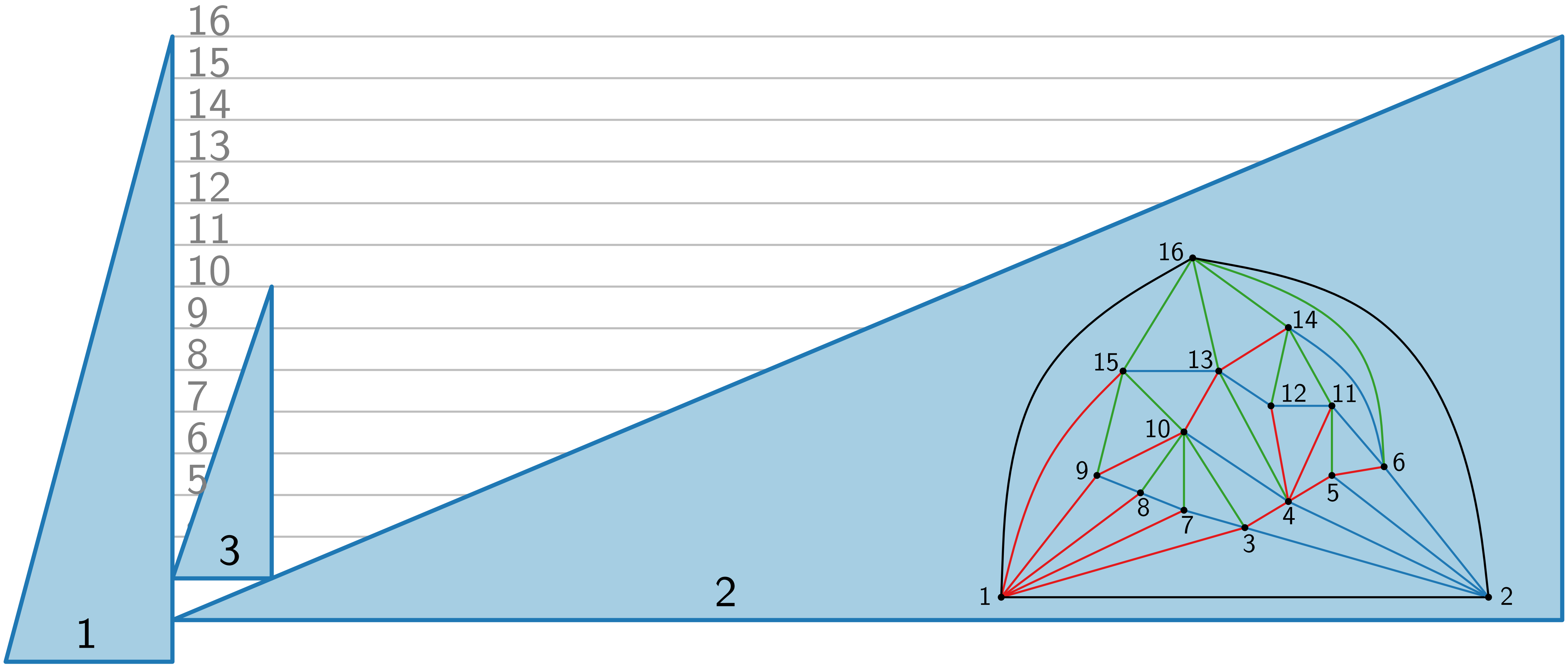
Observation.

- Can set base of triangle at height equal to position in canonical order.
- Triangle tip is precisely at base of triangle corresponding to cover neighbor.
- Outgoing edges in Schnyder forest indicate corner contacts.

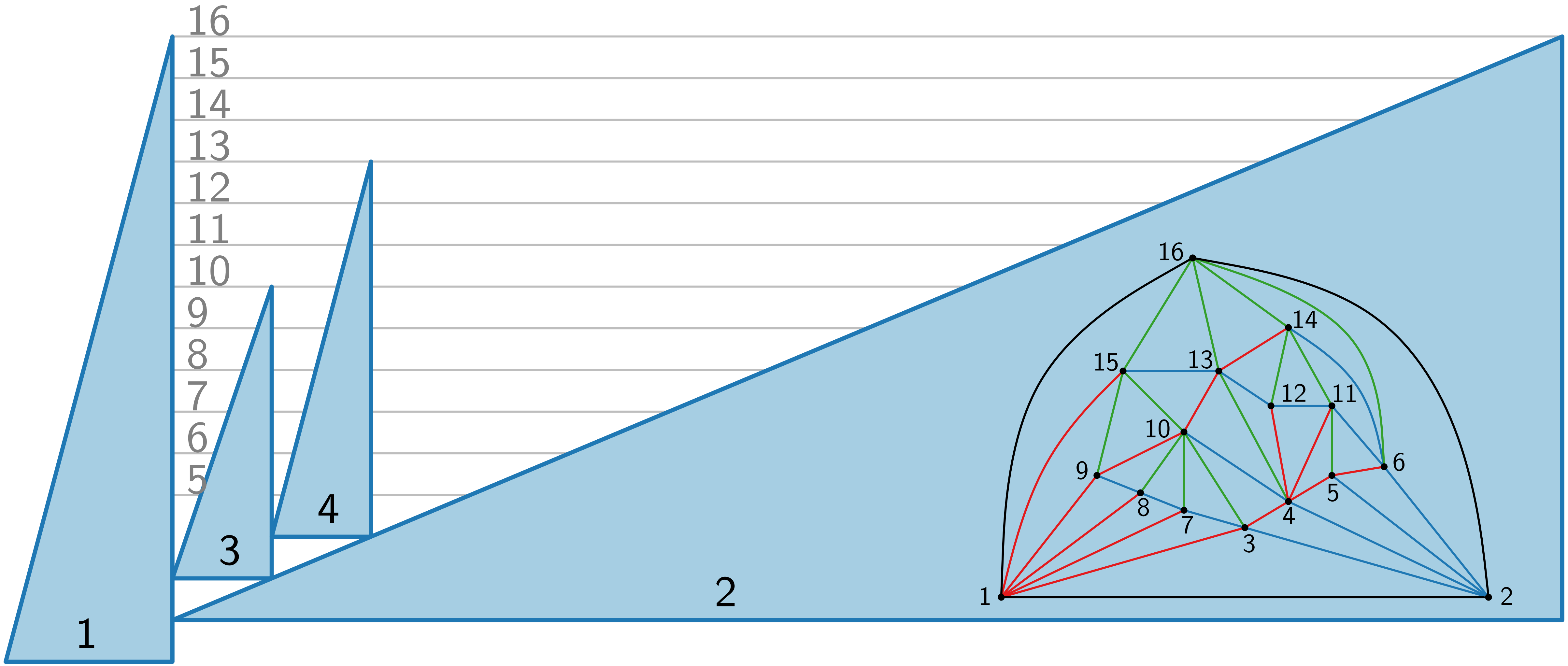
Triangle-contact representation example



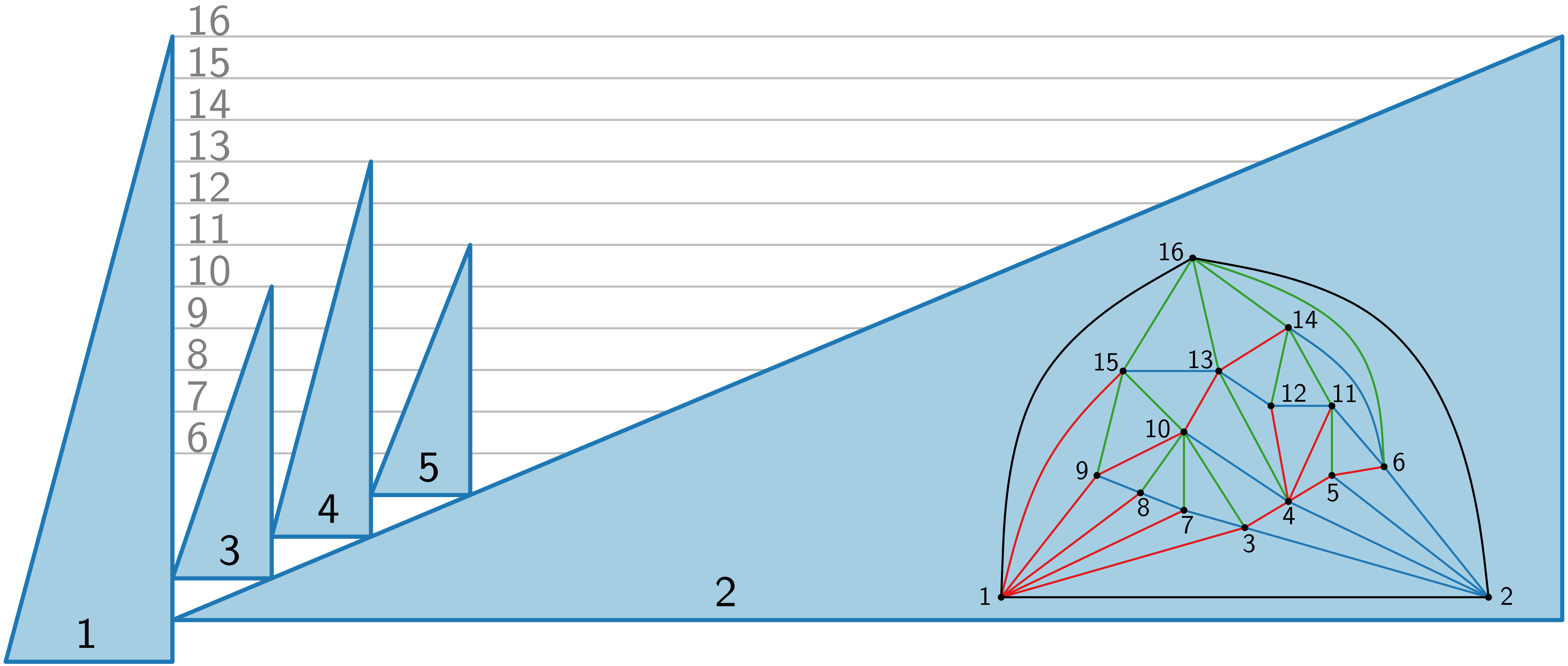
Triangle-contact representation example



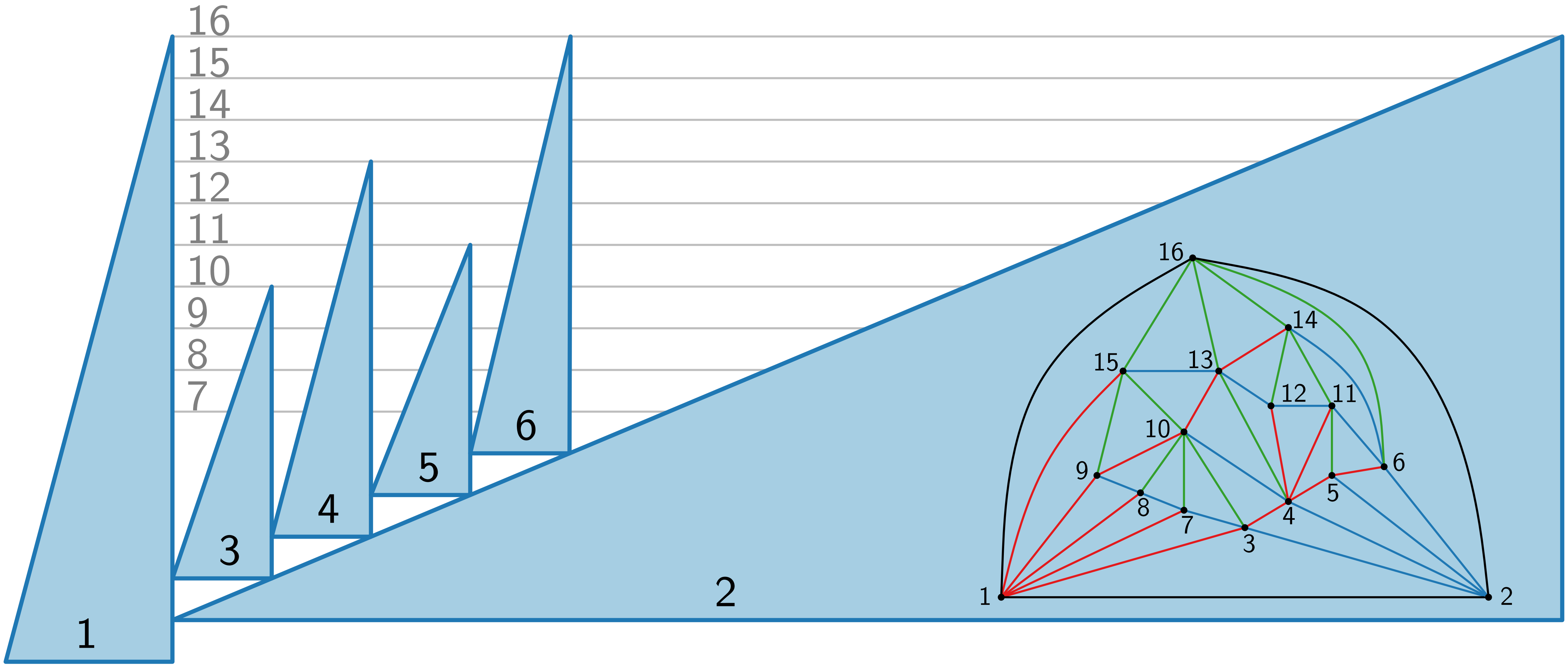
Triangle-contact representation example



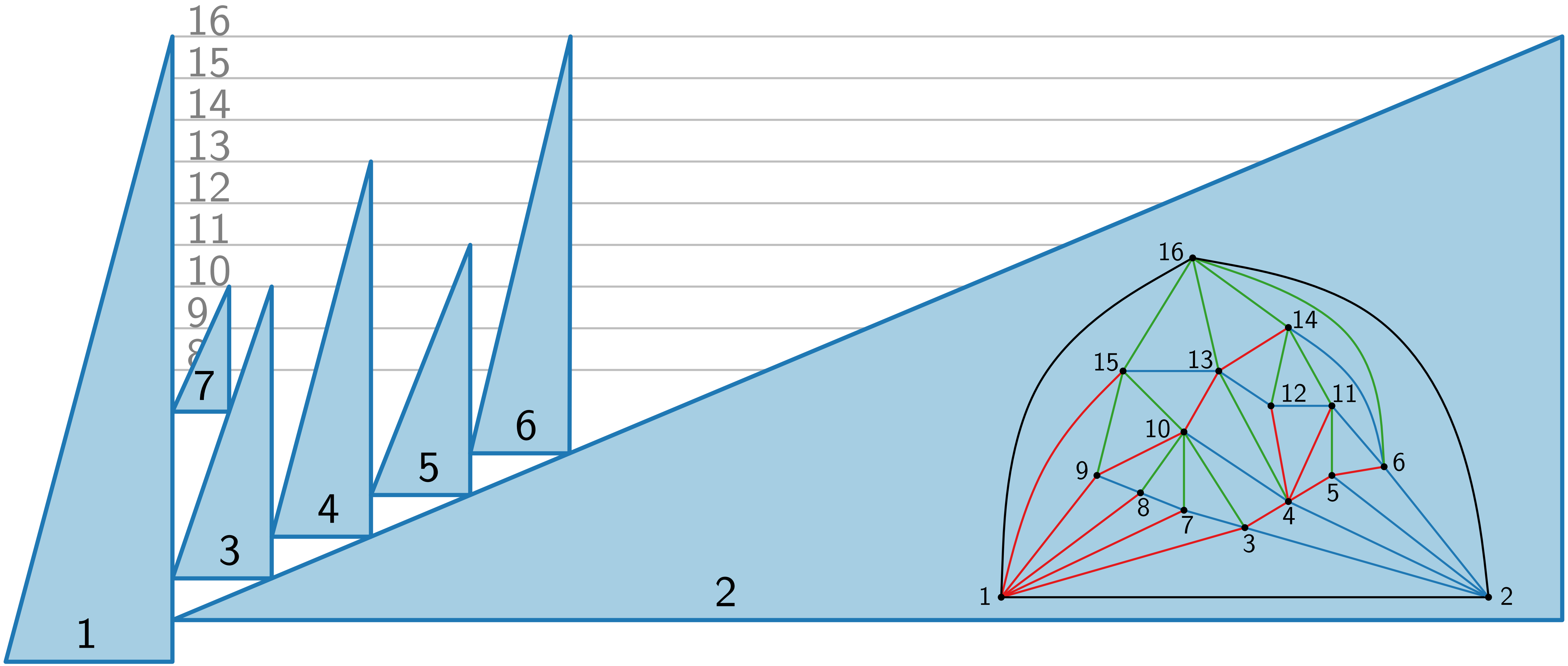
Triangle-contact representation example



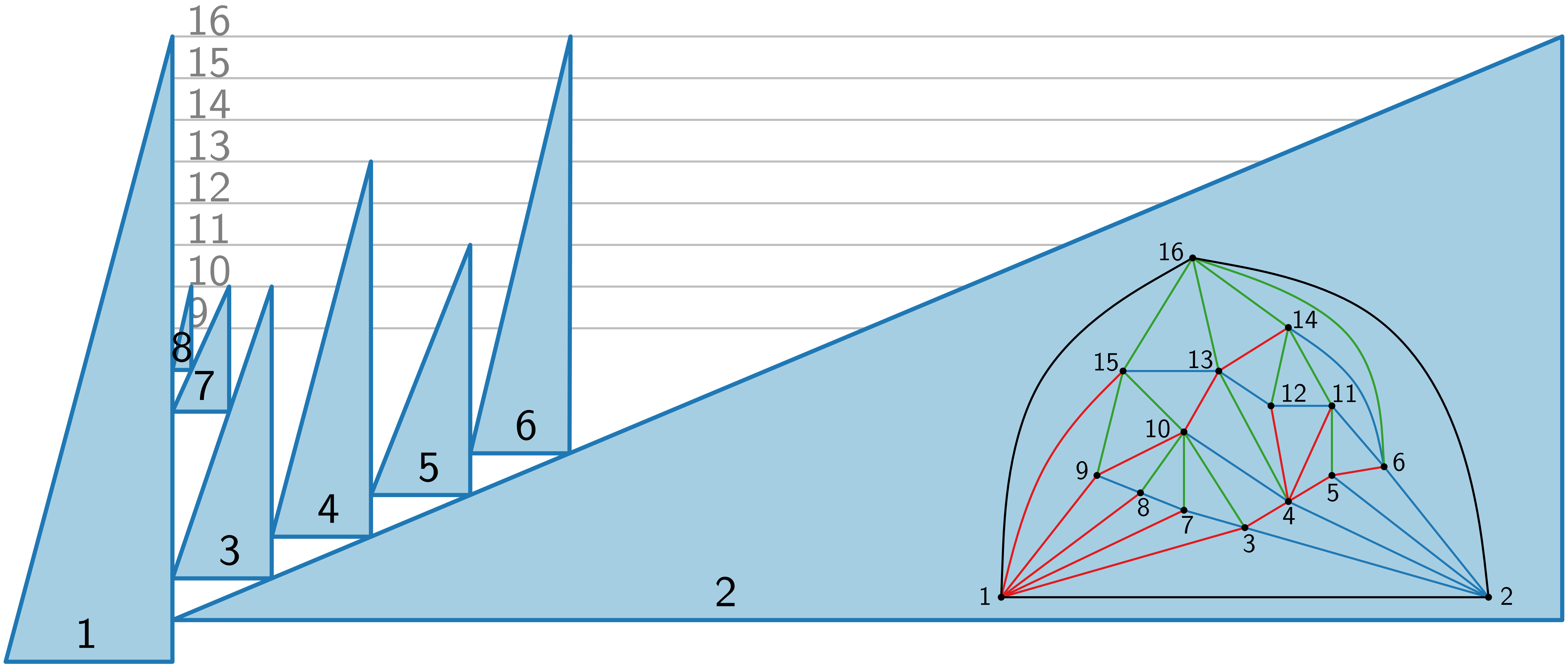
Triangle-contact representation example



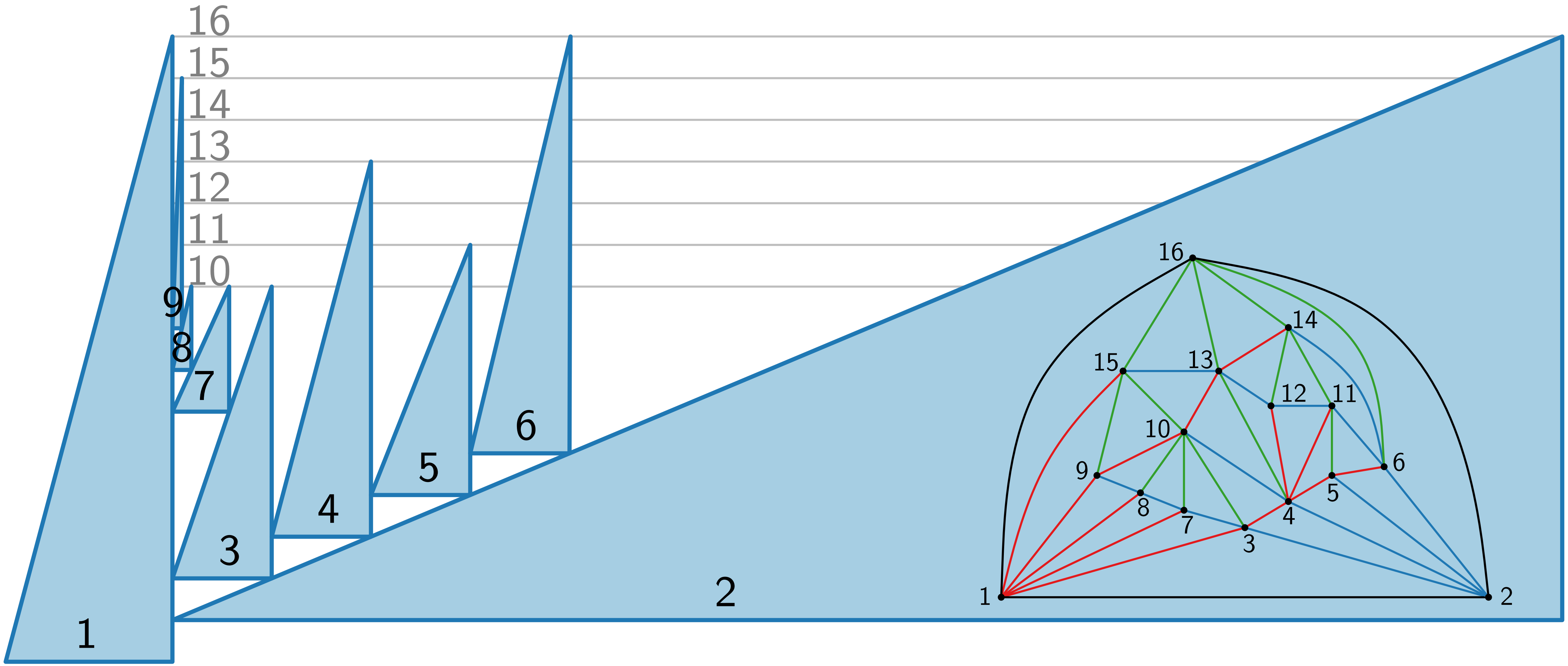
Triangle-contact representation example



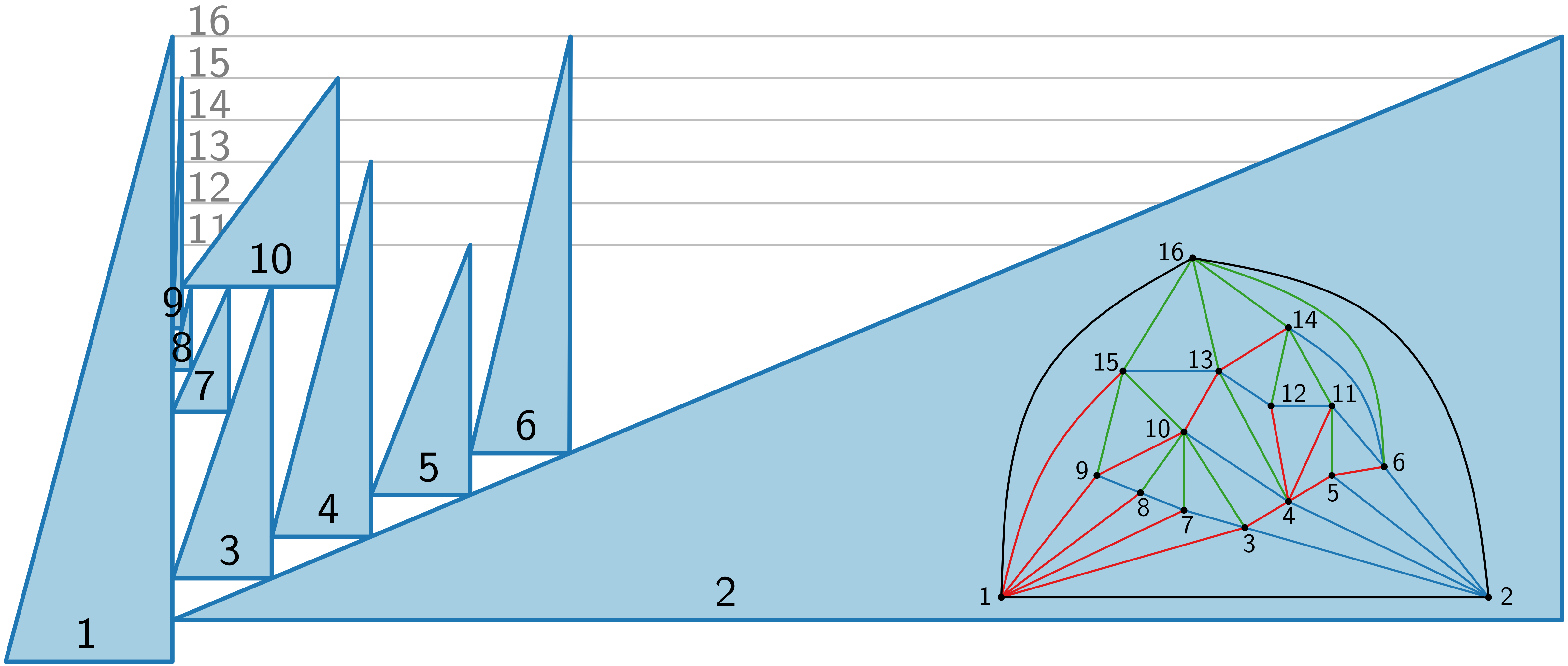
Triangle-contact representation example



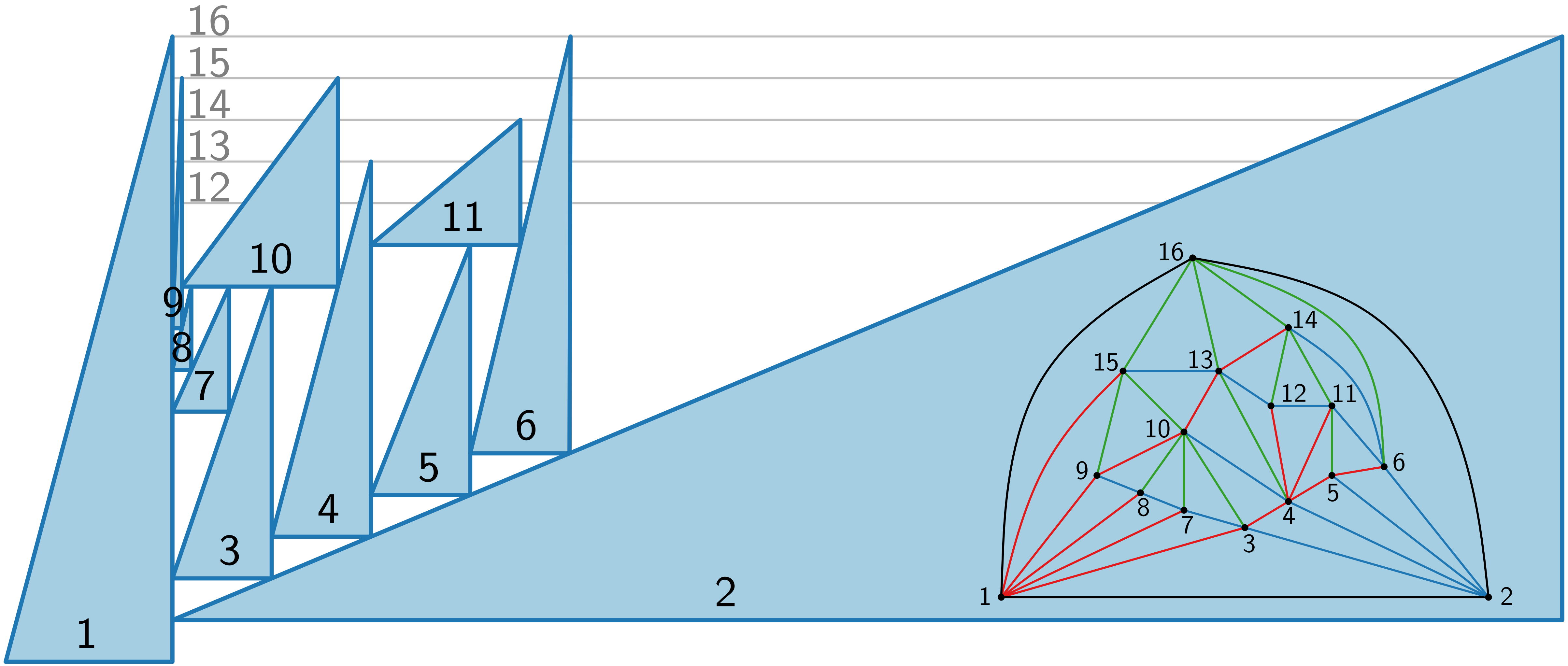
Triangle-contact representation example



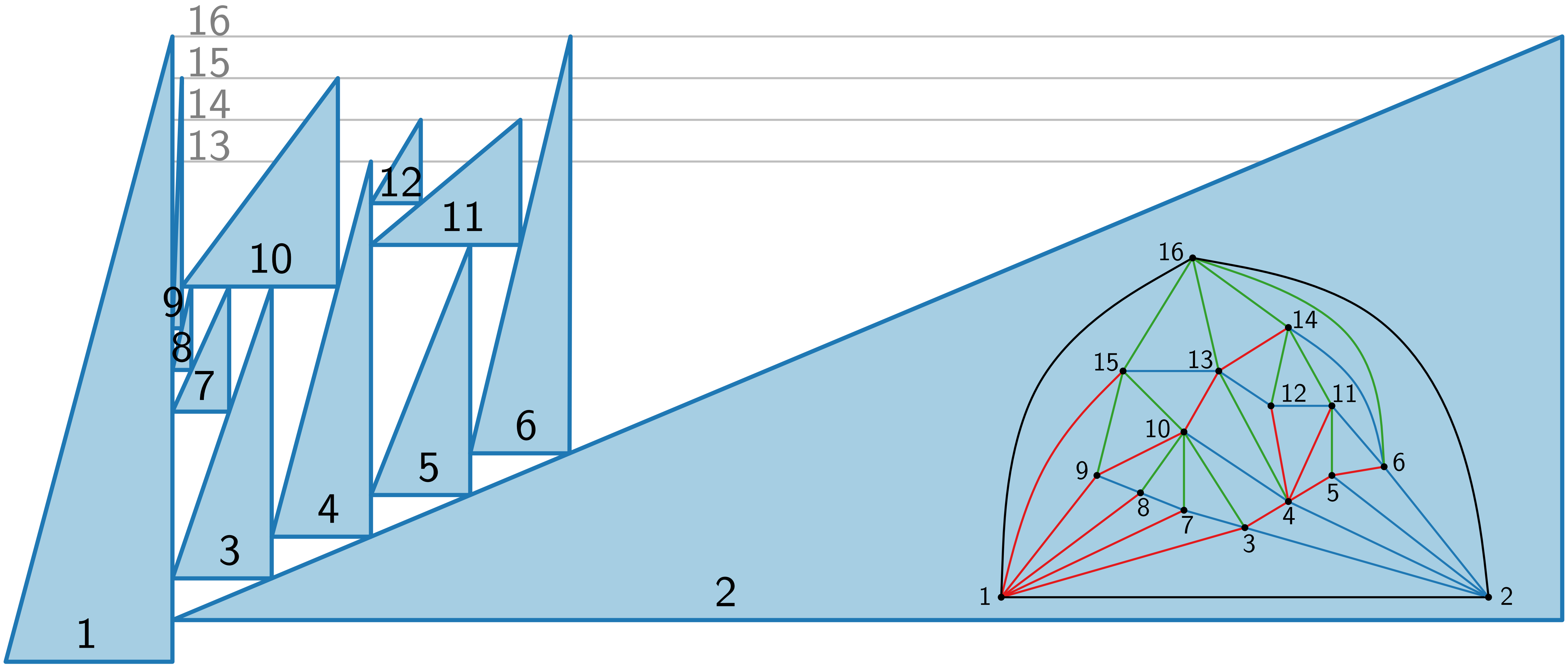
Triangle-contact representation example



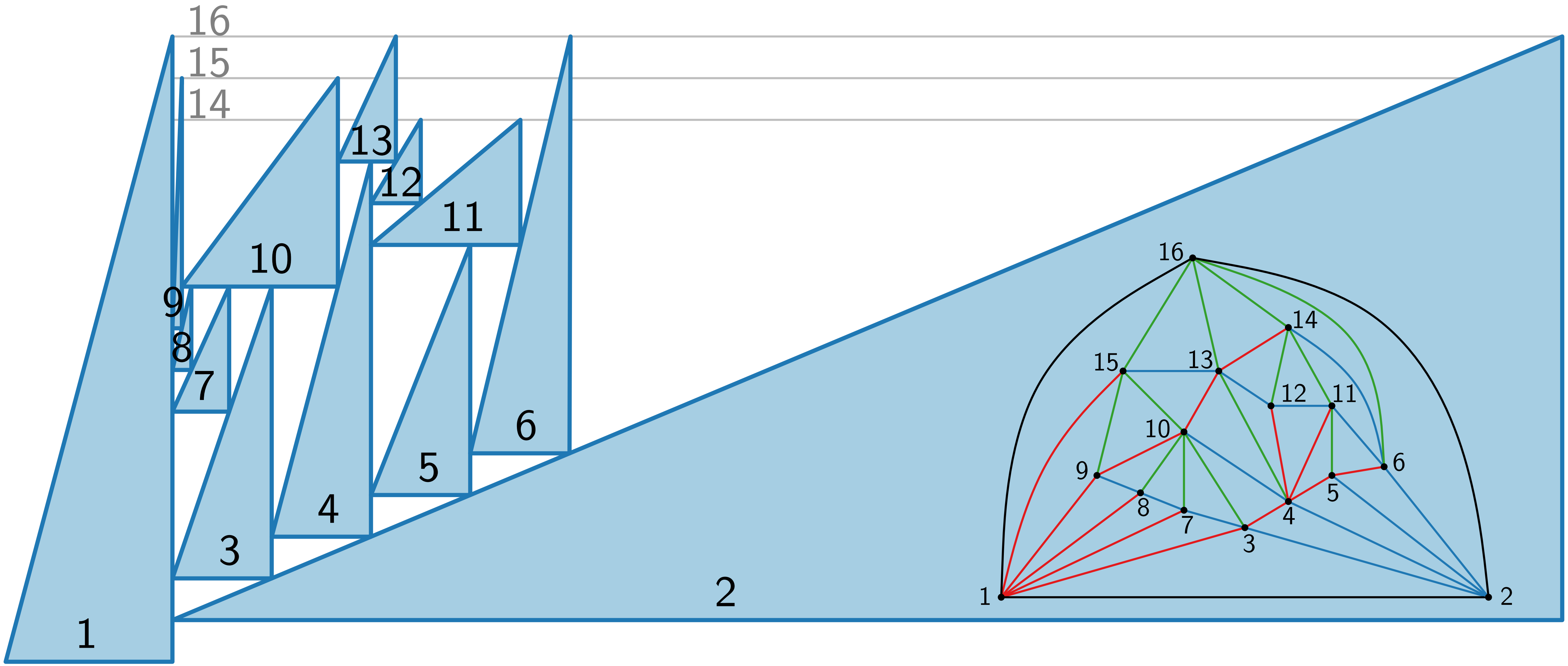
Triangle-contact representation example



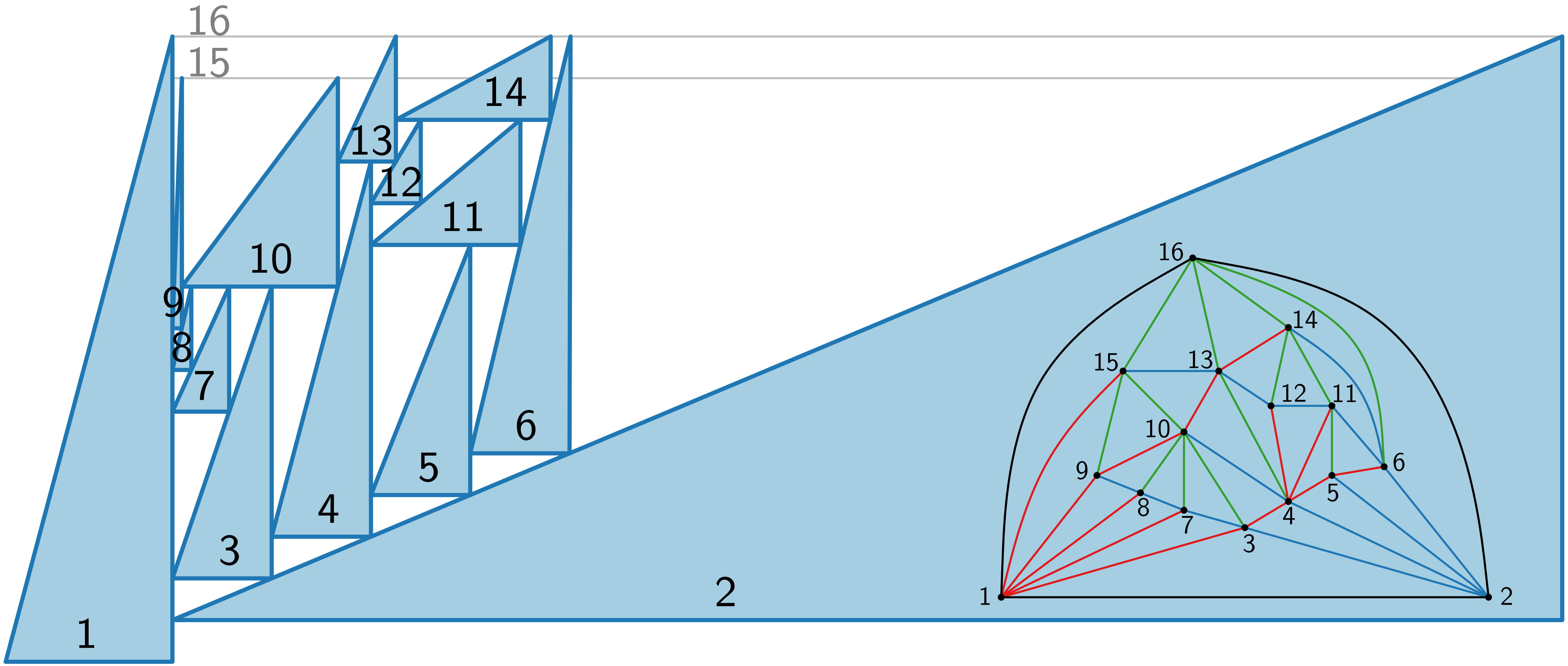
Triangle-contact representation example



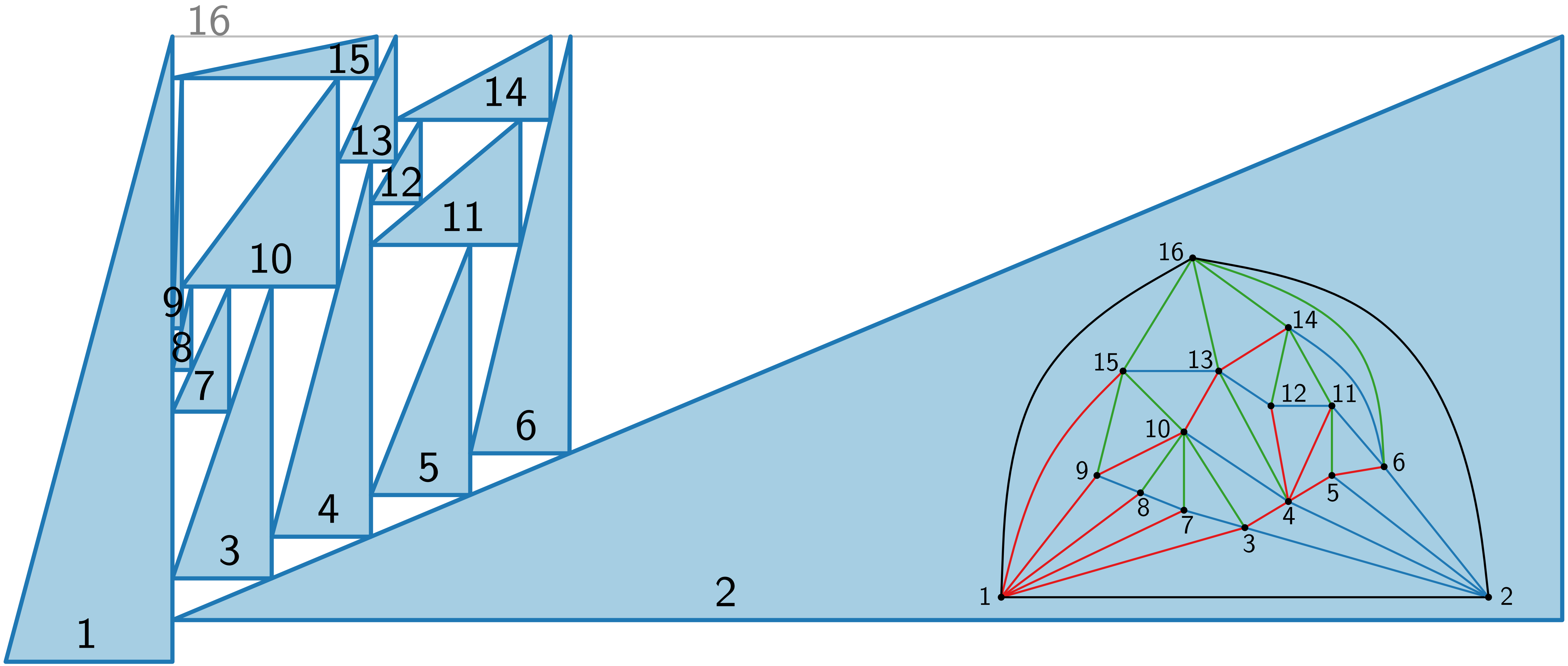
Triangle-contact representation example



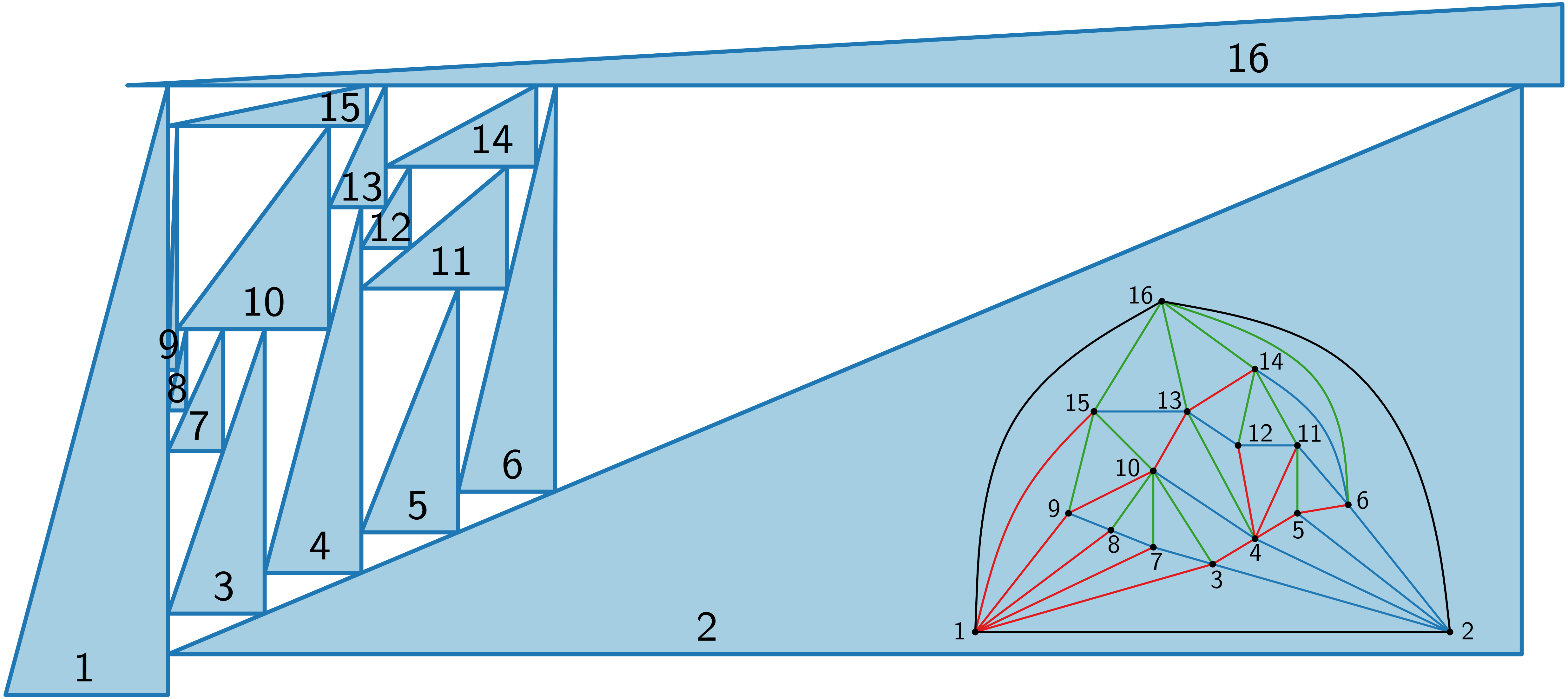
Triangle-contact representation example



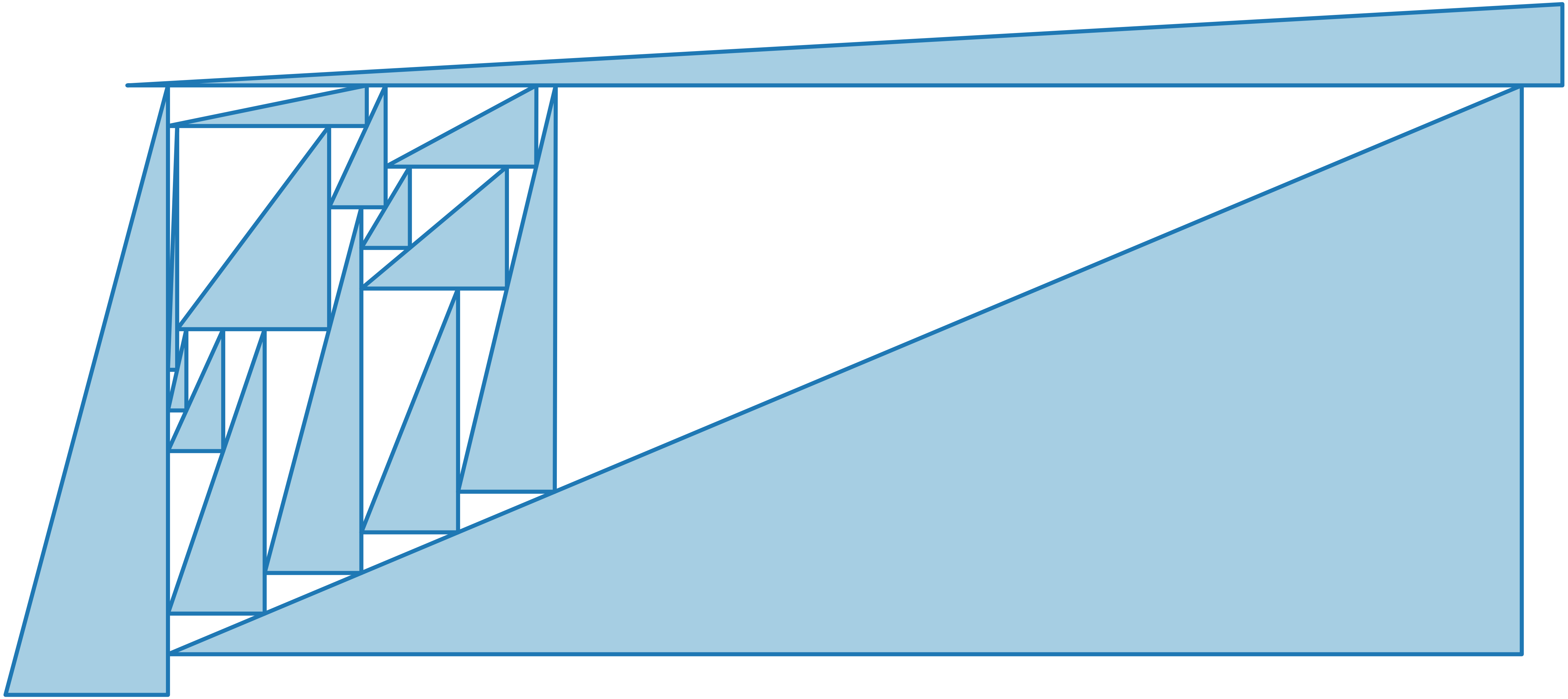
Triangle-contact representation example



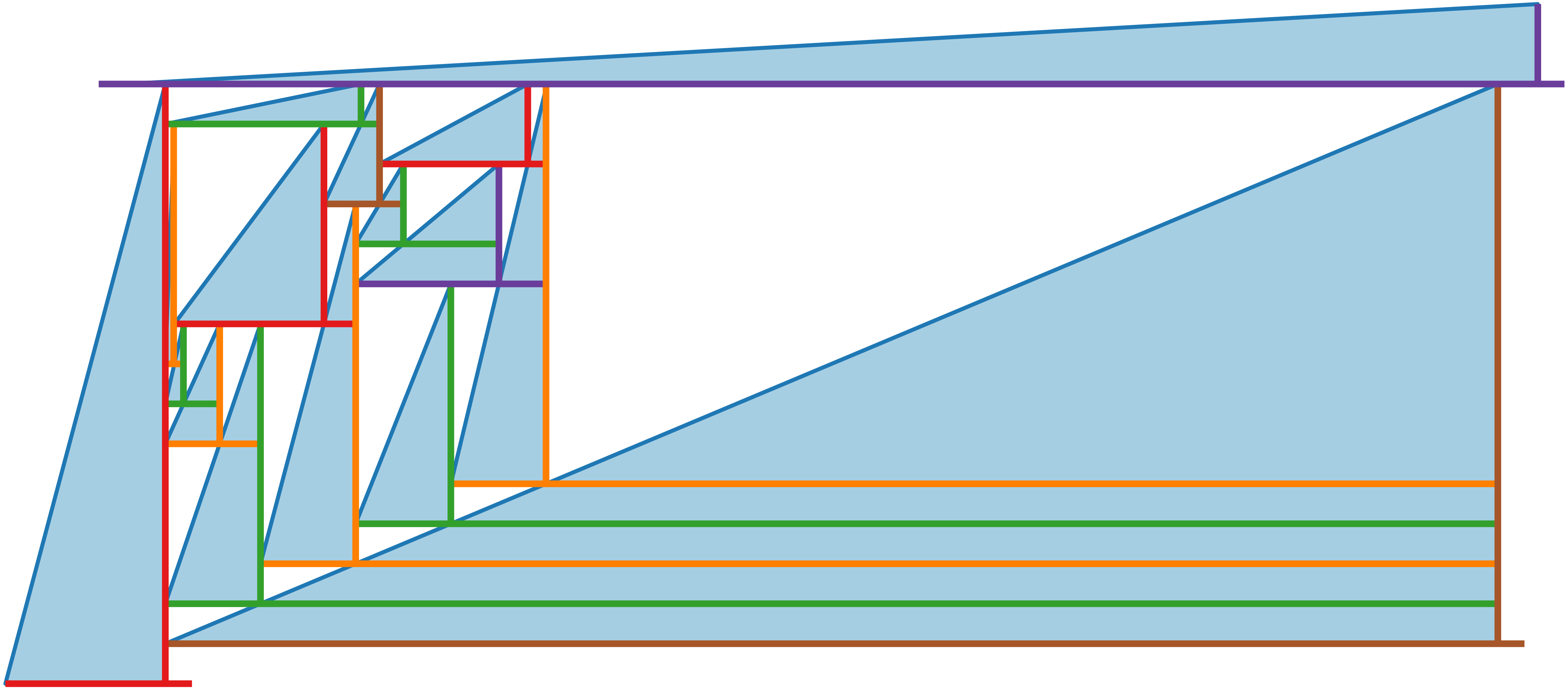
Triangle-contact representation example



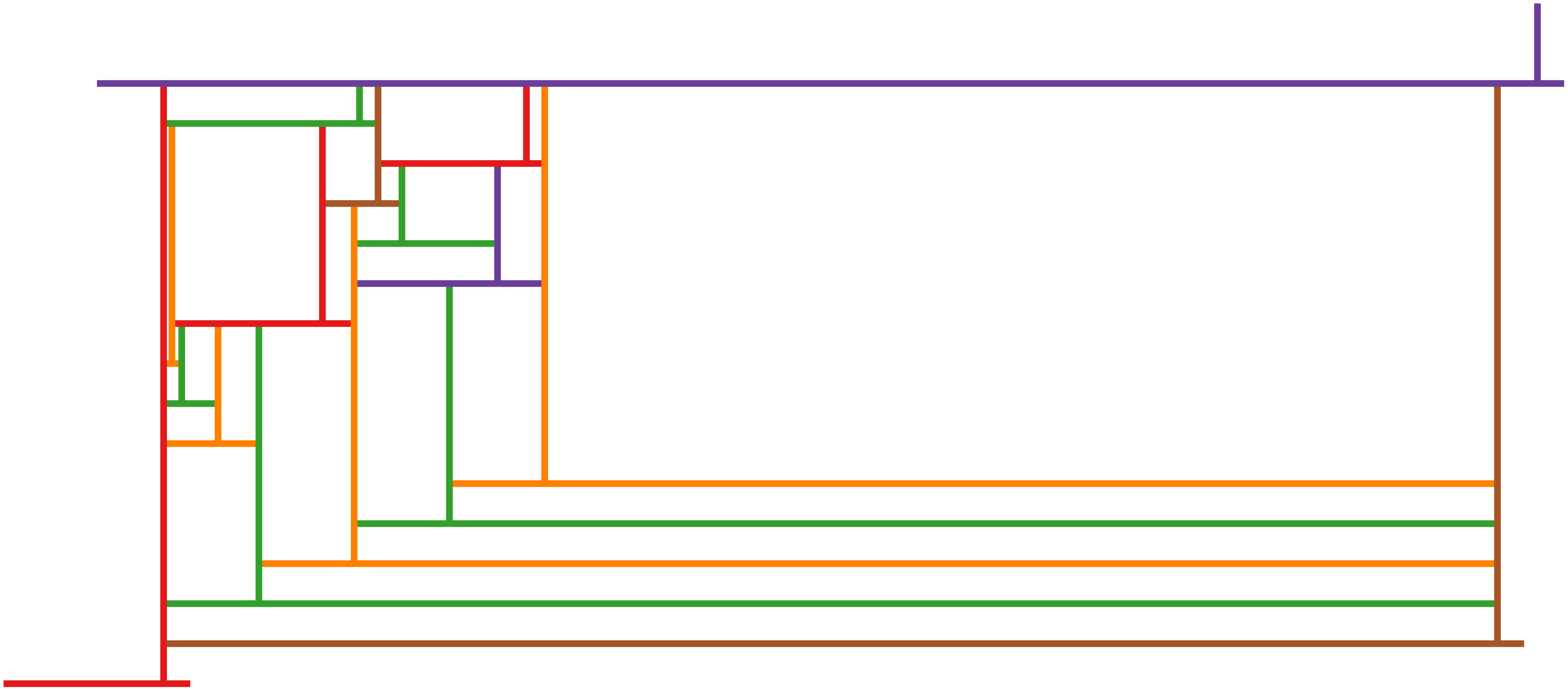
T-shape contact representation



T-shape contact representation



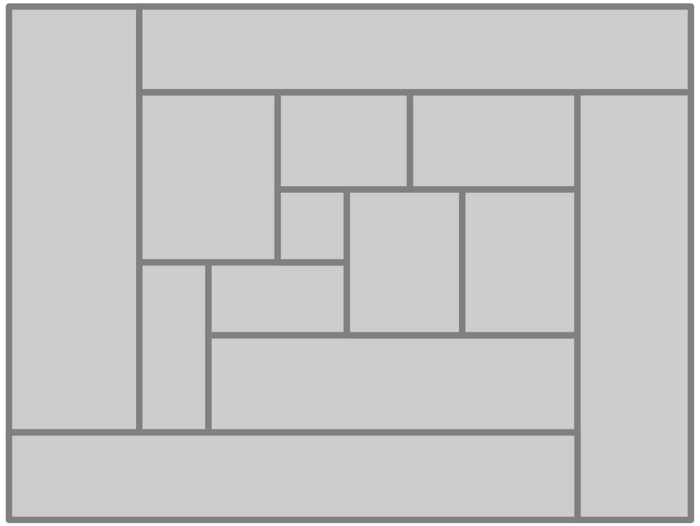
T-shape contact representation



Rectangular dual

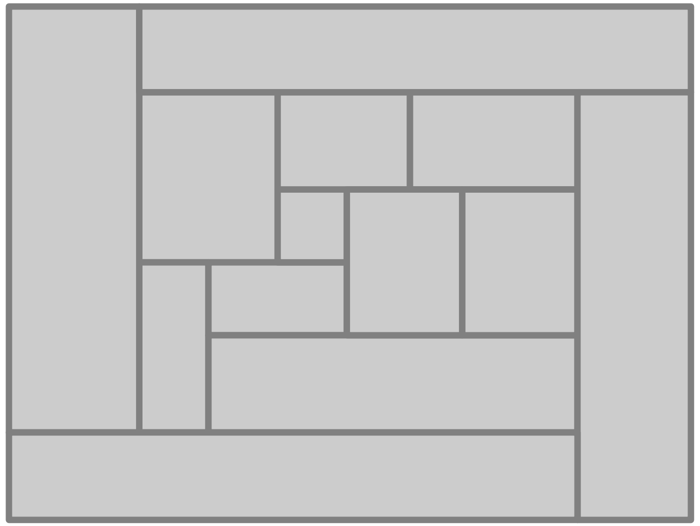
Definition.
A **rectangular dual** of a graph G is a contact representation with axis aligned rectangles s.t.

- no four rectangles share a point, and
- the union of all rectangles is a rectangle.



Rectangular dual

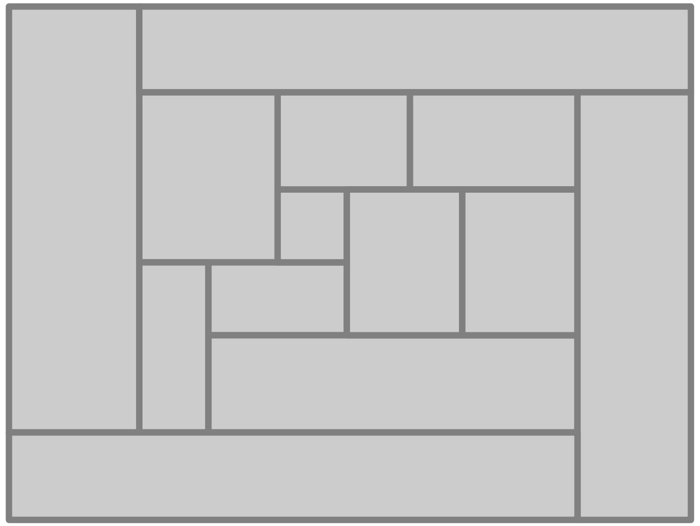
Definition.
A **rectangular dual** of a graph G is a contact representation with axis aligned rectangles s.t.
- no four rectangles share a point, and
- the union of all rectangles is a rectangle.



When does G admit a rectangular dual?

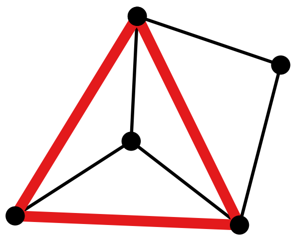
Rectangular dual

Definition.
A **rectangular dual** of a graph G is a contact representation with axis aligned rectangles s.t.
- no four rectangles share a point, and
- the union of all rectangles is a rectangle.



When does G admit a rectangular dual?

Definition.
A triangle C of G whose removal results in at least two connected components is called a **separating triangle**.

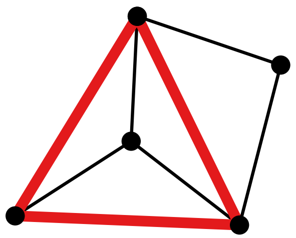


Does not have a rectangular dual. To enclose an area we need at least four rectangles.

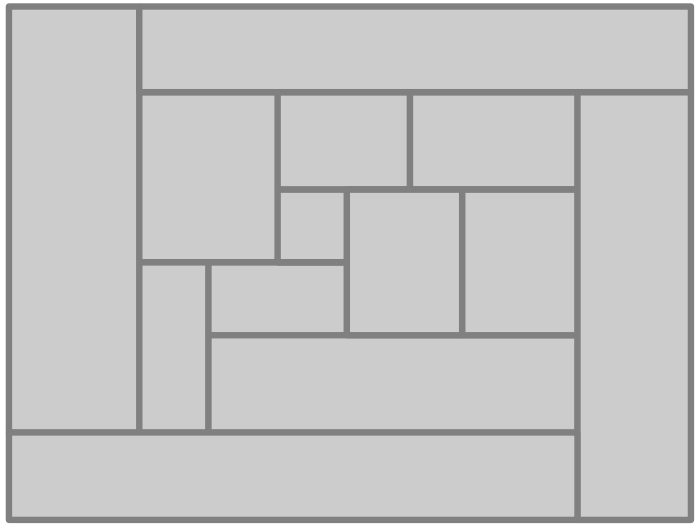
Rectangular dual

Definition.
A **rectangular dual** of a graph G is a contact representation with axis aligned rectangles s.t.
- no four rectangles share a point, and
- the union of all rectangles is a rectangle.

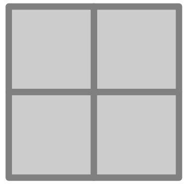
Definition.
A triangle C of G whose removal results in at least two connected components is called a **separating triangle**.



Does not have a rectangular dual. To enclose an area we need at least four rectangles.



When does G admit a rectangular dual?
- G has no separating triangle
- G has at least 4 vertices on outer face; wlog assume this
- each inner face of G must be a triangle



:(

Proper triangular planar graph

Definition.
An internally triangulated, plane graph G without separating triangles and exactly four vertices on the outer face is called **properly triangulated planar (PTP)**.

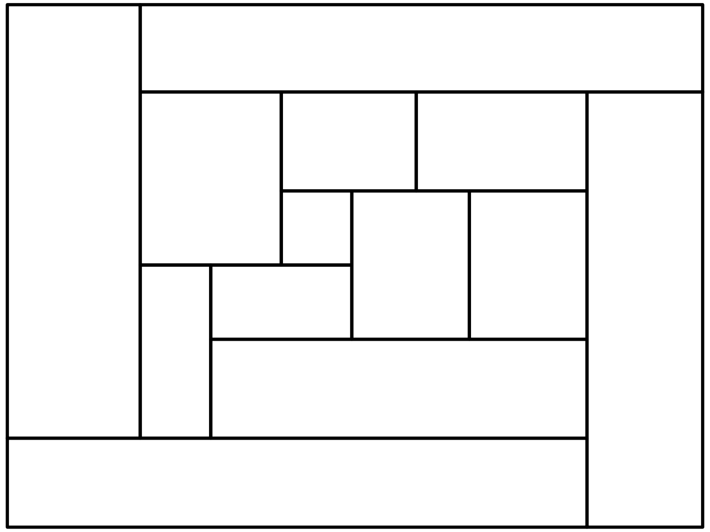
Proper triangular planar graph

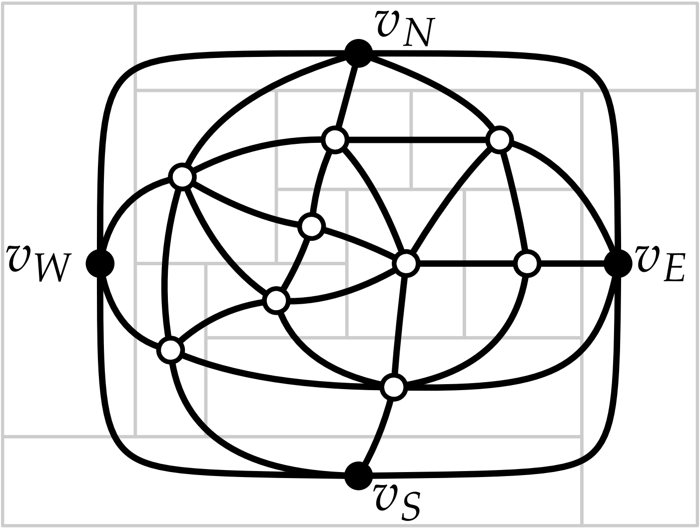
Definition.
An internally triangulated, plane graph G without separating triangles and exactly four vertices on the outer face is called **properly triangulated planar (PTP)**.

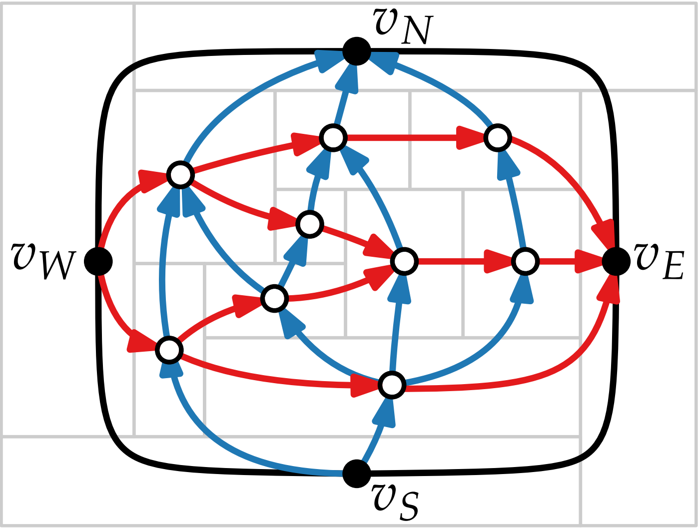
Theorem. [Koźmiński, Kinnen '85]
A graph G has a rectangular dual $\mathcal{R}$ with four rectangles on the boundary of $\mathcal{R}$ if and only if G is a PTP graph.

Regular edge labeling

A rectangular dual gives rise to a 2-coloring and an orientation of the inner edges of G:

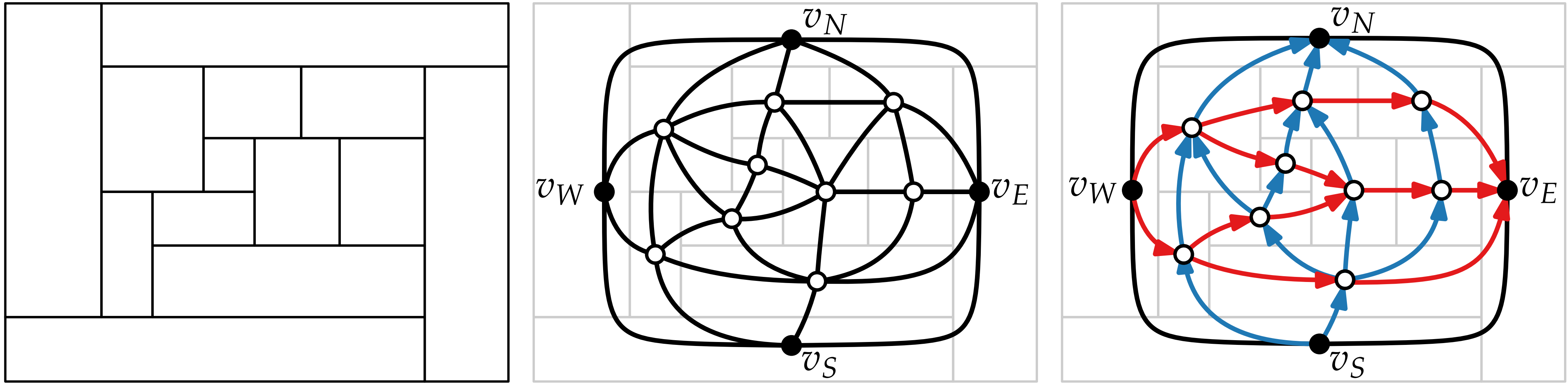






Regular edge labeling

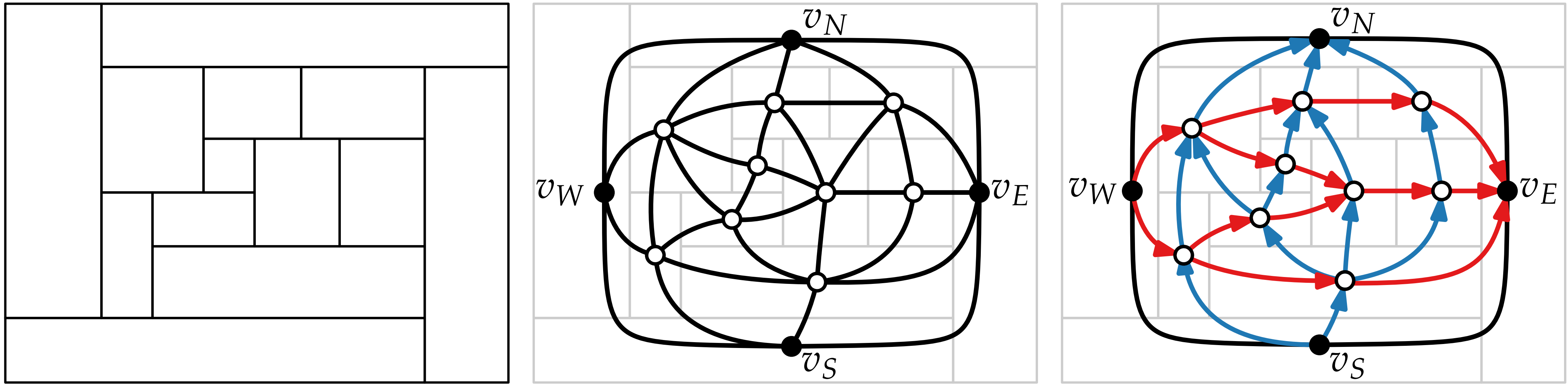
A rectangular dual gives rise to a 2-coloring and an orientation of the inner edges of G:



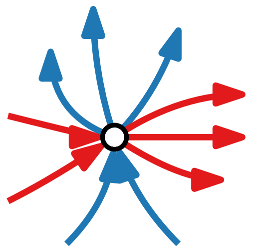
Definition.
A **regular edge labeling (REL)** is a 2-coloring and an orientation of inner edges of G such that

Regular edge labeling

A rectangular dual gives rise to a 2-coloring and an orientation of the inner edges of G:



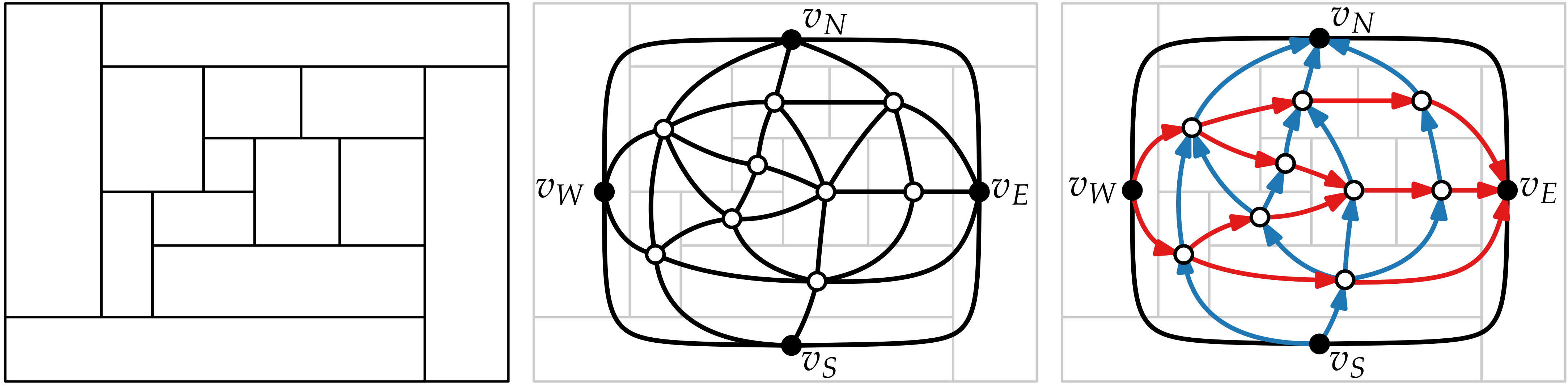
Definition.
A **regular edge labeling (REL)** is a 2-coloring and an orientation of inner edges of G such that



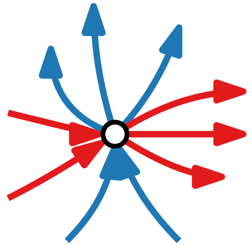
for every inner vertex

Regular edge labeling

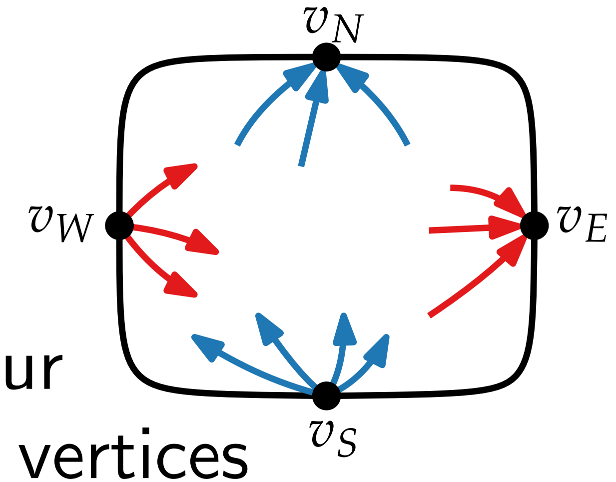
A rectangular dual gives rise to a 2-coloring and an orientation of the inner edges of G:



Definition.
A **regular edge labeling (REL)** is a 2-coloring and an orientation of inner edges of G such that



for every inner vertex

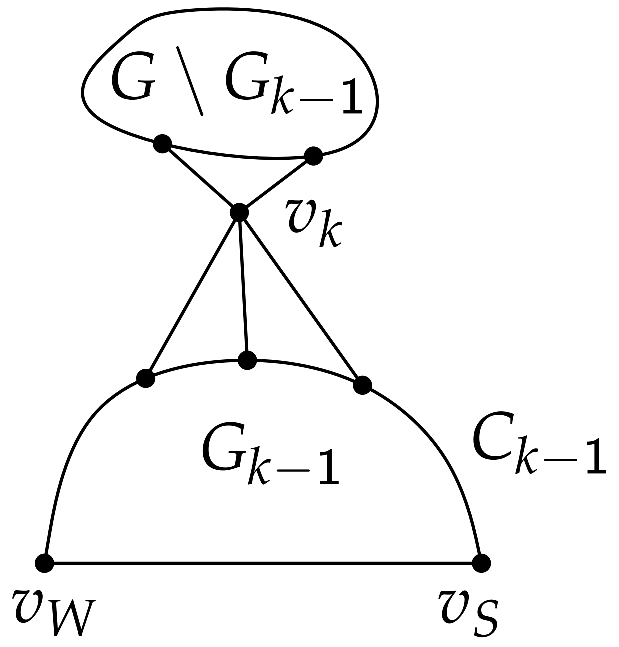


for four outer vertices

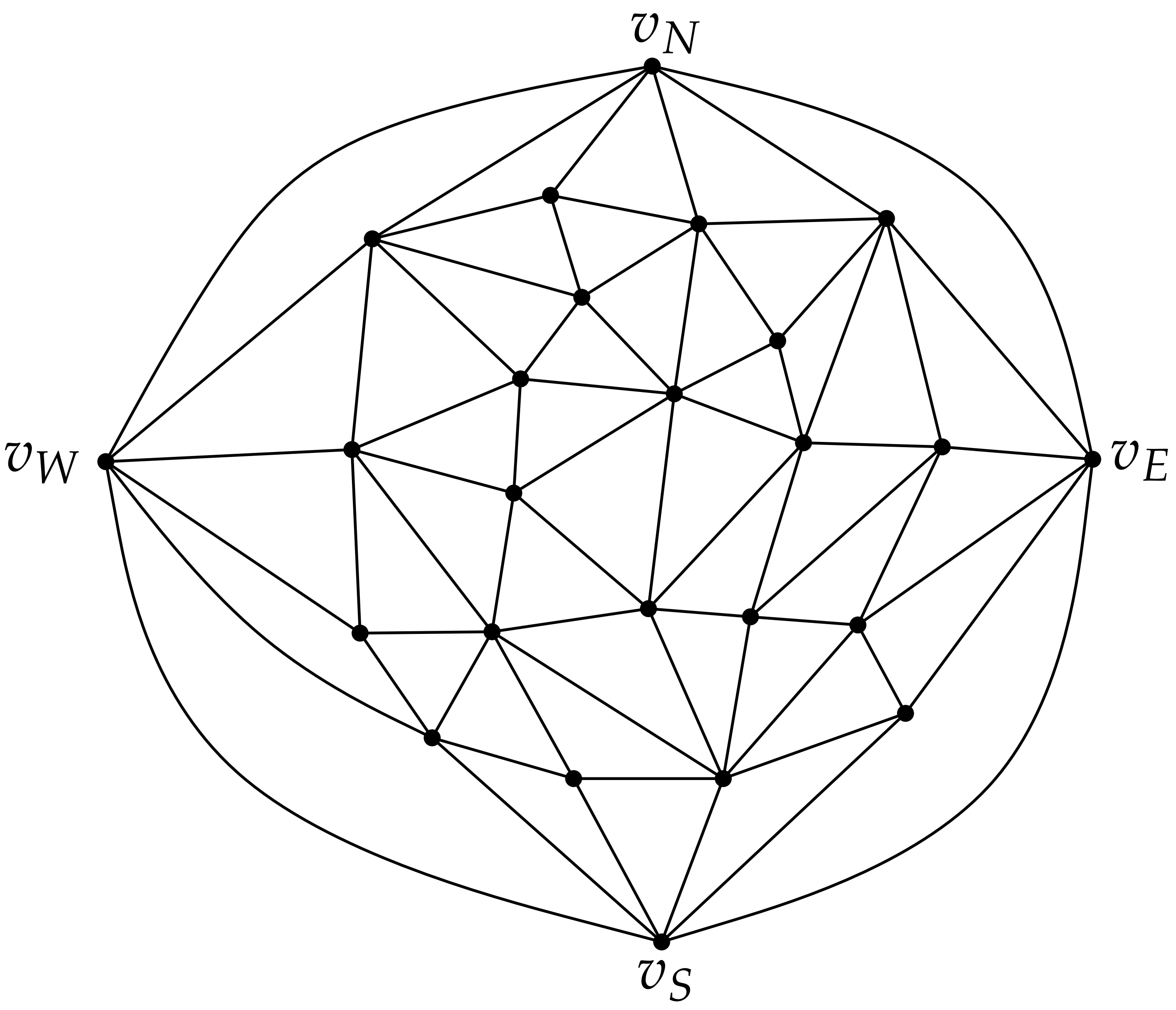
Refined canonical order

Theorem/Definition.
Let G be a PTP graph. There exists a labeling $v_1 = v_S, v_2 = v_W, v_3, \ldots, v_n = v_N$ of the vertices of G such that for every $4 \leq k \leq n$:

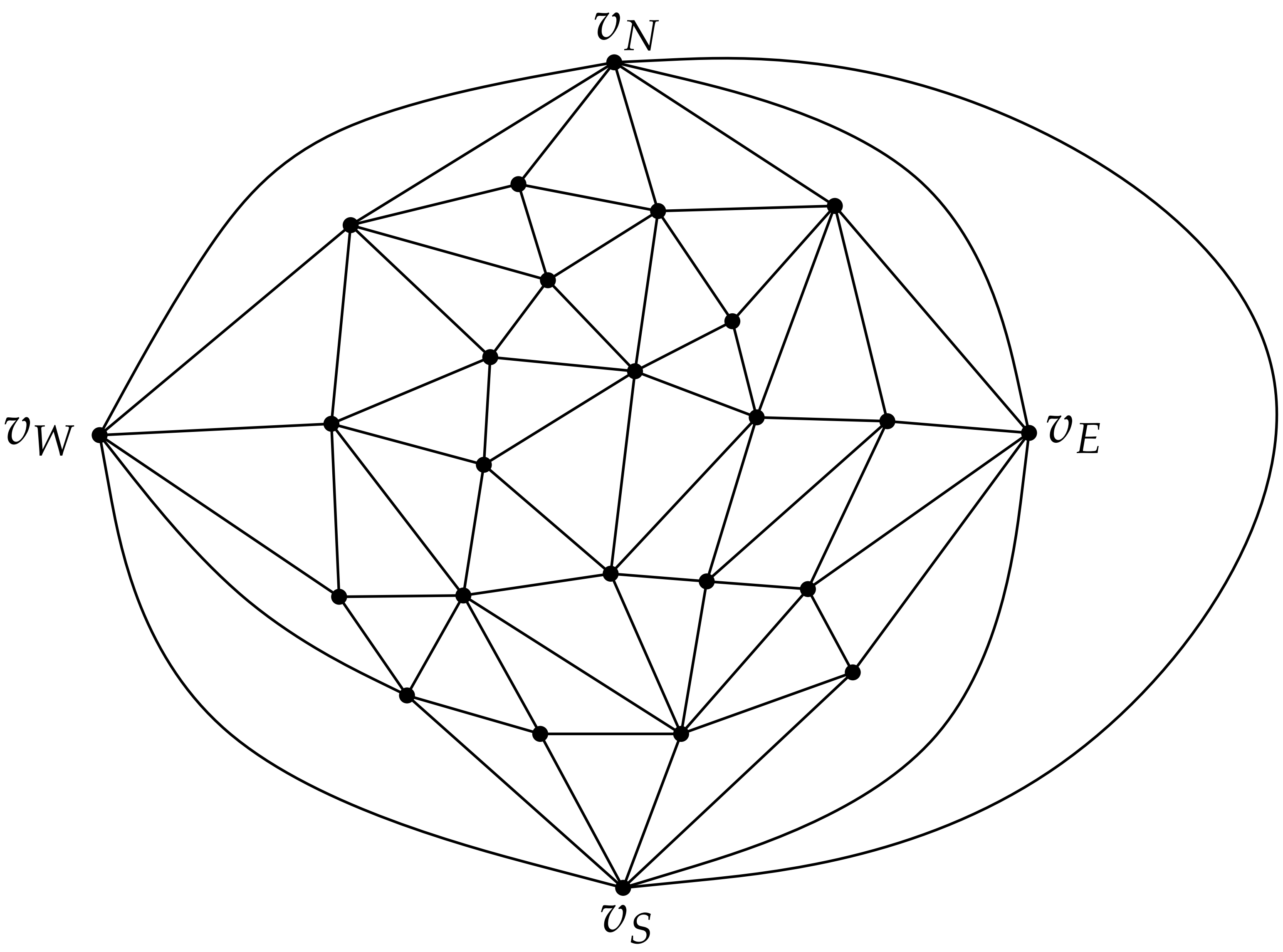
- The subgraph G_{k-1} induced by $v_1, \ldots, v_{k-1}$ is biconnected and boundary C_{k-1} of G_{k-1} contains the edge (v_S, v_W).
- v_k is in exterior face of G_{k-1}, and its neighbors in G_{k-1} form (at least 2-element) subinterval of the path $C_{k-1} \setminus (v_S, v_W)$.
- If $k \leq k-2$, v_k has at least 2 neighbors in $G \setminus G_{k-1}$.



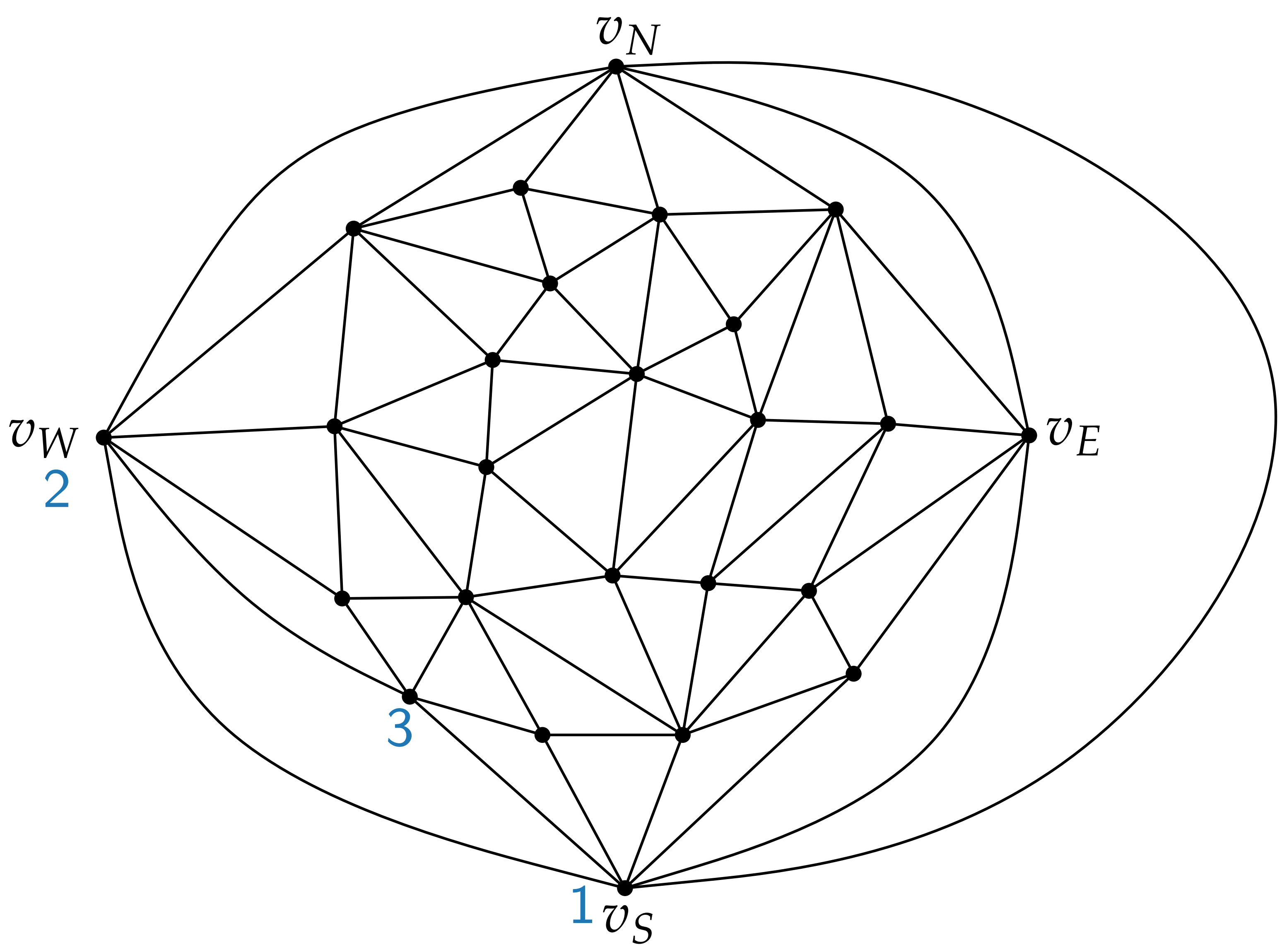
Refined canonical order example



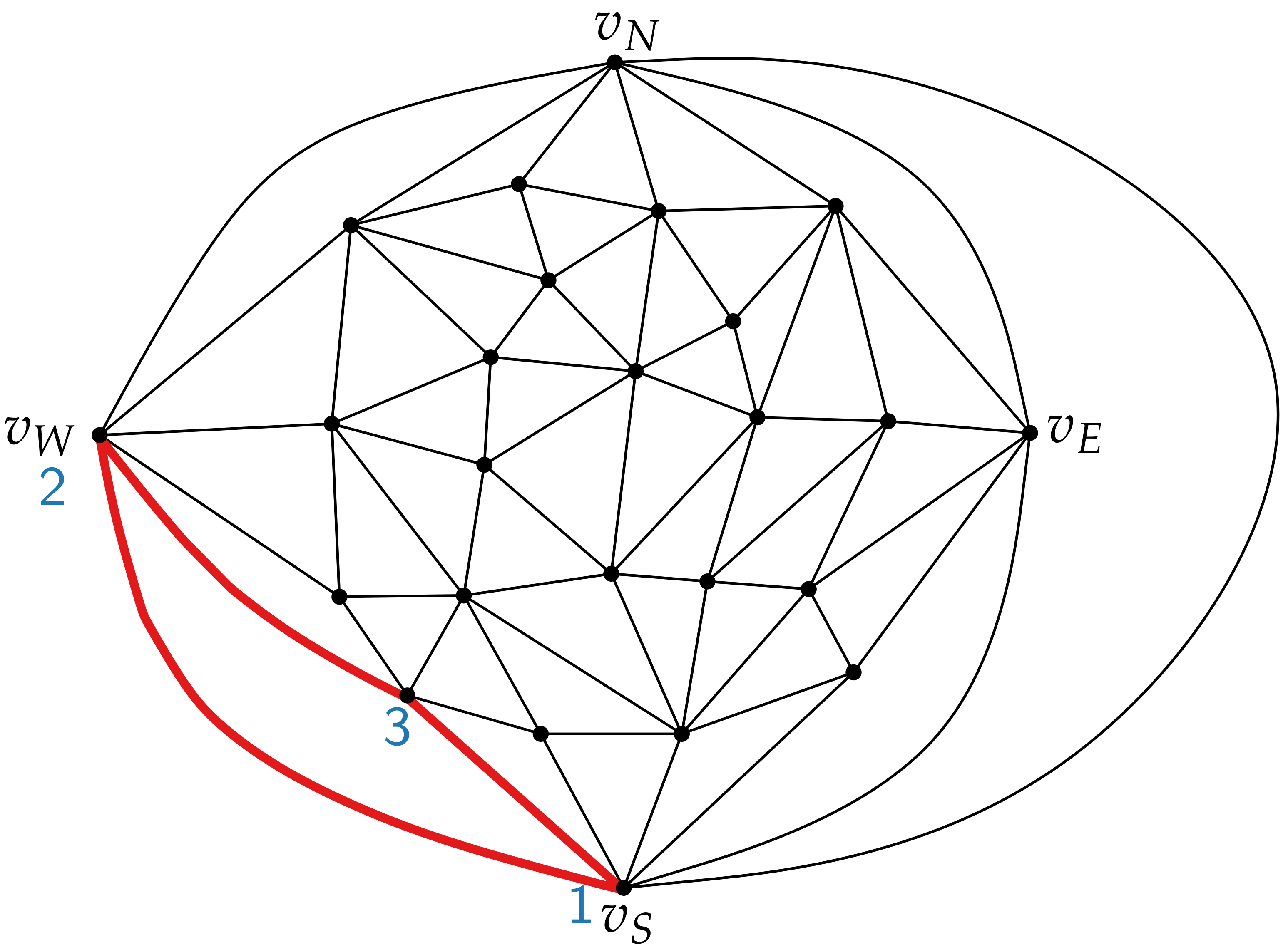
Refined canonical order example



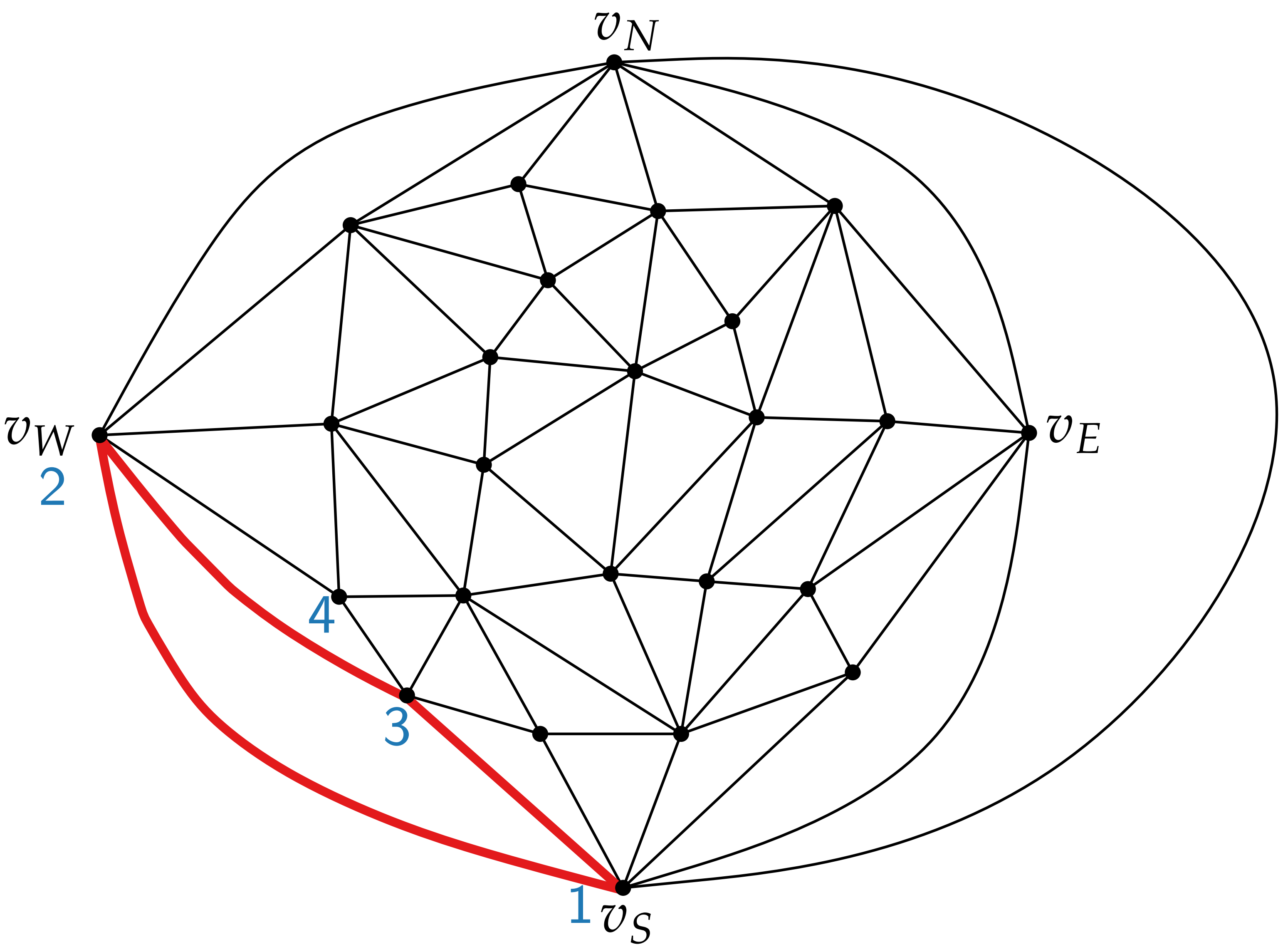
Refined canonical order example



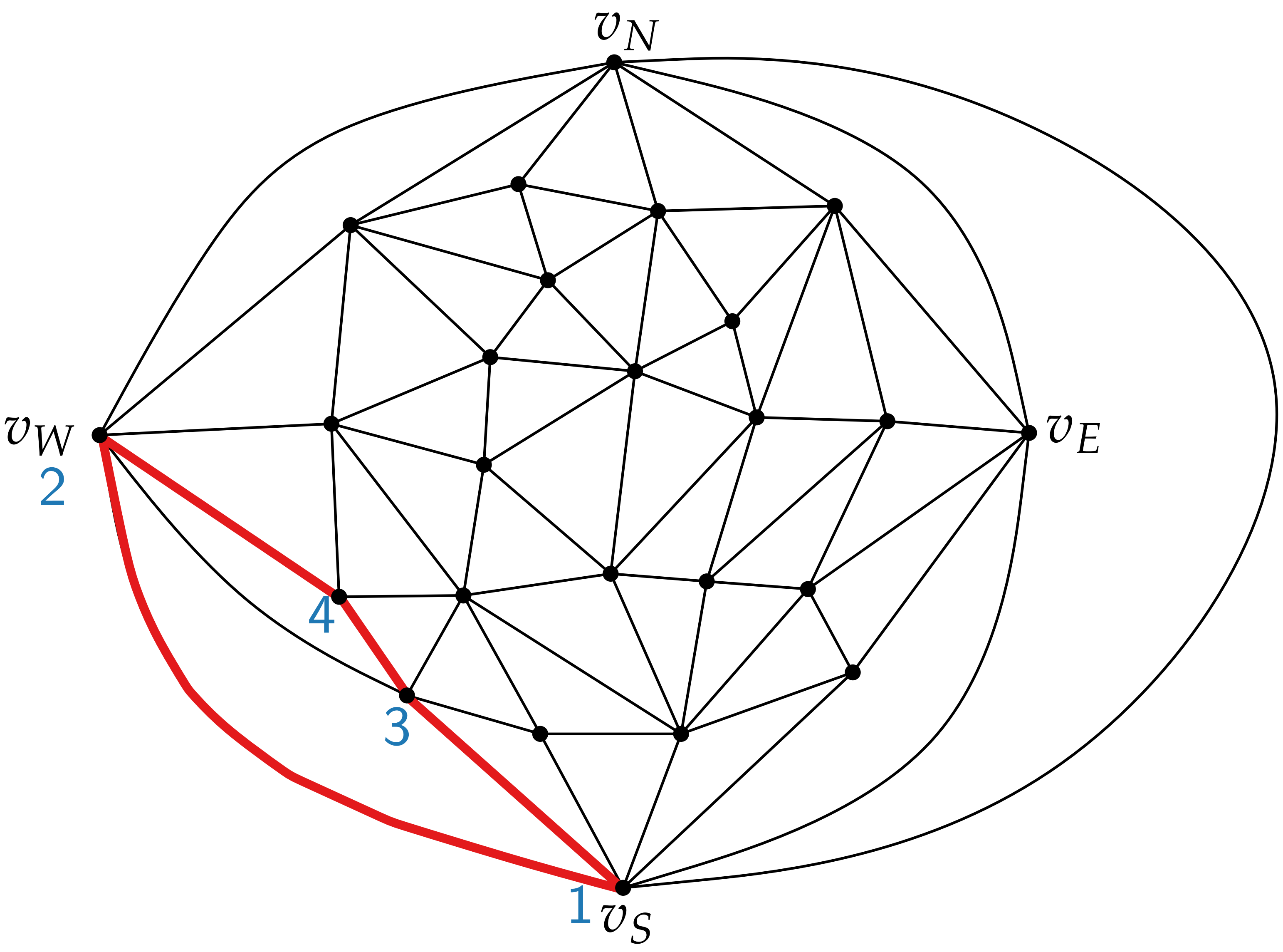
Refined canonical order example



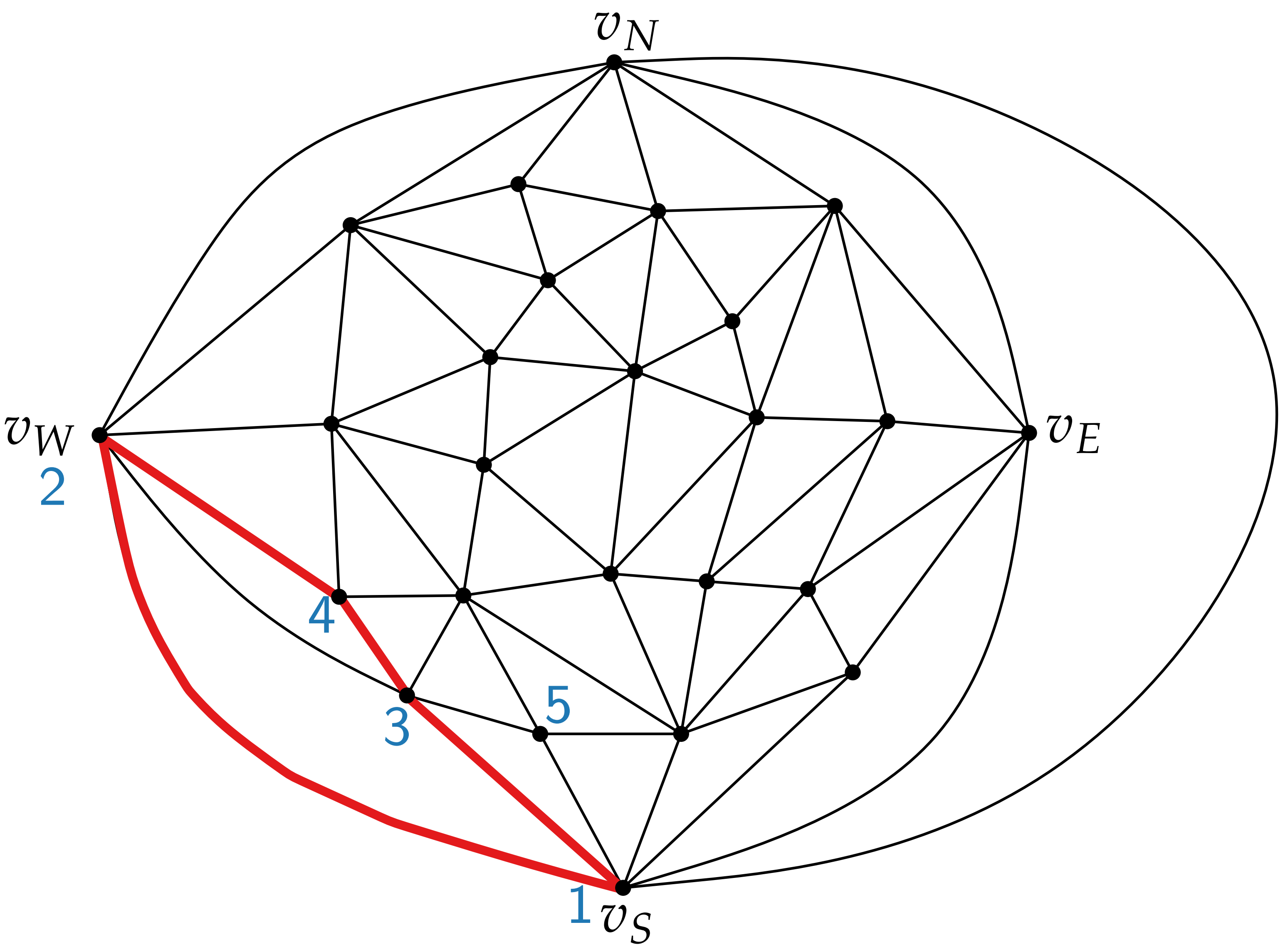
Refined canonical order example



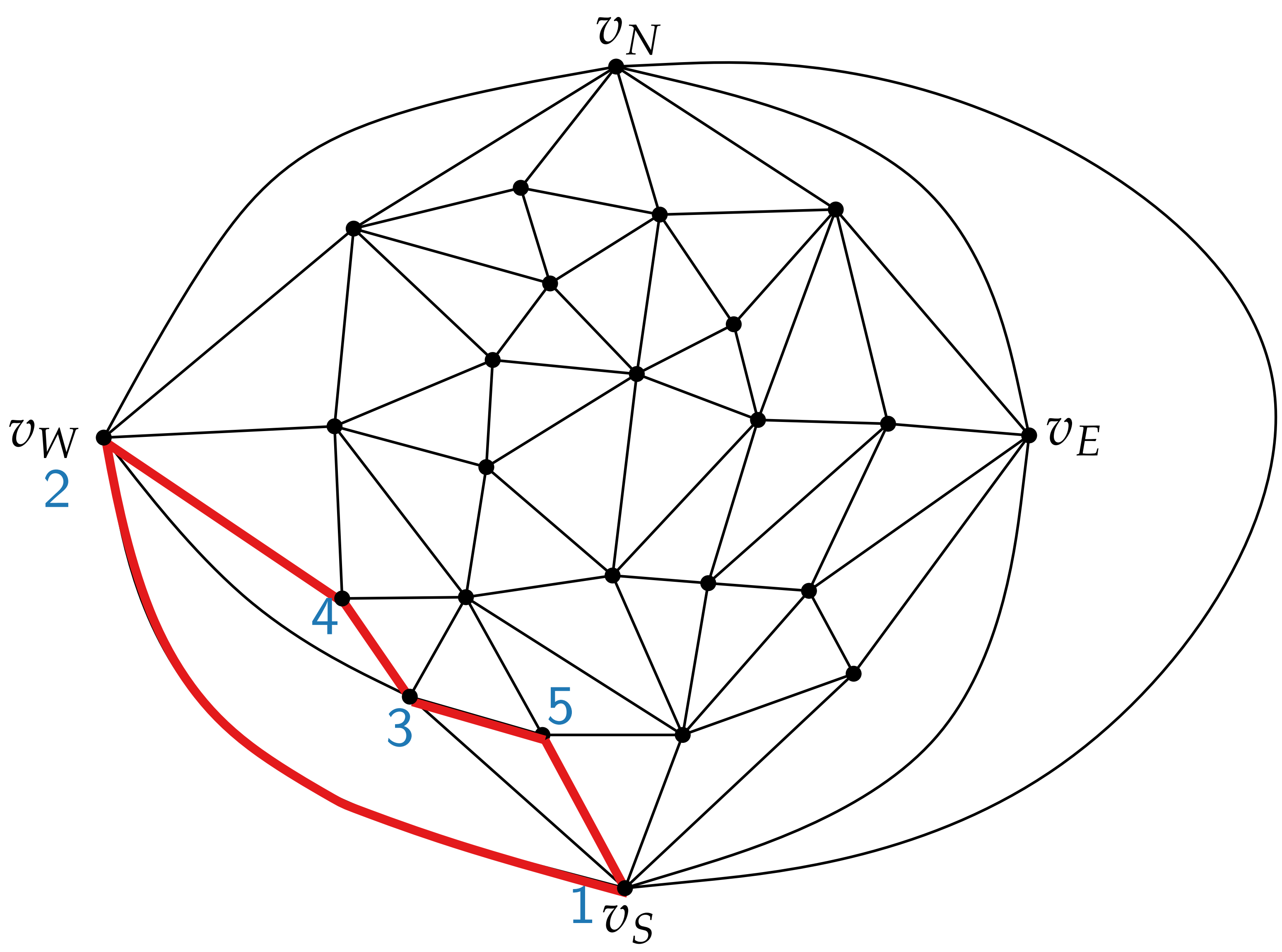
Refined canonical order example



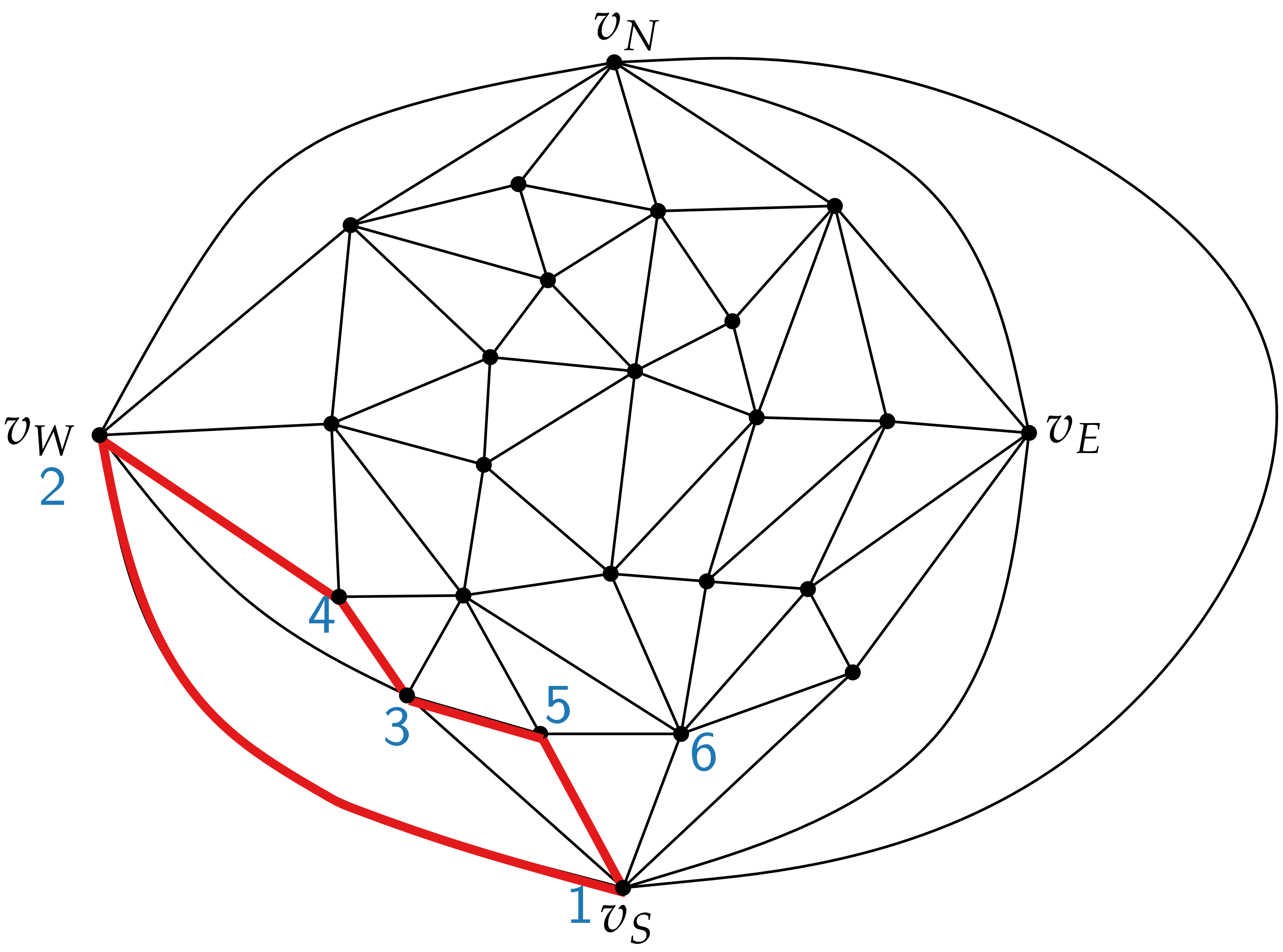
Refined canonical order example



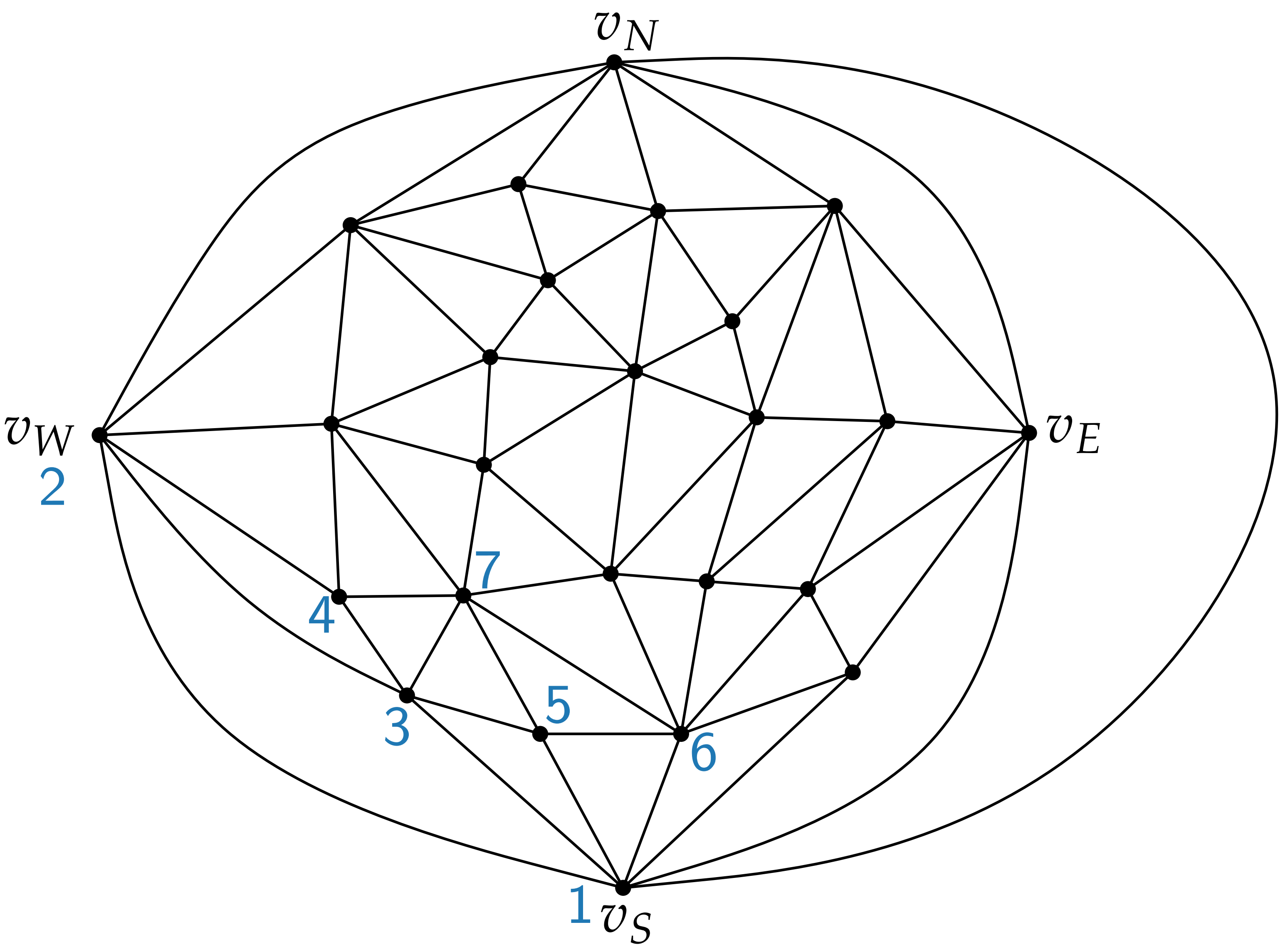
Refined canonical order example



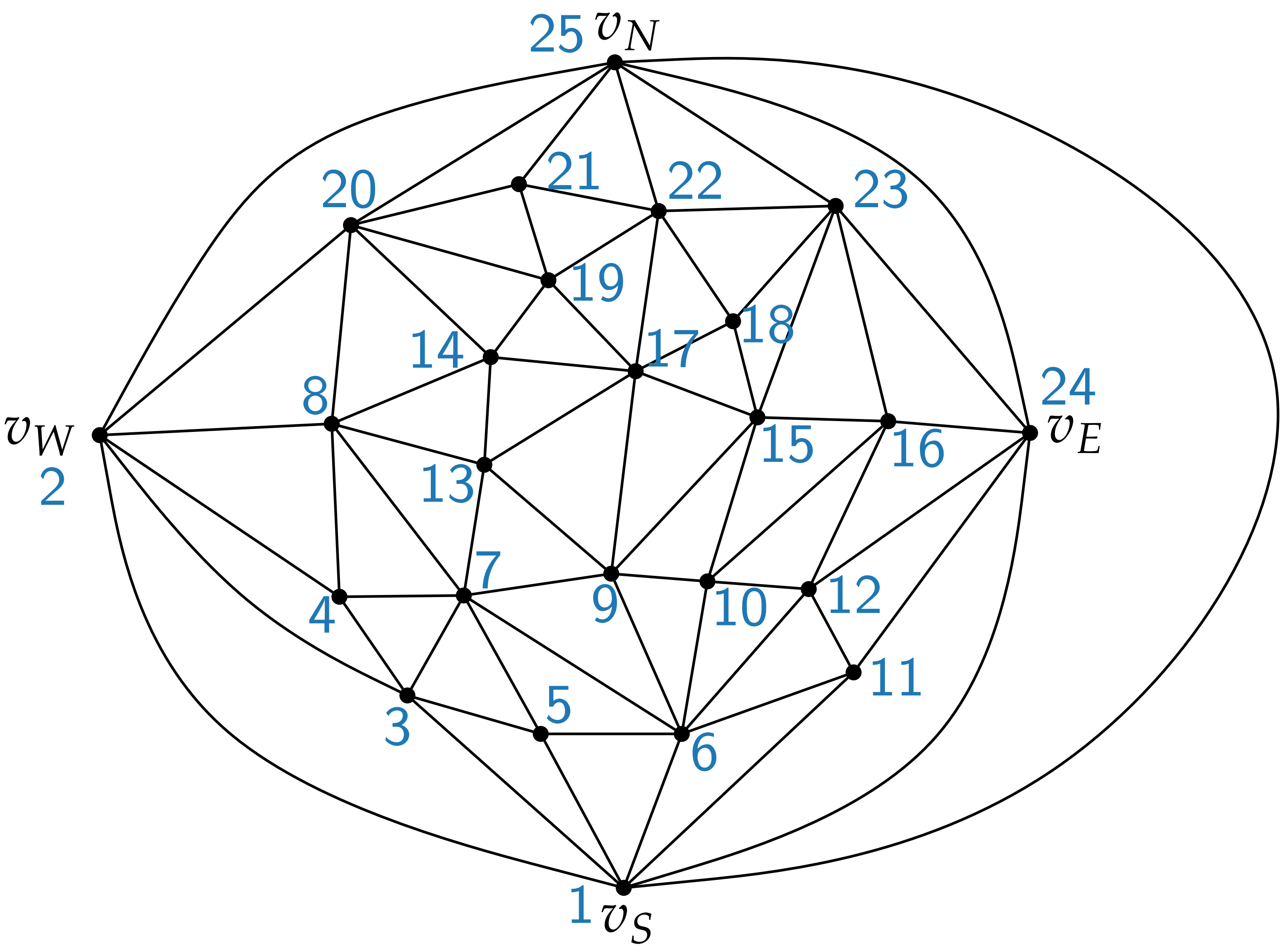
Refined canonical order example



Refined canonical order example



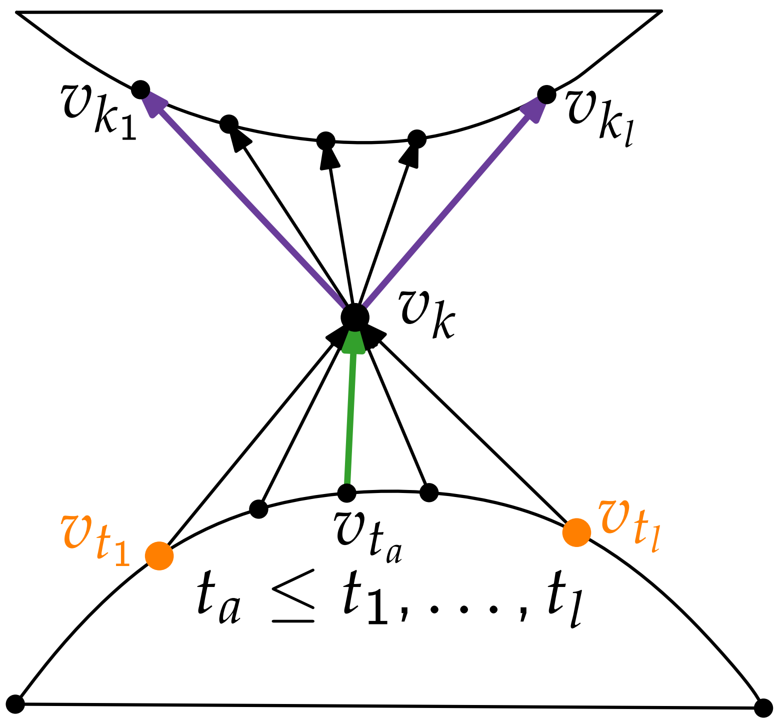
Refined canonical order example



From refined canonical order to REL

Given a refined canonical ordering of G we construct a REL as follows:

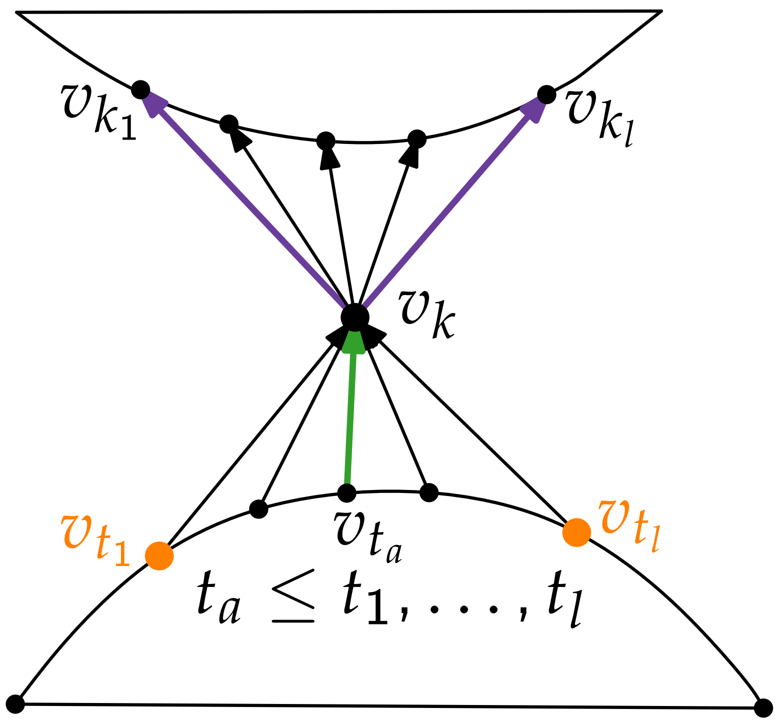
- For $i < j$, orient (v_i, v_j) from v_i to v_j;
- v_k has incoming edges from $v_{t_1}, \ldots, v_{t_l}$, we say that v_{t_1} is **left point** of v_k and v_{t_l} is **right point** of v_k.
- **Base edge** of v_k is (v_{t_a}, v_k), where $t_a < k$ is minimal.
- If $v_{k_1}, \ldots, v_{k_l}$ are higher numbered neighbors of v_k, we call (v_k, v_{k_1}) **left edge** and (v_k, v_{k_l}) **right edge**.



From refined canonical order to REL

Given a refined canonical ordering of G we construct a REL as follows:

- For $i < j$, orient (v_i, v_j) from v_i to v_j;
- v_k has incoming edges from $v_{t_1}, \ldots, v_{t_l}$, we say that v_{t_1} is **left point** of v_k and v_{t_l} is **right point** of v_k.
- **Base edge** of v_k is (v_{t_a}, v_k), where $t_a < k$ is minimal.
- If $v_{k_1}, \ldots, v_{k_l}$ are higher numbered neighbors of v_k, we call (v_k, v_{k_1}) **left edge** and (v_k, v_{k_l}) **right edge**.

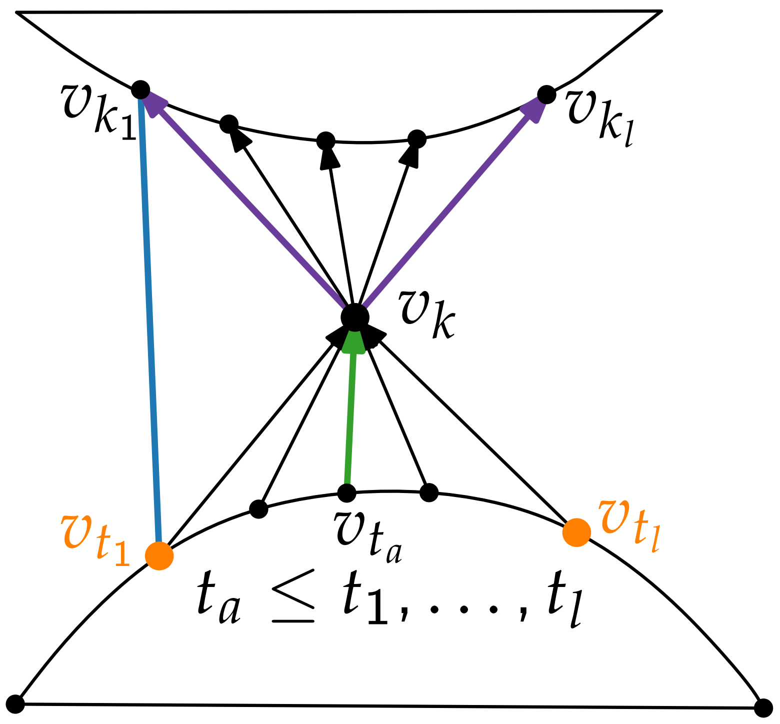


Lemma 1.
Left edge or right edge cannot be a base edge.

From refined canonical order to REL

Given a refined canonical ordering of G we construct a REL as follows:

- For $i < j$, orient (v_i, v_j) from v_i to v_j;
- v_k has incoming edges from $v_{t_1}, \ldots, v_{t_l}$, we say that v_{t_1} is **left point** of v_k and v_{t_l} is **right point** of v_k.
- **Base edge** of v_k is (v_{t_a}, v_k), where $t_a < k$ is minimal.
- If $v_{k_1}, \ldots, v_{k_l}$ are higher numbered neighbors of v_k, we call (v_k, v_{k_1}) **left edge** and (v_k, v_{k_l}) **right edge**.

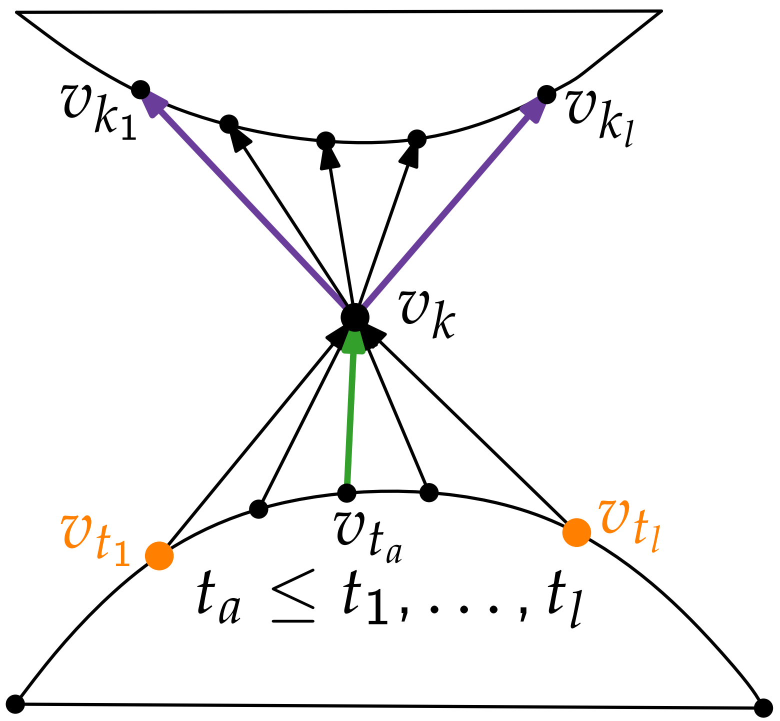


Lemma 1.
Left edge or right edge cannot be a base edge.

Proof. Suppose left edge (v_k, v_{k_1}) is base edge of v_{k_1}. Since G triangulated, $(v_{t_1}, v_{k_1}) \in E(G)$. Contradiction since $v_k > v_{t_1}$.

From refined canonical order to REL

Lemma 2.
An edge is either a left edge, a right edge or a base edge.

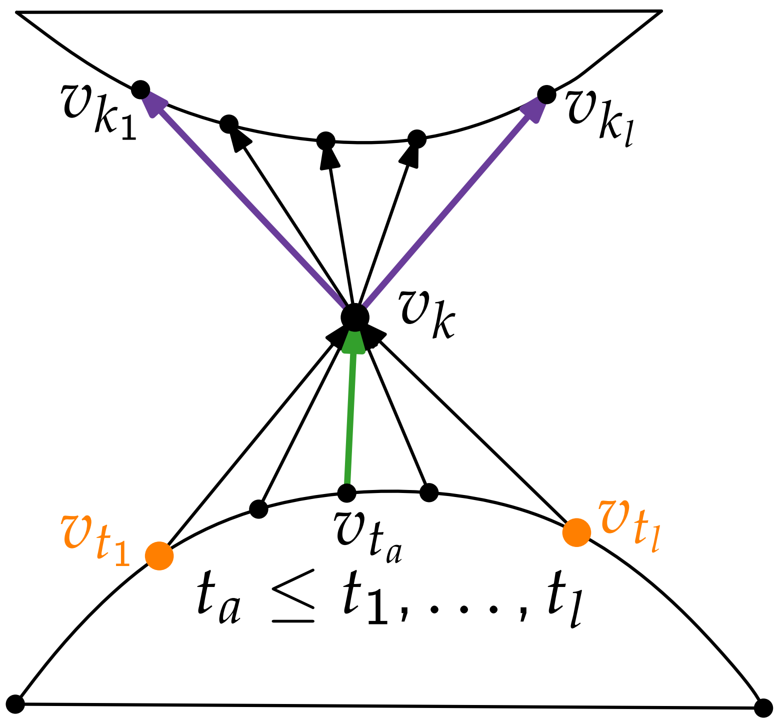


From refined canonical order to REL

Lemma 2.
An edge is either a left edge, a right edge or a base edge.

Proof.

- Exclusive "or" follows from Lemma 1.

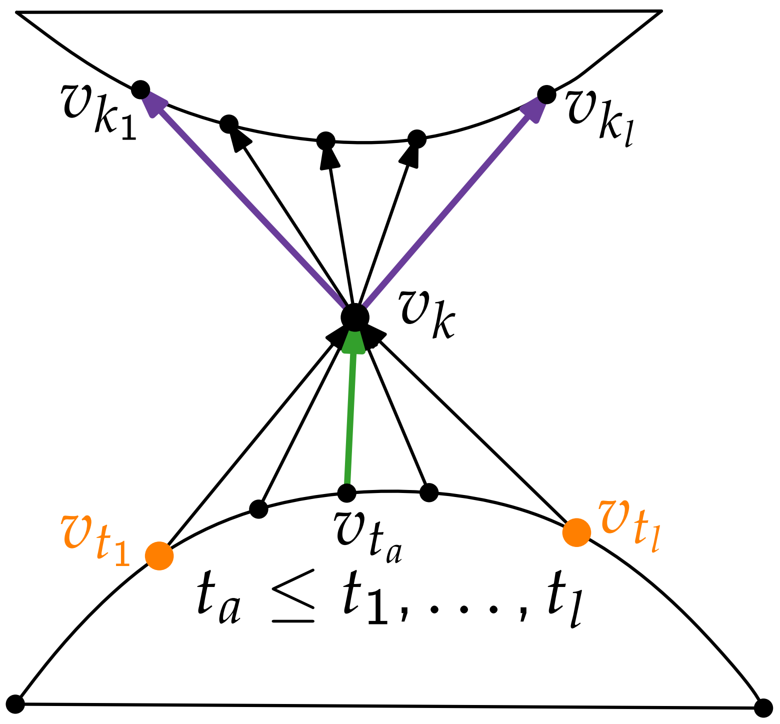


From refined canonical order to REL

Lemma 2.
An edge is either a left edge, a right edge or a base edge.

Proof.

- Exclusive "or" follows from Lemma 1.
- Let (v_{t_a}, v_k) be base edge of v_k.
- v_{t_a} is right point of $v_{t_{a-1}}$; v_{t_i} is right point of $v_{t_{i-1}}$:
  - v_{t_i} has at least two higher-numbered neighbors.
  - One of them is v_k; the other one is either $v_{t_{i-1}}$ or $v_{t_{i+1}}$.
  - For $1 \leq i < a-1$, it is $v_{t_{i-1}}$.

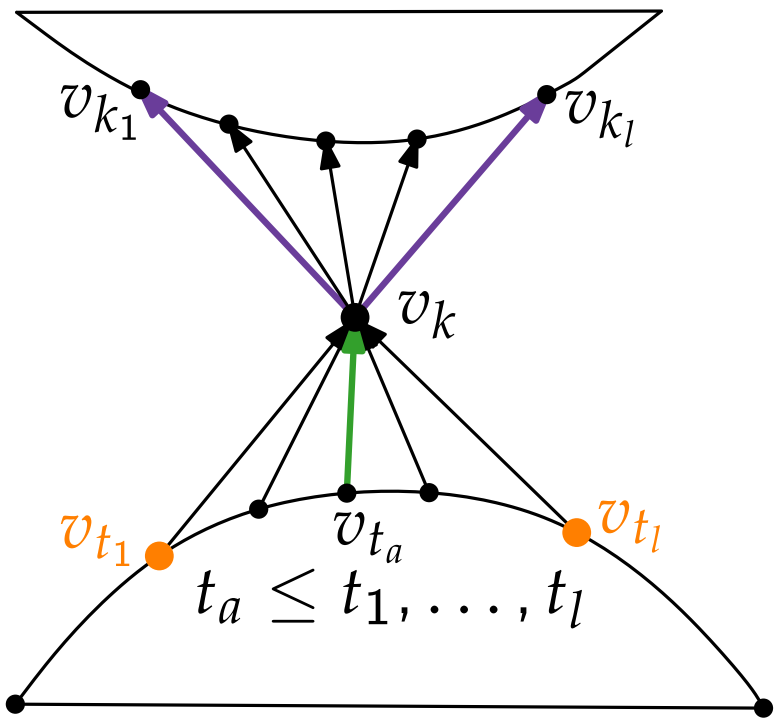


From refined canonical order to REL

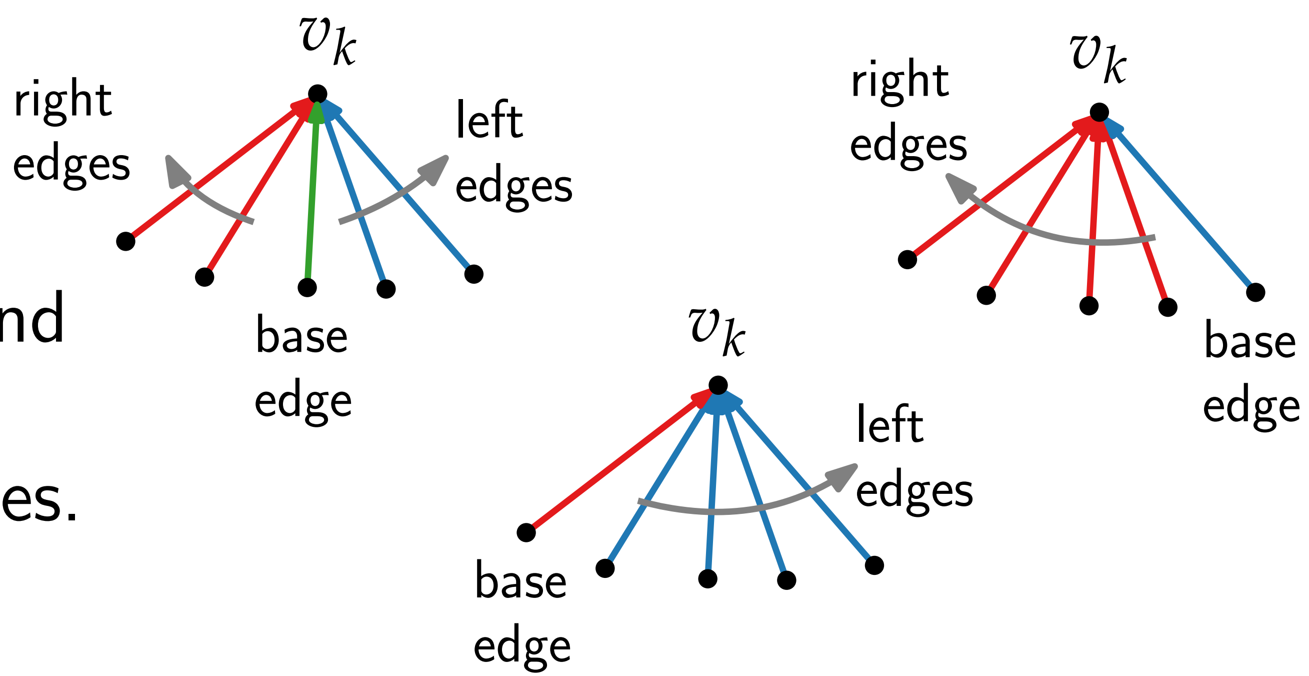
Lemma 2.
An edge is either a left edge, a right edge or a base edge.

Proof.

- Exclusive "or" follows from Lemma 1.
- Let (v_{t_a}, v_k) be base edge of v_k.
- v_{t_a} is right point of $v_{t_{a-1}}$; v_{t_i} is right point of $v_{t_{i-1}}$:
  - v_{t_i} has at least two higher-numbered neighbors.
  - One of them is v_k; the other one is either $v_{t_{i-1}}$ or $v_{t_{i+1}}$.
  - For $1 \leq i < a-1$, it is $v_{t_{i-1}}$.
- Edges (v_{t_i}, v_k), $1 \leq i < a-1$, are right edges.
- Similarly, (v_{t_i}, v_k), for $a+1 \leq i \leq l$, are left edges.



From refined canonical order to REL



Coloring.

- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

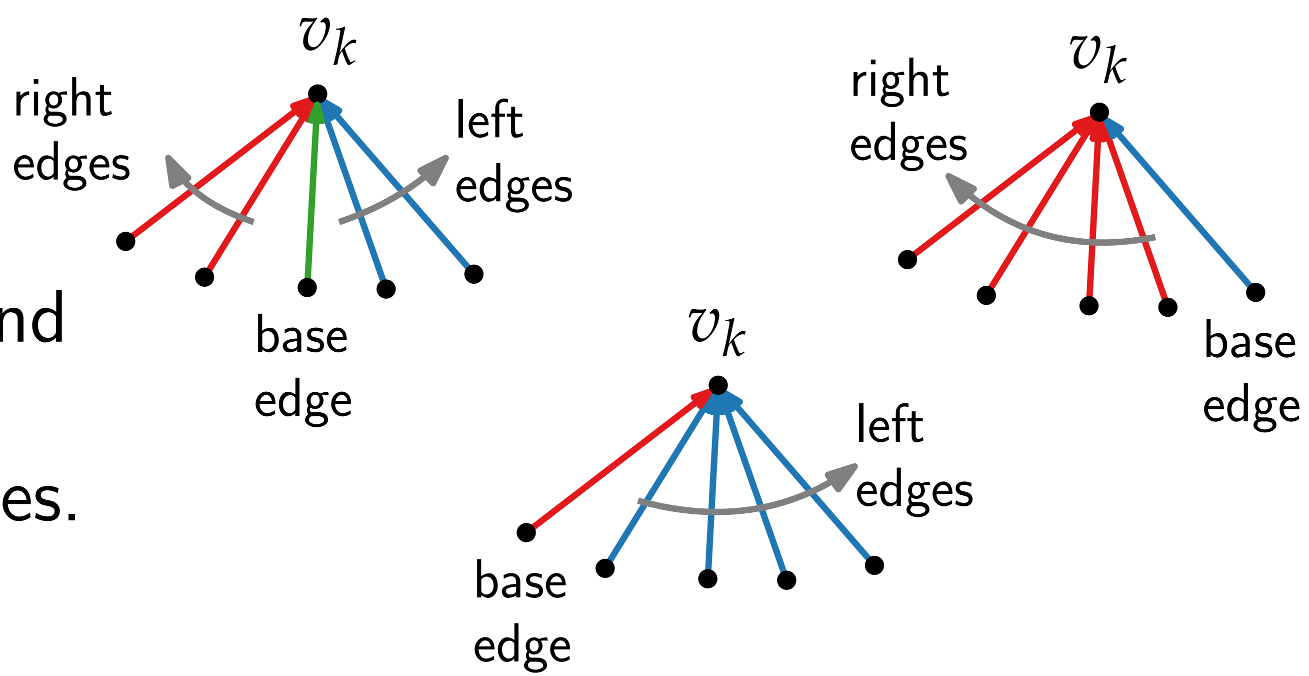
Let T_r be the red edges and T_b the blue edges.

From refined canonical order to REL

Coloring.

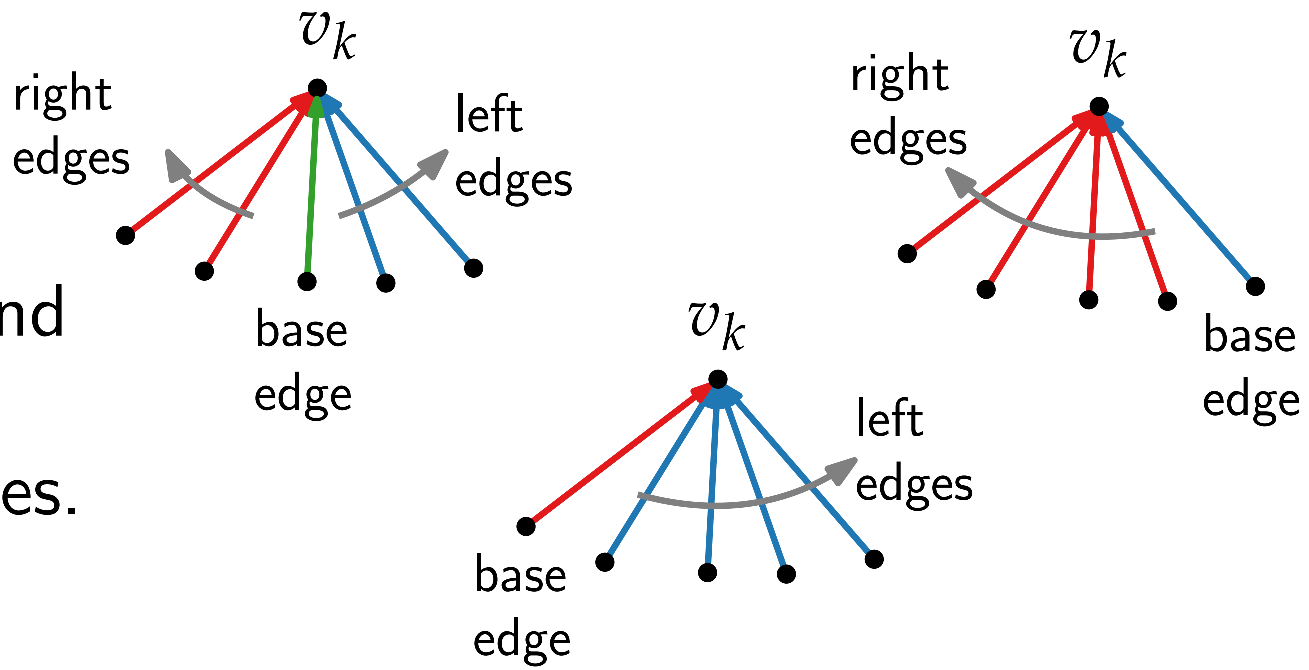
- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

Let T_r be the red edges and T_b the blue edges.



Lemma 3.
$\{T_r, T_b\}$ is a regular edge labeling.

From refined canonical order to REL



Coloring.

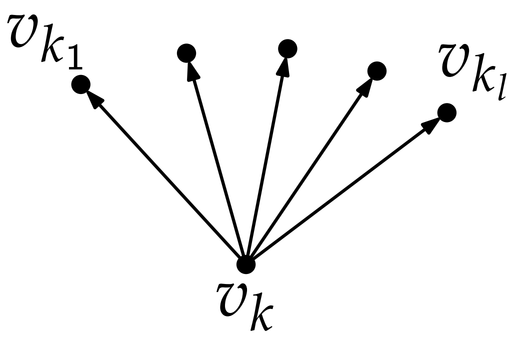
- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

Let T_r be the red edges and T_b the blue edges.

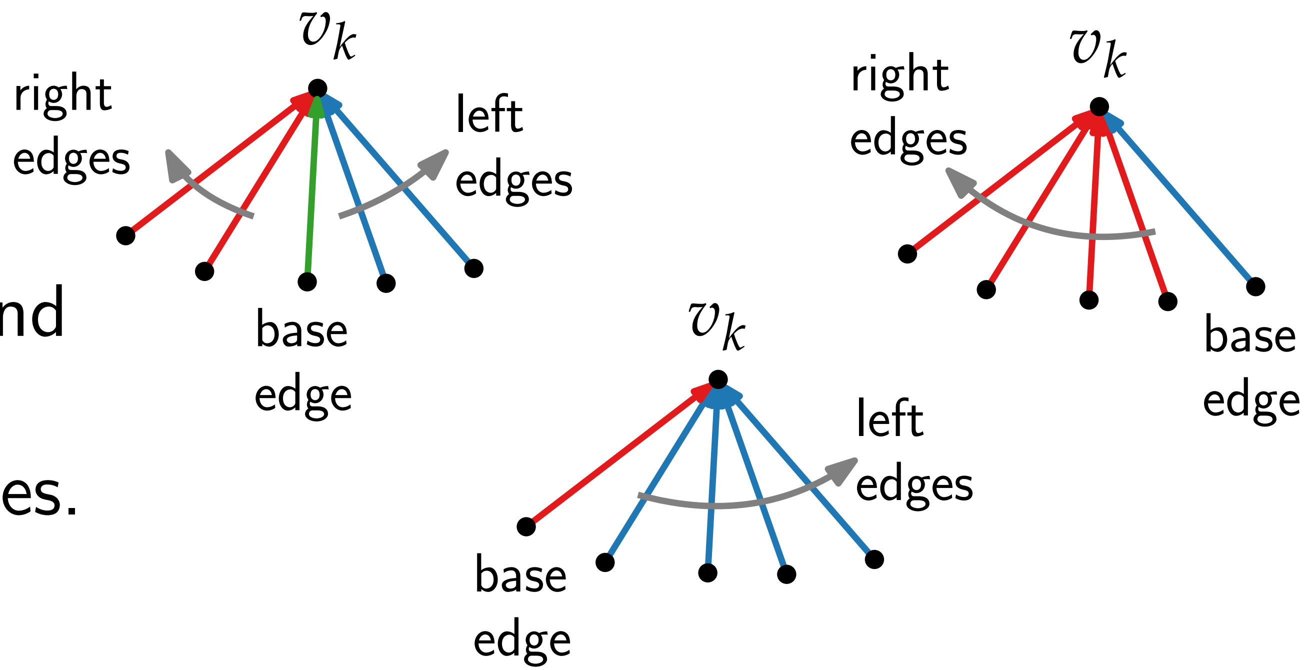
Lemma 3.
$\{T_r, T_b\}$ is a regular edge labeling.

Proof.

$k_l \geq 2$



From refined canonical order to REL



Coloring.

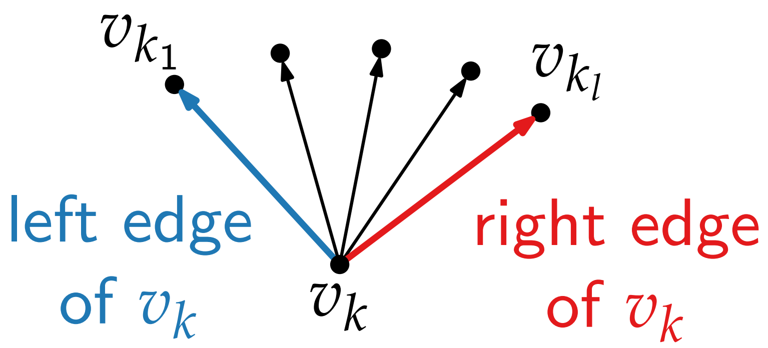
- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

Let T_r be the red edges and T_b the blue edges.

Lemma 3.
$\{T_r, T_b\}$ is a regular edge labeling.

Proof.

$k_l \geq 2$

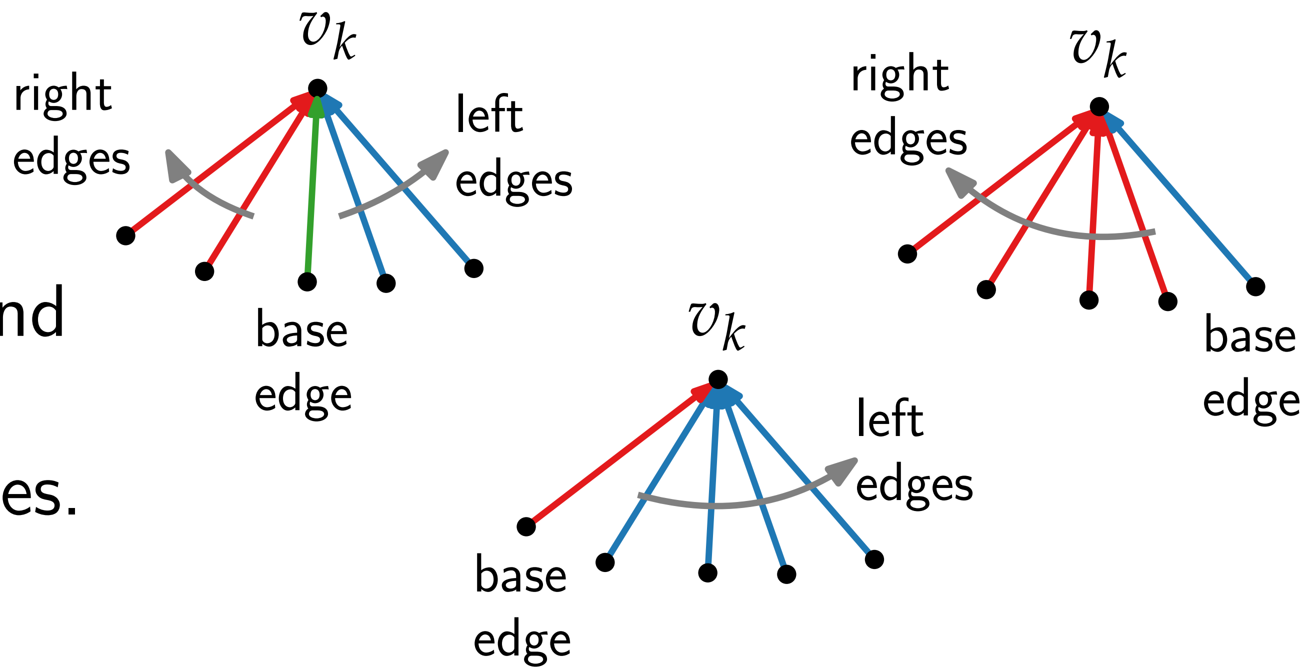


From refined canonical order to REL

Coloring.

- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

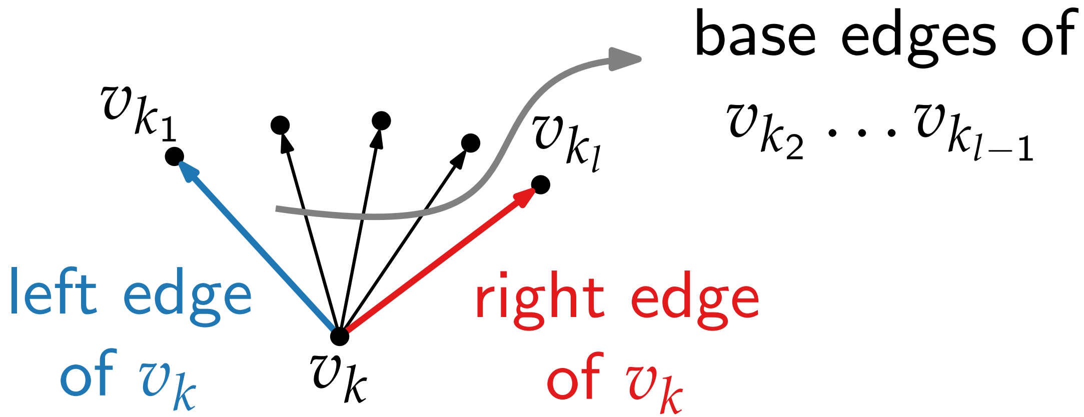
Let T_r be the red edges and T_b the blue edges.



Lemma 3.
$\{T_r, T_b\}$ is a regular edge labeling.

Proof.

$k_l \geq 2$

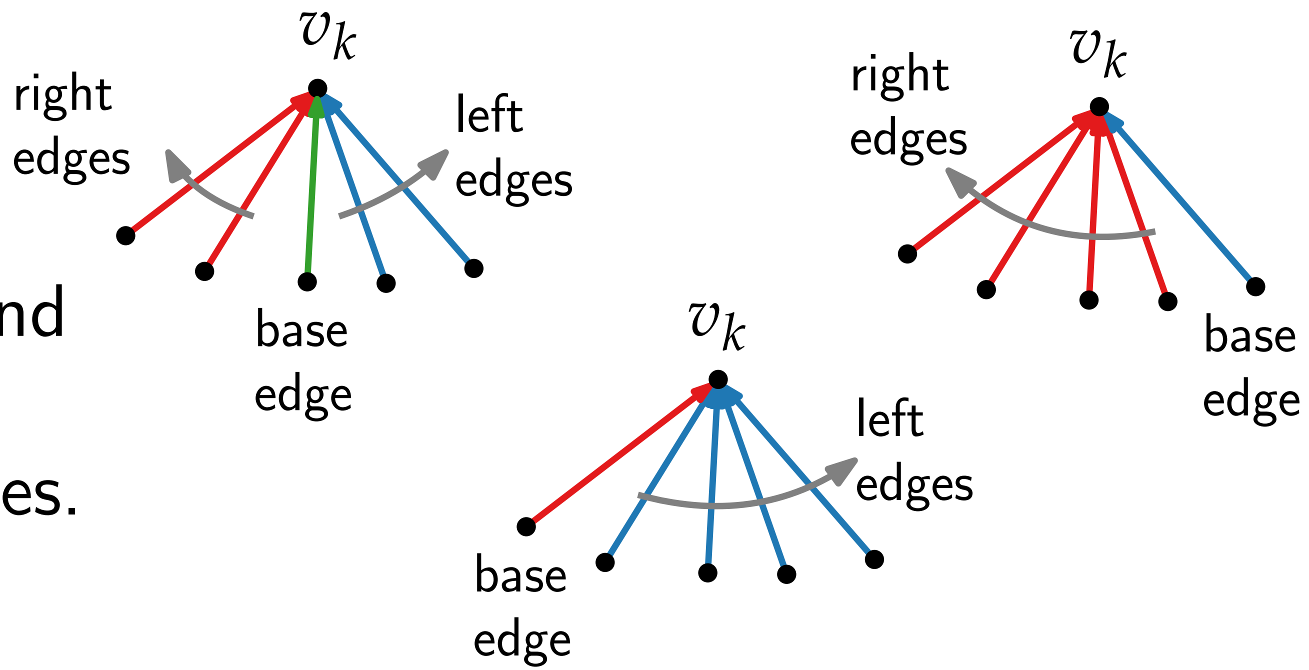


From refined canonical order to REL

Coloring.

- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

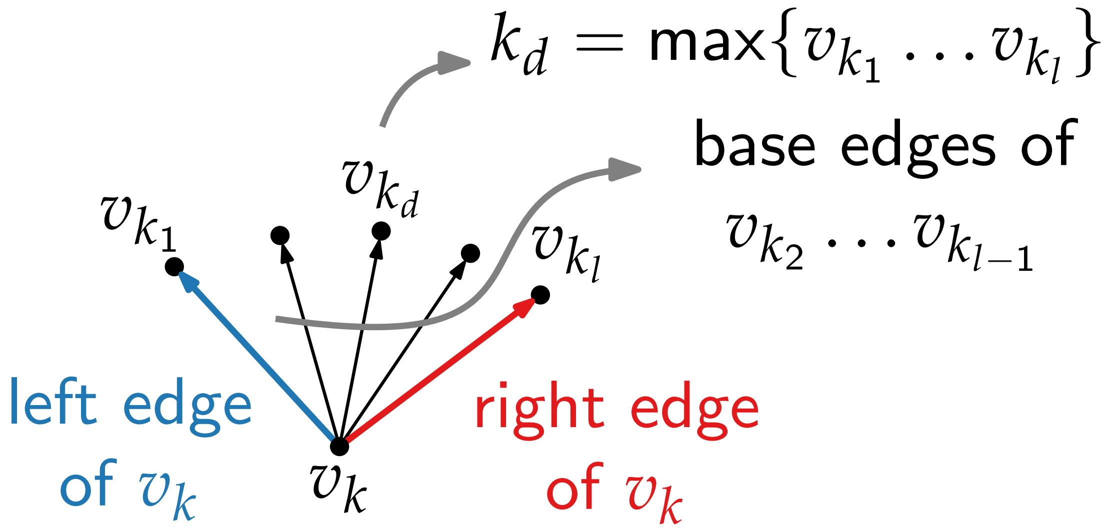
Let T_r be the red edges and T_b the blue edges.



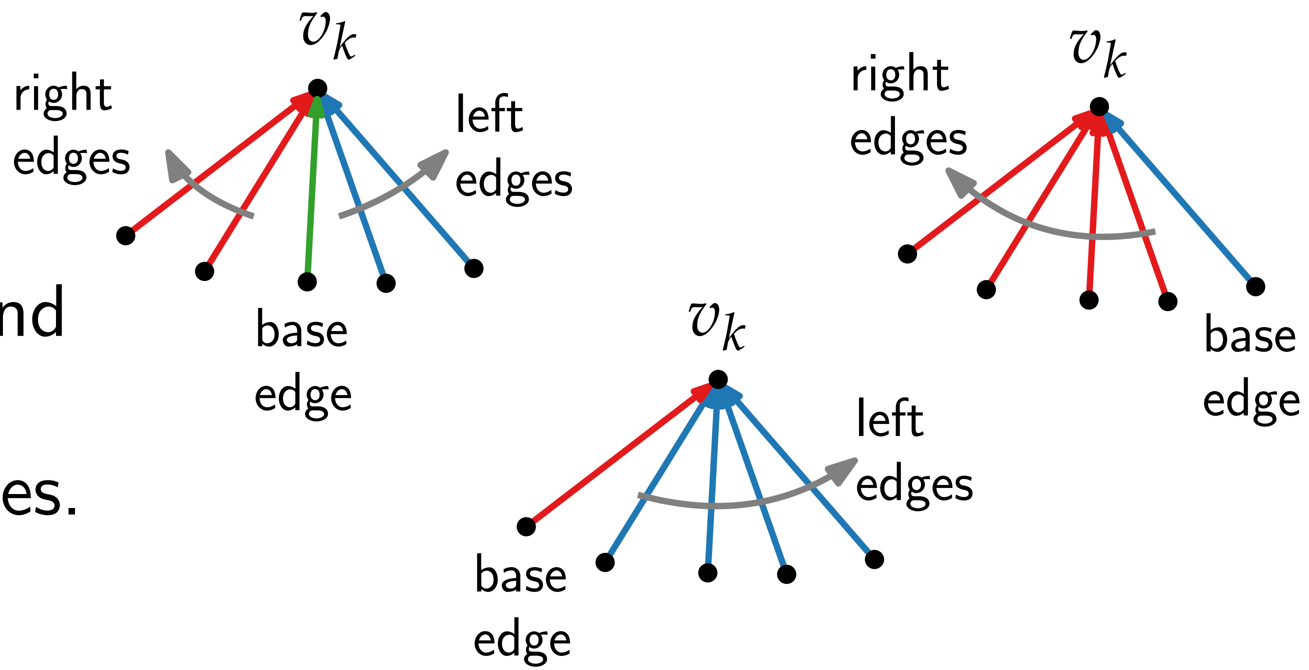
> **Lemma 3.**
> $\{T_r, T_b\}$ is a regular edge labeling.

Proof.

$k_l \geq 2$



From refined canonical order to REL



Coloring.

- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

Let T_r be the red edges and T_b the blue edges.

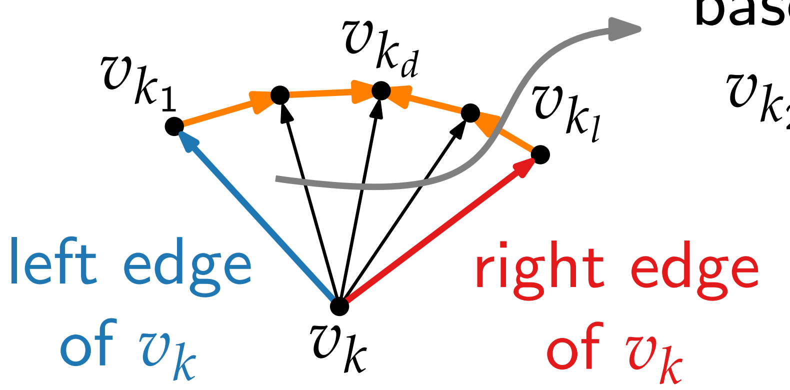
> **Lemma 3.**
> $\{T_r, T_b\}$ is a regular edge labeling.

Proof.

$k_l \geq 2$

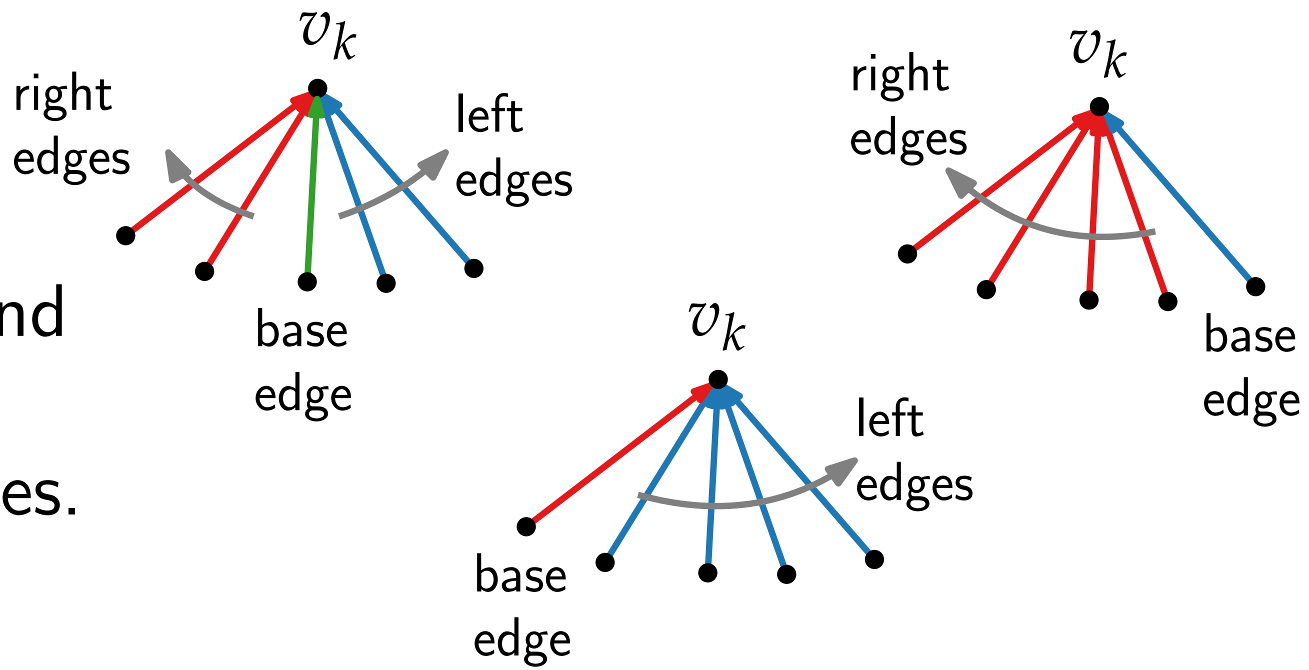
$k_d = \max\{v_{k_1} \dots v_{k_l}\}$

base edges of $v_{k_2} \dots v_{k_{l-1}}$



- $k_1 < k_2 < \dots < k_d$ and $k_d > k_{d+1} > \dots > k_l$

From refined canonical order to REL



Coloring.

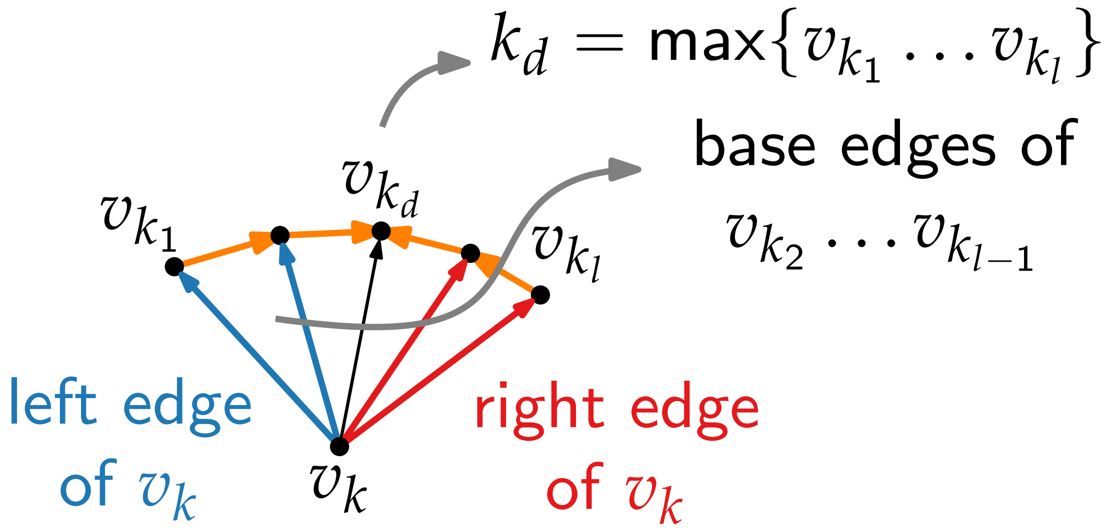
- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

Let T_r be the red edges and T_b the blue edges.

> **Lemma 3.**
> $\{T_r, T_b\}$ is a regular edge labeling.

Proof.

$k_l \geq 2$



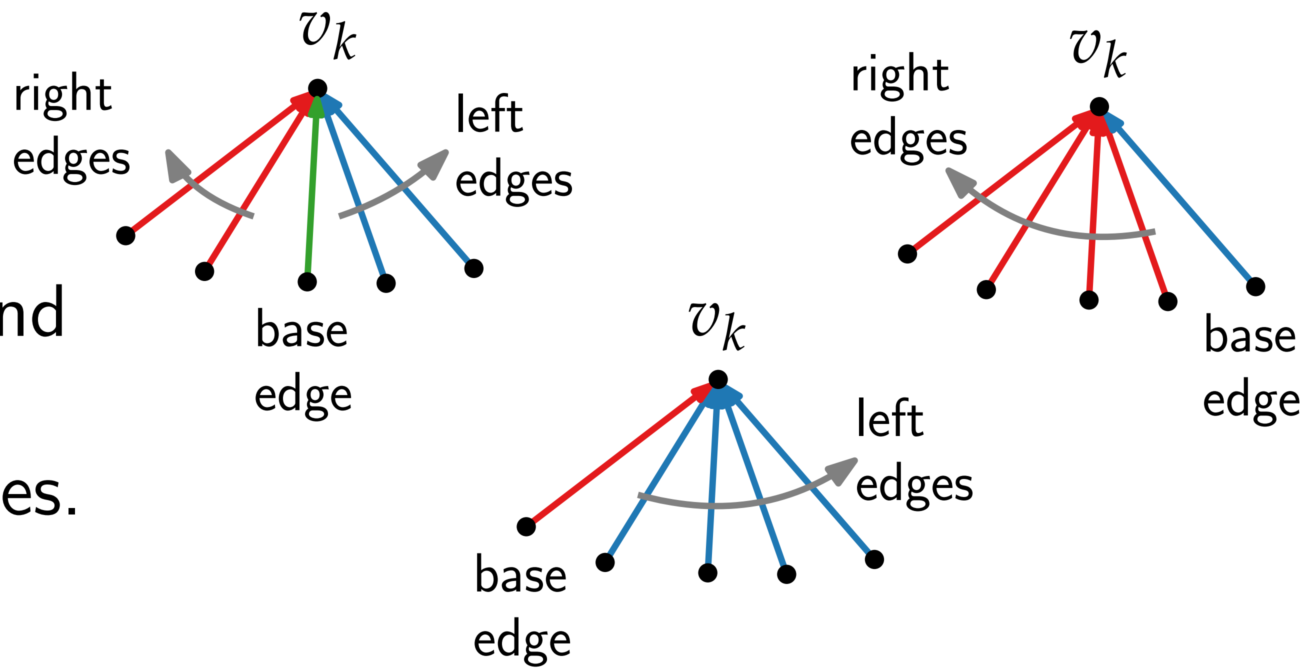
- $k_1 < k_2 < \dots < k_d$ and $k_d > k_{d+1} > \dots > k_l$
- (v_k, v_{k_i}), $2 \leq i \leq d-1$ are red
- (v_k, v_{k_i}), $d+1 \leq i \leq l-1$ are blue
- (v_k, v_{k_d}) is either red or blue

From refined canonical order to REL

Coloring.

- Color right (left) edges in red (blue).
- Color a base edge (v_{t_i}, v_k) red if $i = 1$ and blue if $i = l$ and otherwise arbitrarily.

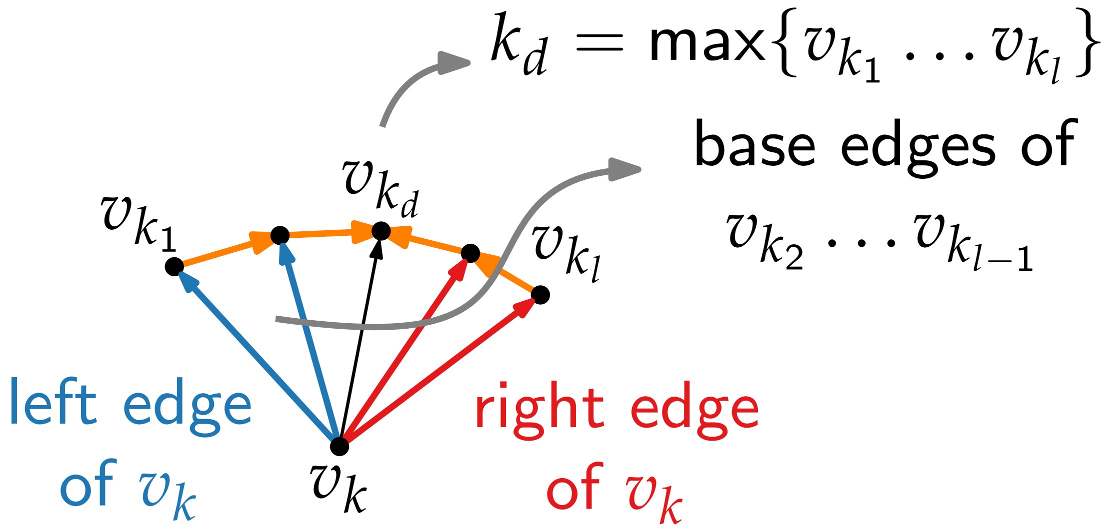
Let T_r be the red edges and T_b the blue edges.



Lemma 3.
$\{T_r, T_b\}$ is a regular edge labeling.

Proof.

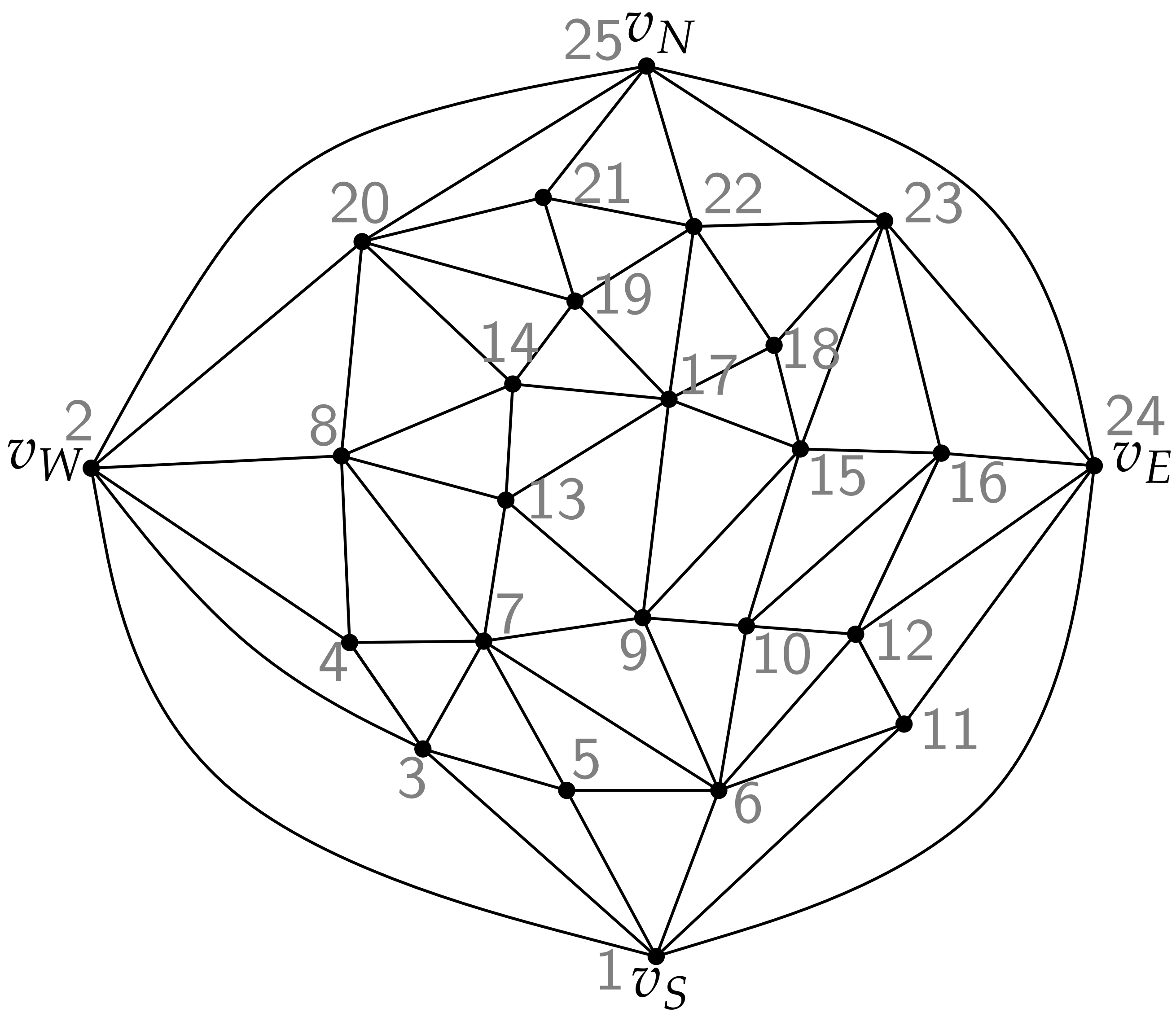
$k_l \geq 2$



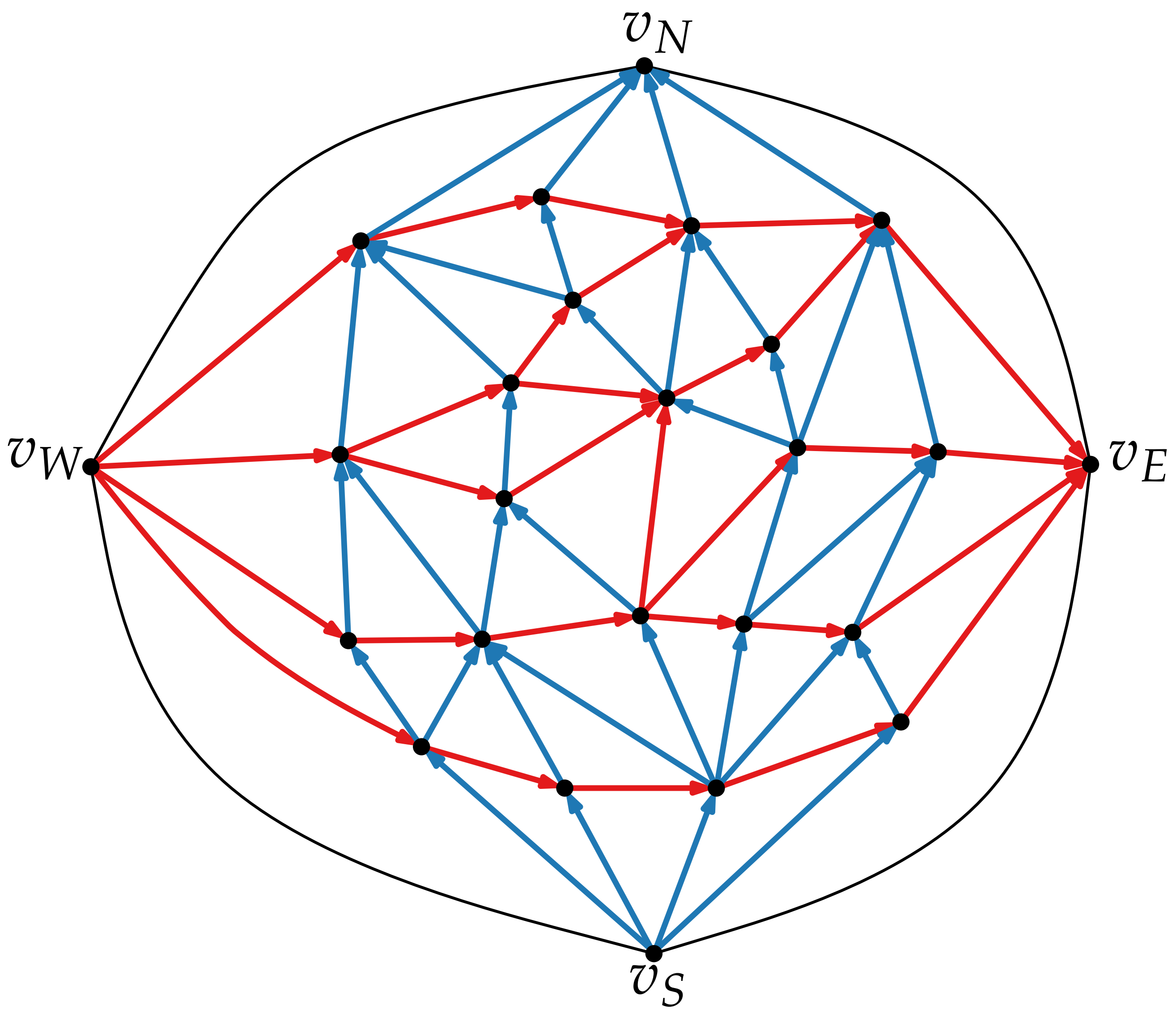
- $k_1 < k_2 < \ldots < k_d$ and $k_d > k_{d+1} > \ldots > k_l$
- (v_k, v_{k_i}), $2 \leq i \leq d-1$ are red
- (v_k, v_{k_i}), $d+1 \leq i \leq l-1$ are blue
- (v_k, v_{k_d}) is either red or blue

$\Rightarrow$ circular order of outgoing edges of v_k correct

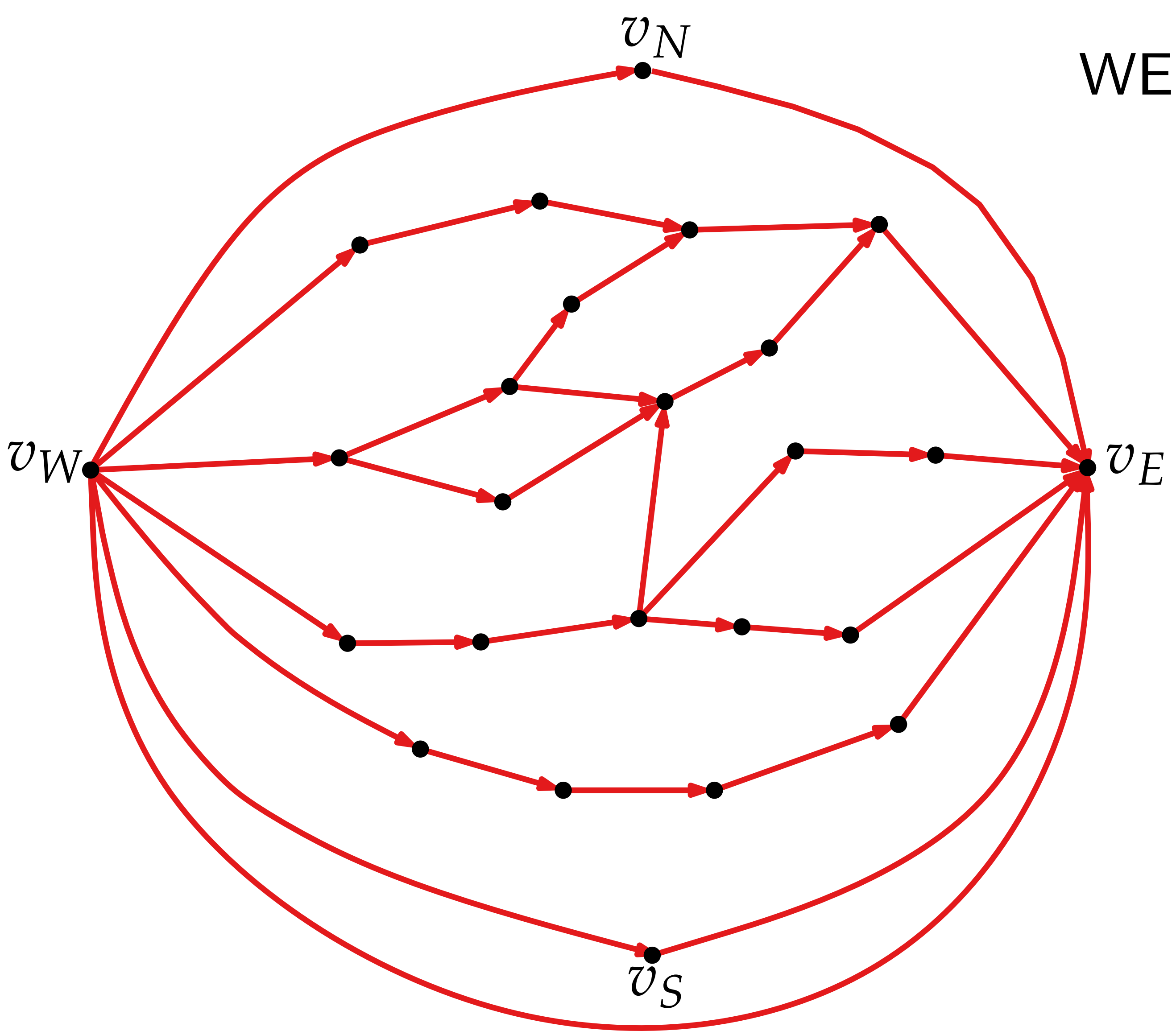
From REL to st-digraphs to coordinates



From REL to st-digraphs to coordinates

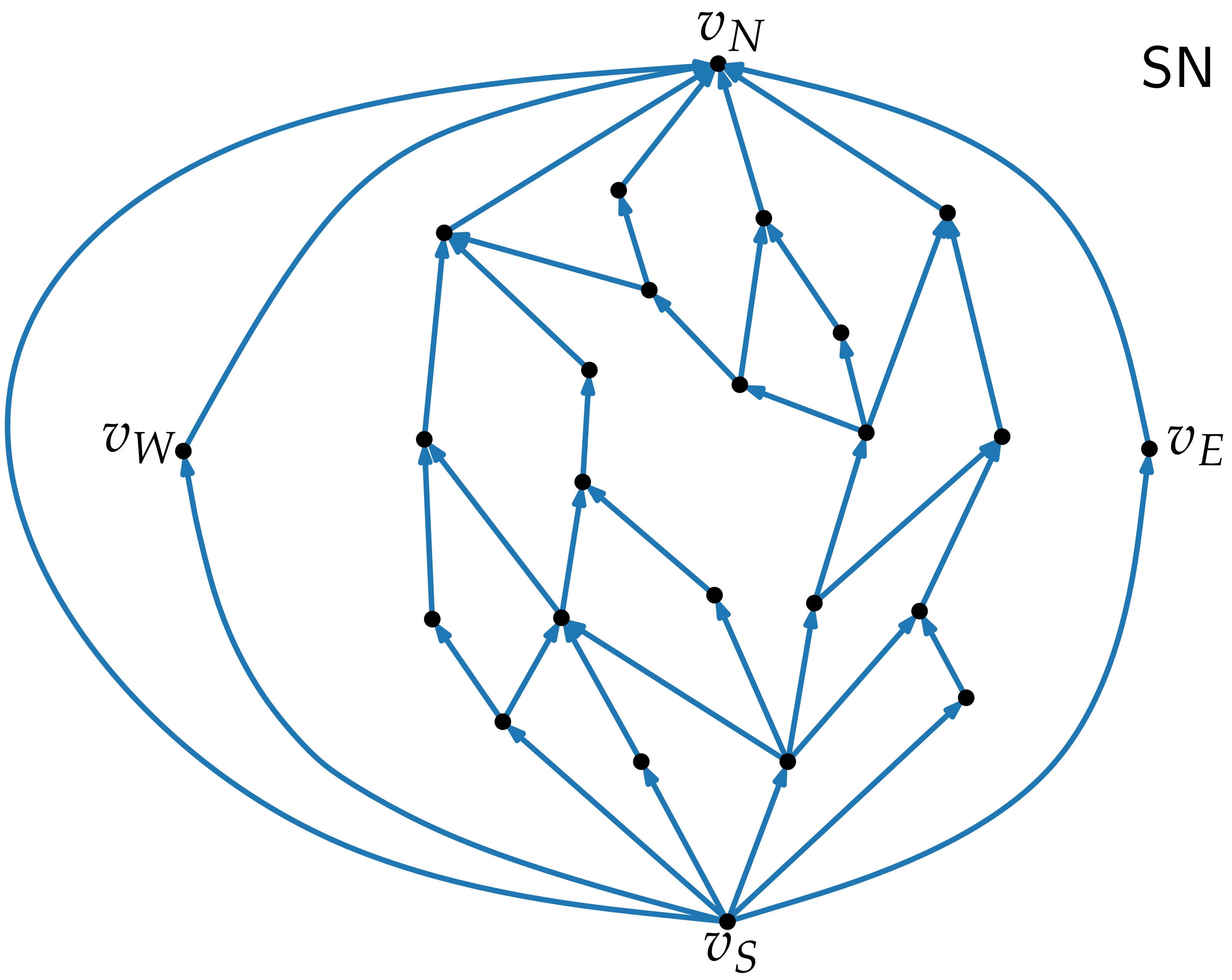


From REL to st-digraphs to coordinates



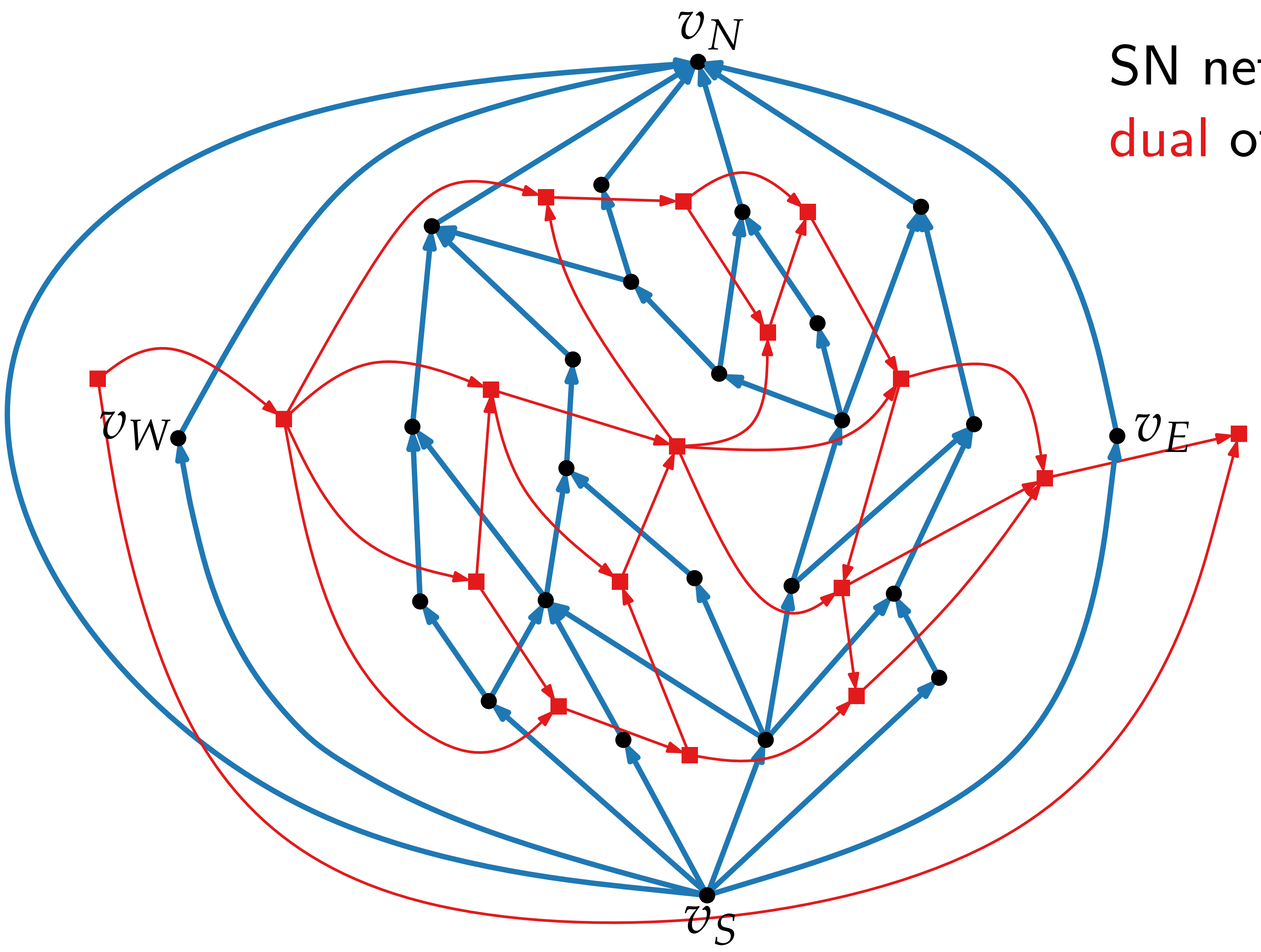
WE network G_{hor}

From REL to st-digraphs to coordinates



SN network G_{ver}

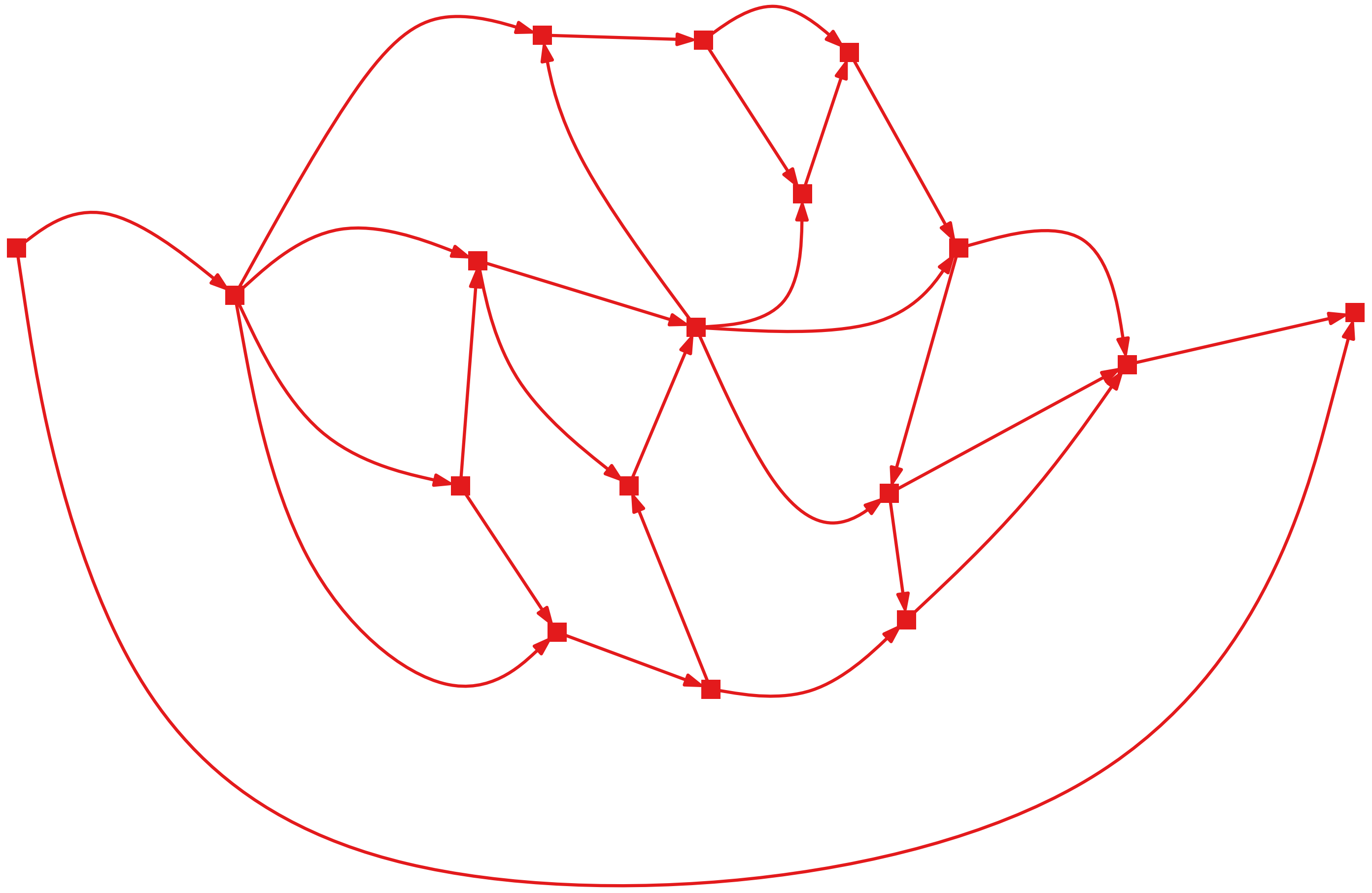
From REL to st-digraphs to coordinates



SN network G_{ver}

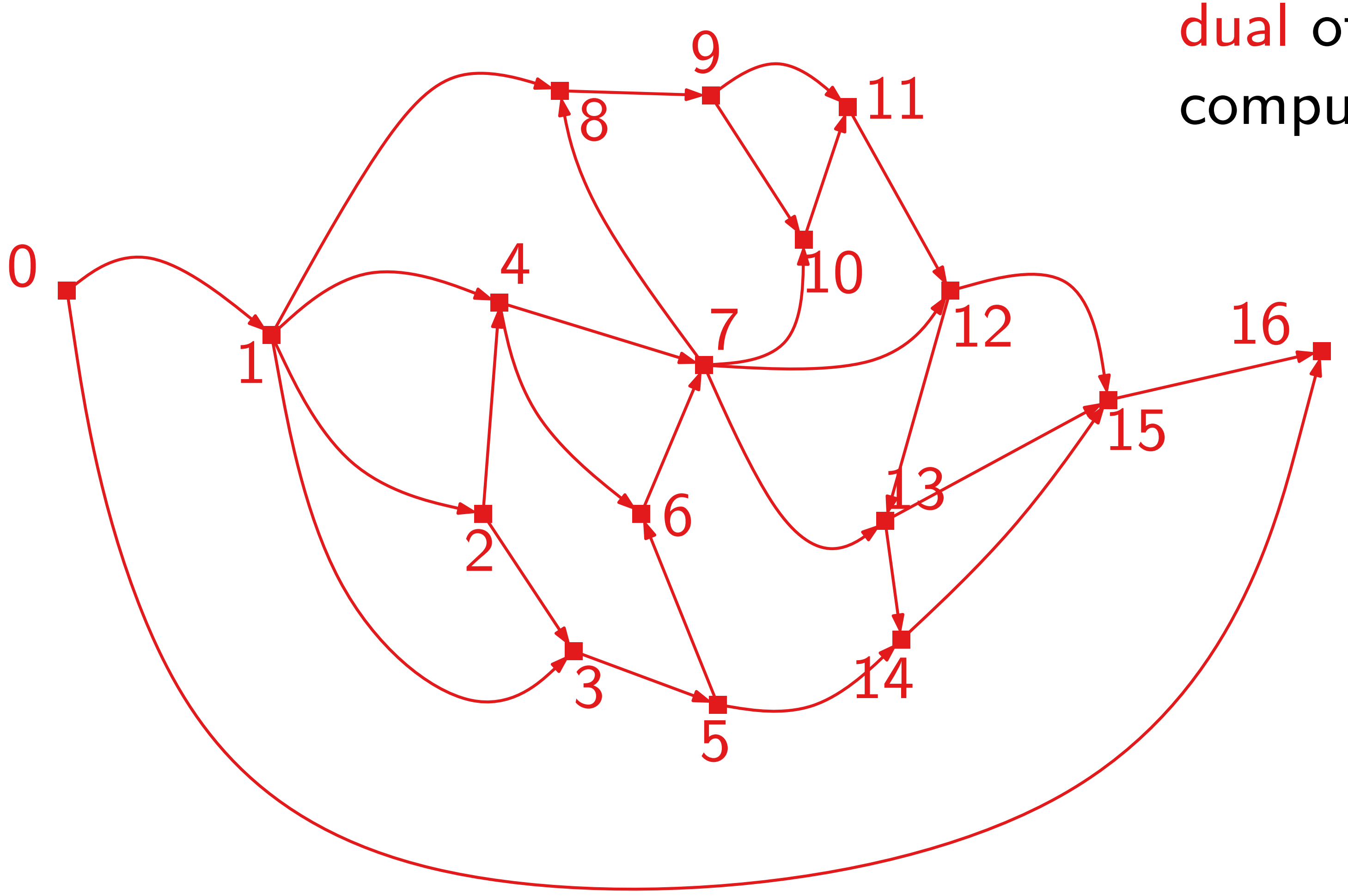
dual of G_{ver}

From REL to st-digraphs to coordinates



dual of G_{ver}

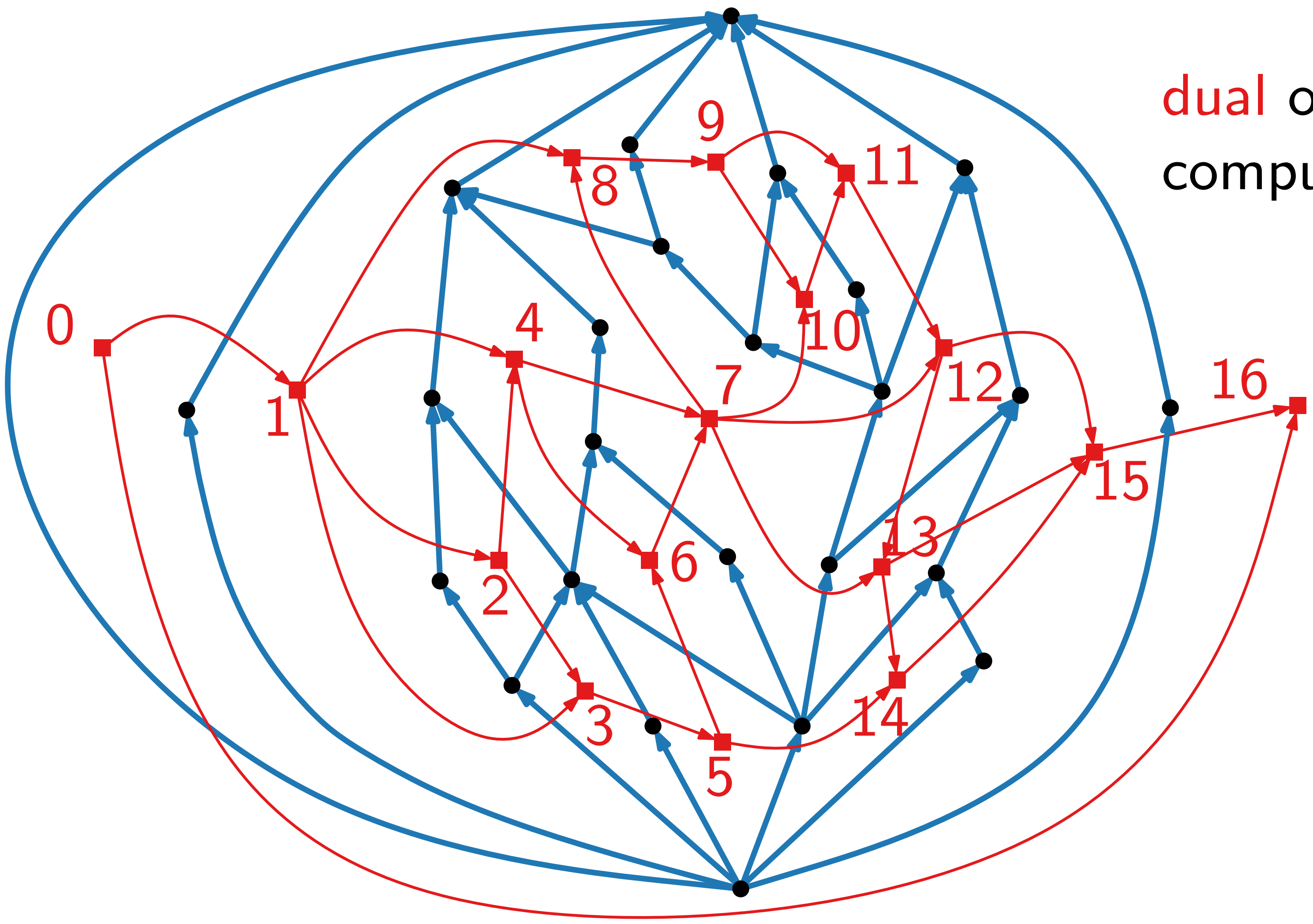
From REL to st-digraphs to coordinates



dual of G_{ver}

compute topological order

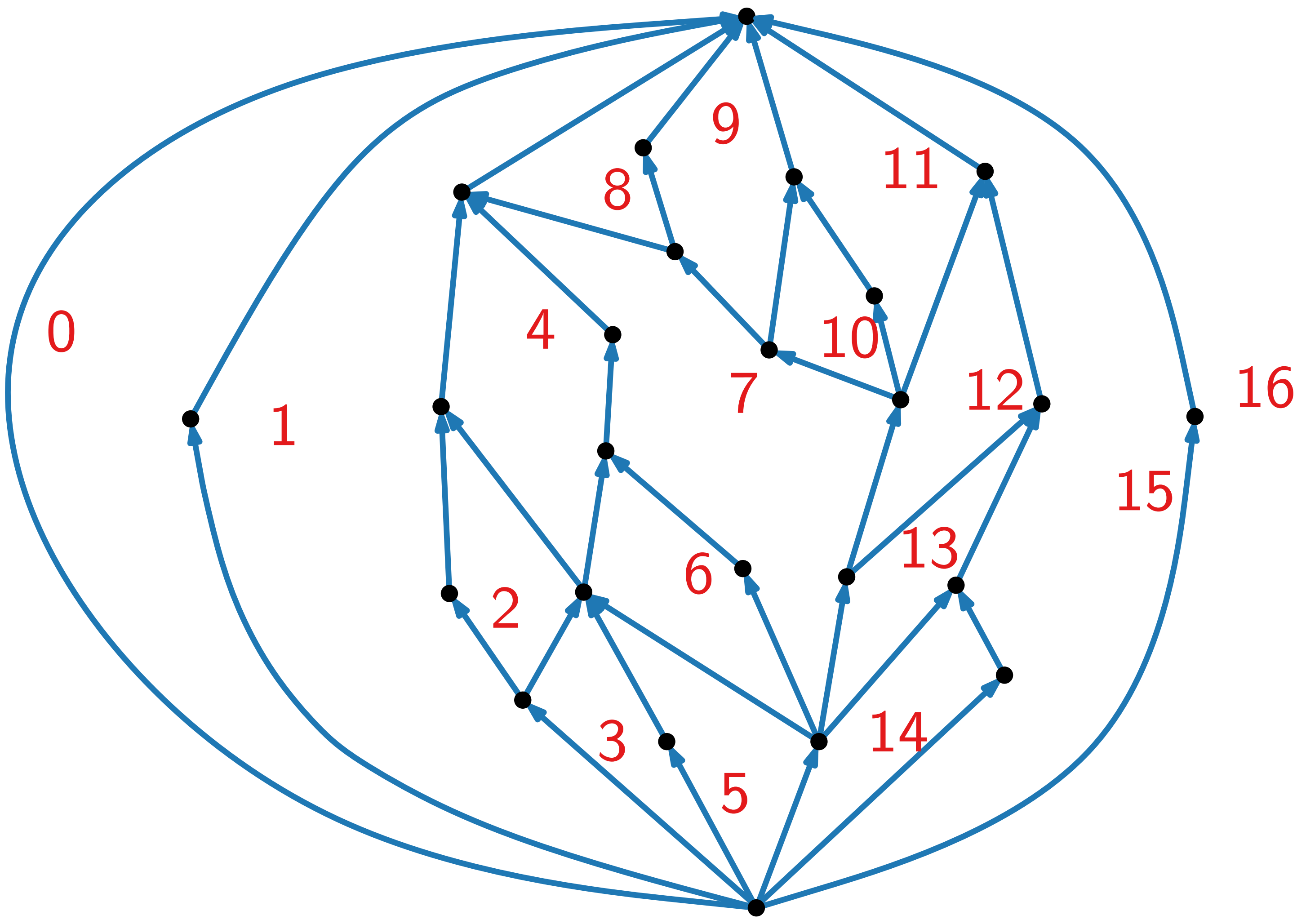
From REL to st-digraphs to coordinates



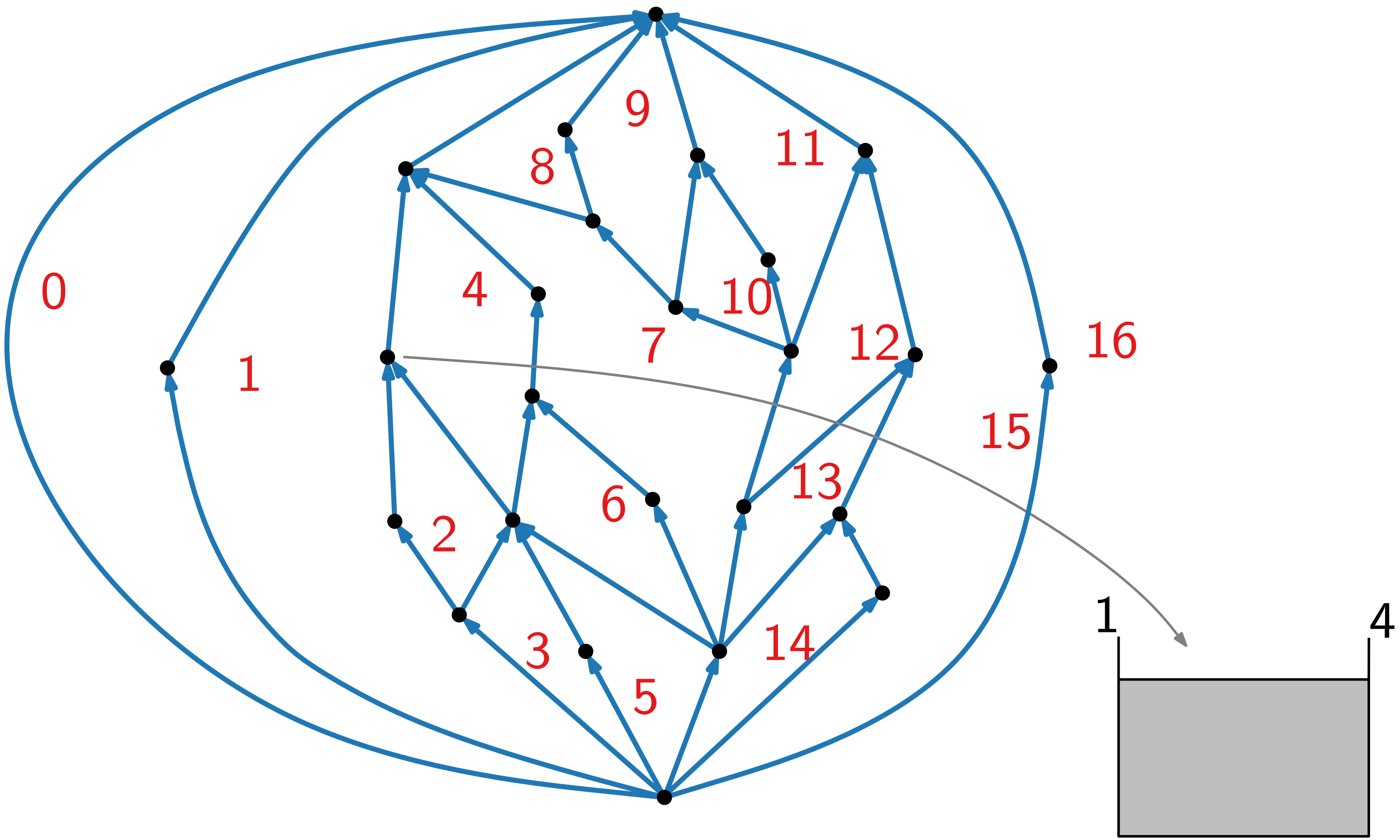
dual of G_{ver}

compute topological order

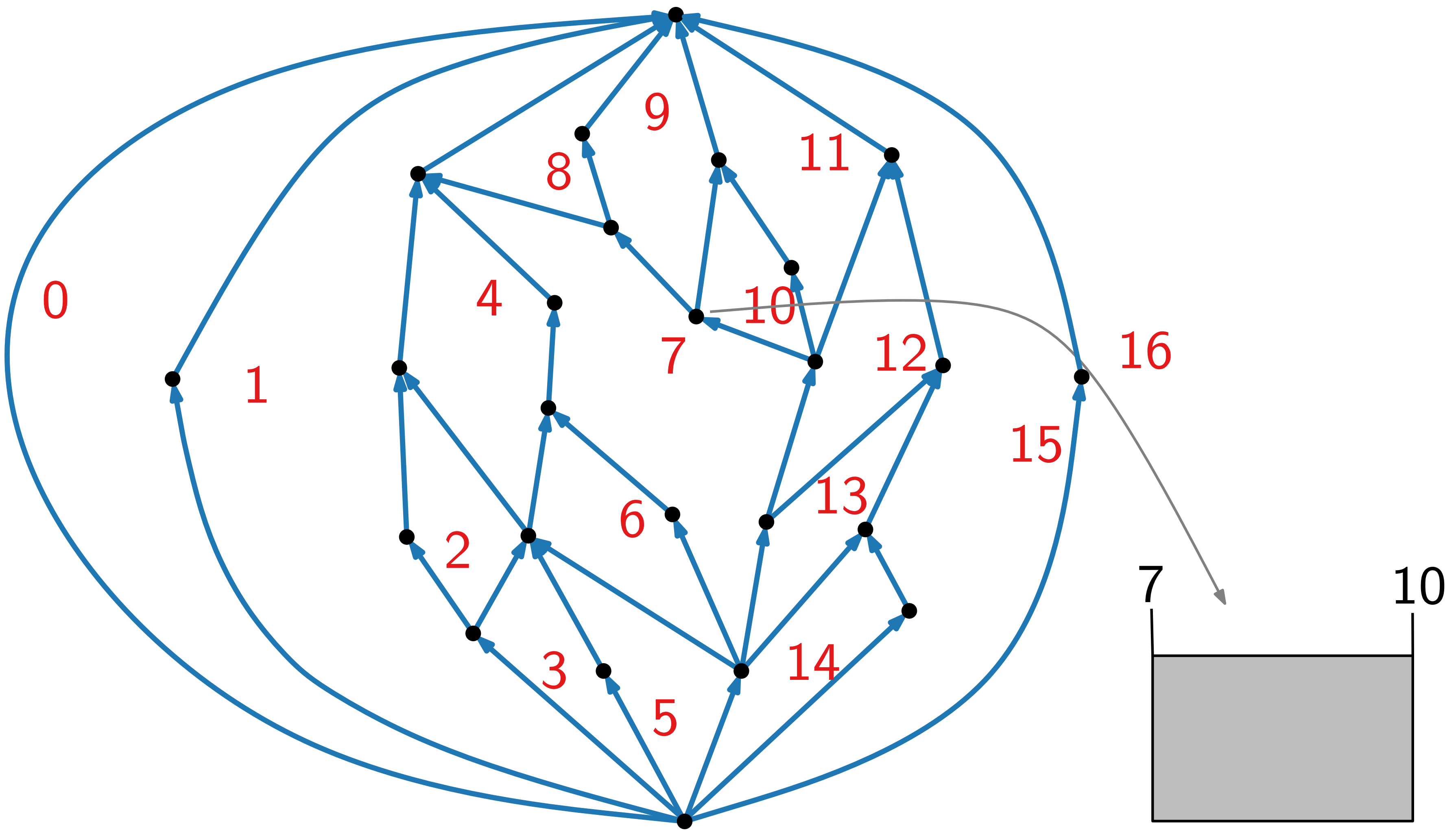
From REL to st-digraphs to coordinates



From REL to st-digraphs to coordinates



From REL to st-digraphs to coordinates



Rectangular dual algorithm

For a PTP graph $G = (V, E)$:

- Find a REL T_r, T_b of G;
- Construct a SN network G_{ver} of G (consists of T_b plus outer edges)
- Construct the dual $G^{\star}_{\text{ver}}$ of G_{ver} and compute a topological ordering f_{ver} of $G^{\star}_{\text{ver}}$
- For each vertex $v \in V$, let g and h be the face on the left and face on the right of v. Set $x_1(v) = f_{\text{ver}}(g)$ and $x_2(v) = f_{\text{ver}}(h)$.
- Define $x_1(v_N) = x_1(v_S) = 1$ and $x_2(v_N) = x_2(v_S) = \max f_{\text{ver}} - 1$

Rectangular dual algorithm

For a PTP graph $G = (V, E)$:

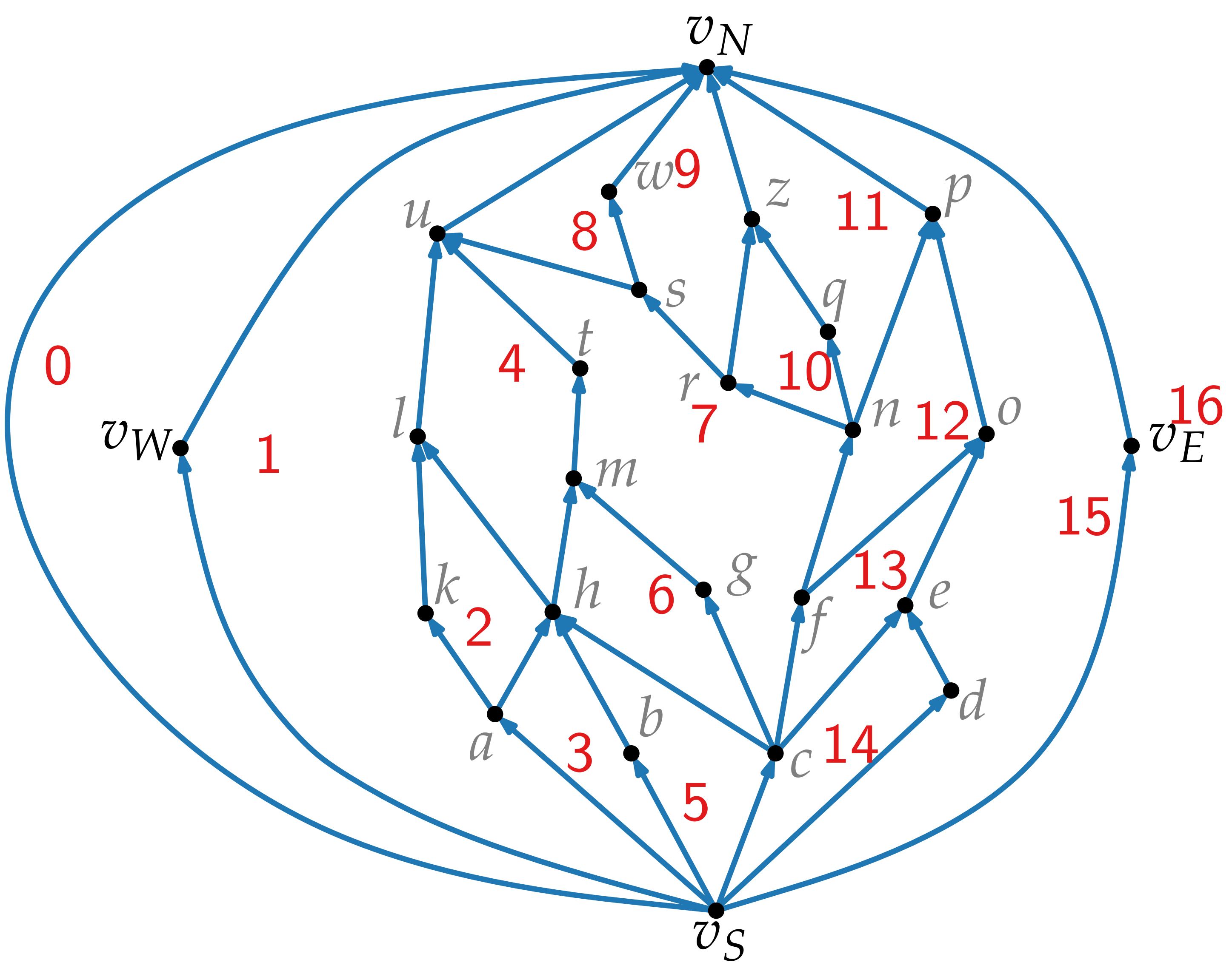
- Find a REL T_r, T_b of G;
- Construct a SN network G_{ver} of G (consists of T_b plus outer edges)
- Construct the dual $G^{\star}_{\text{ver}}$ of G_{ver} and compute a topological ordering f_{ver} of $G^{\star}_{\text{ver}}$
- For each vertex $v \in V$, let g and h be the face on the left and face on the right of v. Set $x_1(v) = f_{\text{ver}}(g)$ and $x_2(v) = f_{\text{ver}}(h)$.
- Define $x_1(v_N) = x_1(v_S) = 1$ and $x_2(v_N) = x_2(v_S) = \max f_{\text{ver}} - 1$
- Analogously compute y_1 and y_2 with G_{hor}.

Rectangular dual algorithm

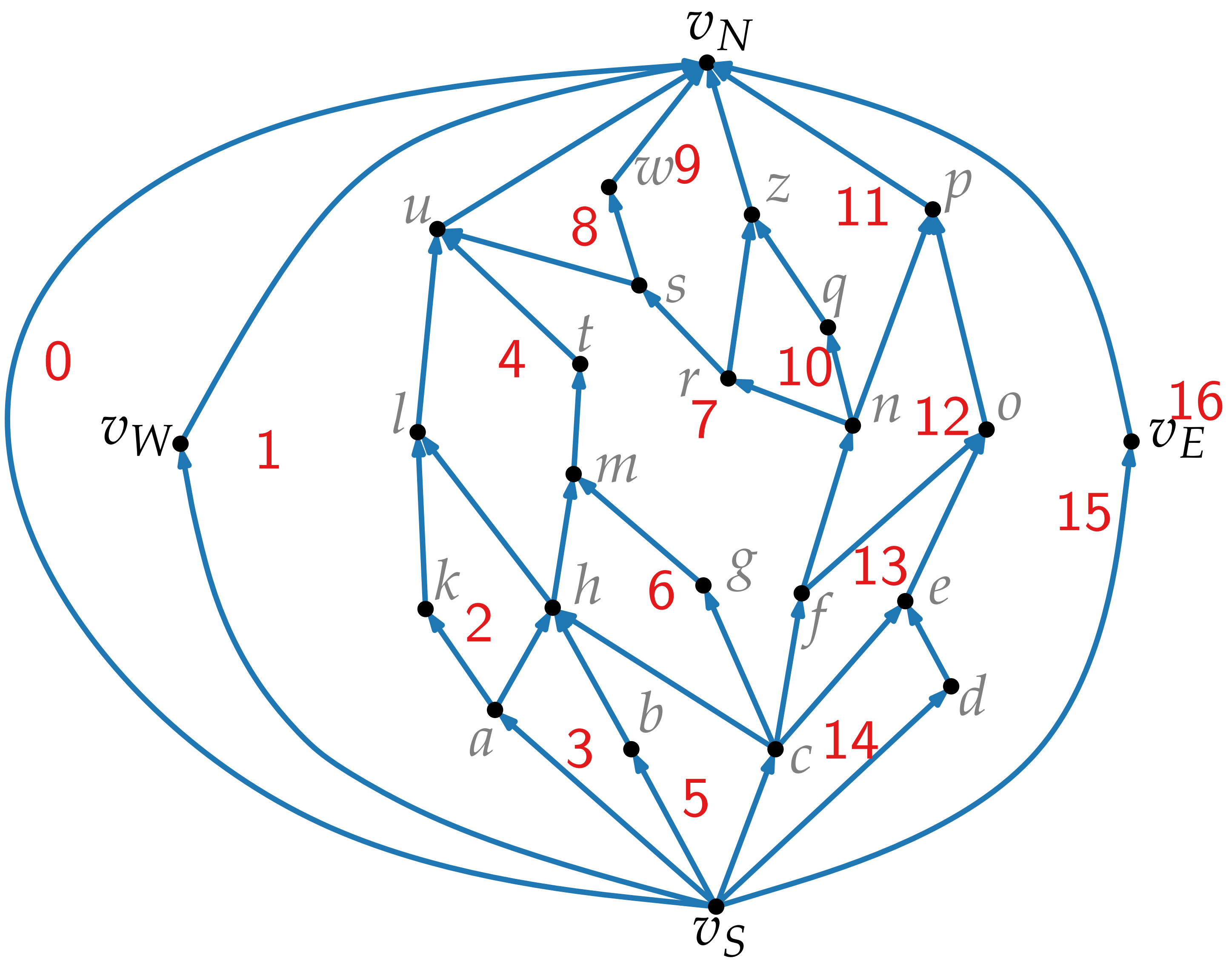
For a PTP graph $G = (V, E)$:

- Find a REL T_r, T_b of G;
- Construct a SN network G_{ver} of G (consists of T_b plus outer edges)
- Construct the dual $G^{\star}_{\text{ver}}$ of G_{ver} and compute a topological ordering f_{ver} of $G^{\star}_{\text{ver}}$
- For each vertex $v \in V$, let g and h be the face on the left and face on the right of v. Set $x_1(v) = f_{\text{ver}}(g)$ and $x_2(v) = f_{\text{ver}}(h)$.
- Define $x_1(v_N) = x_1(v_S) = 1$ and $x_2(v_N) = x_2(v_S) = \max f_{\text{ver}} - 1$
- Analogously compute y_1 and y_2 with G_{hor}.
- For each $v \in V$, assign a rectangle $R(v)$ bounded by x-coordinates $x_1(v),\ x_2(v)$ and y-coordinates $y_1(v),\ y_2(v)$.

Reading off coordinates to get rectangular dual

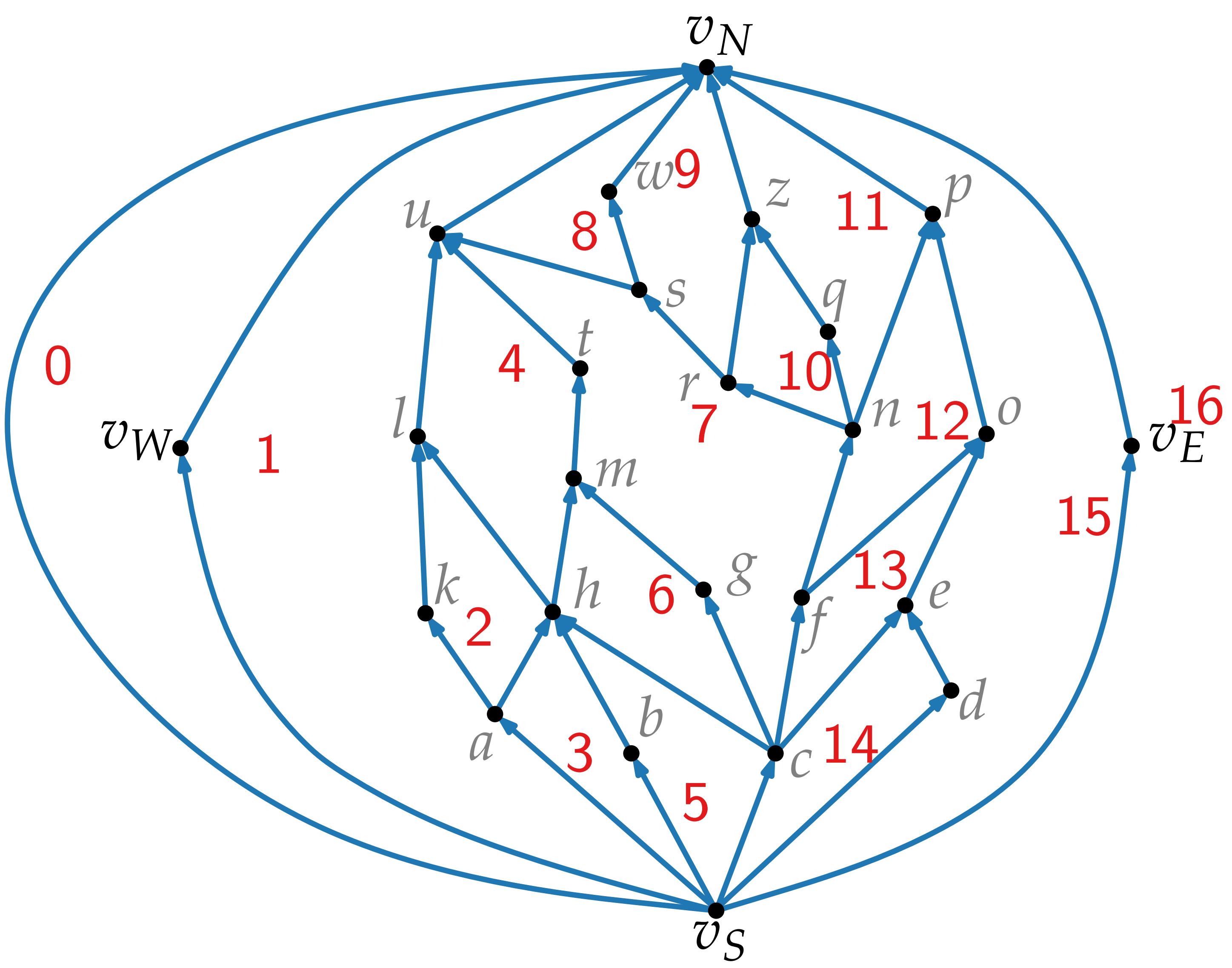


Reading off coordinates to get rectangular dual



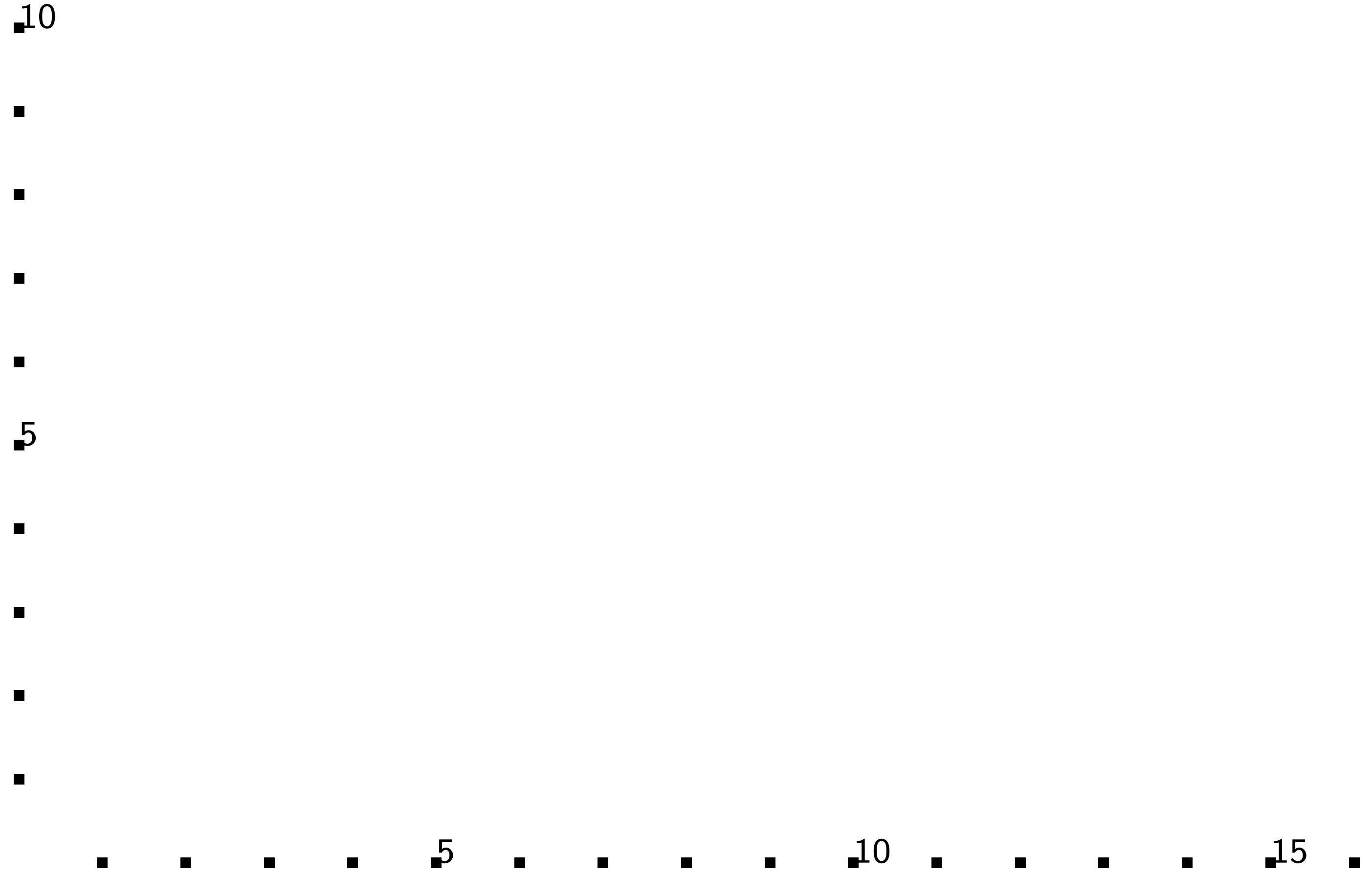
$x_1(v_N) = 1,\ x_2(v_N) = 15$
$x_1(v_S) = 1,\ x_2(v_S) = 15$
$x_1(v_W) = 0, x_2(v_W) = 1$
$x_1(v_E) = 15,\ x_2(v_E) = 16$
$x_1(a) = 1,\ x_2(a) = 3$
$x_1(b) = 3,\ x_2(b) = 5$
$x_1(c) = 5,\ x_2(c) = 14$
$x_1(d) = 14,\ x_2(d) = 15$
$x_1(e) = 13,\ x_2(e) = 15$
$\ldots$

Reading off coordinates to get rectangular dual



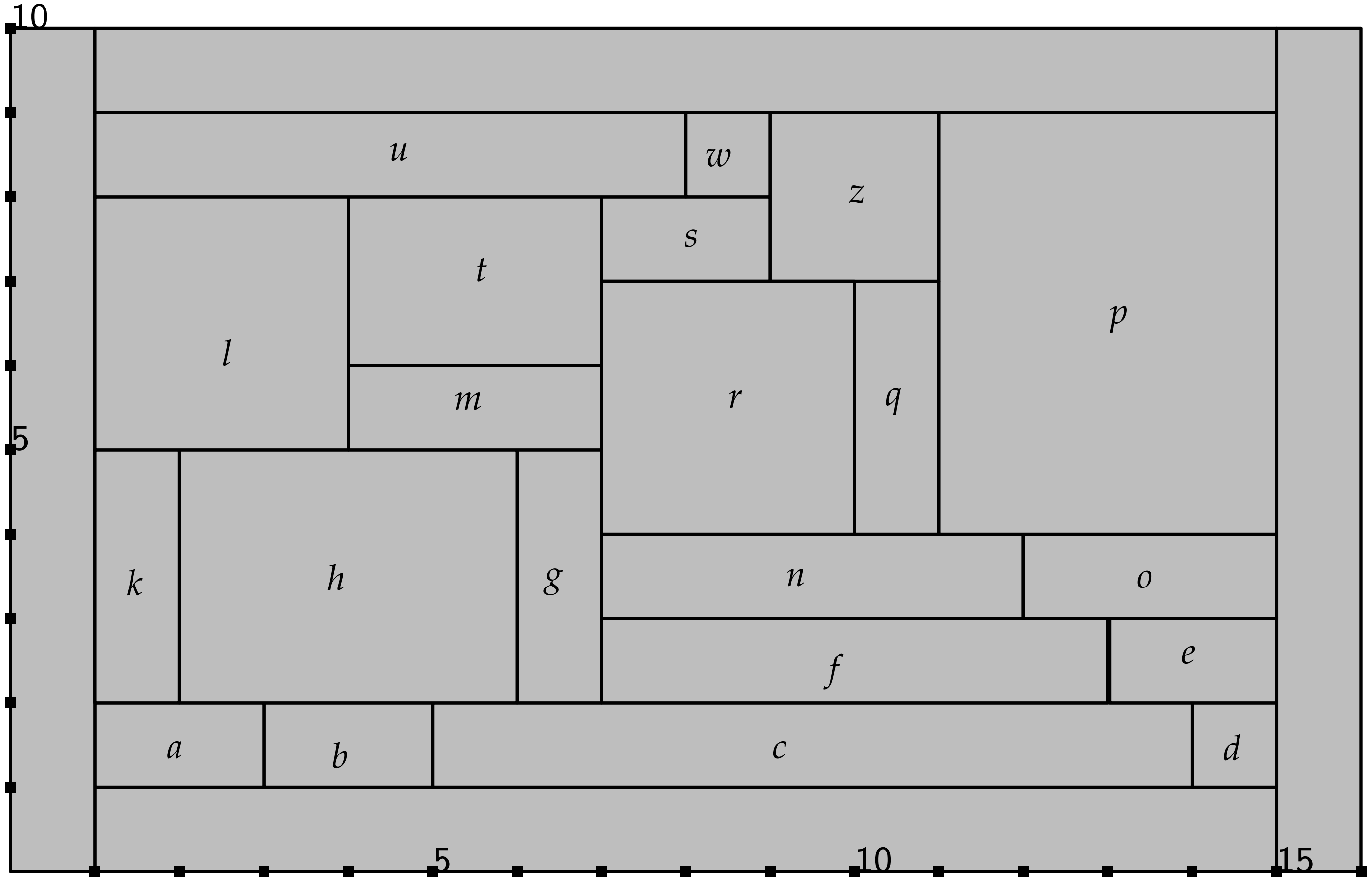
$x_1(v_N) = 1,\ x_2(v_N) = 15$
$x_1(v_S) = 1,\ x_2(v_S) = 15$
$x_1(v_W) = 0, x_2(v_W) = 1$
$x_1(v_E) = 15,\ x_2(v_E) = 16$
$x_1(a) = 1,\ x_2(a) = 3$
$x_1(b) = 3,\ x_2(b) = 5$
$x_1(c) = 5,\ x_2(c) = 14$
$x_1(d) = 14,\ x_2(d) = 15$
$x_1(e) = 13,\ x_2(e) = 15$
...
$y_1(v_W) = 0, y_2(v_W) = 10$
$y_1(v_E) = 0,\ y_2(v_E) = 10$
$y_1(v_N) = 9,\ y_2(v_N) = 10$
$y_1(v_S) = 0,\ y_2(v_S) = 1$
$y_1(a) = 1,\ y_2(a) = 2$
$y_1(b) = 1,\ y_2(b) = 2$
...

Reading off coordinates to get rectangular dual



$x_1(v_N) = 1,\ x_2(v_N) = 15$
$x_1(v_S) = 1,\ x_2(v_S) = 15$
$x_1(v_W) = 0, x_2(v_W) = 1$
$x_1(v_E) = 15,\ x_2(v_E) = 16$
$x_1(a) = 1,\ x_2(a) = 3$
$x_1(b) = 3,\ x_2(b) = 5$
$x_1(c) = 5,\ x_2(c) = 14$
$x_1(d) = 14,\ x_2(d) = 15$
$x_1(e) = 13,\ x_2(e) = 15$
$\ldots$
$y_1(v_W) = 0, y_2(v_W) = 10$
$y_1(v_E) = 0,\ y_2(v_E) = 10$
$y_1(v_N) = 9,\ y_2(v_N) = 10$
$y_1(v_S) = 0,\ y_2(v_S) = 1$
$y_1(a) = 1,\ y_2(a) = 2$
$y_1(b) = 1,\ y_2(b) = 2$
$\ldots$

Reading off coordinates to get rectangular dual



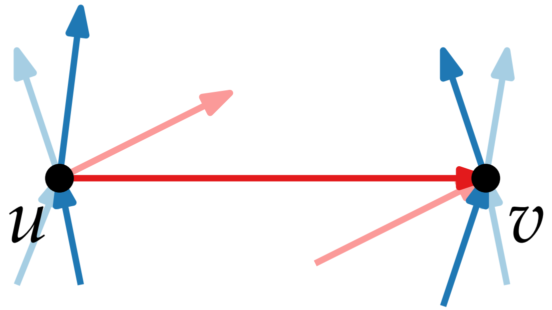
$x_1(v_N) = 1,\ x_2(v_N) = 15$
$x_1(v_S) = 1,\ x_2(v_S) = 15$
$x_1(v_W) = 0, x_2(v_W) = 1$
$x_1(v_E) = 15,\ x_2(v_E) = 16$
$x_1(a) = 1,\ x_2(a) = 3$
$x_1(b) = 3,\ x_2(b) = 5$
$x_1(c) = 5,\ x_2(c) = 14$
$x_1(d) = 14,\ x_2(d) = 15$
$x_1(e) = 13,\ x_2(e) = 15$
...
$y_1(v_W) = 0, y_2(v_W) = 10$
$y_1(v_E) = 0,\ y_2(v_E) = 10$
$y_1(v_N) = 9,\ y_2(v_N) = 10$
$y_1(v_S) = 0,\ y_2(v_S) = 1$
$y_1(a) = 1,\ y_2(a) = 2$
$y_1(b) = 1,\ y_2(b) = 2$
...

Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$

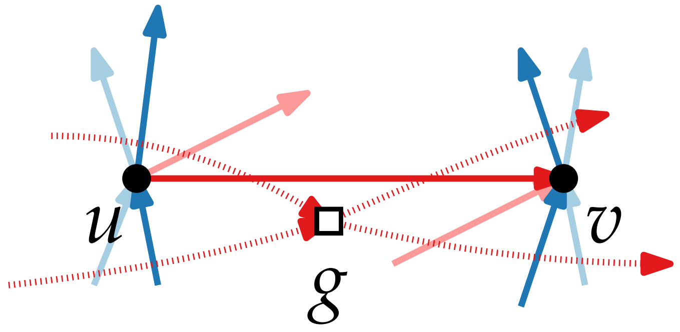
Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$



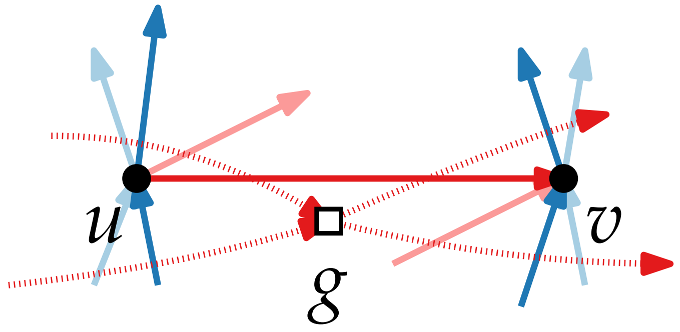
Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$



Correctness of algorithm (sketch)

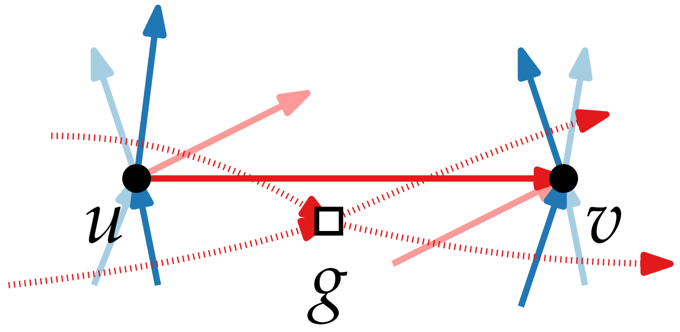
- If edge (u, v) existens, then $x_2(u) = x_1(v)$



$$x_2(u) = f_{\text{ver}}(g) = x_1(v)$$

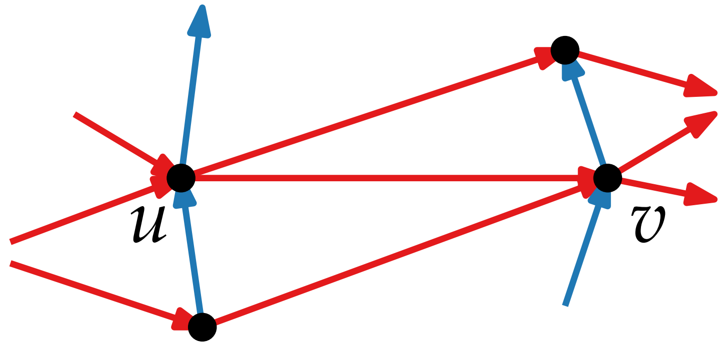
Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$



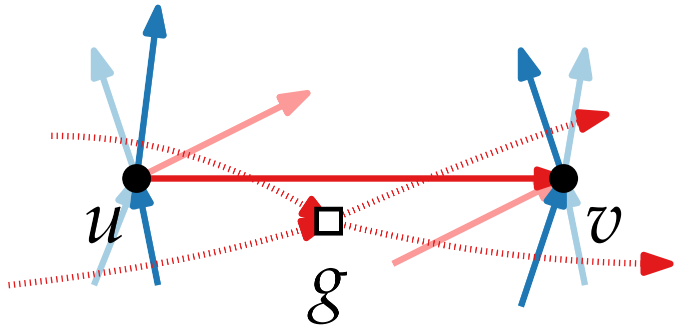
$$x_2(u) = f_{\text{ver}}(g) = x_1(v)$$

- and their veritcal segment of their rectangles overlap.



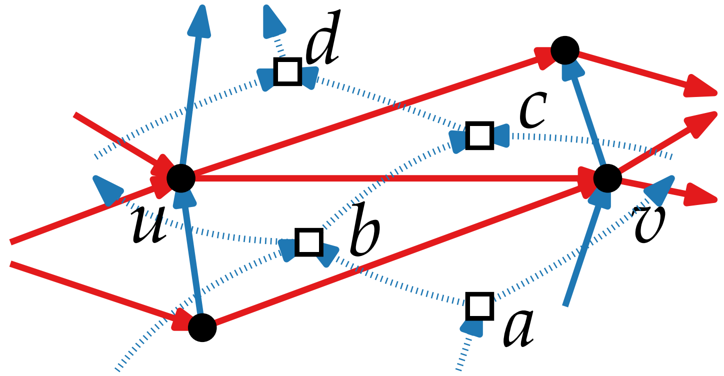
Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$



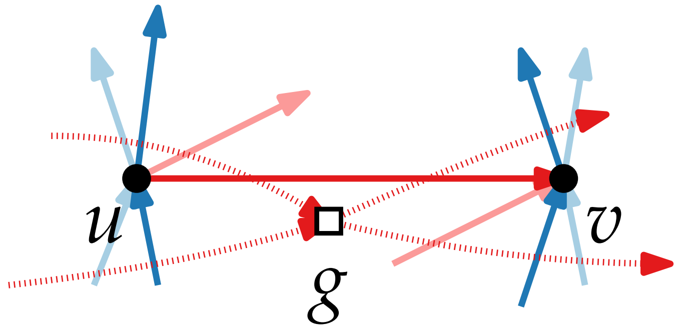
$$x_2(u) = f_{\text{ver}}(g) = x_1(v)$$

- and their veritcal segment of their rectangles overlap.



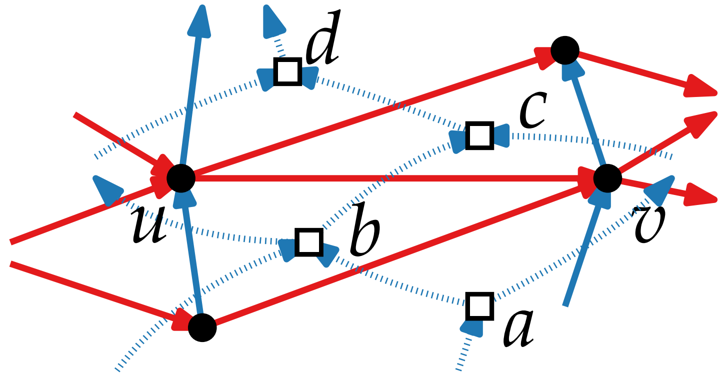
Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$



$$x_2(u) = f_{\text{ver}}(g) = x_1(v)$$

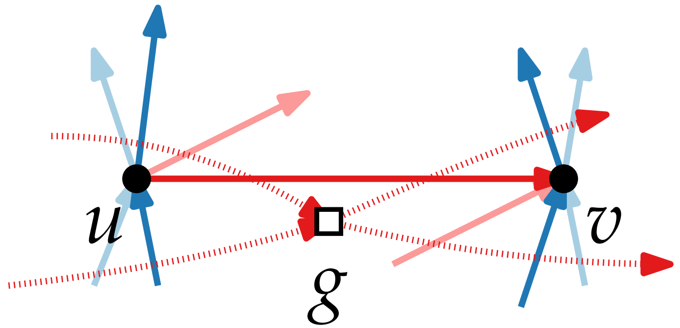
- and their veritcal segment of their rectangles overlap.



$$y_1(v) = f_{\text{hor}}(a) < y_1(u) = f_{\text{hor}}(b) < y_2(v) = f_{\text{hor}}(c) < y_2(u) = f_{\text{hor}}(d)$$

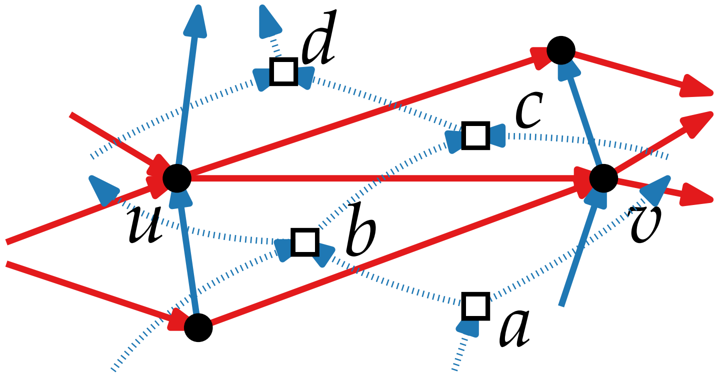
Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$



$$x_2(u) = f_{\text{ver}}(g) = x_1(v)$$

- and their veritcal segment of their rectangles overlap.

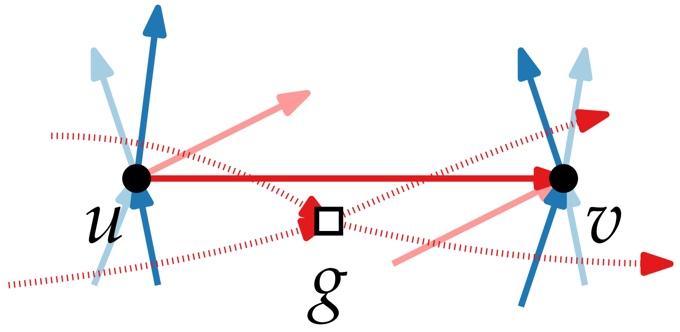


$$y_1(v) = f_{\text{hor}}(a) < y_1(u) = f_{\text{hor}}(b) < y_2(v) = f_{\text{hor}}(c) < y_2(u) = f_{\text{hor}}(d)$$

- If path from u to v in red at least two edges long, then $x_2(u) < x_1(v)$.

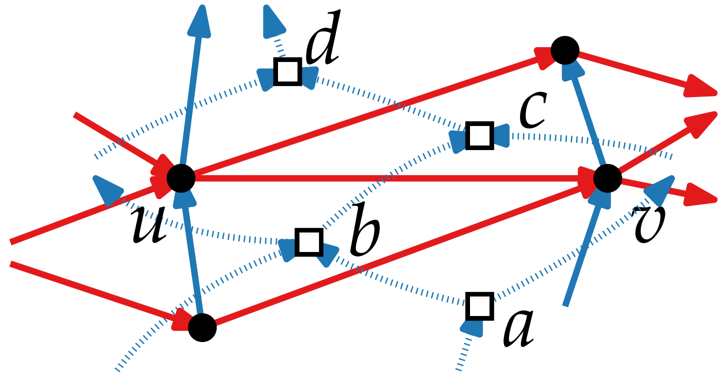
Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$



$$x_2(u) = f_{\text{ver}}(g) = x_1(v)$$

- and their veritcal segment of their rectangles overlap.

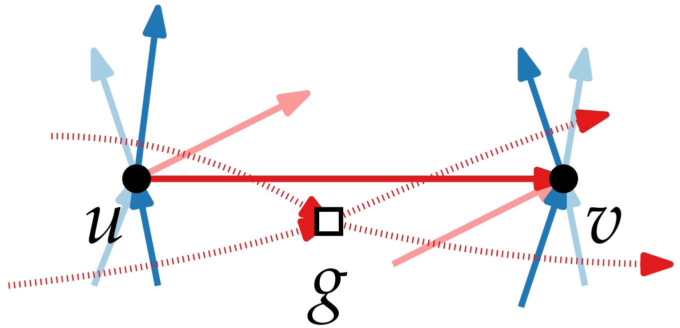


$$y_1(v) = f_{\text{hor}}(a) < y_1(u) = f_{\text{hor}}(b) < y_2(v) = f_{\text{hor}}(c) < y_2(u) = f_{\text{hor}}(d)$$

- If path from u to v in red at least two edges long, then $x_2(u) < x_1(v)$.
- No two boxes overlap.

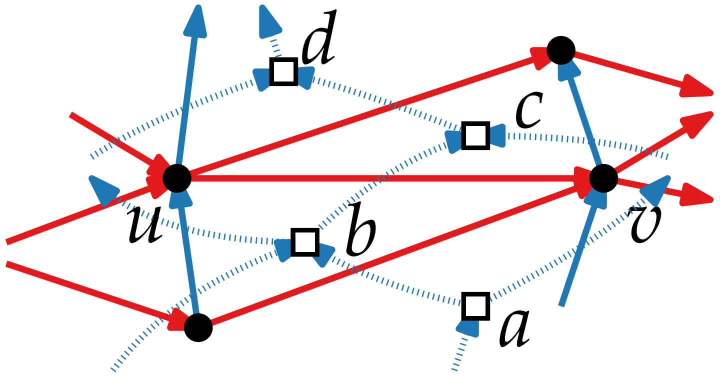
Correctness of algorithm (sketch)

- If edge (u, v) existens, then $x_2(u) = x_1(v)$



$$x_2(u) = f_{\text{ver}}(g) = x_1(v)$$

- and their veritcal segment of their rectangles overlap.



$$y_1(v) = f_{\text{hor}}(a) < y_1(u) = f_{\text{hor}}(b) <$$
$$y_2(v) = f_{\text{hor}}(c) < y_2(u) = f_{\text{hor}}(d)$$

- If path from u to v in red at least two edges long, then $x_2(u) < x_1(v)$.
- No two boxes overlap.

for details see He's paper [He '93]

Rectangular dual result

Theorem.
Every PTP graph G has a rectangular dual, which can be computed in linear time.

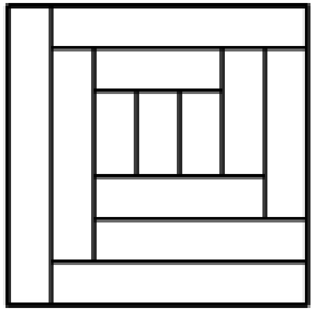
Proof.

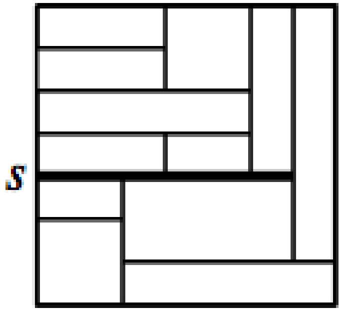
- Compute a planar embedding of G.
- Compute a refined canonical ordering of G.
- Traverse the graph and color the edges.
- Construct G_{ver} and G_{hor}.
- Construct their duals $G^\star_{\text{ver}}$ and $G^\star_{\text{hor}}$.
- Compute a topological ordering for vertices of $G^\star_{\text{ver}}$ and $G^\star_{\text{hor}}$.
- Assing coordinates to the rectangles representing vertices.

Discussion

- A layout is area-universal if any assignment of areas to rectangles can be realized by a combinatorially equivalent rectangular layout.
- A rectangular layout is **area-universal** if and only if it is **one-sided**. [Eppstein et al. SIAM J. Comp. 2012]

one-sided



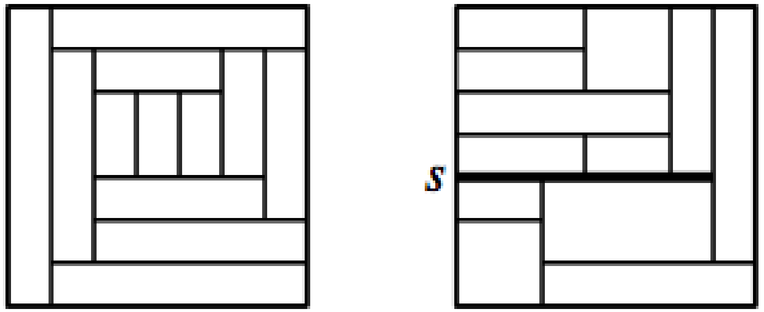


not one-sided

Discussion

- A layout is area-universal if any assignment of areas to rectangles can be realized by a combinatorially equivalent rectangular layout.
- A rectangular layout is **area-universal** if and only if it is **one-sided**. [Eppstein et al. SIAM J. Comp. 2012]

one-sided



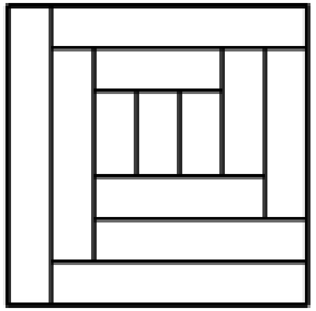
not one-sided

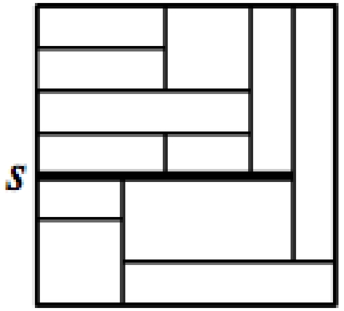
- Area universal **rectlinear** representation - possible for all planar graphs
- Alam et al. 2013: 8 sides (matches the lower bound)

Discussion

- A layout is area-universal if any assignment of areas to rectangles can be realized by a combinatorially equivalent rectangular layout.
- A rectangular layout is **area-universal** if and only if it is **one-sided**. [Eppstein et al. SIAM J. Comp. 2012]

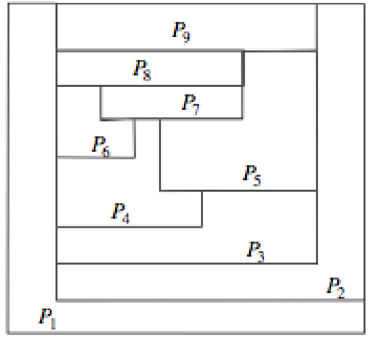
one-sided





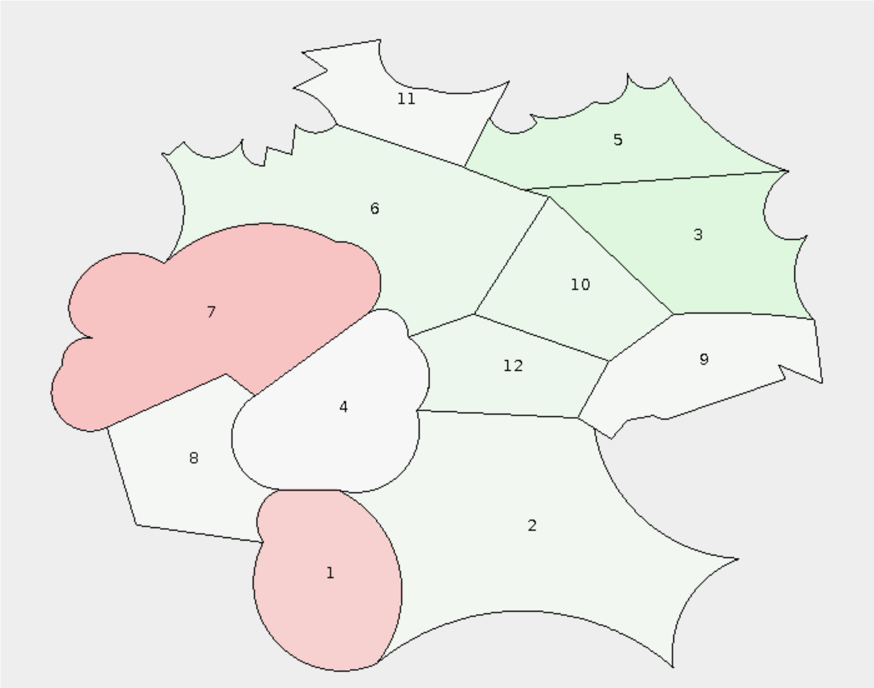
not one-sided

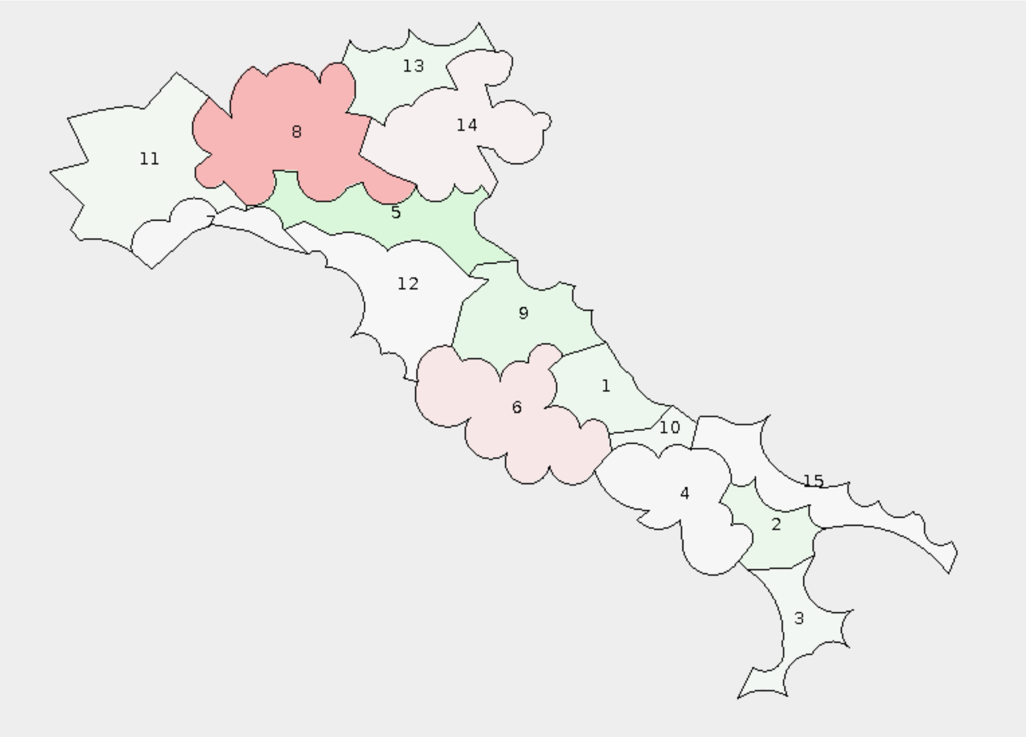
- Area universal **rectlinear** representation - possible for all planar graphs
- Alam et al. 2013: 8 sides (matches the lower bound)



Discussion

- Circular Arc Cartograms [Kämper, Kobourov, Nöllenburg. IEEE PasViz 2013]





Source: `http://cartogram.cs.arizona.edu`

Literature

Construction of triangle contact representations based on

- [de Fraysseix, de Mendez, Rosenstiehl ’94] On Triangle Contact Graphs

Construction of rectangular dual based on

- [He ’93] On Finding the Rectangular Duals of Planar Triangulated Graphs
- [Kant, He ’94] Two algorithms for finding rectangular duals of planar graphs

and originally from

- [Koźmiński, Kinnen ’85] Rectangular Duals of Planar Graphs