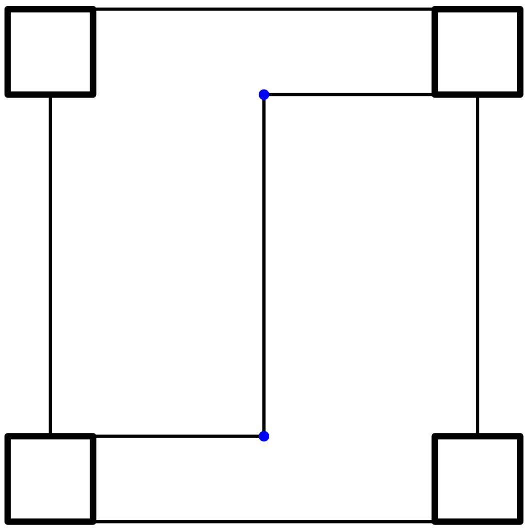
Smoothening of Kandinsky Drawings

Benjamin Ulvi Çoban

Wilhelm-Schickard-Institut für Informatik
Universität Tübingen, Germany

Kandinsky drawings

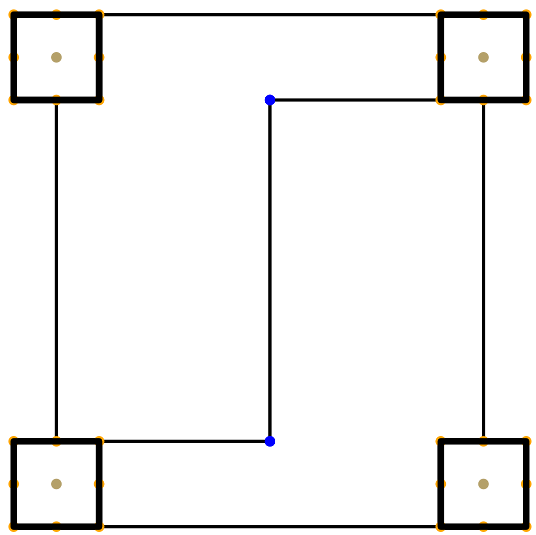
- Well-established orthogonal graph drawing model



Kandinsky
Complexity 3

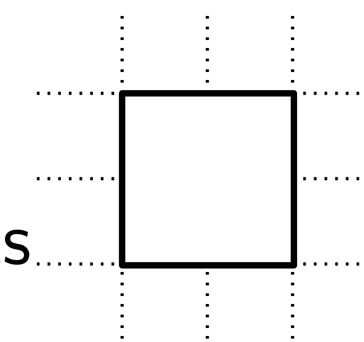
Kandinsky drawings

- Well-established orthogonal graph drawing model



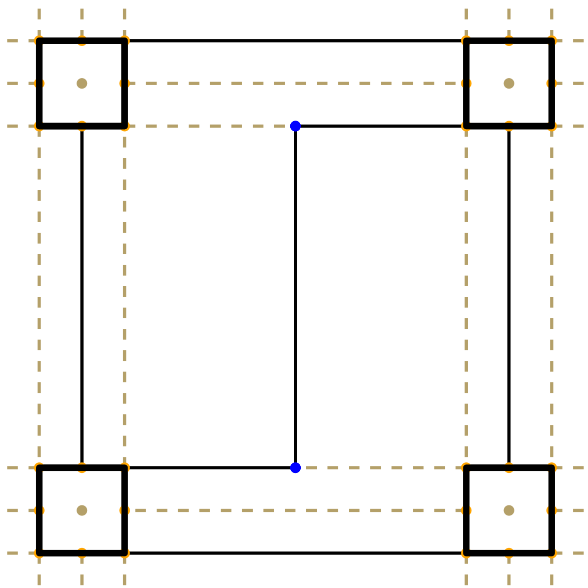
Kandinsky
Complexity 3

- underlying grid embedding
- vertices as boxes of uniform size, inherit ports



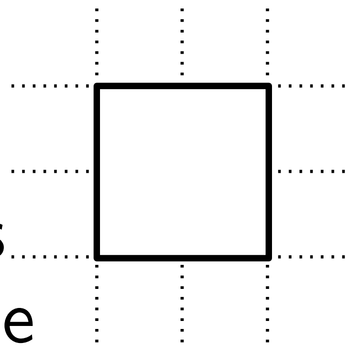
Kandinsky drawings

- Well-established orthogonal graph drawing model



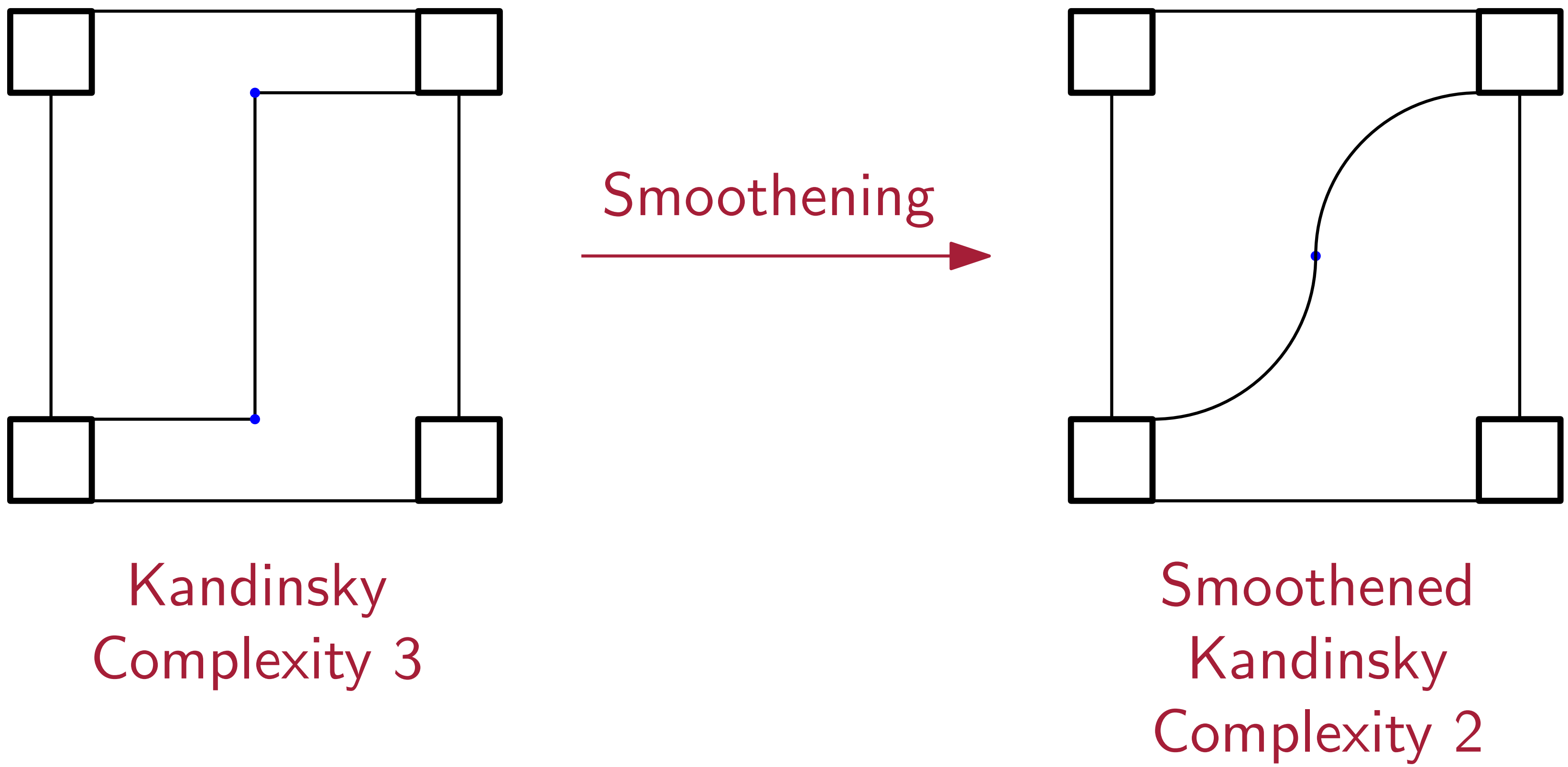
Kandinsky
Complexity 3

- underlying grid embedding
- vertices as boxes of uniform size, inherit ports
- polyedges as non-empty line segment sequence



Kandinsky drawings

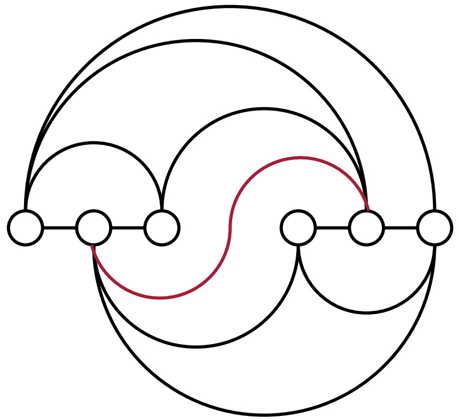
- Well-established orthogonal graph drawing model



- new segments - quarter / semi circular arcs
- Clarity + Aesthetics combined

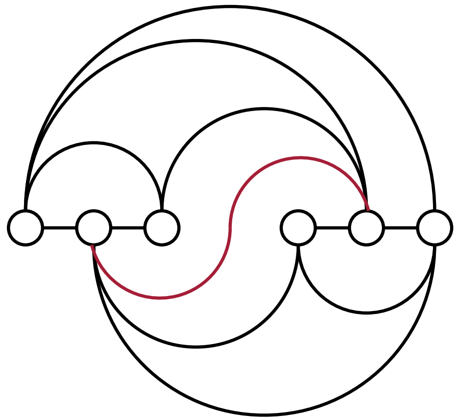
Previous results

- **Kandinsky drawings**
  - admit a complexity-2 smoothened drawing in $\mathcal{O}(n^2)$ area inspired by book embeddings [Bekos et al. 2013]

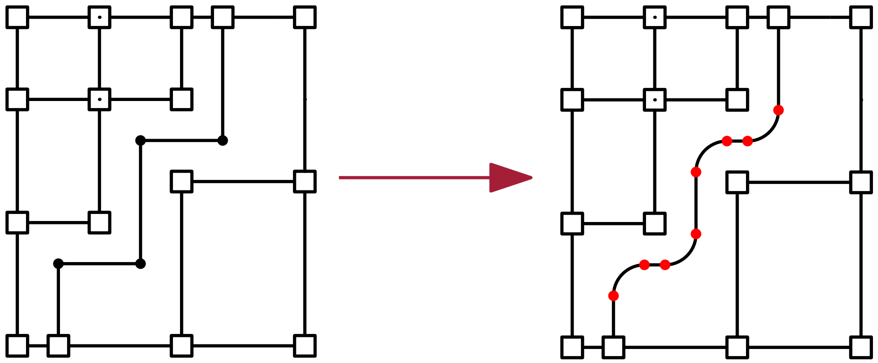


Previous results

- **Kandinsky drawings**
  - admit a complexity-2 smoothened drawing in $\mathcal{O}(n^2)$ area inspired by book embeddings [Bekos et al. 2013]

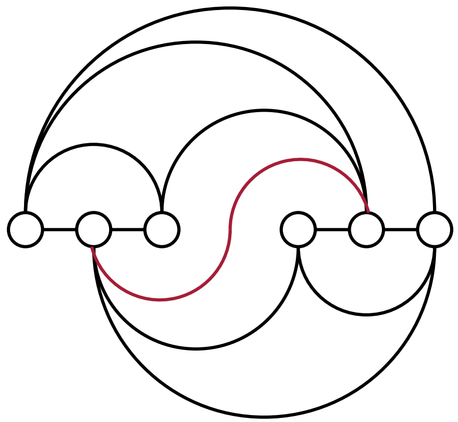


- **Orthogonal drawings with max degree 4**
  - Fixation of the vertex boxes lead to a high complexity increase [Bekos et al. 2013]

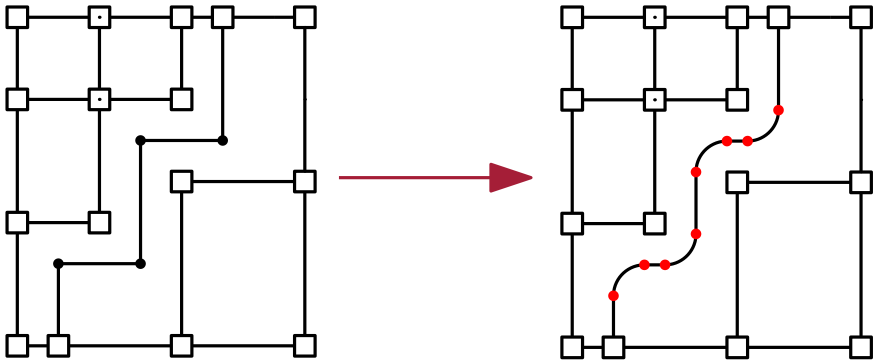


Previous results

- **Kandinsky drawings**
  - admit a complexity-2 smoothened drawing in $\mathcal{O}(n^2)$ area inspired by book embeddings [Bekos et al. 2013]



- **Orthogonal drawings with max degree 4**
  - Fixation of the vertex boxes lead to a high complexity increase [Bekos et al. 2013]



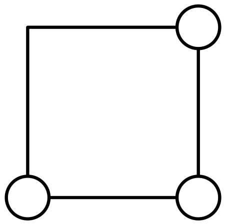
- **Disadvantages**
  - Either shape alternation or high complexity increase

Stretching technique

- **Orthogonal drawings with max degree 4 can be smoothened with reasonable complexity increase and area consumption** [Bekos et al. 2013]
- Stretching guarantees area for quarter circular arc substitution

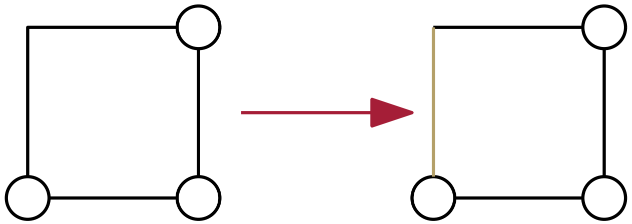
Stretching technique

- **Orthogonal drawings with max degree 4 can be smoothened with reasonable complexity increase and area consumption** [Bekos et al. 2013]
- Stretching guarantees area for quarter circular arc substitution
- Plane Sweep implementation
  - horizontal line segments as events
  - elongated by the length of the longest vertical segment



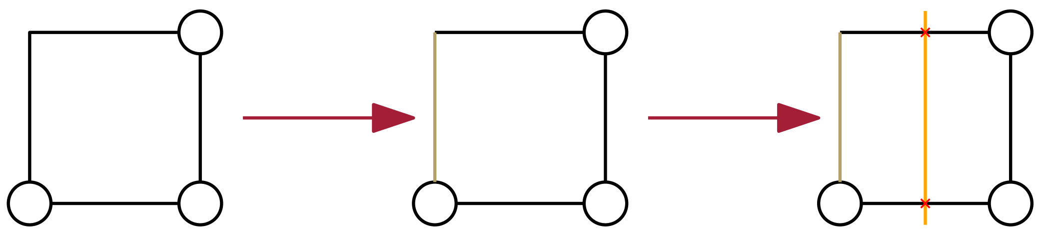
Stretching technique

- **Orthogonal drawings with max degree 4 can be smoothened with reasonable complexity increase and area consumption** [Bekos et al. 2013]
- Stretching guarantees area for quarter circular arc substitution
- Plane Sweep implementation
  - horizontal line segments as events
  - elongated by the length of the longest vertical segment



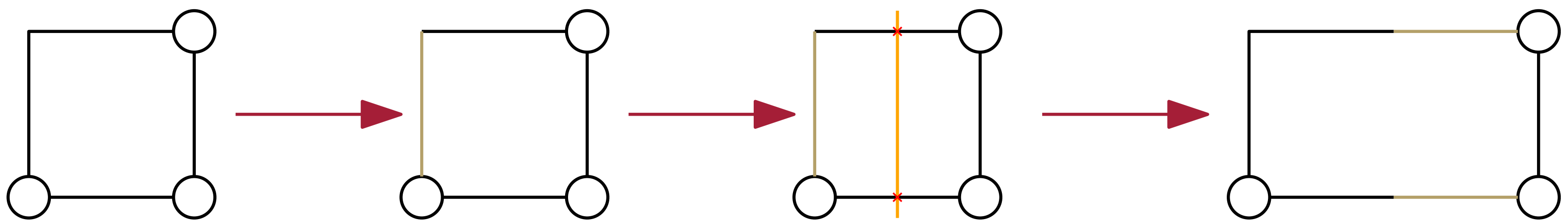
Stretching technique

- **Orthogonal drawings with max degree 4 can be smoothened with reasonable complexity increase and area consumption** [Bekos et al. 2013]
- Stretching guarantees area for quarter circular arc substitution
- Plane Sweep implementation
  - horizontal line segments as events
  - elongated by the length of the longest vertical segment



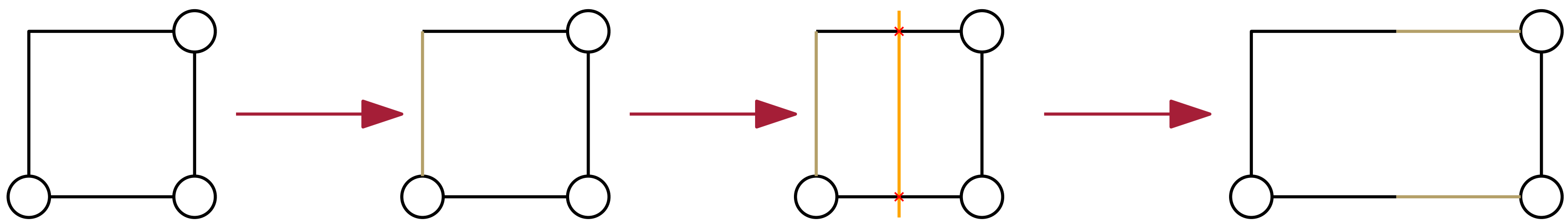
Stretching technique

- **Orthogonal drawings with max degree 4 can be smoothened with reasonable complexity increase and area consumption** [Bekos et al. 2013]
- Stretching guarantees area for quarter circular arc substitution
- Plane Sweep implementation
  - horizontal line segments as events
  - elongated by the length of the longest vertical segment

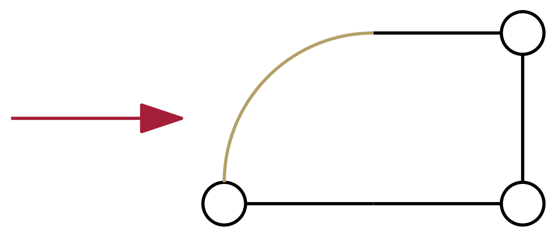


Stretching technique

- **Orthogonal drawings with max degree 4 can be smoothened with reasonable complexity increase and area consumption** [Bekos et al. 2013]
- Stretching guarantees area for quarter circular arc substitution
- Plane Sweep implementation
  - horizontal line segments as events
  - elongated by the length of the longest vertical segment



- quarter circular arc substitution at vertical segments

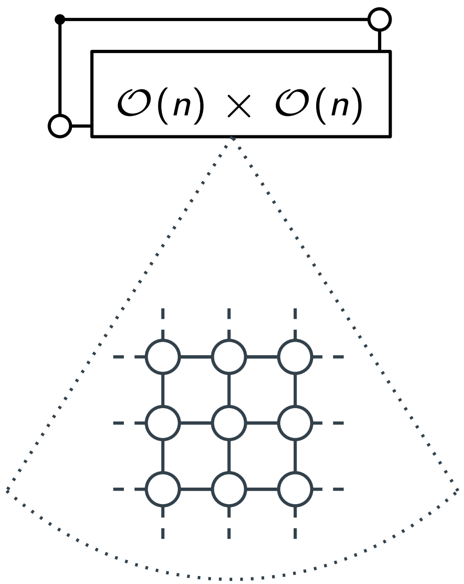


Stretching Technique (cnt.)

- linear runtime relative to the width of the drawing
- Does not alter the drawing drastically

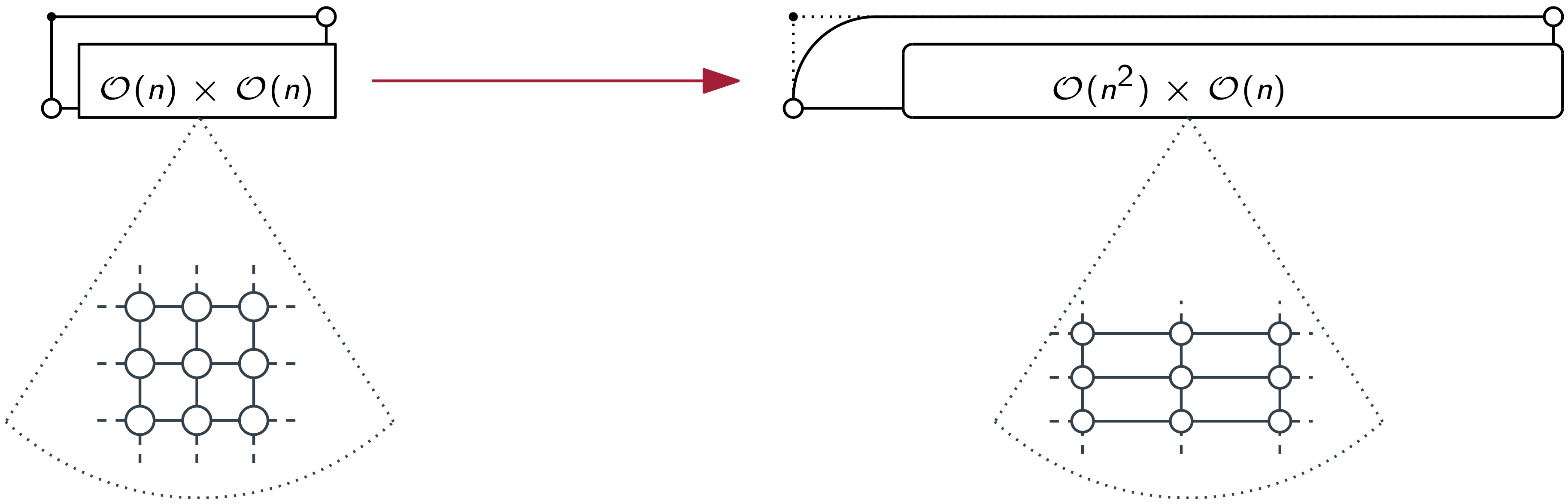
Stretching Technique (cnt.)

- linear runtime relative to the width of the drawing
- Does not alter the drawing drastically
- worst case area: quadratic in width size



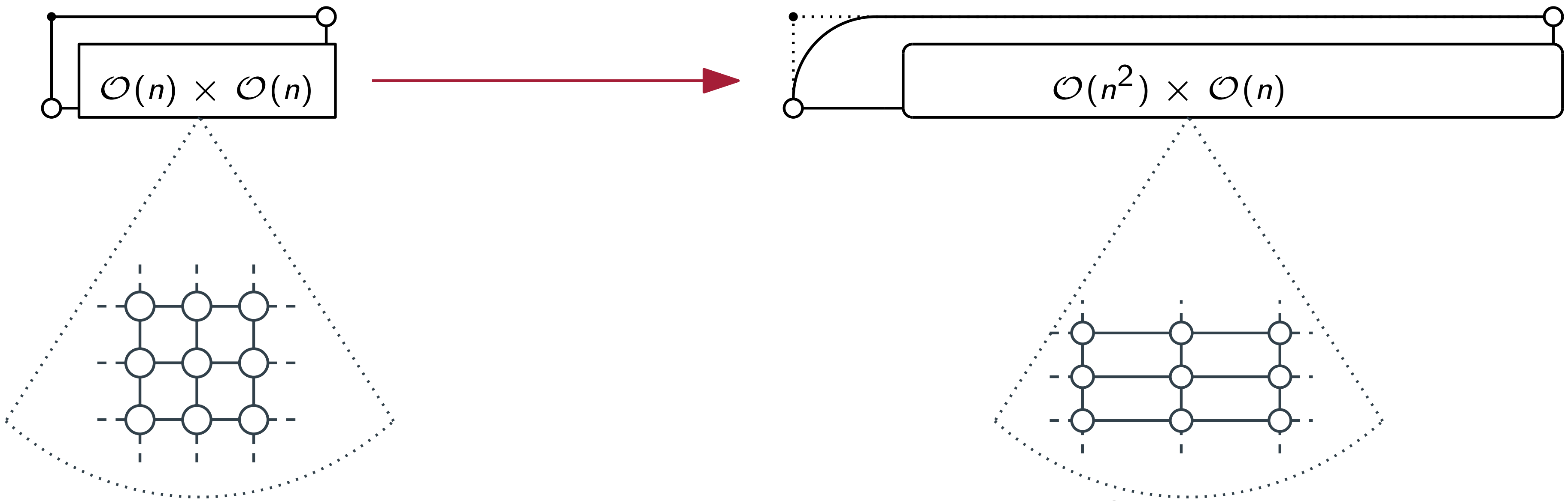
Stretching Technique (cnt.)

- linear runtime relative to the width of the drawing
- Does not alter the drawing drastically
- worst case area: quadratic in width size

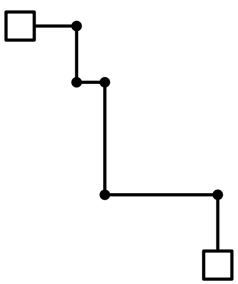


Stretching Technique (cnt.)

- linear runtime relative to the width of the drawing
- Does not alter the drawing drastically
- worst case area: quadratic in width size

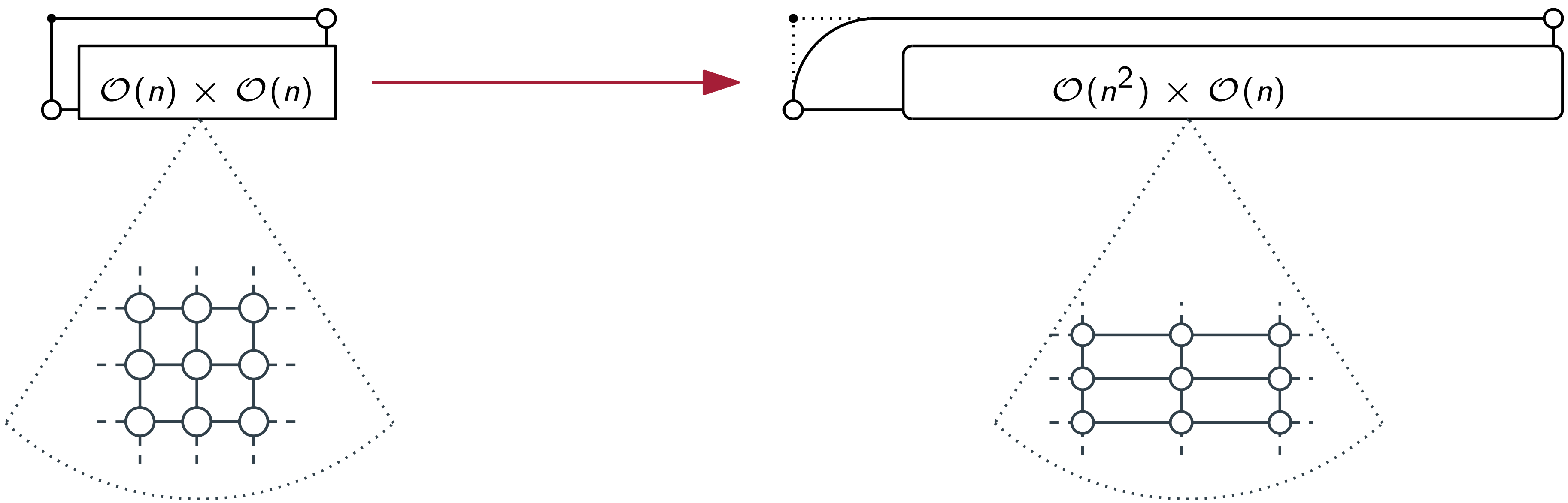


- worst case complexity increase: $k \to \left\lceil \frac{3}{2} k \right\rceil - 1$

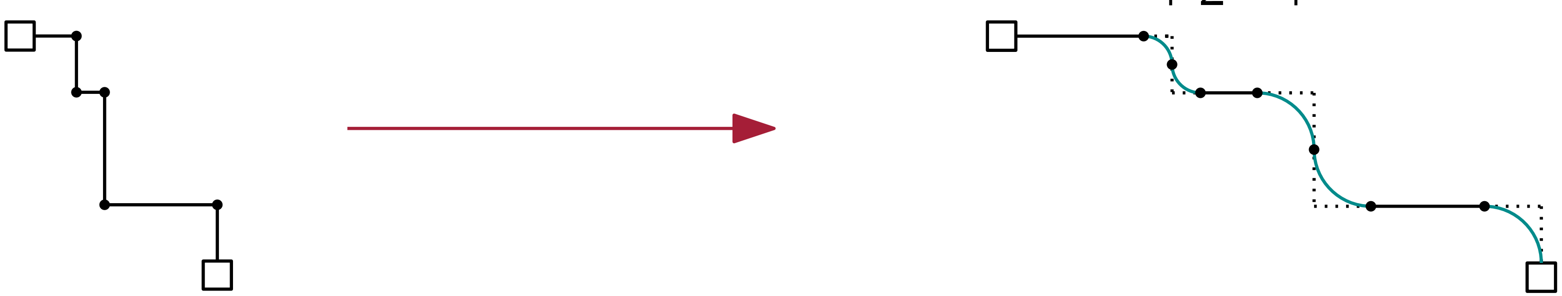


Stretching Technique (cnt.)

- linear runtime relative to the width of the drawing
- Does not alter the drawing drastically
- worst case area: quadratic in width size



- worst case complexity increase: $k \to \left\lceil \frac{3}{2} k \right\rceil - 1$



Smoothening of Kandinsky drawings

- Outline
  - **Stretching Technique**
    - Does the stretching technique smoothening work for Kandinsky drawings of arbitrary degree?
    - Area bounds?
    - Complexity bounds?

Smoothening of Kandinsky drawings

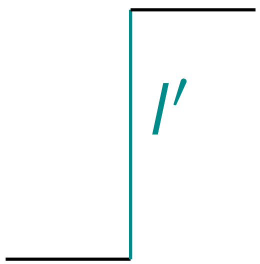
- Outline
  - **Stretching Technique**
    - Does the stretching technique smoothening work for Kandinsky drawings of arbitrary degree?
    - Area bounds?
    - Complexity bounds?
  - **Saving measures**
    - Area saving measures
    - Complexity saving measures

Kandinsky - Stretching

- **The stretching technique smoothening does work for Kandinsky drawings of arbitrary degree**
- Three szenarios for vertical segments to consider:

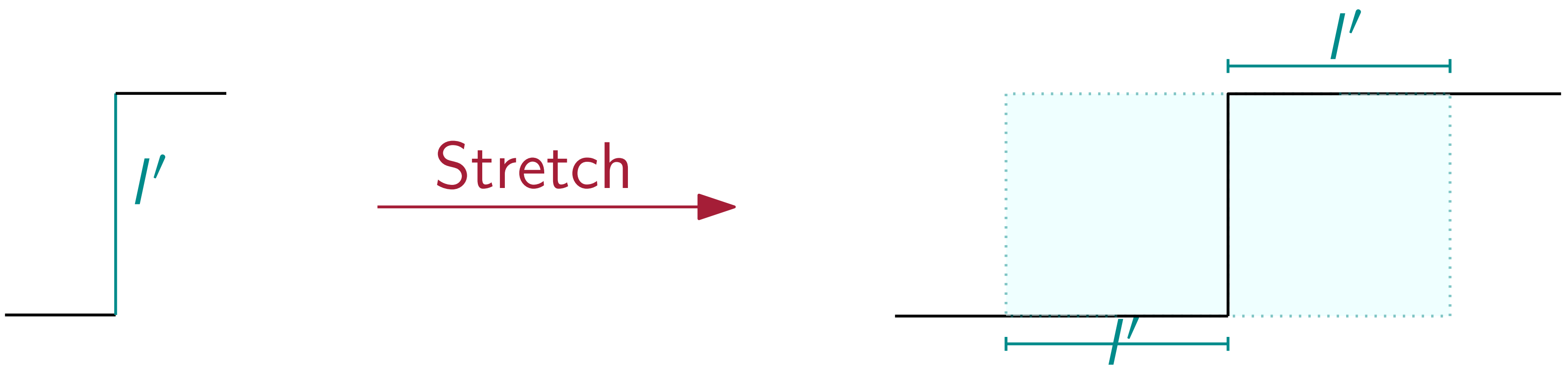
Kandinsky - Stretching

- **The stretching technique smoothening does work for Kandinsky drawings of arbitrary degree**
- Three szenarios for vertical segments to consider:
  - Vertical segment between two horizontal segments



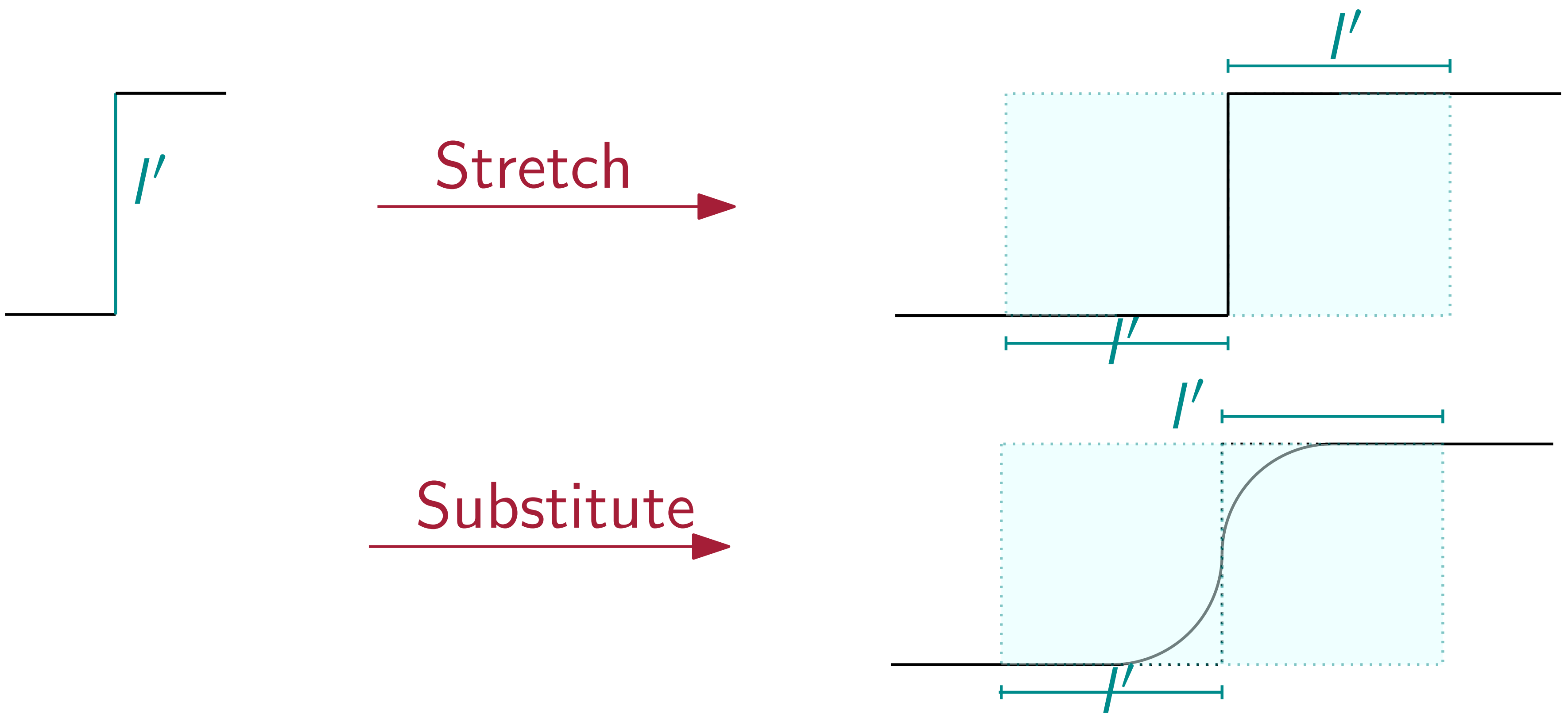
Kandinsky - Stretching

- **The stretching technique smoothening does work for Kandinsky drawings of arbitrary degree**
- Three szenarios for vertical segments to consider:
  - Vertical segment between two horizontal segments



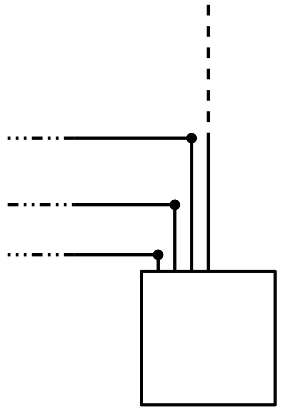
Kandinsky - Stretching

- **The stretching technique smoothening does work for Kandinsky drawings of arbitrary degree**
- Three szenarios for vertical segments to consider:
  - Vertical segment between two horizontal segments



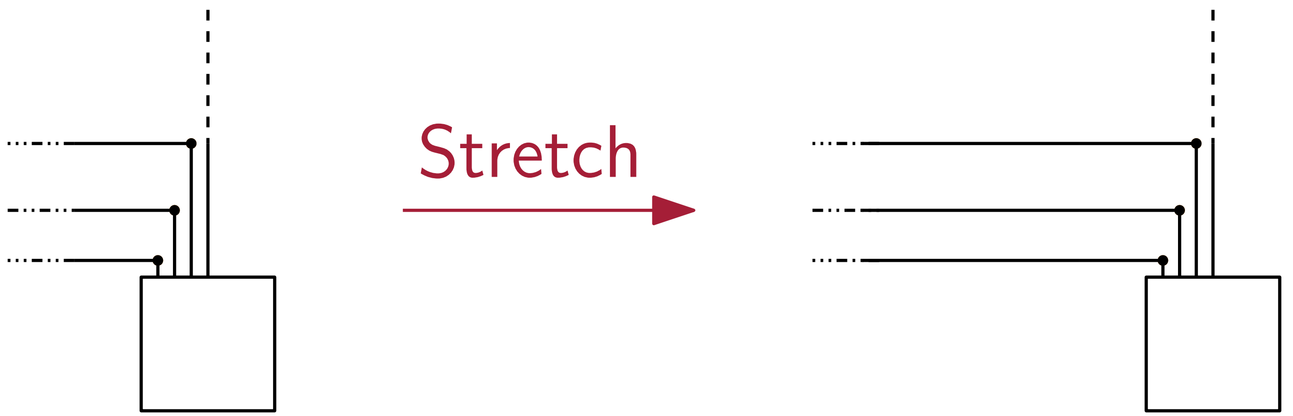
Kandinsky Stretching

- Multiple vertical segments adjacent to a vertex



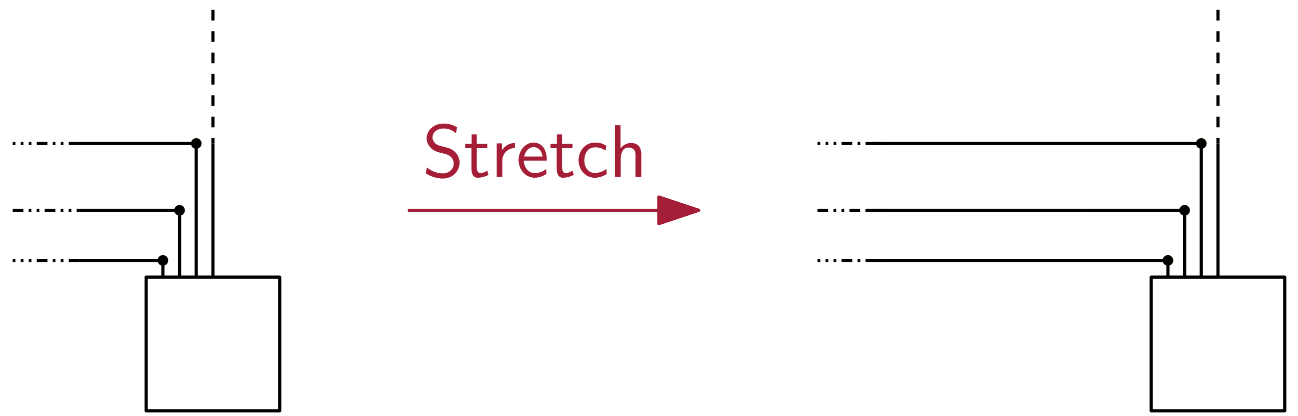
Kandinsky Stretching

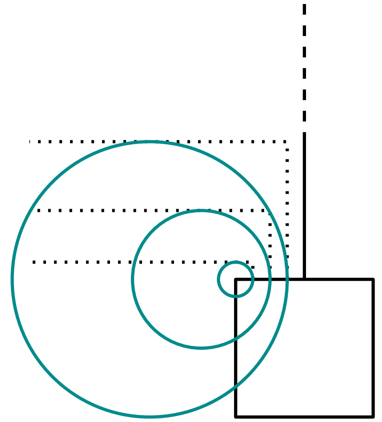
- Multiple vertical segments adjacent to a vertex



Kandinsky Stretching

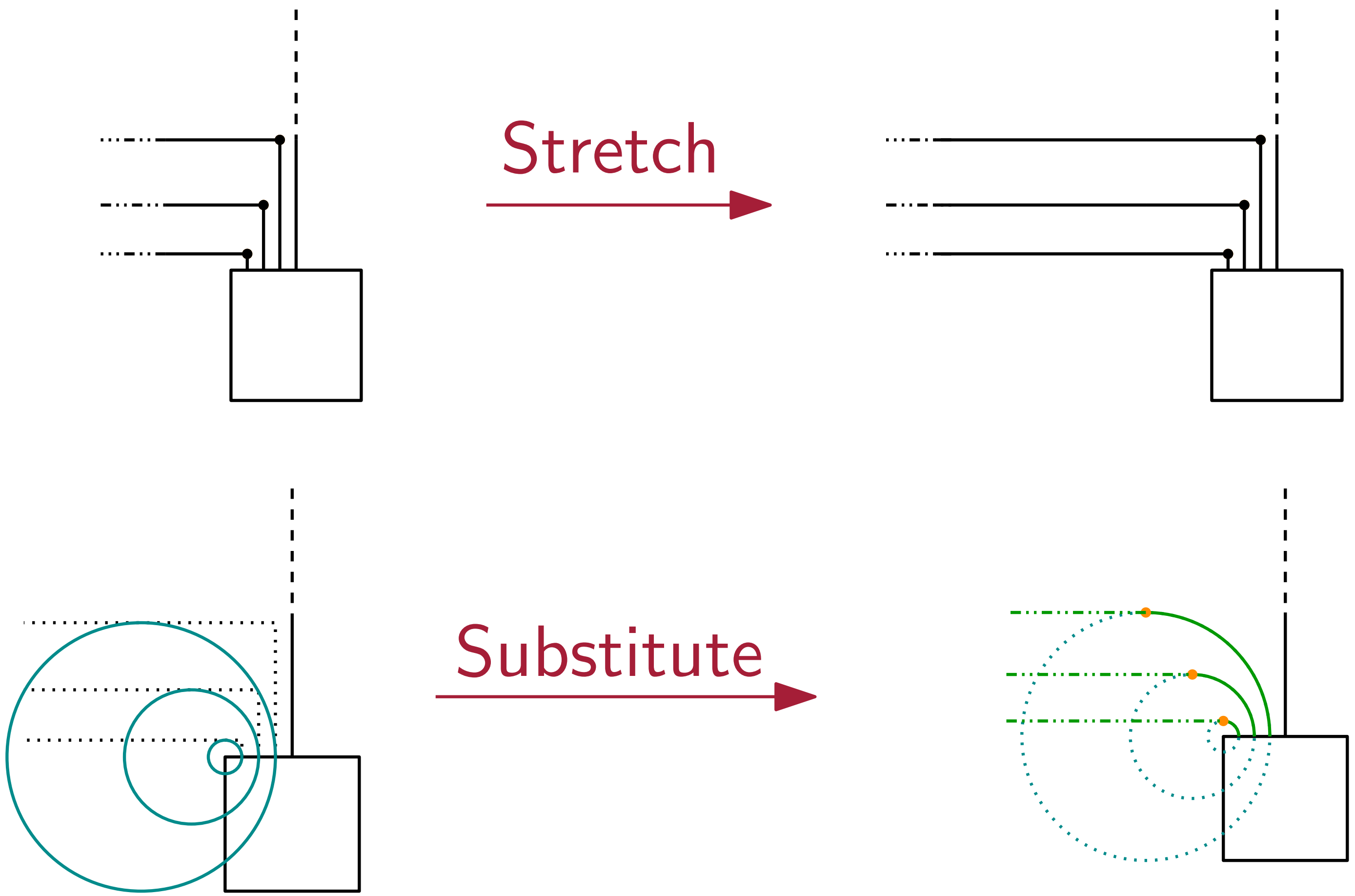
- Multiple vertical segments adjacent to a vertex





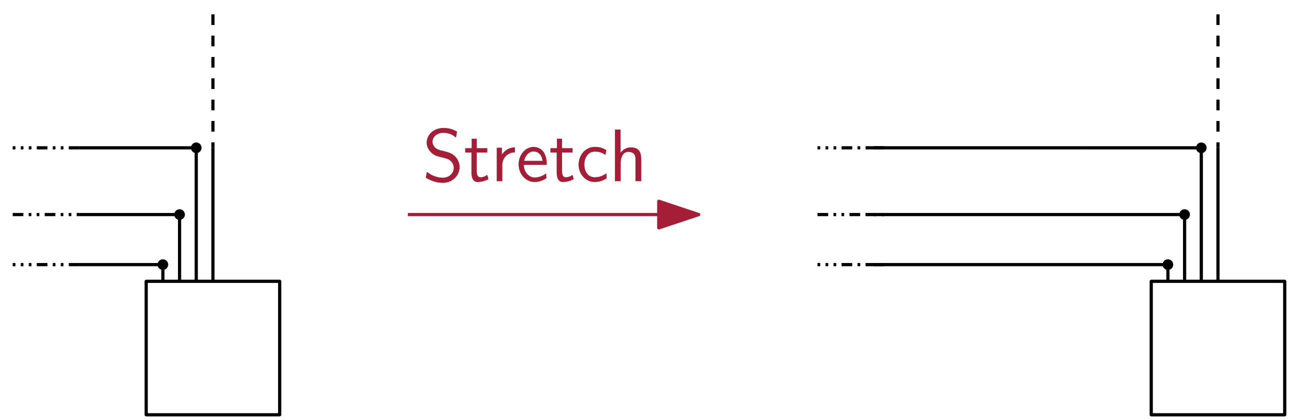
Kandinsky Stretching

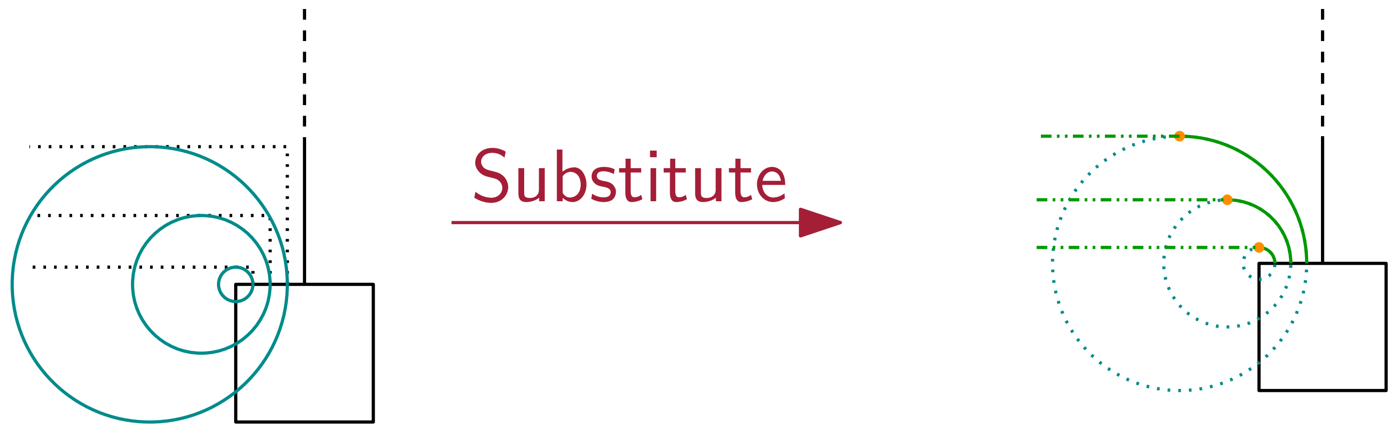
- Multiple vertical segments adjacent to a vertex



Kandinsky Stretching

- Multiple vertical segments adjacent to a vertex





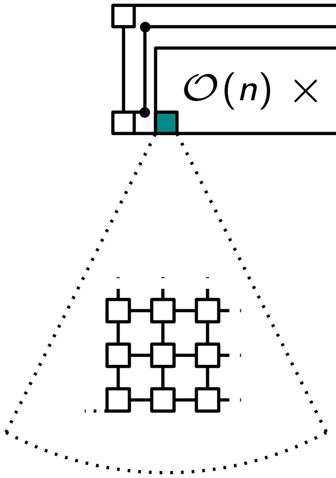
- Single vertical segment between two vertices - do nothing

Kandinsky - Stretching

- Length of every vertical segment l' is bounded by the length of the longest one
  - Therefore, the quarter circular arc substitution works

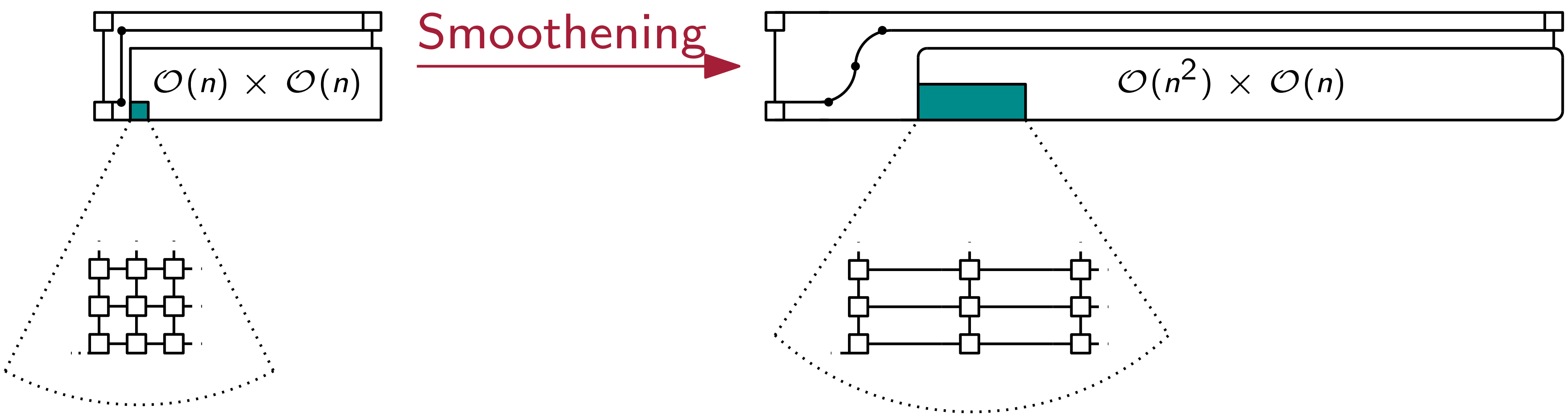
Kandinsky - Stretching

- Length of every vertical segment l' is bounded by the length of the longest one
  - Therefore, the quarter circular arc substitution works
  - Worst case area consumption: $\mathcal{O}(n^2) \times \mathcal{O}(n)$
    - Implied by the orthogonal 4-planar drawing worst case



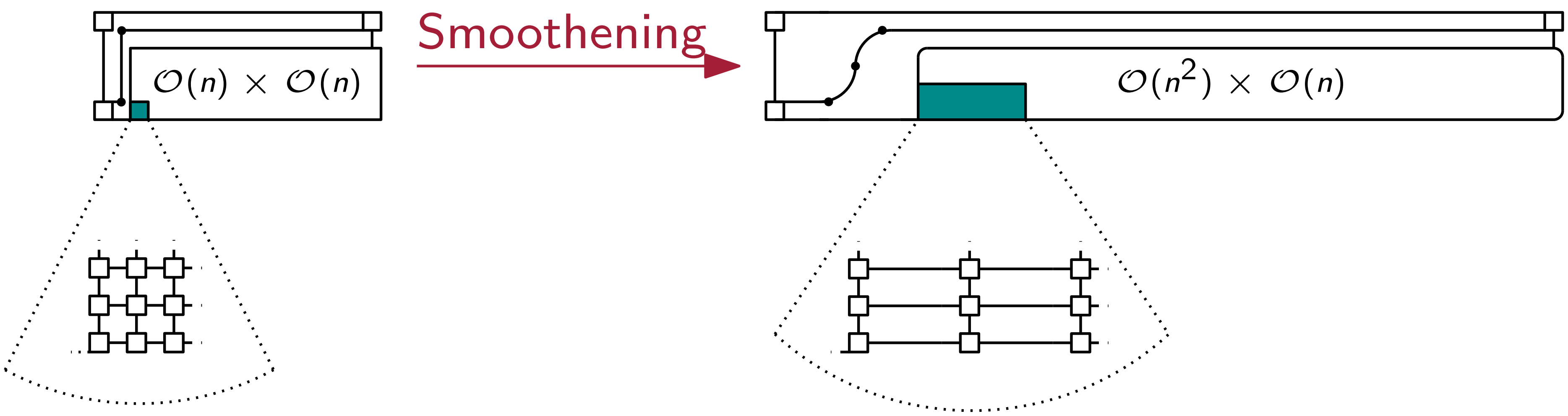
Kandinsky - Stretching

- Length of every vertical segment l' is bounded by the length of the longest one
  - Therefore, the quarter circular arc substitution works
  - Worst case area consumption: $\mathcal{O}(n^2) \times \mathcal{O}(n)$
    - Implied by the orthogonal 4-planar drawing worst case



Kandinsky - Stretching

- Length of every vertical segment l' is bounded by the length of the longest one
  - Therefore, the quarter circular arc substitution works
  - Worst case area consumption: $\mathcal{O}(n^2) \times \mathcal{O}(n)$
    - Implied by the orthogonal 4-planar drawing worst case



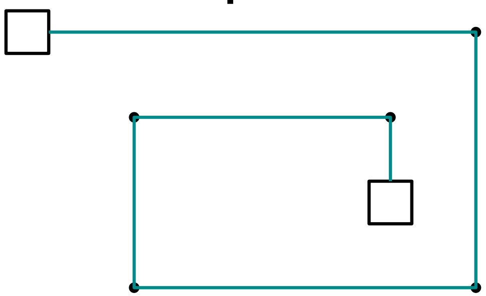
- Consider the complexity increase from three to four

Complexity Investigation

- How does the complexity of polyedges “behave“ by smoothening an orthogonal drawing?

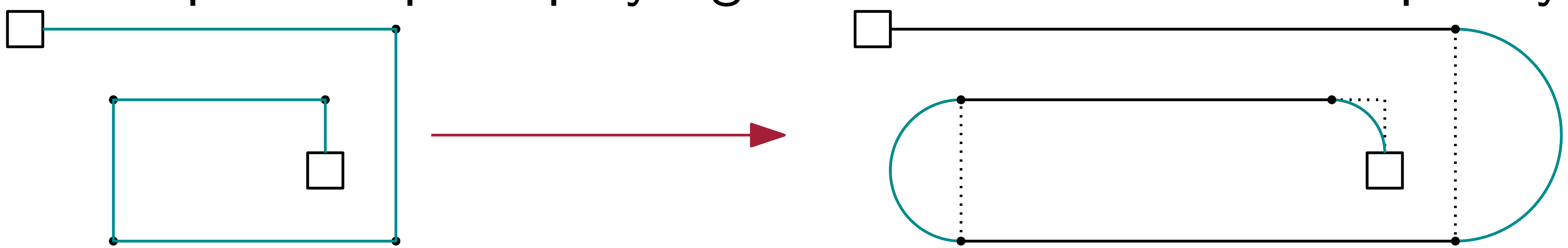
Complexity Investigation

- How does the complexity of polyedges "behave" by smoothening an orthogonal drawing?
- **Known results** [Bekos et al. 2013]
  - "spiral-shaped" polyedges do not increase in complexity



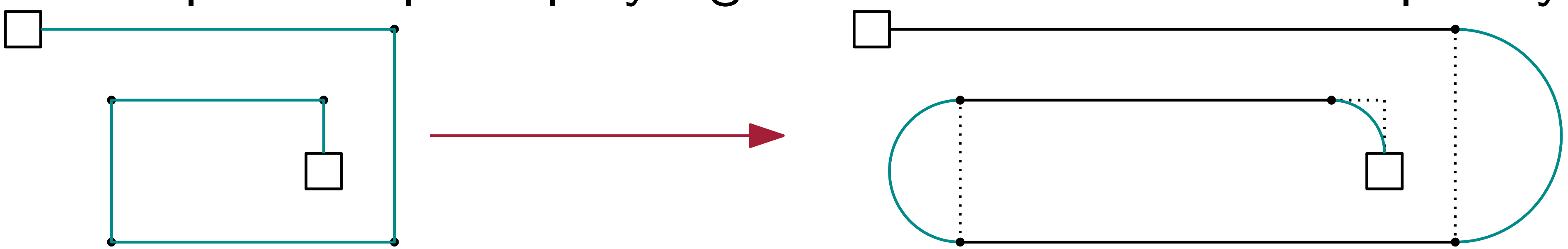
Complexity Investigation

- How does the complexity of polyedges “behave“ by smoothening an orthogonal drawing?
- **Known results** [Bekos et al. 2013]
  - “spiral-shaped“ polyedges do not increase in complexity

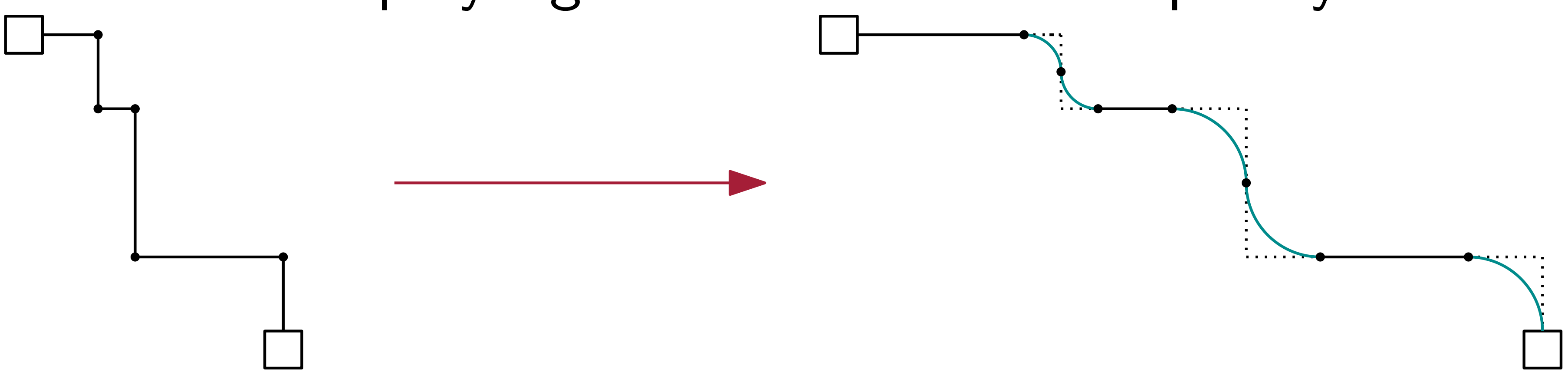


Complexity Investigation

- How does the complexity of polyedges “behave“ by smoothening an orthogonal drawing?
- **Known results** [Bekos et al. 2013]
  - “spiral-shaped“ polyedges do not increase in complexity



  - “staircase“ polyedges **do** increase in complexity

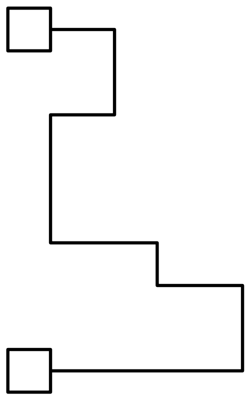


Defining “good“ and “bad“ parts

- Orthogonal polyedge e given
  - “Partition“ e:
    - *Fragment* $:=$ non-empty sequence of segments
    - *Fragmentation* $:=$ non-empty sequence of fragments

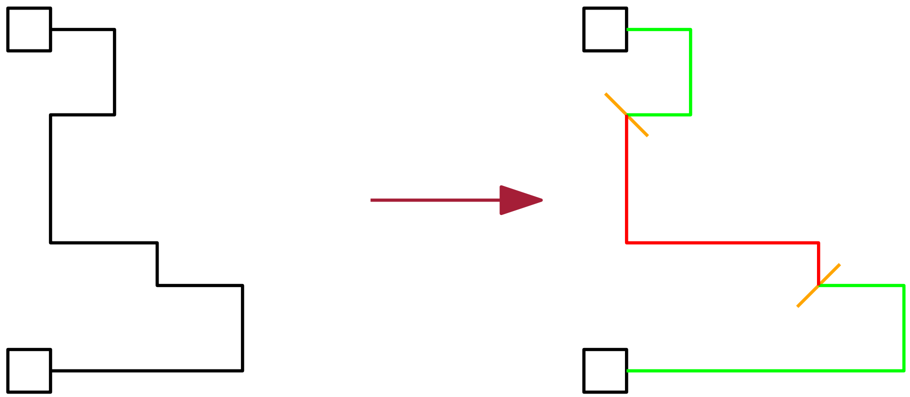
Defining “good“ and “bad“ parts

- Orthogonal polyedge e given
  - “Partition“ e:
    - *Fragment* := non-empty sequence of segments
    - *Fragmentation* := non-empty sequence of fragments



Defining “good“ and “bad“ parts

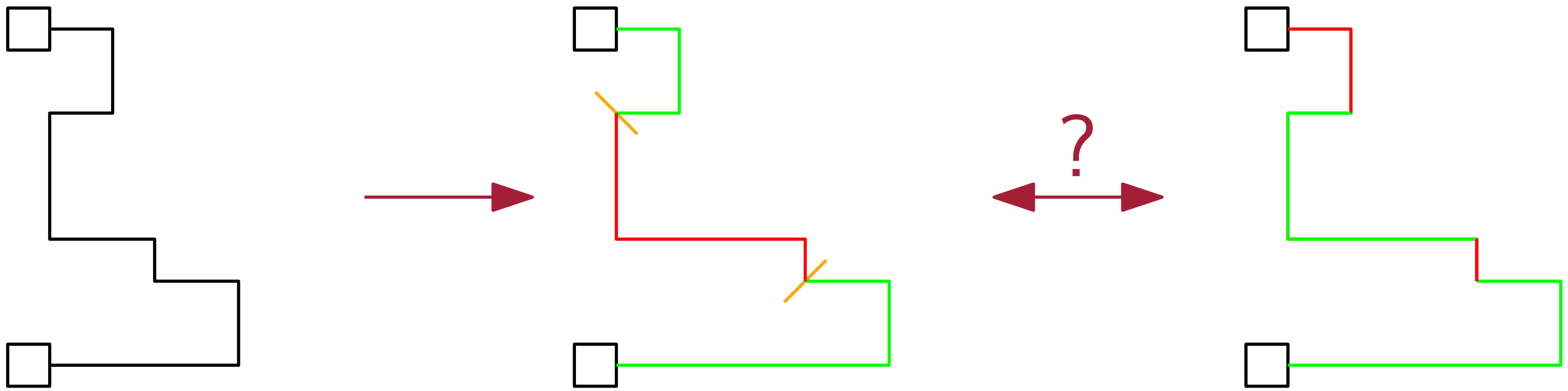
- Orthogonal polyedge e given
  - “Partition“ e:
    - *Fragment* $:=$ non-empty sequence of segments
    - *Fragmentation* $:=$ non-empty sequence of fragments



    - Good part $\triangleq$ spiral shapes $\triangleq$ uniform fragment
    - Bad part $\triangleq$ zig-zag shapes $\triangleq$ alternating fragment

Defining “good“ and “bad“ parts

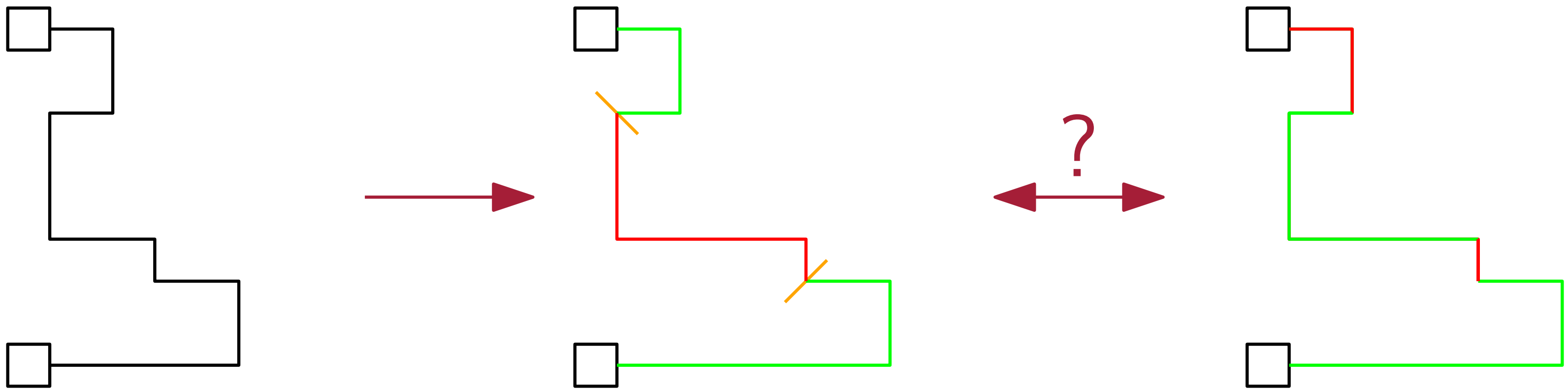
- Orthogonal polyedge e given
  - “Partition“ e:
    - *Fragment* := non-empty sequence of segments
    - *Fragmentation* := non-empty sequence of fragments



- Good part ≙ spiral shapes ≙ uniform fragment
- Bad part ≙ zig-zag shapes ≙ alternating fragment

Defining “good“ and “bad“ parts

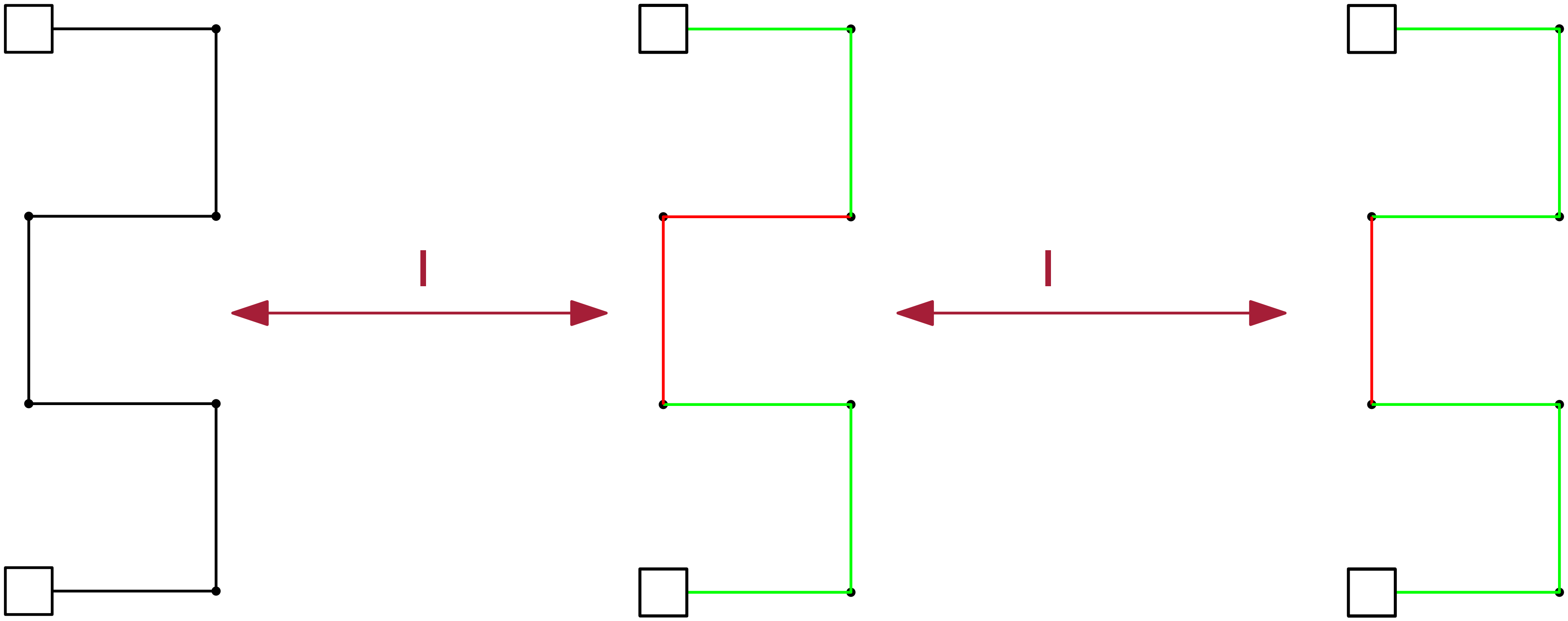
- Orthogonal polyedge e given
  - “Partition“ e:
    - *Fragment* := non-empty sequence of segments
    - *Fragmentation* := non-empty sequence of fragments



    - Good part $\triangleq$ spiral shapes $\triangleq$ uniform fragment
    - Bad part $\triangleq$ zig-zag shapes $\triangleq$ alternating fragment
  - What fragmentation is the “most accurate“?
  - How can we utilize the partitioning for complexity investigaion?

Fragmentation properties

- Multiple valid fragmentations can describe one polyedge

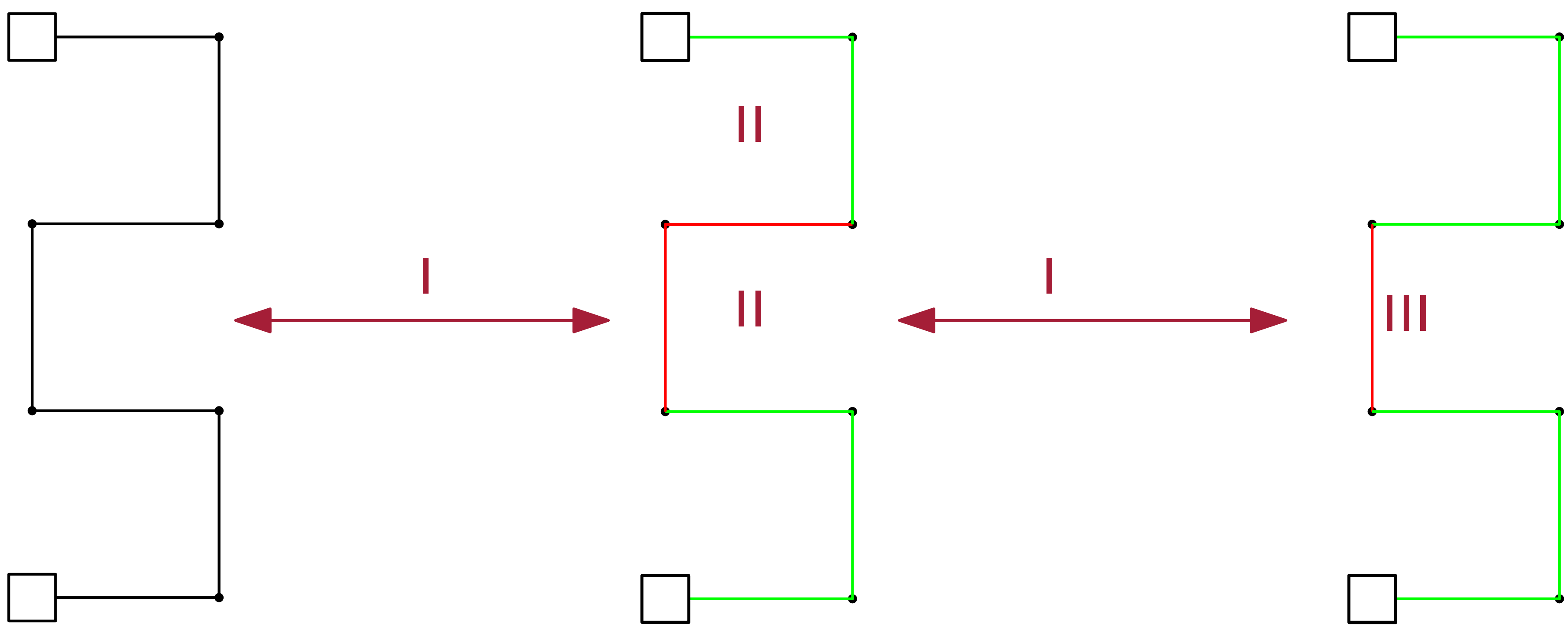


Fragmentation properties

I Multiple valid fragmentations can describe one polyedge

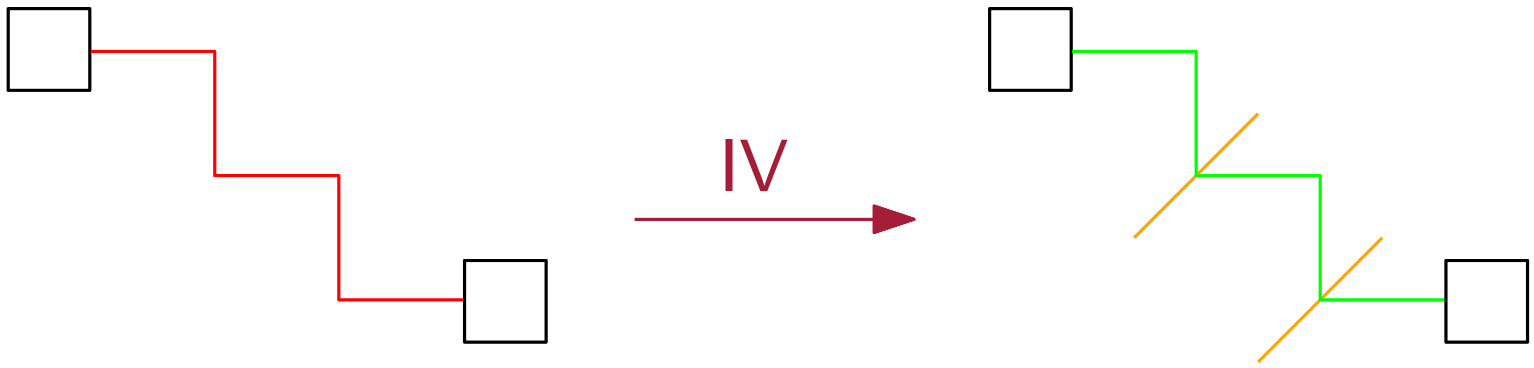
II Fragments of length up to 2 are both uniform *and* alternating

III Two fragments are *incompatible*
$\Leftrightarrow$ they are not further mergable



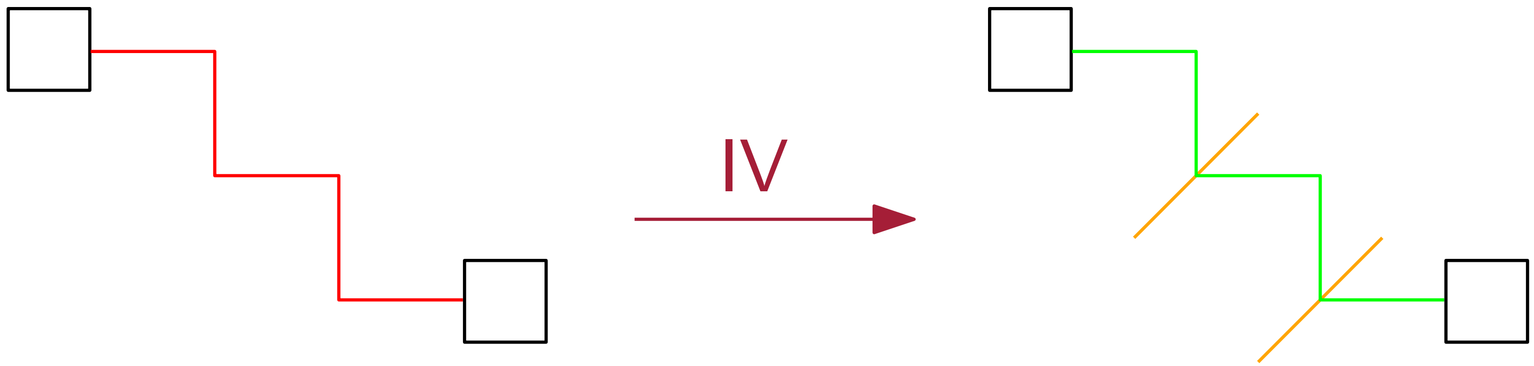
Fragmentation Properties

IV *Alternating* fragments can be decomposed in uniform fragments of length at most two

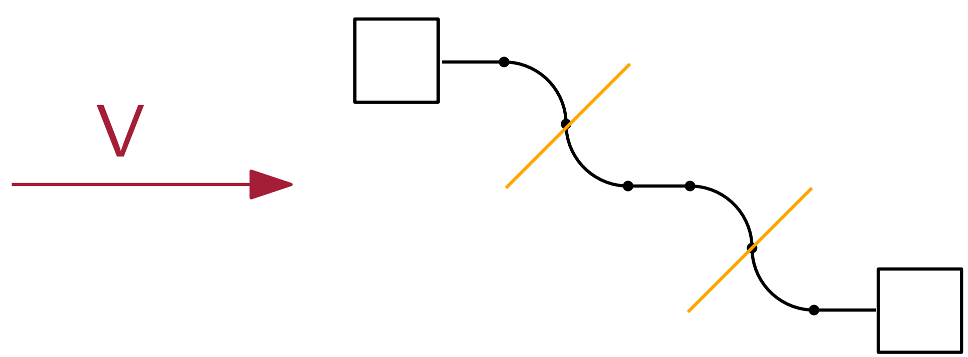


Fragmentation Properties

IV *Alternating* fragments can be decomposed in uniform fragments of length at most two



V Incompatible fragments increase the complexity by 1 in the smoothened drawing



Complexity Investigation

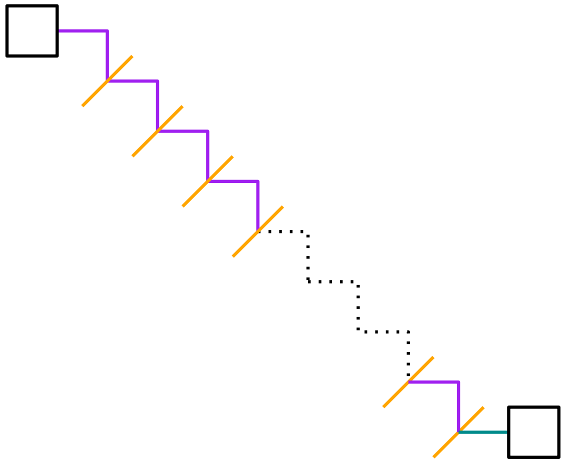
- Uniform-only fragmentations describe “smoothening behaviour“ accurately, linear runtime algorithm

Complexity Investigation

- Uniform-only fragmentations describe “smoothening behaviour“ accurately, linear runtime algorithm
- $k \mathrel{\hat{=}}$ complexity of input orthogonal polyedge
  - $1 \leq$ fragmentation length $\leq \left\lceil \frac{k}{2} \right\rceil$

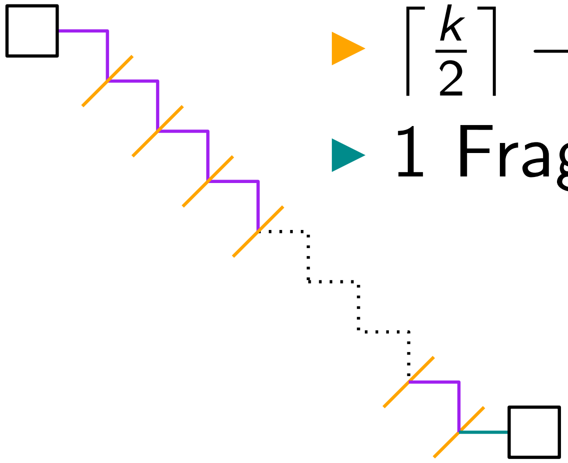
Complexity Investigation

- Uniform-only fragmentations describe “smoothening behaviour“ accurately, linear runtime algorithm
- $k \triangleq$ complexity of input orthogonal polyedge
  - $1 \leq$ fragmentation length $\leq \lceil \frac{k}{2} \rceil$
  - worst case: staircase polyedge
    - “chopped“into fragments of length at most 2
    - k odd:
      - fragmentation length: $\lceil \frac{k}{2} \rceil$



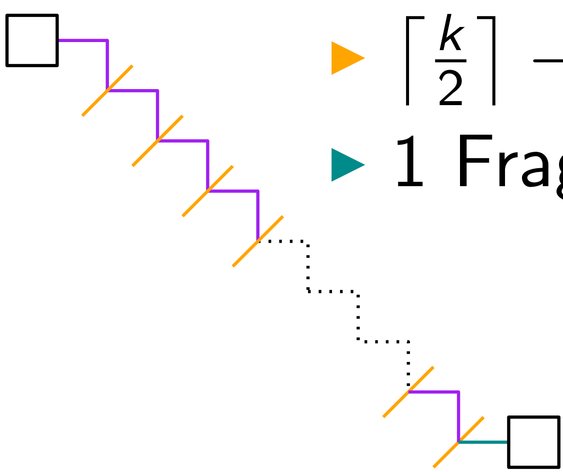
Complexity Investigation

- Uniform-only fragmentations describe “smoothening behaviour“ accurately, linear runtime algorithm
- $k \triangleq$ complexity of input orthogonal polyedge
  - $1 \leq$ fragmentation length $\leq \left\lceil \frac{k}{2} \right\rceil$
  - worst case: staircase polyedge
    - “chopped“into fragments of length at most 2
    - k odd:
      - fragmentation length: $\left\lceil \frac{k}{2} \right\rceil$
      - $\left\lceil \frac{k}{2} \right\rceil - 1$ fragments of length 2
      - $\left\lceil \frac{k}{2} \right\rceil - 1$ incompatible fragment transitions
      - 1 Fragment of length 1



Complexity Investigation

- Uniform-only fragmentations describe “smoothening behaviour“ accurately, linear runtime algorithm
- $k \triangleq$ complexity of input orthogonal polyedge
  - $1 \leq$ fragmentation length $\leq \left\lceil \frac{k}{2} \right\rceil$
  - worst case: staircase polyedge
    - “chopped“into fragments of length at most 2
    - k odd:
      - fragmentation length: $\left\lceil \frac{k}{2} \right\rceil$
      - $\left\lceil \frac{k}{2} \right\rceil - 1$ fragments of length 2
      - $\left\lceil \frac{k}{2} \right\rceil - 1$ incompatible fragment transitions
      - 1 Fragment of length 1



- $\left(\sum_{i=1}^{\left\lceil \frac{k}{2} \right\rceil - 1} 2 \right) + 1 + \left\lceil \frac{k}{2} \right\rceil - 1 = \left\lceil \frac{3k}{2} \right\rceil - 1$

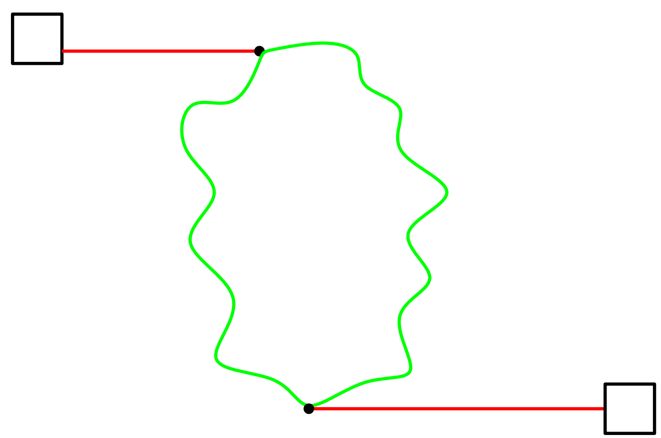
(complexity upper bound)

What about Kandinsky drawings?

- Upper bound holds for low complexity polyedges
- The upper bound can be further improved for polyedges with high complexity

What about Kandinsky drawings?

- Upper bound holds for low complexity polyedges
- The upper bound can be further improved for polyedges with high complexity
- Large polyedges in Kandinsky may inherit two bends which are not deducable
- Polyedge spiral-shaped in-between those bends



- In this case: complexity increase from k to $k+2$

Overall results

- **Kandinsky drawings can be smoothened with the stretching technique**
  - worst case area consumption: quadratic in width size
  - height stays unaltered

Overall results

- **Kandinsky drawings can be smoothened with the stretching technique**
  - worst case area consumption: quadratic in width size
  - height stays unaltered
  - Edge complexity increase:

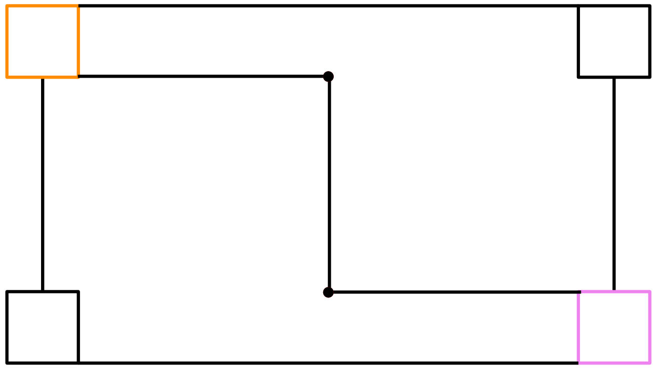
Original Kandinsky k	Smoothened Kandinsky k'
≤ 5	$\lceil \frac{3k}{2} \rceil - 1$
≥ 5	$k + 2$

Saving measures - Complexity

- Port reassignment

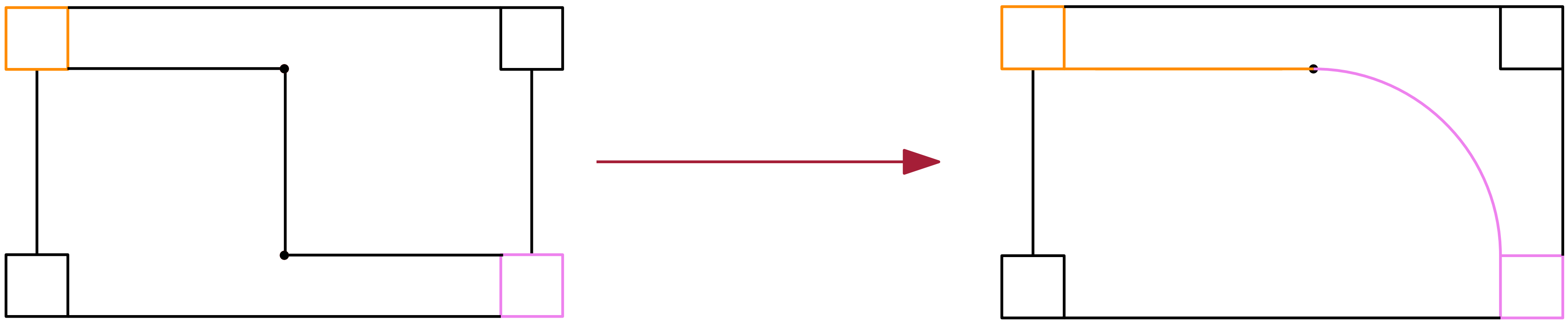
Saving measures - Complexity

- Port reassignment
  - May reduce the edge complexity by one



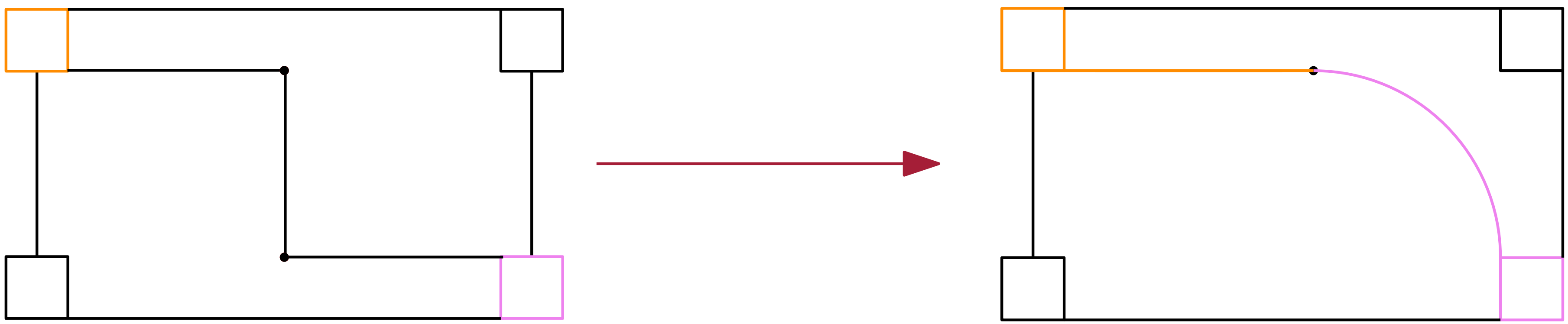
Saving measures - Complexity

- Port reassignment
  - May reduce the edge complexity by one

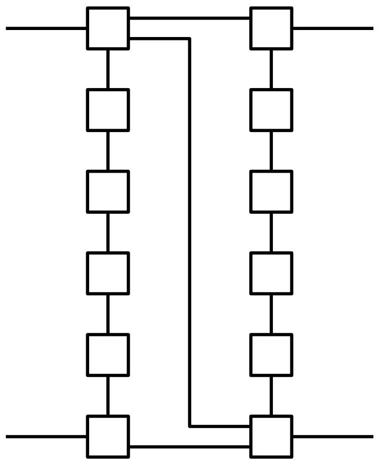


Saving measures - Complexity

- Port reassignment
  - May reduce the edge complexity by one

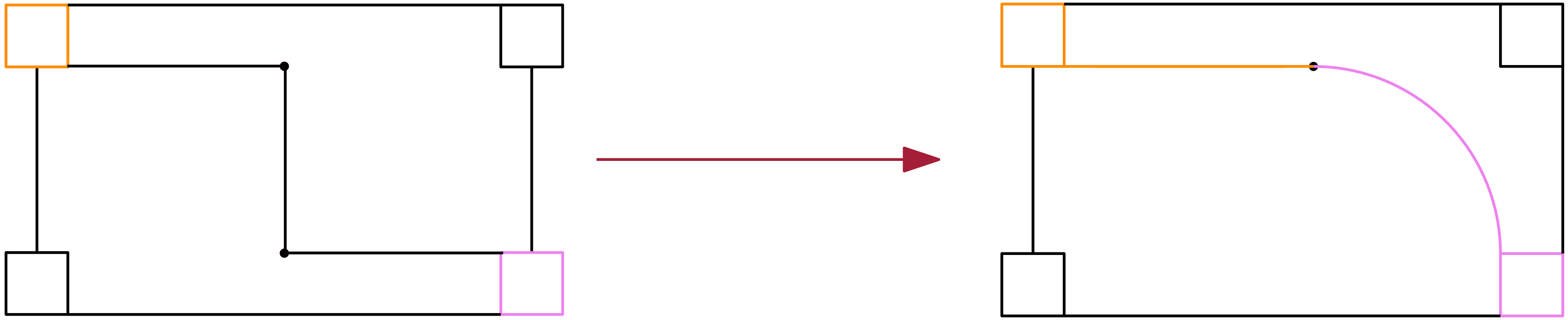


  - However, it does not always work

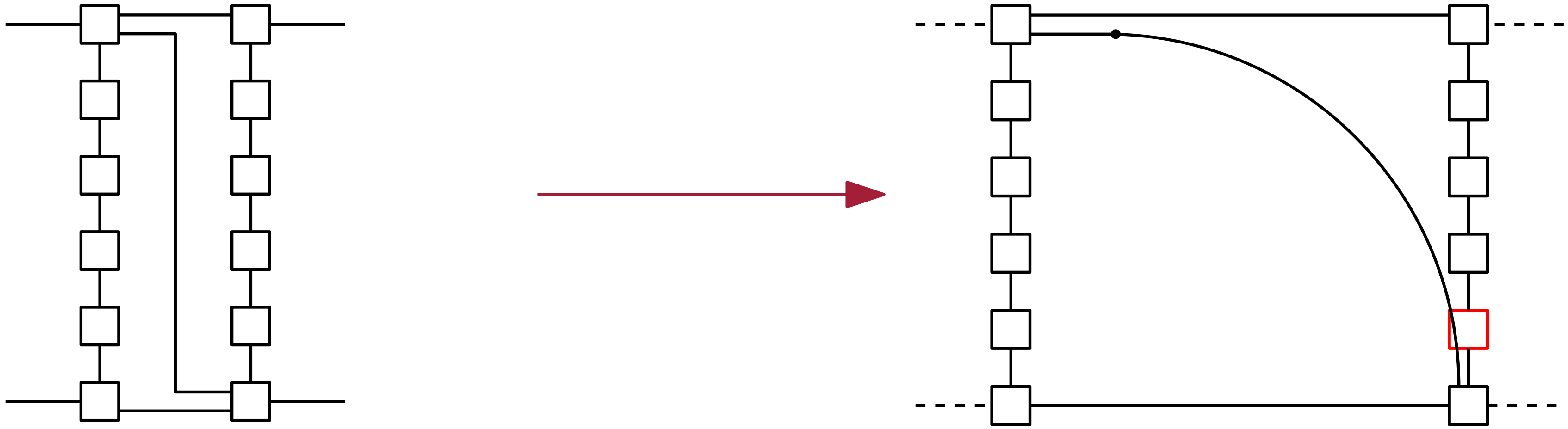


Saving measures - Complexity

- Port reassignment
  - May reduce the edge complexity by one

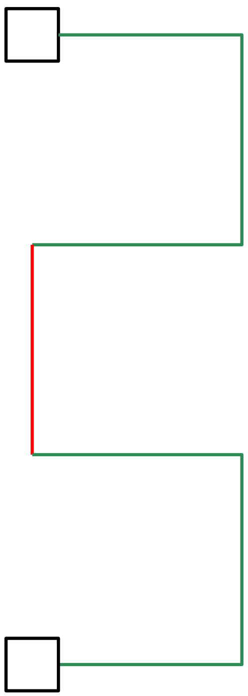


  - However, it does not always work



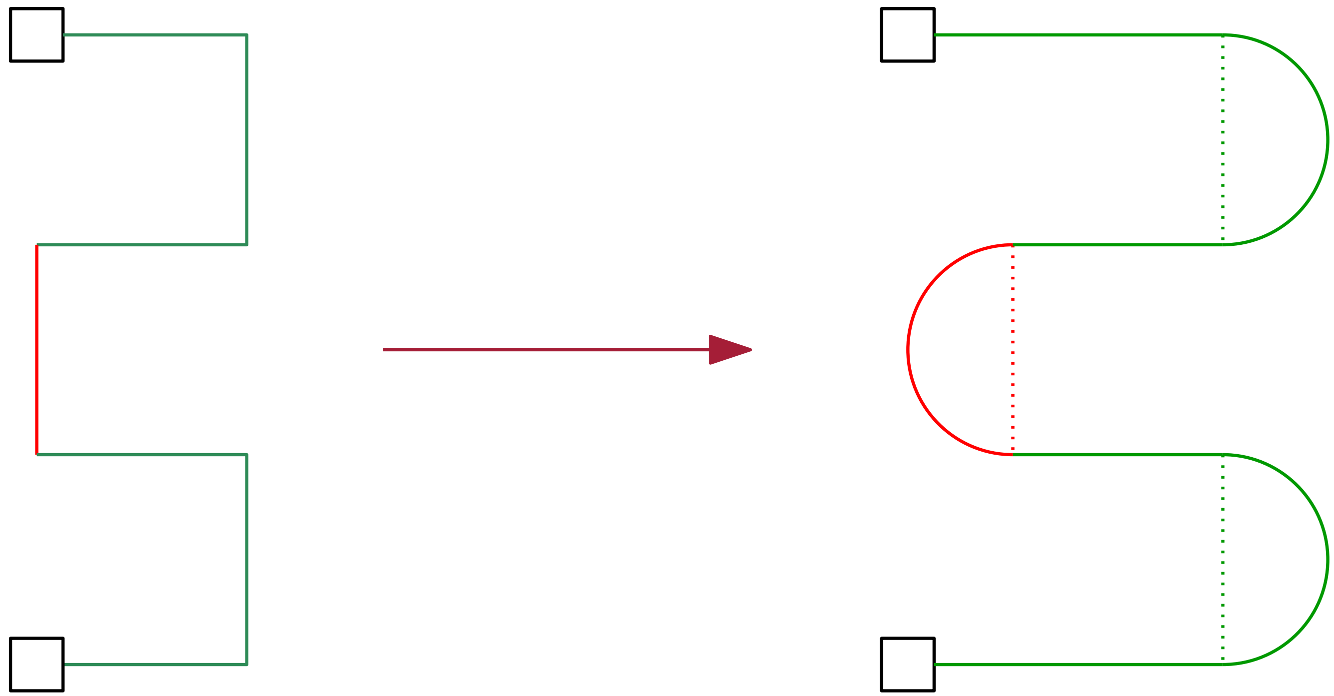
Saving measures - Complexity

- Using the fragmentation
  - If an alternating fragment of length one inherits a vertical fragment, the complexity does not increase



Saving measures - Complexity

- Using the fragmentation
  - If an alternating fragment of length one inherits a vertical fragment, the complexity does not increase



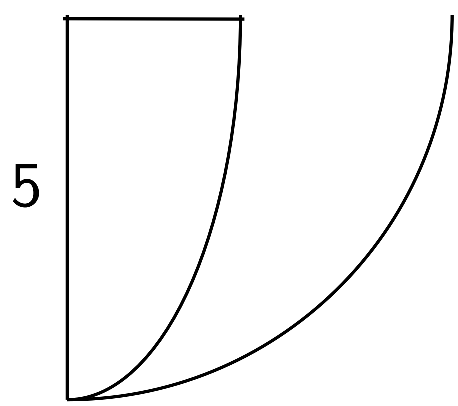
Saving measures - Area

Saving measures - Area

- Stretching technique
  - Stretch by only the square root of the longest vertical segment
  - Worst case area consumption: $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$
  - Substitute quarter circular arcs with:

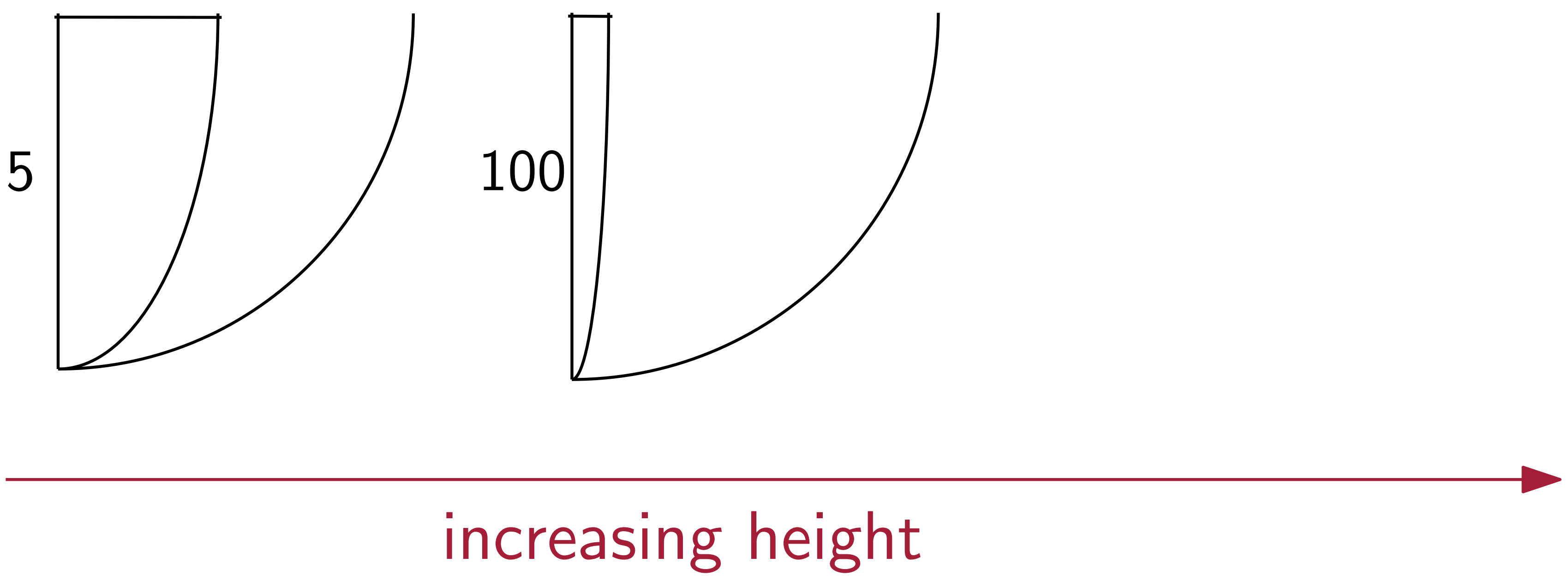
Saving measures - Area

- Stretching technique
  - Stretch by only the square root of the longest vertical segment
  - Worst case area consumption: $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$
  - Substitute quarter circular arcs with:
    - Ellipse arcs



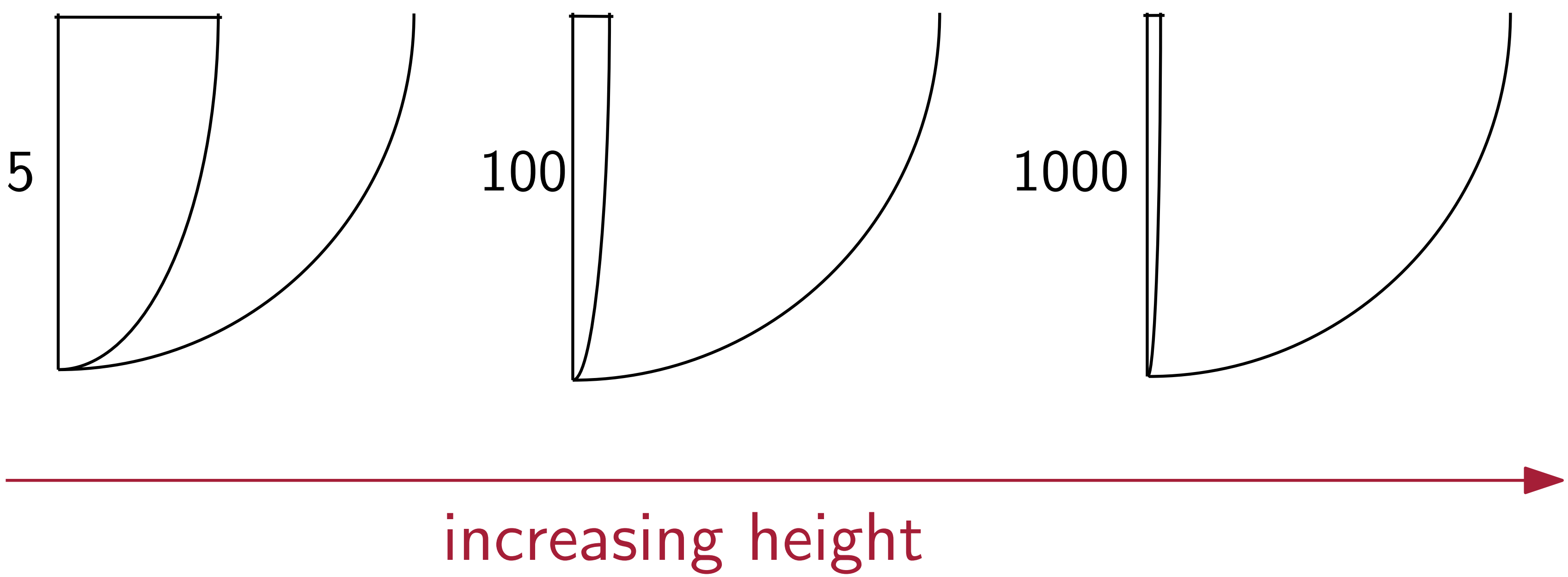
Saving measures - Area

- Stretching technique
  - Stretch by only the square root of the longest vertical segment
  - Worst case area consumption: $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$
  - Substitute quarter circular arcs with:
    - Ellipse arcs
    - Low readability for high values



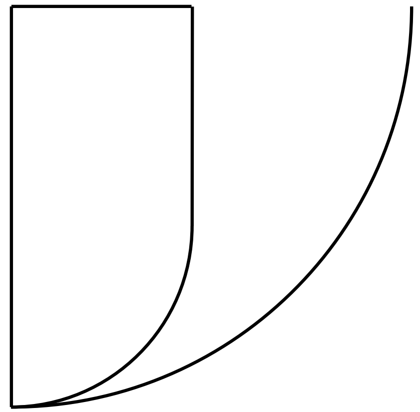
Saving measures - Area

- Stretching technique
  - Stretch by only the square root of the longest vertical segment
  - Worst case area consumption: $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$
  - Substitute quarter circular arcs with:
    - Ellipse arcs
    - Low readability for high values



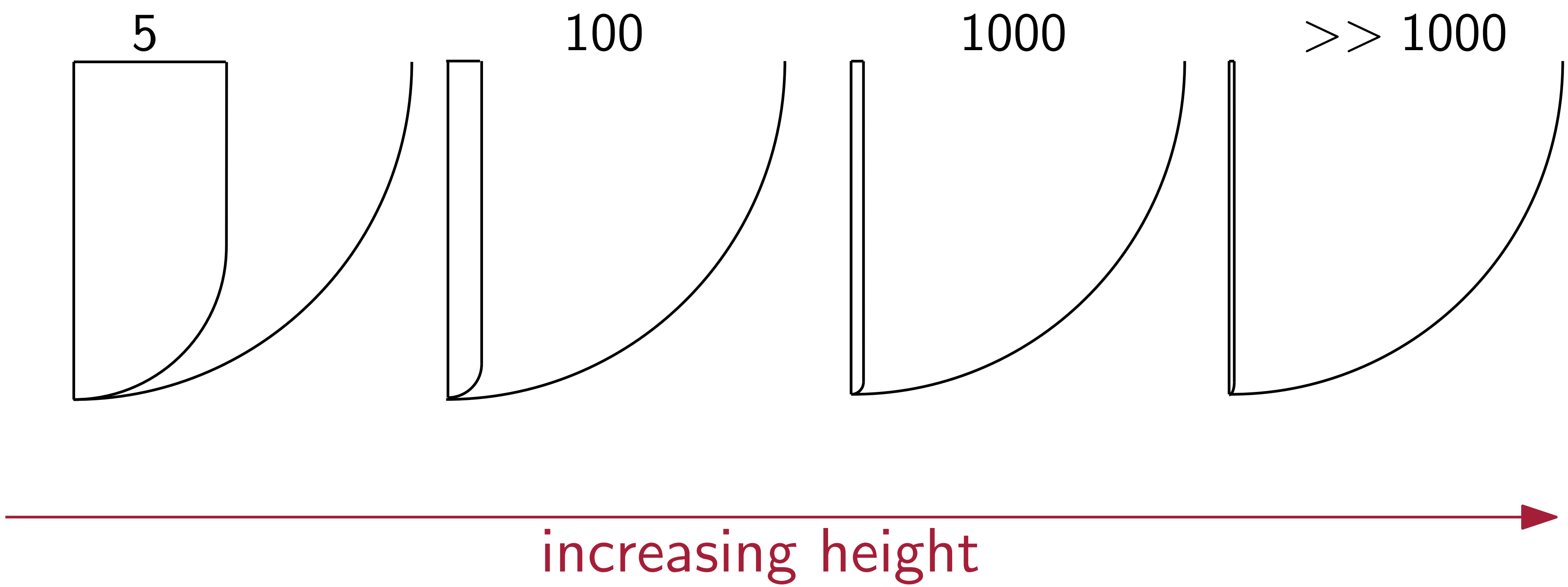
Saving measures - Area

- Stretching technique
  - Stretch by only the square root of the longest vertical segment
  - Worst case area consumption: $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$
  - Substitute quarter circular arcs with:
    - Smaller quarter circular arc $+$ 1 vertical segment



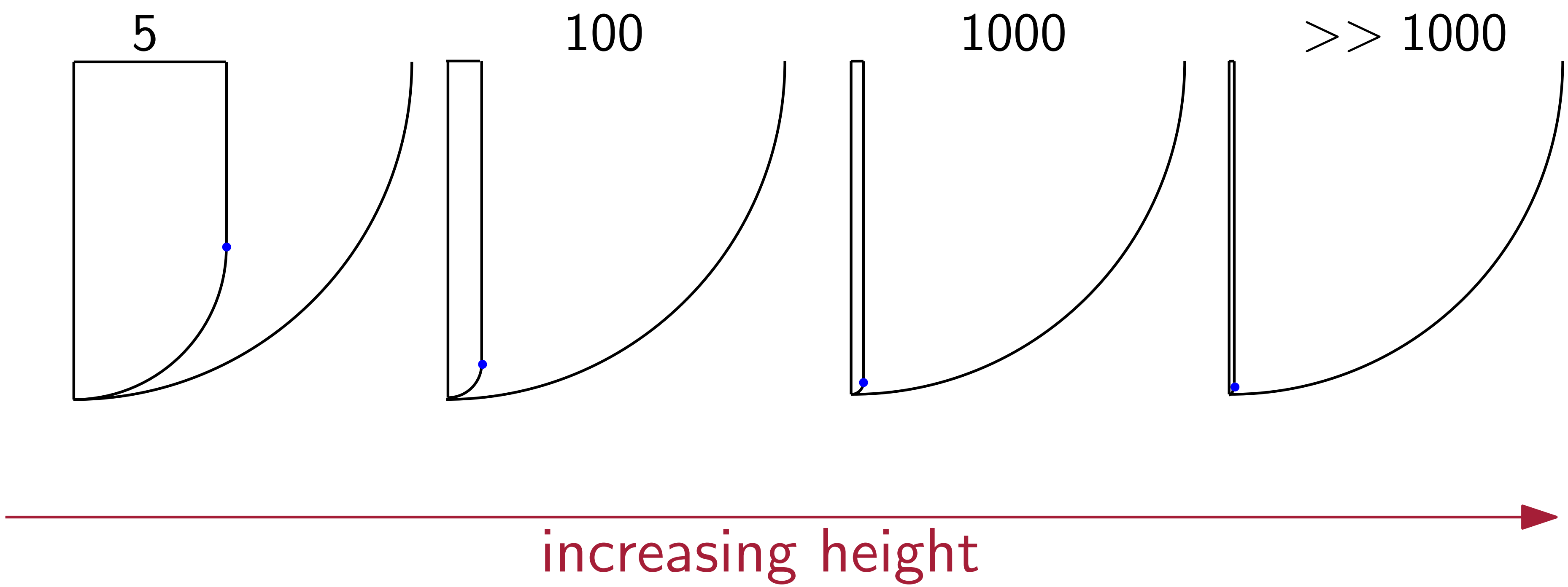
Saving measures - Area

- Stretching technique
  - Stretch by only the square root of the longest vertical segment
  - Worst case area consumption: $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$
  - Substitute quarter circular arcs with:
    - Smaller quarter circular arc $+$ 1 vertical segment



Saving measures - Area

- Stretching technique
  - Stretch by only the square root of the longest vertical segment
  - Worst case area consumption: $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$
  - Substitute quarter circular arcs with:
    - Smaller quarter circular arc $+$ 1 vertical segment
      - Higher readability, but high complexity increase



Area & complexity savings combined

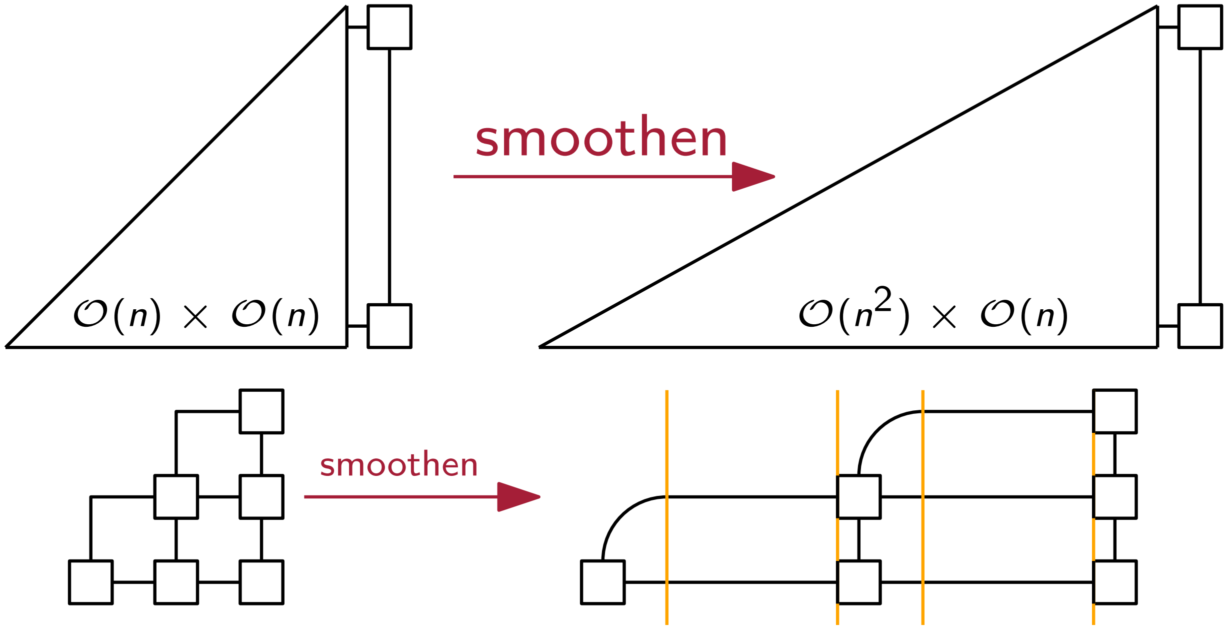
- Saving plane sweep

Area & complexity savings combined

- Saving plane sweep
  - If sweep line only crosses horizontal segments, look for redundancy

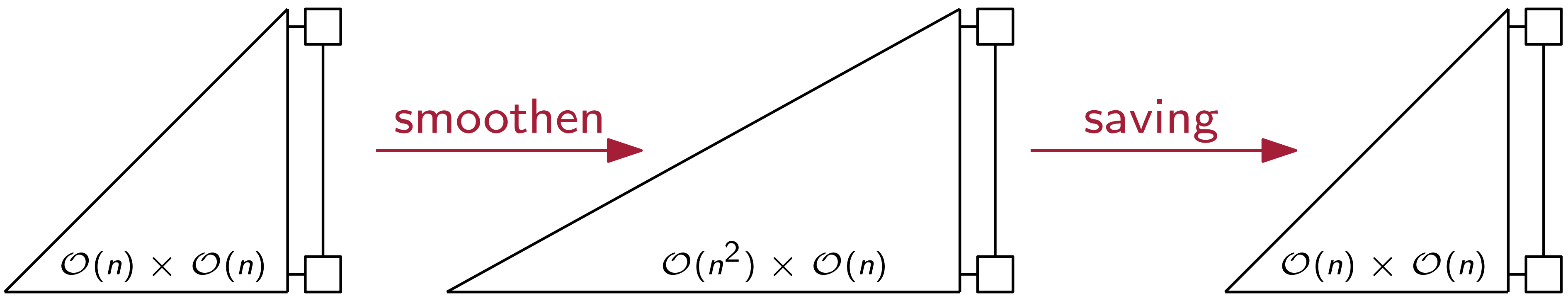
Area & complexity savings combined

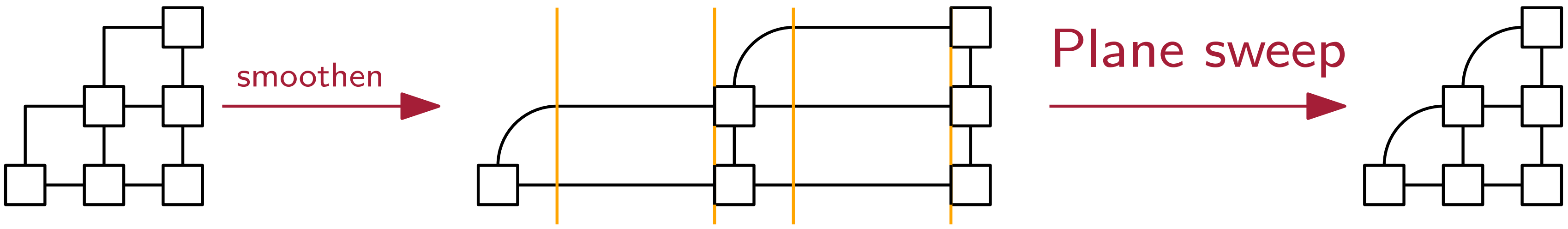
- Saving plane sweep
  - If sweep line only crosses horizontal segments, look for redundancy
  - Reduces width up to its square root and save segments



Area & complexity savings combined

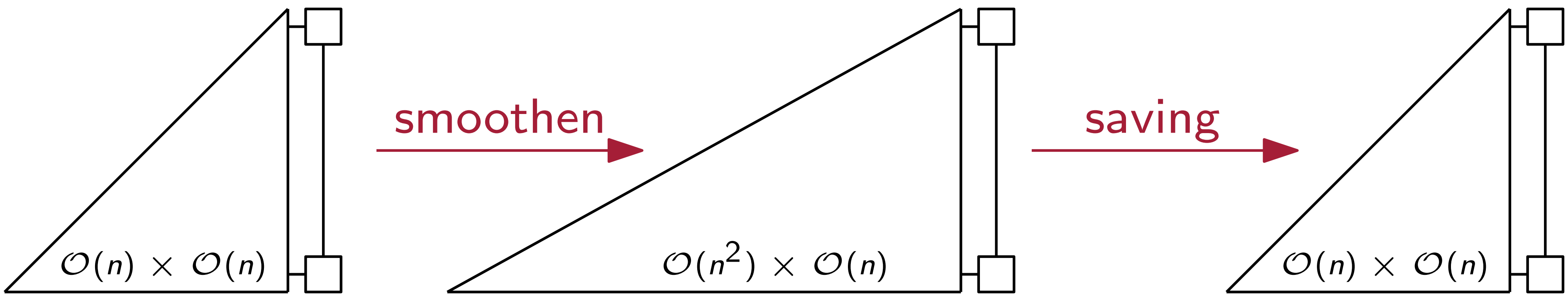
- Saving plane sweep
  - If sweep line only crosses horizontal segments, look for redundancy
  - Reduces width up to its square root and save segments

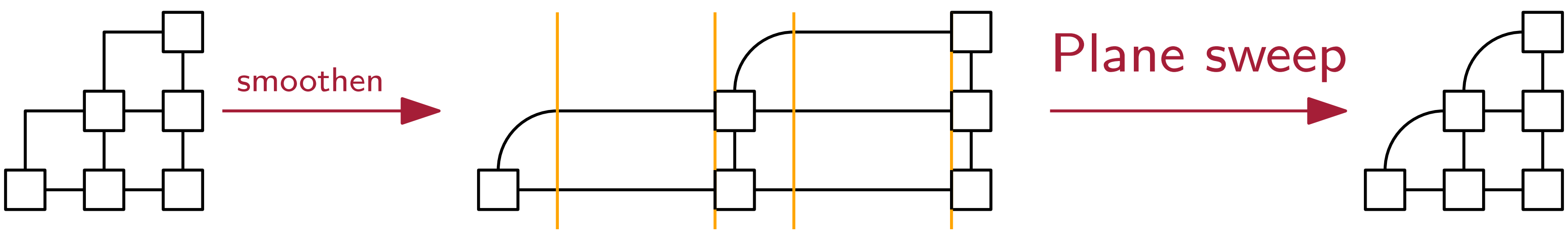




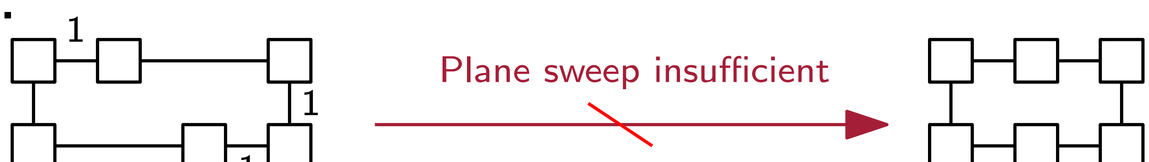
Area & complexity savings combined

- Saving plane sweep
  - If sweep line only crosses horizontal segments, look for redundancy
  - Reduces width up to its square root and save segments





- But...



4M Algorithm - Moving

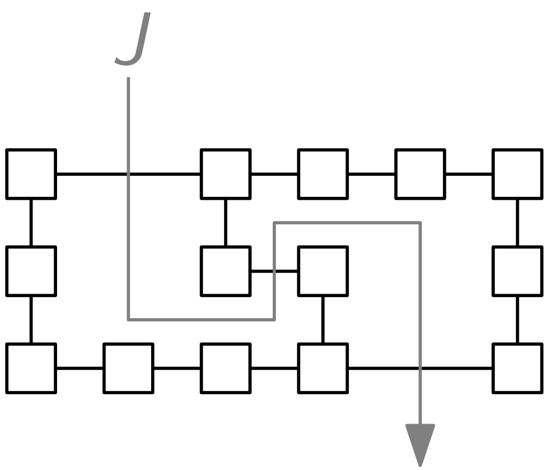
- **Area saving algorithm for orthogonal drawings** [Fößmeier et al. 1998]
- Finds a directed path through horizontal line segments

4M Algorithm - Moving

- **Area saving algorithm for orthogonal drawings** [Fößmeier et al. 1998]
- Finds a directed path through horizontal line segments
  - downwards: reduction by one unit length
  - upwards: elongation by one unit length
  - repeat until one of the segments achieved unit length

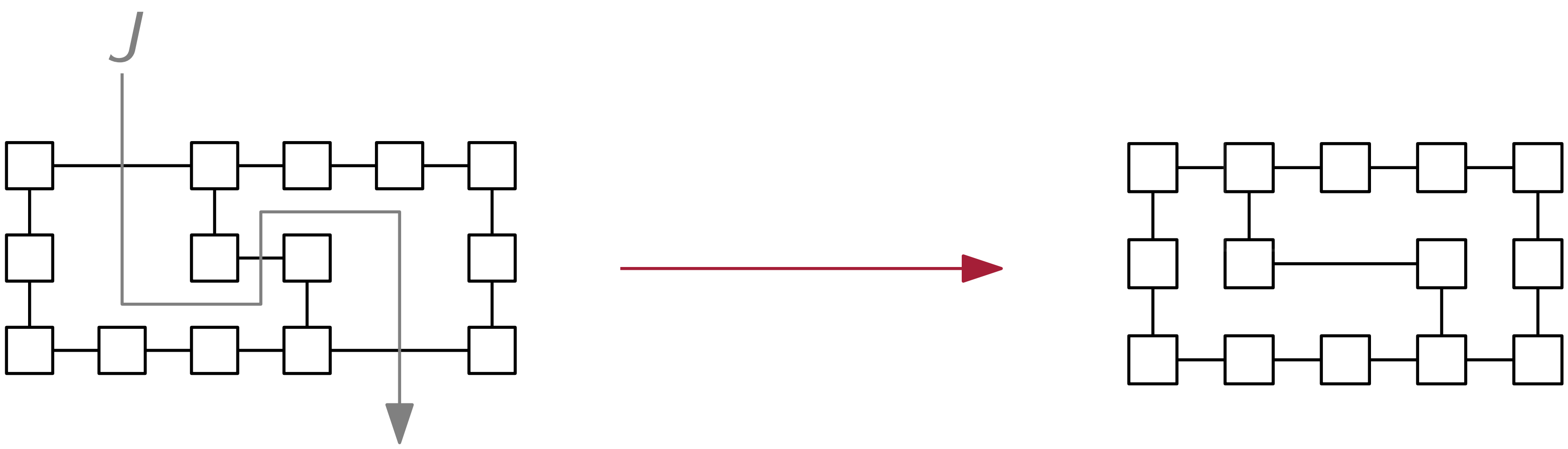
4M Algorithm - Moving

- **Area saving algorithm for orthogonal drawings** [Fößmeier et al. 1998]
- Finds a directed path through horizontal line segments
  - downwards: reduction by one unit length
  - upwards: elongation by one unit length
  - repeat until one of the segments achieved unit length



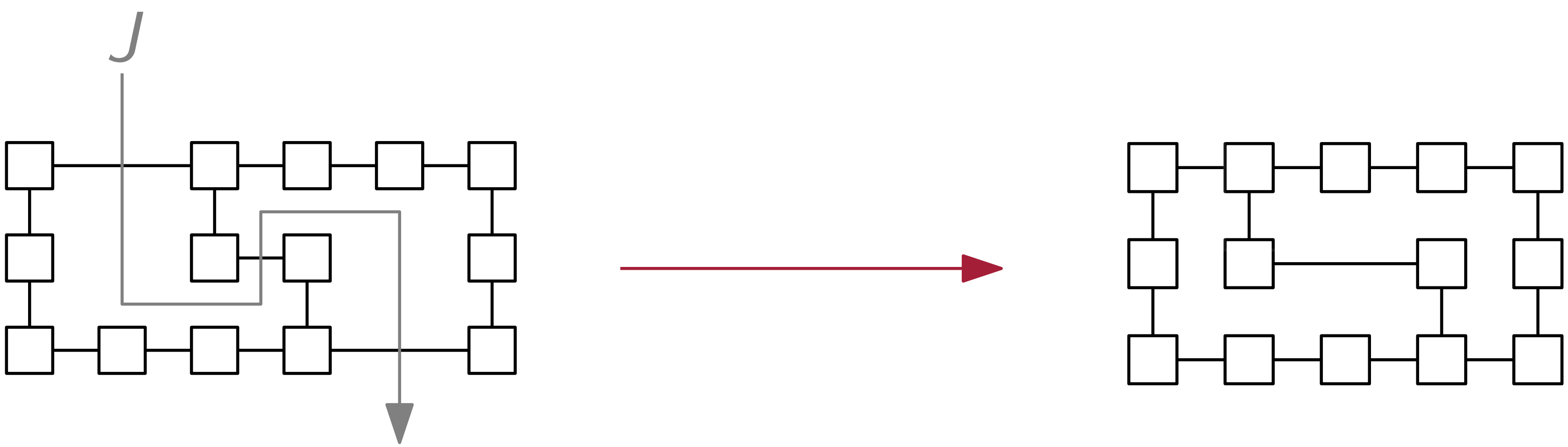
4M Algorithm - Moving

- **Area saving algorithm for orthogonal drawings** [Fößmeier et al. 1998]
- Finds a directed path through horizontal line segments
  - downwards: reduction by one unit length
  - upwards: elongation by one unit length
  - repeat until one of the segments achieved unit length



4M Algorithm - Moving

- **Area saving algorithm for orthogonal drawings** [Fößmeier et al. 1998]
- Finds a directed path through horizontal line segments
  - downwards: reduction by one unit length
  - upwards: elongation by one unit length
  - repeat until one of the segments achieved unit length



- Can save area either horizontally or vertically
- Can be modified for Smoothened Kandinsky drawings

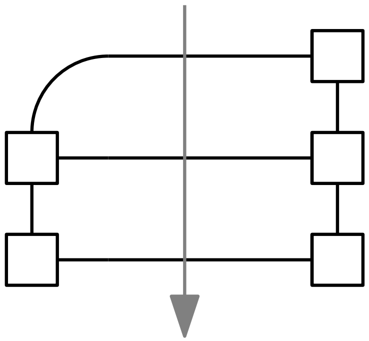
4M Algorithm - Moving modification

4M Algorithm - Moving modification

- Finds a directed path through horizontal line segments **and quarter circular arcs**

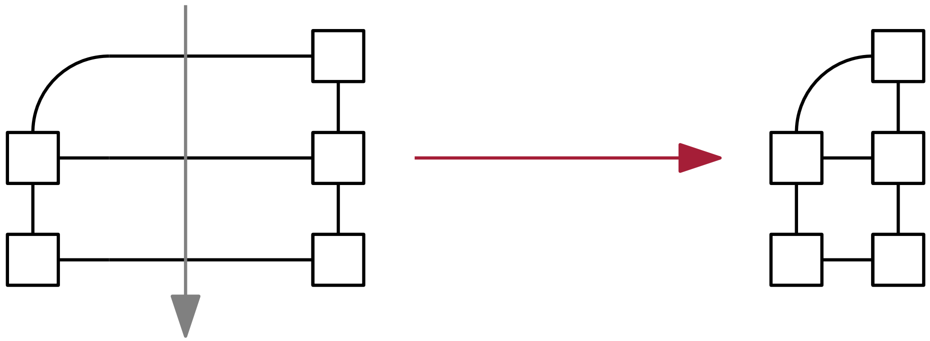
4M Algorithm - Moving modification

- Finds a directed path through horizontal line segments **and quarter circular arcs**
  - Try to cross line segments
    - Complexity might decrease



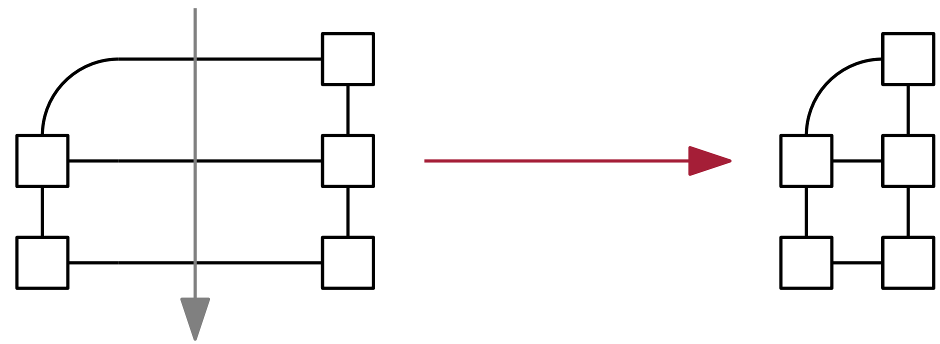
4M Algorithm - Moving modification

- Finds a directed path through horizontal line segments **and quarter circular arcs**
  - Try to cross line segments
    - Complexity might decrease

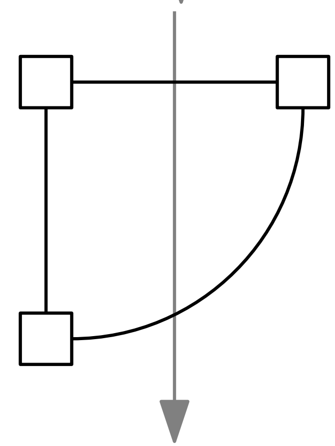


4M Algorithm - Moving modification

- Finds a directed path through horizontal line segments **and quarter circular arcs**
  - Try to cross line segments
    - Complexity might decrease

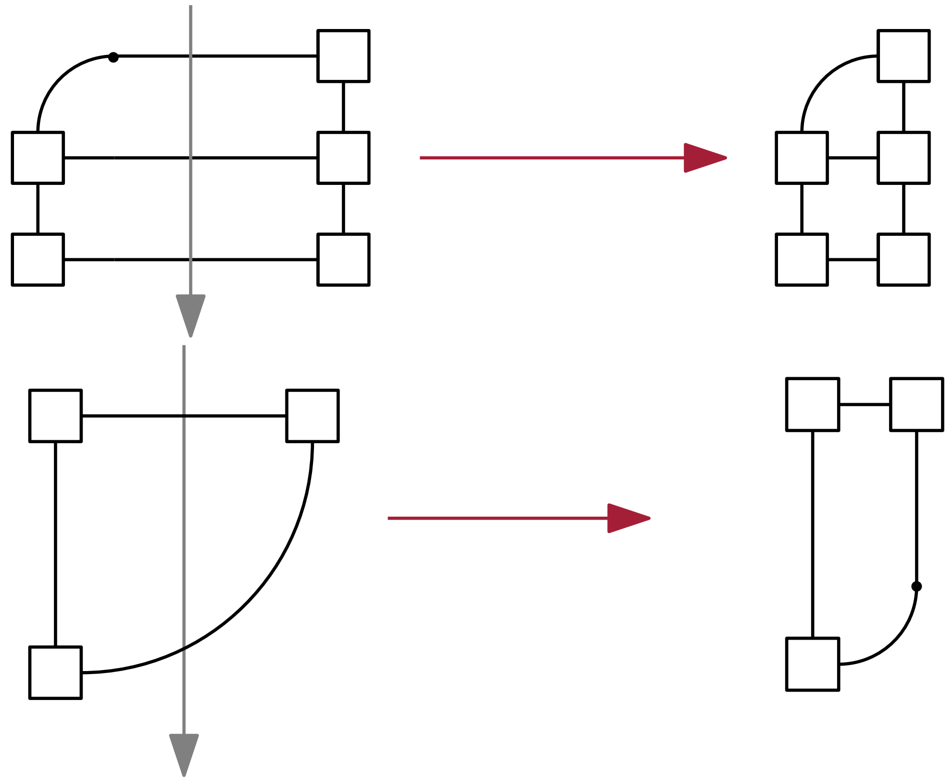


  - Else cross circular arc
    - Complexity might increase



4M Algorithm - Moving modification

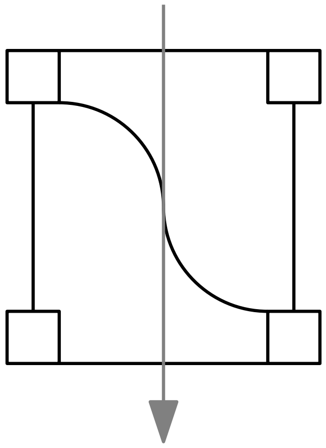
- Finds a directed path through horizontal line segments **and quarter circular arcs**
  - Try to cross line segments
    - Complexity might decrease
  - Else cross circular arc
    - Complexity might increase



  - Quarter circular arcs substituted
    - downwards: smaller circular arc, line segment
    - upwards: same-sized circular arc, line segment

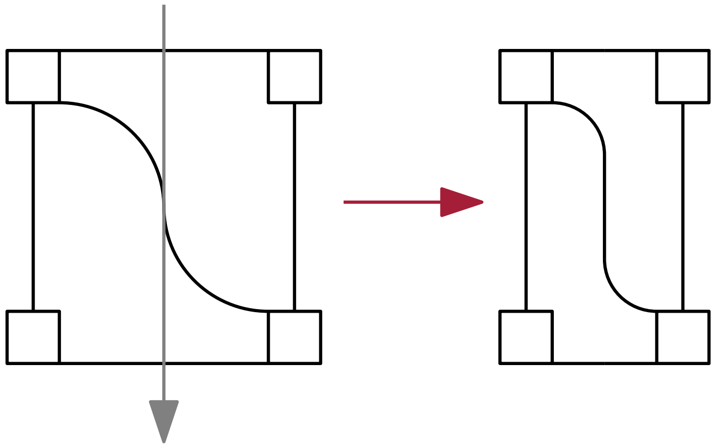
4M - Example

- Horizontal saving...



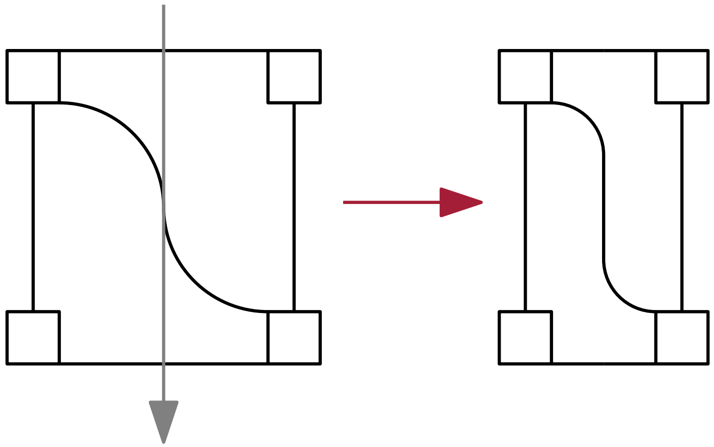
4M - Example

- Horizontal saving...

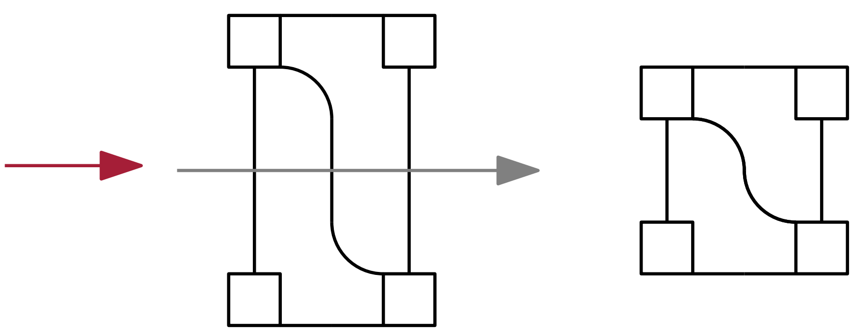


4M - Example

- Horizontal saving...

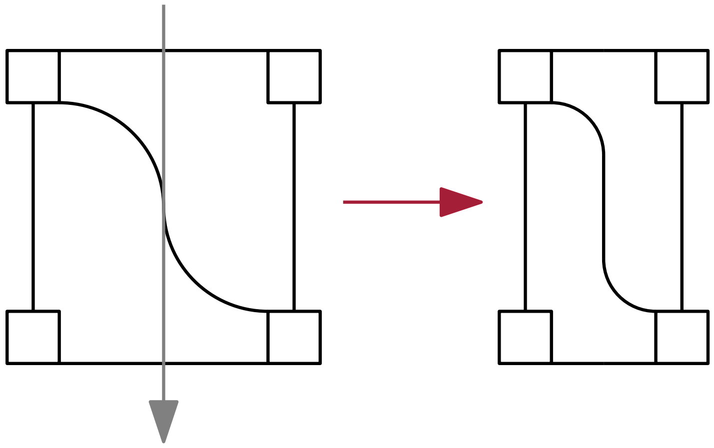


- Followed by vertical saving

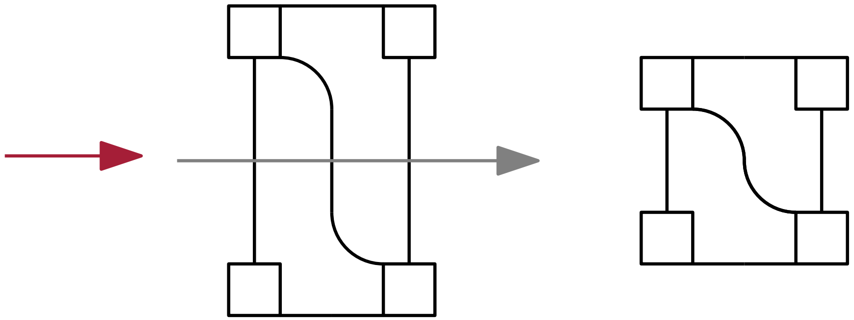


4M - Example

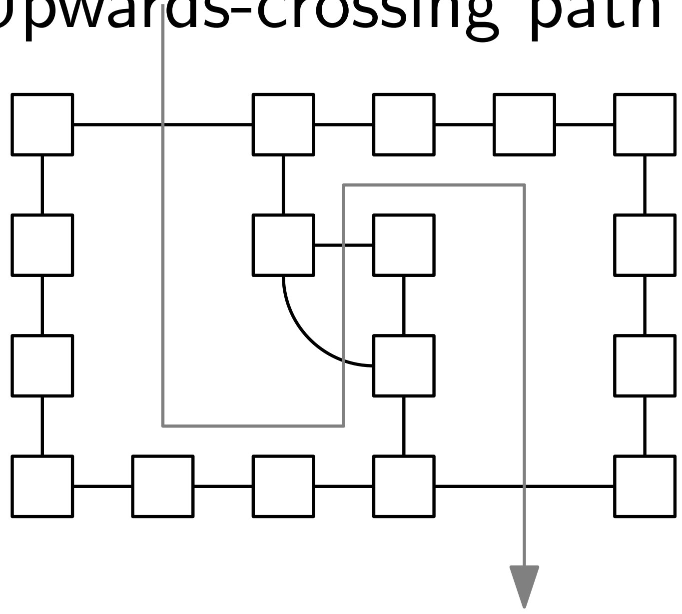
- Horizontal saving...



- Followed by vertical saving

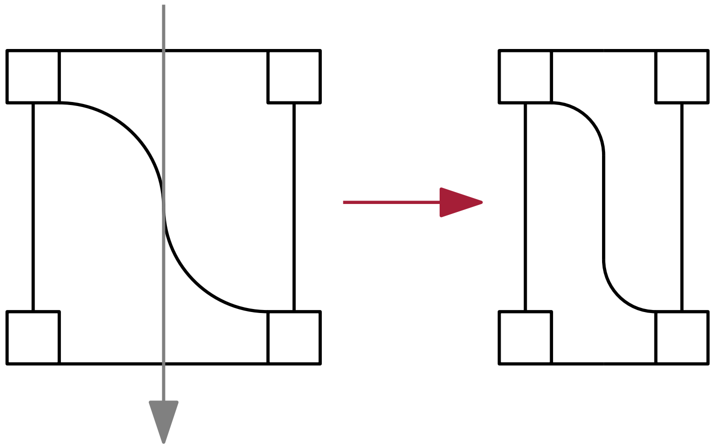


- Upwards-crossing path

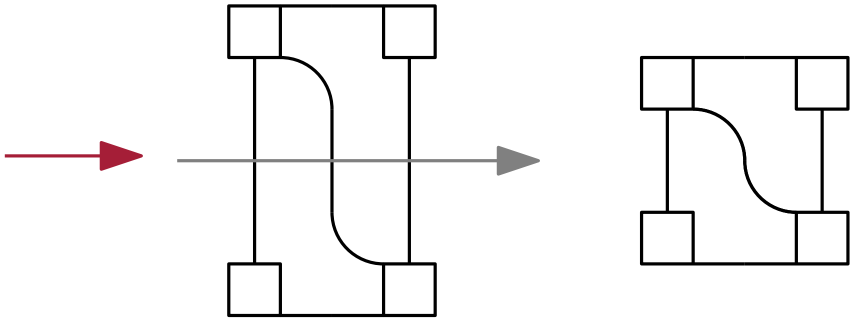


4M - Example

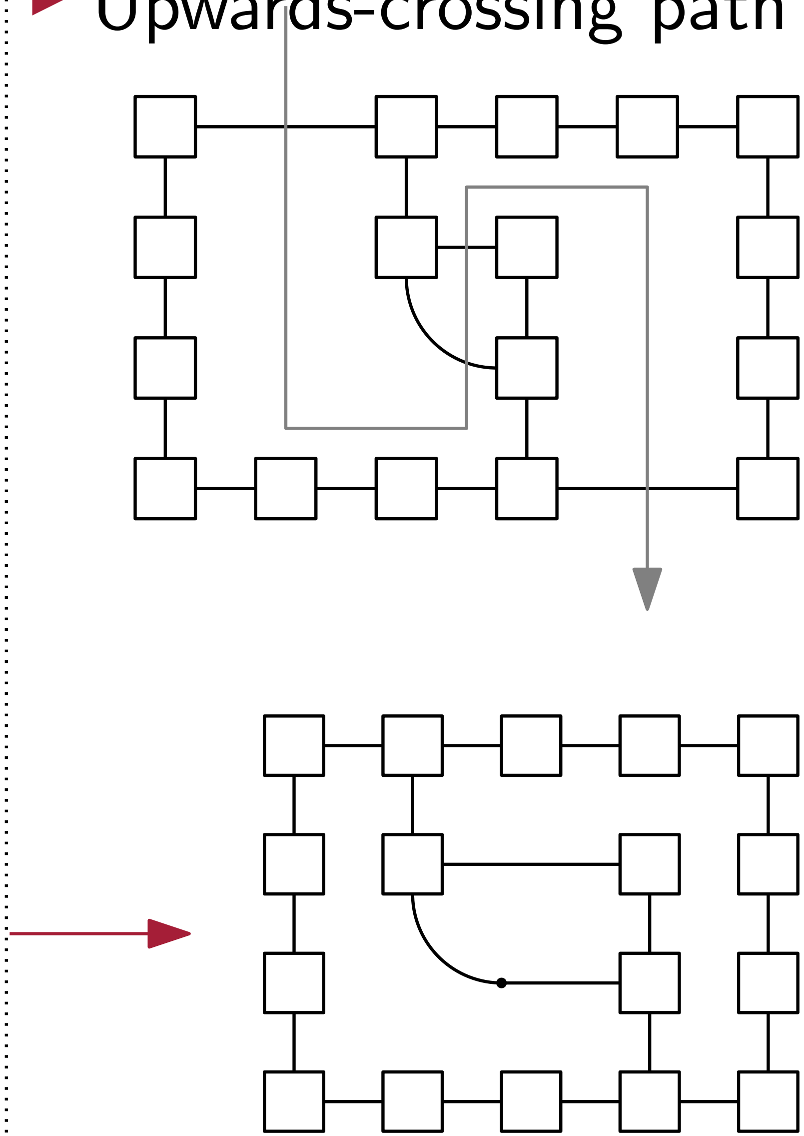
- Horizontal saving...



- Followed by vertical saving



- Upwards-crossing path



Summary

- Area saving often requires complexity increase
  - Square root stretching would suffice, needs complexity increase for clarity
  - 4M always increases complexity, when crossing a circular arc, but saves area consequently

Summary

- Area saving often requires complexity increase
  - Square root stretching would suffice, needs complexity increase for clarity
  - 4M always increases complexity, when crossing a circular arc, but saves area consequently

- Area saving plane sweep not sufficient
- 4M Moving Modification may suit well for saving measures

Summary

- Area saving often requires complexity increase
  - Square root stretching would suffice, needs complexity increase for clarity
  - 4M always increases complexity, when crossing a circular arc, but saves area consequently

- Area saving plane sweep not sufficient
- 4M Moving Modification may suit well for saving measures

- Port reassignment may decrease complexity
  - hard to determine, whether possible or not

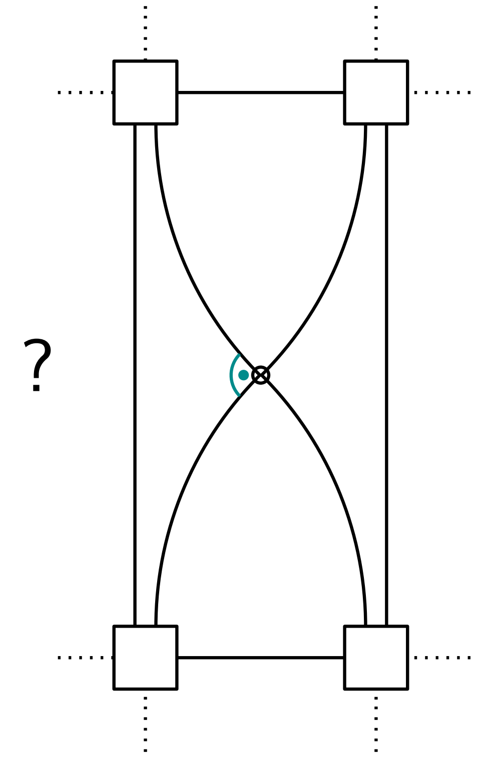
Open Problems

- **Implementation**
  - Useful for heuristics
    - How do smoothened Kandinsky drawings look like?
    - How efficient are the saving measures?

Open Problems

- **Implementation**
  - Useful for heuristics
    - How do smoothened Kandinsky drawings look like?
    - How efficient are the saving measures?

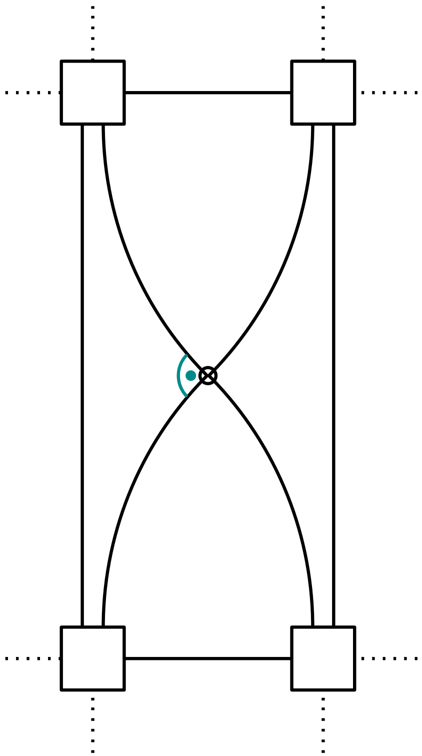
- **Graphs with crossings**



Open Problems

- **Implementation**
  - Useful for heuristics
    - How do smoothened Kandinsky drawings look like?
    - How efficient are the saving measures?
- **Graphs with crossings**

?



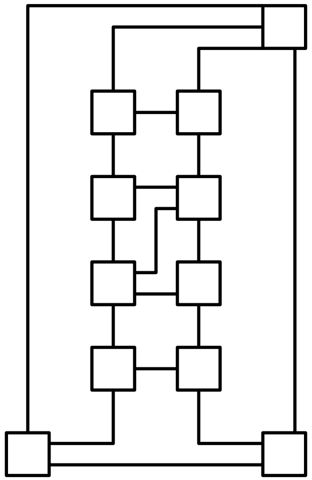
- **Further saving approaches**

Questions?

?
?
?
?
?
?
?
?
?
?
?

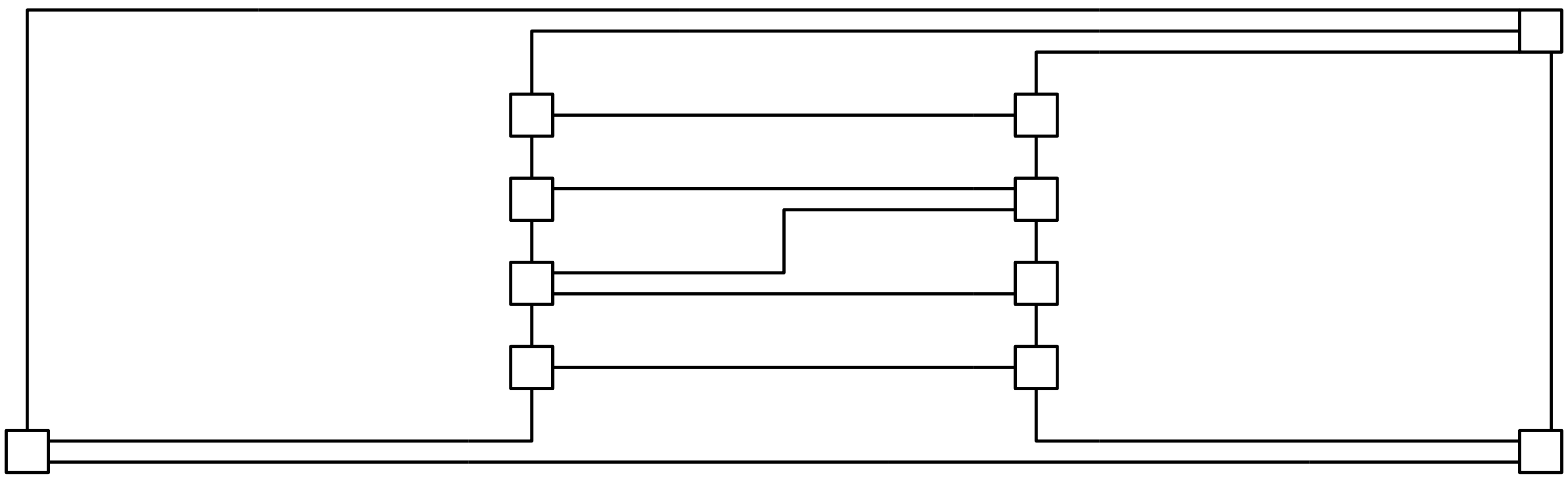
Appendix - Let's see, how it should look

- Example
  - Input drawing



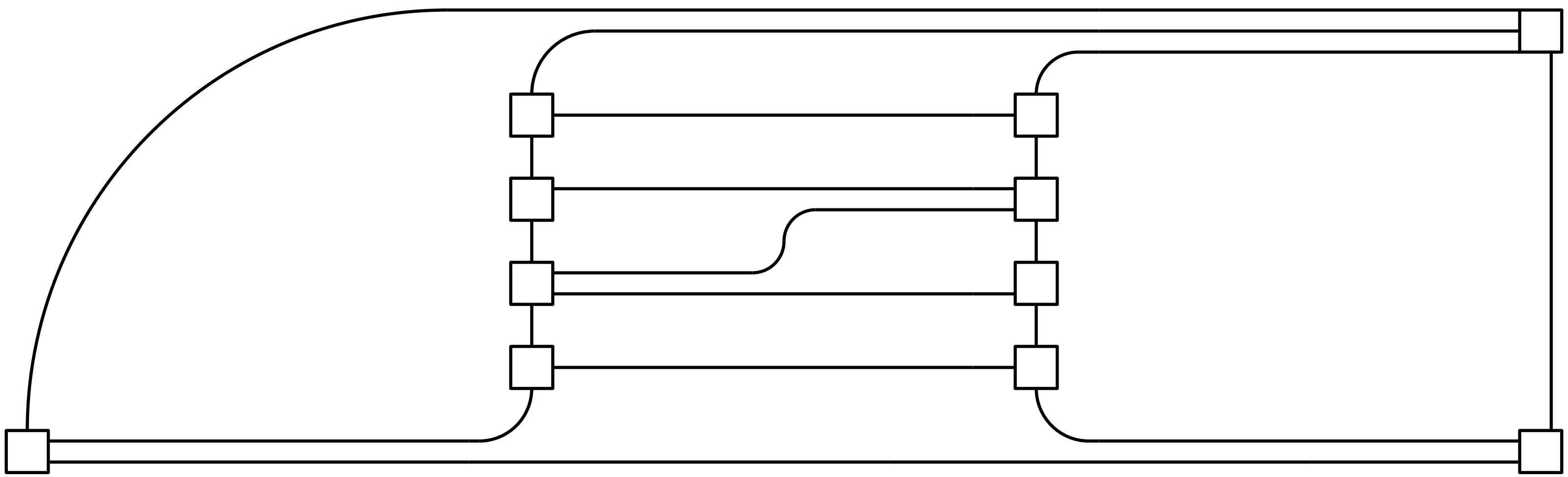
Appendix

- Example
  - Stretched by the longest vertical segment



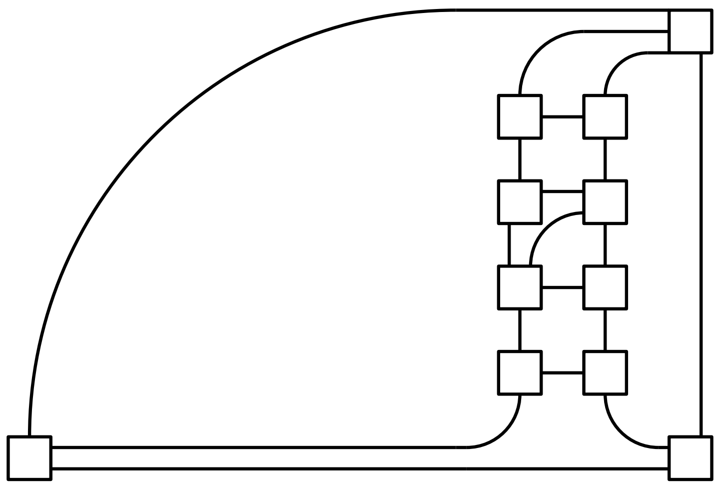
Appendix

- Example
  - Substituted



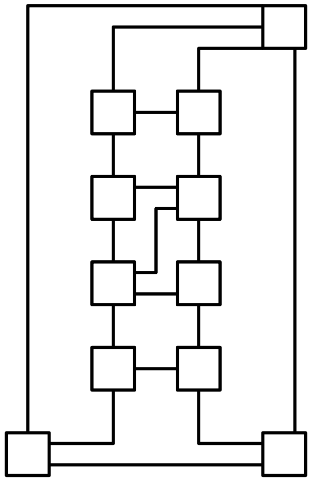
Appendix

- Example
  - Optimized



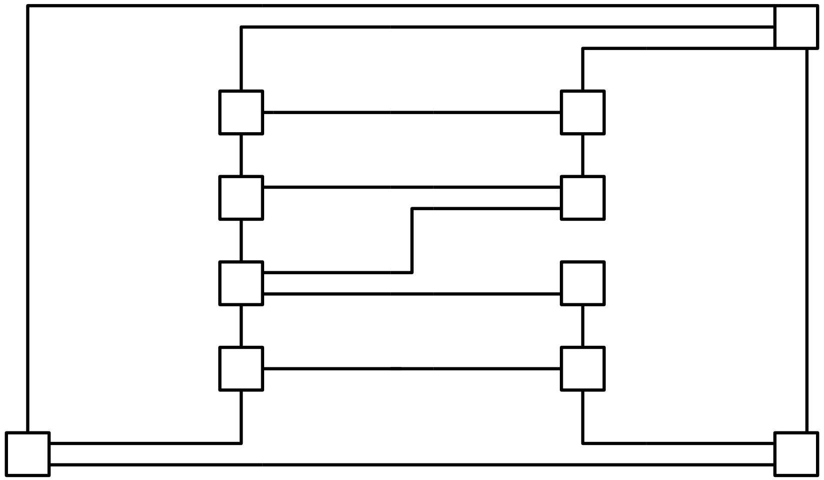
Appendix

- Example
  - Input drawing



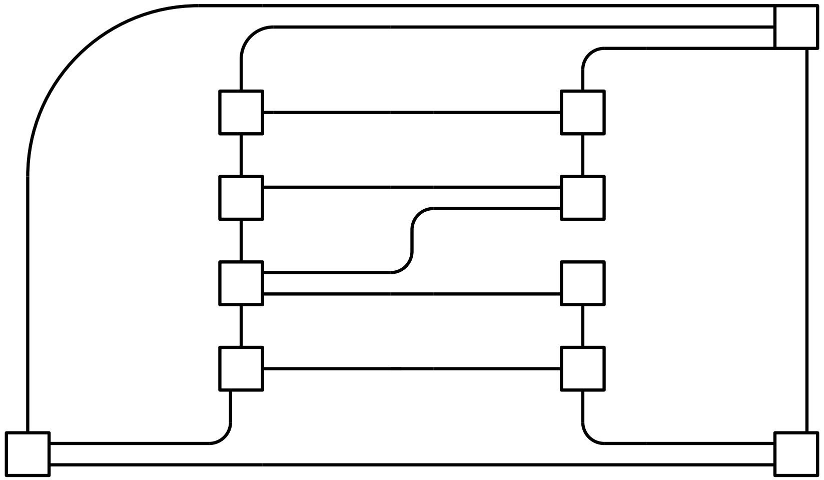
Appendix

- Example
  - Stretched by the sqrt of longest vertical segment



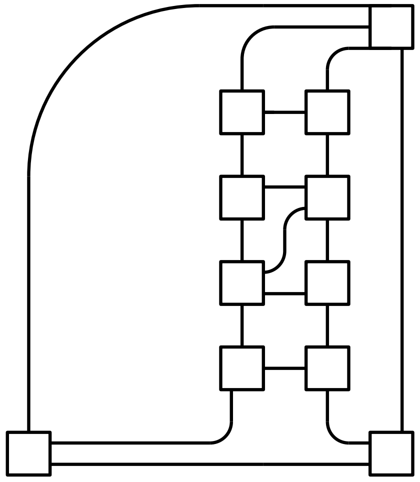
Appendix

- Example
  - Substitution of circular arcs with vertical segments



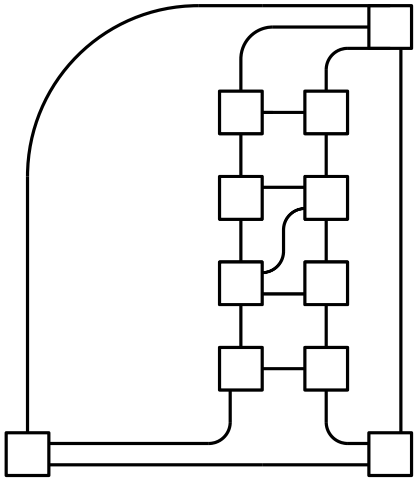
Appendix

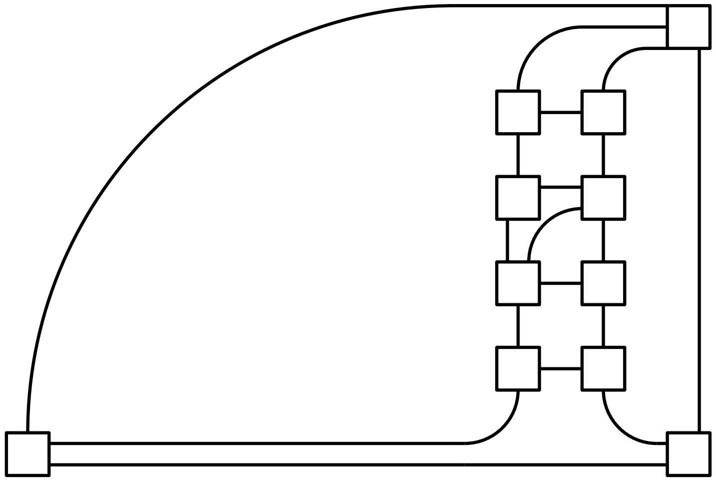
- Example
  - Optimization



Appendix

- Example
  - Comparism





Thank you!

:)