Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chair for
INFORMATICS I
Efficient Algorithms and
Knowledge-Based Systems

Institute for Informatics

Bachelor Colloquium

Rectangular Representation of Weighted Outerplanar Graphs

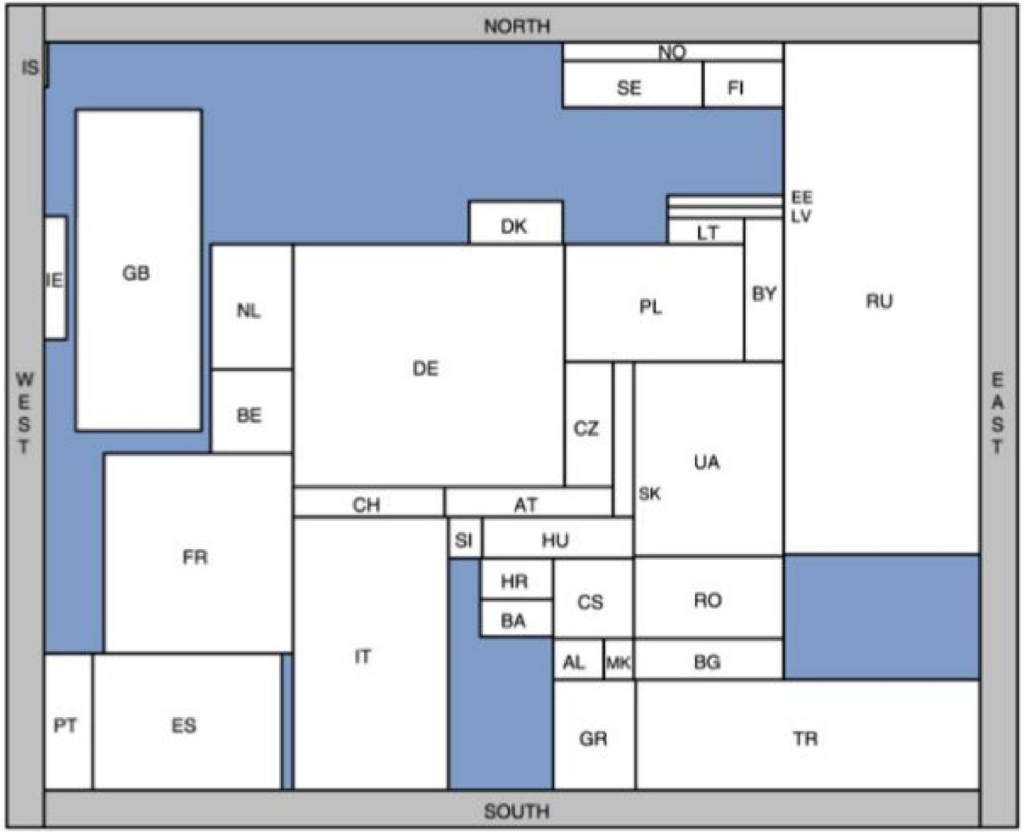
Lorenz Reinhart
October 01, 2014

Supervisors
Philipp Kindermann & Alexander Wolff
Chair of Computer Science I
Universität Würzburg

Motivation

Rectangular Cartograms:

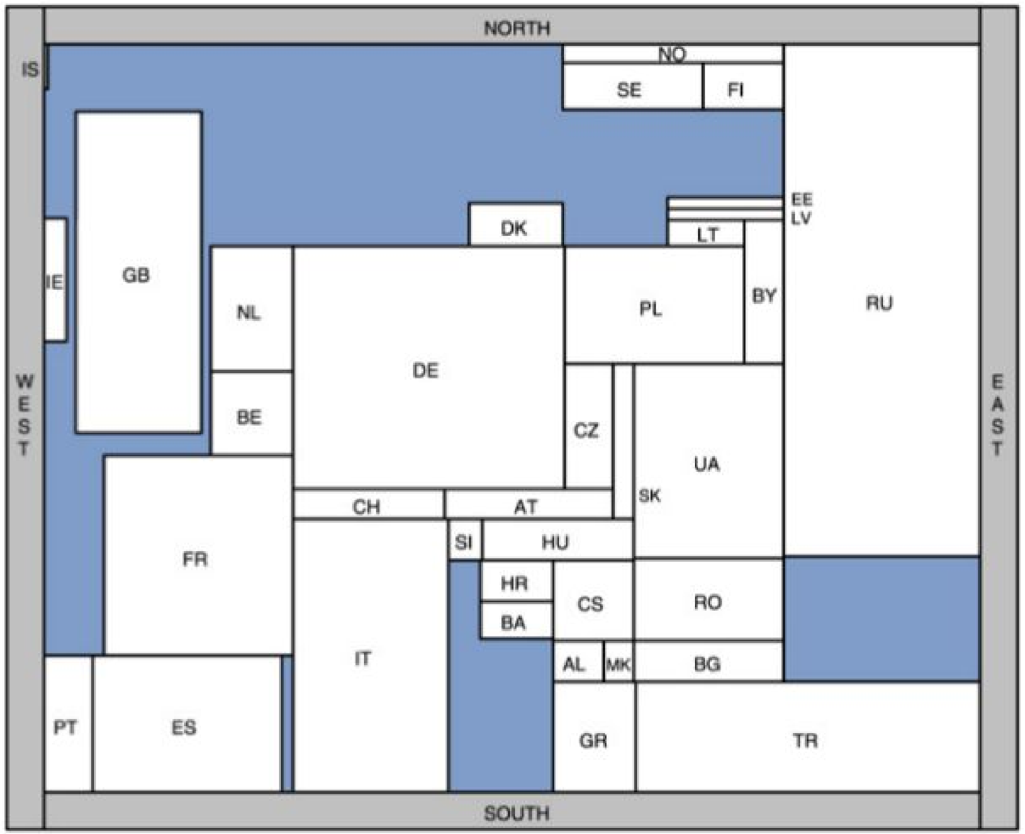
Representation of maps where every region is represented by a rectangle.



Motivation

Rectangular Cartograms:

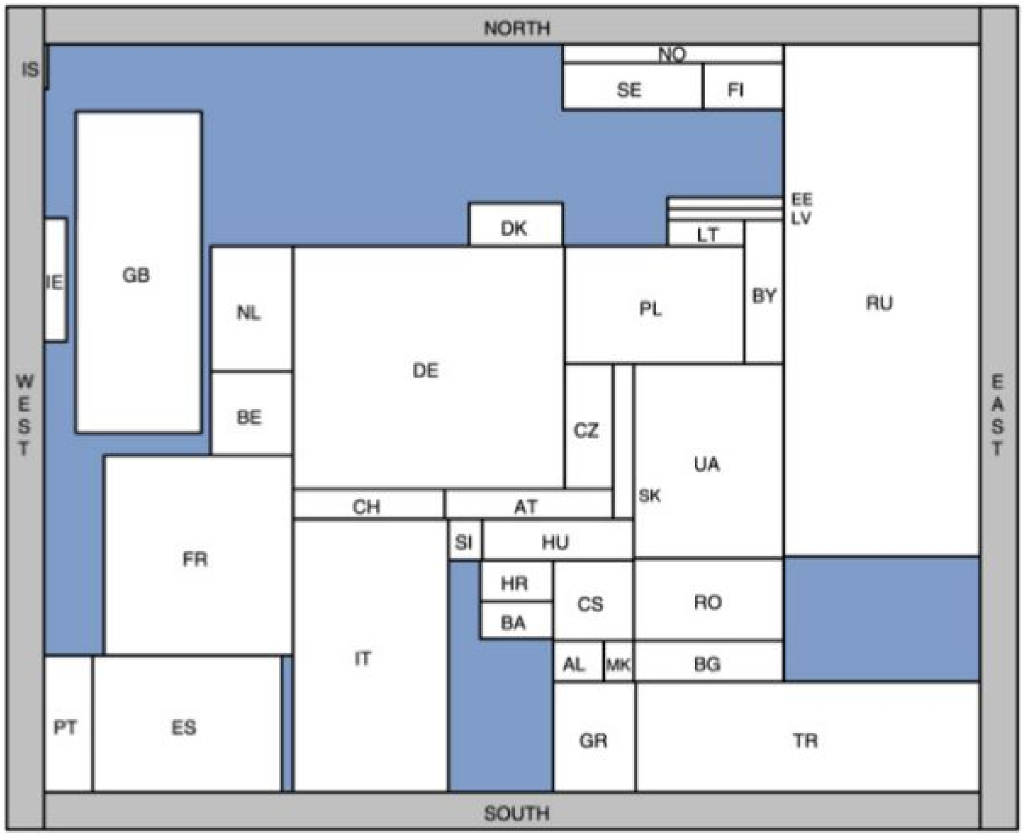
Representation of maps where every region is represented by a rectangle.



Motivation

Rectangular Cartograms:

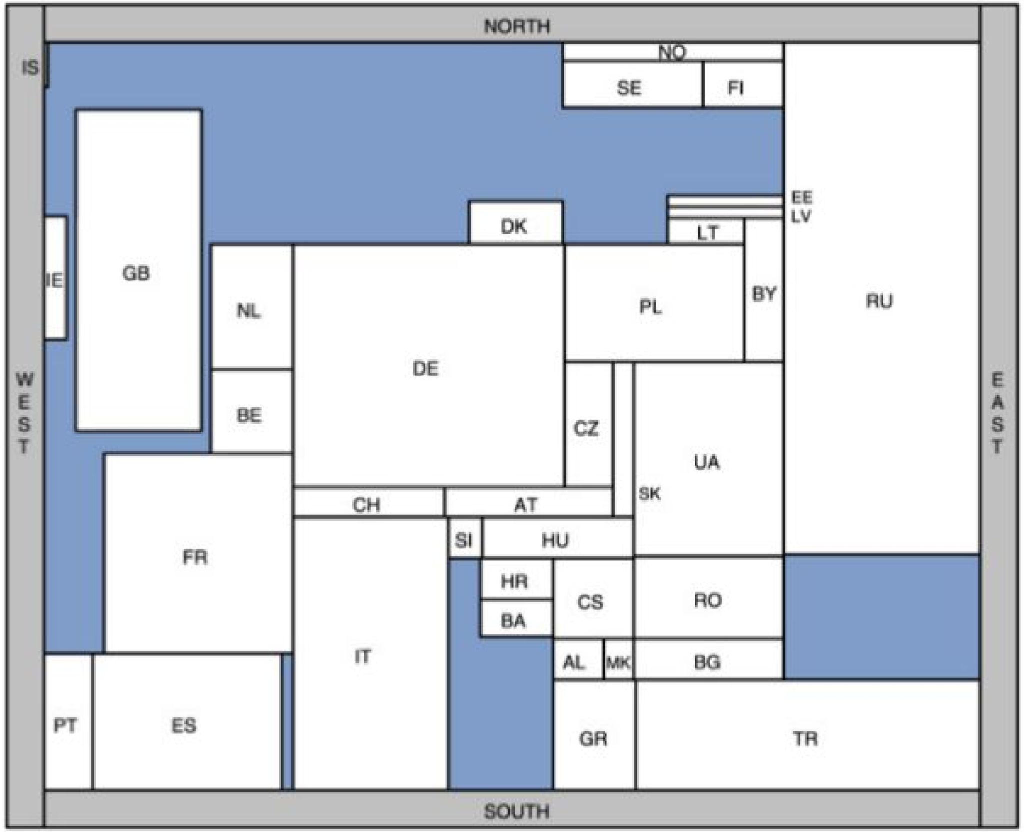
Representation of maps where every region is represented by a rectangle.

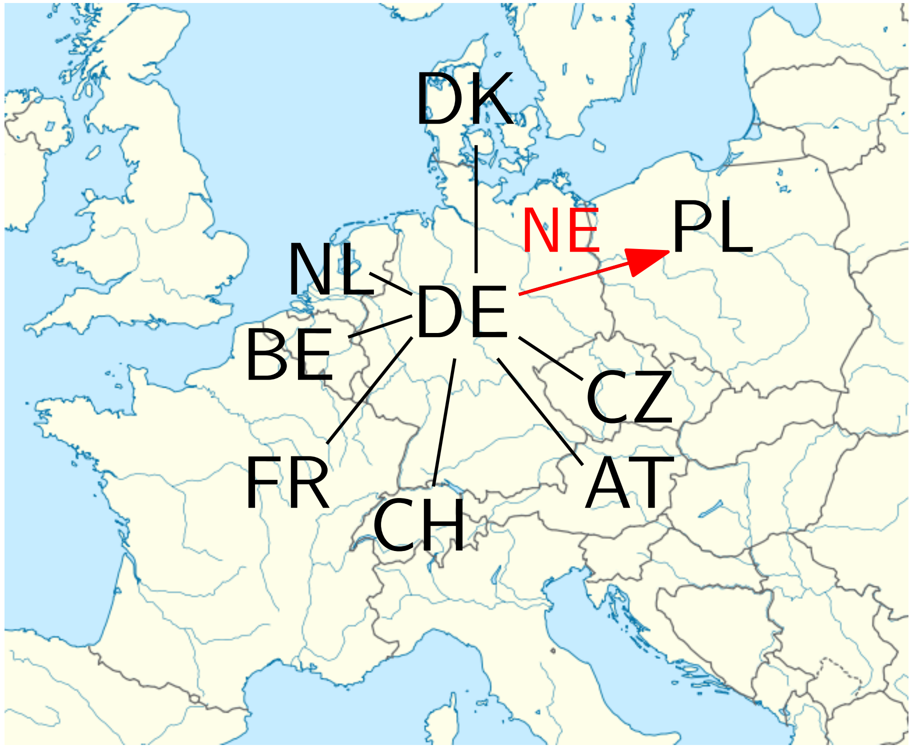


Motivation

Rectangular Cartograms:

Representation of maps where every region is represented by a rectangle.

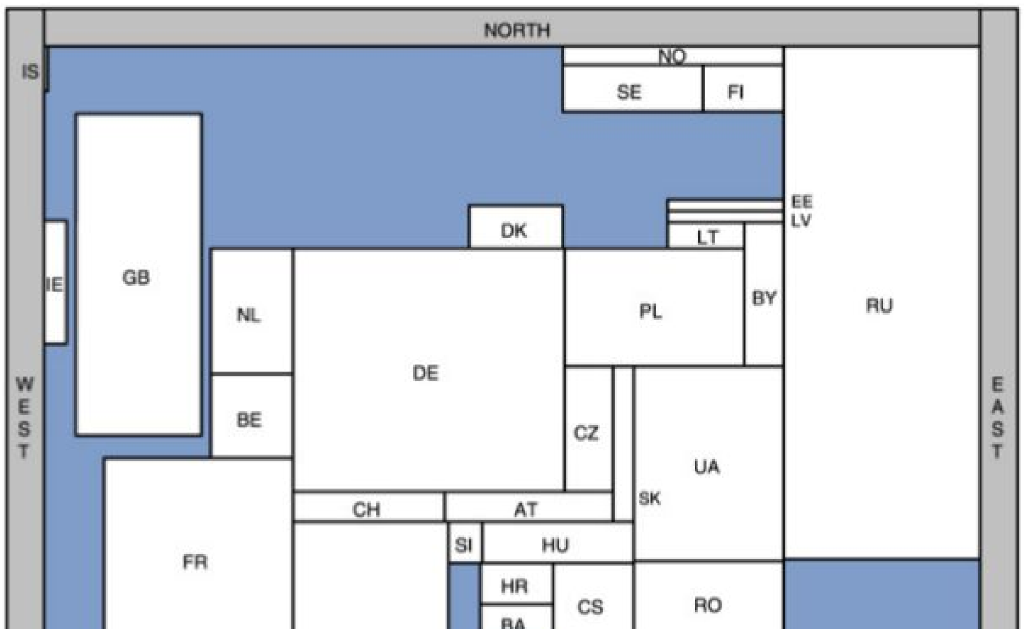


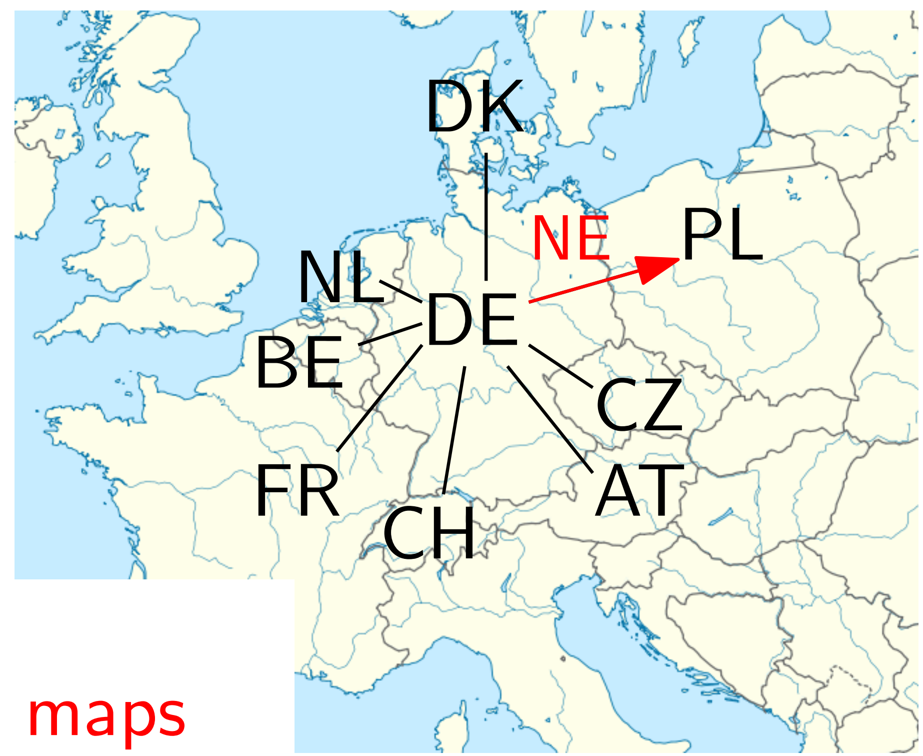


Motivation

Rectangular Cartograms:

Representation of maps where every region is represented by a rectangle.





Problems:

- wrong representation of some maps
- graphs without geographic information

Targets

- draw the rectangles with the desired size

Targets

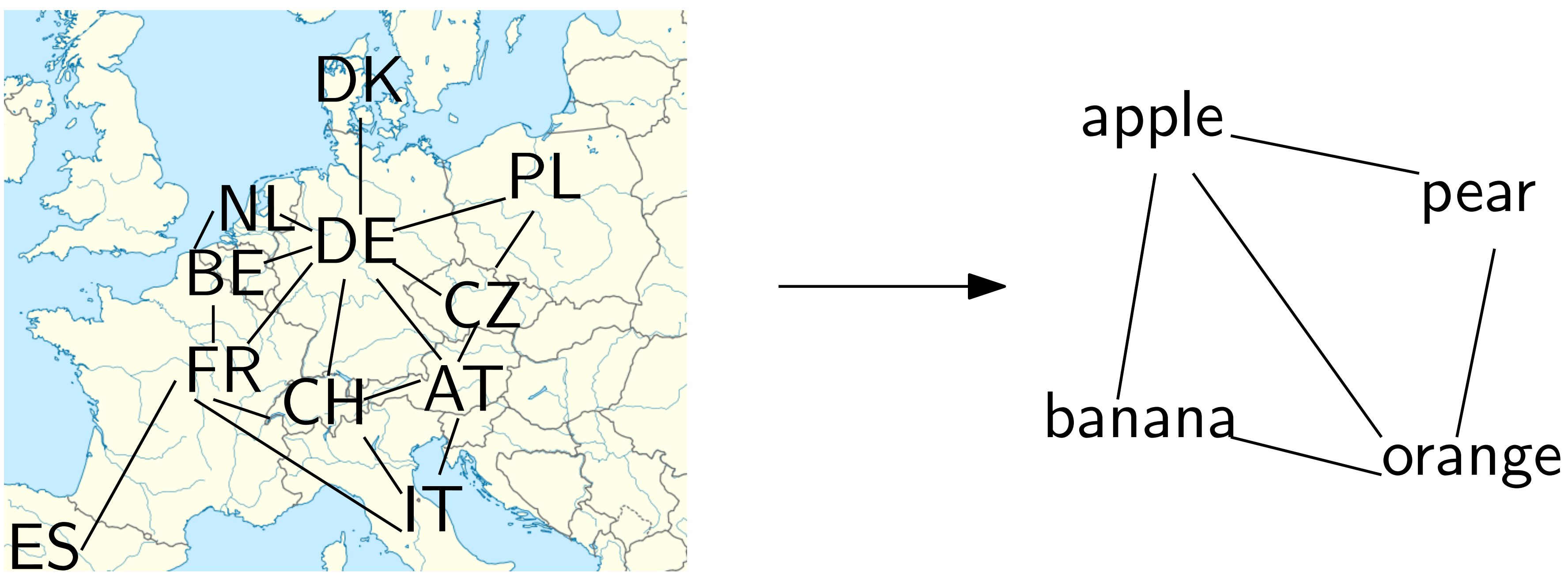
- draw the rectangles with the desired size
- every rectangular representation should be possible to draw

Targets

- draw the rectangles with the desired size
- every rectangular representation should be possible to draw
- no need for a surrounding rectangle

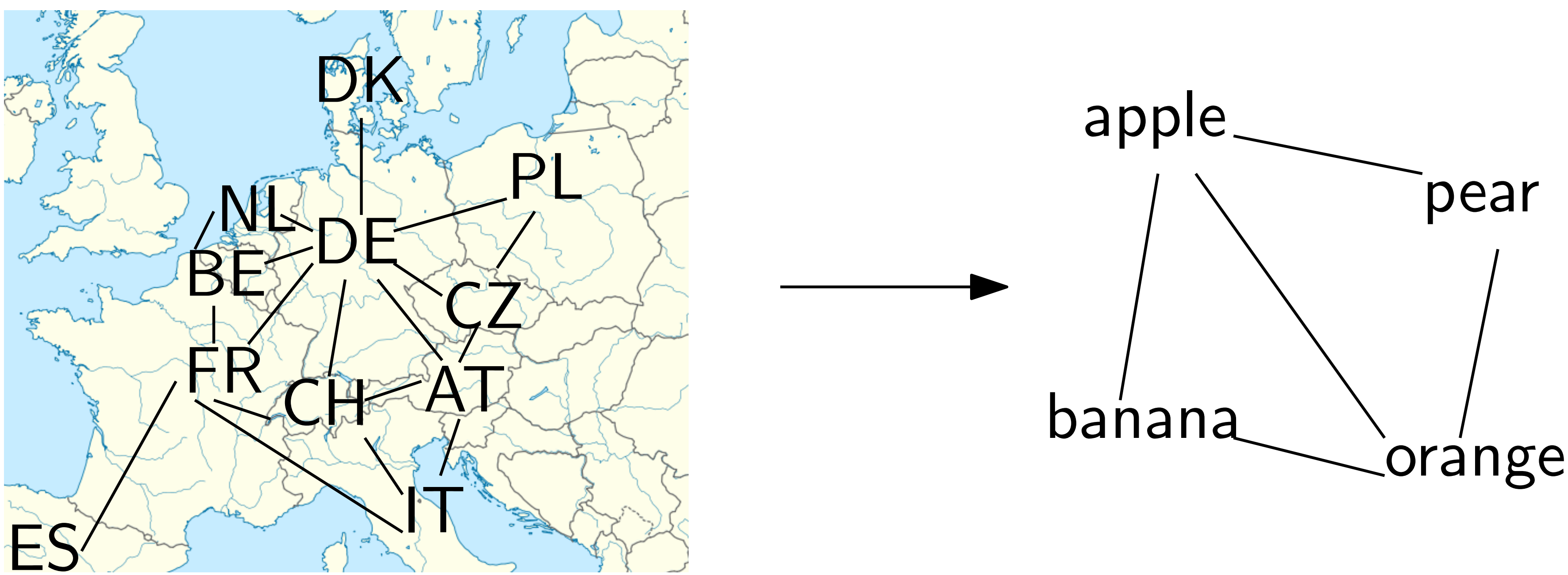
Targets

- draw the rectangles with the desired size
- every rectangular representation should be possible to draw
- no need for a surrounding rectangle
- extension to general graphs



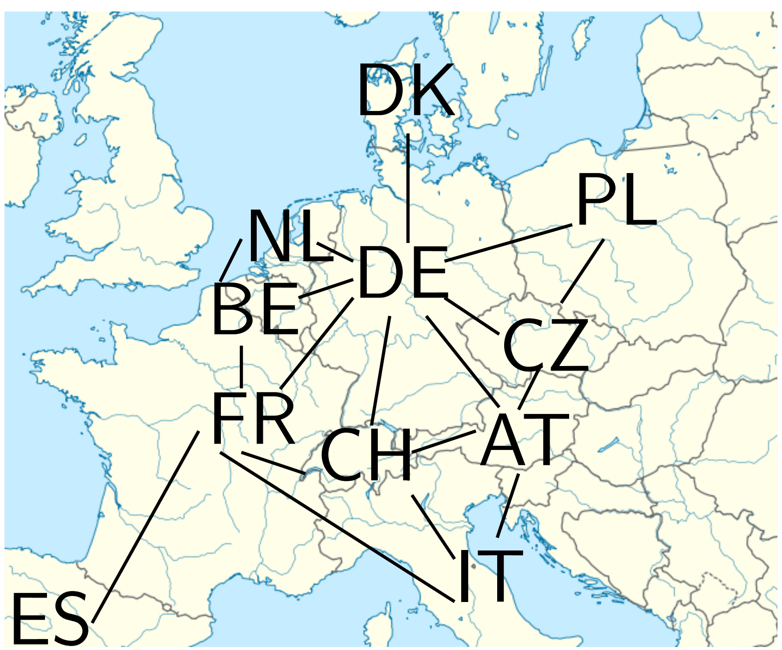
Targets

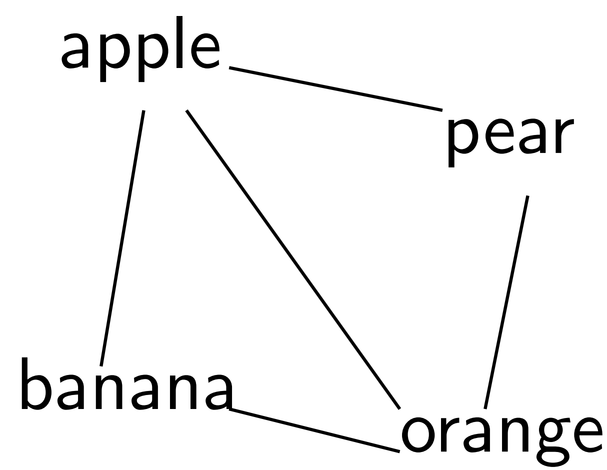
- draw the rectangles with the desired size
- every rectangular representation should be possible to draw
- no need for a surrounding rectangle
- extension to general graphs?



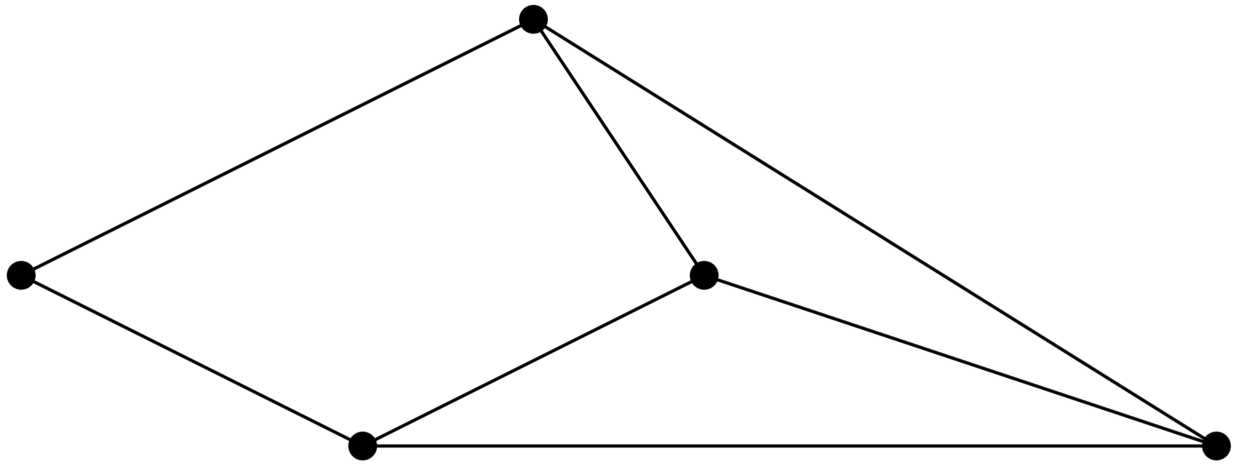
Targets

- draw the rectangles with the desired size
- every rectangular representation should be possible to draw
- no need for a surrounding rectangle
- extension to general graph classes!



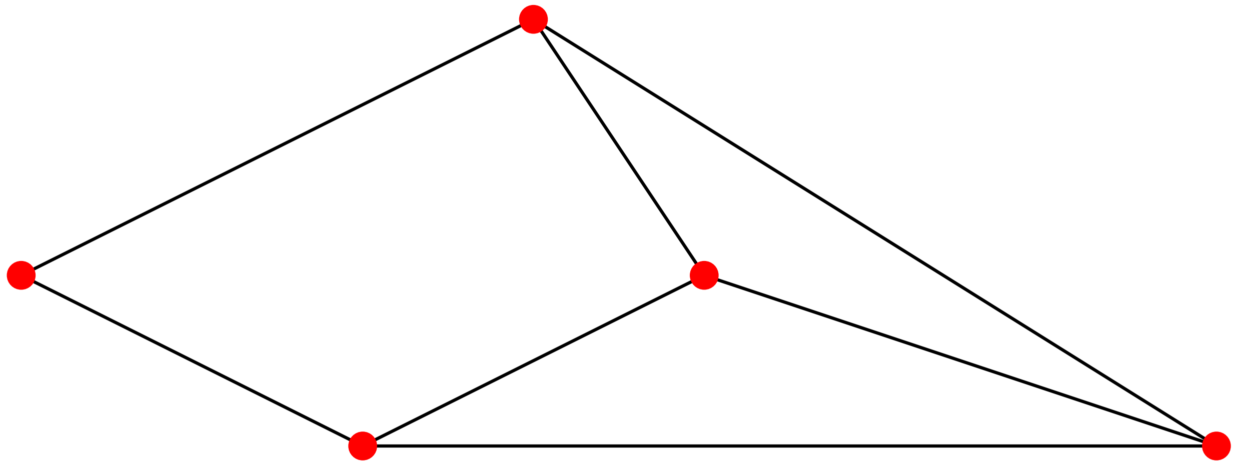


Basic Graphs



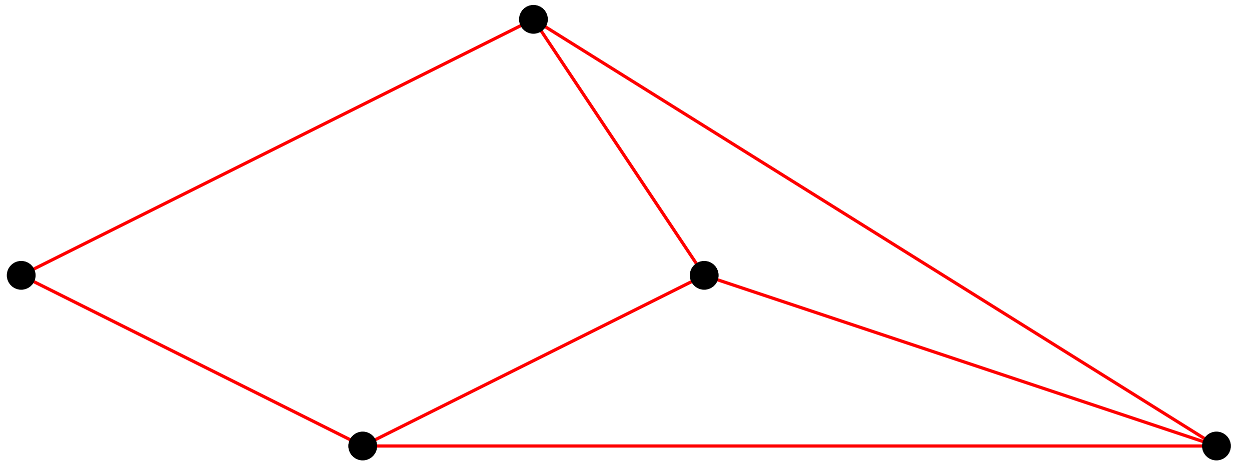
Graph $G = (V, E)$

Basic Graphs



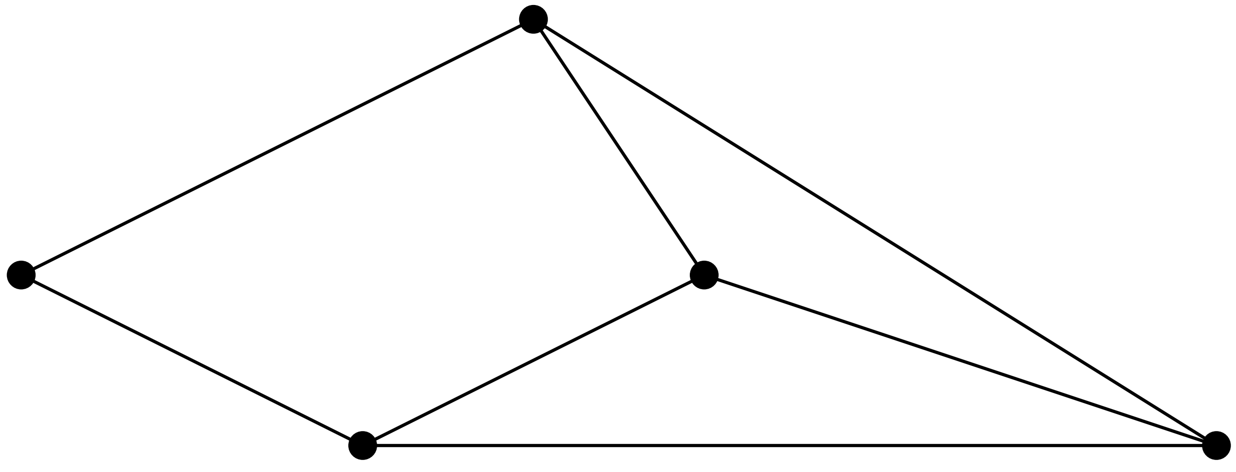
$Graph\ G = (V, E)$

Basic Graphs



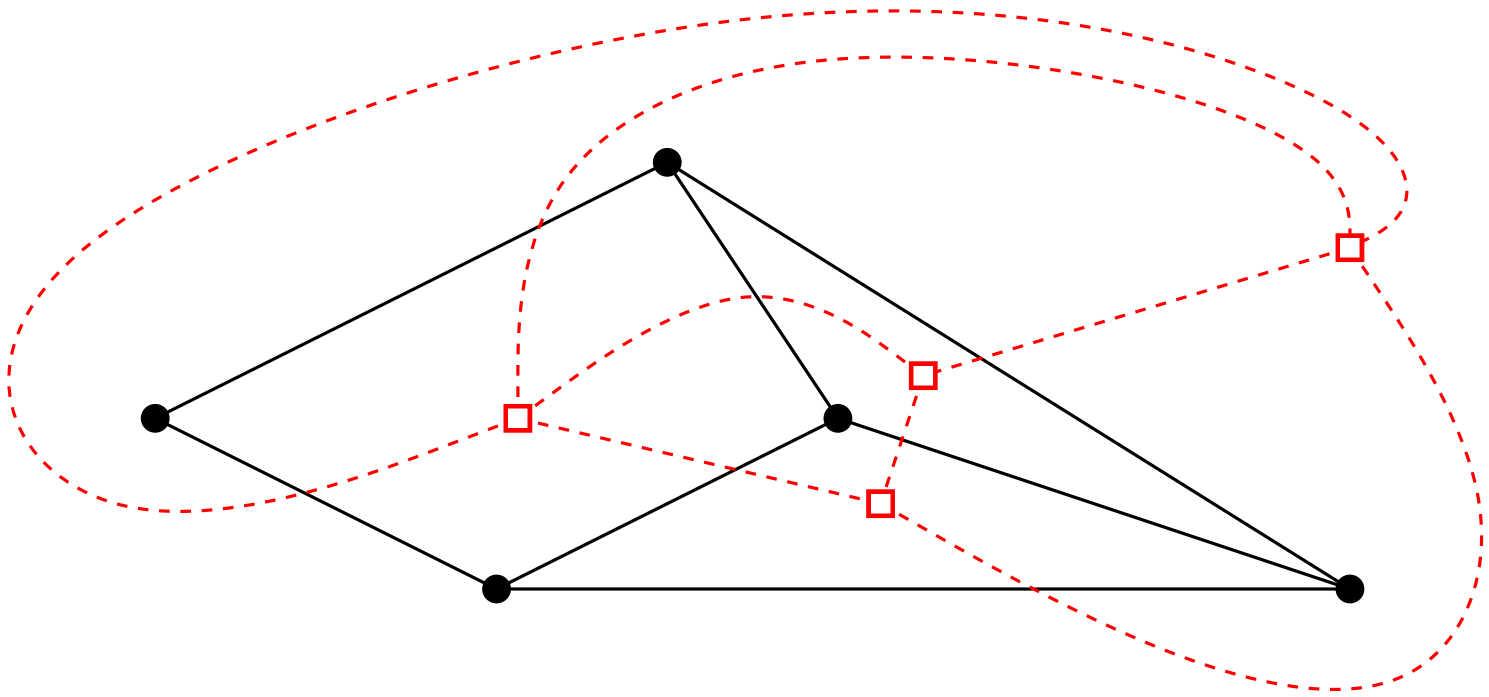
$Graph\ G = (V, E)$

Basic Graphs



A *Planar Graph* $G = (V, E)$ can be drawn in such a way that no edges cross each other

Basic Graphs

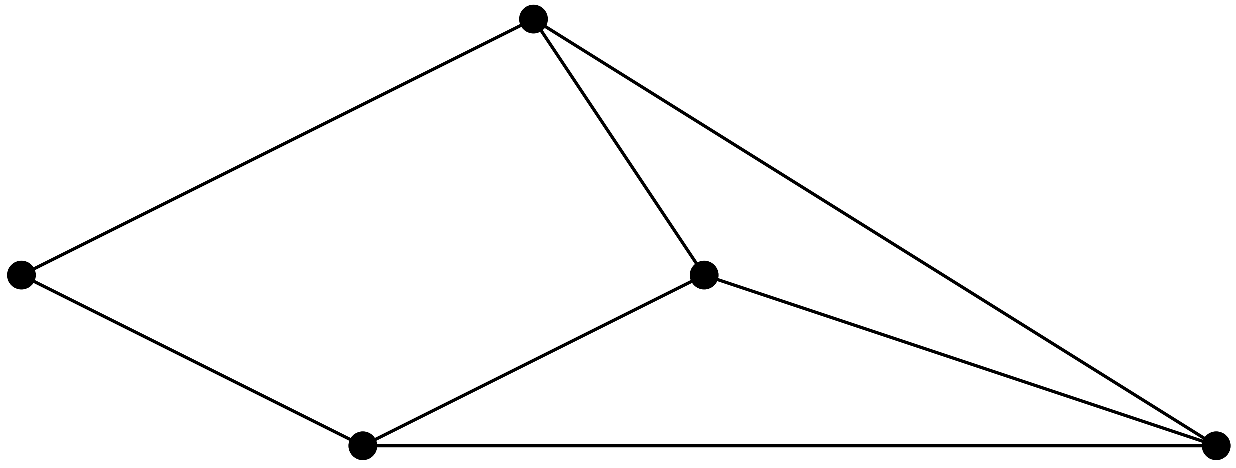


A *Planar Graph* $G = (V, E)$ can be drawn in such a way that no edges cross each other

Dual Graph G*=(V*,E*):

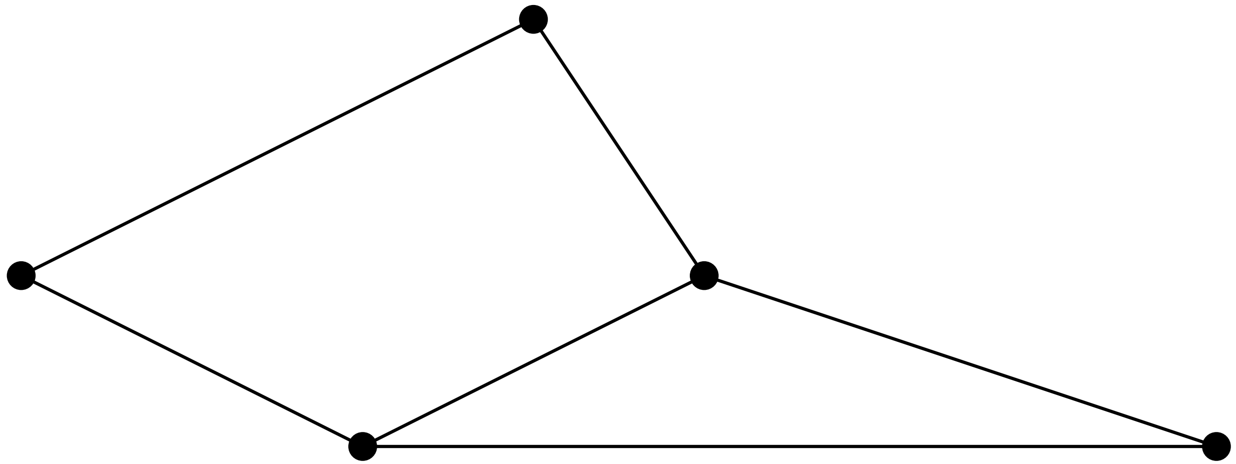
- a vertex corresponding to each face of G
- an edge joining two neighboring faces for each edge in G

Maximal Outerplanar Graphs



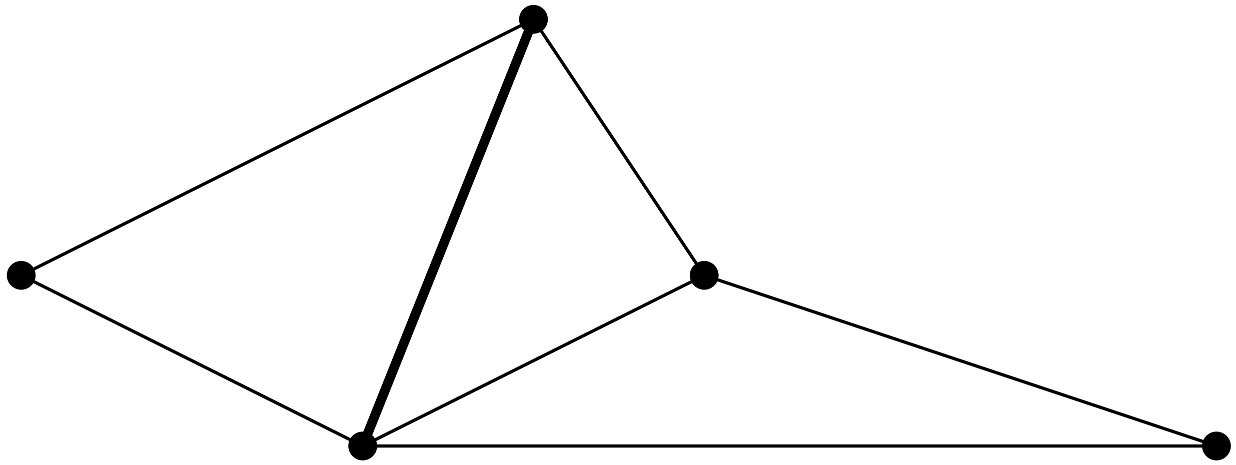
Outerplanar graph: all vertices belong to the unbounded face

Maximal Outerplanar Graphs



Outerplanar graph: all vertices belong to the unbounded face

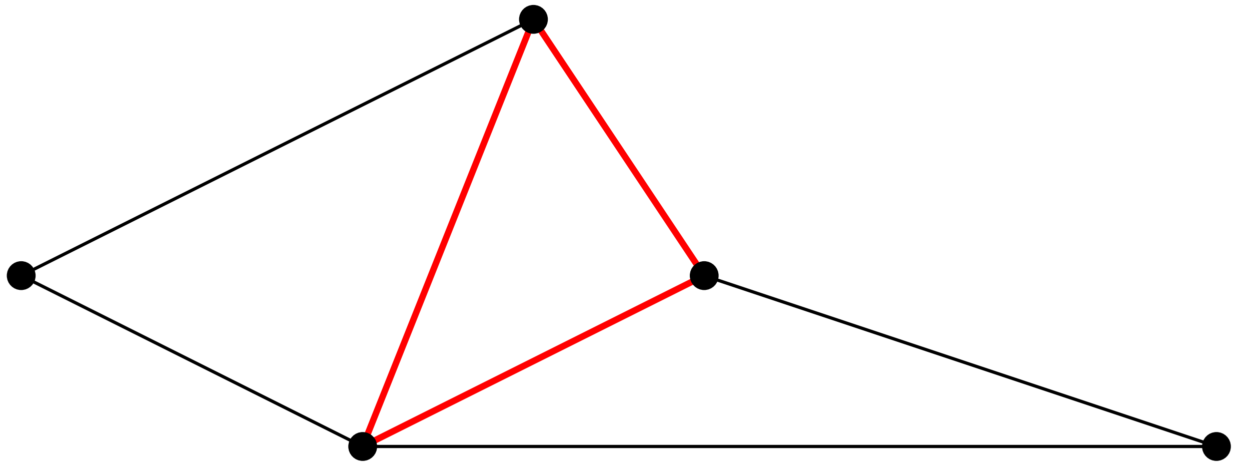
Maximal Outerplanar Graphs



Outerplanar graph: all vertices belong to the unbounded face

Maximal outerplanar Graph: cannot have any additional edges while preserving outerplanarity

Maximal Outerplanar Graphs



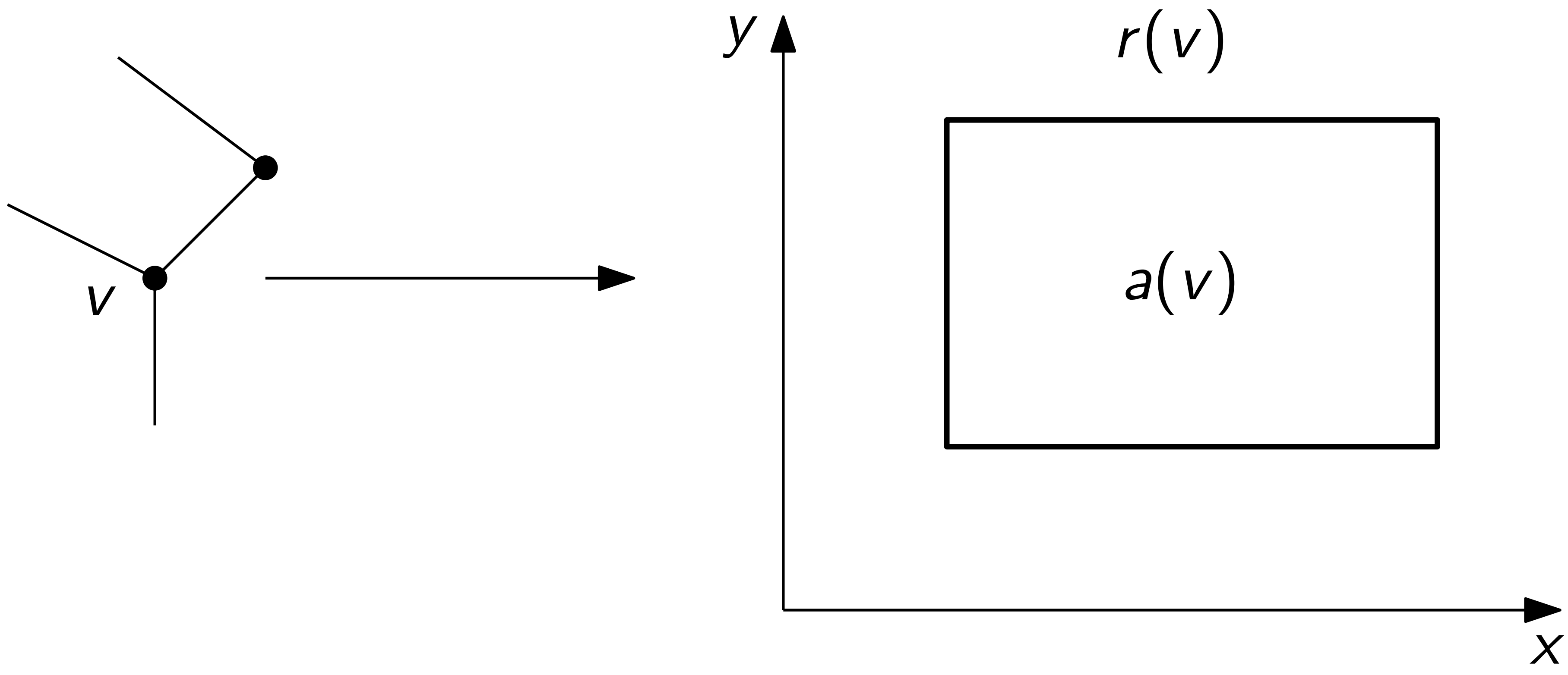
Outerplanar graph: all vertices belong to the unbounded face

Maximal outerplanar Graph: cannot have any additional edges while preserving outerplanarity

Every surface is surrounded by a triangle!

Rectangles

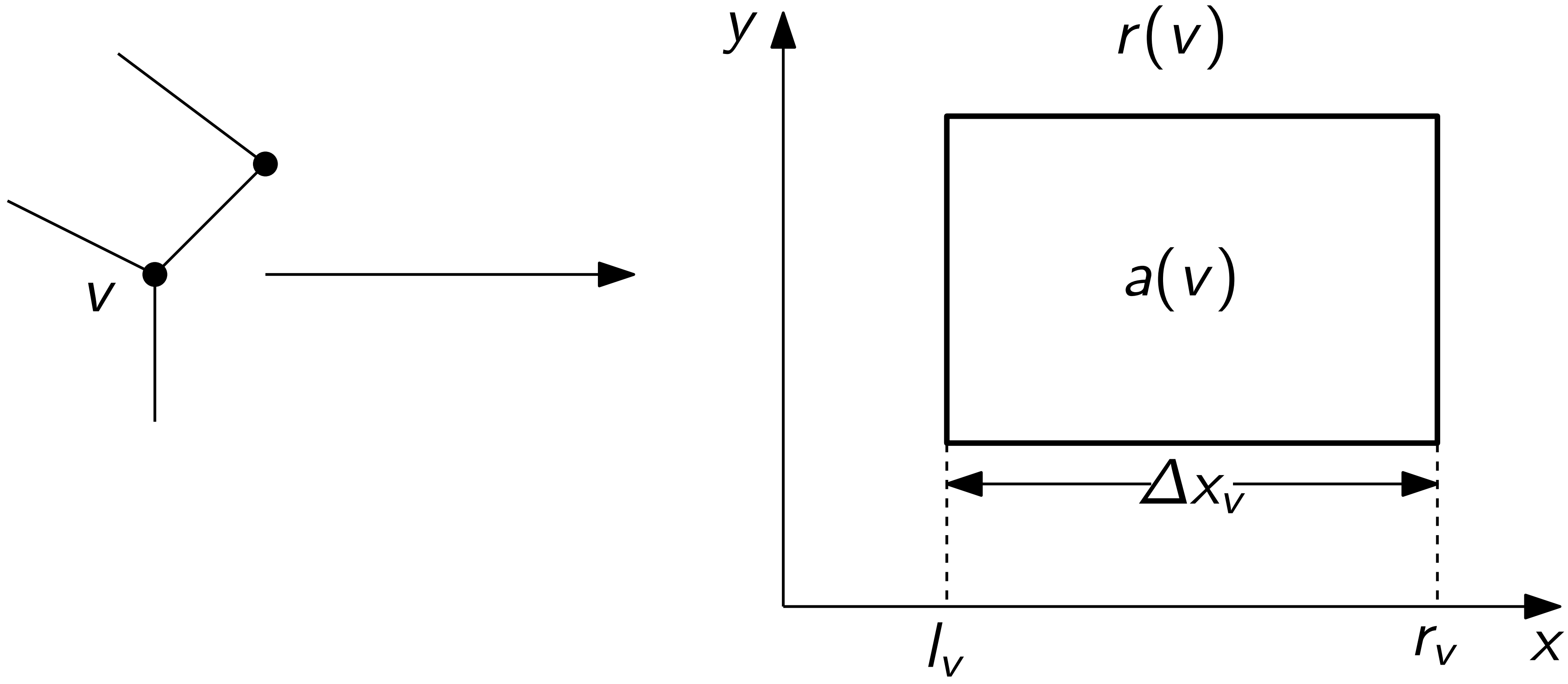
Rectangular representation



A vertex v is corresponding to a rectangle $r(v)$ with size $a(v)$

Rectangles

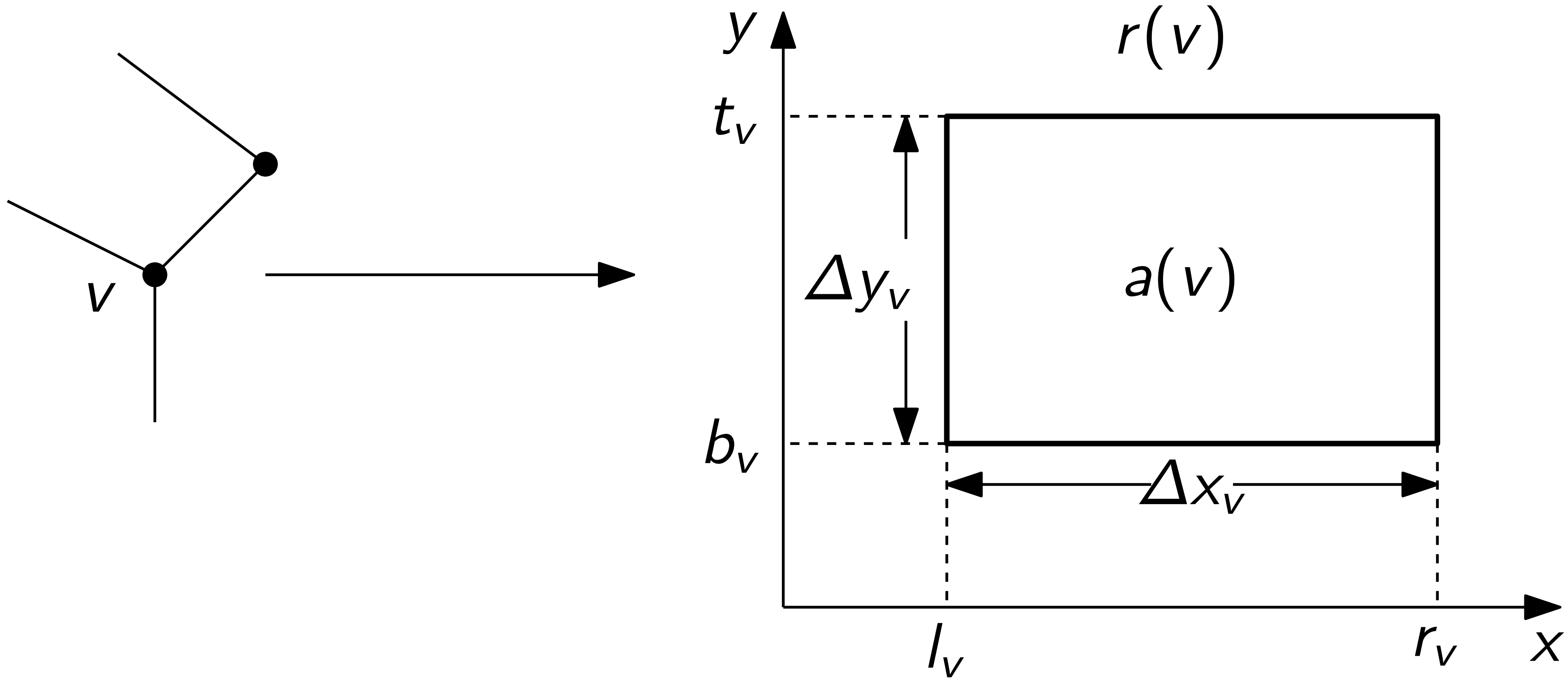
Rectangular representation



A vertex v is corresponding to a rectangle $r(v)$ with size $a(v)$

Rectangles

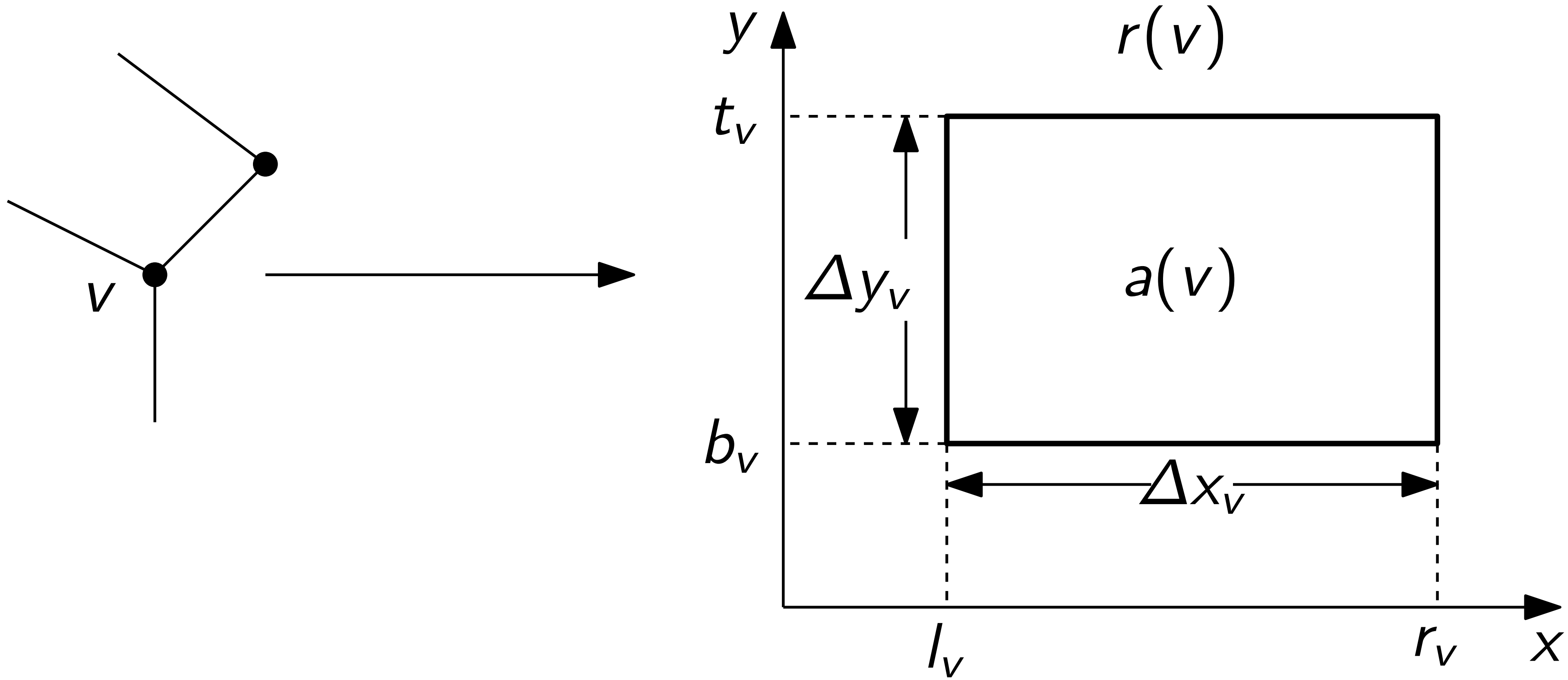
Rectangular representation



A vertex v is corresponding to a rectangle $r(v)$ with size $a(v)$

Rectangles

Rectangular representation

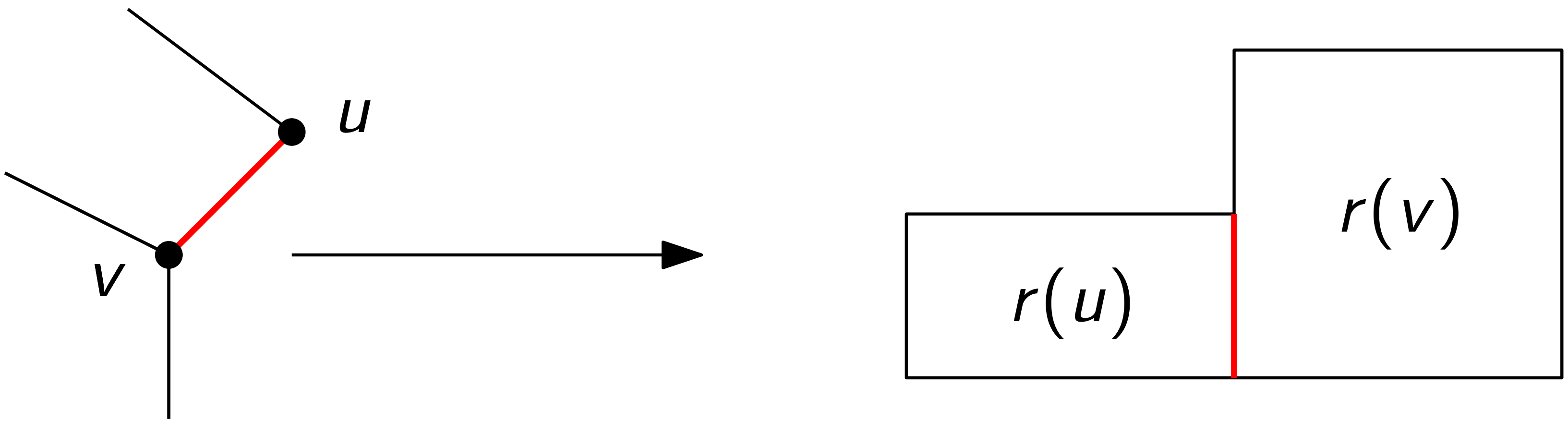


A vertex v is corresponding to a rectangle $r(v)$ with size $a(v)$

$$v \in V \xrightarrow{r : V \to R} r \in R \qquad\qquad v \in V \xrightarrow{a : V \to A} a \in A$$

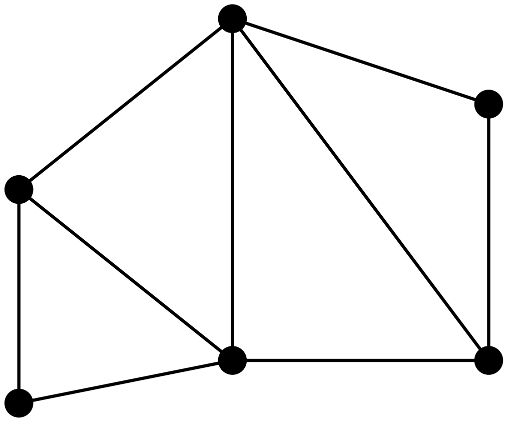
Rectangles

Rectangular representation

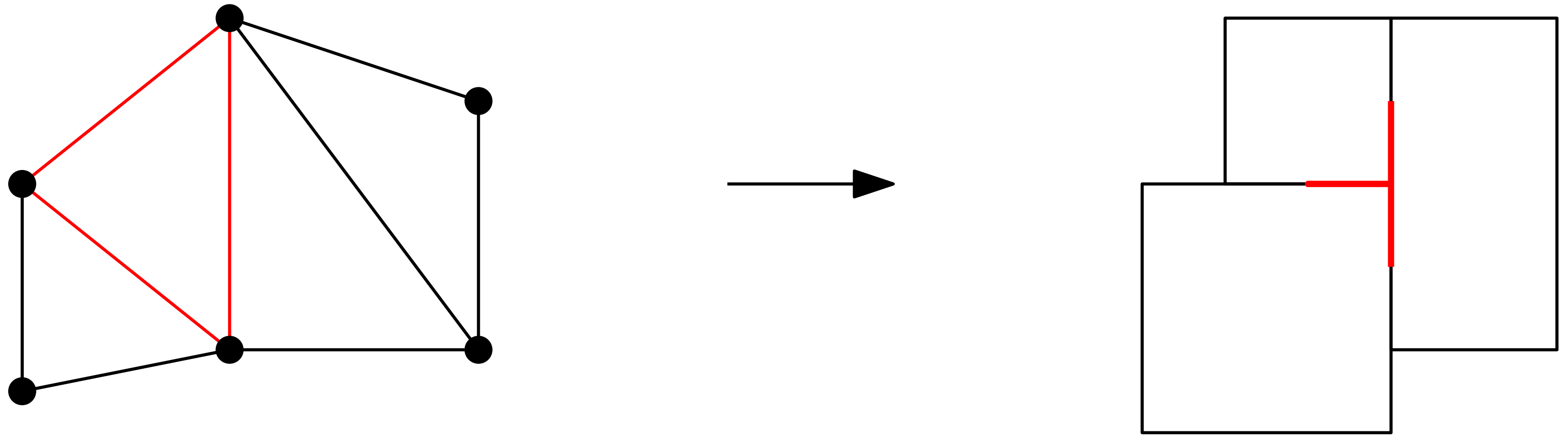


$u, v \in V$ and $uv \in E \rightarrow r(u)$ and $r(v)$ are adjacent

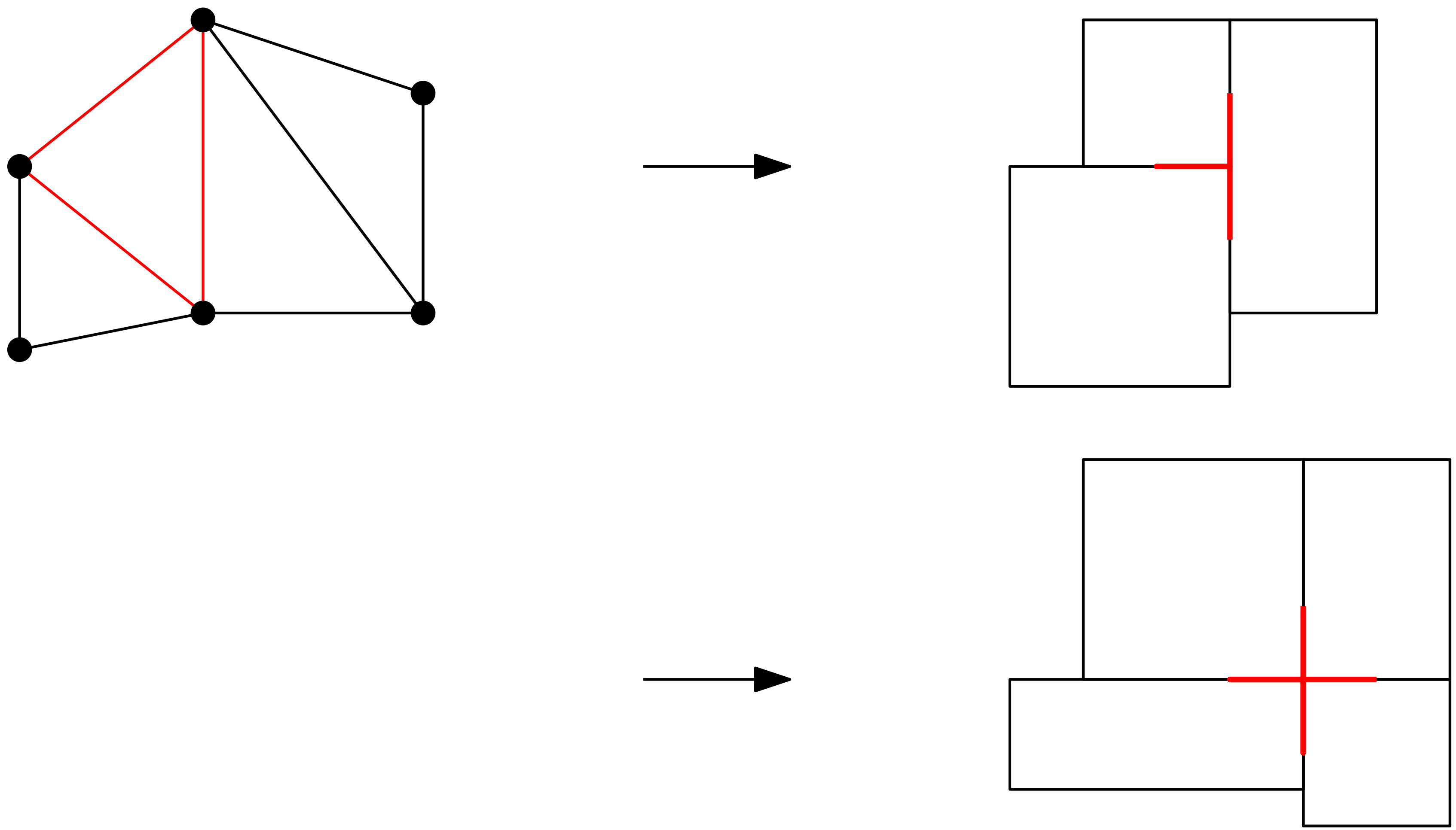
Faces and Corners



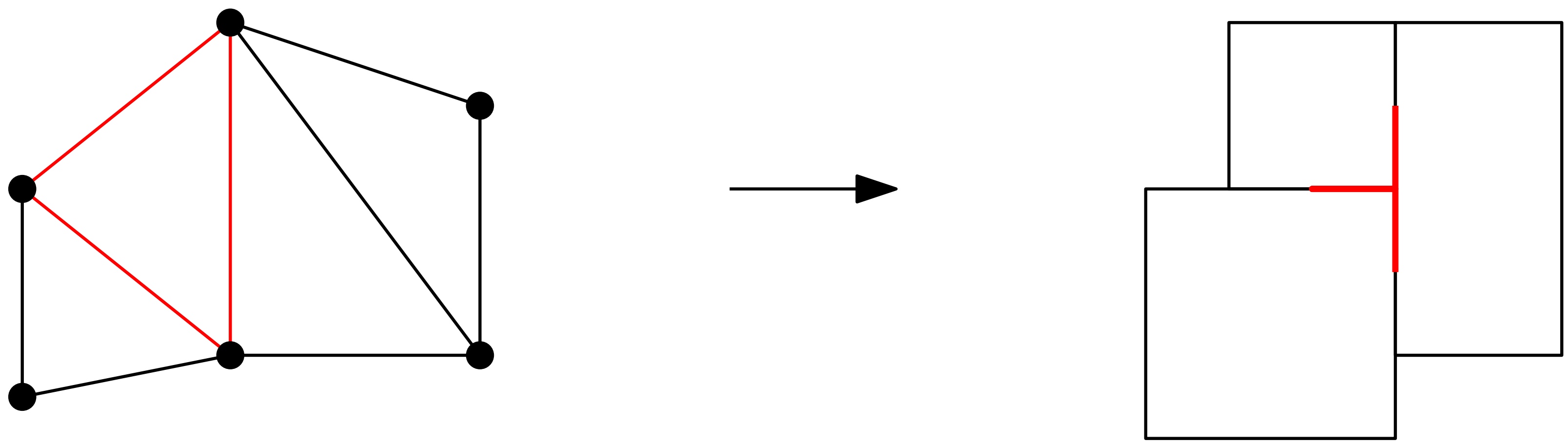
Faces and Corners

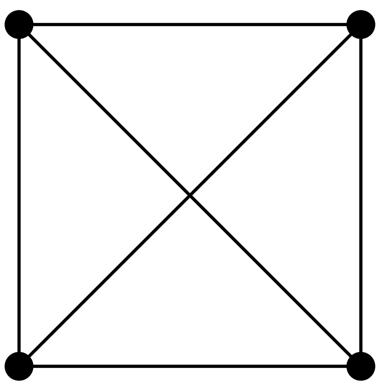


Faces and Corners

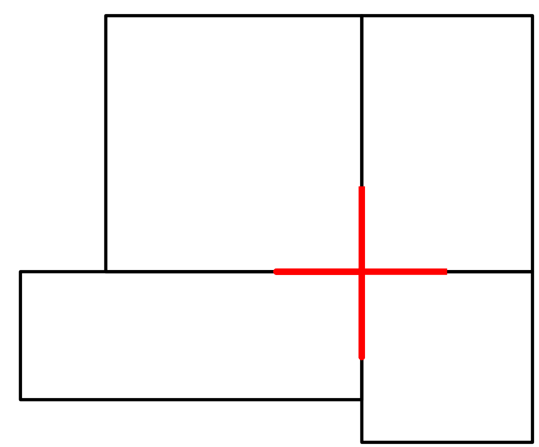


Faces and Corners

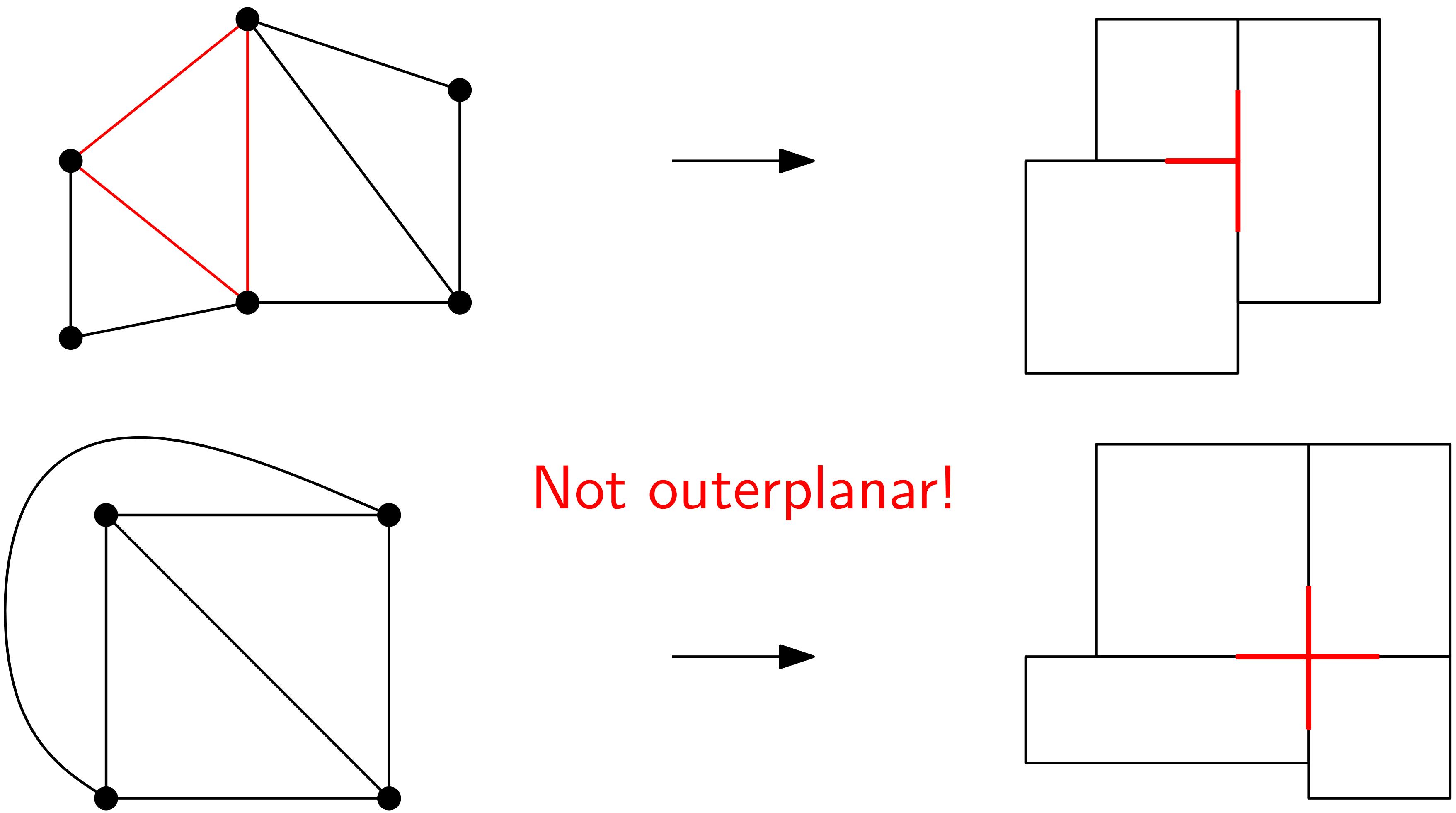




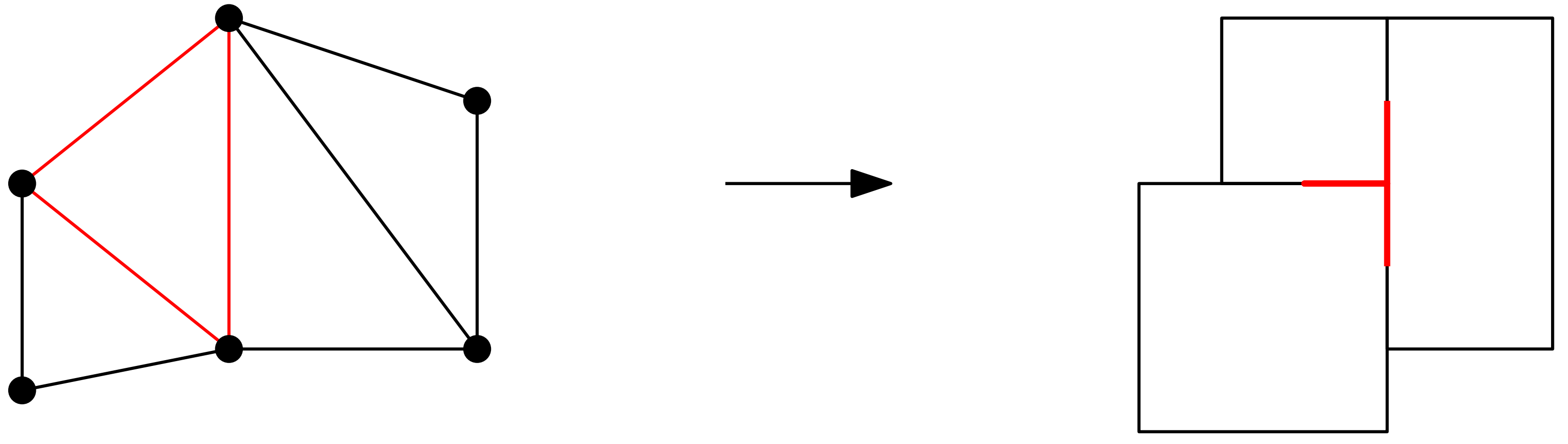
Not planar!



Faces and Corners

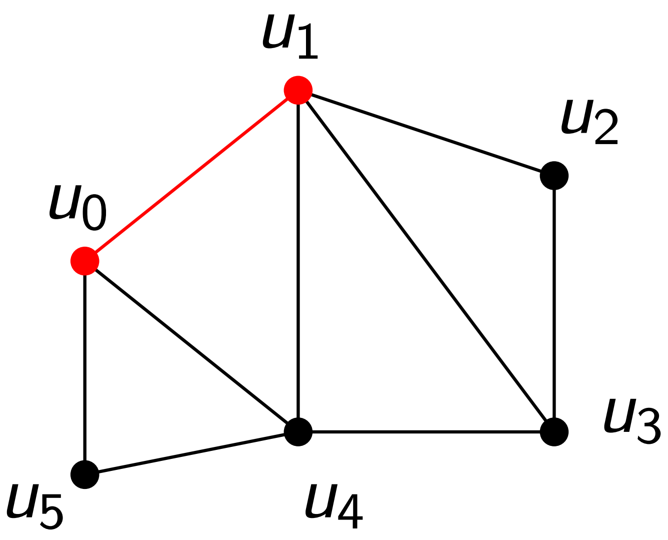


Faces and Corners

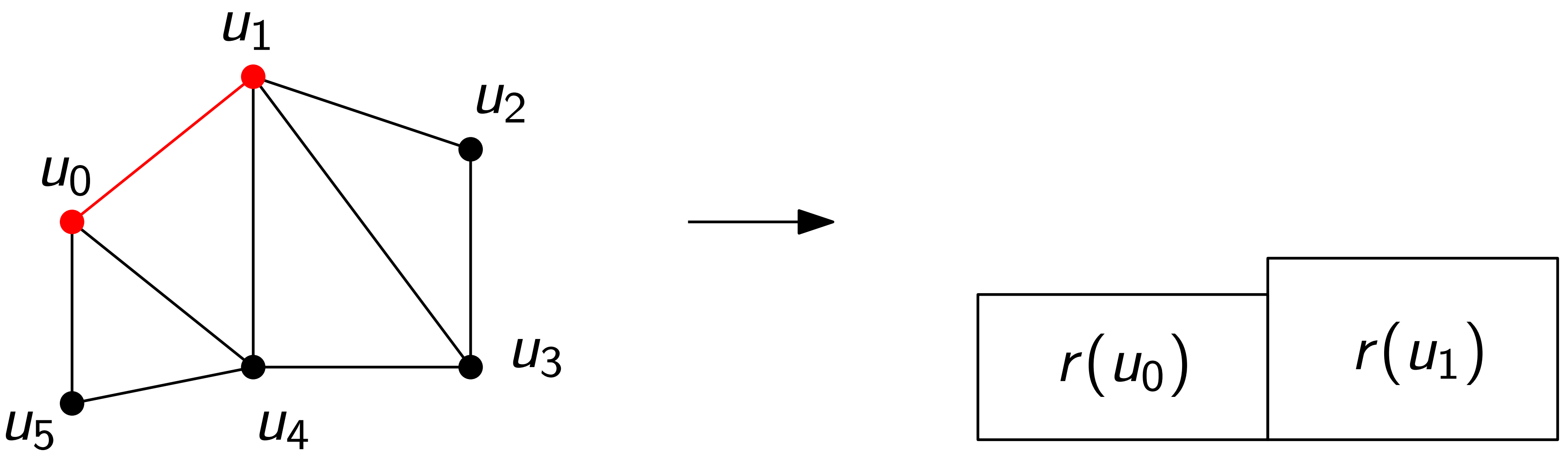


There are only corners with at most three involved rectangles!

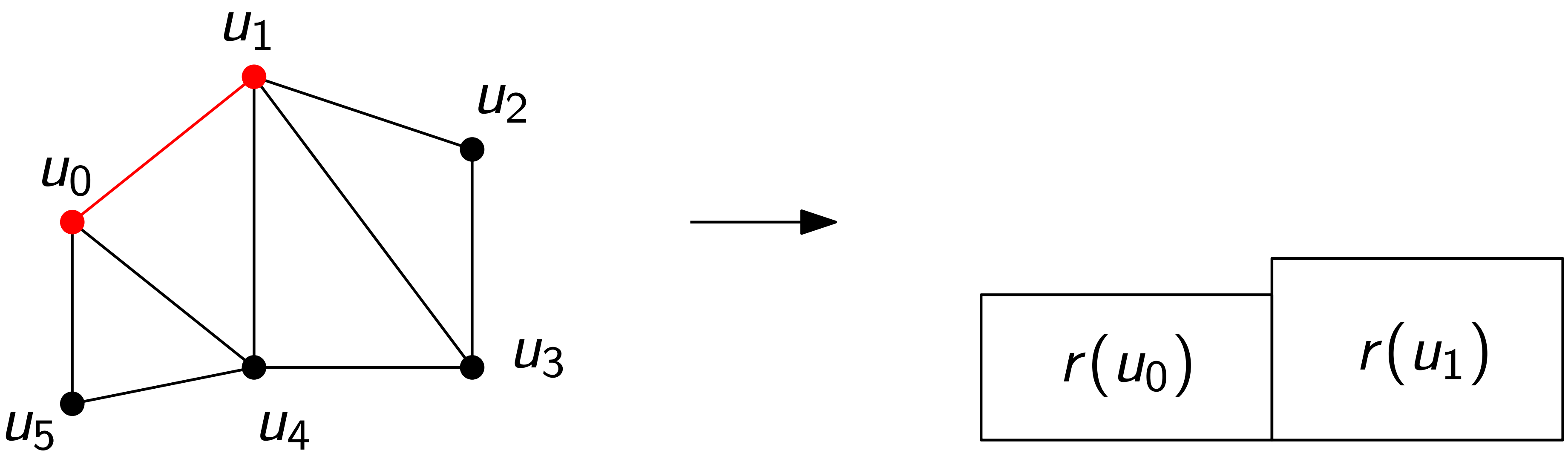
Placing the first two rectangles



Placing the first two rectangles



Placing the first two rectangles

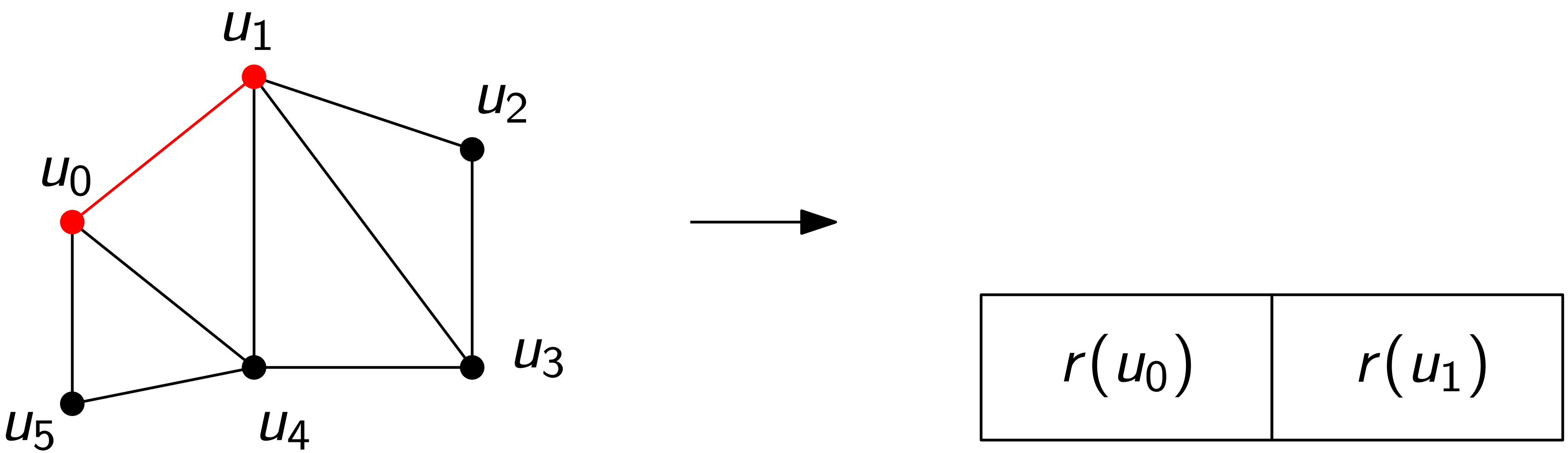


Draw the two rectangles neighboring each other with:

- the same height at the bottom
- and the same width

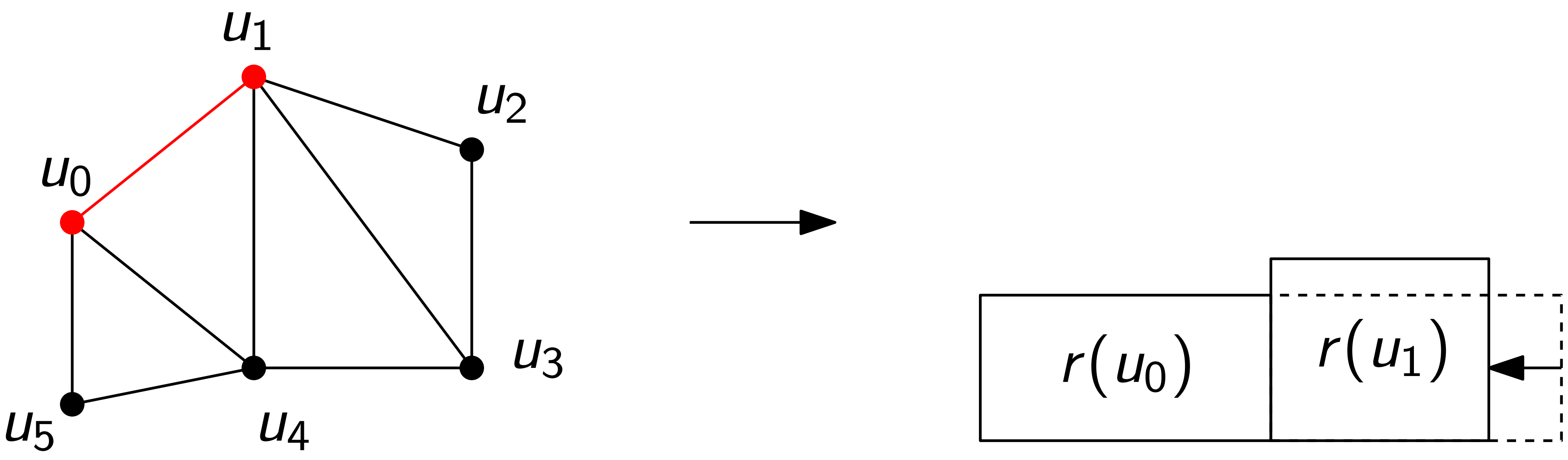
$$b_{u_0} = b_{u_1} \text{ and } \Delta x_{u_0} = \Delta x_{u_1}$$

Placing the first two rectangles



If $a(u_0) = a(u_1)$ we don't get a new edge to place the next rectangle

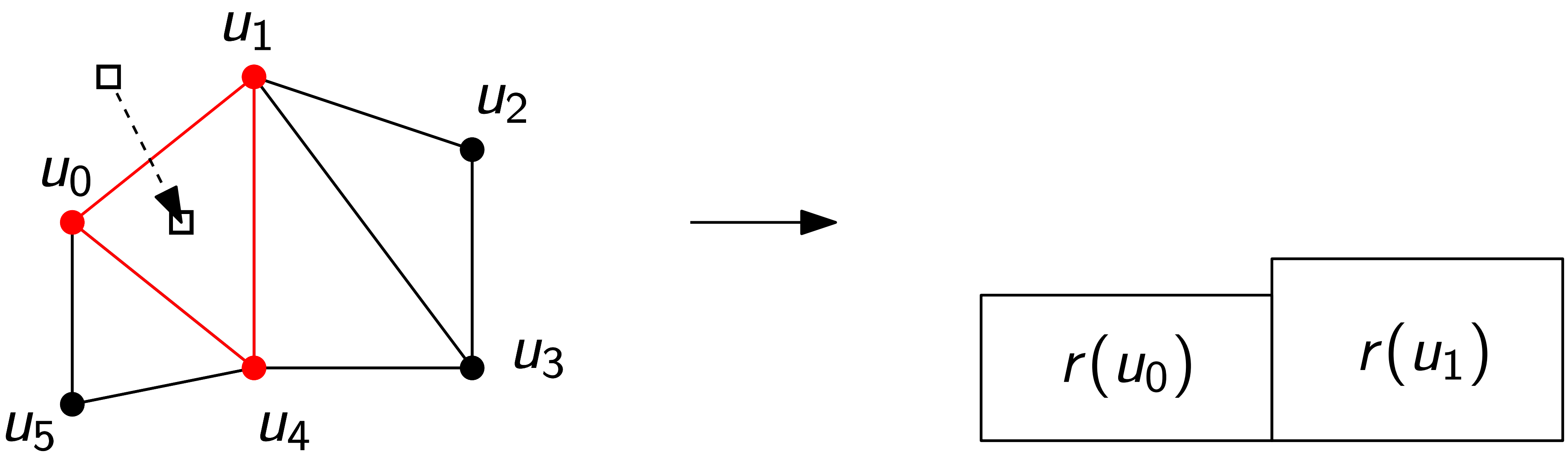
Placing the first two rectangles



If $a(u_0) = a(u_1)$ we don't get a new edge to place the next rectangle

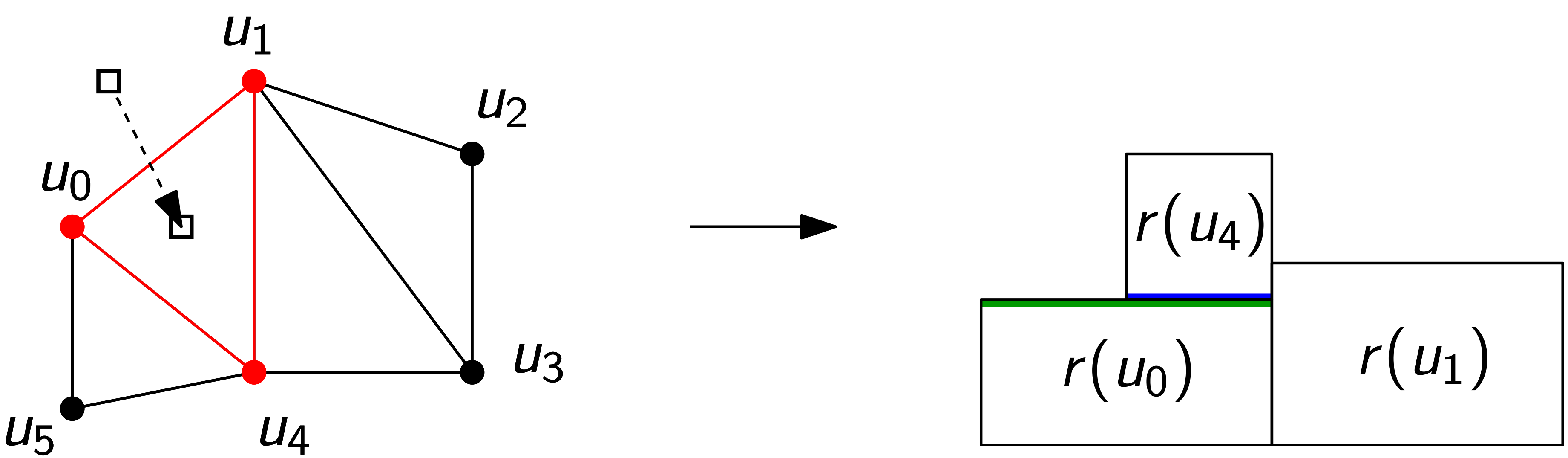
$\rightarrow$ Reduce the width of one rectangle!

Treatment of an inner facet



$r(u_4)$ is adjacent to $r(u_0)$ and $r(u_1)$

Treatment of an inner facet

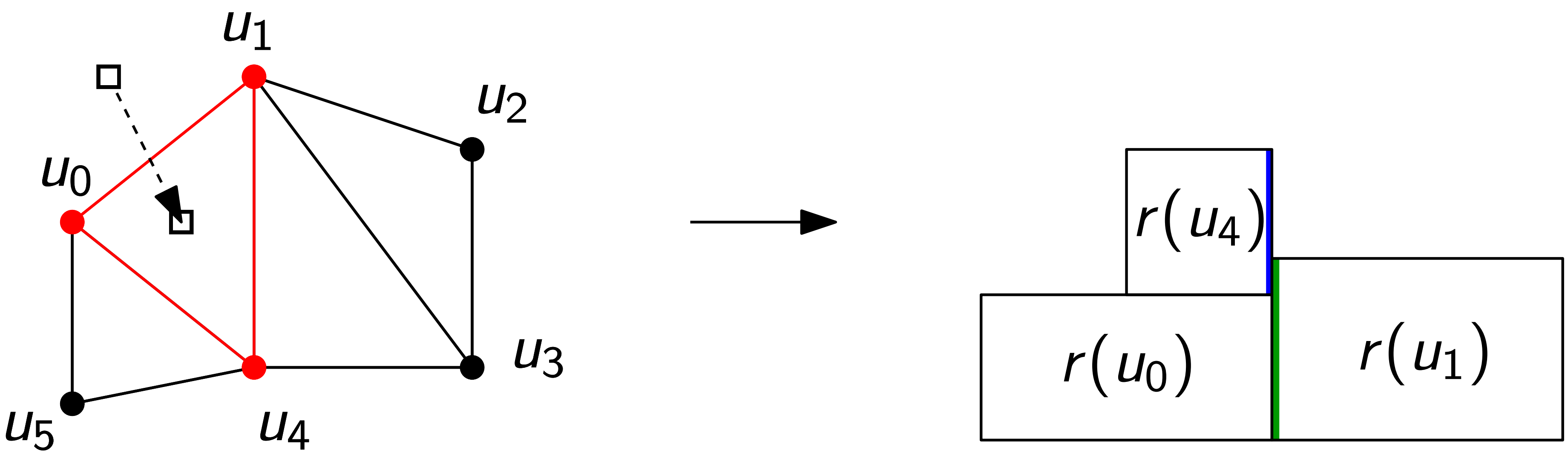


$r(u_4)$ is adjacent to $r(u_0)$ and $r(u_1)$

Place $r(u_4)$ on top of the rectangle with the lower top edge

$$b_{u_4} = t_{u_0}$$

Treatment of an inner facet

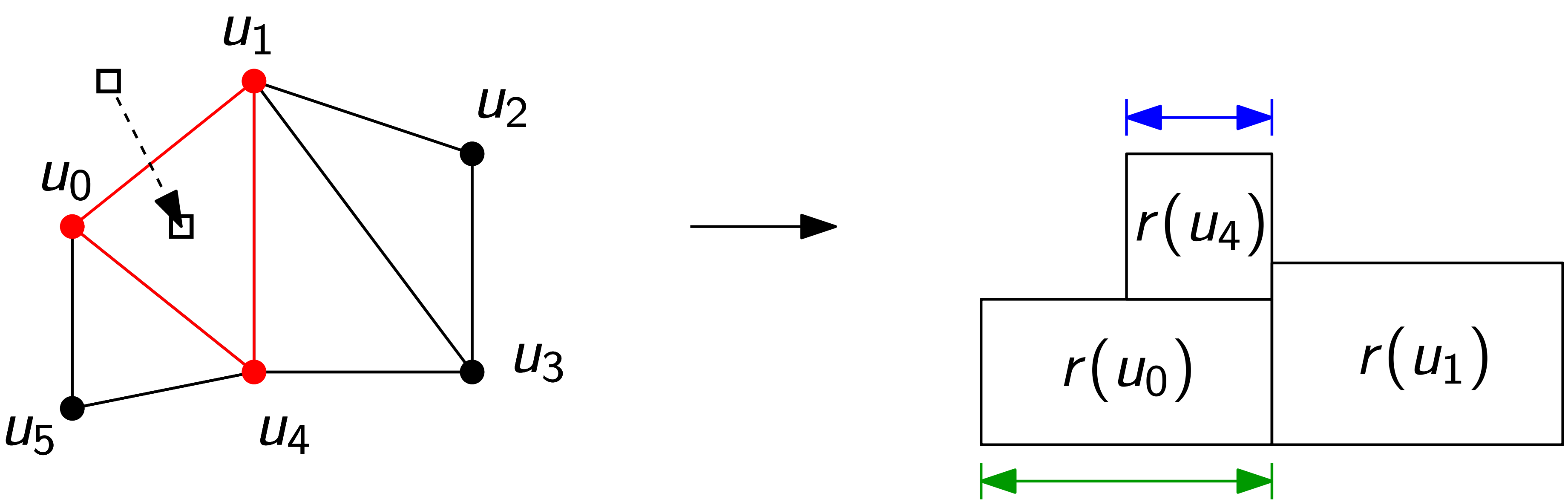


$r(u_4)$ is adjacent to $r(u_0)$ and $r(u_1)$

Place $r(u_4)$ on top of the rectangle with the lower top edge and to the left of the other one

$$b_{u_4} = t_{u_0} \text{ and } r_{u_4} = l_{u_1}$$

Treatment of an inner facet

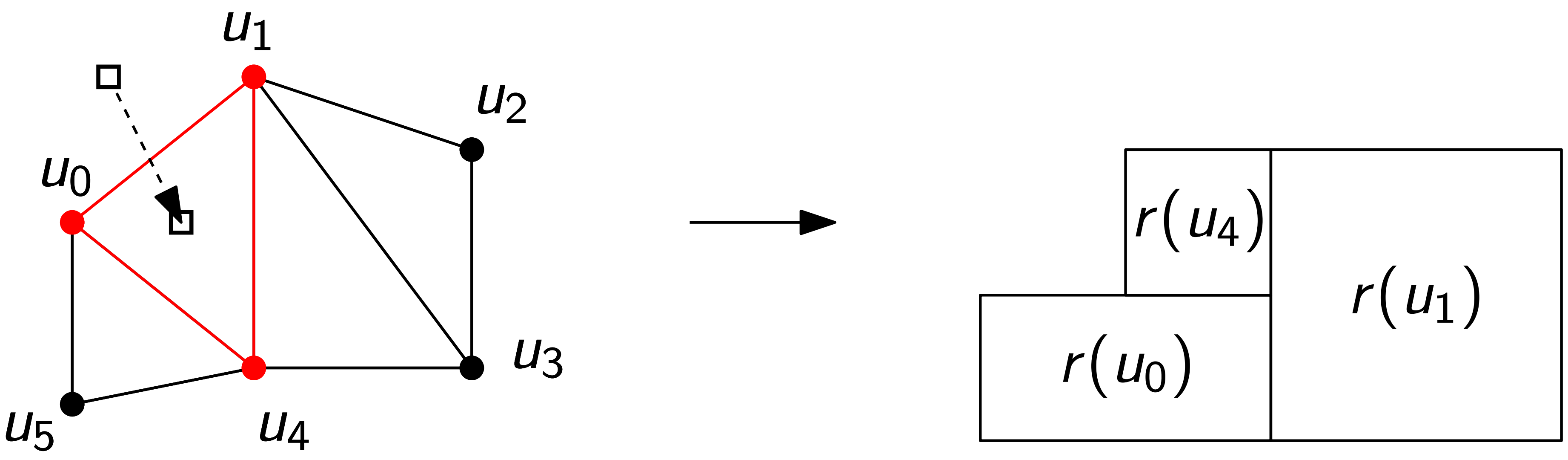


$r(u_4)$ is adjacent to $r(u_0)$ and $r(u_1)$

Leave free space for the rectangles adjacent to $r(u_0)$ and $r(u_4)$ or $r(u_1)$ and $r(u_4)$

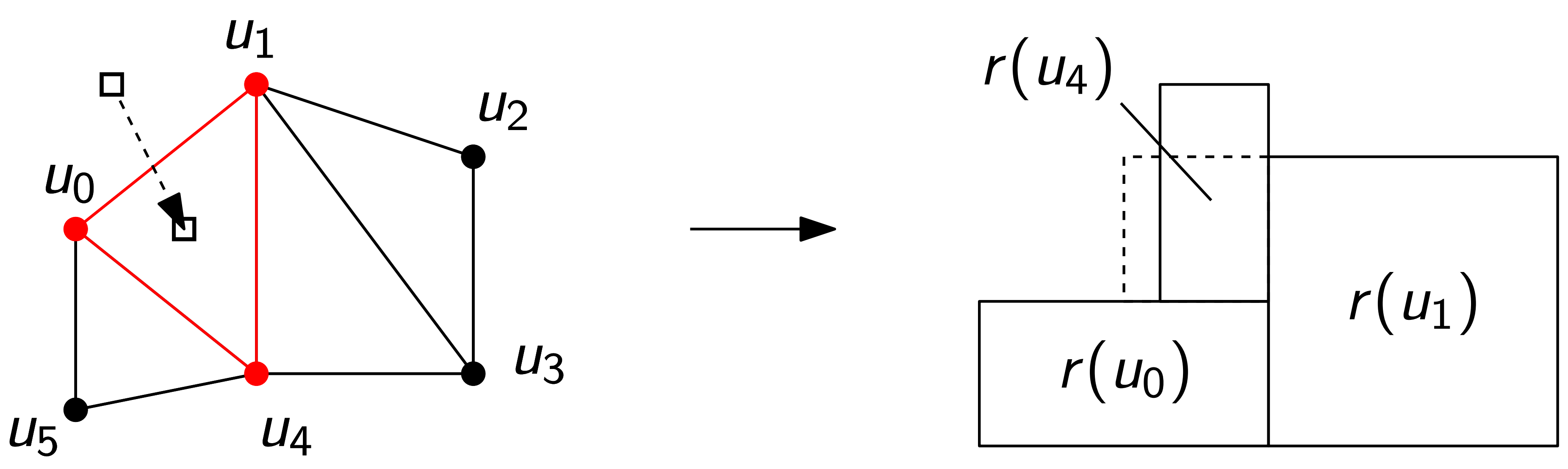
$$\Delta x_{u_4} = 0.5 \cdot Free(u_0)$$

Treatment of an inner facet



What can we do if there is no free space?

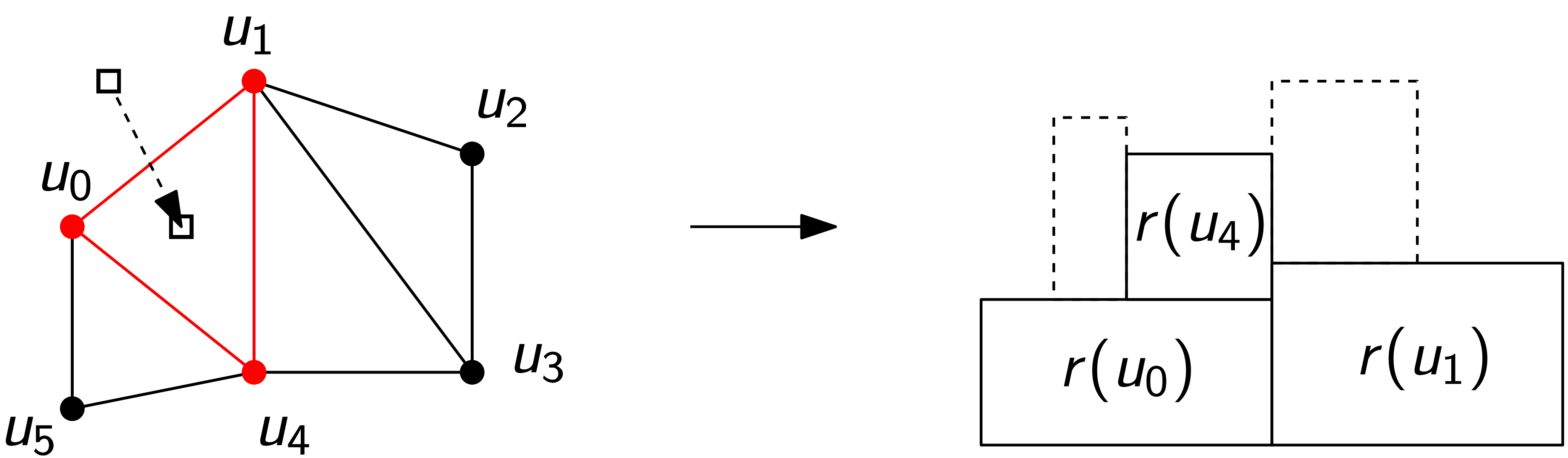
Treatment of an inner facet



What can we do if there is no free space?

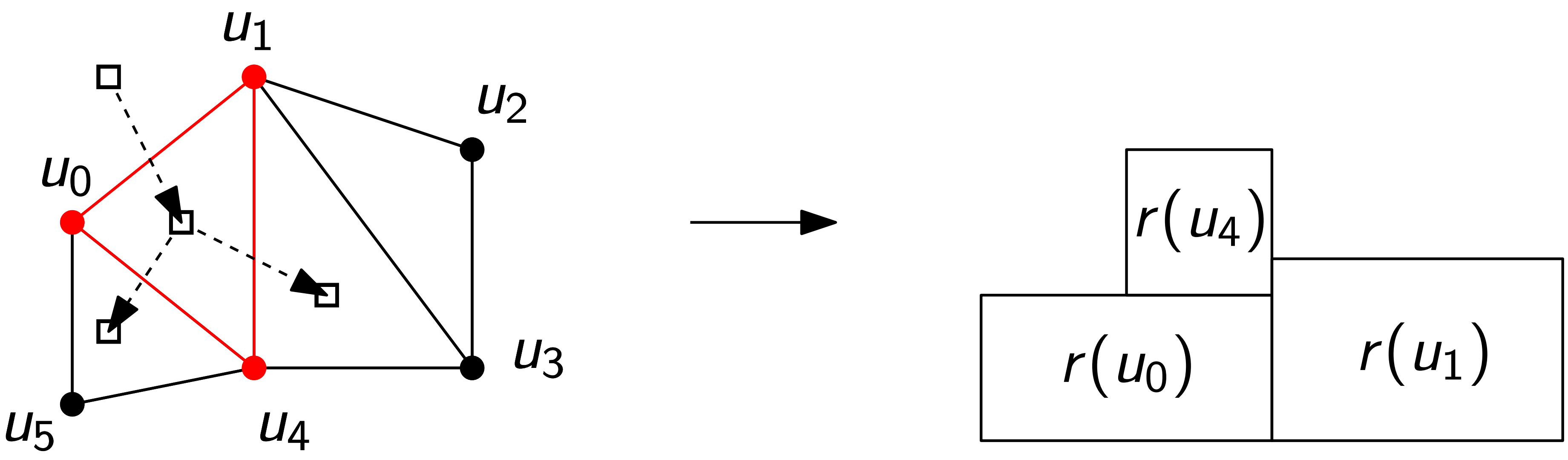
$\rightarrow$ Reduce the width of the new rectangle!

Treatment of an inner facet



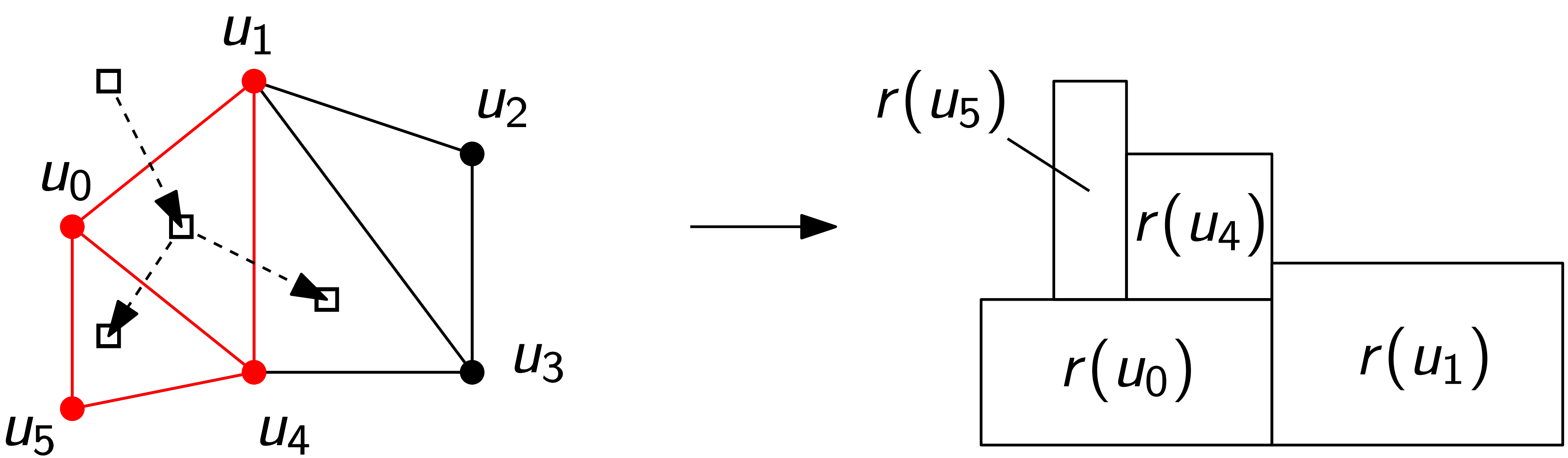
- The new Rectangle $r(u_4)$ is adjacent to $r(u_0)$ and $r(u_1)$
- Possibility to place the two rectangles neighboring $r(u_0)$ and $r(u_4)$ or $r(u_1)$ and $r(u_4)$

Finishing the graph



Choose one direction to work on first

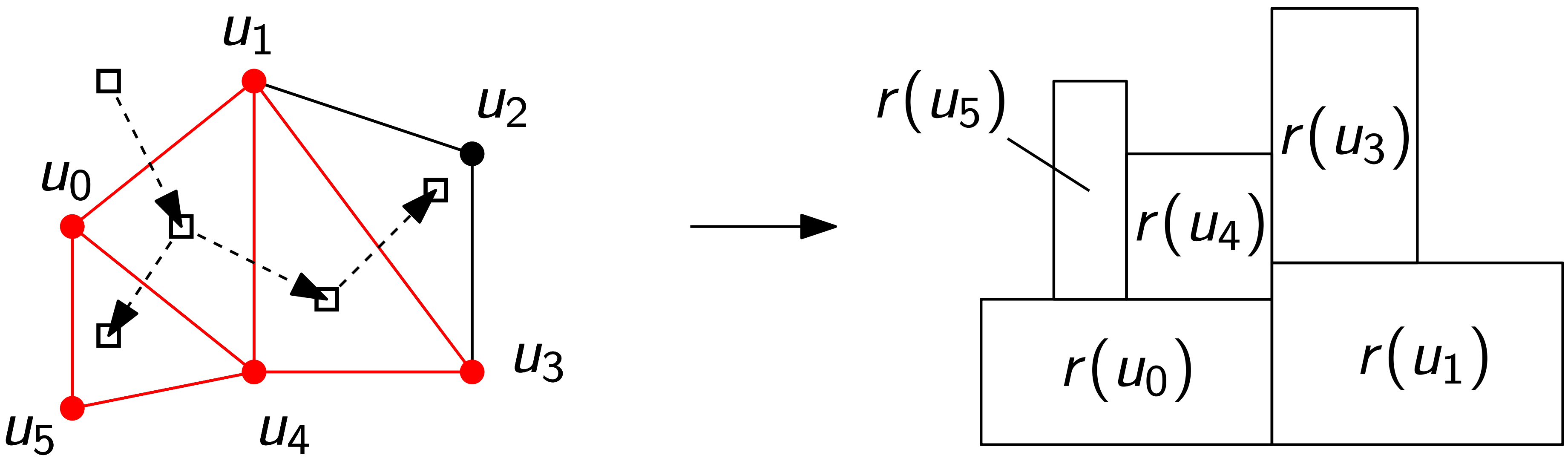
Finishing the graph



Choose one direction to work on first

- Place $r(u_5)$ next to $r(u_0)$ and $r(u_4)$

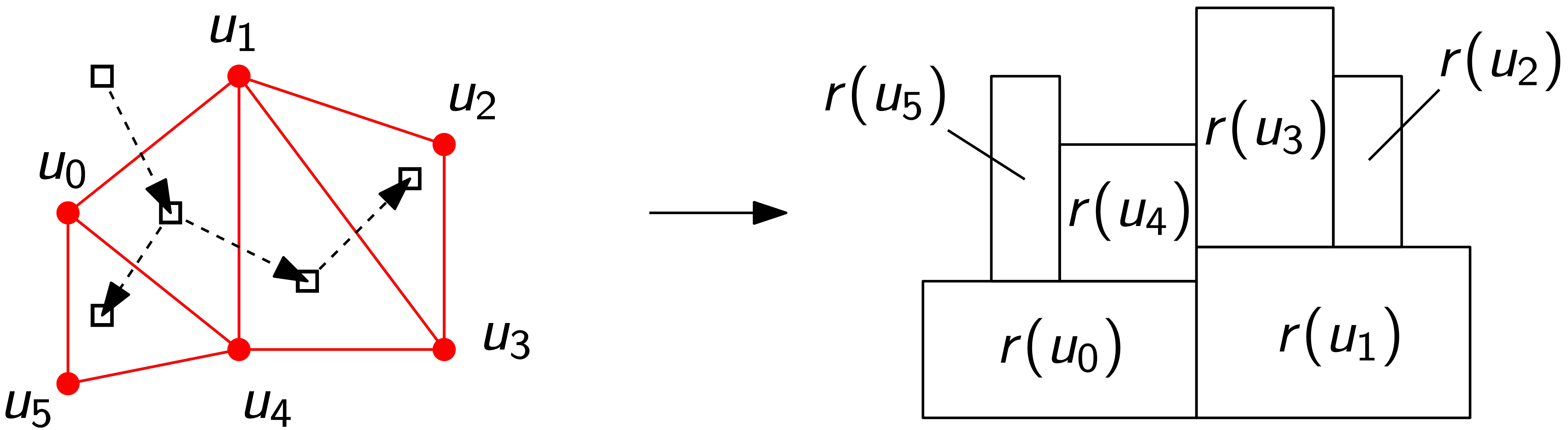
Finishing the graph



Choose one direction to work on first

- Place $r(u_5)$ next to $r(u_0)$ and $r(u_4)$
- Place $r(u_3)$ next to $r(u_1)$ and $r(u_4)$

Finishing the graph



Choose one direction to work on first

- Place $r(u_5)$ next to $r(u_0)$ and $r(u_4)$
- Place $r(u_3)$ next to $r(u_1)$ and $r(u_4)$
- Place $r(u_2)$ next to $r(u_1)$ and $r(u_3)$

Proof of correctness

The correctness is proved by induction on the number of rectangles placed so far.

Proof of correctness

The correctness is proved by induction on the number of rectangles placed so far.

Graph $G_k = (V_k, E_k)$ is a subgraph of the graph $G = (V, E)$ to be drawn with

Proof of correctness

The correctness is proved by induction on the number of rectangles placed so far.

Graph $G_k = (V_k, E_k)$ is a subgraph of the graph $G = (V, E)$ to be drawn with

- $V_k \subseteq V$ and
- $E_k = \{(u, v) \in E \mid u, v \in V_k\}$
- and G_k has a rectangular representation with k placed rectangles.

Proof of correctness

The correctness is proved by induction on the number of rectangles placed so far.

Graph $G_k = (V_k, E_k)$ is a subgraph of the graph $G = (V, E)$ to be drawn with

- $V_k \subseteq V$ and
- $E_k = \{(u, v) \in E \mid u, v \in V_k\}$
- and G_k has a rectangular representation with k placed rectangles.

Invariant:
There is a free corner for the next rectangle $r(w)$ to be placed with $w \in V \setminus V_k$, in such a way that it is adjacent to its predecessors $r(u)$ and $r(v)$ with $u, v \in V_k$.

Proof of correctness

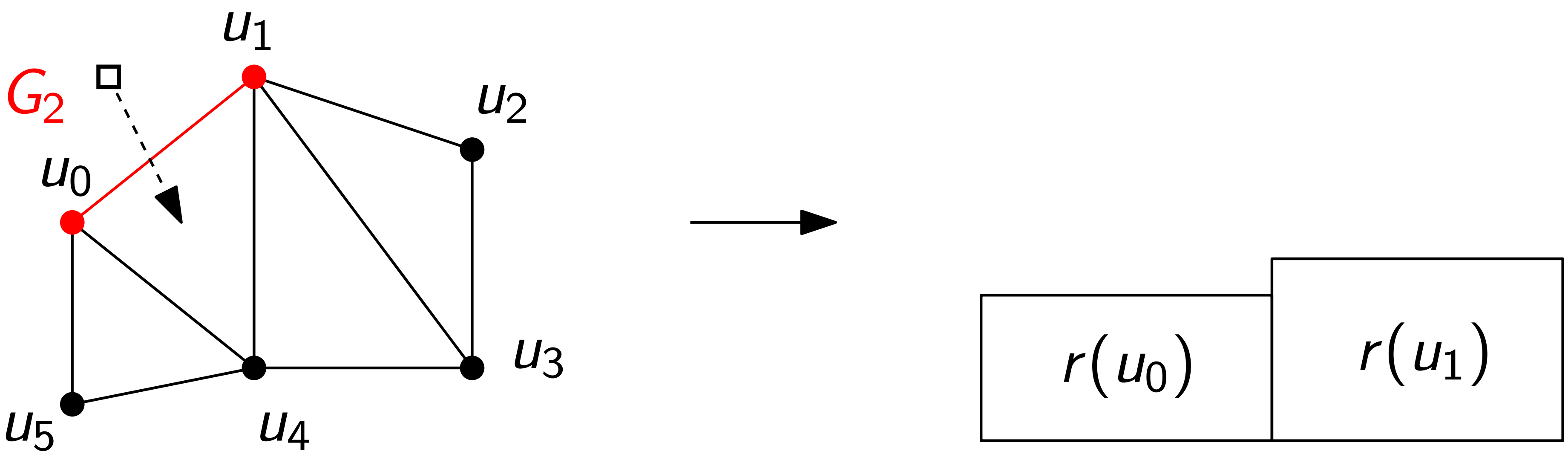
The correctness is proved by induction on the number of rectangles placed so far.

Graph $G_k = (V_k, E_k)$ is a subgraph of the graph $G = (V, E)$ to be drawn with

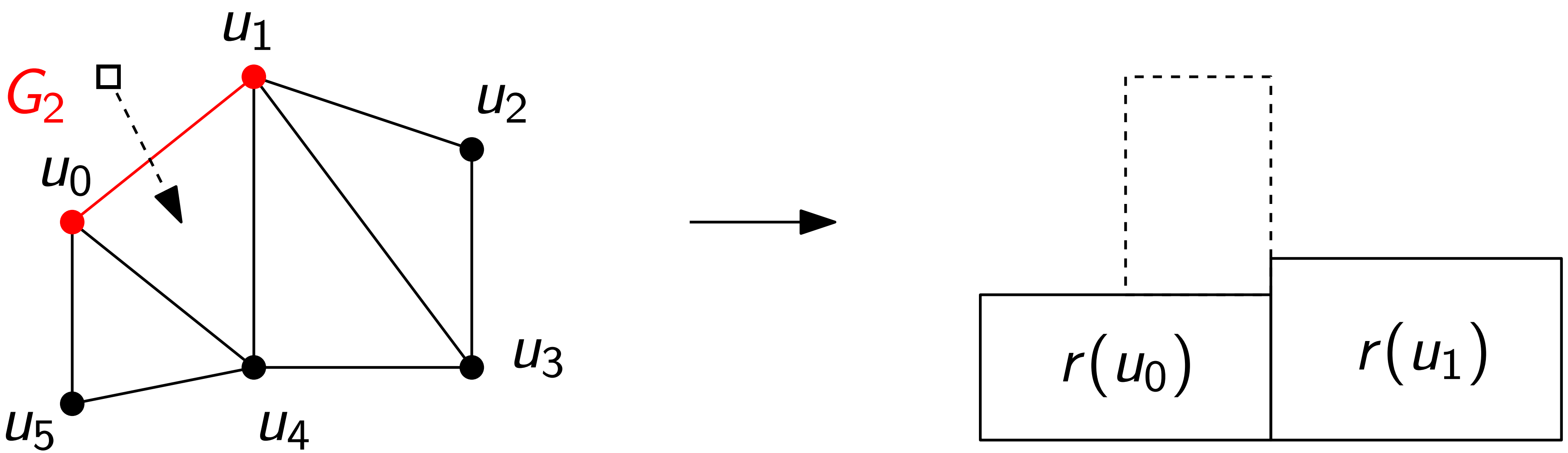
- $V_k \subseteq V$ and
- $E_k = \{(u, v) \in E \mid u, v \in V_k\}$
- and G_k has a rectangular representation with k placed rectangles.

Invariant:
There is a free corner for the next rectangle $r(w)$ to be placed with $w \in V \setminus V_k$, in such a way that it is adjacent to its predecessors $r(u)$ and $r(v)$ with $u, v \in V_k$.

Proof of correctness

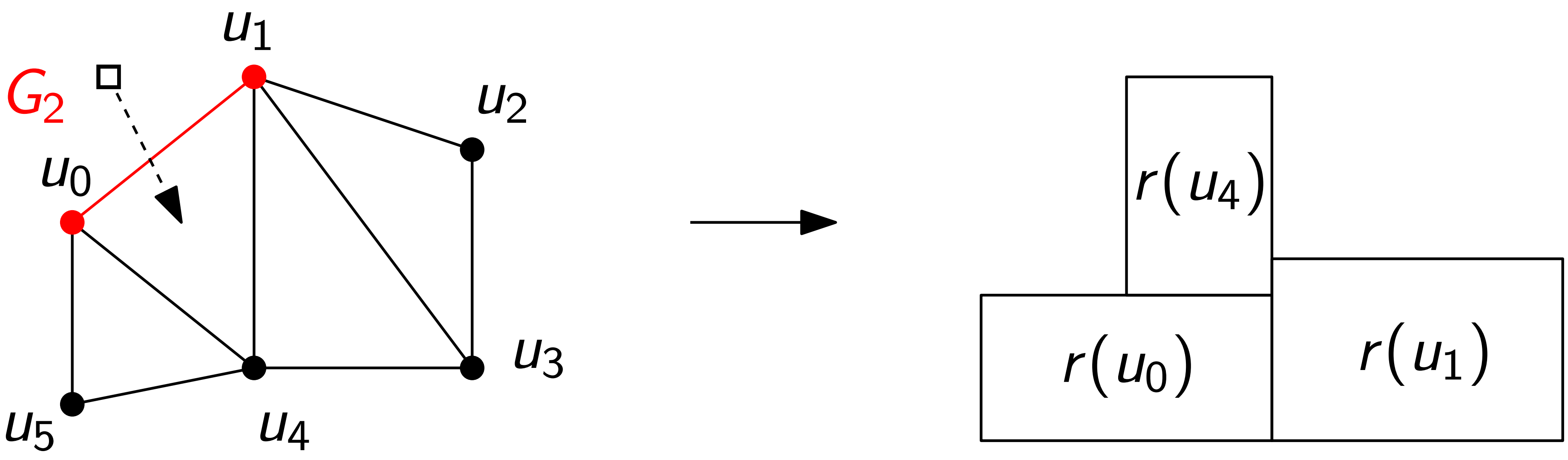


Proof of correctness



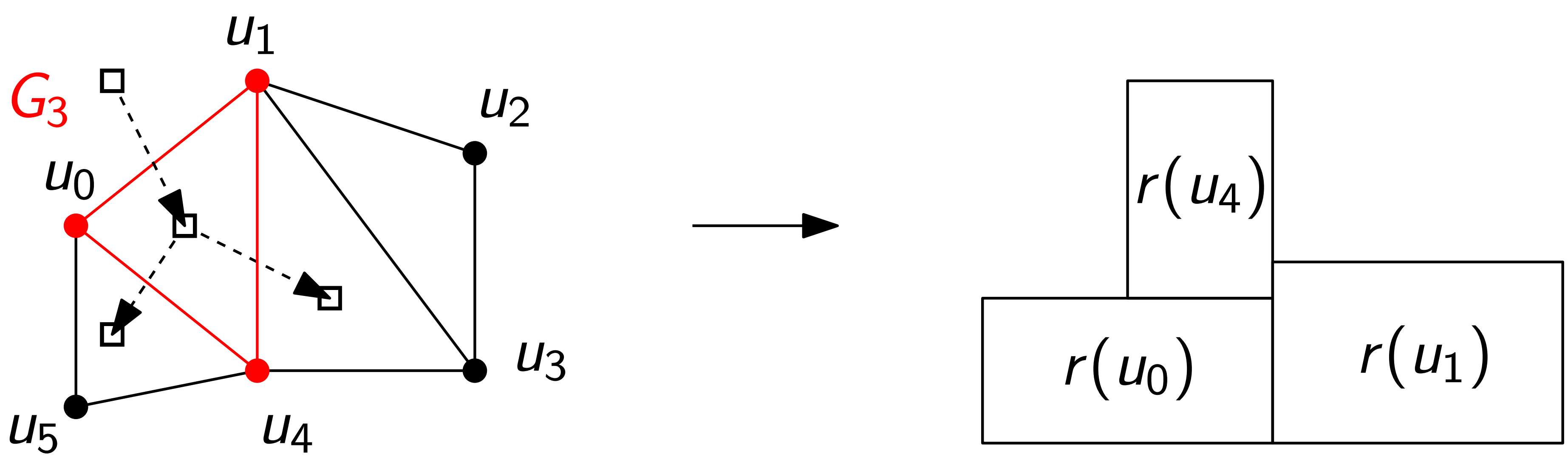
The invariant is fulfilled because we have one corner where $r(u_4)$ can be placed adjacent to its predecessors $r(u_0)$ and $r(u_1)$

Proof of correctness



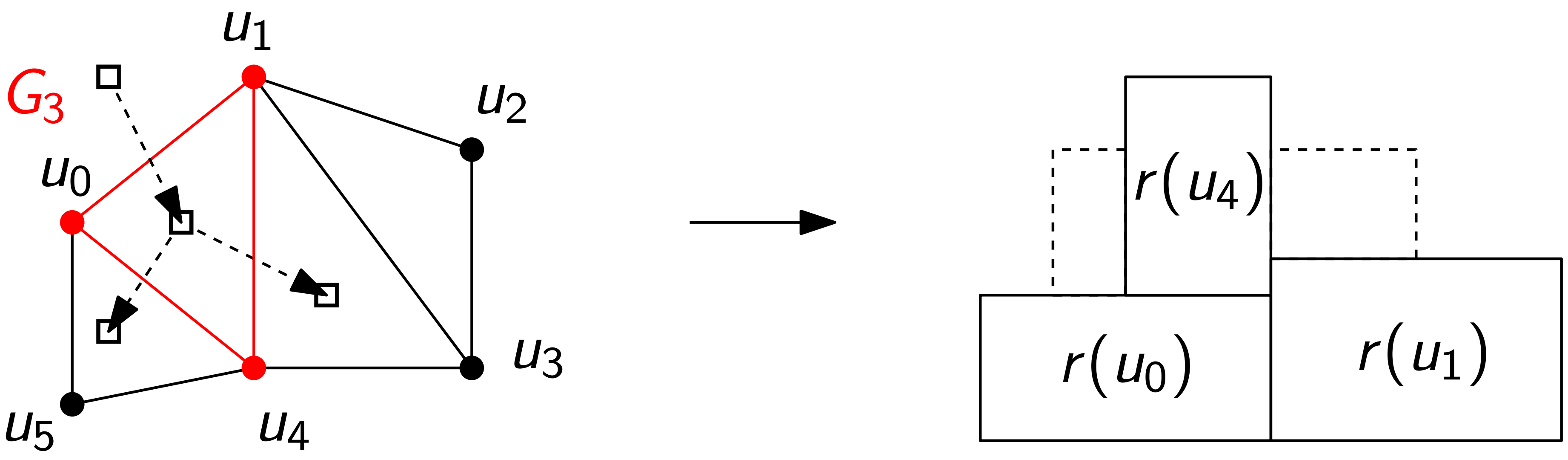
The invariant is fulfilled because we have one corner where $r(u_4)$ can be placed adjacent to its predecessors $r(u_0)$ and $r(u_1)$

Proof of correctness



The invariant is fulfilled because we have one corner where $r(u_4)$ can be placed adjacent to its predecessors $r(u_0)$ and $r(u_1)$

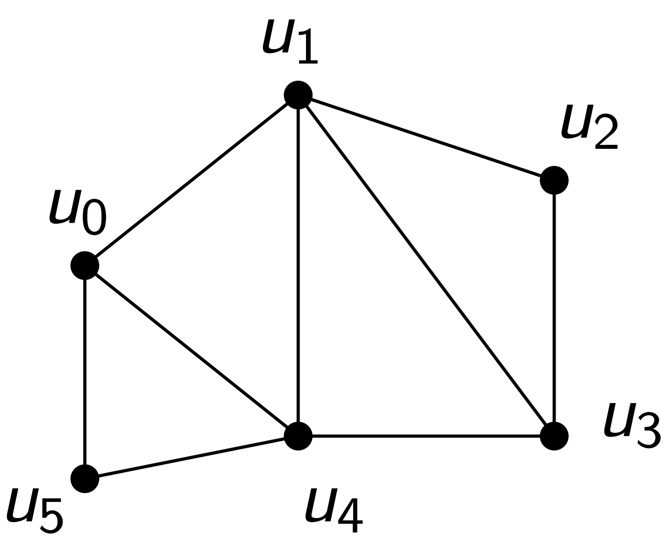
Proof of correctness



The invariant is fulfilled because we have one corner where $r(u_4)$ can be placed adjacent to its predecessors $r(u_0)$ and $r(u_1)$

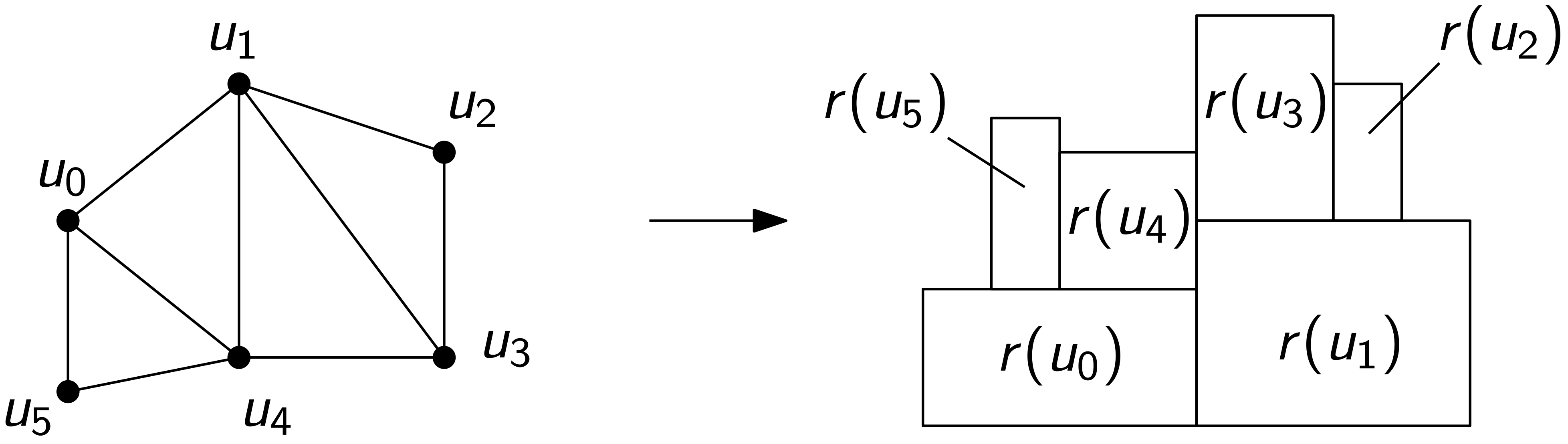
There is a corner for $r(u_5)$ and $r(u_3)$ where they can be placed adjacent to their predecessors.

Outerplanar Graphs



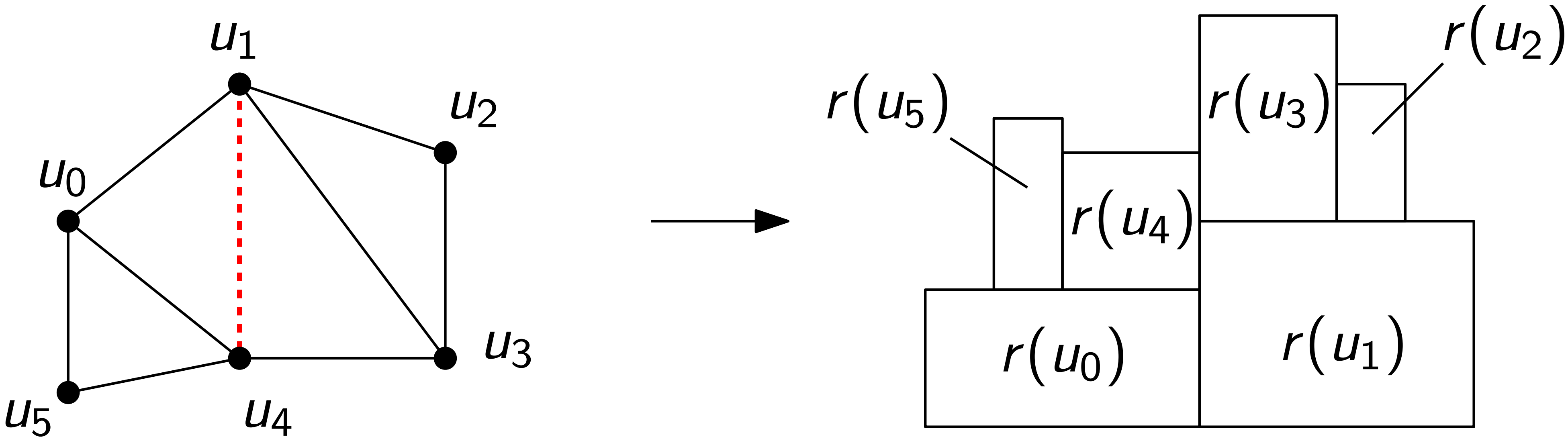
Outerplanar graph: all vertices belong to the unbounded face

Outerplanar Graphs



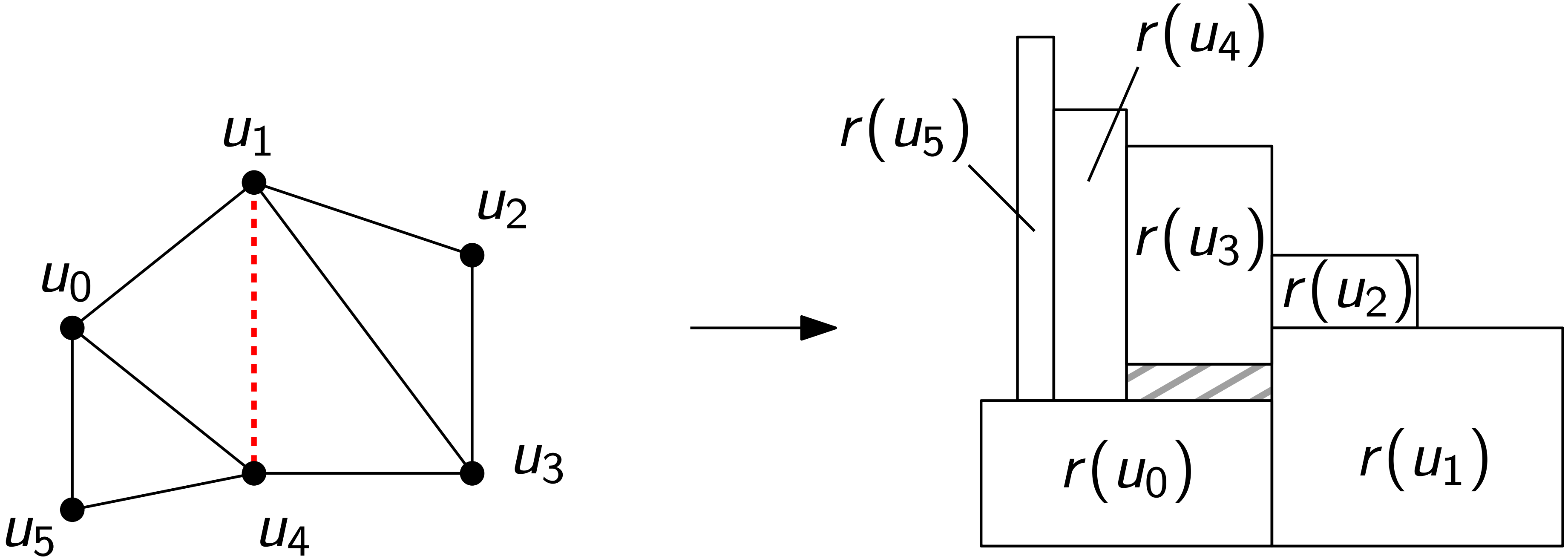
Outerplanar graph: all vertices belong to the unbounded face

Outerplanar Graphs



Outerplanar graph: all vertices belong to the unbounded face but not all faces are triangles!

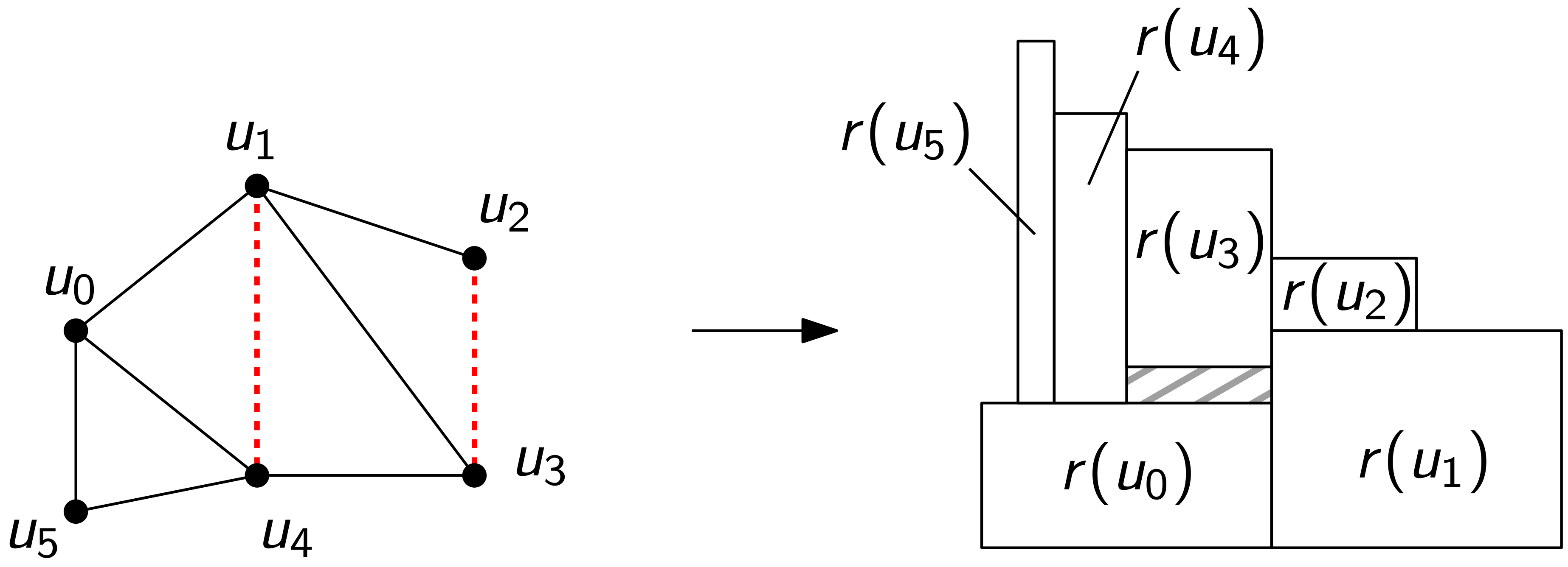
Outerplanar Graphs



Outerplanar graph: all vertices belong to the unbounded face but not all faces are triangles!

$r(u_1)$ and $r(u_4)$ are not adjacent anymore

Outerplanar Graphs

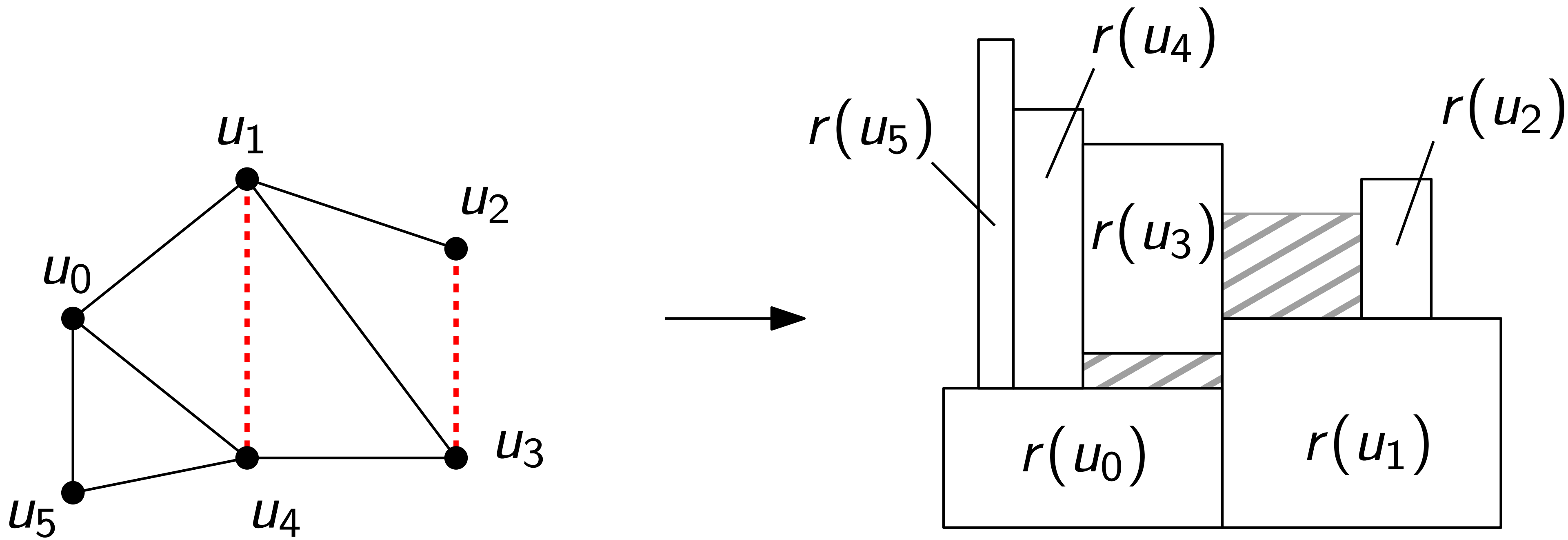


Outerplanar graph: all vertices belong to the unbounded face but not all faces are triangles!

$r(u_1)$ and $r(u_4)$ are not adjacent anymore

There are vertices that are connected with only one edge

Outerplanar Graphs

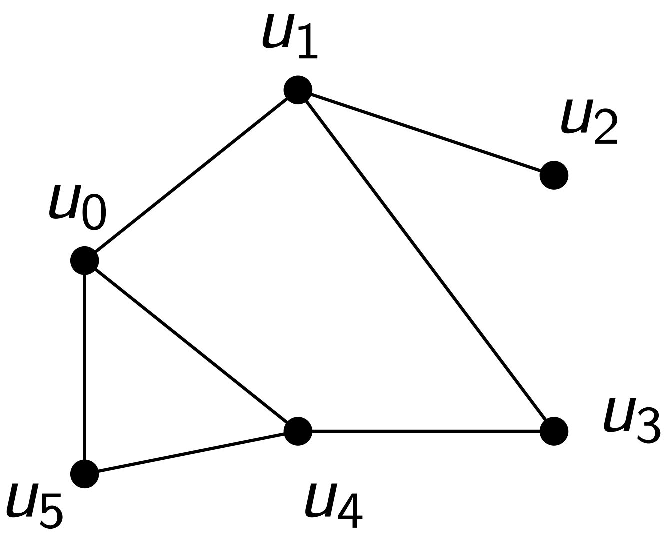


Outerplanar graph: all vertices belong to the unbounded face but not all faces are triangles!

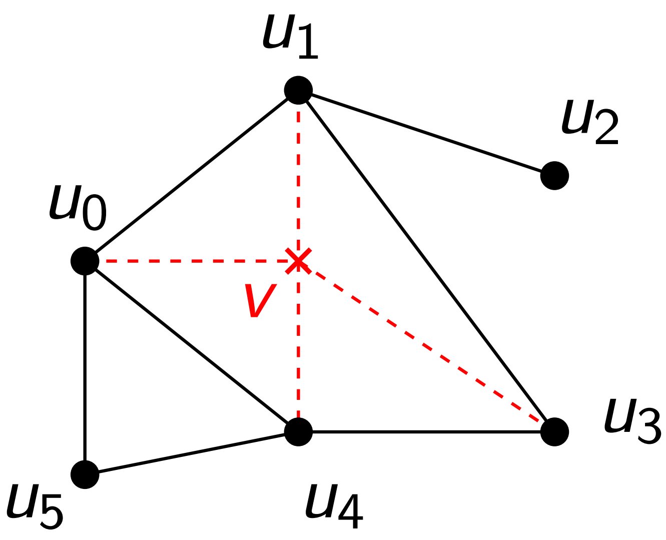
$r(u_1)$ and $r(u_4)$ are not adjacent anymore

There are vertices that are connected with only one edge

Dummy-nodes

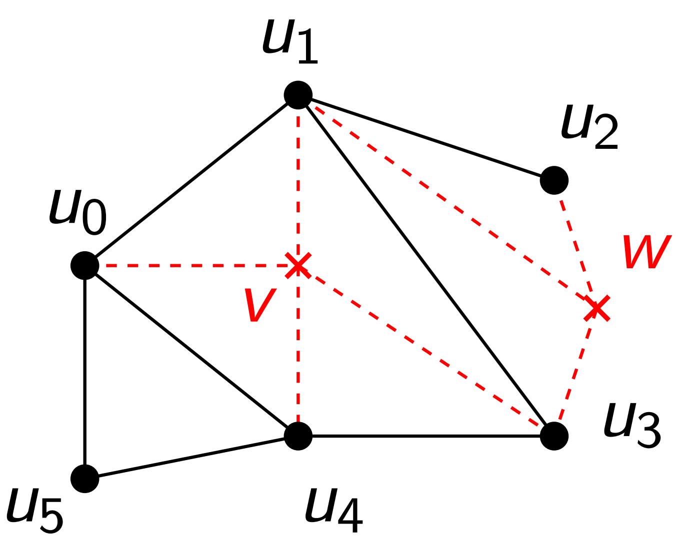


Dummy-nodes



Insert an inner dummy-node $v \in V_i$ for every not triangular inner face and connect it to the surrounding vertices with $vu \in E_i$.

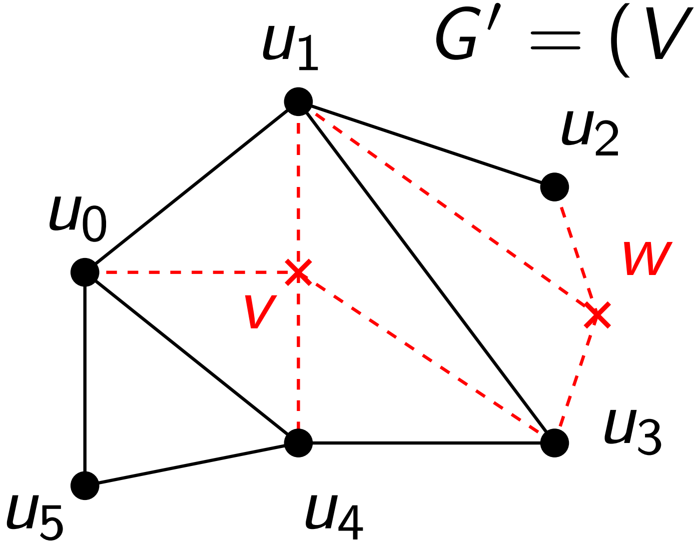
Dummy-nodes



Insert an inner dummy-node $v \in V_i$ for every not triangular inner face and connect it to the surrounding vertices with $vu \in E_i$.

Insert an outer dummy-node $w \in V_a$ for every separating vertex in the graph and add the edges $wu \in E_a$.

Dummy-nodes

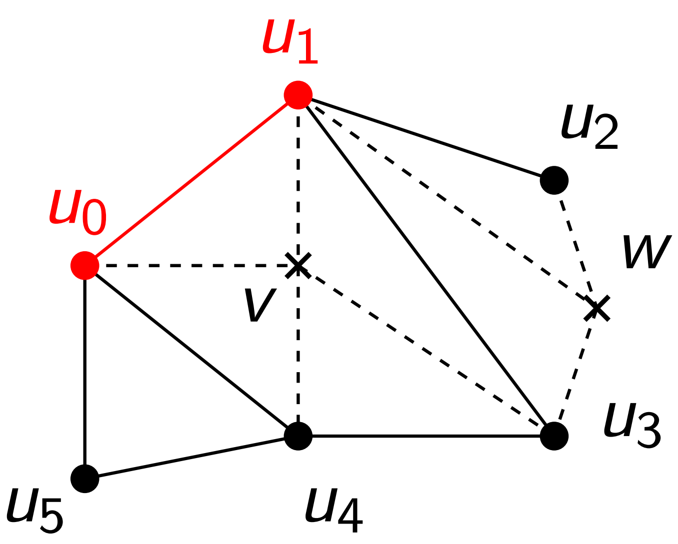


$$G' = (V', E') = (V \cup V_i \cup V_a, E \cup E_i \cup E_a)$$

Insert an inner dummy-node $v \in V_i$ for every not triangular inner face and connect it to the surrounding vertices with $vu \in E_i$.

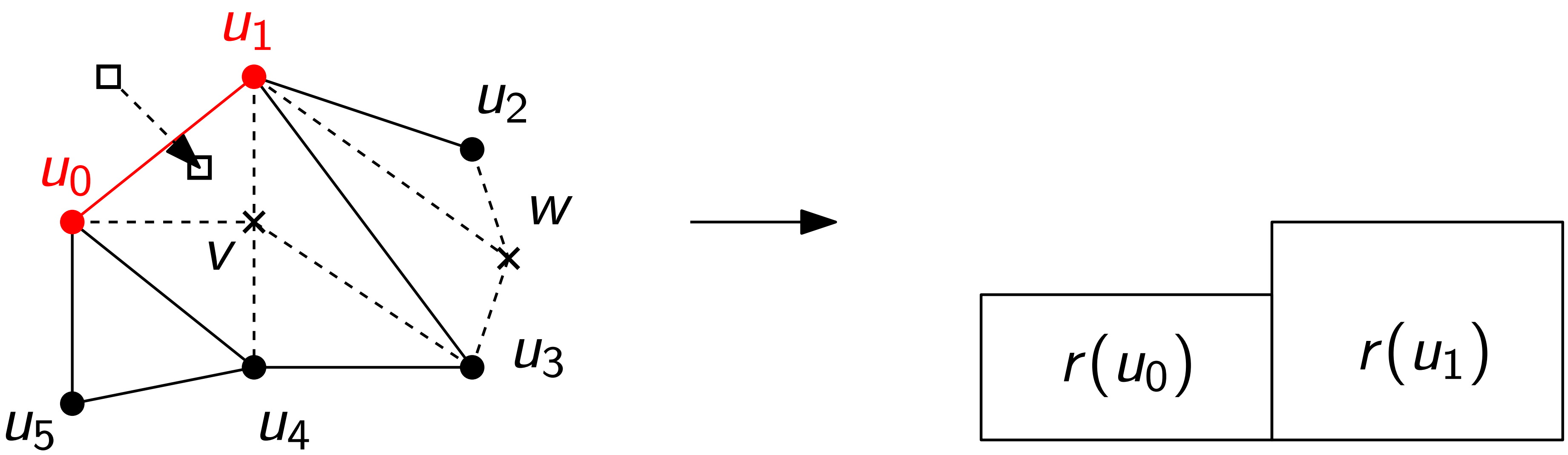
Insert an outer dummy-node $w \in V_a$ for every separating vertex in the graph and add the edges $wu \in E_a$.

Placing the first two rectangles



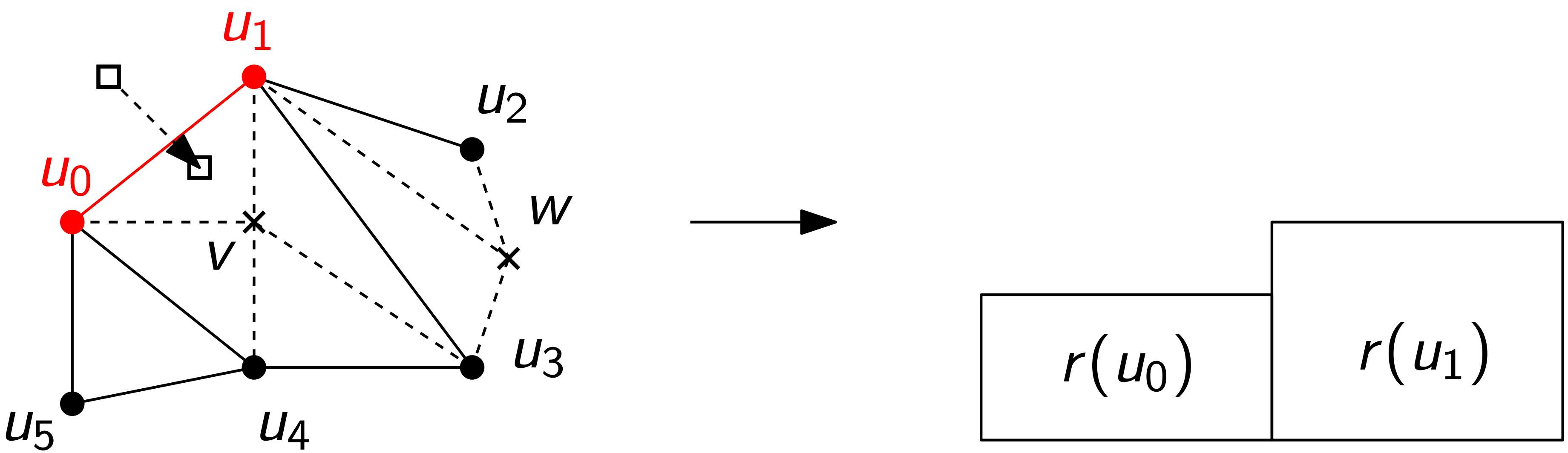
Start with an edge in $E = E' \setminus \{E_a \cup E_i\}$

Placing the first two rectangles



Start with an edge in $E = E' \setminus \{E_a \cup E_i\}$

Placing the first two rectangles

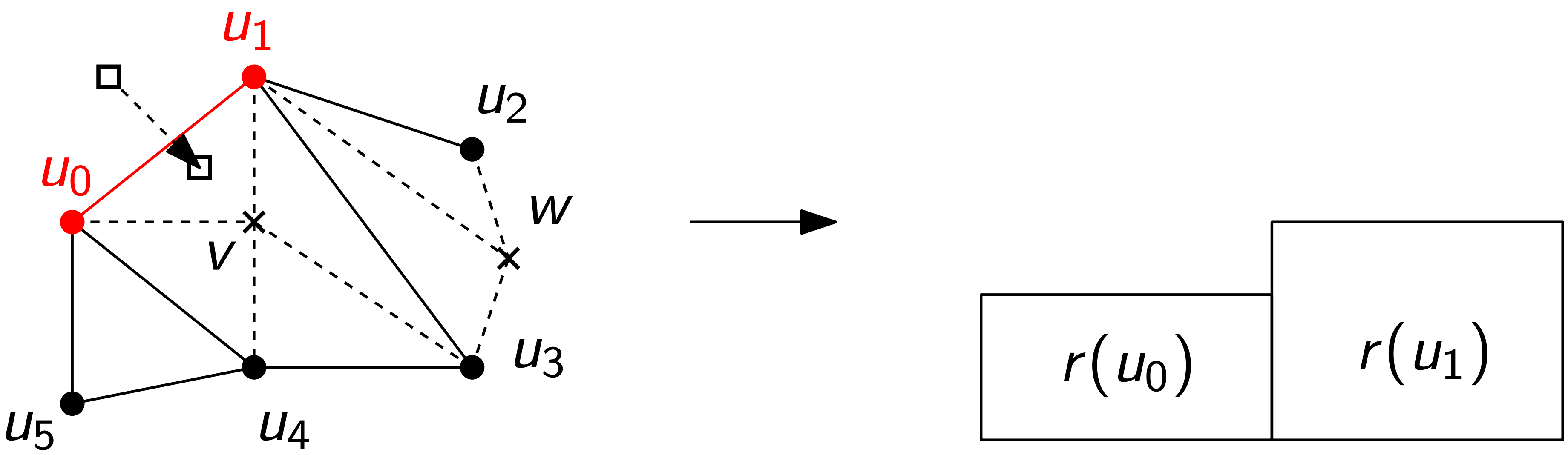


Start with an edge in $E = E' \setminus \{E_a \cup E_i\}$

Differentiate between three cases:

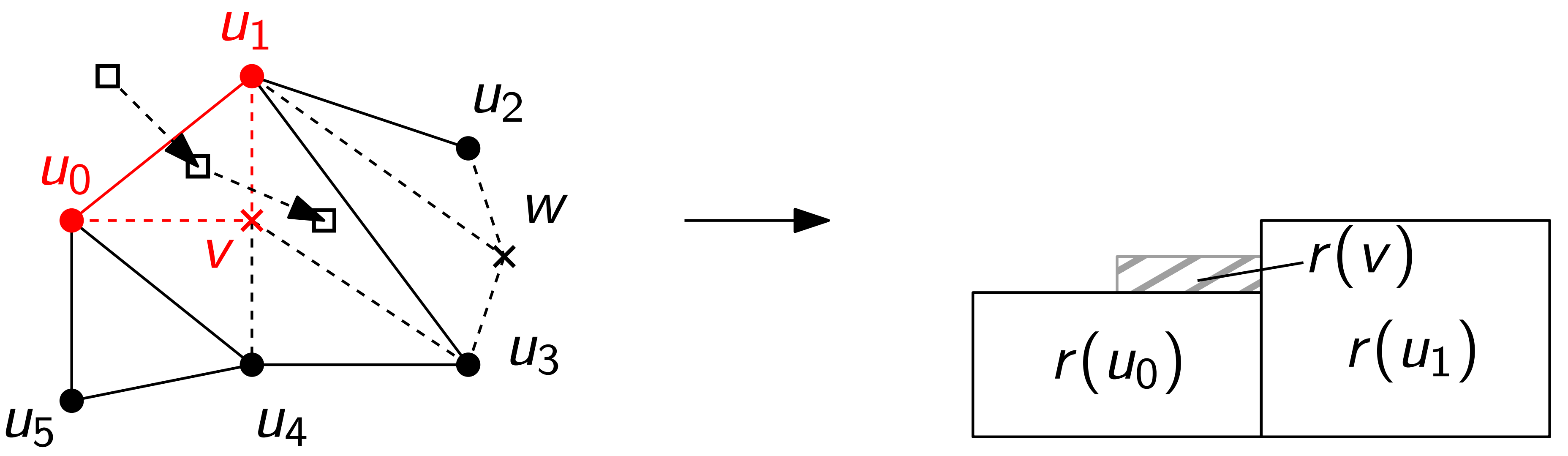
- Inner dummy-node $v \in V_i$
- Outer dummy-node $v \in V_a$
- Normal vertex $v \in V = V' \setminus \{V_i \cup V_a\}$

An inner dummy-node



Place the free space corresponding to the node v

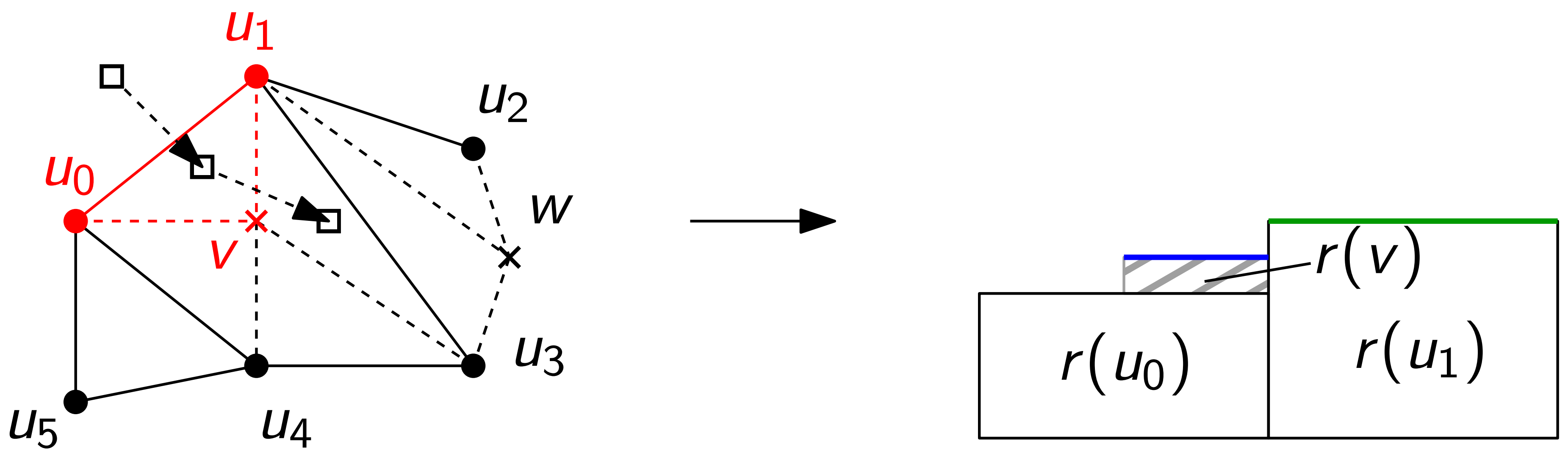
An inner dummy-node



Place the free space corresponding to the node v

- Place $r(v)$ on top of the rectangle with the lower top edge

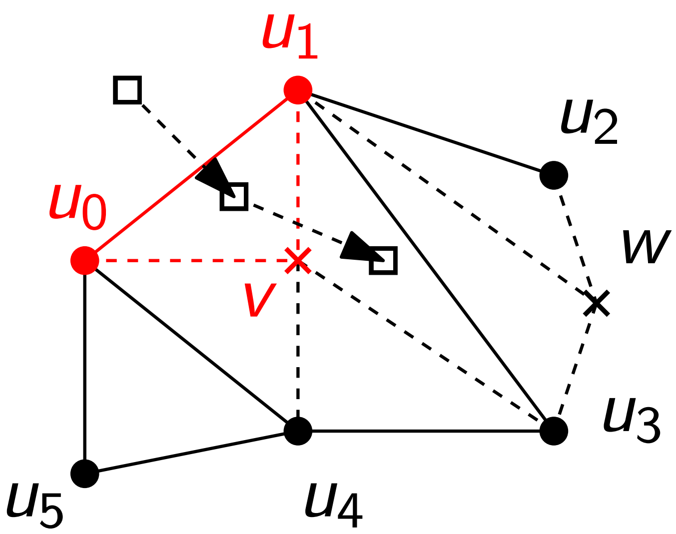
An inner dummy-node

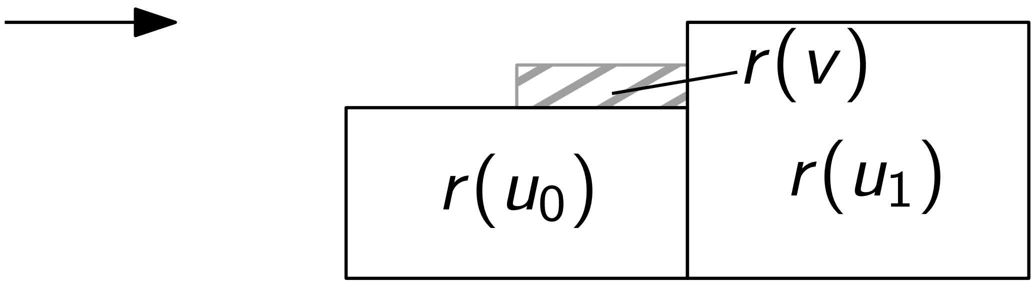


Place the free space corresponding to the node v

- Place $r(v)$ on top of the rectangle with the lower top edge
- The top edge of $r(v)$ must be below the top edge of $r(u_1)$ with $t_v = 1/2 \cdot (t_{u_0} + t_{u_1})$

An inner dummy-node



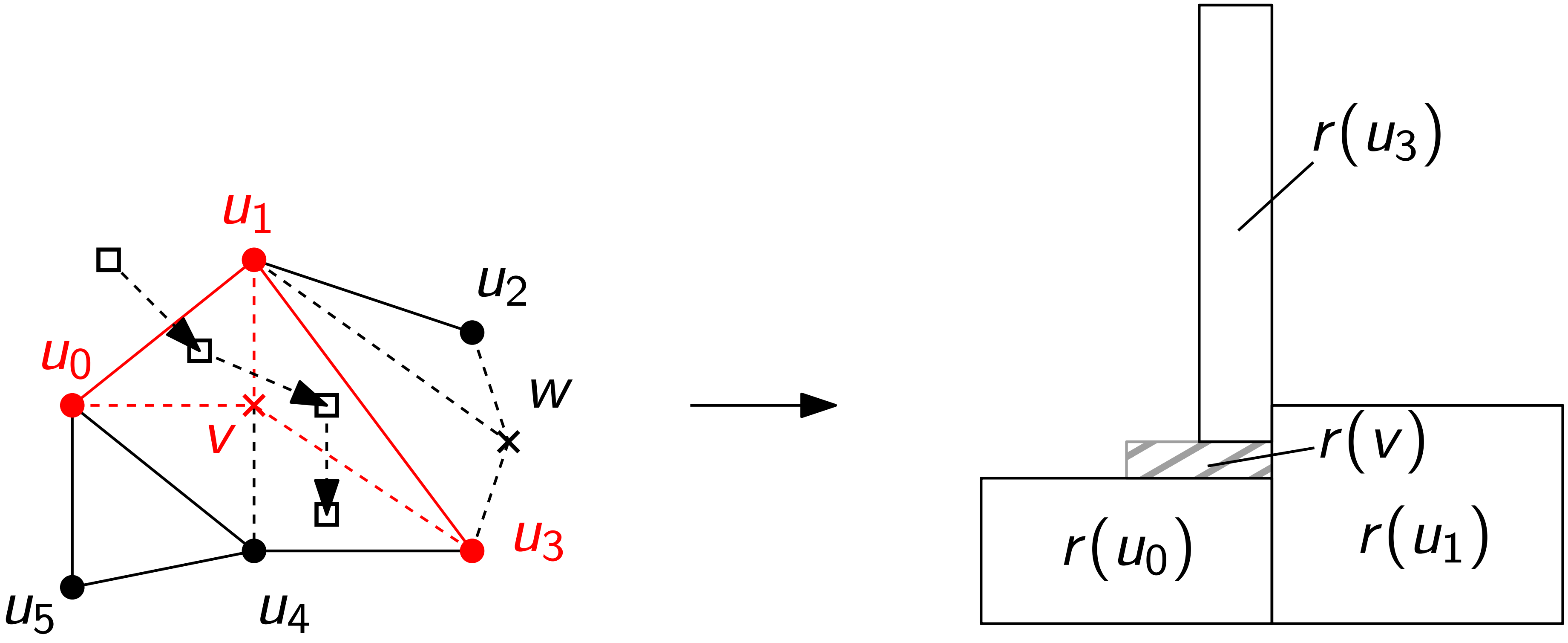


Place the free space corresponding to the node v

- Place $r(v)$ on top of the rectangle with the lower top edge
- The top edge of $r(v)$ must be below the top edge of $r(u_1)$ with $t_v = 1/2 \cdot (t_{u_0} + t_{u_1})$

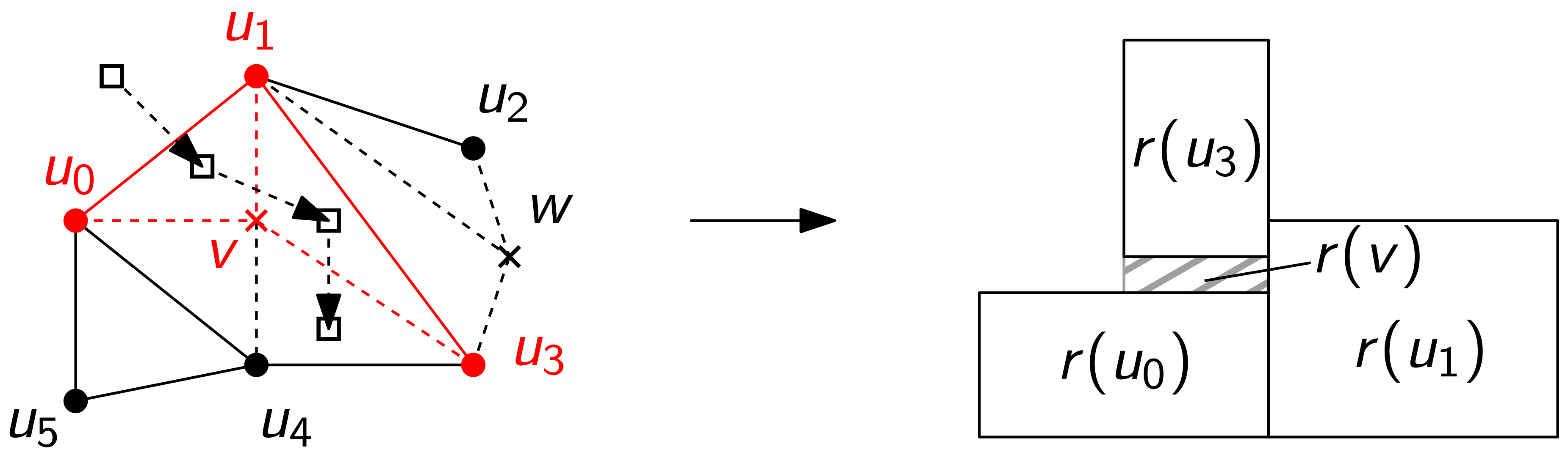
Walk clockwise around the inner dummy-node

An inner dummy-node



Place all following vertices on top of $r(v)$ except the last one

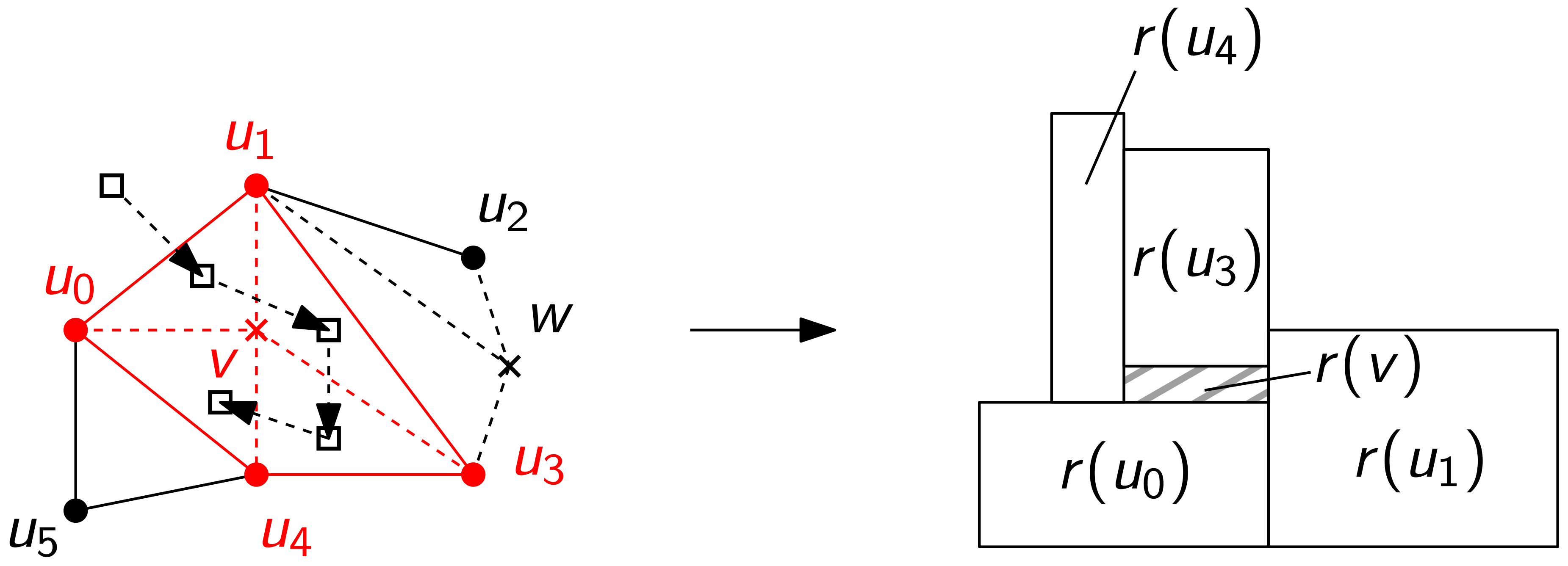
An inner dummy-node



Place all following vertices on top of $r(v)$ except the last one

The last but one rectangle, in this case $r(u_3)$, must fill the remaining free space on top of $r(v)$

An inner dummy-node

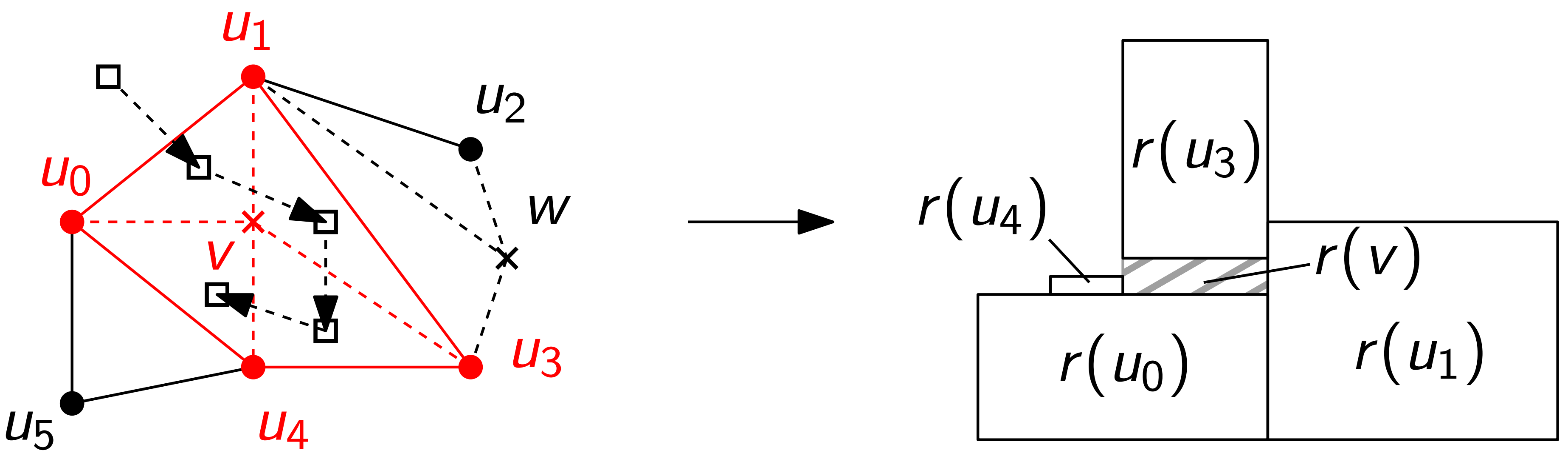


Place all following vertices on top of $r(v)$ except the last one

The last but one rectangle, in this case $r(u_3)$, must fill the remaining free space on top of $r(v)$

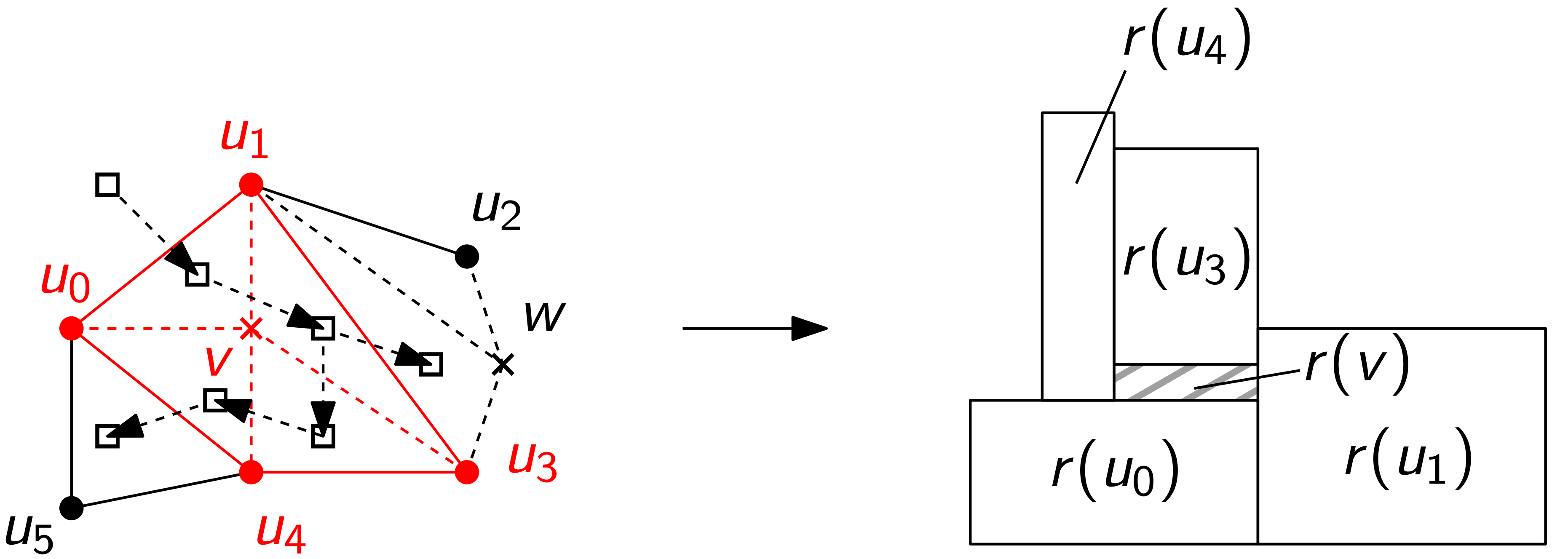
The last rectangle must connect its predecessor with the first rectangle

An inner dummy-node



If $r(u_4)$ is too small to connect both rectangles reduce the width to increase its height

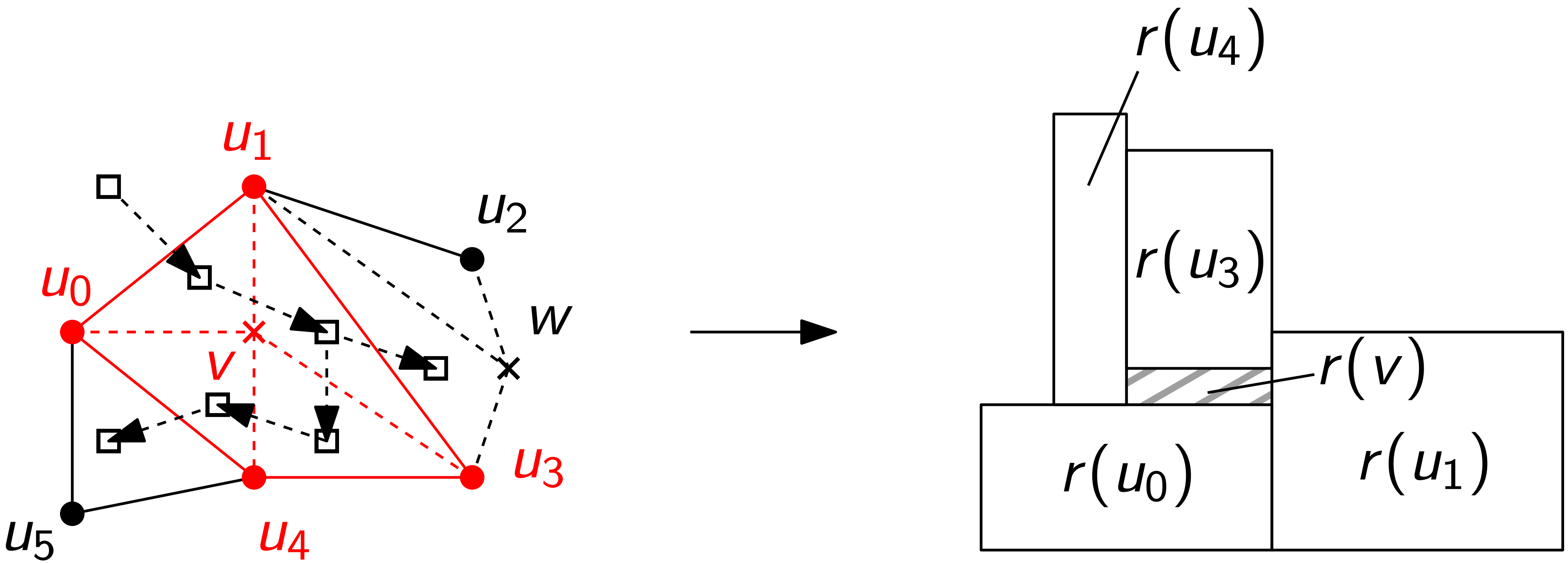
An inner dummy-node



If $r(u_4)$ is too small to connect both rectangles reduce the width to increase its height

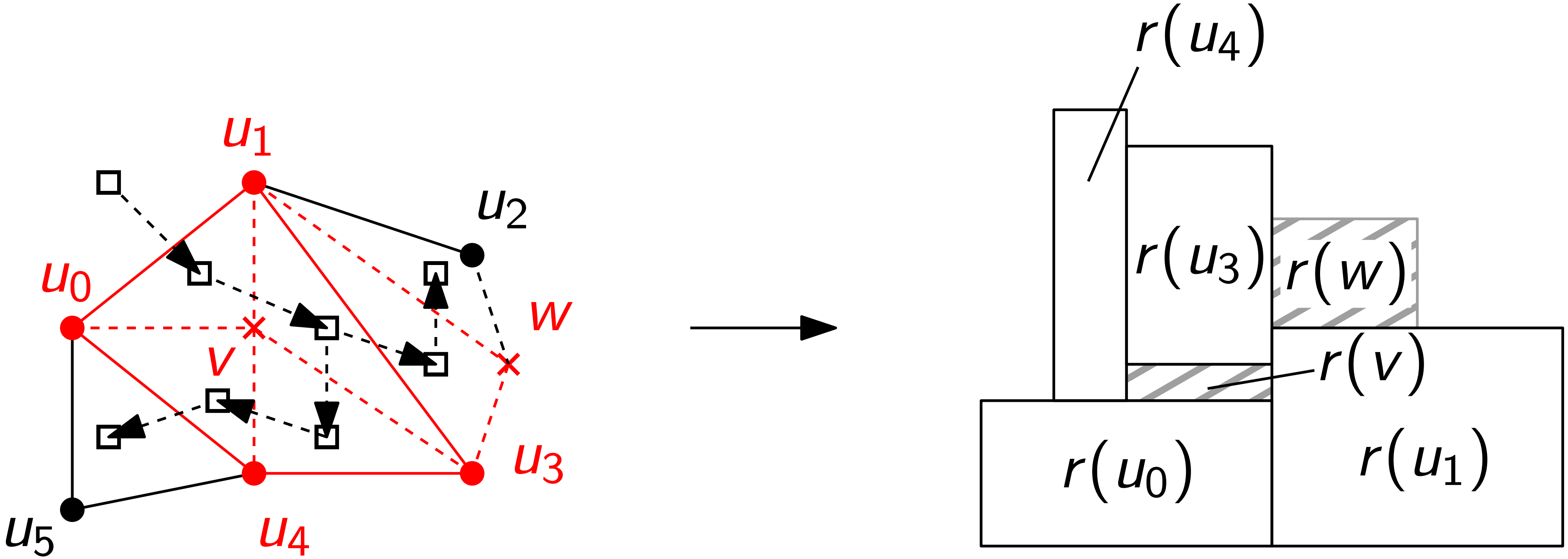
All rectangles adjacent to $r(v)$ have been placed correctly
$\rightarrow$ Finally visit all adjacent faces in G'

An outer dummy-node



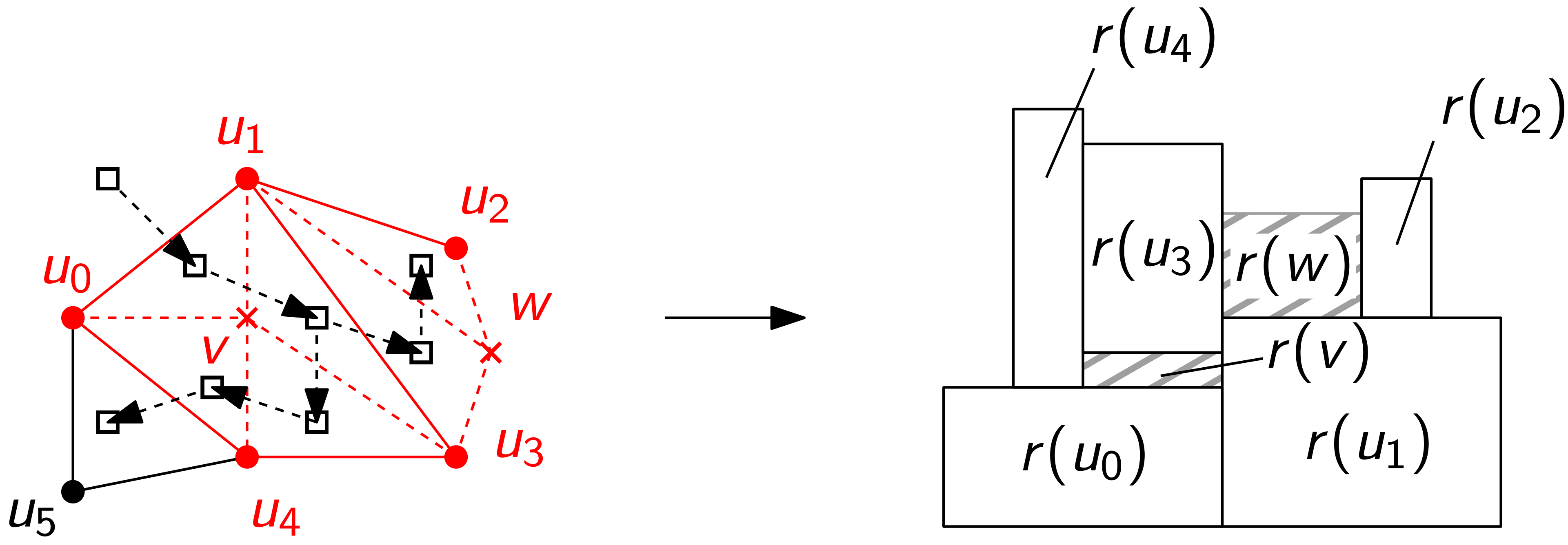
Insert the outer dummy-node into the rectangular representation with size $a(w) = 1$

An outer dummy-node



Insert the outer dummy-node into the rectangular representation with size $a(w) = 1$

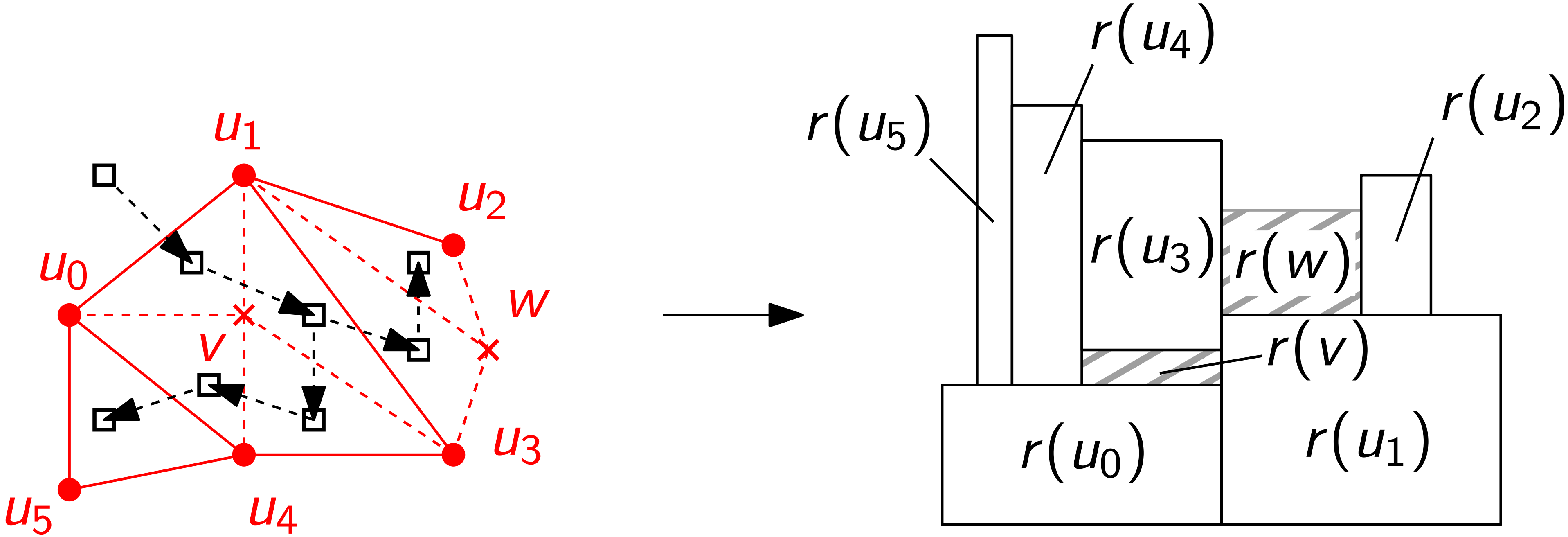
An outer dummy-node



Insert the outer dummy-node into the rectangular representation with size $a(w) = 1$

Add the rectangle $r(u_2)$ adjacent to the dummy-node and the separating vertex

An outer dummy-node



Insert the outer dummy-node into the rectangular representation with size $a(w) = 1$

Add the rectangle $r(u_2)$ adjacent to the dummy-node and the separating vertex

Proof of correctness for outerplanar graphs

The correctness is proved similarly to the maximal outerplanar graphs by induction on the number of placed rectangles

Proof of correctness for outerplanar graphs

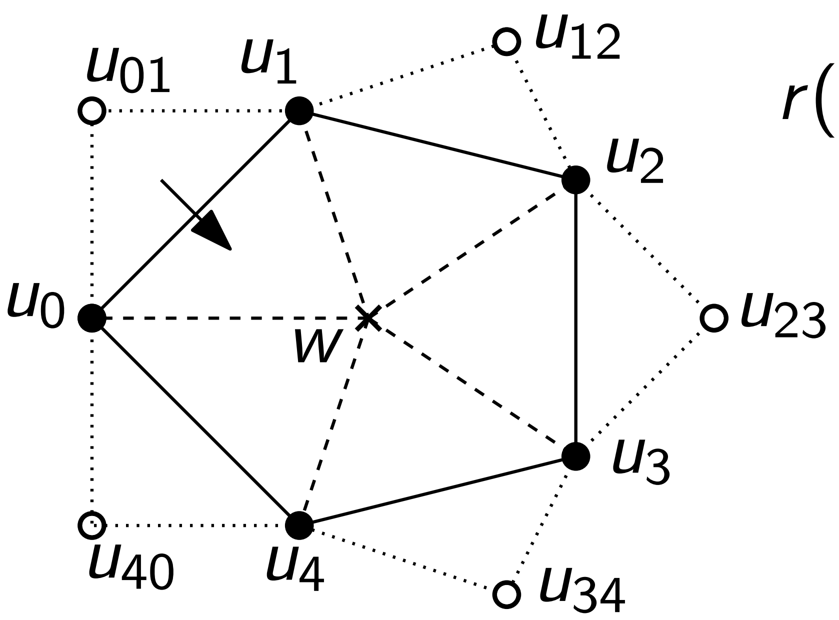
The correctness is proved similarly to the maximal outerplanar graphs by induction on the number of placed rectangles

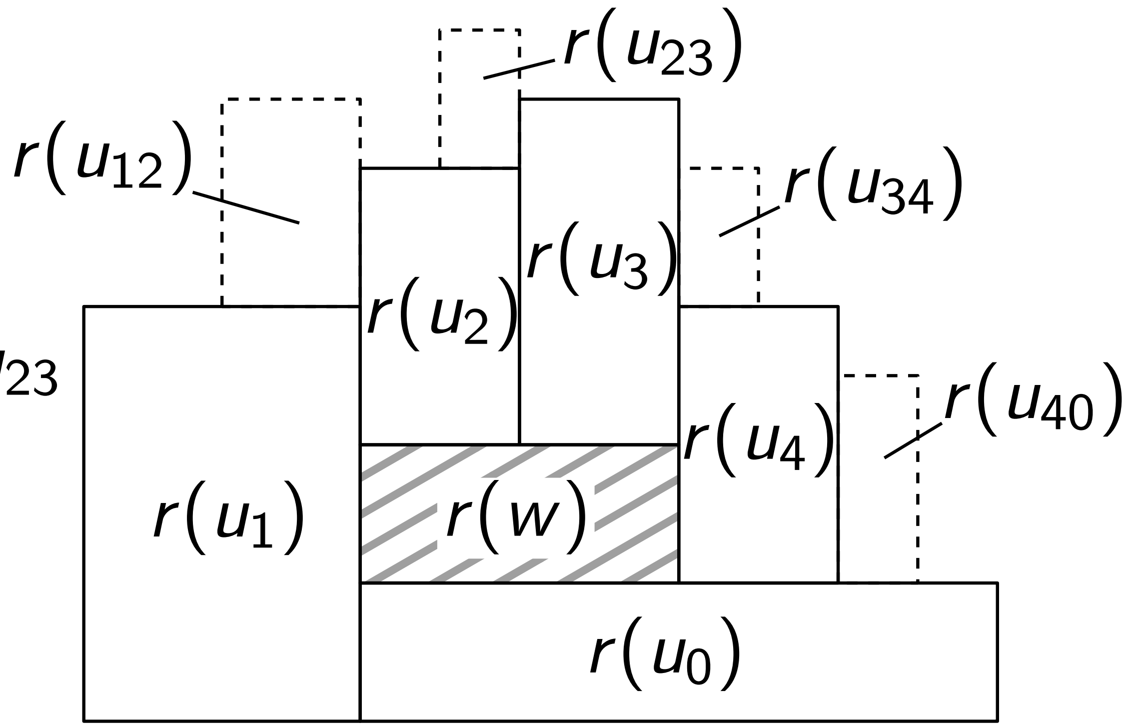
Distinguish between normal node, inner and outer dummy-node:
$\rightarrow$ Process all faces adjacent to a dummy-node in one step!

Proof of correctness for outerplanar graphs

The correctness is proved similarly to the maximal outerplanar graphs by induction on the number of placed rectangles

Distinguish between normal node, inner and outer dummy-node:
→ Process all faces adjacent to a dummy-node in one step!

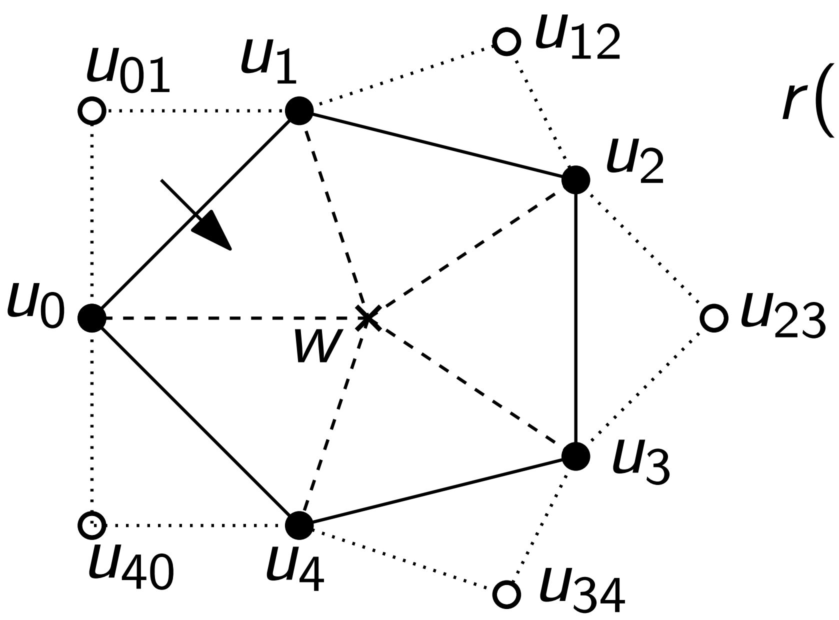


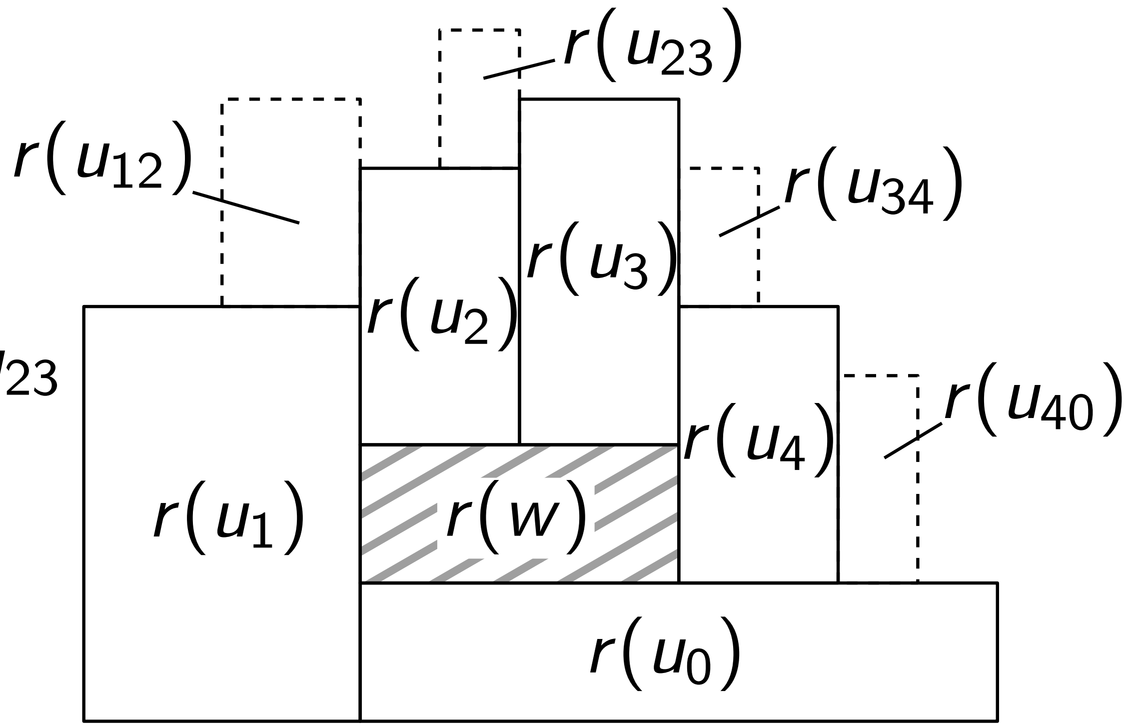


Proof of correctness for outerplanar graphs

Invariant:

There is a *free corner* *for the next rectangle* $r(w)$ *to be placed with* $w \in V \setminus V_k$*, in such a way that it is* *adjacent to its predecessors* $r(u)$ *and* $r(v)$ *with* $u, v \in V_k$*.*



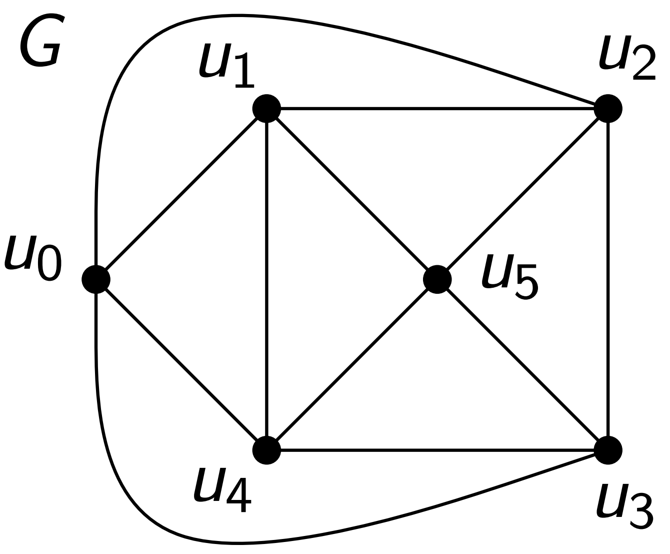


K-outerplanar graphs

A graph is k-outerplanar if removing all vertices on the outer face in its embedding results in a $(k-1)$-outerplanar graph

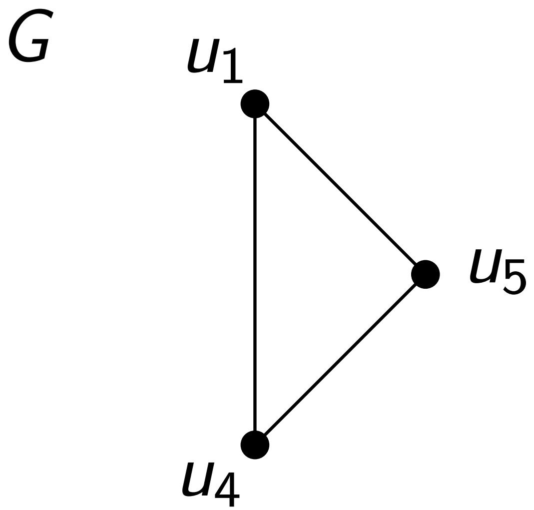
K-outerplanar graphs

A graph is k-outerplanar if removing all vertices on the outer face in its embedding results in a $(k-1)$-outerplanar graph



K-outerplanar graphs

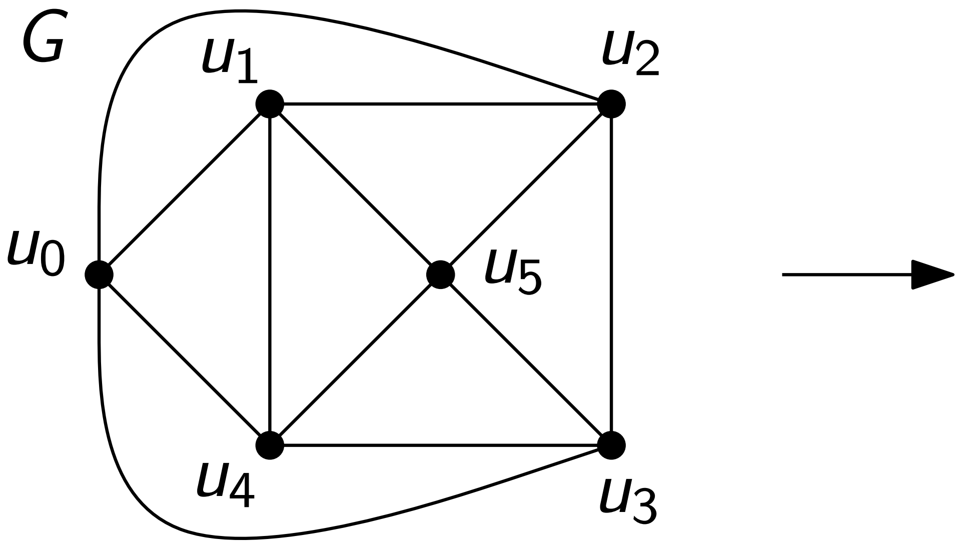
A graph is k-outerplanar if removing all vertices on the outer face in its embedding results in a $(k-1)$-outerplanar graph



G is a 2-outerplanar graph: removing all vertices on the unbounded face results in a 1-outerplanar graph

K-outerplanar graphs

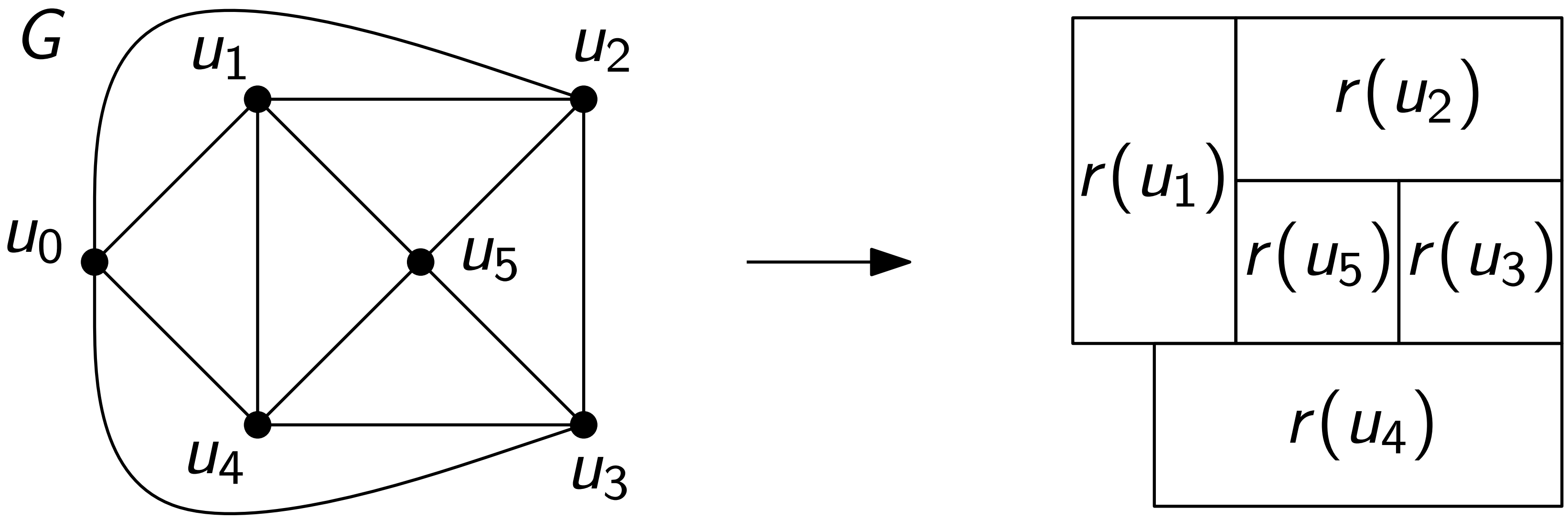
A graph is k-outerplanar if removing all vertices on the outer face in its embedding results in a $(k-1)$-outerplanar graph



G is a 2-outerplanar graph: removing all vertices on the unbounded face results in a 1-outerplanar graph

K-outerplanar graphs

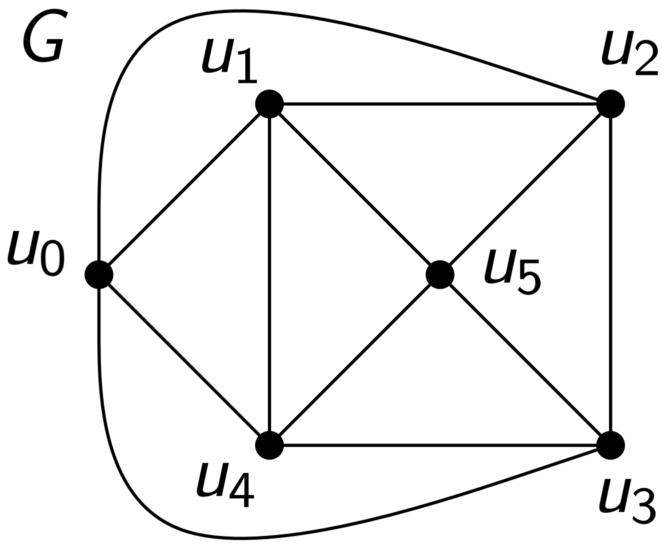
A graph is k-outerplanar if removing all vertices on the outer face in its embedding results in a $(k-1)$-outerplanar graph

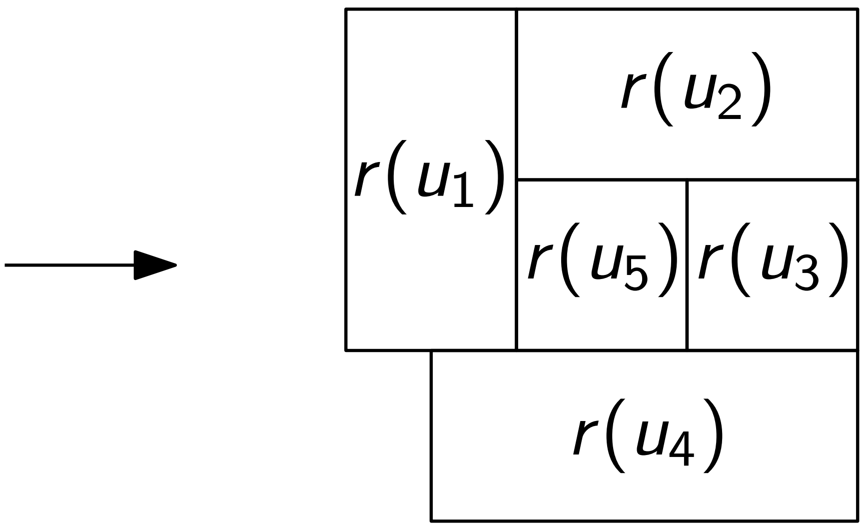


G is a 2-outerplanar graph: removing all vertices on the unbounded face results in a 1-outerplanar graph

K-outerplanar graphs

A graph is k-outerplanar if removing all vertices on the outer face in its embedding results in a $(k-1)$-outerplanar graph



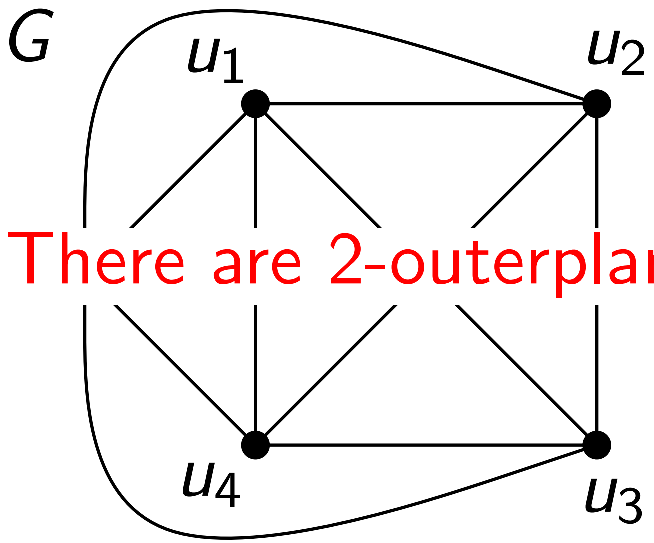


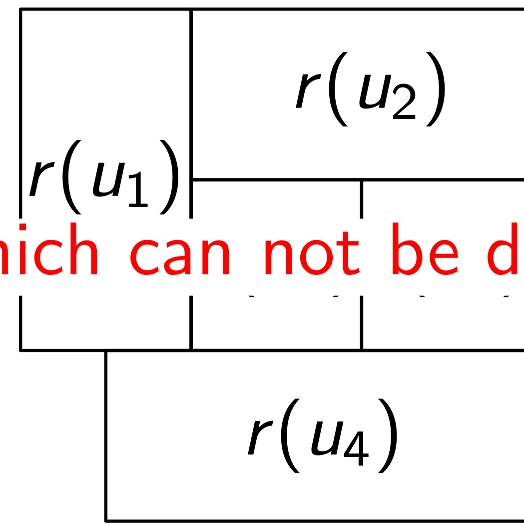
G is a 2-outerplanar graph: removing all vertices on the unbounded face results in a 1-outerplanar graph

It is not possible to draw $r(u_0)$ adjacent to all rectangles surrounding $r(u_5)$!

K-outerplanar graphs

A graph is k-outerplanar if removing all vertices on the outer face in its embedding results in a $(k-1)$-outerplanar graph





There are 2-outerplanar graphs, which can not be drawn!

G is a 2-outerplanar graph: removing all vertices on the unbounded face results in a 1-outerplanar graph

It is not possible to draw $r(u_0)$ adjacent to all rectangles surrounding $r(u_5)$!

Recapitulation

- Draw the rectangles with the desired size

Recapitulation

- Draw the rectangles with the desired size ✓
- No need for a surrounding rectangle

Recapitulation

- Draw the rectangles with the desired size ✓
- No need for a surrounding rectangle ✓
- Extension to general graph classes

Recapitulation

- Draw the rectangles with the desired size ✓
- No need for a surrounding rectangle ✓
- Extension to general graph classes ✓
- Every rectangular representation is possible to draw

Recapitulation

- Draw the rectangles with the desired size ✓
- No need for a surrounding rectangle ✓
- Extension to general graph classes ✓
- Every rectangular representation is possible to draw ✓

Recapitulation

- Draw the rectangles with the desired size ✓
- No need for a surrounding rectangle ✓
- Extension to general graph classes ✓
- Every rectangular representation is possible to draw ✓

Future work:

- Improve the aspect ratio of the rectangles with a better distribution of the free space on top of the rectangles

Recapitulation

- Draw the rectangles with the desired size ✓
- No need for a surrounding rectangle ✓
- Extension to general graph classes ✓
- Every rectangular representation is possible to draw ✓

Future work:

- Improve the aspect ratio of the rectangles with a better distribution of the free space on top of the rectangles
- Consider other graph classes, e.g. series-parallel graphs

Recapitulation

- Draw the rectangles with the desired size ✓
- No need for a surrounding rectangle ✓
- Extension to general graph classes ✓
- Every rectangular representation is possible to draw ✓

Future work:

- Improve the aspect ratio of the rectangles with a better distribution of the free space on top of the rectangles
- Consider other graph classes, e.g. series-parallel graphs

Thank you for your attention!
Questions?