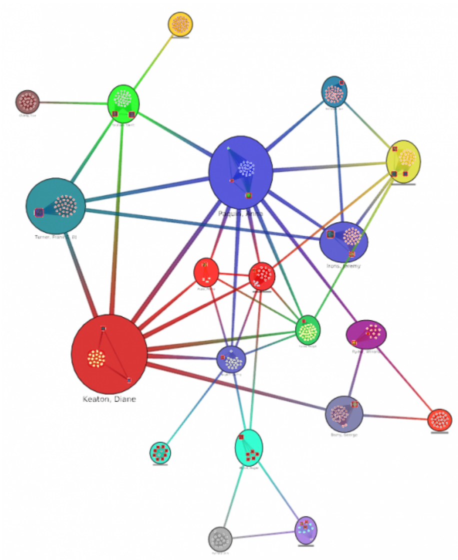
Rook-drawing for planar graphs

Claire Pennarun
From joint work with David Auber, Nicolas Bonichon and Paul Dorbec

LaBRI, Bordeaux

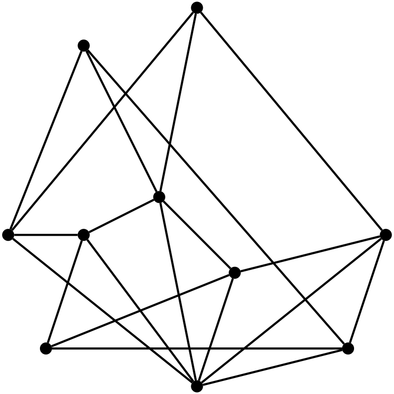
Labyrinth Day
April 3^{rd}, 2015

Visualization of graphs



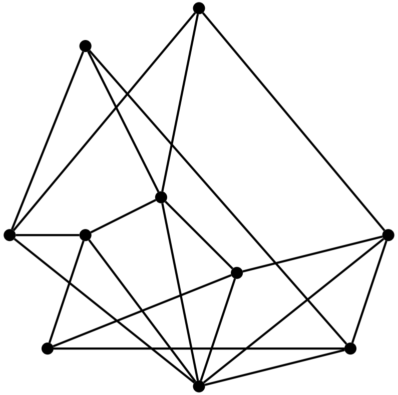
Drawing graphs

We want to draw large graphs representing dynamic data :



Drawing graphs

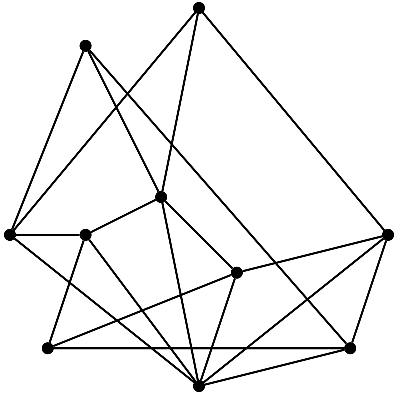
We want to draw large graphs representing **dynamic data** :



Drawing graphs

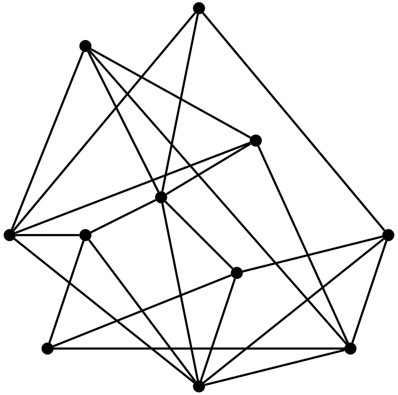
We want to draw large graphs representing **dynamic data** :

- addition/deletion of nodes



We want to draw large graphs representing **dynamic data** :

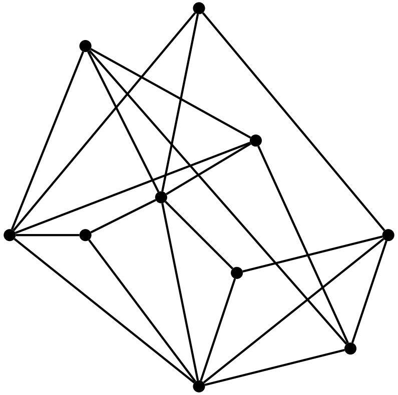
- addition/deletion of nodes



Drawing graphs

We want to draw large graphs representing **dynamic data** :

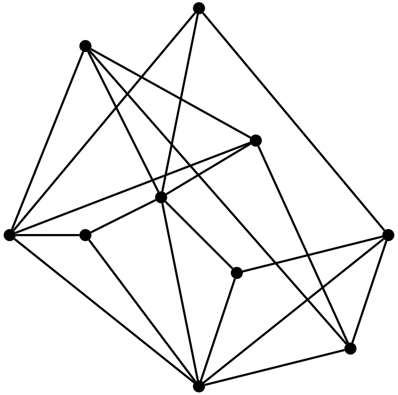
- addition/deletion of nodes



Drawing graphs

We want to draw large graphs representing **dynamic data** :

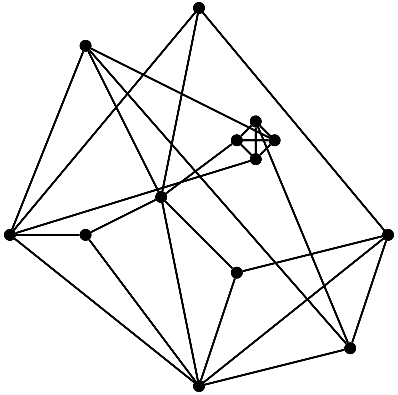
- addition/deletion of nodes
- a metanode = a group of nodes



Drawing graphs

We want to draw large graphs representing **dynamic data** :

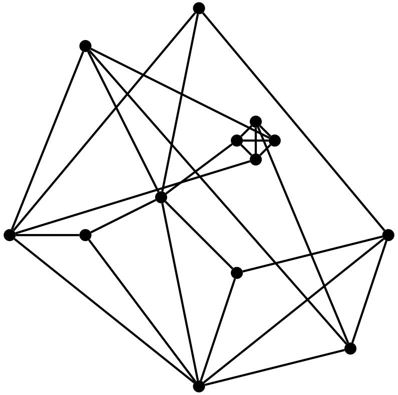
- addition/deletion of nodes
- a metanode = a group of nodes



Drawing graphs

We want to draw **large** graphs representing dynamic data :

- addition/deletion of nodes
- a metanode = a group of nodes

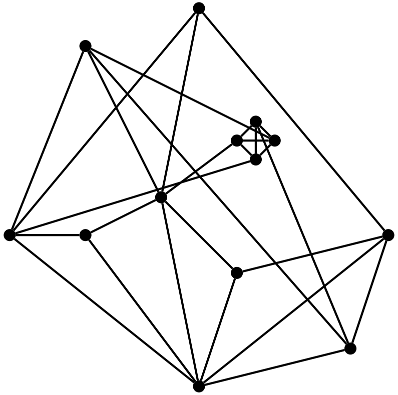


Drawing graphs

We want to draw **large** graphs representing dynamic data :

- addition/deletion of nodes
- a metanode = a group of nodes

- preserve the relative positions of nodes

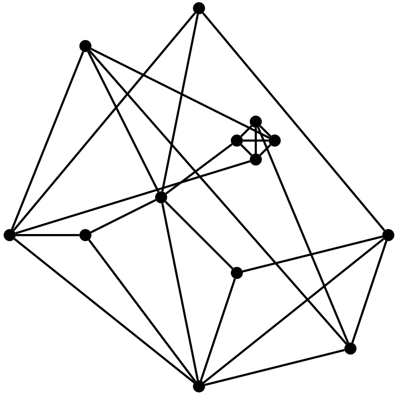


Drawing graphs

We want to draw **large** graphs representing dynamic data :

- addition/deletion of nodes
- a metanode = a group of nodes

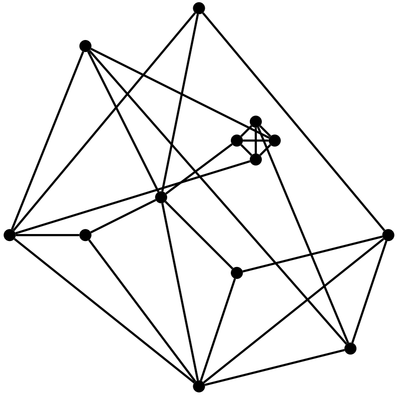
- preserve the relative positions of nodes
- low complexity of algorithms (linear, if possible)



Drawing graphs

We want to draw **large** graphs representing dynamic data :

- addition/deletion of nodes
- a metanode = a group of nodes



- preserve the relative positions of nodes
- low complexity of algorithms (linear, if possible)

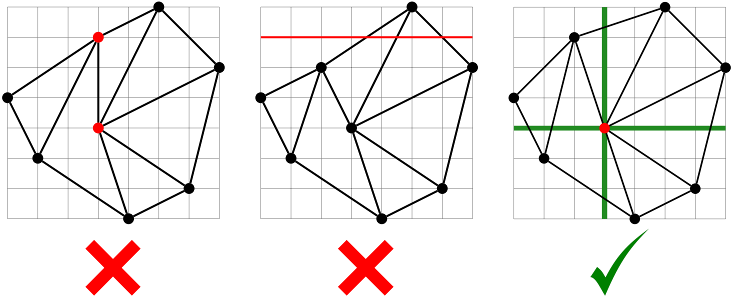
→ new type of drawing with constraints : rook-drawing

Rook-drawing

Definition

A rook-drawing of a graph of n vertices :

- Straight-lines
- Regular grid $n \times n$
- One vertex per line and column exactly

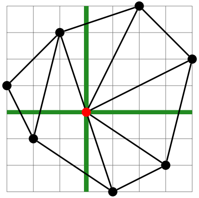


Rook-drawing

Definition

A rook-drawing of a graph of n vertices :

- Straight-lines
- Regular grid $n \times n$
- One vertex per line and column exactly

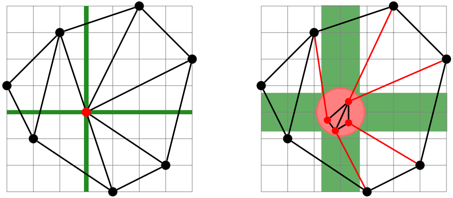


Rook-drawing

Definition

A rook-drawing of a graph of n vertices :

- Straight-lines
- Regular grid $n \times n$
- One vertex per line and column exactly

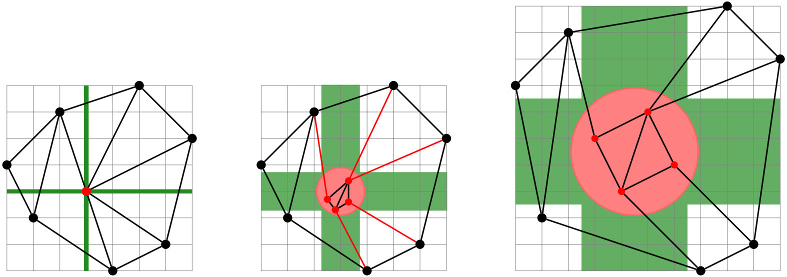


Rook-drawing

Definition

A rook-drawing of a graph of n vertices :

- Straight-lines
- Regular grid $n \times n$
- One vertex per line and column exactly

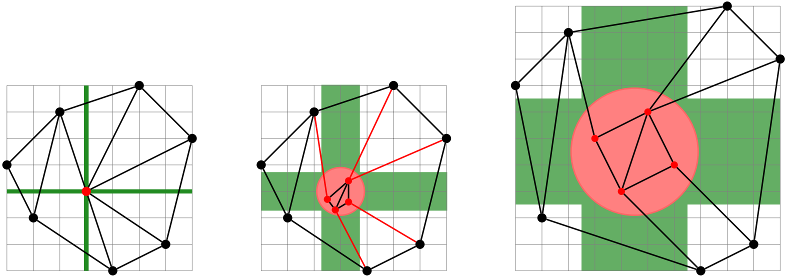


Rook-drawing

Definition

A rook-drawing of a graph of n vertices :$\rightarrow$ two orders on the nodes

- Straight-lines
- Regular grid $n \times n$
- One vertex per line and column exactly



Planar rook-drawing

Is there a **planar** rook-drawing for every planar graph ?

Planar rook-drawing

Is there a **planar** rook-drawing for every planar graph ?

What we already know :

- Straight-lines drawing ([Fáry, 1948] : every planar graph)
- Grid drawing ([de Fraysseix, 1988], [Schnyder, 1990] : every planar graph on an $(n-2) \times (n-2)$ grid)

Planar rook-drawing

Is there a **planar** rook-drawing for every planar graph ?

What we already know :

- Straight-lines drawing ([Fáry, 1948] : every planar graph)
- Grid drawing ([de Fraysseix, 1988], [Schnyder, 1990] : every planar graph on an $(n-2) \times (n-2)$ grid)

Idea : addition of constraints, but 2 more lines and columns !

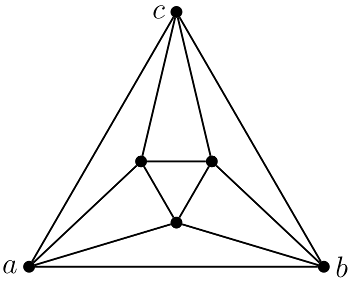
Planar rook-drawing

Is there a **planar** rook-drawing for every planar graph ?

What we already know :

- Straight-lines drawing ([Fáry, 1948] : every planar graph)
- Grid drawing ([de Fraysseix, 1988], [Schnyder, 1990] : every planar graph on an $(n-2) \times (n-2)$ grid)

Idea : addition of constraints, but 2 more lines and columns !



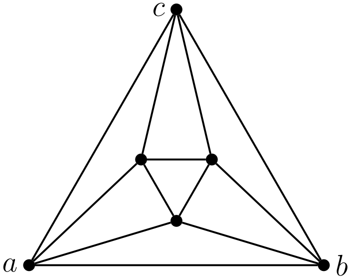
Planar rook-drawing

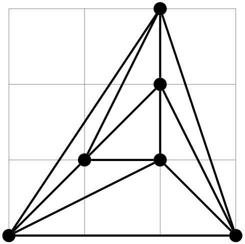
Is there a **planar** rook-drawing for every planar graph ?

What we already know :

- Straight-lines drawing ([Fáry, 1948] : every planar graph)
- Grid drawing ([de Fraysseix, 1988], [Schnyder, 1990] : every planar graph on an $(n-2) \times (n-2)$ grid)

Idea : addition of constraints, but 2 more lines and columns !





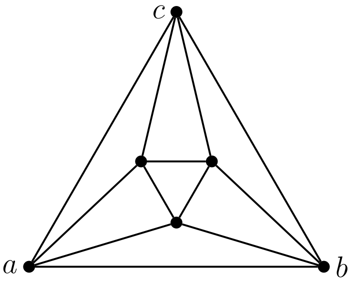
Planar rook-drawing

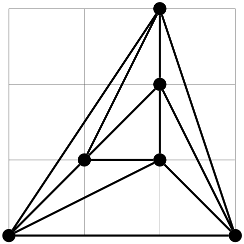
Is there a **planar** rook-drawing for every planar graph ? No !

What we already know :

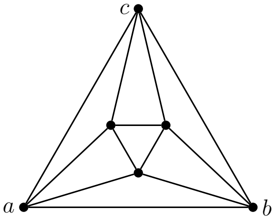
- Straight-lines drawing ([Fáry, 1948] : every planar graph)
- Grid drawing ([de Fraysseix, 1988], [Schnyder, 1990] : every planar graph on an $(n-2) \times (n-2)$ grid)

Idea : addition of constraints, but 2 more lines and columns !



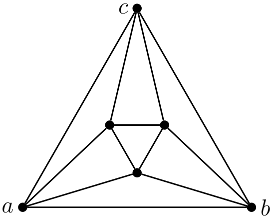


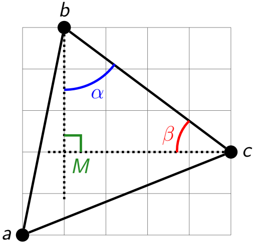
A counter-example



Three exterior nodes *a*, *b* and *c*. Inner nodes : inside the area delimited by (*ab*), (*bc*) and (*ca*).

A counter-example

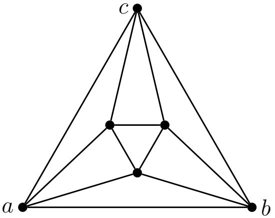




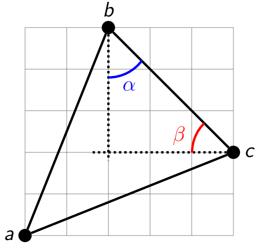
Three exterior nodes *a*, *b* and *c*. Inner nodes : inside the area delimited by (*ab*), (*bc*) and (*ca*).

$\alpha \geq 45°$, $\beta \geq 45°$. *Mbc* right-angled

A counter-example

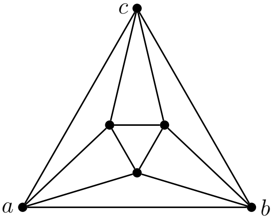


Three exterior nodes *a*, *b* and *c*. Inner nodes : inside the area delimited by (*ab*), (*bc*) and (*ca*).

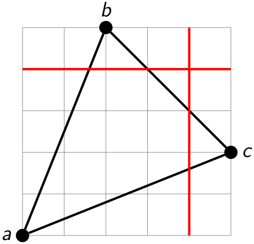


$\alpha \geq 45°$, $\beta \geq 45°$. *Mbc* right-angled
$\alpha = \beta = 45° \rightarrow x(b) = y(c)$.

A counter-example

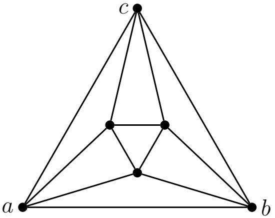


Three exterior nodes *a*, *b* and *c*. Inner nodes : inside the area delimited by (*ab*), (*bc*) and (*ca*).

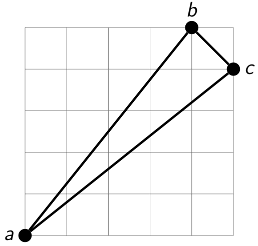


$\alpha \geq 45°$, $\beta \geq 45°$. *Mbc* right-angled
$\alpha = \beta = 45° \rightarrow x(b) = y(c)$.
(*bc*) prevents any node to fill the line under *b* or the column at the left of *c*.

A counter-example

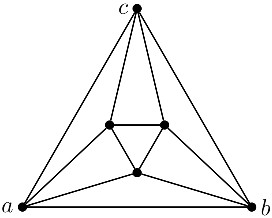


Three exterior nodes a, b and c. Inner nodes : inside the area delimited by (ab), (bc) and (ca).

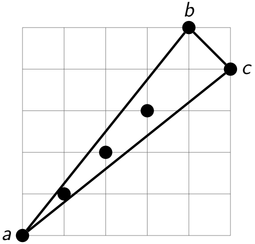


$\alpha \geq 45°$, $\beta \geq 45°$. Mbc right-angled
$\alpha = \beta = 45° \rightarrow x(b) = y(c)$.
(bc) prevents any node to fill the line under b or the column at the left of c.
Fill these line and column with c and b !

A counter-example



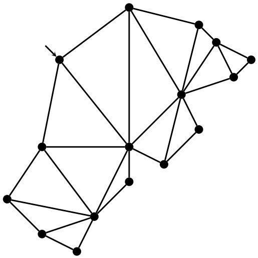
Three exterior nodes *a*, *b* and *c*. Inner nodes : inside the area delimited by (*ab*), (*bc*) and (*ca*).



$\alpha \geq 45°$, $\beta \geq 45°$. *Mbc* right-angled
$\alpha = \beta = 45° \rightarrow x(b) = y(c)$.
(*bc*) prevents any node to fill the line under *b* or the column at the left of *c*.
Fill these line and column with *c* and *b* !
Inner nodes : along a diagonal → problem with the edges...

Rook-drawing for outerplanar graphs

A graph is outerplanar if it has a planar drawing such that all its vertices are on the outer face.



Result

Every outerplanar graph has a rook-drawing which can be computed in linear time.

Rook-drawing for outerplanar graphs

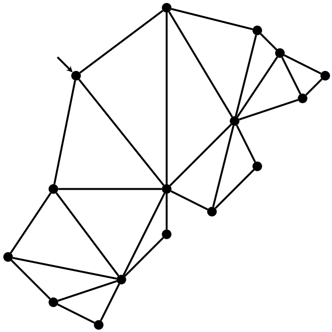
[Bonichon, Gavoille, Hanusse, 2005]

- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.

Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

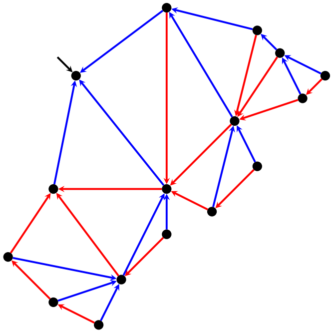
- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.



Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

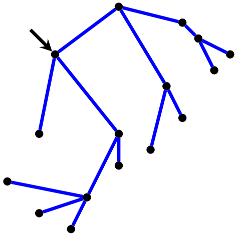
- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.



Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

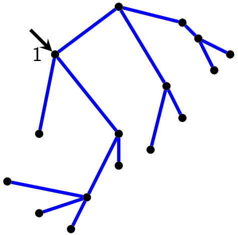
- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.



Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.

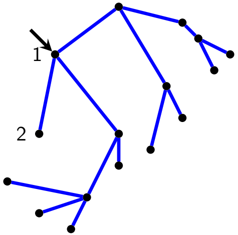


- x : ccw pre-order depth-first
- y : ccw post-order depth-first

Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

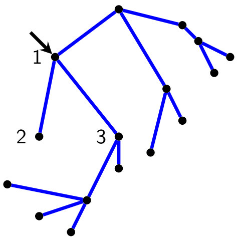
- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.



- x : ccw pre-order depth-first
- y : ccw post-order depth-first

[Bonichon, Gavoille, Hanusse, 2005]

- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.

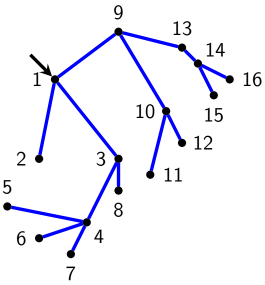


- x : ccw pre-order depth-first
- y : ccw post-order depth-first

Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

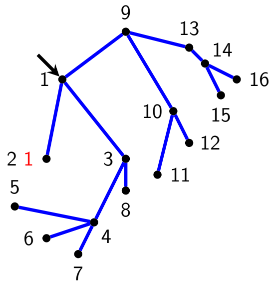
- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.



- x : ccw pre-order depth-first
- y : ccw post-order depth-first

[Bonichon, Gavoille, Hanusse, 2005]

- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.

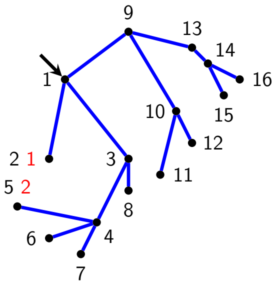


- x : ccw pre-order depth-first
- y : ccw post-order depth-first

Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.

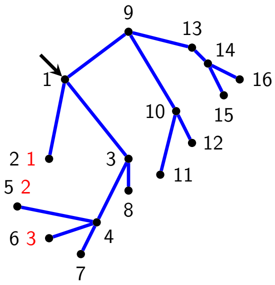


- x : ccw pre-order depth-first
- y : ccw post-order depth-first

Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.

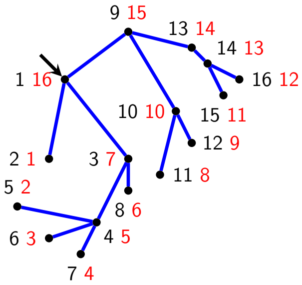


- x : ccw pre-order depth-first
- y : ccw post-order depth-first

Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

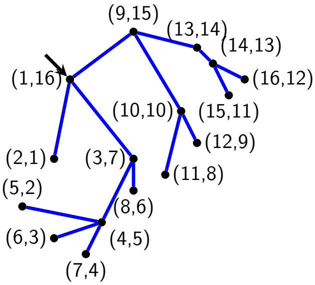
- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.

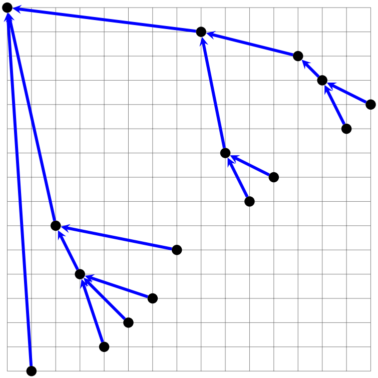


- x : ccw pre-order depth-first
- y : ccw post-order depth-first

[Bonichon, Gavoille, Hanusse, 2005]

- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.



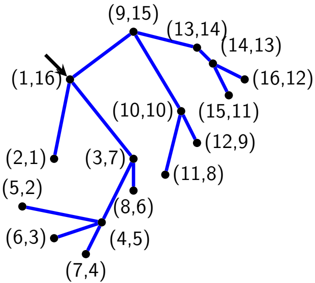


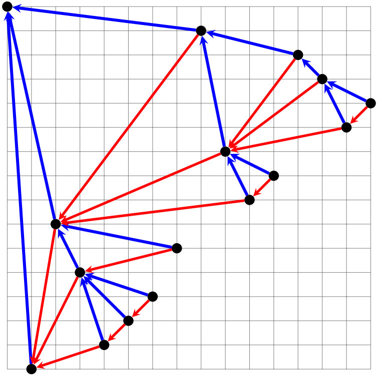
- x : ccw pre-order depth-first
- y : ccw post-order depth-first

Rook-drawing for outerplanar graphs

[Bonichon, Gavoille, Hanusse, 2005]

- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and its first non-descendant found after u in a clockwise pre-ordering of T_b.

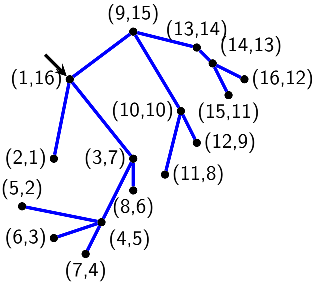




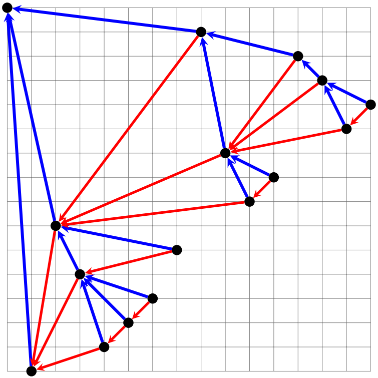
- x : ccw pre-order depth-first
- y : ccw post-order depth-first

[Bonichon, Gavoille, Hanusse, 2005]

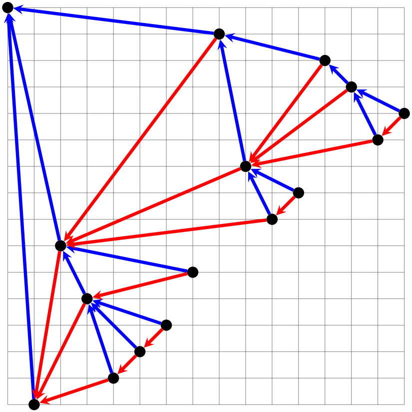
- edges of G outerplanar map $\rightarrow$ T_r, T_b
- edges of T_r : between u and the first node below u, unrelated in T_b.



- x : ccw pre-order depth-first
- y : ccw post-order depth-first



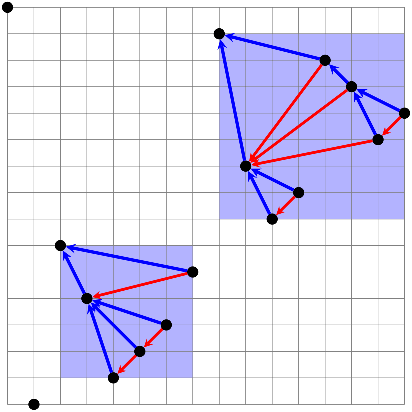
Rook-drawing for outerplanar graphs



Hypothesis : T_v of depth k (+ red edges) admits a planar rook-drawing in the grid $[x(v), x(v) + |T_v| - 1] \times [y(v) - |T_v| + 1, y(v)]$.

Proof by induction on depth :

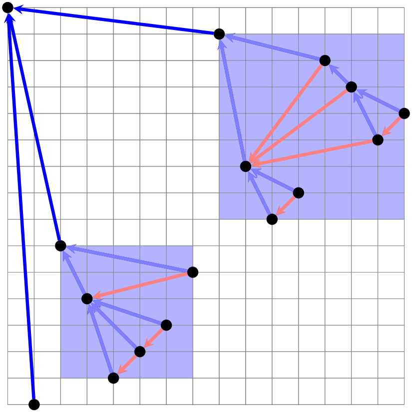
Rook-drawing for outerplanar graphs



Hypothesis : T_v of depth k (+ red edges) admits a planar rook-drawing in the grid $[x(v), x(v) + |T_v| - 1] \times [y(v) - |T_v| + 1, y(v)]$.

Proof by induction on depth :
The children subtrees are placed in distinct areas and are "well" drawn.

Rook-drawing for outerplanar graphs

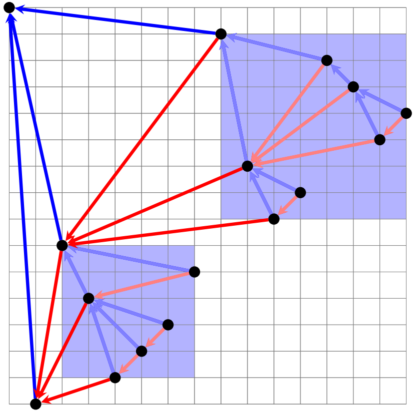


Hypothesis : T_v of depth k (+ red edges) admits a planar rook-drawing in the grid $[x(v), x(v) + |T_v| - 1] \times [y(v) - |T_v| + 1, y(v)]$.

Proof by induction on depth :
The children subtrees are placed in distinct areas and are "well" drawn.

Edges from v to its children : no crossings !

Rook-drawing for outerplanar graphs



Hypothesis : T_v of depth k (+ red edges) admits a planar rook-drawing in the grid $[x(v), x(v) + |T_v| - 1] \times [y(v) - |T_v| + 1, y(v)]$.

Proof by induction on depth :
The children subtrees are placed in distinct areas and are "well" drawn.

Edges from v to its children : no crossings !

Additional red edges are between children subtrees : no crossings !

Polyline rook-drawing for planar graphs

Main result

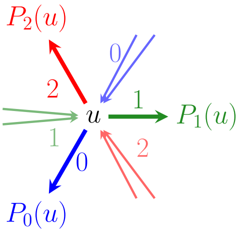
Every planar graph with n nodes admits a planar polyline rook-drawing, with at most $n-3$ bends (at most one per edge). Such a drawing can be computed in linear time.

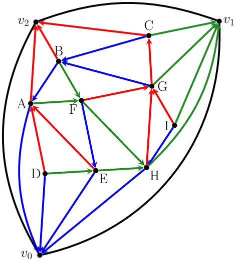
G a triangulation (else, make it triangulated and remove the edges at the end) with exterior nodes v_0, v_1 and v_2

Proof : based on an algorithm of [Bonichon, Mosbah, Le Saëc, 2002] optimizing the area of a polyline drawing.

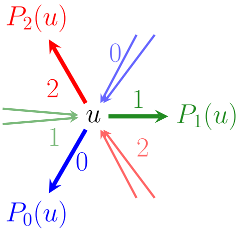
Schnyder woods

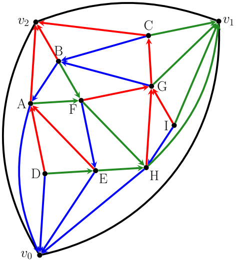
A Schnyder wood is a partition of the internal edges of a triangulation in three trees T_0, T_1 and T_2 (directed toward the root) and with a particular configuration around each inner node :





A Schnyder wood is a partition of the internal edges of a triangulation in three trees T_0, T_1 and T_2 (directed toward the root) and with a particular configuration around each inner node :

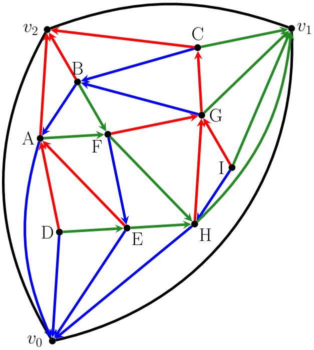




[Schnyder 1989]

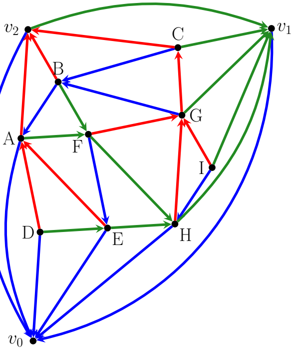
Every plane triangulation admits at least one Schnyder wood, and it can be computed in linear time.

- (T_0, T_1, T_2) : Schnyder wood of G.



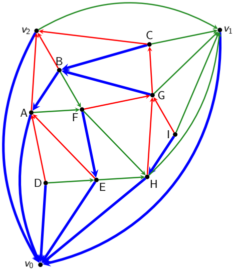
Planar polyline rook-drawing - Nodes

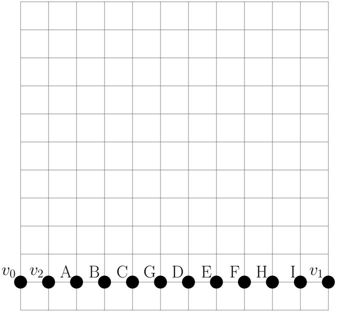
- (T_0, T_1, T_2) : Schnyder wood of G.
- $(v_1 v_0)$, $(v_2 v_0)$, $(v_2 v_1)$



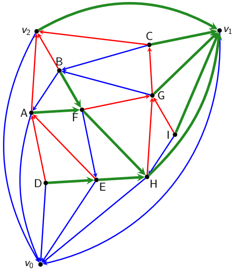
Planar polyline rook-drawing - Nodes

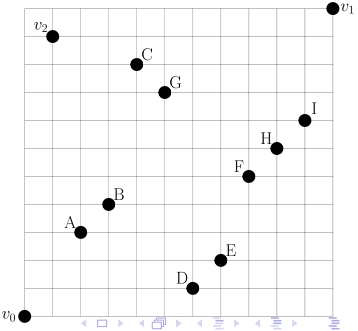
- (T_0, T_1, T_2) : Schnyder wood of G.
- $(v_1 v_0)$, $(v_2 v_0)$, $(v_2 v_1)$
- x : clockwise preordering of T_0 $= \{v_0 v_2 ABCGDEFHIv_1\}$.



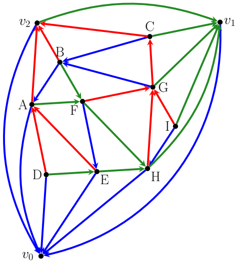


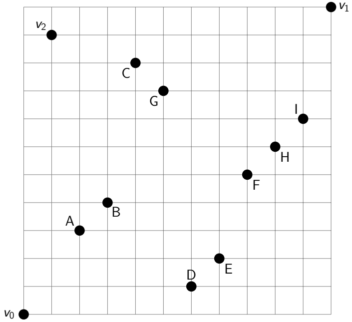
- (T_0, T_1, T_2) : Schnyder wood of G.
- $(v_1 v_0)$, $(v_2 v_0)$, $(v_2 v_1)$
- x : clockwise preordering of T_0 $= \{v_0 v_2 ABCGDEFHIv_1\}$.
- y : clockwise postordering of T_1 $= \{DEABFHIGCv_2 v_1\}$ $(v_0 = 0)$.





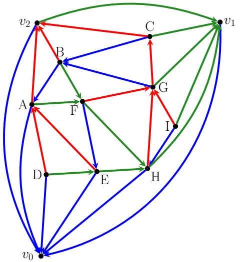
Planar polyline rook-drawing - Edges

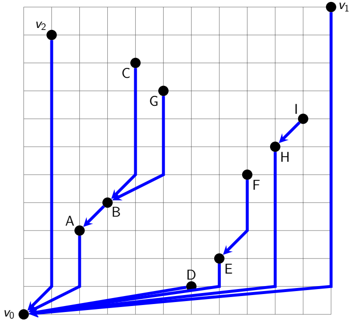




Planar polyline rook-drawing - Edges

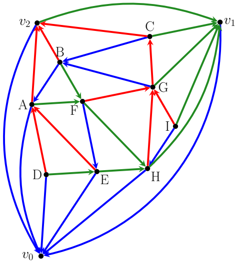
- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$

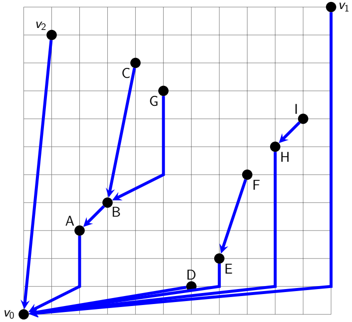




Planar polyline rook-drawing - Edges

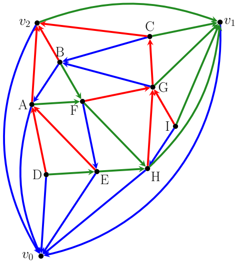
- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$
- First blue child : directly right to its father $\rightarrow$ straighten

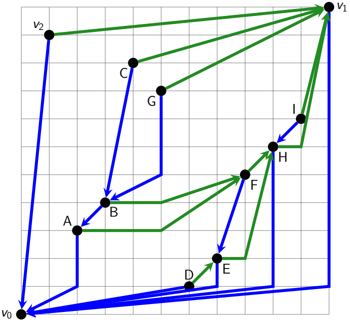




Planar polyline rook-drawing - Edges

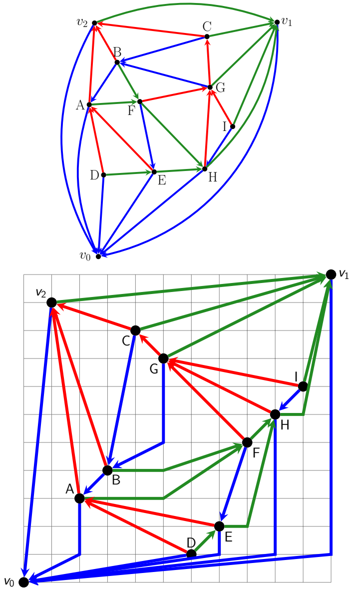
- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$
- First blue child : directly right to its father $\rightarrow$ straighten
- The edges $(u, P_1(u))$ are bent at $(x(\text{last descendant}_0(u)), y(u))$ (no bend if u is a leaf of T_0)





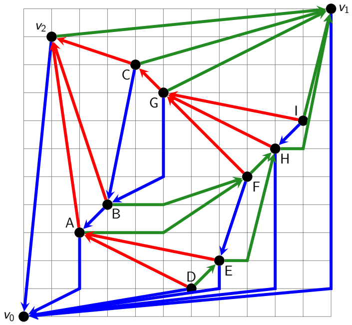
Planar polyline rook-drawing - Edges

- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$
- First blue child : directly right to its father $\rightarrow$ straighten
- The edges $(u, P_1(u))$ are bent at $(x(\text{last descendant}_0(u)), y(u))$ (no bend if u is a leaf of T_0)
- Edges of T_2 : not bent



Planar polyline rook-drawing - Edges

- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$
- First blue child : directly right to its father $\rightarrow$ straighten
- The edges $(u, P_1(u))$ are bent at $(x(\text{last descendant}_0(u)), y(u))$ (no bend if u is a leaf of T_0)
- Edges of T_2 : not bent

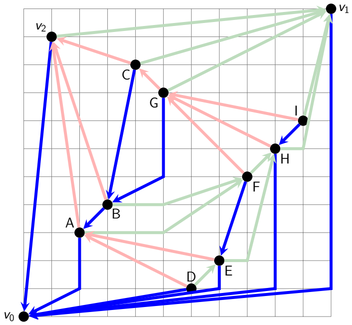


k = number of leaves in T_0

$T_0 = n - 1$ edges
$T_1 = n - 2$ edges
$T_2 = n - 3$ edges

Planar polyline rook-drawing - Edges

- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$
- First blue child : directly right to its father $\rightarrow$ straighten
- The edges $(u, P_1(u))$ are bent at $(x(\text{last descendant}_0(u)), y(u))$ (no bend if u is a leaf of T_0)
- Edges of T_2 : not bent



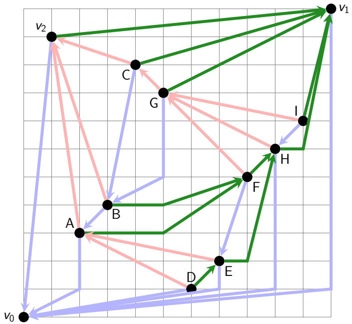
$T_0 = n - 1 - (n - k)$ bends.

k = number of leaves in T_0

$T_0 = n - 1$ edges
$T_1 = n - 2$ edges
$T_2 = n - 3$ edges

Planar polyline rook-drawing - Edges

- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$
- First blue child : directly right to its father $\rightarrow$ straighten
- The edges $(u, P_1(u))$ are bent at $(x(\text{last descendant}_0(u)), y(u))$ (no bend if u is a leaf of T_0)
- Edges of T_2 : not bent



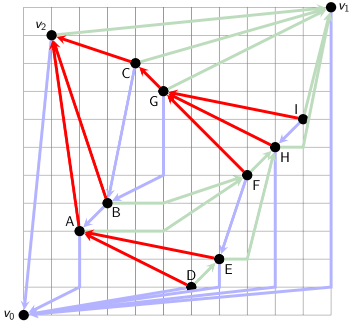
k = number of leaves in T_0

$$T_0 = n - 1 \text{ edges}$$
$$T_1 = n - 2 \text{ edges}$$
$$T_2 = n - 3 \text{ edges}$$

$$T_0 = n - 1 - (n - k) \text{ bends.}$$
$$T_1 = n - 2 - k \text{ bends.}$$

Planar polyline rook-drawing - Edges

- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$
- First blue child : directly right to its father $\rightarrow$ straighten
- The edges $(u, P_1(u))$ are bent at $(x(\text{last descendant}_0(u)), y(u))$ (no bend if u is a leaf of T_0)
- Edges of T_2 : not bent



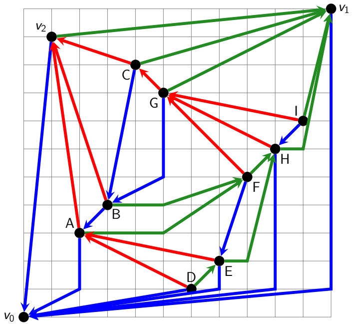
k = number of leaves in T_0

$$T_0 = n - 1 \text{ edges}$$
$$T_1 = n - 2 \text{ edges}$$
$$T_2 = n - 3 \text{ edges}$$

$$T_0 = n - 1 - (n - k) \text{ bends.}$$
$$T_1 = n - 2 - k \text{ bends.}$$
$$T_2 = 0 \text{ bends.}$$

Planar polyline rook-drawing - Edges

- The edges $(u, P_0(u))$ are bent at $(x(u), y(P_0(u)) + 1)$
- First blue child : directly right to its father $\rightarrow$ straighten
- The edges $(u, P_1(u))$ are bent at $(x(\text{last descendant}_0(u)), y(u))$ (no bend if u is a leaf of T_0)
- Edges of T_2 : not bent



k = number of leaves in T_0

$T_0 = n - 1$ edges

$T_1 = n - 2$ edges

$T_2 = n - 3$ edges

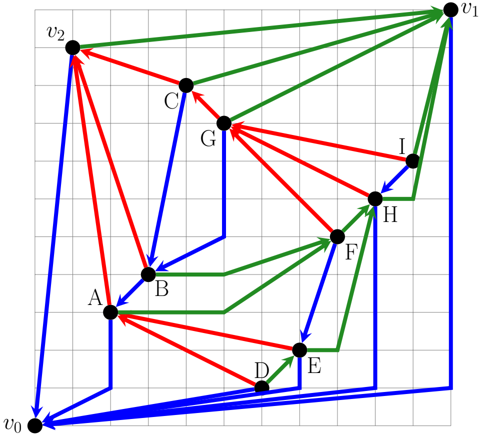
$T_0 = n - 1 - (n - k)$ bends.

$T_1 = n - 2 - k$ bends.

$T_2 = 0$ bends.

$\rightarrow n - 3$ bends in the drawing of G.

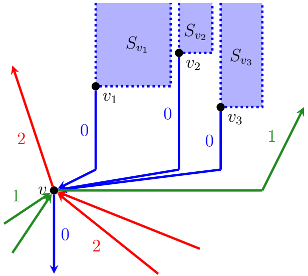
Proof of planarity (some ideas)



Edges direction

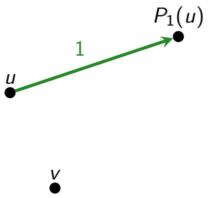
For each inner node v :

- $P_0(v)$ is left and below v.
- $P_1(v)$ is right and above v.
- $P_2(v)$ is left and above v.



Properties of green/red edges

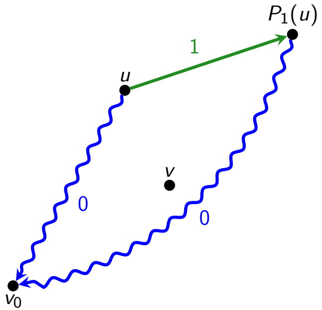
Every node v with $x(u) < x(v) < x(P_1(u))$ has $y(v) < y(u)$ if v is not a descendant of u in T_0.



v : between u and $P_1(u)$.

Properties of green/red edges

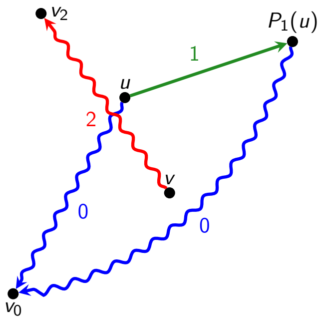
Every node v with $x(u) < x(v) < x(P_1(u))$ has $y(v) < y(u)$ if v is not a descendant of u in T_0.



v : between u and $P_1(u)$. $\rightarrow$ v in area (v_0, u), $(u, P_1(u))$, $(P_1(u), v_0)$.

Properties of green/red edges

Every node v with $x(u) < x(v) < x(P_1(u))$ has $y(v) < y(u)$ if v is not a descendant of u in T_0.

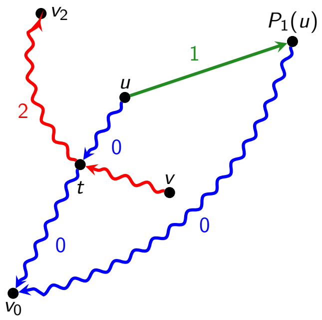


v : between u and $P_1(u)$. $\rightarrow$ v in area (v_0, u), $(u, P_1(u))$, $(P_1(u), v_0)$.

Path $v \rightarrow v_2$ has to "leave" the area !

Properties of green/red edges

Every node v with $x(u) < x(v) < x(P_1(u))$ has $y(v) < y(u)$ if v is not a descendant of u in T_0.



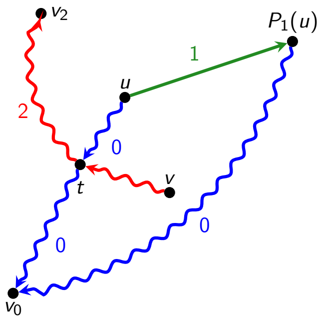
v : between u and $P_1(u)$. $\rightarrow$ v in area (v_0, u), $(u, P_1(u))$, $(P_1(u), v_0)$.

Path $v \rightarrow v_2$ has to "leave" the area ! Intersection t on path (v_0, u).

Paths $v \rightarrow t$ and $u \rightarrow t$.

Properties of green/red edges

Every node v with $x(u) < x(v) < x(P_1(u))$ has $y(v) < y(u)$ if v is not a descendant of u in T_0.



v : between u and $P_1(u)$. $\rightarrow$ v in area (v_0, u), $(u, P_1(u))$, $(P_1(u), v_0)$.

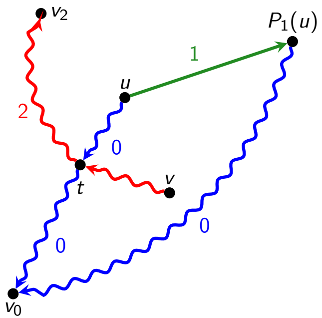
Path $v \rightarrow v_2$ has to "leave" the area ! Intersection t on path (v_0, u).

Paths $v \rightarrow t$ and $u \rightarrow t$.

$\rightarrow$ path from v to u is going upwards $= y(v) < y(u)$.

Properties of green/red edges

Every node v with $x(u) < x(v) < x(P_1(u))$ has $y(v) < y(u)$ if v is not a descendant of u in T_0.



v : between u and $P_1(u)$. $\rightarrow$ v in area (v_0, u), $(u, P_1(u))$, $(P_1(u), v_0)$.

Path $v \rightarrow v_2$ has to "leave" the area ! Intersection t on path (v_0, u).

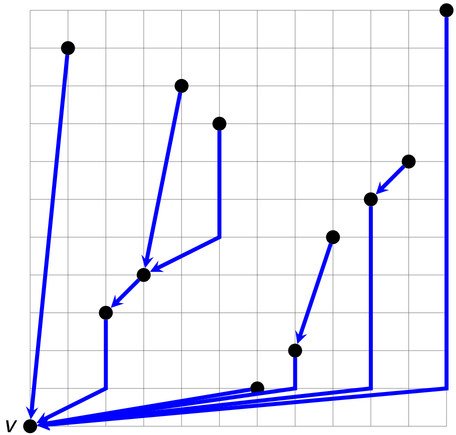
Paths $v \rightarrow t$ and $u \rightarrow t$.

$\rightarrow$ path from v to u is going upwards $= y(v) < y(u)$.

Every node v with $x(P_2(u)) < x(v) < x(u))$ has $y(v) < y(u)$ if v is not a descendant of $P_2(u)$ in T_0.

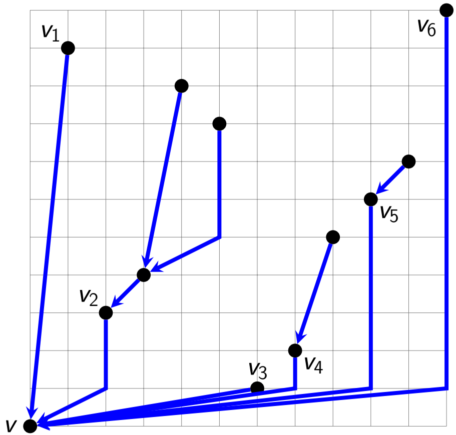
Non-crossing - blue

The edges of T_0 do not cross each other.



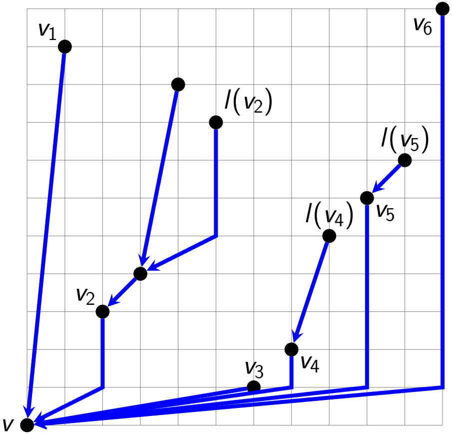
Non-crossing - blue

The edges of T_0 do not cross each other.



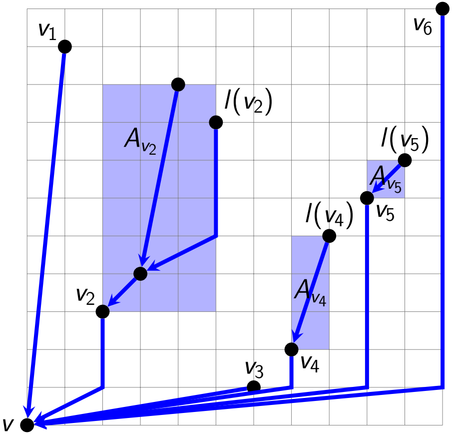
Non-crossing - blue

The edges of T_0 do not cross each other.



Non-crossing - blue

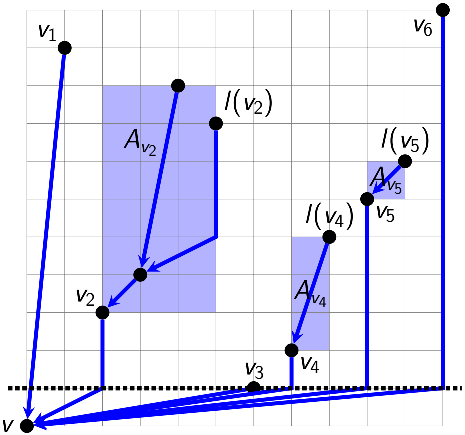
The edges of T_0 do not cross each other.



The subtrees of children "live" in different areas of width $(x(l(v_i)) - x(v_i))$.

Non-crossing - blue

The edges of T_0 do not cross each other.



The subtrees of children "live" in different areas of width $(x(l(v_i)) - x(v_i))$.

The edges to the children can not cross each other.

Non-crossing - green

The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

Non-crossing - green

The edges of T_1 do not cross each other.

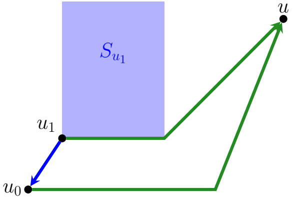
Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

u_{i+1} descendant of u_i in T_0

The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

u_{i+1} descendant of u_i in T_0

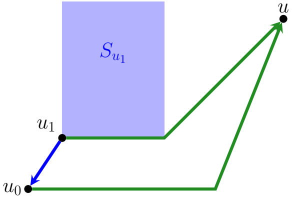


Non-crossing - green

The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

u_{i+1} descendant of u_i in T_0

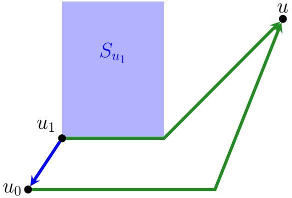


The last descendant of u_i in T_0 is on the right to the one of u_{i+1}.

The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

u_{i+1} descendant of u_i in T_0



The last descendant of u_i in T_0 is on the right to the one of u_{i+1}.

u_{i+1} not descendant of u_i in T_0

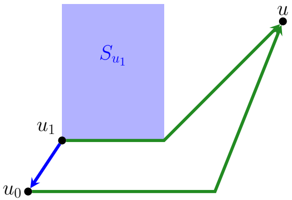
$x(u_{i+1}) > x(u_i)$

Non-crossing - green

The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

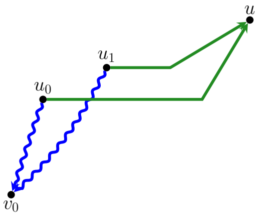
u_{i+1} descendant of u_i in T_0



The last descendant of u_i in T_0 is on the right to the one of u_{i+1}.

u_{i+1} not descendant of u_i in T_0

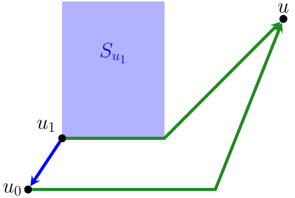
$x(u_{i+1}) > x(u_i)$



The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

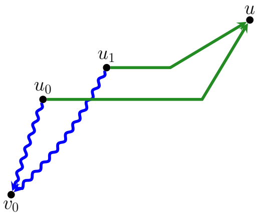
u_{i+1} descendant of u_i in T_0



The last descendant of u_i in T_0 is on the right to the one of u_{i+1}.

u_{i+1} not descendant of u_i in T_0

$x(u_{i+1}) > x(u_i)$



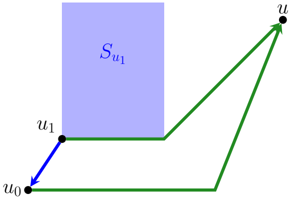
u_{i+1} should be below the edge (u_i, u)

Non-crossing - green

The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

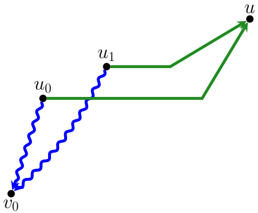
u_{i+1} descendant of u_i in T_0



The last descendant of u_i in T_0 is on the right to the one of u_{i+1}.

u_{i+1} not descendant of u_i in T_0

$x(u_{i+1}) > x(u_i)$



u_{i+1} should be below the edge (u_i, u)

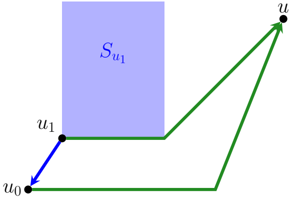
$x(u_{i+1}) < x(u_i)$

Non-crossing - green

The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

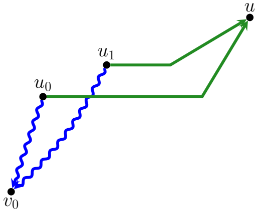
u_{i+1} descendant of u_i in T_0



The last descendant of u_i in T_0 is on the right to the one of u_{i+1}.

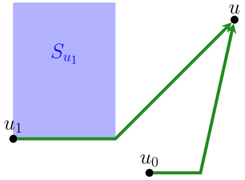
u_{i+1} not descendant of u_i in T_0

$x(u_{i+1}) > x(u_i)$



u_{i+1} should be below the edge (u_i, u)

$x(u_{i+1}) < x(u_i)$

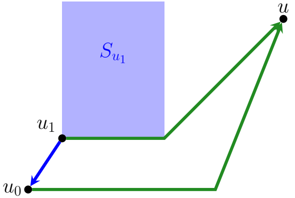


Non-crossing - green

The edges of T_1 do not cross each other.

Subtrees live in different areas (by construction). The bends : y-decreasing (by construction) ; x-increasing :

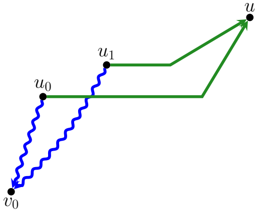
u_{i+1} descendant of u_i in T_0



The last descendant of u_i in T_0 is on the right to the one of u_{i+1}.

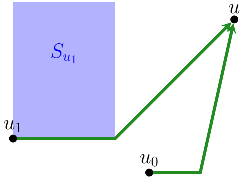
u_{i+1} not descendant of u_i in T_0

$x(u_{i+1}) > x(u_i)$



u_{i+1} should be below the edge (u_i, u)

$x(u_{i+1}) < x(u_i)$



Descendants of u_{i+1} are between $x(u_{i+1})$ and $x(u_i)$.

Conclusion

Open questions :

- Reduce the number of bends necessary to draw a given planar graph ?
- Caracterization of planar graphs for which a straight-lines rook-drawing is (not) possible
- What is the minimum grid size requested to draw a planar straight-lines rook-drawing for a given planar graph ?

Thank you for your attention !