Coloring triangle-free rectangular frame intersection graphs with O(log log n) colors

Tomasz Krawczyk (speaker)   Arkadiusz Pawlik
Bartosz Walczak

THEORETICAL COMPUTER SCIENCE
Jagiellonian University

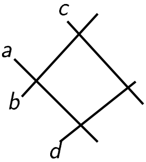
{krawczyk, pawlik, walczak}@tcs.uj.edu.pl

Forum Informatyki Teoretycznej
Toruń, 11-14 April, 2013

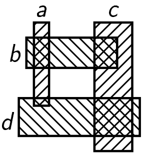
Geometric intersection graphs

$\mathcal{S}$ — a set of compact, arcwise connected objects in the plane.

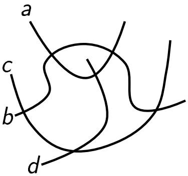
Geometric intersection graph of $\mathcal{S}$:
vertices $\leftrightarrow$ objects,   edges $\leftrightarrow$ intersecting objects.



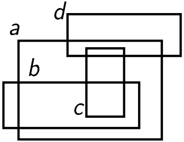
line segments
segment graph



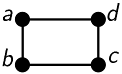
rectangles



curves
string graph



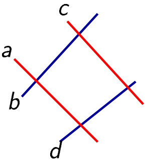
frames



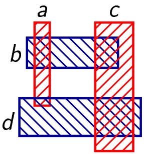
Geometric intersection graphs

$\mathcal{S}$ — a set of compact, arcwise connected objects in the plane.

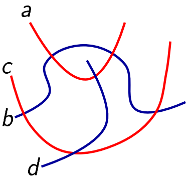
Geometric intersection graph of $\mathcal{S}$:
vertices $\leftrightarrow$ objects, edges $\leftrightarrow$ intersecting objects.



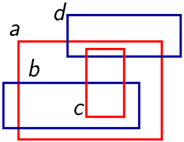
line segments
segment graph



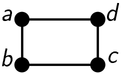
rectangles



curves
string graph



frames



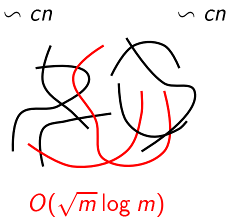
chromatic number - upper bounds

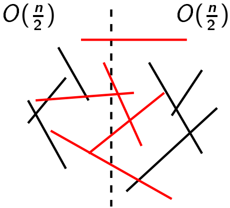
Upper bounds:

- string graphs – $O(\log^{\log \omega} n)$ – Fox, Pach, 2013
  - Separator Theorem for string graphs,
- segment graphs – $O(\log n)$ - Suk, 2012
  - segments piercing a common line have bounded chromatic number,
- rectangles – $O(\omega^2)$ – Asplund and Grünbaum.

Comments:

- nothing better than $O(\log n)$ was known.





chromatic number - lower bounds

Lower bounds:

- $\Omega(\log \log n)$ – Pawlik, Kozik, T.K., Lasoń, Micek, Walczak, Trotter, 2012
  - triangle-free segment/string graphs,
  - objects obtained by horizontal scaling, vertical scaling, and translation of some fixed object (but not rectangles),
- rectangles – 3ω – Kostochka.

Comments:

- obtained via on-line coloring games on intervals in the line!.

result

Theorem (T.K., Pawlik, Walczak, 2012)

Every triangle-free intersection graph of frames with n vertices can be colored with $O(\log \log n)$ colors.

Comments:

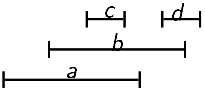
- the first algorithm that beats $O(\log n)$ bound,
- describes precisely the structure of frame intersection graphs,
- uses on-line coloring algorithms.

Problems:

- replace triangle-free with K_d-free,
- extend the method on segment graphs, L-shaped graphs,
- limitations of our method?

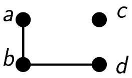
overlap graphs

Overlap (circle) graph:
vertices $\leftrightarrow$ intervals in the line, edges $\leftrightarrow$ overlapping intervals.



Theorem (Kostochka, Kratochvíl, 1997)

Every K_ω-free overlap graph can be colored with $50 \cdot 2^\omega$ colors!



overlap coloring game

K_ω-free overlap coloring game:

- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.

- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.

overlap coloring game

K_ω-free overlap coloring game:

- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.

overlap coloring game

K_ω-free overlap coloring game:

- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.

- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.

overlap coloring game

K_ω-free overlap coloring game:

- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.

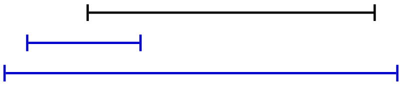
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.

overlap coloring game

K_ω-free overlap coloring game:

- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.

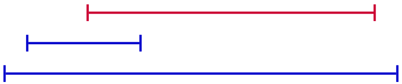
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

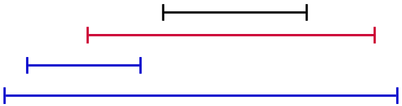
- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

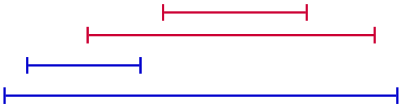
- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

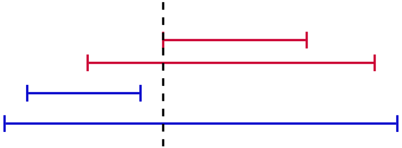
- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

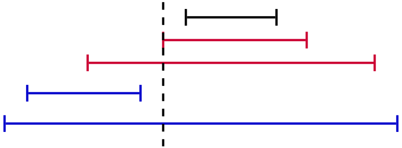
- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

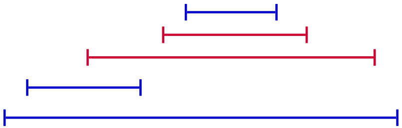
- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

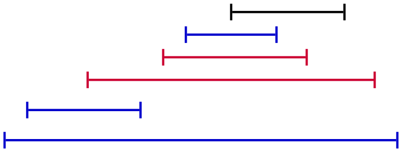
- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

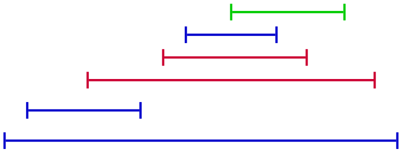
- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.

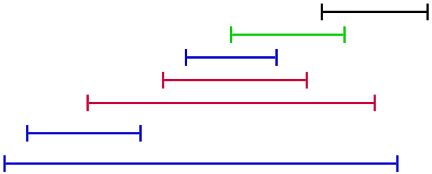


overlap coloring game

K_ω-free overlap coloring game:

- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.

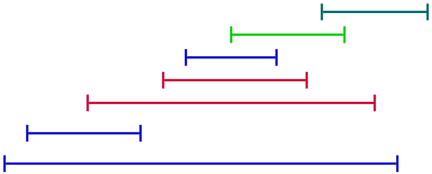
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

K_ω-free overlap coloring game:

- played by Presenter and Algorithm in rounds,
- Presenter builds a K_ω-free overlap graph:
  - one interval per round,
  - presentation order – consistent with left-endpoints relation.
- Algorithm colors intervals (immediately and irrevocably) so that no two overlapping intervals have the same color.



overlap coloring game

Observation (Pawlik, Kozik, T.K., Lasoń, Micek, Walczak, Trotter, 2012)

If Presenter has a strategy to force Algorithm to use c colors in h rounds of K_ω-free overlap coloring game

$$\Downarrow$$

There is a K_ω-free frame intersection graph with $2^{poly(h)}$ vertices and chromatic number at least c.

universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.

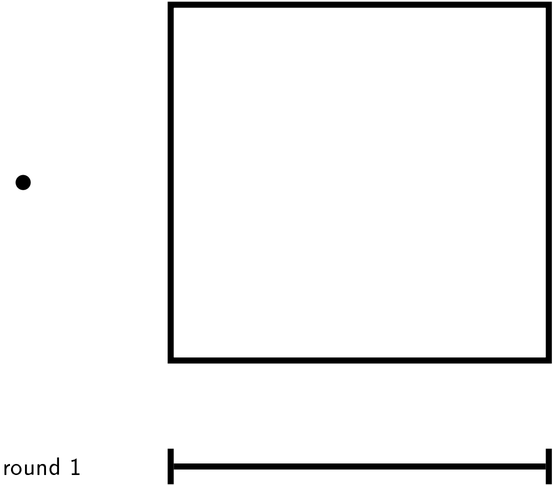
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.

round 1

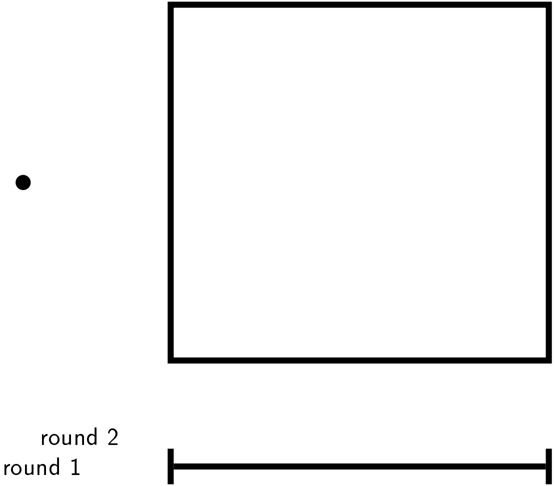
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



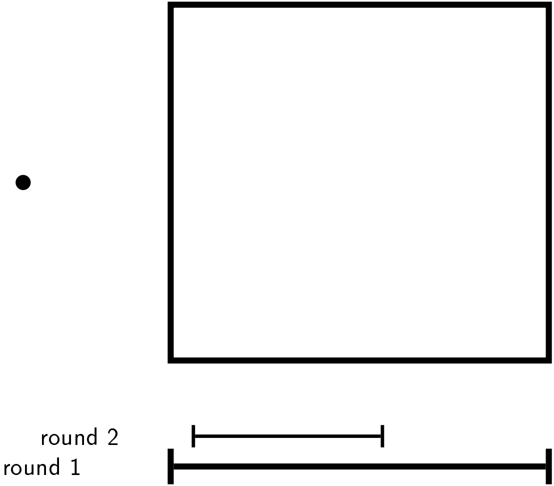
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



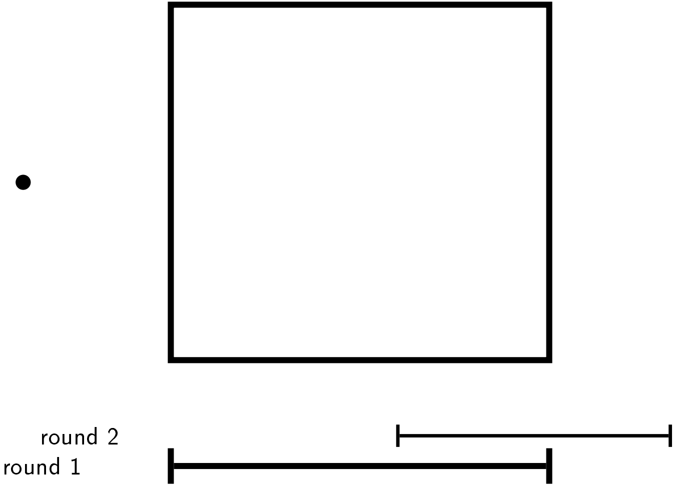
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



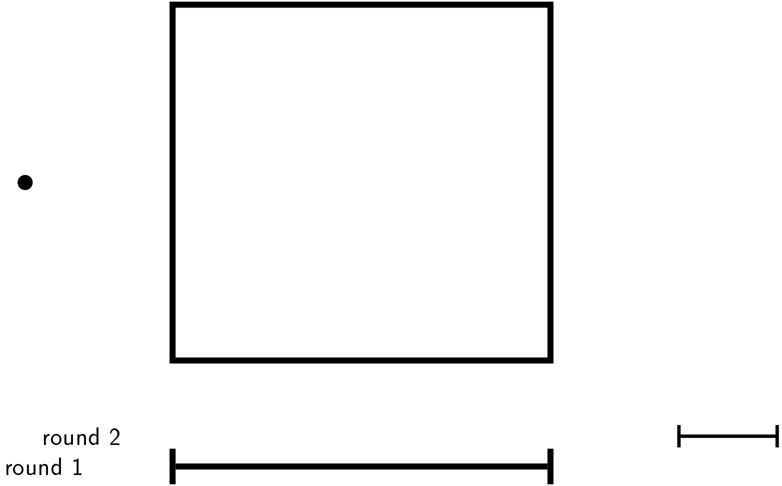
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



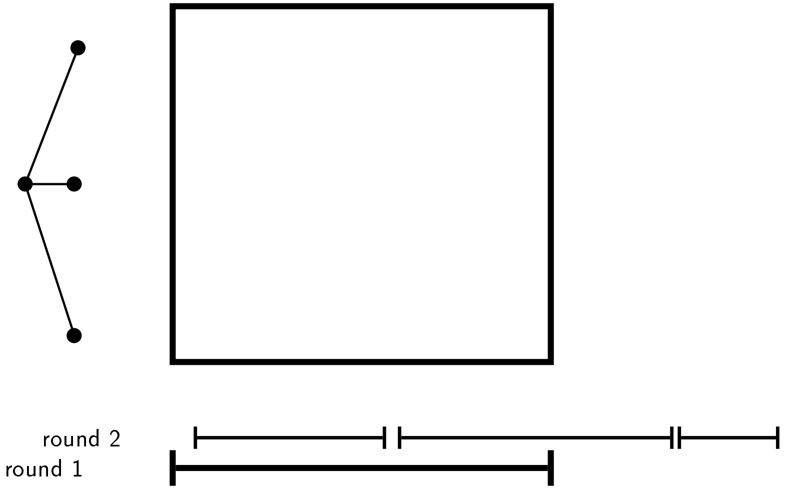
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



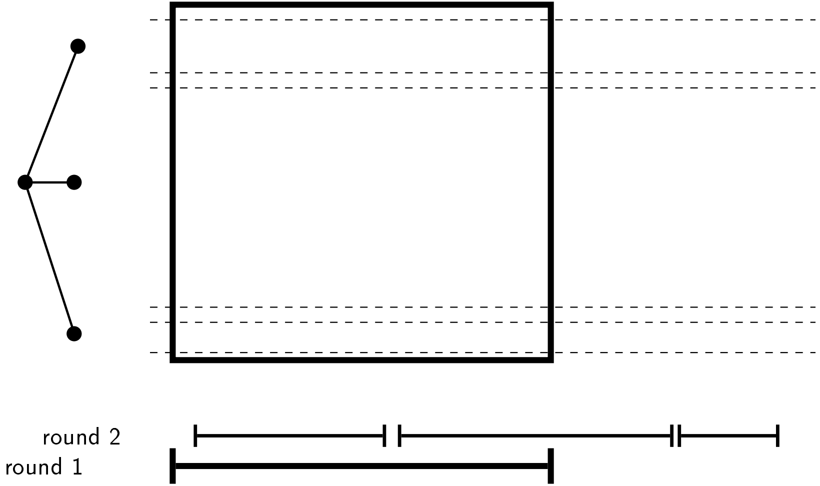
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



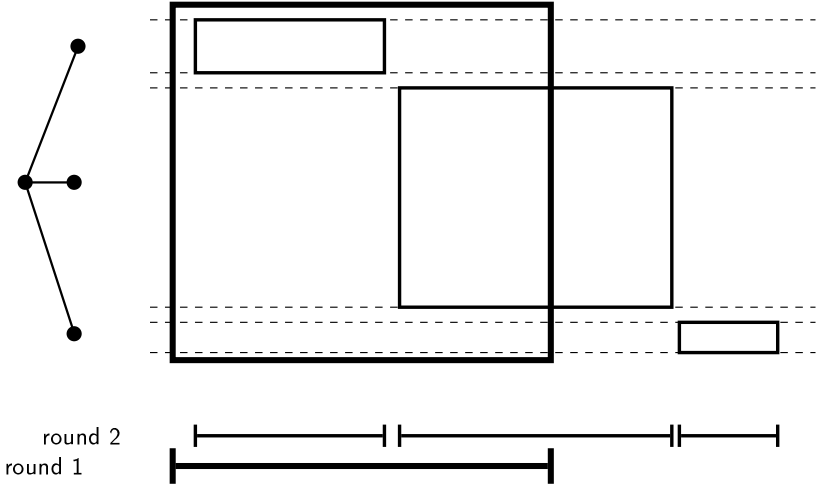
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



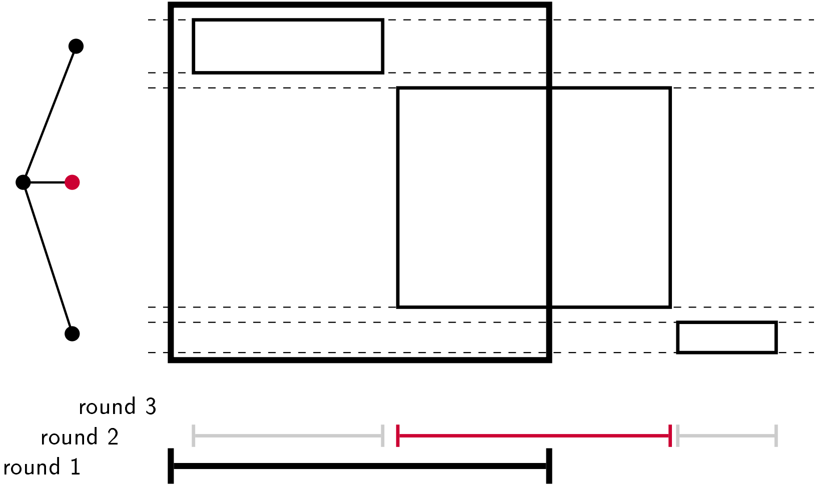
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



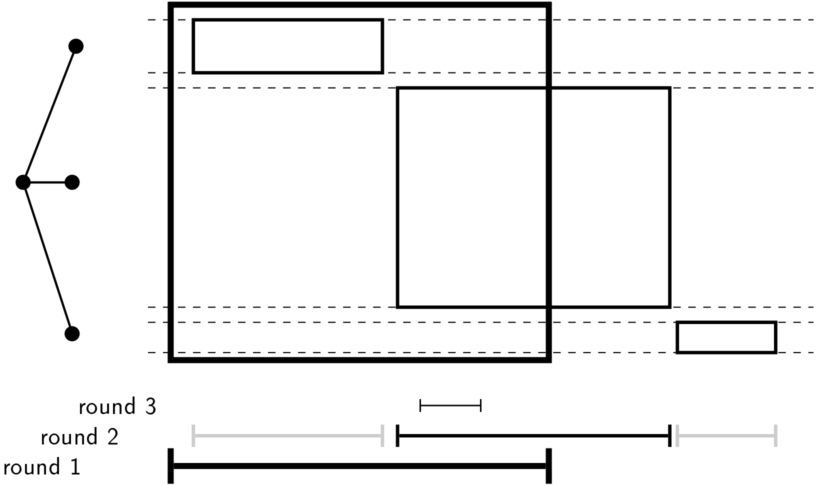
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



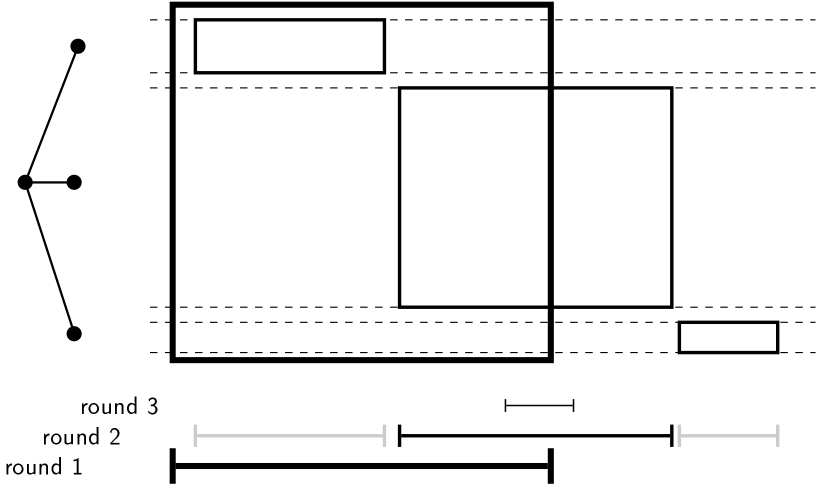
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



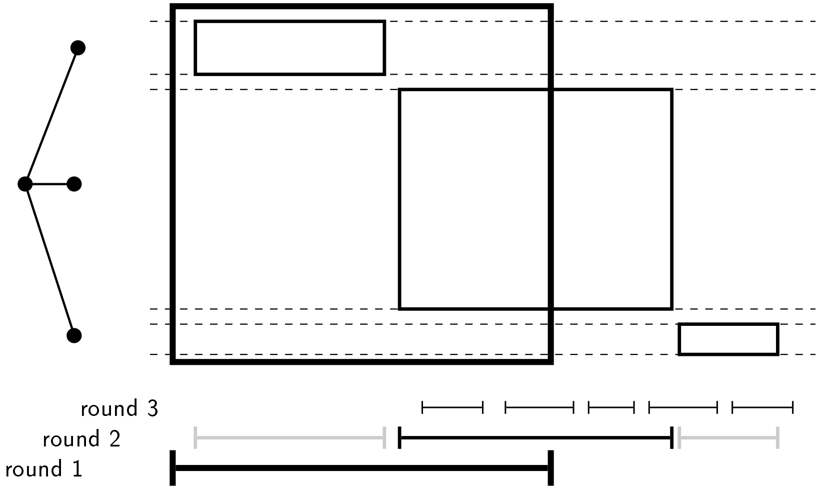
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



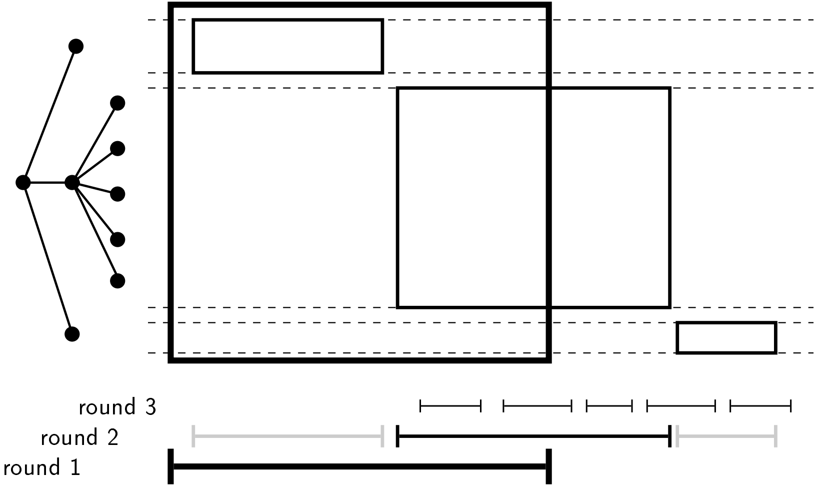
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



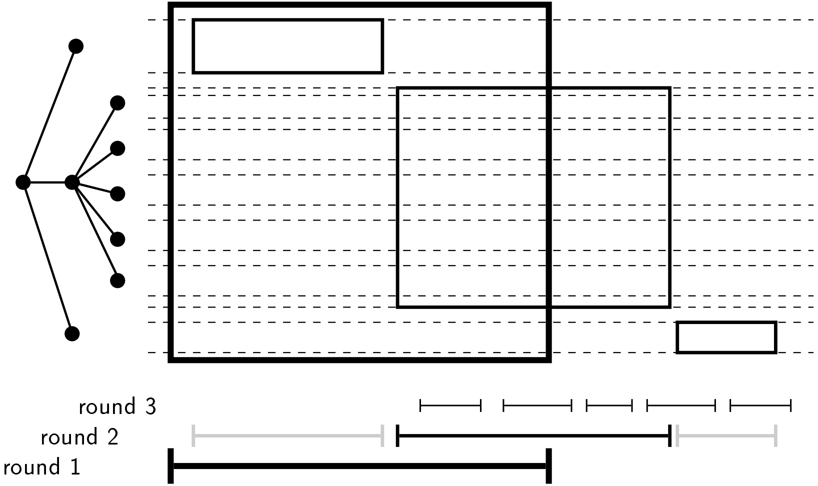
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



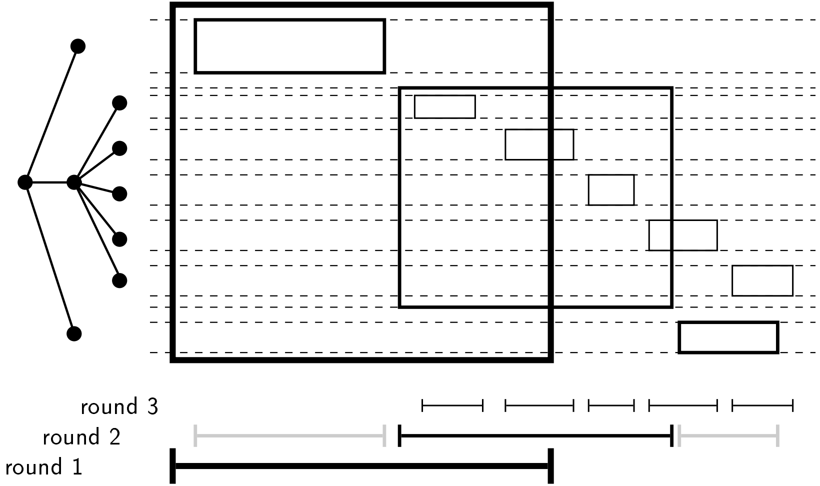
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



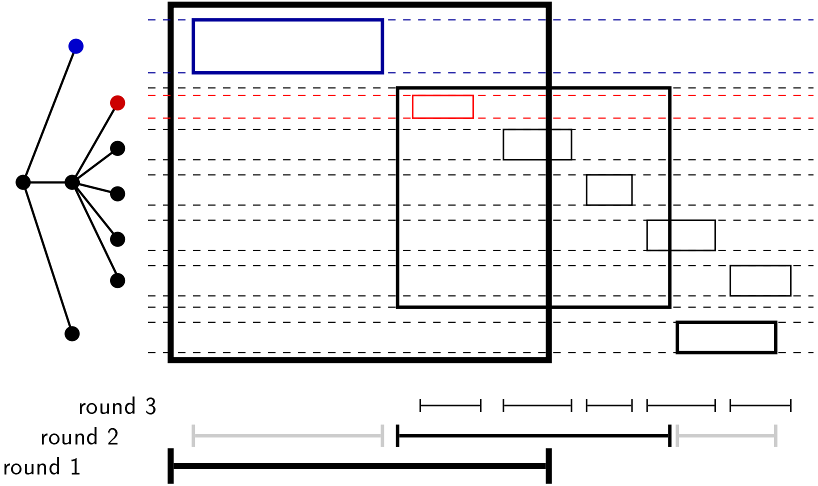
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



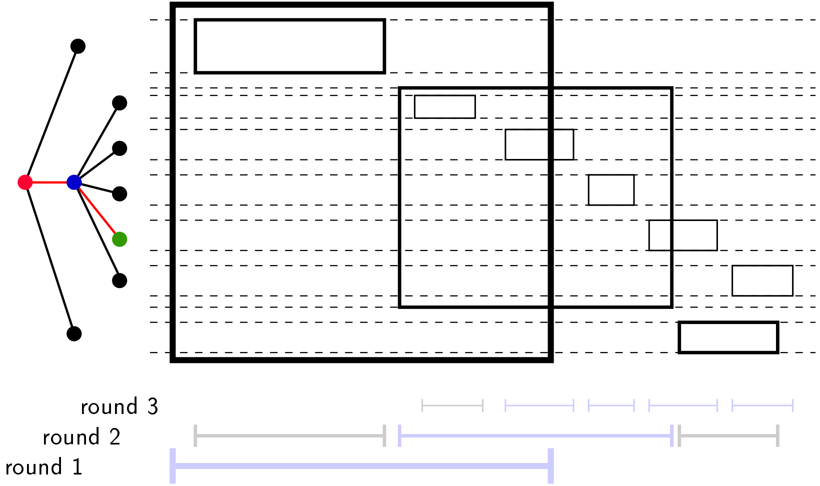
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



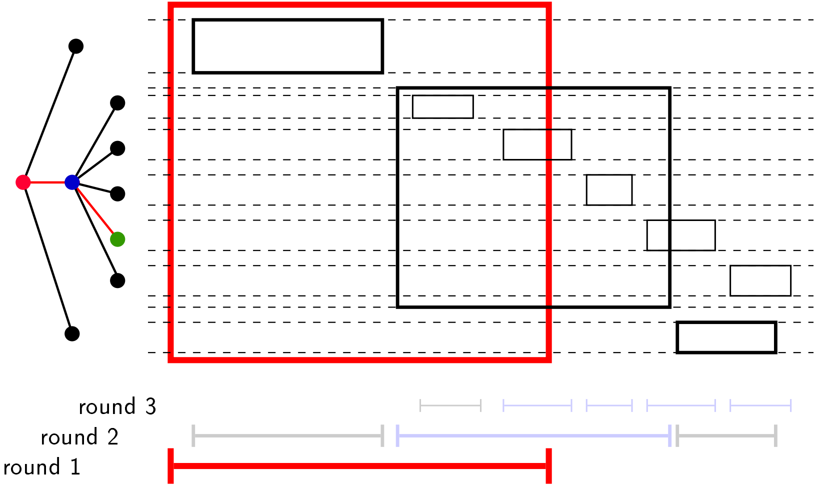
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



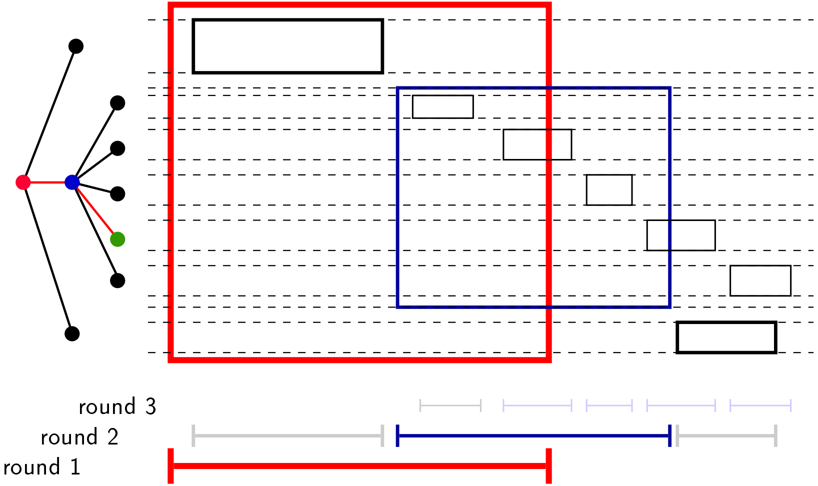
universal graph

h-universal graph – ‘encodes’ all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



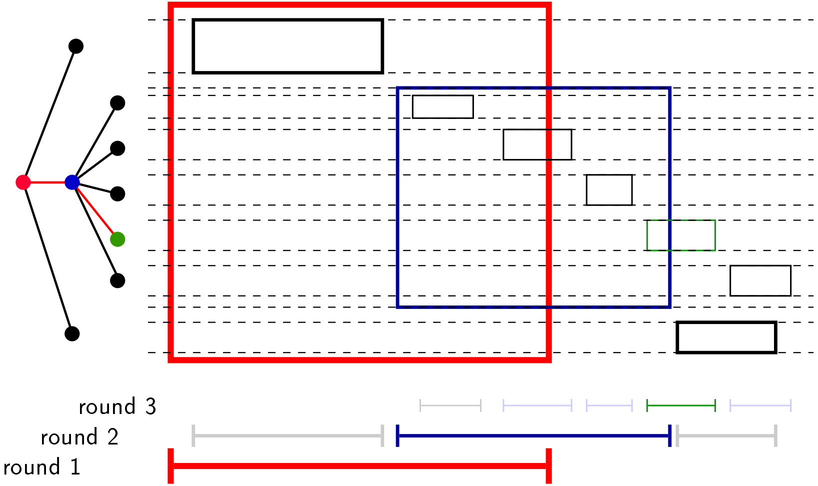
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



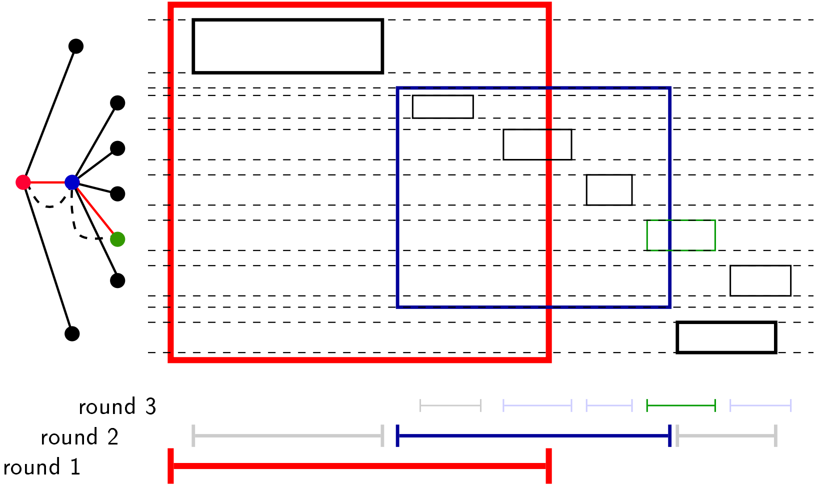
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



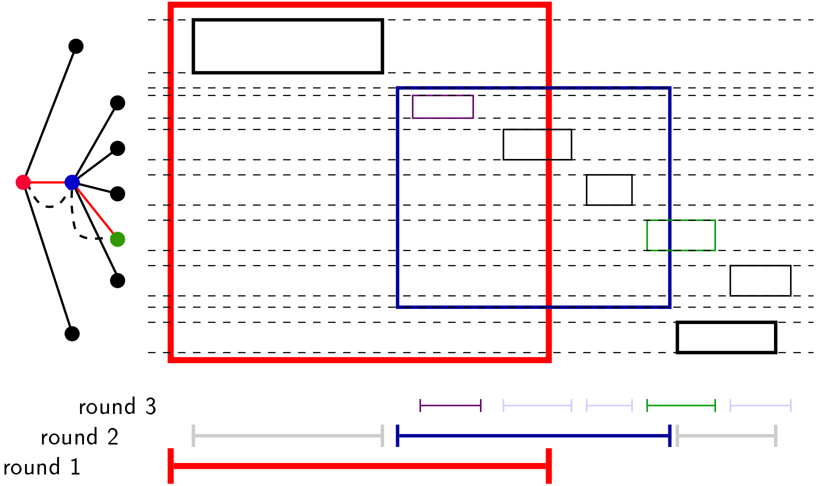
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



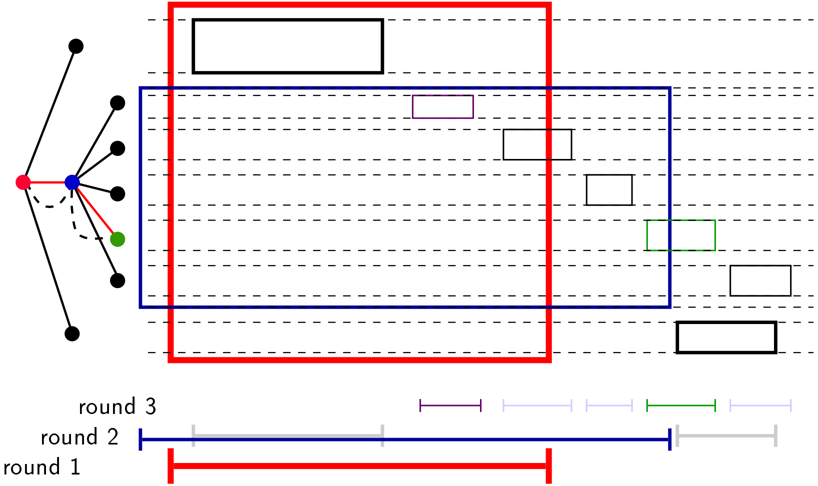
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



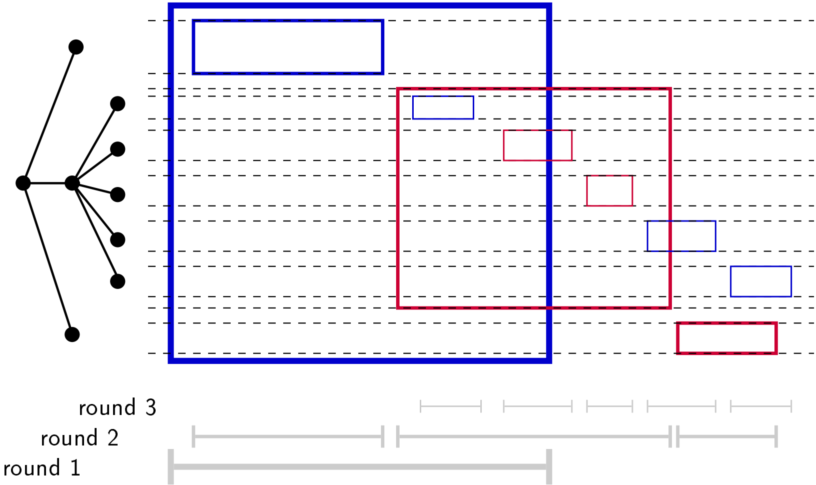
universal graph

h-universal graph – 'encodes' all possible moves of Presenter in the first h rounds of K_ω-free o.c. game.



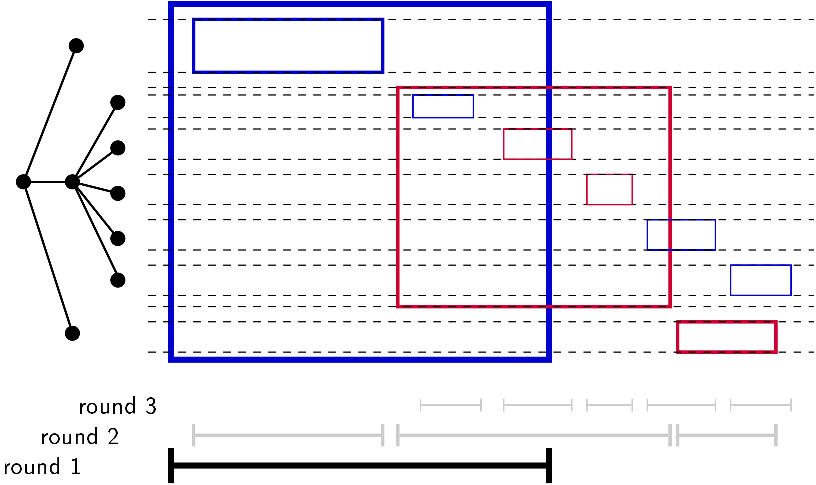
observation - proof

Correspondence: On-line algorithms – Proper colorings.



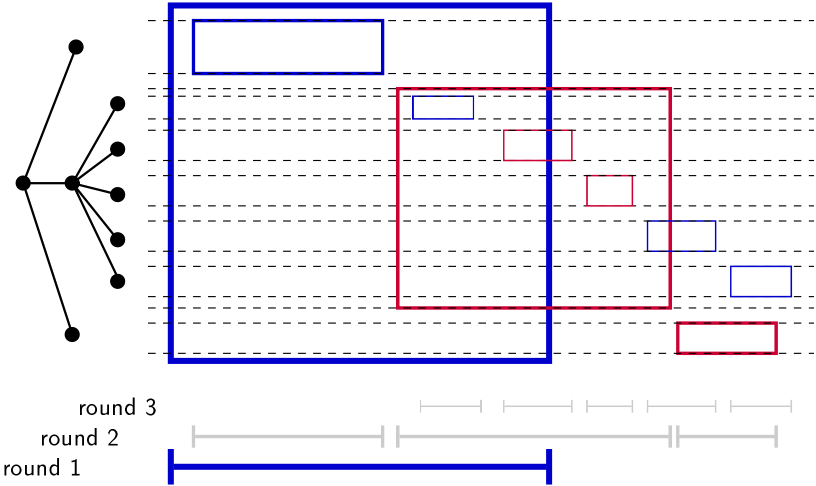
observation - proof

Correspondence: On-line algorithms – Proper colorings.



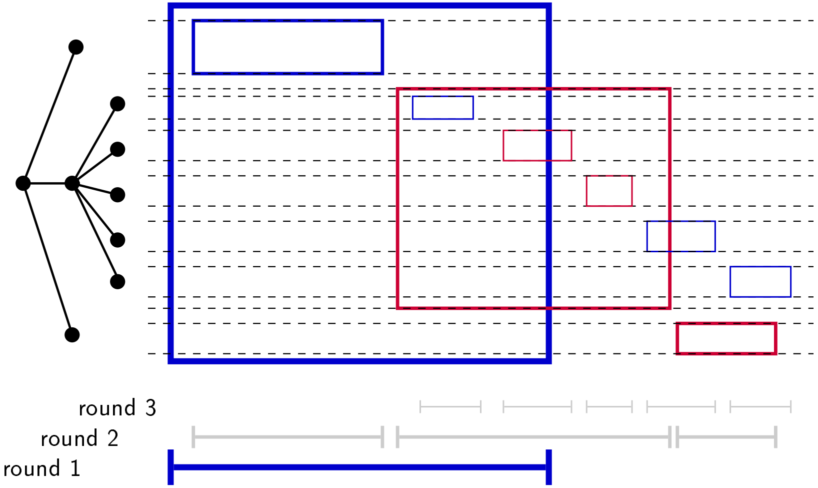
observation - proof

Correspondence: On-line algorithms – Proper colorings.



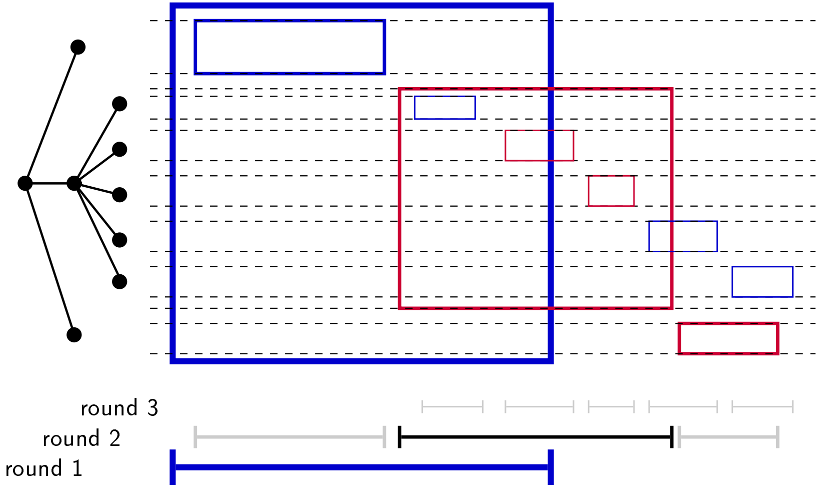
observation - proof

Correspondence: On-line algorithms – Proper colorings.



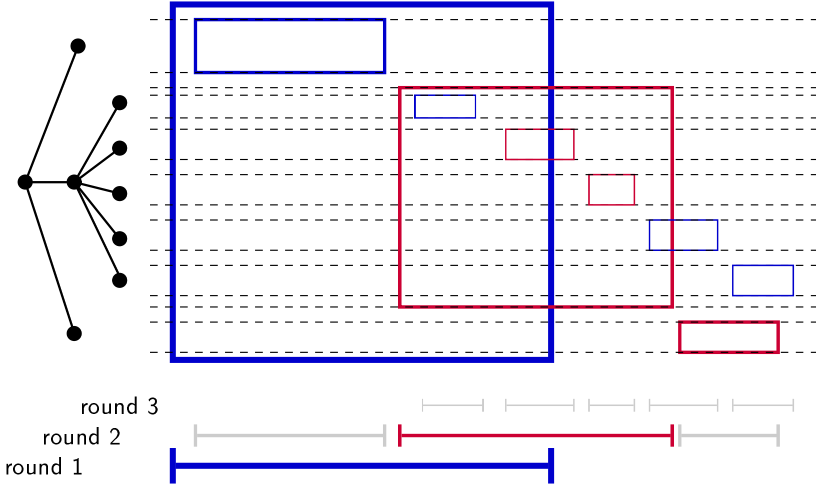
observation - proof

Correspondence: On-line algorithms – Proper colorings.



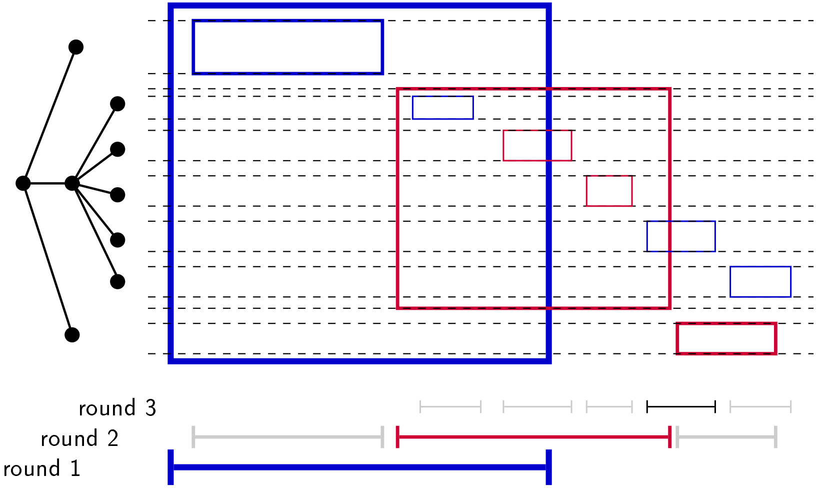
observation - proof

Correspondence: On-line algorithms – Proper colorings.



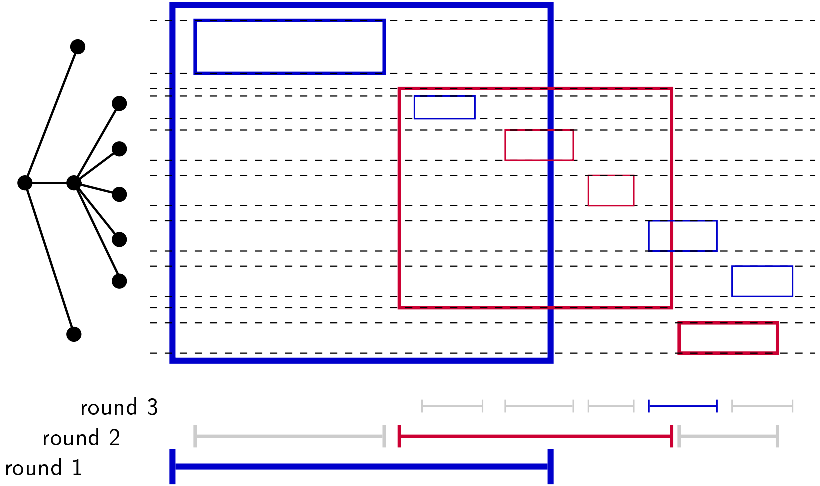
observation - proof

Correspondence: On-line algorithms – Proper colorings.



observation - proof

Correspondence: On-line algorithms – Proper colorings.



overlap coloring game - lower bounds

Observation (Pawlik, Kozik, T.K., Lasoń,Micek, Walczak, Trotter, 2012)

There is a strategy for Presenter that forces Algorithm to use $\log h$ *colors in* h *rounds of the triangle-free o.c. game.*

Comments:

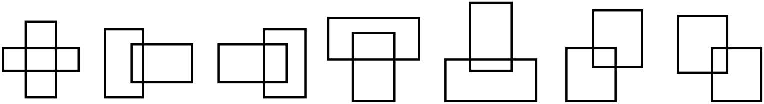
- there are triangle-free frame intersection graphs (universal graphs) with n vertices and chromatic number $\Omega(\log \log n)$,
- universal graphs can be represented also by segments in the plane.

upper bound for frames

On-line algorithms from overlap coloring games are also useful in frame coloring.

frames - intersection types

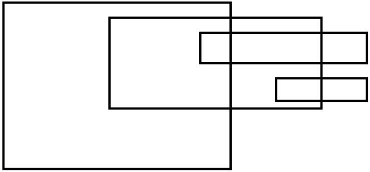
Intersection types:



a crossing rightward- leftward- downward- upward- diagonal intersections
directed

directed families of frames

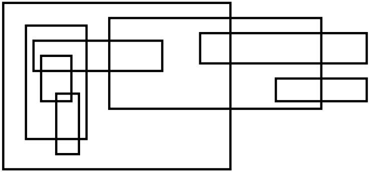
A family $\mathcal{F}$ of frames is rightward-directed (leftward-, upward-, downward-directed) if the intersection of every two frames from $\mathcal{F}$ is rightward-directed (leftward-, upward-, downward-directed).



decomposition theorem

Theorem (T.K., Pawlik, Walczak, 2012)

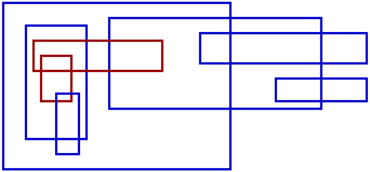
Every K_ω-free family of frames $\mathcal{F}$ can be partitioned into $f(\omega)$ componentwise directed subfamilies.



decomposition theorem

Theorem (T.K., Pawlik, Walczak, 2012)

Every K_ω-free family of frames $\mathcal{F}$ can be partitioned into $f(\omega)$ componentwise directed subfamilies.

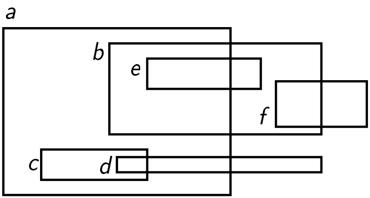


Overlap game graph – a connected, directed component.

- Projections of every two frames on OY-axis are either nested or disjoint,
- encode a 'partial' overlap coloring game (Presenter's power is limited).

Comments:

- encode a 'partial' overlap coloring game (Presenter's power is limited).

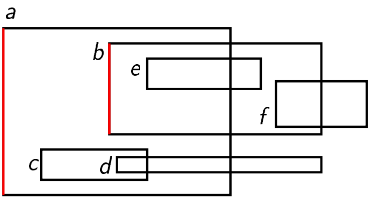


Overlap game graph – a connected, directed component.

- Projections of every two frames on OY-axis are either nested or disjoint,
- encode a 'partial' overlap coloring game (Presenter's power is limited).

Comments:

- encode a 'partial' overlap coloring game (Presenter's power is limited).

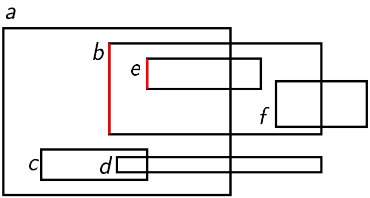


Overlap game graph – a connected, directed component.

- Projections of every two frames on OY-axis are either nested or disjoint,
- encode a 'partial' overlap coloring game (Presenter's power is limited).

Comments:

- encode a 'partial' overlap coloring game (Presenter's power is limited).

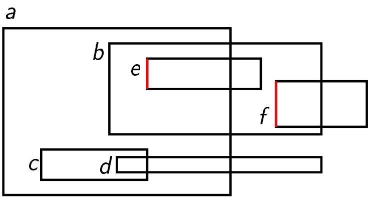


Overlap game graph – a connected, directed component.

- Projections of every two frames on OY-axis are either nested or disjoint,
- encode a 'partial' overlap coloring game (Presenter's power is limited).

Comments:

- encode a 'partial' overlap coloring game (Presenter's power is limited).

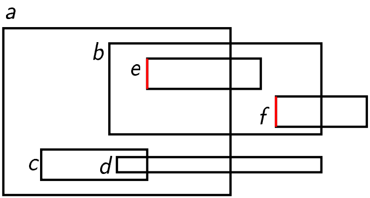


Overlap game graph – a connected, directed component.

- Projections of every two frames on OY-axis are either nested or disjoint,
- encode a 'partial' overlap coloring game (Presenter's power is limited).

Comments:

- encode a 'partial' overlap coloring game (Presenter's power is limited).

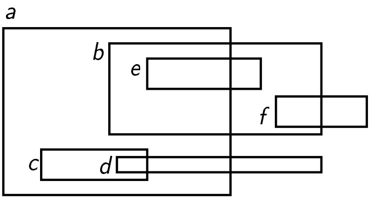


Overlap game graph – a connected, directed component.

- Projections of every two frames on OY-axis are either nested or disjoint,
- encode a 'partial' overlap coloring game (Presenter's power is limited).

Comments:

- encode a 'partial' overlap coloring game (Presenter's power is limited).

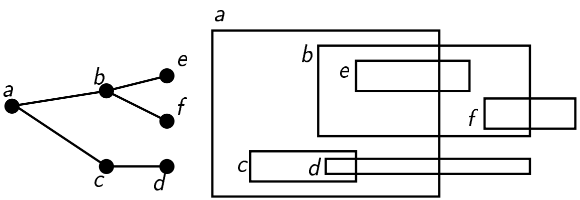


Overlap game graph – a connected, directed component.

- Projections of every two frames on OY-axis are either nested or disjoint,
- encode a 'partial' overlap coloring game (Presenter's power is limited).

Comments:

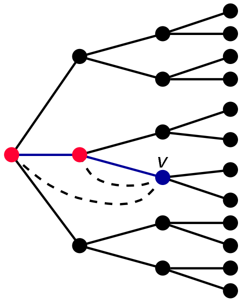
- encode a 'partial' overlap coloring game (Presenter's power is limited).



use of on-line algorithms

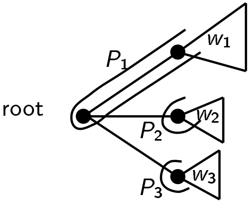
Coloring universal graphs:

- h-universal graph - overlap game graph that encodes the first h rounds of the game, ($2^{\mathrm{poly}(h)}$ vertices)
- use on-line algorithm! $O(\log h)$ colors.
- the chromatic number of universal graph is $O(\log \log n)$.



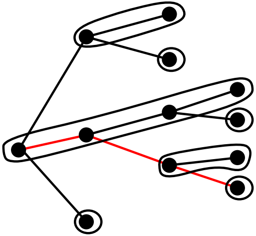
heavy-light decomposition

- introduced by Sleator and Tarjan, 1983,
- a partition of a rooted tree into heavy paths,
- $w_1 \in P_1 \implies |w_i| \leqslant |w_1|$
- if a path from the root to a leaf intersects k heavy paths, then the tree contains at least $\sim 2^k$ vertices.



heavy-light decomposition

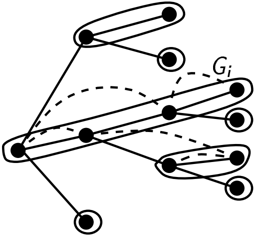
- introduced by Sleator and Tarjan, 1983,
- a partition of a rooted tree into heavy paths,
- $w_1 \in P_1 \implies |w_i| \leqslant |w_1|$
- if a path from the root to a leaf intersects k heavy paths, then the tree contains at least $\sim 2^k$ vertices.



algorithm

Unbalanced trees:

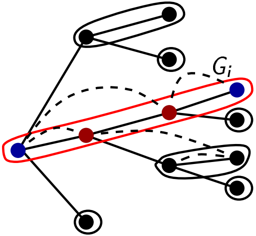
- heavy-light decomposition,
- color vertices in each heavy paths (overlap graph) (first coordinate)
- use modified on-line algorithm,
- if the k-th color is used then the path from the root to the vertex intersects at least $\backsim 2^k$ heavy paths,
- the graph contains at least $\backsim 2^{2^k}$ vertices.



algorithm

Unbalanced trees:

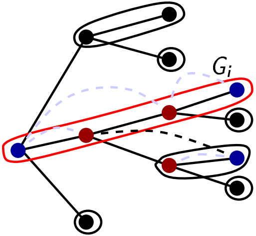
- heavy-light decomposition,
- color vertices in each heavy paths (overlap graph) (first coordinate)
- use modified on-line algorithm,
- if the k-th color is used then the path from the root to the vertex intersects at least $\backsim 2^k$ heavy paths,
- the graph contains at least $\backsim 2^{2^k}$ vertices.



algorithm

Unbalanced trees:

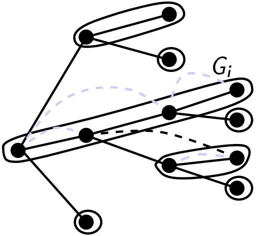
- heavy-light decomposition,
- color vertices in each heavy paths (overlap graph) (first coordinate)
- use modified on-line algorithm,
- if the k-th color is used then the path from the root to the vertex intersects at least $\backsim 2^k$ heavy paths,
- the graph contains at least $\backsim 2^{2^k}$ vertices.



algorithm

Unbalanced trees:

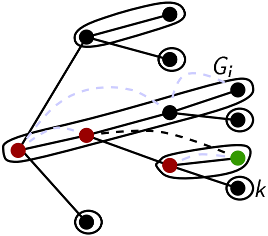
- heavy-light decomposition,
- color vertices in each heavy paths (overlap graph) (first coordinate)
- use modified on-line algorithm,
- if the k-th color is used then the path from the root to the vertex intersects at least $\backsim 2^k$ heavy paths,
- the graph contains at least $\backsim 2^{2^k}$ vertices.



algorithm

Unbalanced trees:

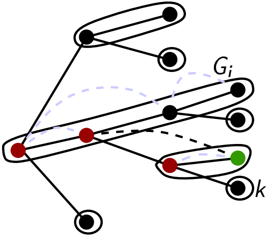
- heavy-light decomposition,
- color vertices in each heavy paths (overlap graph) (first coordinate)
- use modified on-line algorithm,
- if the k-th color is used then the path from the root to the vertex intersects at least $\backsim 2^k$ heavy paths,
- the graph contains at least $\backsim 2^{2^k}$ vertices.



algorithm

Unbalanced trees:

- heavy-light decomposition,
- color vertices in each heavy paths (overlap graph) (first coordinate)
- use modified on-line algorithm,
- if the k-th color is used then the path from the root to the vertex intersects at least $\backsim 2^k$ heavy paths,
- the graph contains at least $\backsim 2^{2^k}$ vertices.



Theorem (T.K., Pawlik, Walczak, 2012)

- *There is a strategy for Algorithm that uses $O(\log k)$ colors in k rounds of the triangle-free overlap coloring game.*
- *The above strategy can be adapted to work also with heavy-light decomposition.*
- *We can color triangle-free frame intersection graph with n vertices using $O(\log \log n)$ colors.*

Question:

- Is there a strategy for Algorithm in K_ω-free overlap coloring game that uses $O(\log n)$ colors?

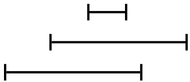
partial results

Theorem (T.K., Pawlik, Wlaczak, 2013)

There is a strategy for Algorithm that uses $O(\log n)$ *colors provided Presenter builds clean overlap graphs .*

Theorem (Kostochka, Kratochvíl, 1997)

Every K_ω *overlap graph can be colored with* $50 \cdot 2^\omega$ *colors!*



Theorem (Kostochka, Milans, 2010)

Every K_ω *clean overlap graph can be colored with* $2\omega - 1$ *colors!*

Questions:

- Does every K_ω-free segment graph (L-shaped intersection graph) partitions into $f(\omega)$ overlap game graphs?
- Does every segment graph with high chromatic number contains a large (of linear size) overlap game graph?
- (Fox, Pach) Does every K_ω-free segment/string graph contains an independent set of linear size (cn for some $c > 0$)?
- Does every K_ω-free frame intersection graph contains an independent set of linear size (cn for some $c > 0$)?