Widths, Games and Logic

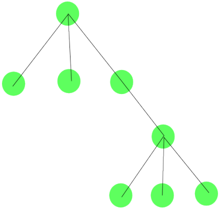
Robert Ganian

June 28, 2015

ALGORITHMS AND COMPLEXITY GROUP

Trees

Trees



Nice and simple – most problems are easy on trees.

Treewidth

Treewidth measures how "tree-like" a graph is.
Why?

Treewidth

Treewidth measures how "tree-like" a graph is.
Why?

- "tree-like" structure can be exploited for algorithms
- Many problems FPT parameterized by treewidth
- Applications in all kinds of fields, many real-world systems have low treewidth
- $\sim$9k hits on google scholar, $\sim$100k hits on google

Undisputed king of structural parameters

Does my graph have bounded treewidth?

Several equivalent definitions of treewidth

This talk: Definition via Cops and Robber game
(Easy to use) (Also: most fun)

Does my graph have bounded treewidth?

Several equivalent definitions of treewidth

This talk: Definition via Cops and Robber game
(Easy to use) (Also: most fun)

- k cops vs. 1 robber
- Robber is always on one vertex and can move along edges, but is extremely fast
- Each cop can land on one vertex. Vertex is blocked and cannot be used by robber until cop lifts off
  - Before cop actually lands, the robber can still move.

How many cops do we need to catch the robber?

Does my graph have bounded treewidth?

Several equivalent definitions of treewidth

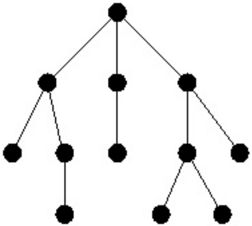
This talk: Definition via Cops and Robber game
(Easy to use) (Also: most fun)

- k cops vs. 1 robber
- Robber is always on one vertex and can move along edges, but is extremely fast
- Each cop can land on one vertex. Vertex is blocked and cannot be used by robber until cop lifts off
    - Before cop actually lands, the robber can still move.

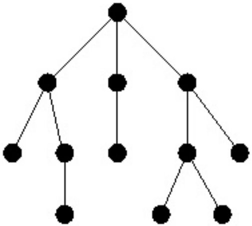
How many cops do we need to catch the robber? Subtract 1 and you get treewidth.

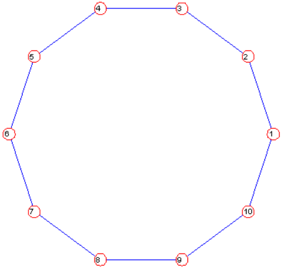
- There are many algorithms for computing treewidth: FPT, approximation, heuristics...

Cops and Robber examples

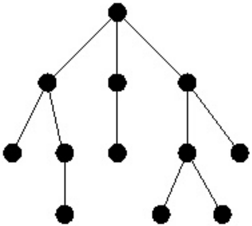


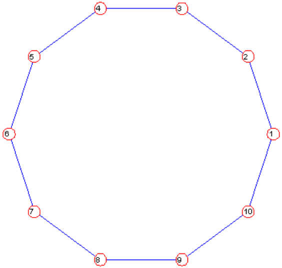
Cops and Robber examples

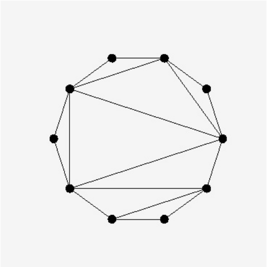




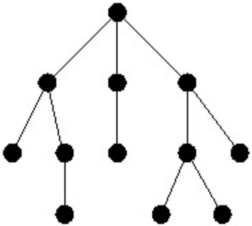
Cops and Robber examples

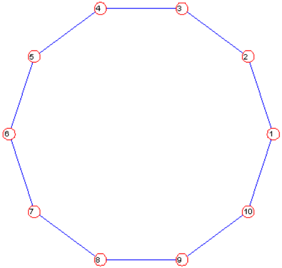


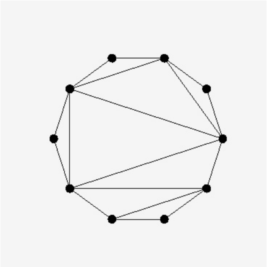


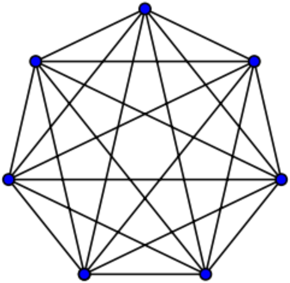


Cops and Robber examples









Using treewidth

Any problem which can be expressed by a small MSO formula is FPT par. by treewidth

Example of **Algorithmic Meta-Theorem**... we also develop them

Using treewidth

Any problem which can be expressed by a small MSO formula is FPT par. by treewidth

Example of **Algorithmic Meta-Theorem**... we also develop them

MSO formulas

- Quantifiers over individual vertices/edges, vertex/edge sets
- Can check $\neq$, $\in$, vertex-edge incidence "inc" (and vertex-vertex and edge-edge adjacency "adj")
- standard logical connectives

Using treewidth

Any problem which can be expressed by a small MSO formula is FPT par. by treewidth

Example of **Algorithmic Meta-Theorem**... we also develop them

MSO formulas

- Quantifiers over individual vertices/edges, vertex/edge sets
- Can check $\neq$, $\in$, vertex-edge incidence "inc" (and vertex-vertex and edge-edge adjacency "adj")
- standard logical connectives

Can express simple stuff: Each vertex has some incident edge

$$\forall v \exists e : \mathrm{inc}(v, e)$$

Using treewidth

Any problem which can be expressed by a small MSO formula is FPT par. by treewidth

Example of **Algorithmic Meta-Theorem**... we also develop them

MSO formulas

- Quantifiers over individual vertices/edges, vertex/edge sets
- Can check $\neq$, $\in$, vertex-edge incidence "inc" (and vertex-vertex and edge-edge adjacency "adj")
- standard logical connectives

Can express simple stuff: Each vertex has some incident edge

$$\forall v \exists e : \mathrm{inc}(v, e)$$

There exists an independent set of vertices

$$\exists C \forall a, b : (a \in C \land b \in C) \implies \neg \mathrm{adj}(a, b)$$

Using MSO logic

Can also express NP-hard problems: 3-colorability

Using MSO logic

Can also express NP-hard problems: 3-colorability

$$\exists A, B, C : \text{independent}(A) \wedge \text{independent}(B) \wedge \text{independent}(C) \wedge$$

$$\wedge (\forall v : v \in A \vee v \in B \vee v \in C)$$

What about connectivity?

Using MSO logic

Can also express NP-hard problems: 3-colorability

$$\exists A, B, C : \text{independent}(A) \wedge \text{independent}(B) \wedge \text{independent}(C) \wedge$$

$$\wedge (\forall v : v \in A \vee v \in B \vee v \in C)$$

What about connectivity?

- For every A and B, if A and B are complements then there exists an edge between $a \in A$ and $b \in B$

Hamiltonian cycle?

Using MSO logic

Can also express NP-hard problems: 3-colorability

$$\exists A, B, C : \text{independent}(A) \wedge \text{independent}(B) \wedge \text{independent}(C) \wedge$$

$$\wedge (\forall v : v \in A \vee v \in B \vee v \in C)$$

What about connectivity?

- For every A and B, if A and B are complements then there exists an edge between $a \in A$ and $b \in B$

Hamiltonian cycle?

- There exists an edge-set H such that each vertex is incident to two edges from H, and H is connected.

Using MSO logic

Can also express NP-hard problems: 3-colorability

$$\exists A, B, C : \text{independent}(A) \wedge \text{independent}(B) \wedge \text{independent}(C) \wedge$$

$$\wedge (\forall v : v \in A \vee v \in B \vee v \in C)$$

What about connectivity?

- For every A and B, if A and B are complements then there exists an edge between $a \in A$ and $b \in B$

Hamiltonian cycle?

- There exists an edge-set H such that each vertex is incident to two edges from H, and H is connected.

Can also optimize over size of sets

- Find smallest/largest set A such that ...
- Vertex Cover, Independent Set, Dominating Set...

Clique-width

Treewidth is bad for dense graphs

- But dense graphs have structure too!

Clique-width

Treewidth is bad for dense graphs

- But dense graphs have structure too!

Solution: Clique-width

- Clique-width measures how close a graph is to... not just cliques
- More general than treewidth
  - Whenever treewidth is bounded, then so is clique-width
  - Can also be small on graphs with high treewidth (cliques)
- Allows solution of many problems (but less than treewidth)

Does my graph have small clique-width?

Building game (like Lego)

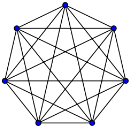
You can use labels $1 \ldots k$ on vertices, and have these operations:

- Create a new graph with one vertex with label i
- Make disjoint union of graphs
- add all edges between labels i and j
- change all labels i to j

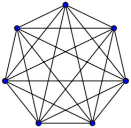
Clique-width = minimum number of labels you need to build the graph

Examples

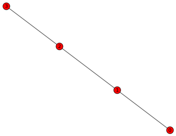
- Complete graph



Examples

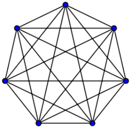


- Complete graph

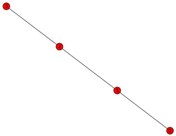


- Path

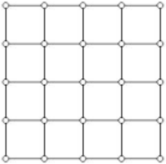
Examples



- Complete graph



- Path



- Grid
- Tree? (Think when bored)

Using clique-width

Any problem expressible in a “weaker” MSO logic is FPT par. by clique-width

- Cannot speak about edges

Can we express 3-Colorability?

Using clique-width

Any problem expressible in a "weaker" MSO logic is FPT par. by clique-width

- Cannot speak about edges

Can we express 3-Colorability? Vertex Cover?

Using clique-width

Any problem expressible in a “weaker” MSO logic is FPT par. by clique-width

- Cannot speak about edges

Can we express 3-Colorability? Vertex Cover? Hamiltonian Cycle?

Using clique-width

Any problem expressible in a “weaker” MSO logic is FPT par. by clique-width

- Cannot speak about edges

Can we express 3-Colorability? Vertex Cover? Hamiltonian Cycle?

rank-width $\sim$ clique-width 2.0

- Computing rank-width $\times$ Computing clique-width

Relation to our research:

- Use treewidth, clique-width, rank-width in our results
- Study other measures of structure (parameter ecology)
- Develop new meta-theorems

Relation to our research:

- Use treewidth, clique-width, rank-width in our results
- Study other measures of structure (parameter ecology)
- Develop new meta-theorems

Thank you for your attention.