Reu Problems of the Czech Group

Adam Juraszek, Tomáš Masařík, Jitka Novotná,
Martin Töpfer, Tomáš Toufar, Jan Voborník, Peter Zeman

DIMACS REU 2015, Piscataway

Complexity of Fair Deletion Problems

Complexity of Graph Deletion Problems

Some important algorithmic problems in graph theory can be stated as deletion problems – i.e. find a set such that graph satisfies certain property after deletion of such set

- Vertex Cover - Find a set $W \subseteq V$ such that $G[V \setminus W]$ contains no edge

Complexity of Graph Deletion Problems

Some important algorithmic problems in graph theory can be stated as deletion problems – i.e. find a set such that graph satisfies certain property after deletion of such set

- Vertex Cover - Find a set $W \subseteq V$ such that $G[V \setminus W]$ contains no edge
- Feedback Vertex Set - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is acyclic

Complexity of Graph Deletion Problems

Some important algorithmic problems in graph theory can be stated as deletion problems – i.e. find a set such that graph satisfies certain property after deletion of such set

- Vertex Cover - Find a set $W \subseteq V$ such that $G[V \setminus W]$ contains no edge
- Feedback Vertex Set - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is acyclic
- Feedback Arc Set - Find a set $F \subseteq E$ such that $(V, E \setminus F)$ is acyclic

Complexity of Graph Deletion Problems

Some important algorithmic problems in graph theory can be stated as deletion problems – i.e. find a set such that graph satisfies certain property after deletion of such set

- Vertex Cover - Find a set $W \subseteq V$ such that $G[V \setminus W]$ contains no edge
- Feedback Vertex Set - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is acyclic
- Feedback Arc Set - Find a set $F \subseteq E$ such that $(V, E \setminus F)$ is acyclic
- Odd Cycle Transversal - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is bipartite

Complexity of Graph Deletion Problems

Some important algorithmic problems in graph theory can be stated as deletion problems – i.e. find a set such that graph satisfies certain property after deletion of such set

- VERTEX COVER - Find a set $W \subseteq V$ such that $G[V \setminus W]$ contains no edge
- FEEDBACK VERTEX SET - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is acyclic
- FEEDBACK ARC SET - Find a set $F \subseteq E$ such that $(V, E \setminus F)$ is acyclic
- ODD CYCLE TRANSVERSAL - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is bipartite

The goal is to find smallest such set

Complexity of Graph Deletion Problems

Some important algorithmic problems in graph theory can be stated as deletion problems – i.e. find a set such that graph satisfies certain property after deletion of such set

- Vertex Cover - Find a set $W \subseteq V$ such that $G[V \setminus W]$ contains no edge
- Feedback Vertex Set - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is acyclic
- Feedback Arc Set - Find a set $F \subseteq E$ such that $(V, E \setminus F)$ is acyclic
- Odd Cycle Transversal - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is bipartite

The goal is to find smallest such set
All those problems are NP-complete

Complexity of Graph Deletion Problems

Some important algorithmic problems in graph theory can be stated as deletion problems – i.e. find a set such that graph satisfies certain property after deletion of such set

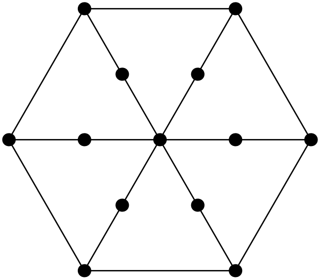
- Vertex Cover - Find a set $W \subseteq V$ such that $G[V \setminus W]$ contains no edge
- Feedback Vertex Set - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is acyclic
- Feedback Arc Set - Find a set $F \subseteq E$ such that $(V, E \setminus F)$ is acyclic
- Odd Cycle Transversal - Find a set $W \subseteq V$ such that $G[V \setminus W]$ is bipartite

The goal is to find smallest such set
All those problems are NP-complete
However, they are fixed-parameter tractable (in short FPT)

Fair Versions of Deletion Problems

Sometimes the smallest set is not what we want

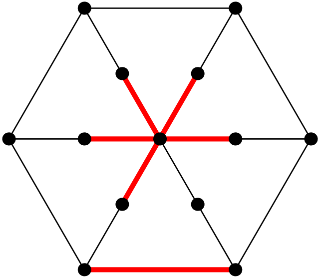
- We have a communication network, we want to remove redundancies
- We don't want to reduce the capability of each vertex by too much (a local condition in contrast to the previous global one)



Fair Versions of Deletion Problems

Sometimes the smallest set is not what we want

- We have a communication network, we want to remove redundancies
- We don't want to reduce the capability of each vertex by too much (a local condition in contrast to the previous global one)

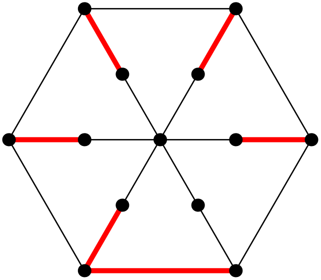


This solution removes a lot of links from single vertex

Fair Versions of Deletion Problems

Sometimes the smallest set is not what we want

- We have a communication network, we want to remove redundancies
- We don't want to reduce the capability of each vertex by too much (a local condition in contrast to the previous global one)



A much better solution

Our Aim

FPT results concerning fair versions of problems are still not very well explored

Our Aim

FPT results concerning fair versions of problems are still not very well explored

Our aim:

- There is a result stating that every problem definable in a modification of MSO logic can be solved by an FPT algorithm (Kolman et al.)
- Find fair versions of problems that are definable in MSO logic or prove their inexpressibility via Ehrenfeucht-Fräissé games

Our Aim

FPT results concerning fair versions of problems are still not very well explored

Our aim:

- There is a result stating that every problem definable in a modification of MSO logic can be solved by an FPT algorithm (Kolman et al.)
- Find fair versions of problems that are definable in MSO logic or prove their inexpressibility via Ehrenfeucht-Fräissé games
- The result also works only for a specific fairness condition (number of deleted edges incident to a vertex), we want to explore other reasonable fairness conditions
- We plan to tackle this problem by extending the so-called homomorphism-based method by Borie et al.

Our Aim

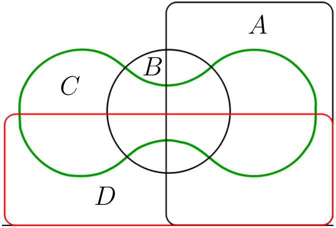
FPT results concerning fair versions of problems are still not very well explored

Our aim:

- There is a result stating that every problem definable in a modification of MSO logic can be solved by an FPT algorithm (Kolman et al.)
- Find fair versions of problems that are definable in MSO logic or prove their inexpressibility via Ehrenfeucht-Fräissé games
- The result also works only for a specific fairness condition (number of deleted edges incident to a vertex), we want to explore other reasonable fairness conditions
- We plan to tackle this problem by extending the so-called homomorphism-based method by Borie et al.
- The current parameterization is by tree-width; we want to explore different parameterizations that will lead to FPT algorithms

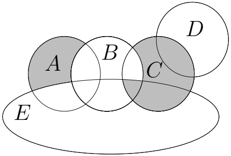
Simplification of the Inclusion-exclusion Formula

Inclusion-exclusion Principle



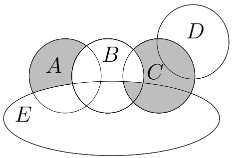
$$
\begin{aligned}
|A \cup B \cup C \cup D| = & \\
& |A| + |B| + |C| + |D| \\
& -|A \cap B| - |A \cap C| - |A \cap D| - |B \cap C| - |B \cap D| - |C \cap D| \\
& +|A \cap B \cap C| + |A \cap B \cap D| + |A \cap C \cap D| + |B \cap C \cap D| \\
& -|A \cap B \cap C \cap D|
\end{aligned}
$$

Simplified Formula



$$|A \cup B \cup C \cup D| = |A| + |B| + |C| + |D| - 3|A \cap B \cap C \cap D|$$

Simplified Formula



$$|A \cup B \cup C \cup D| = |A| + |B| + |C| + |D| - 3|A \cap B \cap C \cap D|$$

Short formulas can be found by methods from linear algebra.

Simplified Formula 2

Problem: In practice, evaluating formulae with large coefficients amplify measurement errors.

Simplified Formula 2

Problem: In practice, evaluating formulae with large coefficients amplify measurement errors.

We want the formula to be

- short,

Simplified Formula 2

Problem: In practice, evaluating formulae with large coefficients amplify measurement errors.

We want the formula to be

- short,
- with small coefficients,

Simplified Formula 2

Problem: In practice, evaluating formulae with large coefficients amplify measurement errors.

We want the formula to be

- short,
- with small coefficients,
- easy to find.

Research Problems

Considering formulas with coefficients from the set $\{-1, 1, 0\}$.

Q: Given m nonempty regions what is the shortest formula?

Q: Is it NP-hard to decide whether there is a formula with at most k terms?

Known results:

Lower Bound: There are families of sets requiring $\Omega(m^{2-\varepsilon})$ terms.

Upper Bound: There is an algorithm which finds a formula of size $m^{\mathcal{O}(\log^2(n))}$ in expected running time $m^{\mathcal{O}(\log^2(n))}$.

Stronger results for specific families of sets (e.g. disks in a plane).

Our Aim

We will focus on

- Complexity of the decision problem about existence of a short formula with $\{-1, 0, 1\}$ coefficients.

Our Aim

We will focus on

- Complexity of the decision problem about existence of a short formula with $\{-1, 0, 1\}$ coefficients.
- Studying structural properties of specific set systems described by Venn diagram regions.

Our Aim

We will focus on

- Complexity of the decision problem about existence of a short formula with $\{-1, 0, 1\}$ coefficients.
- Studying structural properties of specific set systems described by Venn diagram regions.
- Generalization of the results for specific families of set systems.

Our Aim

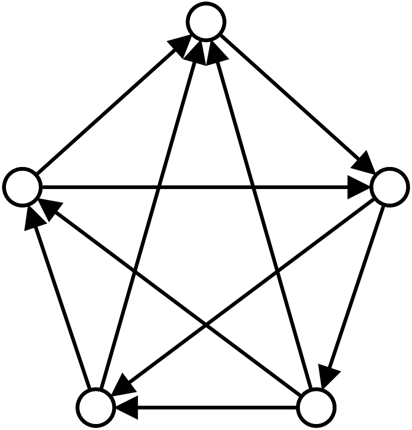
We will focus on

- Complexity of the decision problem about existence of a short formula with $\{-1, 0, 1\}$ coefficients.
- Studying structural properties of specific set systems described by Venn diagram regions.
- Generalization of the results for specific families of set systems.
- Improving the upper bound.

Dominating Sets on Colored Tournaments

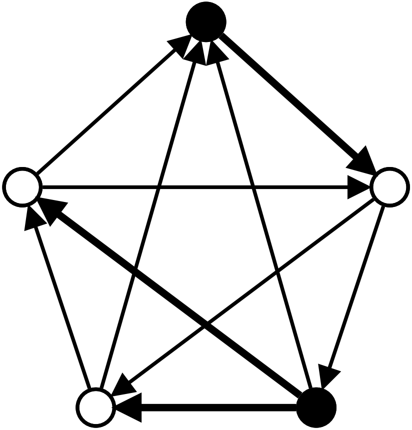
Dominating Set on Tournaments

- A tournament is a complete graph with orientated edges.
- A vertex v is dominated by u if there is an edge $u \to v$.
- A dominating set is a set of vertices called dominators such that every vertex of the graph is a dominator or is dominated by at least one.



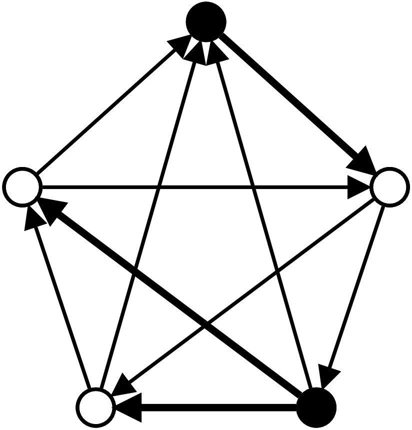
Dominating Set on Tournaments

- A tournament is a complete graph with orientated edges.
- A vertex v is dominated by u if there is an edge $u \to v$.
- A dominating set is a set of vertices called dominators such that every vertex of the graph is a dominator or is dominated by at least one.



Dominating Set on Tournaments

- A tournament is a complete graph with orientated edges.
- A vertex v is dominated by u if there is an edge $u \rightarrow v$.
- A dominating set is a set of vertices called dominators such that every vertex of the graph is a dominator or is dominated by at least one.

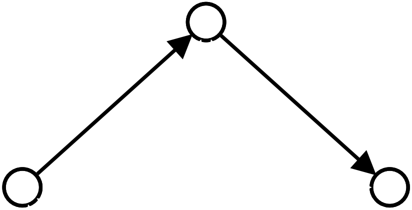


Our goal

We are looking for an upper bound of the size of the dominating set on colored tournaments.

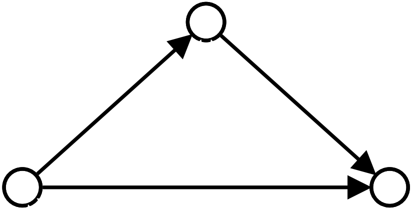
Edge-transitive Tournaments

- An edge-transitive digraph: $u \to v \& v \to w \implies u \to w$
- This means the graph must be acyclic.
- On transitive tournaments, one vertex dominates all.



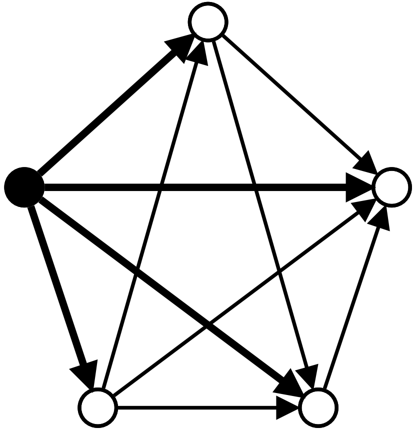
Edge-transitive Tournaments

- An edge-transitive digraph: $u \to v \& v \to w \implies u \to w$
- This means the graph must be acyclic.
- On transitive tournaments, one vertex dominates all.



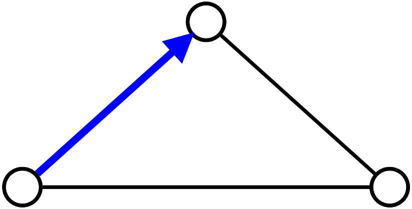
Edge-transitive Tournaments

- An edge-transitive digraph:
  $u \to v \& v \to w \implies u \to w$
- This means the graph must be acyclic.
- On transitive tournaments, one vertex dominates all.



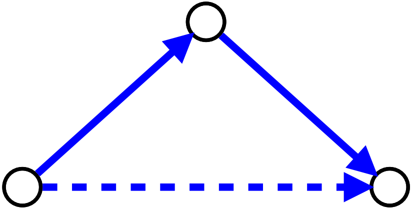
2-colored Transitive Tournaments

- Every edge is either red, or blue.
- Edges of the same color satisfy transitivity.
- Then whole graph is transitive and it has a single dominator.



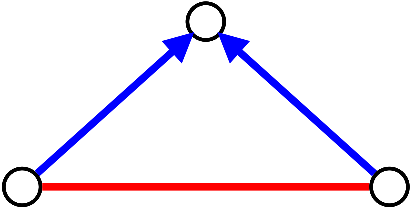
2-colored Transitive Tournaments

- Every edge is either red, or blue.
- Edges of the same color satisfy transitivity.
- Then whole graph is transitive and it has a single dominator.



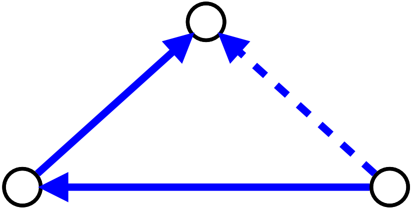
2-colored Transitive Tournaments

- Every edge is either red, or blue.
- Edges of the same color satisfy transitivity.
- Then whole graph is transitive and it has a single dominator.



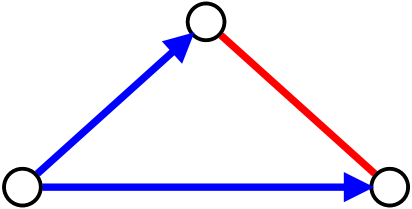
2-colored Transitive Tournaments

- Every edge is either red, or blue.
- Edges of the same color satisfy transitivity.
- Then whole graph is transitive and it has a single dominator.



2-colored Transitive Tournaments

- Every edge is either red, or blue.
- Edges of the same color satisfy transitivity.
- Then whole graph is transitive and it has a single dominator.



k-colored Transitive Tournaments

What if we assumed more colors?

- What would be the upper bound for the size of the dominating set?

k-colored Transitive Tournaments

What if we assumed more colors?

- What would be the upper bound for the size of the dominating set?
- It is conjectured that it depends only on number of colors.

k-colored Transitive Tournaments

What if we assumed more colors?

- What would be the upper bound for the size of the dominating set?
- It is conjectured that it depends only on number of colors.

Plan

We will test small cases algorithmically and then try to generalize the result for tournaments of an arbitrary size.

Subclasses of Chordal Graphs

Symmetries of Graphs

The study of symmetries of geometrical objects is an ancient topic in mathematics and its precise formulation led to group theory.

Symmetries of Graphs

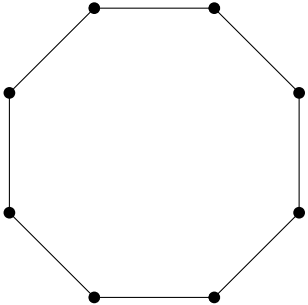
The study of symmetries of geometrical objects is an ancient topic in mathematics and its precise formulation led to group theory.

Symmetries of graphs are described by their automorphisms.

Symmetries of Graphs

The study of symmetries of geometrical objects is an ancient topic in mathematics and its precise formulation led to group theory.

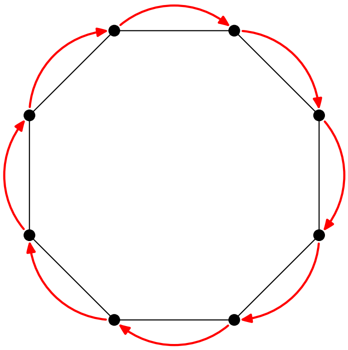
Symmetries of graphs are described by their automorphisms.



Symmetries of Graphs

The study of symmetries of geometrical objects is an ancient topic in mathematics and its precise formulation led to group theory.

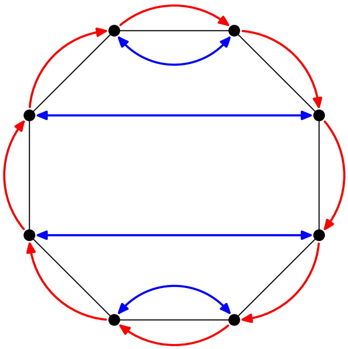
Symmetries of graphs are described by their automorphisms.



Symmetries of Graphs

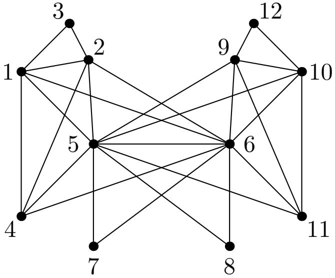
The study of symmetries of geometrical objects is an ancient topic in mathematics and its precise formulation led to group theory.

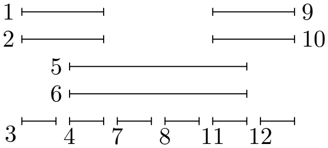
Symmetries of graphs are described by their automorphisms.



Interval Graphs (INT)

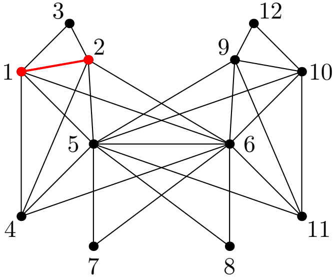
Interval graphs are intersection graphs of intervals of the real line.

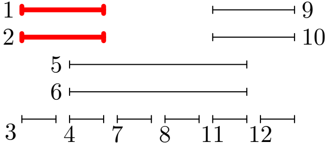




Interval Graphs (INT)

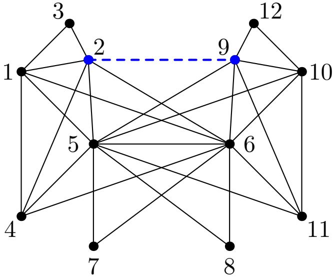
Interval graphs are intersection graphs of intervals of the real line.

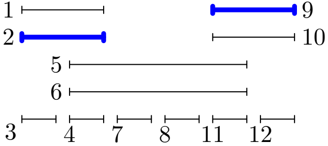




Interval Graphs (INT)

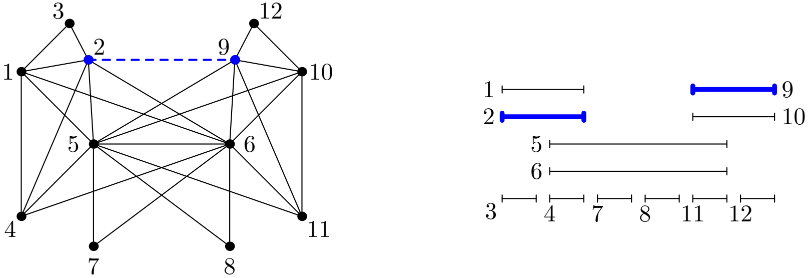
Interval graphs are intersection graphs of intervals of the real line.





Interval Graphs (INT)

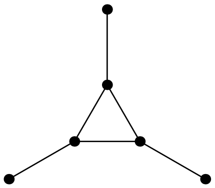
Interval graphs are intersection graphs of intervals of the real line.



Many computational problems are polynomially solvable for interval graphs (graph isomorphism, maximum clique, *k*-coloring, maximum independent set, ...).

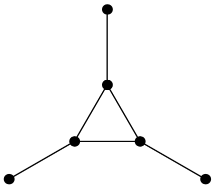
Chordal Graphs (CHOR)

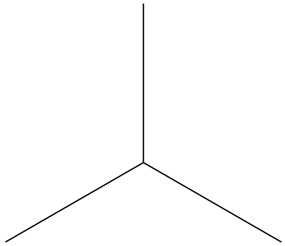
Chordal graphs are intersection graphs of subtrees of a tree.



Chordal Graphs (CHOR)

Chordal graphs are intersection graphs of subtrees of a tree.



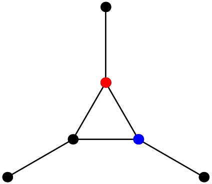


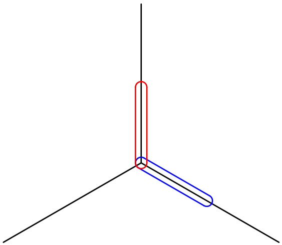
Chordal Graphs (CHOR)

Chordal graphs are intersection graphs of subtrees of a tree.

Chordal Graphs (CHOR)

Chordal graphs are intersection graphs of subtrees of a tree.



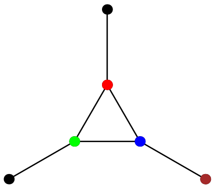


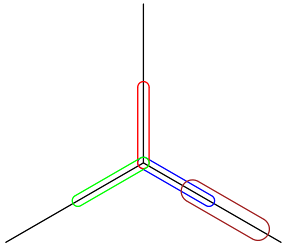
Chordal Graphs (CHOR)

Chordal graphs are intersection graphs of subtrees of a tree.

Chordal Graphs (CHOR)

Chordal graphs are intersection graphs of subtrees of a tree.



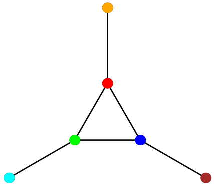


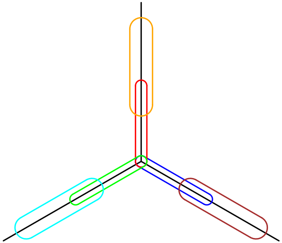
Chordal Graphs (CHOR)

Chordal graphs are intersection graphs of subtrees of a tree.

Chordal Graphs (CHOR)

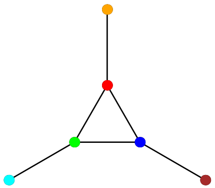
Chordal graphs are intersection graphs of subtrees of a tree.

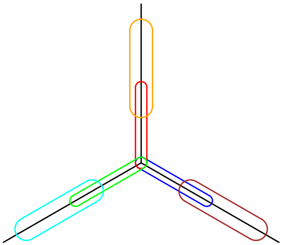




Chordal Graphs (CHOR)

Chordal graphs are intersection graphs of subtrees of a tree.





It is easy to see that INT $\subsetneq$ CHOR.

T-GRAPH

For a fixed tree T, the class of all intersection graphs of subtrees of T will be denoted by T-GRAPH.

T-GRAPH

For a fixed tree T, the class of all intersection graphs of subtrees of T will be denoted by T-GRAPH.

We obtain an infinite hierarchy between INT and CHOR, since

$$\text{INT} \subseteq T\text{-GRAPH} \subsetneq \text{CHOR}.$$

T-GRAPH

For a fixed tree T, the class of all intersection graphs of subtrees of T will be denoted by T-GRAPH.

We obtain an infinite hierarchy between INT and CHOR, since

$$\text{INT} \subseteq T\text{-GRAPH} \subsetneq \text{CHOR}.$$

The automorphism groups of interval graphs were recently characterized (P. Klavík, P. Zeman).

The automorphism groups of chordal graphs are known to be universal (K. Booth, G. Lueker).

T-GRAPH

For a fixed tree T, the class of all intersection graphs of subtrees of T will be denoted by T-GRAPH.

We obtain an infinite hierarchy between INT and CHOR, since

$$\mathrm{INT} \subseteq T\text{-GRAPH} \subsetneq \mathrm{CHOR}.$$

The automorphism groups of interval graphs were recently characterized (P. Klavík, P. Zeman).

The automorphism groups of chordal graphs are known to be universal (K. Booth, G. Lueker).

Concerning the class T-GRAPH, not much is known in comparison to INT and CHOR.

Main Aim

Our main aim is to study the structural and computational properties of the classes T-GRAPH.

First, we will try to understand the classes T-GRAPH, for some special tree T (for example $K_{1,n}$).

It is possible that some techniques developed for interval graphs could be used.

Thank you!