Algorithmic Meta-Theorems

192.122

WS21/22

Jan Dreier

dreier@ac.tuwien.ac.at

ALGORITHMS AND COMPLEXITY GROUP

TU WIEN Informatics

Algorithmic Meta-Theorems

192.122
WS21/22

Jan Dreier
dreier@ac.tuwien.ac.at

ALGORITHMS AND COMPLEXITY GROUP
TU WIEN Informatics

Content

Algorithmic Meta-Theorems lie at intersection of

- Algorithms,
- Graphs,
- Logic.

Content

Algorithmic Meta-Theorems lie at intersection of

- Algorithms,
- Graphs,
- Logic.

“[Algorithmic Meta-Theorems] are results of the form: every computational problem that can be formalised in a given logic L can be solved efficiently on every class C of structures satisfying certain conditions.”

Stefan Kreuzer

Content

We learn how to

Content

We learn how to

- express graph problems as logical formulas (MSO, FO),

Content

We learn how to

- express graph problems as logical formulas (MSO, FO),
- evaluate MSO formulas on tree-like graphs,

Content

We learn how to

- express graph problems as logical formulas (MSO, FO),
- evaluate MSO formulas on tree-like graphs,
- evaluate FO formulas on sparse graphs,

Content

We learn how to

- express graph problems as logical formulas (MSO, FO),
- evaluate MSO formulas on tree-like graphs,
- evaluate FO formulas on sparse graphs,
- (transfer these results to dense graphs).

Content

We learn how to

- express graph problems as logical formulas (MSO, FO),
- evaluate MSO formulas on tree-like graphs,
- evaluate FO formulas on sparse graphs,
- (transfer these results to dense graphs).

To do so, we

- use tools from logic (Gaifman, Feferman–Vaught, Ehrenfeucht-Fraïssé, ...),

Content

We learn how to

- express graph problems as logical formulas (MSO, FO),
- evaluate MSO formulas on tree-like graphs,
- evaluate FO formulas on sparse graphs,
- (transfer these results to dense graphs).

To do so, we

- use tools from logic (Gaifman, Feferman–Vaught, Ehrenfeucht-Fraïssé, …),
- learn to work with treewidth

Content

We learn how to

- express graph problems as logical formulas (MSO, FO),
- evaluate MSO formulas on tree-like graphs,
- evaluate FO formulas on sparse graphs,
- (transfer these results to dense graphs).

To do so, we

- use tools from logic (Gaifman, Feferman–Vaught, Ehrenfeucht-Fraïssé, . . .),
- learn to work with treewidth
- discover the right notion of "sparse graphs",

Content

We learn how to

- express graph problems as logical formulas (MSO, FO),
- evaluate MSO formulas on tree-like graphs,
- evaluate FO formulas on sparse graphs,
- (transfer these results to dense graphs).

To do so, we

- use tools from logic (Gaifman, Feferman–Vaught, Ehrenfeucht-Fraïssé, …),
- learn to work with treewidth
- discover the right notion of "sparse graphs",
- work through central proofs (from a high level perspective).

Pros and Cons

Should *You* use algorithmic meta-theorems?

Pros and Cons

Should *You* use algorithmic meta-theorems?

Pros:

- Quick and easy way to check if something is tractable in principle.

Pros and Cons

Should *You* use algorithmic meta-theorems?

Pros:

- Quick and easy way to check if something is tractable in principle.
- Powerful algorithms to use as a subroutine in other problems.

Pros and Cons

Should *You* use algorithmic meta-theorems?

Pros:

- Quick and easy way to check if something is tractable in principle.
- Powerful algorithms to use as a subroutine in other problems.

Cons:

- For specific problems much slower than “handcrafted” algorithms.

Why do *I* care about them?

Why do *I* care about them?

Central piece in the quest for a general theory of *tractable graph classes*.

Structure and Dates

- Semester hours: 2.0
- Credits: 3.0

- Weekly Lectures: Tuesdays, 11:00 – 12:30
  - Seminarraum FAV 01 B (Seminarraum 187/2)
  - also online (link in Tuwel)
  - The slides will be available on Tuwel.

- Bi-Weekly Exercises: Fridays, 11:00 – 12:30
  - Seminarraum FAV 01 B (Seminarraum 187/2)
  - also online (link in Tuwel)
  - You get problems related to lecture and solve them together.
  - Separate session at 09:15 for online participants?

- No mandatory homework exercises.

Grading

- You get a passing grade “befriedigend” if you actively participate in four all bi-weekly exercise sessions.
- Active partitiicipation means
  - being present (either online or in person),
  - being familiar with relevant lecture-material,
  - trying to make meaningful contributions.
- Cannot actively participate in some session? Reach out as soon as possible!
- If you got a passing grade "befriedigend" you may take an optional oral exam at the end of the lecture period. The grade attained for the oral exam then replaces the previous grade.

Me

- Jan Dreier (just call me Jan)
- studied CS and did my PhD at RWTH Aachen University
- joined TU Wien as PostDoc beginning of the year.
- I like working on algorithmic meta-theorems, parameterized algorithms, structural graph theory, randomness, …
- This is the first time this lecture is held and my first lecture in general. Things may not always go smooth.

You

- What's your background?
- What are you interested in?
- What do you expect?

Next Dates

- 19.10. Lecture
- 26.10. ~~Lecture~~ (National Holiday)
- 29.10. Exercise
- …

Crash Course

We start with some simple examples that illustrate a bit of the "philosophy" behind algorithmic meta-theorems.

Graphs and Structure

- This lecture only deals only with *graph problems*.
  - Many problems can be described using graphs (use edges to describe conflicts, friendships, dependencies, ...).
  - Many ways to restrict their structure (trees, planarity, bounded degree, ...).

Graphs and Structure

- This lecture only deals only with *graph problems*.
  - Many problems can be described using graphs (use edges to describe conflicts, friendships, dependencies, …).
  - Many ways to restrict their structure (trees, planarity, bounded degree, …).

- Many problems are are tractable if you assume some extra structure.
  - independent set on trees
  - coloring on interval graphs
  - …

Using Graph Structure to Solve Independent Set

How does graph structure help us solve problems? We will explore this using the independent problem as an example.

An *independent set* in a graph G is a vertex set $S \subseteq V(G)$ such that no vertices in S are adjacent.

INDEPENDENTSET

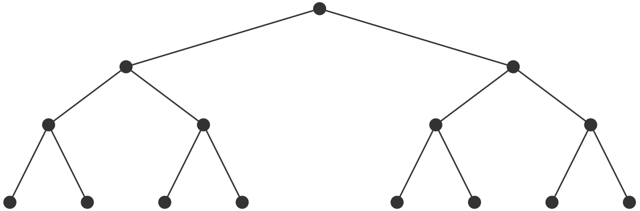
Input:	Graph G and integer k
Question:	Does G have an independent set of size k?

INDEPENDENTSET is NP-complete. This means every problem in NP can be reduced to this problem. Unless P$=$NP, INDEPENDENTSET cannot be solved in polynomial time.

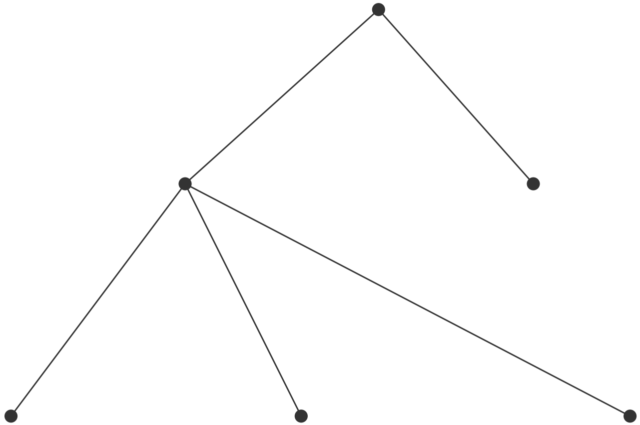
Independent Set on Trees

INDEPENDENTSET can be solved in linear time on trees.

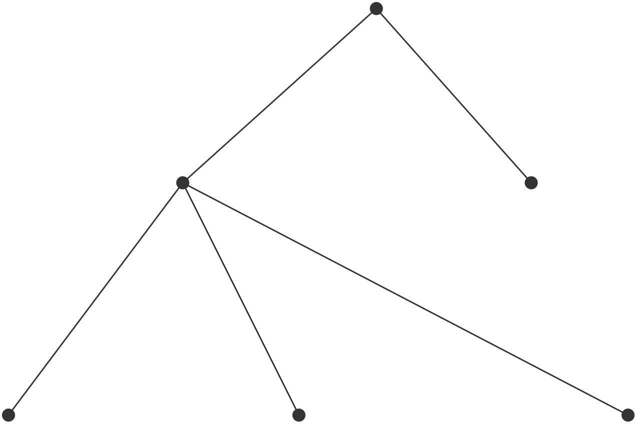
Idea: Root the tree and do dynamic programming. Starting at the leafs, compute for each subtree the maximum size of a solution with and without its root.



Independent Set on Trees



Independent Set on Trees

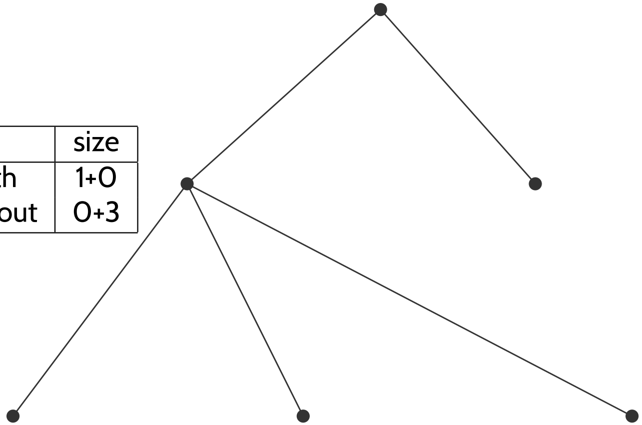


	size
with	1
without	0

	size
with	1
without	0

	size
with	1
without	0

Independent Set on Trees



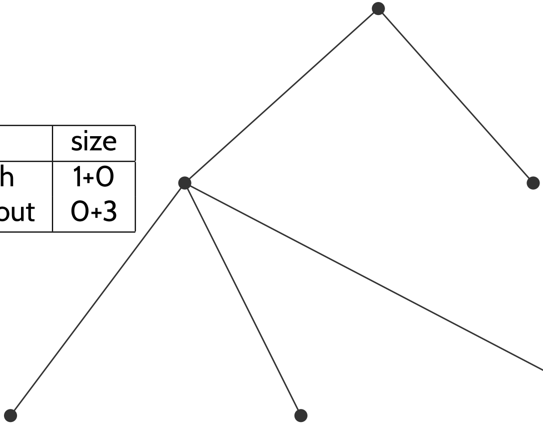
	size
with	1+0
without	0+3

	size
with	1
without	0

	size
with	1
without	0

	size
with	1
without	0

Independent Set on Trees



	size
with	1+0
without	0+3

	size
with	1
without	0

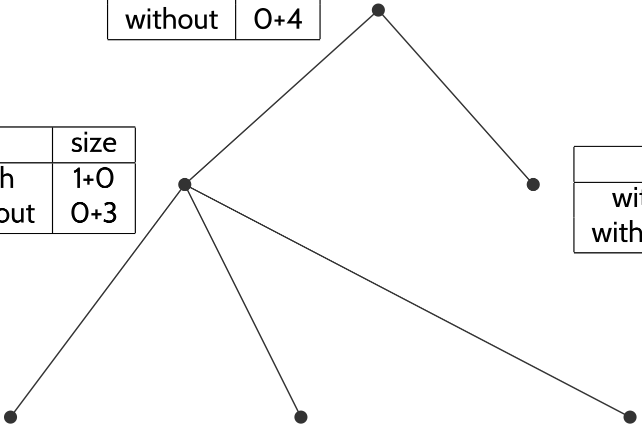
	size
with	1
without	0

	size
with	1
without	0

	size
with	1
without	0

Independent Set on Trees

	size
with	1+3
without	0+4



	size
with	1+0
without	0+3

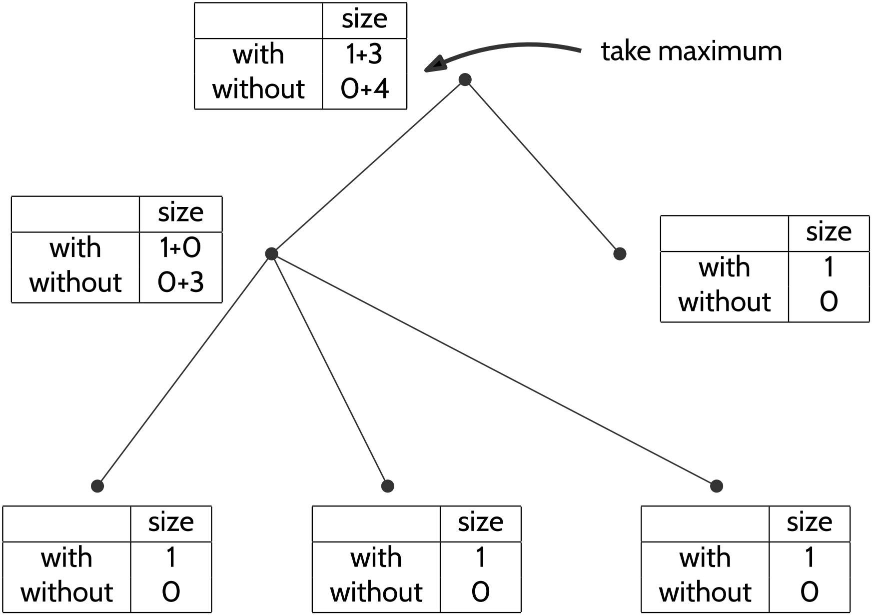
	size
with	1
without	0

	size
with	1
without	0

	size
with	1
without	0

	size
with	1
without	0

Independent Set on Trees



	size
with	1+3
without	0+4

	size
with	1+0
without	0+3

	size
with	1
without	0

	size
with	1
without	0

	size
with	1
without	0

	size
with	1
without	0

Generalization

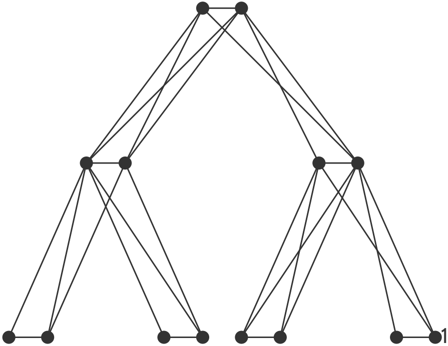
- This approach can be extended to tree-like graphs (bounded treewidth).

Generalization

- This approach can be extended to tree-like graphs (bounded treewidth).
- First main result of the lecture (Courcelle's theorem): Every problem definable in monadic second-order logic can be solved in linear time on graphs of bounded treewidth.

Generalization

- This approach can be extended to tree-like graphs (bounded treewidth).
- First main result of the lecture (Courcelle's theorem): Every problem definable in monadic second-order logic can be solved in linear time on graphs of bounded treewidth.
- This includes
  - coloring
  - independent set
  - clique
  - dominating set
  - feedback vertex set
  - hamilton path
  - ...



Second-Order Logic

Second-order logic on graphs has . . .

- variables for vertices (x, y, z) and sets of vertices (X, Y, Z)
- quantifiers $\exists$ and $\forall$, as well as operators $\wedge$, $\vee$ and $\neg$
- relations $=$, $x \sim y$ (adjacency) and $x \in X$ (membership)

Second-Order Logic

Second-order logic on graphs has ...

- variables for vertices (x, y, z) and sets of vertices (X, Y, Z)
- quantifiers $\exists$ and $\forall$, as well as operators $\wedge$, $\vee$ and $\neg$
- relations $=$, $x \sim y$ (adjacency) and $x \in X$ (membership)

A problem is *expressible* if we can write down a sentence that is true if and only if the instance is a yes-instance.

Second-Order Logic

Second-order logic on graphs has . . .

- variables for vertices (x, y, z) and sets of vertices (X, Y, Z)
- quantifiers $\exists$ and $\forall$, as well as operators $\wedge$, $\vee$ and $\neg$
- relations $=$, $x \sim y$ (adjacency) and $x \in X$ (membership)

A problem is *expressible* if we can write down a sentence that is true if and only if the instance is a yes-instance.

Instead of finding an algorithm, we only have to find a formula that expresses the problem!

Expressiveness

What does this formula express?

$$\exists x \exists y \exists z (x \sim y \land x \sim z \land y \sim z)$$

Expressiveness

What does this formula express?

$$\exists x \exists y \exists z (x \sim y \wedge x \sim z \wedge y \sim z)$$

Or this one?

$$\exists x \forall y (x = y \vee x \sim y)$$

Expressiveness

What does this formula express?

$$\exists x \exists y \exists z (x \sim y \wedge x \sim z \wedge y \sim z)$$

Or this one?

$$\exists x \forall y (x = y \vee x \sim y)$$

Or this?

$$\varphi(X) = \neg \exists x \exists y (x \in X \wedge y \in X \wedge x \sim y)$$

Expressiveness

What does this formula express?

$$\exists x \exists y \exists z (x \sim y \wedge x \sim z \wedge y \sim z)$$

Or this one?

$$\exists x \forall y (x = y \vee x \sim y)$$

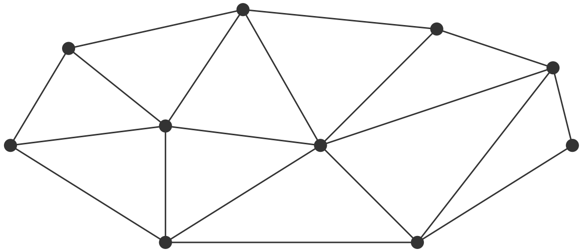
Or this?

$$\varphi(X) = \neg \exists x \exists y (x \in X \wedge y \in X \wedge x \sim y)$$

Later on, we will prove Courcelle's theorem and identify which problems can be expressed in monadic second-order logic.

Independent Set on Planar Graphs

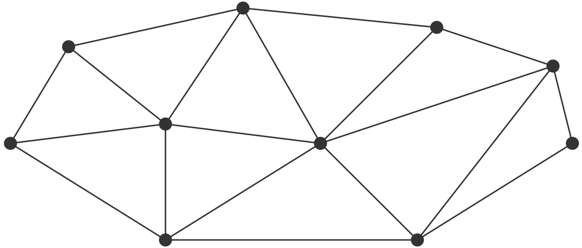
How about planar graphs?



Independent Set on Planar Graphs

How about planar graphs?

INDEPENDENTSET is NP-complete on planar graphs.



Easy and Hard Instances

- The distinction between P and NP may be too coarse.
- Even though a problem is NP-complete, certain instances may be tractable.
- How can we describe these instances?

Parameterized Complexity

Assign each instance a number, called the *parameter*. We hope that

- we can solve the instance if the parameter is small,
- interesting instances have a small parameter.

NP-hard problems may still be tractable for small parameter values!

Parameterized Independent Set

PARAMETERIZED INDEPENDENTSET

Input:	Graph G and integer k
Parameter:	k
Question:	Does G have an independent set of size k?

A parameterized problem is *fixed parameter tractable* (fpt) if instances with parameter k and size n can be solved in time $f(k)n^c$ (for some function f and constant c).

Is PARAMETERIZED INDEPENDENTSET fixed parameter tractable?

Parameterized Hardness

PARAMETERIZED INDEPENDENTSET is not fixed parameter tractable (unless FPT $=$ W[1]).

Parameterized Hardness

PARAMETERIZED INDEPENDENTSET is not fixed parameter tractable (unless FPT $=$ W[1]).

The best known algorithm takes time $n^{\Theta(k)}$.

Parameterized Hardness

PARAMETERIZED INDEPENDENTSET is not fixed parameter tractable (unless FPT $=$ W[1]).

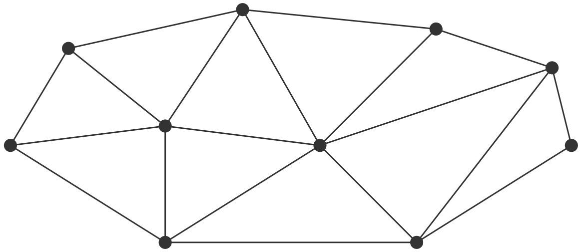
The best known algorithm takes time $n^{\Theta(k)}$.

for $v_1 \in V$
  for $v_2 \in V$
    ...
      for $v_k \in V$
        check if $v_1, \ldots, v_k$ is an IS of size k

This is optimal (under certain complexity assumptions).

Independent Set on Planar Graphs

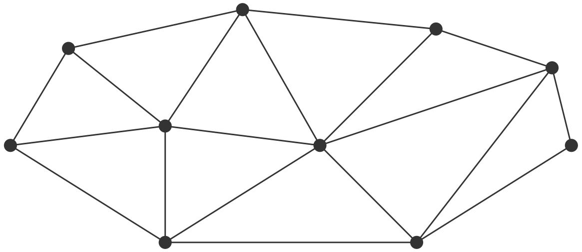
How about parameterized independent set on planar graphs?



Independent Set on Planar Graphs

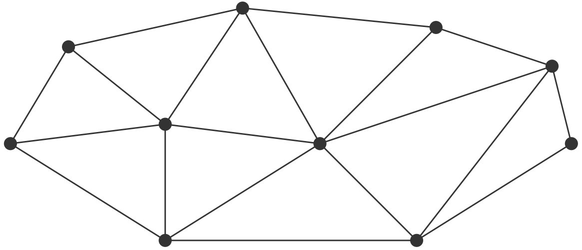
How about parameterized independent set on planar graphs?

PARAMETERIZED INDEPENDENTSET is fixed parameter tractable on planar graphs.



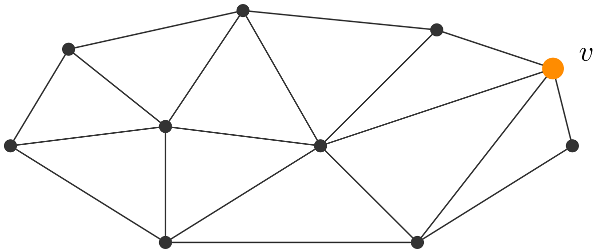
Independent Set on Planar Graphs

- We want to find an independent set of size k.



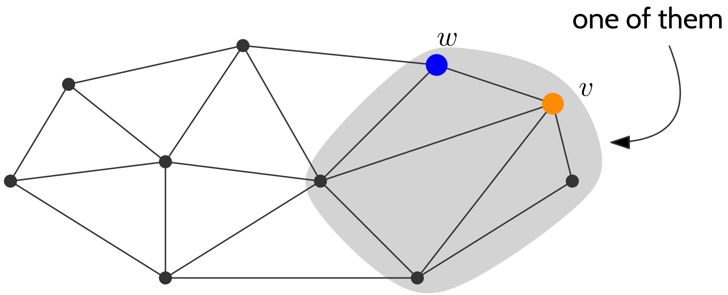
Independent Set on Planar Graphs

- We want to find an independent set of size k.
- In planar graphs there is always a vertex v with degree ≤ 5.



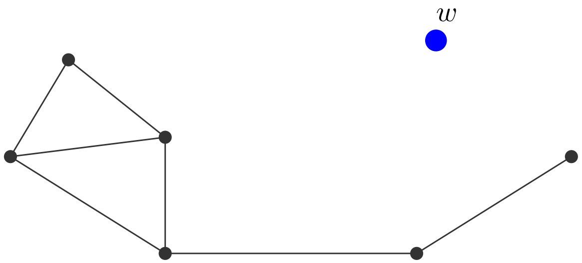
Independent Set on Planar Graphs

- We want to find an independent set of size k.
- In planar graphs there is always a vertex v with degree ≤ 5.
- At least one vertex w from $N(v)$ is in a maximal independent set.



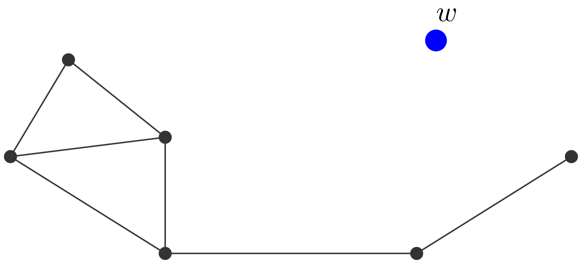
Independent Set on Planar Graphs

- We want to find an independent set of size k.
- In planar graphs there is always a vertex v with degree ≤ 5.
- At least one vertex w from $N(v)$ is in a maximal independent set.
- We guess w, place w in solution and remove $N(w)$.



Independent Set on Planar Graphs

- We want to find an independent set of size k.
- In planar graphs there is always a vertex v with degree ≤ 5.
- At least one vertex w from $N(v)$ is in a maximal independent set.
- We guess w, place w in solution and remove $N(w)$.
- Then find a solution of size $k-1$ in remaining graph.



Algorithm

```
IS(G, k):
   if G is empty return k == 0
   find vertex v with degree ≤ 5 in G
   for all w ∈ N(v):
      if IS(G \ N(w), k − 1) return True
   return False
```

This solves PARAMETERIZED INDEPENDENTSET on planar graphs in time $O(6^k n)$.

Sparse Graphs

- Trees have at most $(n - 1)$ edges and planar graphs at most $3n$ edges.

Sparse Graphs

- Trees have at most $(n-1)$ edges and planar graphs at most $3n$ edges.
- Do problems generally become simpler if there are few edges in the graph?

Sparse Graphs

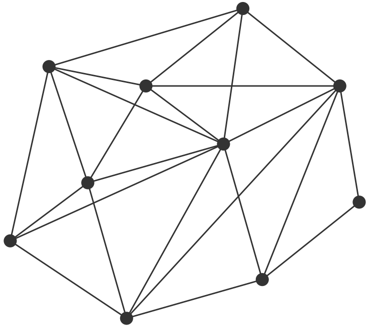
- Trees have at most $(n-1)$ edges and planar graphs at most $3n$ edges.
- Do problems generally become simpler if there are few edges in the graph?
- Yes and no.
- We discuss *Sparsity theory* which gives a rigorous answer to this question.

Sparse Graphs

- Trees have at most $(n-1)$ edges and planar graphs at most $3n$ edges.
- Do problems generally become simpler if there are few edges in the graph?
- Yes and no.
- We discuss *Sparsity theory* which gives a rigorous answer to this question.
- *Nowhere dense graph classes* are the top of the sparsity hierarchy.

Generalization

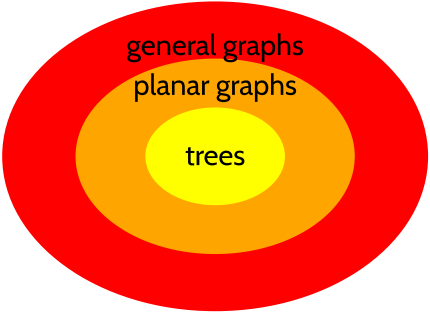
- Second main result of the lecture: Every problem definable in first-order logic can be solved in fpt time on nowhere dense graphs.
- This includes
  - param. independent set
  - param. clique
  - param. dominating set
  - param. scattered set
  - …



Summary

INDEPENDENTSET is hard on general graphs. However,

- on trees, we can solve it in linear time
- on planar graphs, it is still fixed parameter tractable.

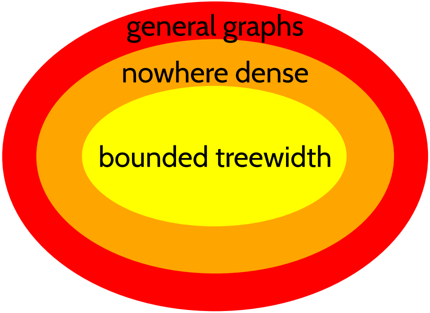


We will observe a similar behaviour for many other problems!

Summary

INDEPENDENTSET is hard on general graphs. However,

- on bounded treewidth, we can solve it in linear time
- on nowhere dense graphs, it is still fixed parameter tractable.



We will observe a similar behaviour for many other problems!