Parameterized complexity of constraint satisfaction problems

Dániel Marx

Budapest University of Technology and Economics

`dmarx@cs.bme.hu`

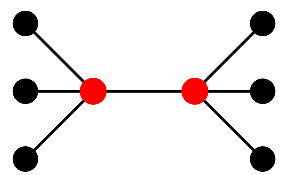
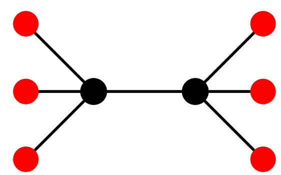
Presented at University of Freiburg, Germany

June 7, 2004

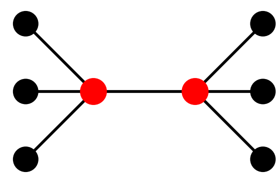
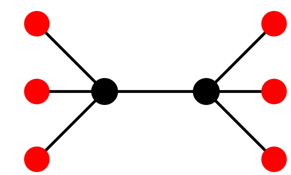
Outline of the talk

- Parameterized complexity
- Schaefer's Dichotomy Theorem
- A parameterized dichotomy theorem
- Sketch of proof
- Planar formulae

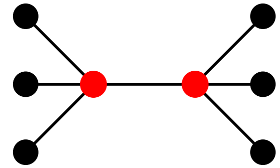
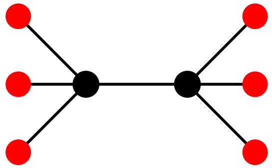
Parameterized complexity

Problem:	MINIMUM VERTEX COVER	MAXIMUM INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Is it possible to cover the edges with k vertices?	Is it possible to find k independent vertices?
		
Complexity:	**NP**-complete	**NP**-complete

Parameterized complexity

Problem:	MINIMUM VERTEX COVER	MAXIMUM INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Is it possible to cover the edges with k vertices?	Is it possible to find k independent vertices?
		
Complexity:	**NP**-complete	**NP**-complete
Complete enumeration:	$O(n^k)$ possibilities	$O(n^k)$ possibilities

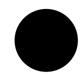
Parameterized complexity

Problem:	MINIMUM VERTEX COVER	MAXIMUM INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Is it possible to cover the edges with k vertices? 	Is it possible to find k independent vertices? 
Complexity:	**NP**-complete	**NP**-complete
Complete enumeration:	$O(n^k)$ possibilities	$O(n^k)$ possibilities
	$O(2^k n^2)$ algorithm exists 	No $n^{o(k)}$ algorithm known 

Bounded search tree method

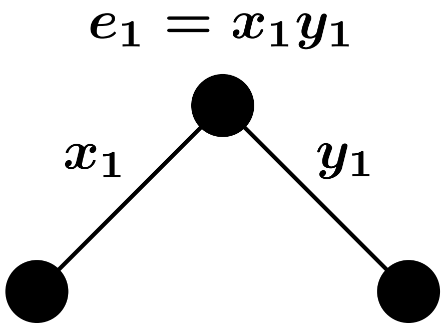
Algorithm for MINIMUM VERTEX COVER:

$$e_1 = x_1 y_1$$



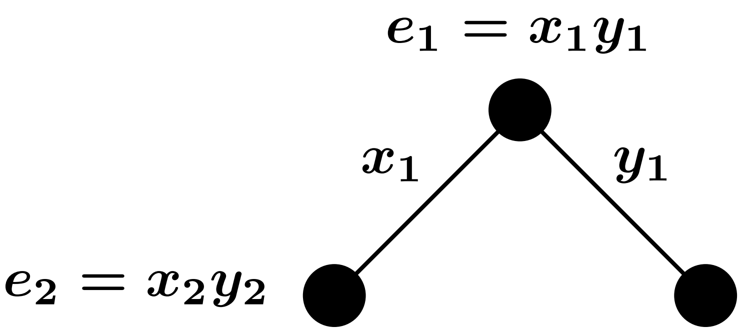
Bounded search tree method

Algorithm for MINIMUM VERTEX COVER:



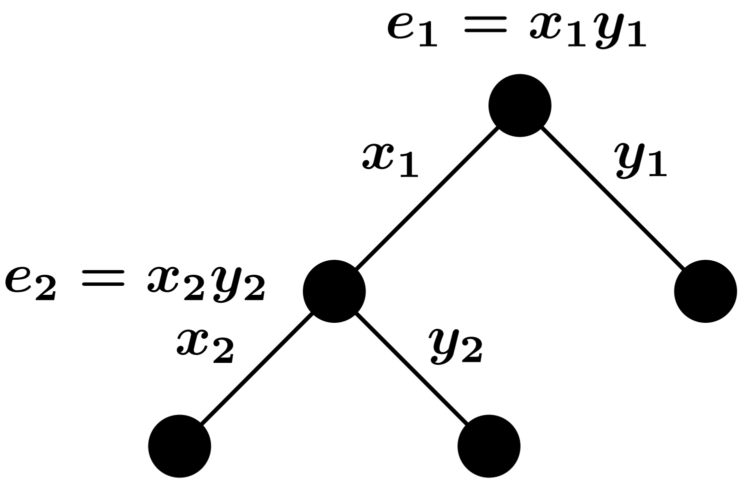
Bounded search tree method

Algorithm for MINIMUM VERTEX COVER:



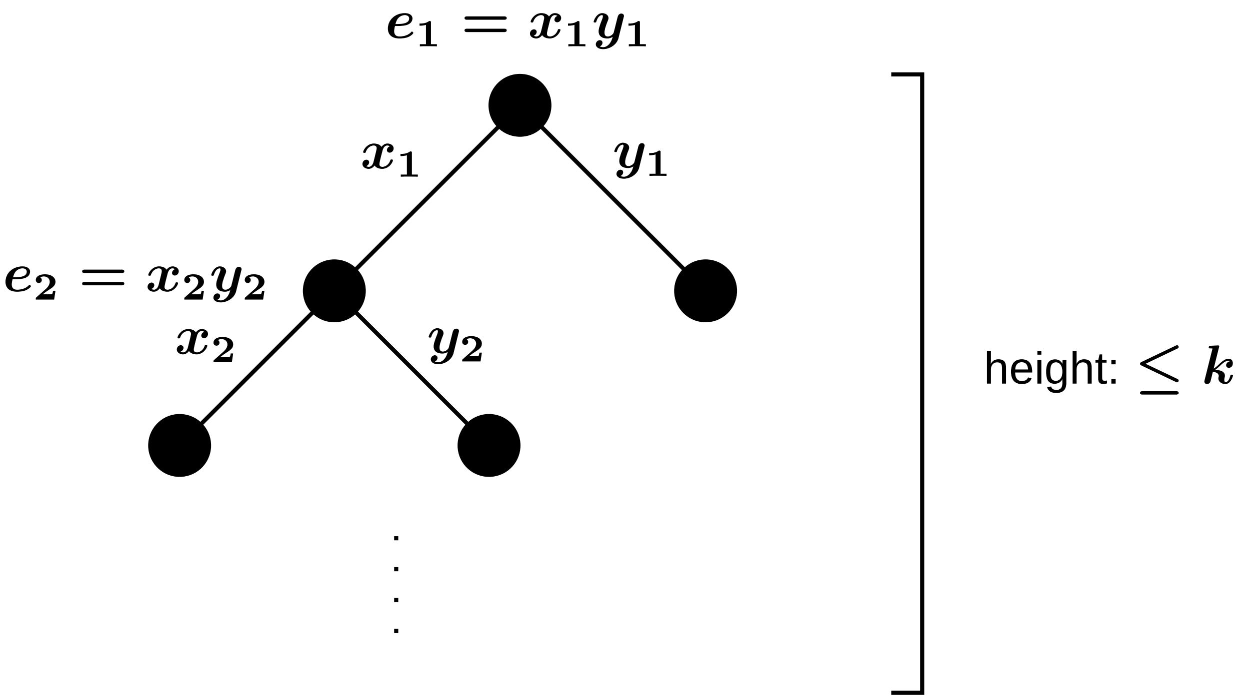
Bounded search tree method

Algorithm for MINIMUM VERTEX COVER:



Bounded search tree method

Algorithm for MINIMUM VERTEX COVER:



Height of the search tree is $\leq k \Rightarrow$ number of nodes is $O(2^k) \Rightarrow$ complete search requires $2^k \cdot$ poly steps.

Fixed-parameter tractability

Definition: a **parameterized problem** is **fixed-parameter tractable (FPT)** if there is an $f(k)n^c$ time algorithm for some constant c.

We have seen that MINIMUM VERTEX COVER is in FPT. Best known algorithm: $O(1.2832^k k + k|V|)$ [Niedermeier, Rossmanith, 2003]

Main goal of parameterized complexity: to find fixed-parameter tractable problems.

Fixed-parameter tractability

Definition: a **parameterized problem** is **fixed-parameter tractable (FPT)** if there is an $f(k)n^c$ time algorithm for some constant c.

We have seen that MINIMUM VERTEX COVER is in FPT. Best known algorithm: $O(1.2832^k k + k|V|)$ [Niedermeier, Rossmanith, 2003]

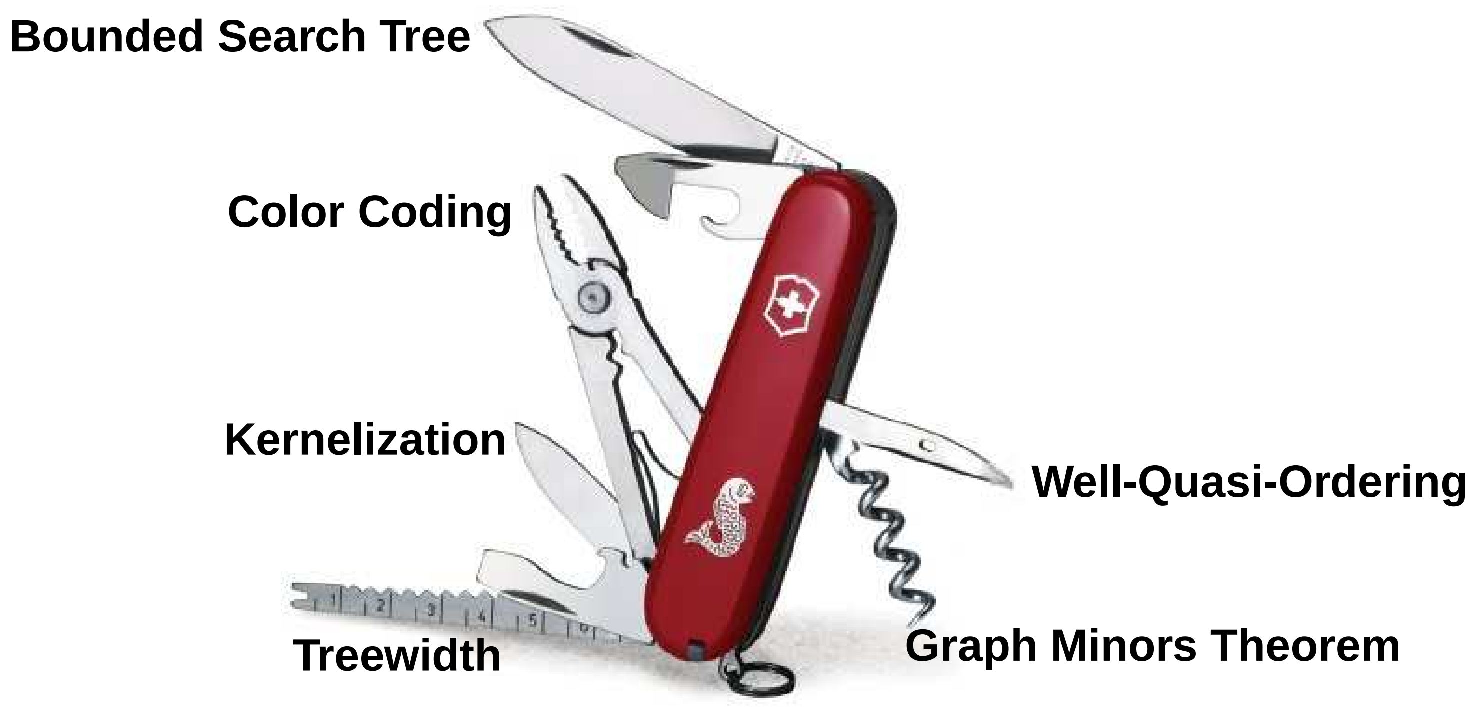
Main goal of parameterized complexity: to find fixed-parameter tractable problems.

Examples of **NP**-hard problems that are in FPT:

- LONGEST PATH
- DISJOINT TRIANGLES
- FEEDBACK VERTEX SET
- GRAPH GENUS
- etc.

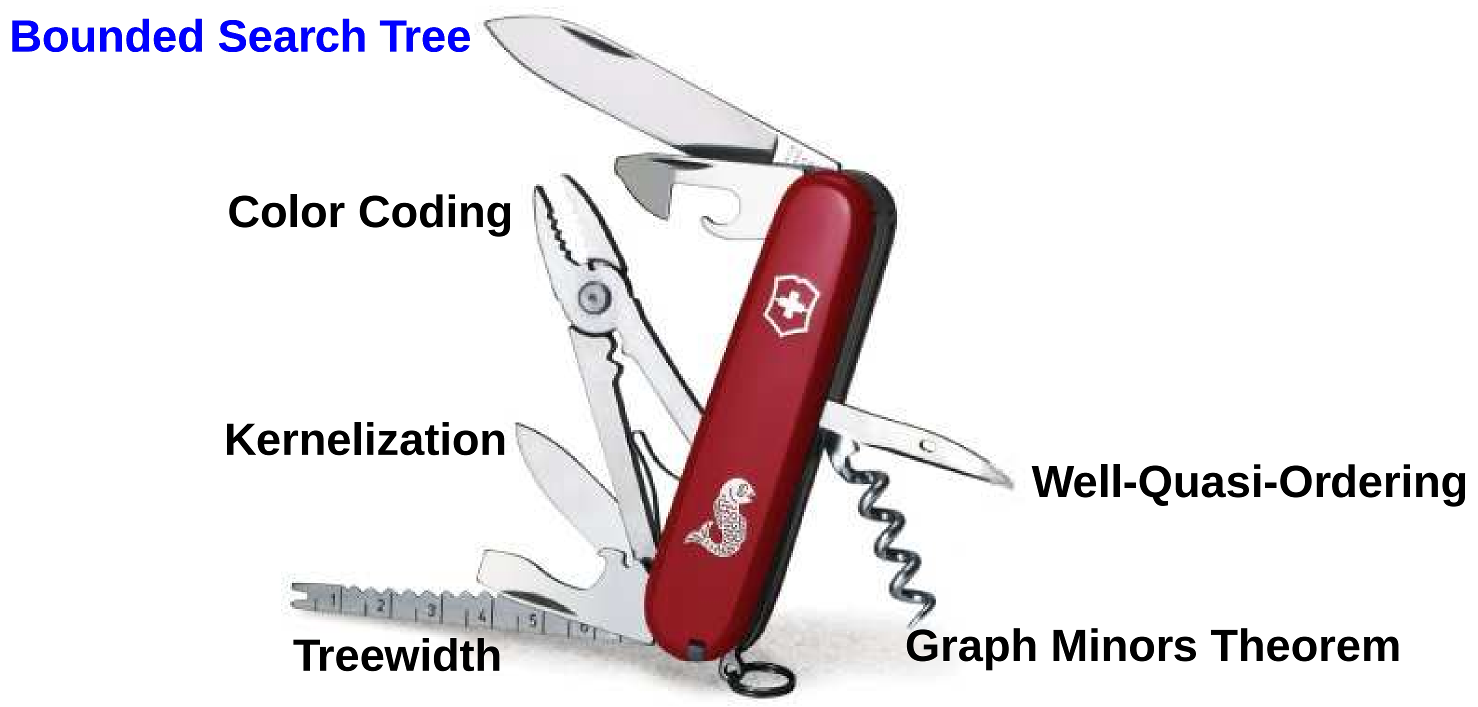
Fixed-parameter tractability (cont.)

- Practical importance: efficient algorithms for small values of k.
- Powerful toolbox for designing FPT algorithms:



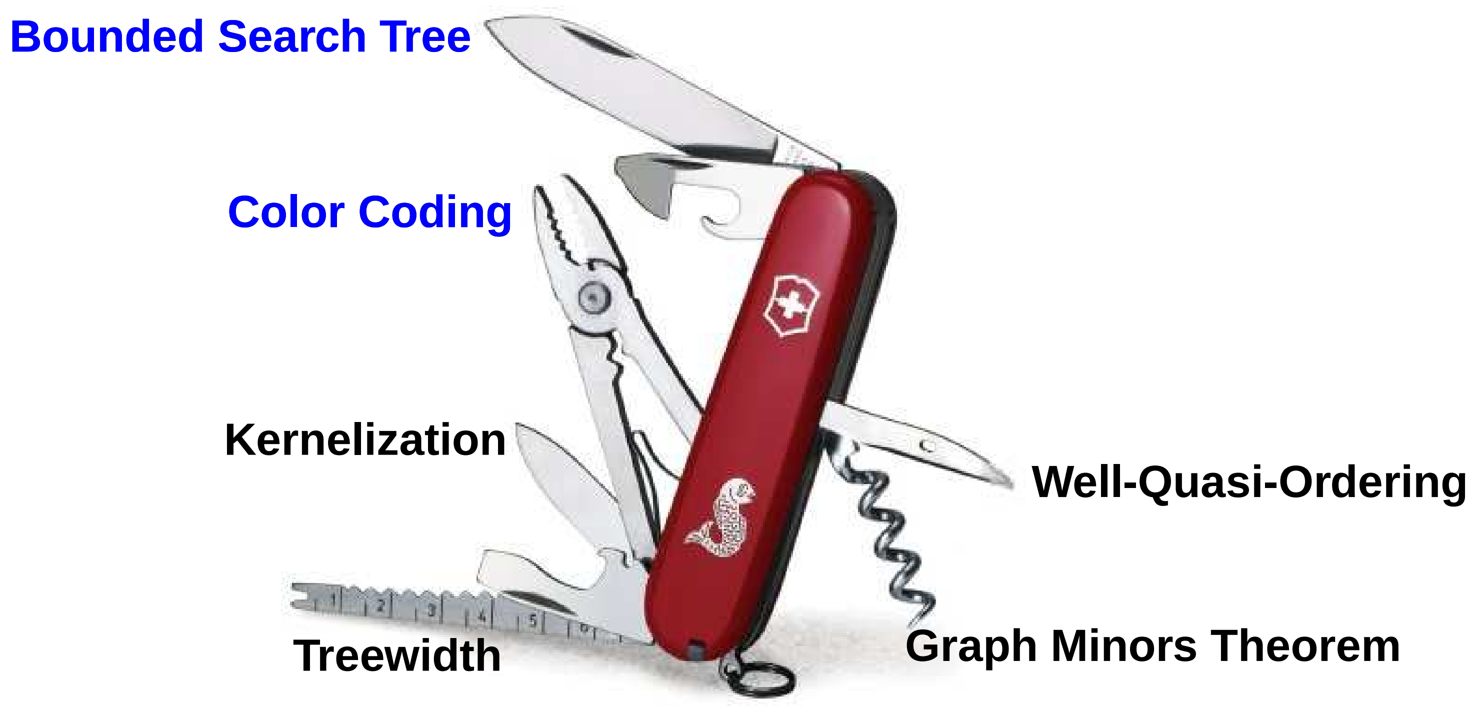
Fixed-parameter tractability (cont.)

- Practical importance: efficient algorithms for small values of k.
- Powerful toolbox for designing FPT algorithms:



Fixed-parameter tractability (cont.)

- Practical importance: efficient algorithms for small values of k.
- Powerful toolbox for designing FPT algorithms:

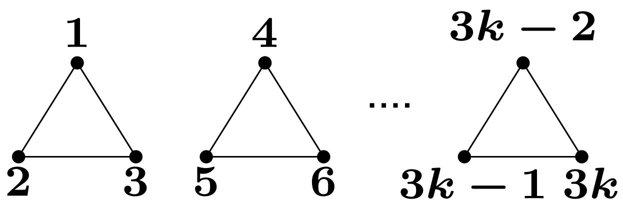


Color Coding: Disjoint Triangles

Task: Find k vertex disjoint triangles in a graph G.

Method:

- Assign random labels $\mathbf{1}, \mathbf{2}, \ldots, \mathbf{3k}$ to the vertices.
- Are there k triangles such that



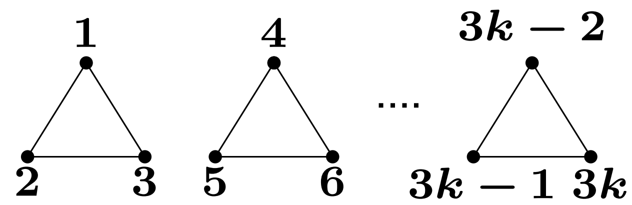
?

The existence of such triangles is easy to check.

Color Coding: Disjoint Triangles

Task: Find k vertex disjoint triangles in a graph G.

Method:

- Assign random labels $1, 2, \ldots, 3k$ to the vertices.
- Are there k triangles such that 

?

  The existence of such triangles is easy to check.

If there are k disjoint triangles

$\Rightarrow$ with probability $1/(3k)^{3k}$ they are labeled as on the figure

$\Rightarrow$ we need on average $(3k)^{3k}$ random assignments to find the k triangles!

Color coding: useful if we want to select a **small** number of disjoint **small** objects from a **large** list.

Method can be derandomized using families of k-perfect hash functions.

Parameterized intractability

We expect that MAXIMUM INDEPENDENT SET is not fixed-parameter tractable, no $n^{o(k)}$ algorithm is known.

W[1]-complete $\approx$ "as hard as MAXIMUM INDEPENDENT SET"

Parameterized intractability

We expect that MAXIMUM INDEPENDENT SET is not fixed-parameter tractable, no $n^{o(k)}$ algorithm is known.

W[1]-complete $\approx$ “as hard as MAXIMUM INDEPENDENT SET”

Parameterized reductions: L_1 is reducible to L_2, if there is a function f that transforms (x, k) to (x', k') such that

- $(x, k) \in L_1$ if and only if $(x', k') \in L_2$,
- f can be computed in $f(k)|x|^c$ time,
- **k' depends only on k**

If L_1 is reducible to L_2, and L_2 is in FPT, then L_1 is in FPT as well.

Most **NP**-completeness proofs are not good for parameterized reductions.

Parameterized Complexity: Summary

Two key concepts:

- A parameterized problem is **fixed-parameter tractable** if it has an $f(k)n^c$ time algorithm.
- To show that a problem L is hard, we have to give a **parameterized reduction** from a known **W[1]-complete** problem to L.

Constraint satisfaction problems

Let $\mathcal{R}$ be a set Boolean of relations. An $\mathcal{R}$-formula is a conjunction of relations in $\mathcal{R}$:

$$R_1(x_1, x_4, x_5) \wedge R_2(x_2, x_1) \wedge R_1(x_3, x_3, x_3) \wedge R_3(x_5, x_1, x_4, x_1)$$

$\mathcal{R}$-SAT

- Given: an $\mathcal{R}$-formula φ
- Find: a variable assignment satisfying φ

Constraint satisfaction problems

Let $\mathcal{R}$ be a set Boolean of relations. An $\mathcal{R}$-formula is a conjunction of relations in $\mathcal{R}$:

$$R_1(x_1, x_4, x_5) \wedge R_2(x_2, x_1) \wedge R_1(x_3, x_3, x_3) \wedge R_3(x_5, x_1, x_4, x_1)$$

$\mathcal{R}$-SAT

- Given: an $\mathcal{R}$-formula φ
- Find: a variable assignment satisfying φ

$\mathcal{R} = \{a \neq b\} \Rightarrow \mathcal{R}$-SAT = 2-coloring of a graph
$\mathcal{R} = \{a \vee b,\ a \vee \bar{b},\ \bar{a} \vee \bar{b}\} \Rightarrow \mathcal{R}$-SAT = 2SAT
$\mathcal{R} = \{a \vee b \vee c, a \vee b \vee \bar{c}, a \vee \bar{b} \vee \bar{c}, \bar{a} \vee \bar{b} \vee \bar{c}\} \Rightarrow \mathcal{R}$-SAT = 3SAT

Question: $\mathcal{R}$-SAT is polynomial time solvable for which $\mathcal{R}$?

It is **NP**-complete for which $\mathcal{R}$?

Schaefer's Dichotomy Theorem (1978)

For every $\mathcal{R}$, the $\mathcal{R}$-SAT problem is polynomial time solvable if one of the following holds, and **NP**-complete otherwise:

- Every relation is satisfied by the all 0 assignment
- Every relation is satisfied by the all 1 assignment
- Every relation can be expressed by a 2SAT formula
- Every relation can be expressed by a Horn formula
- Every relation can be expressed by an anti-Horn formula
- Every relation is an affine subspace over $GF(2)$

Schaefer's Dichotomy Theorem (1978)

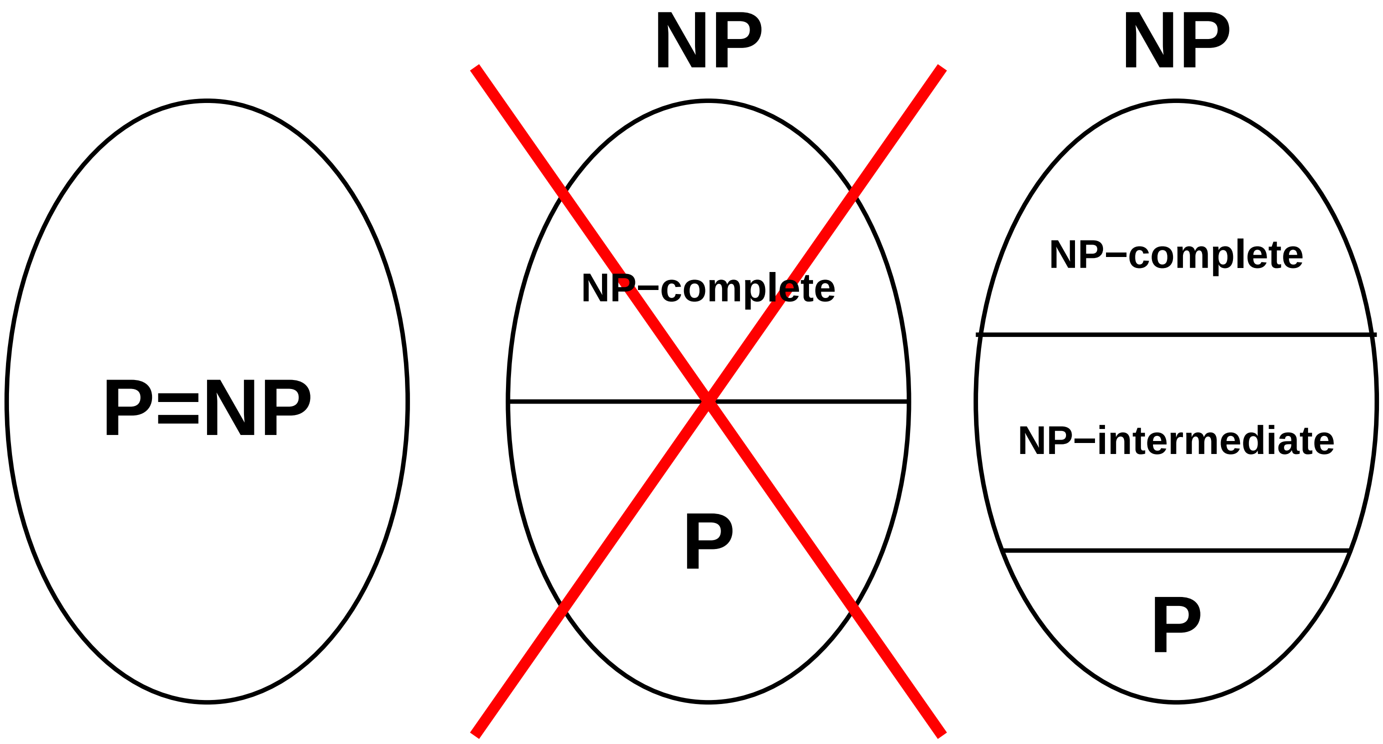
For every $\mathcal{R}$, the $\mathcal{R}$-SAT problem is polynomial time solvable if one of the following holds, and **NP**-complete otherwise:

- Every relation is satisfied by the all 0 assignment
- Every relation is satisfied by the all 1 assignment
- Every relation can be expressed by a 2SAT formula
- Every relation can be expressed by a Horn formula
- Every relation can be expressed by an anti-Horn formula
- Every relation is an affine subspace over $GF(2)$

Why is it surprising?

Ladner's Theorem (1975)

If **P** $\neq$ **NP**, then there is a language $L \in$ **NP** $\setminus$ **P** that is not **NP**-complete.



Other dichotomy results

- Approximability of MAX-SAT, MIN-UNSAT [Khanna et al., 2001]
- Approximability of MAX-ONES, MIN-ONES [Khanna et al., 2001]
- Generalization to 3 valued variables [Bulatov, 2002]
- Inverse satisfiability [Kavvadias and Sideri, 1999]
- etc.

Other dichotomy results

- Approximability of MAX-SAT, MIN-UNSAT [Khanna et al., 2001]
- Approximability of MAX-ONES, MIN-ONES [Khanna et al., 2001]
- Generalization to 3 valued variables [Bulatov, 2002]
- Inverse satisfiability [Kavvadias and Sideri, 1999]
- etc.

Our contribution: parameterized analogue of Schaefer's dichotomy theorem.

Parameterized version

Parameterized $\mathcal{R}$-SAT

- **Input:** an $\mathcal{R}$-formula φ, an integer k
- **Parameter:** k
- **Question:** Does φ have a satisfying assignment of weight exactly k?

For which $\mathcal{R}$ is there an $f(k) \cdot n^c$ algorithm for $\mathcal{R}$-SAT?

Main theorem: For every constraint family $\mathcal{R}$, the parameterized $\mathcal{R}$-SAT problem is either fixed-parameter tractable or W[1]-complete.
(+ simple characterization of FPT cases)

Technical notes

- Are constants allowed in the formula?
  E.g., $R(x_1, 0, 1) \wedge R(1, x_2, x_3)$
- Can a variable appear multiple times in a constraint?
  E.g., $R(x_1, x_1, x_2) \wedge R(x_3, x_3, x_3)$
- Constraints that are not satisfied by the all 0 assignment can be handled easily (bounded search tree).

Weak separability

Definition: $\boldsymbol{R}$ is weakly separable if

1. the union of two disjoint satisfying assignments is also satisfying, and
2. if a satisfying assignment contains a smaller satisfying assignment, then their difference is also satisfying.

Example of 1:

$$\boldsymbol{R(1,1,1,1,0,0,0,0,0) = 1}$$
$$\boldsymbol{R(0,0,0,0,1,1,0,0,0) = 1}$$
$$\Downarrow$$
$$\boldsymbol{R(1,1,1,1,1,1,0,0,0) = 1}$$

Example of 2:

$$\boldsymbol{R(1,1,1,1,1,1,0,0) = 1}$$
$$\boldsymbol{R(0,0,1,1,1,1,0,0) = 1}$$
$$\Downarrow$$
$$\boldsymbol{R(1,1,0,0,0,0,0,0) = 1}$$

Main theorem: $\mathcal{R}$-SAT is FPT if and only if every constraint is weakly separable, and W[1]-complete otherwise.

Weak separability: examples

The constraint EVEN is weakly separable:

Property 1:

$$R(\overbrace{1,1,1,1}^{\text{even}},0,0,0,0,0)=1$$

$$R(0,0,0,0,\underbrace{1,1}_{\text{even}},0,0,0)=1$$

$$\Downarrow$$

$$R(\underbrace{1,1,1,1,1,1}_{\text{even}},0,0,0)=1$$

Property 2:

$$R(\overbrace{1,1,1,1,1,1}^{\text{even}},0,0)=1$$

$$R(0,0,\underbrace{1,1,1,1}_{\text{even}},0,0)=1$$

$$\Downarrow$$

$$R(\underbrace{1,1}_{\text{even}},0,0,0,0,0,0)=1$$

More generally: every **affine** constraint is weakly separable.

Weak separability: examples (cont.)

The following constraint is trivially weakly separable:

$R(0,0,0,0,0) = 1$
$R(1,1,1,0,0) = 1$
$R(0,1,1,1,0) = 1$
$R(0,0,1,1,1) = 1$
$R(x_1,x_2,x_3,x_4,x_5) = 0$ otherwise.

Reason: Property 1 and 2 vacuously hold, no disjoint sets, no subsets.

More generally: if the non-zero satisfying assignments are **intersecting** and form a **clutter**, then it is weakly separable.

Example: $R(x_1,\ldots,x_n) = 1$ if and only if 0 or exactly t out of n variables are 1 ($t > n/2$)

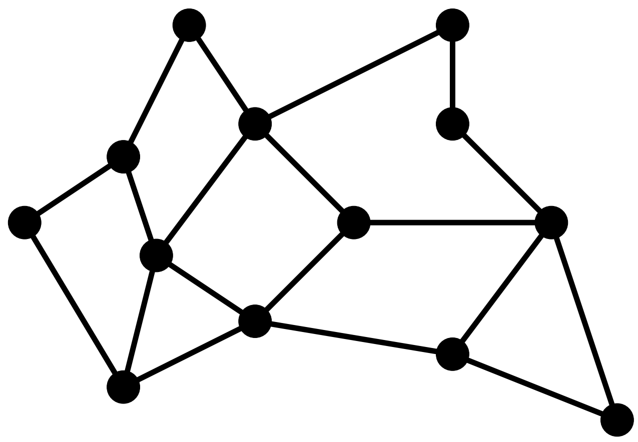
Parameterized vs. classical

The easy and hard cases are different in the classical and the parameterized version:

Constraint	**Classical**	**Parameterized**
$x \vee y$	in P	FPT (VERTEX COVER)
$\bar{x} \vee \bar{y}$	in P	W[1]-complete (MAXIMUM INDEPENDENT SET)
affine	in P	FPT
2-in-3	NP-complete	FPT

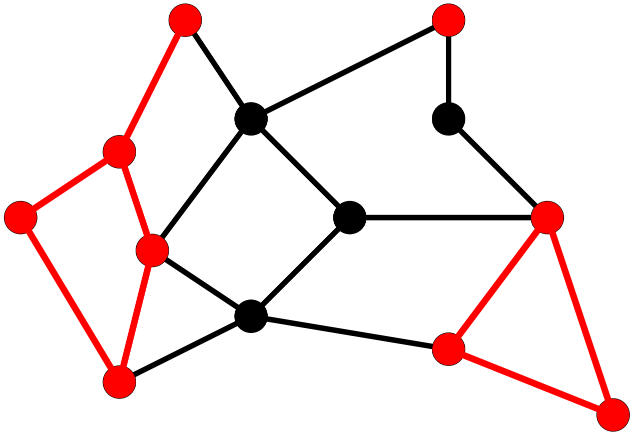
Bounded number of occurrences

Primal graph: Vertices are the variables, two variables are connected if they appear in some clause together.



Bounded number of occurrences

Primal graph: Vertices are the variables, two variables are connected if they appear in some clause together.



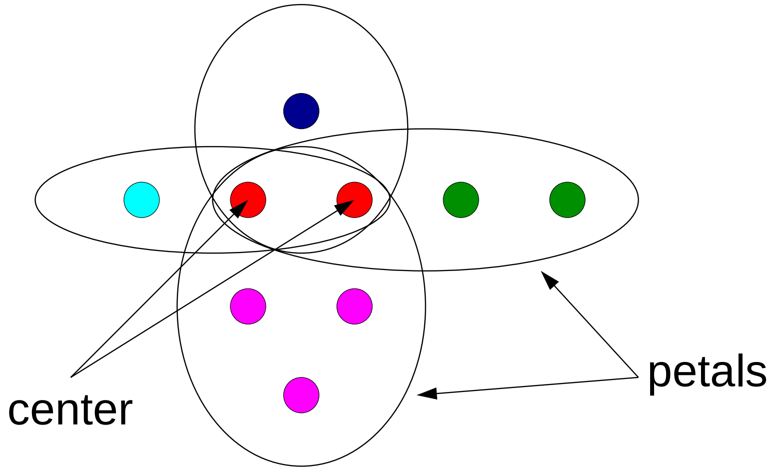
Every satisfying assignment is composed of **connected satisfying assignments**.

Lemma: There are at most $(rd)^{k^2} \cdot n$ connected satisfying assignments of size at most k. (r is the maximum arity, d is the maximum no. of occurrences)

Algorithm: Use color coding to put together the connected assignments to obtain a size k assignment.

The sunflower lemma

Definition: Sets S_1, S_2, ..., S_k form a **sunflower** if the sets $S_i \setminus (S_1 \cap S_2 \cap \cdots \cap S_k)$ are disjoint.



Lemma (Erdős and Rado, 1960): If the size of a set system is greater than $(p-1)^\ell \cdot \ell!$ and it contains only sets of size at most ℓ, then the system contains a sunflower with p petals.

Sunflower of clauses

Definition: A **sunflower** is a set of k clauses such that for every i

- either the same variable appears at position i in every clause,
- or every clause “owns” its ith variable.

$$R(x_1, x_2, x_3, x_4, x_5, x_6)$$
$$R(x_1, x_2, x_3, x_7, x_8, x_9)$$
$$R(x_1, x_2, x_3, x_{10}, x_{11}, x_{12})$$
$$R(x_1, x_2, x_3, x_{13}, x_{14}, x_{15})$$

Lemma: If a variable occurs more than $c_{\mathcal{R}}(k)$ times in an $\mathcal{R}$-formula, then the formula contains a sunflower of clauses with more than k petals.

Plucking the sunflower

For weakly separable constraints, the formula can be reduced if there is a sunflower with $k+1$ petals. Example:

$$k+1 \begin{cases} \text{EVEN}(x_1, x_2, x_3, x_4, x_5, x_6) \\ \text{EVEN}(x_1, x_2, x_3, x_7, x_8, x_9) \\ \text{EVEN}(x_1, x_2, x_3, x_{10}, x_{11}, x_{12}) \\ \text{EVEN}(x_1, x_2, x_3, x_{13}, x_{14}, x_{15}) \end{cases}$$

Plucking the sunflower

For weakly separable constraints, the formula can be reduced if there is a sunflower with $k+1$ petals. Example:

$$k+1 \begin{cases} \text{EVEN}(x_1, x_2, x_3, x_4, x_5, x_6) \\ \text{EVEN}(x_1, x_2, x_3, x_7, x_8, x_9) \\ \text{EVEN}(x_1, x_2, x_3, 0, 0, 0) \\ \text{EVEN}(x_1, x_2, x_3, x_{13}, x_{14}, x_{15}) \end{cases}$$

Plucking the sunflower

For weakly separable constraints, the formula can be reduced if there is a sunflower with $k+1$ petals. Example:

$$k+1 \begin{cases} \text{EVEN}(x_1, x_2, x_3, x_4, x_5, x_6) \\ \text{EVEN}(x_1, x_2, x_3, x_7, x_8, x_9) \\ \text{EVEN}(x_1, x_2, x_3, 0, 0, 0) \\ \text{EVEN}(x_1, x_2, x_3, x_{13}, x_{14}, x_{15}) \end{cases}$$

$$\Downarrow$$

$$\text{EVEN}(x_1, x_2, x_3)$$

Plucking the sunflower

For weakly separable constraints, the formula can be reduced if there is a sunflower with $k+1$ petals. Example:

$$k+1 \begin{cases} \text{EVEN}(x_1, x_2, x_3, x_4, x_5, x_6) \\ \text{EVEN}(x_1, x_2, x_3, x_7, x_8, x_9) \\ \text{EVEN}(x_1, x_2, x_3, 0, 0, 0) \\ \text{EVEN}(x_1, x_2, x_3, x_{13}, x_{14}, x_{15}) \end{cases}$$

$$\Downarrow$$

$$\text{EVEN}(x_1, x_2, x_3)$$

$$\text{EVEN}(x_4, x_5, x_6)$$

$$\text{EVEN}(x_7, x_8, x_9)$$

$$\text{EVEN}(x_{10}, x_{11}, x_{12})$$

$$\text{EVEN}(x_{13}, x_{14}, x_{15})$$

The algorithm

- If there is a variable that occurs more than $c_{\mathcal{R}}(k)$ times:
  - Find a sunflower with $k+1$ petals
  - Pluck the sunflower $\Rightarrow$ shorter formula
- If every variable occurs at most $c_{\mathcal{R}}(k)$ times:
  - Apply the bounded occurrence algorithm

Running time: $2^{k^{r+2} \cdot 2^{2^{O(r)}}} \cdot n \log n$, where r is the maximum arity in the constraint family $\mathcal{R}$.

Hardness results: case 1

Definition: $\boldsymbol{R}$ is weakly separable if

1. the union of two disjoint satisfying assignments is also satisfying, and
2. if a satisfying assignment contains a smaller satisfying assignment, then their difference is also satisfying.

Hardness results: case 1

Definition: $\boldsymbol{R}$ is weakly separable if

1. the union of two disjoint satisfying assignments is also satisfying, and
2. if a satisfying assignment contains a smaller satisfying assignment, then their difference is also satisfying.

If property 1 is violated:

$$R(0,0,0,0,0,0,0,0) = 1$$

$$R(1,1,1,0,0,0,0,0) = 1$$

$$R(0,0,0,1,1,0,0,0) = 1$$

$$R(1,1,1,1,1,0,0,0) = 0$$

Hardness results: case 1

Definition: R is weakly separable if

1. the union of two disjoint satisfying assignments is also satisfying, and
2. if a satisfying assignment contains a smaller satisfying assignment, then their difference is also satisfying.

If property 1 is violated:

$$R(0,0,0,0,0,0,0,0) = 1$$

$$R(1,1,1,0,0,0,0,0) = 1$$

$$R(0,0,0,1,1,0,0,0) = 1$$

$$R(1,1,1,1,1,0,0,0) = 0$$

$$\Downarrow$$

$$R(x,x,x,y,y,0,0,0) = 1 \iff \bar{x} \vee \bar{y}$$

Hardness results: case 1

Definition: $\boldsymbol{R}$ is weakly separable if

1. the union of two disjoint satisfying assignments is also satisfying, and
2. if a satisfying assignment contains a smaller satisfying assignment, then their difference is also satisfying.

If property 1 is violated:

$$\boldsymbol{R(0,0,0,0,0,0,0,0) = 1}$$

$$\boldsymbol{R(1,1,1,0,0,0,0,0) = 1}$$

$$\boldsymbol{R(0,0,0,1,1,0,0,0) = 1}$$

$$\boldsymbol{R(1,1,1,1,1,0,0,0) = 0}$$

$$\Downarrow$$

MAXIMUM INDEPENDENT SET

$$\boldsymbol{R(x,x,x,y,y,0,0,0) = 1 \iff \bar{x} \vee \bar{y}}$$

$\Rightarrow$ can be expressed!

Hardness results: case 2

Definition: R is weakly separable if

1. the union of two disjoint satisfying assignments is also satisfying, and
2. if a satisfying assignment contains a smaller satisfying assignment, then their difference is also satisfying.

If property 2 is violated:

$$R(0,0,0,0,0,0,0,0) = 1$$

$$R(1,1,1,1,1,0,0,0) = 1$$

$$R(0,0,0,1,1,0,0,0) = 1$$

$$R(1,1,1,0,0,0,0,0) = 0$$

Hardness results: case 2

Definition: R is weakly separable if

1. the union of two disjoint satisfying assignments is also satisfying, and
2. if a satisfying assignment contains a smaller satisfying assignment, then their difference is also satisfying.

If property 2 is violated:

$$R(0,0,0,0,0,0,0,0) = 1$$
$$R(1,1,1,1,1,0,0,0) = 1$$
$$R(0,0,0,1,1,0,0,0) = 1$$
$$R(1,1,1,0,0,0,0,0) = 0$$
$$\Downarrow$$
$$R(x,x,x,y,y,0,0,0) = 1 \iff x \to y$$

Hardness results: case 2

Definition: R is weakly separable if

1. the union of two disjoint satisfying assignments is also satisfying, and
2. if a satisfying assignment contains a smaller satisfying assignment, then their difference is also satisfying.

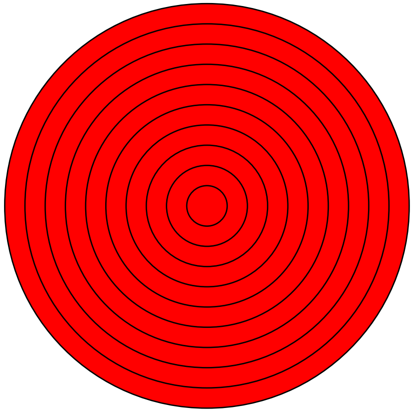
If property 2 is violated:

$$R(0,0,0,0,0,0,0,0) = 1$$
$$R(1,1,1,1,1,0,0,0) = 1$$
$$R(0,0,0,1,1,0,0,0) = 1$$
$$R(1,1,1,0,0,0,0,0) = 0$$
$$\Downarrow$$
$$R(x,x,x,y,y,0,0,0) = 1 \iff x \rightarrow y$$

Lemma: The problem is W[1]-complete for the constraint $\longrightarrow$.

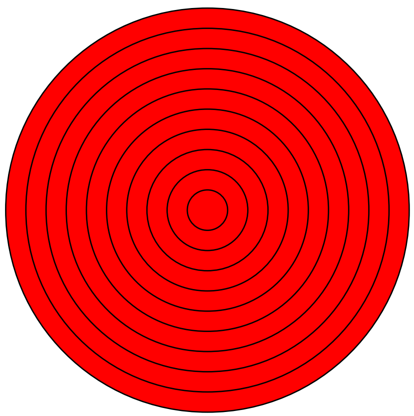
Planar formulae

If the primal graph of the formula is **planar**, then the layering method of Baker can be used.



Planar formulae

If the primal graph of the formula is **planar**, then the layering method of Baker can be used.

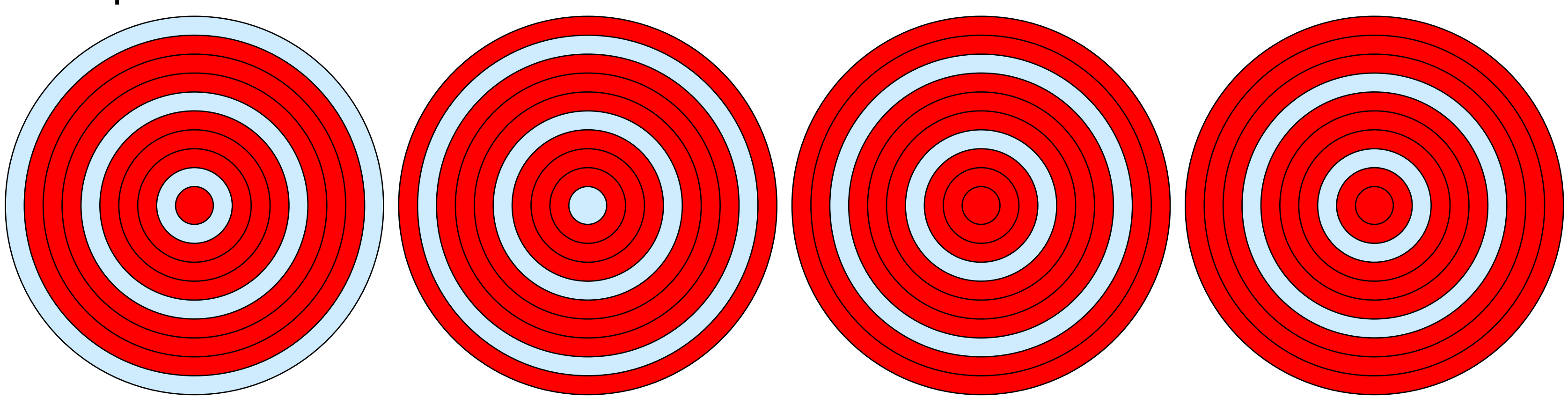


Set to 0 the variables in every $(k+1)$th layer.

There are $k+1$ ways of doing this.

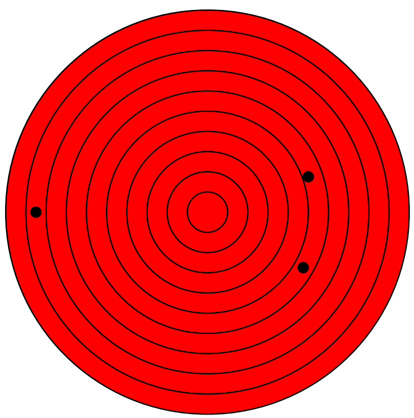
One of them will not hurt the solution.

Example with $k=3$:



Planar formulae

If the primal graph of the formula is **planar**, then the layering method of Baker can be used.

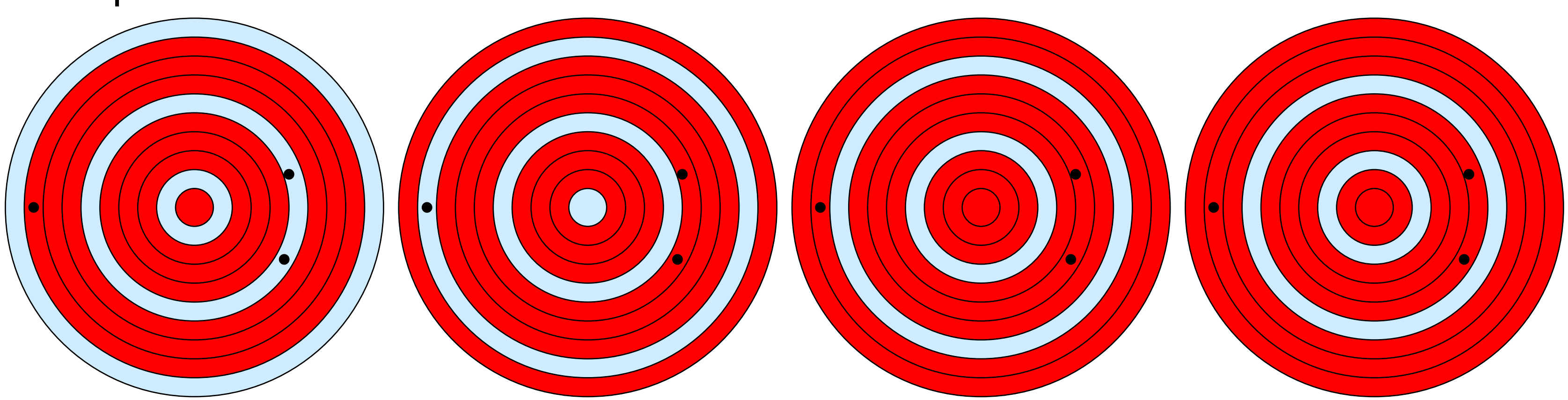


Set to 0 the variables in every $(k+1)$th layer.

There are $k+1$ ways of doing this.

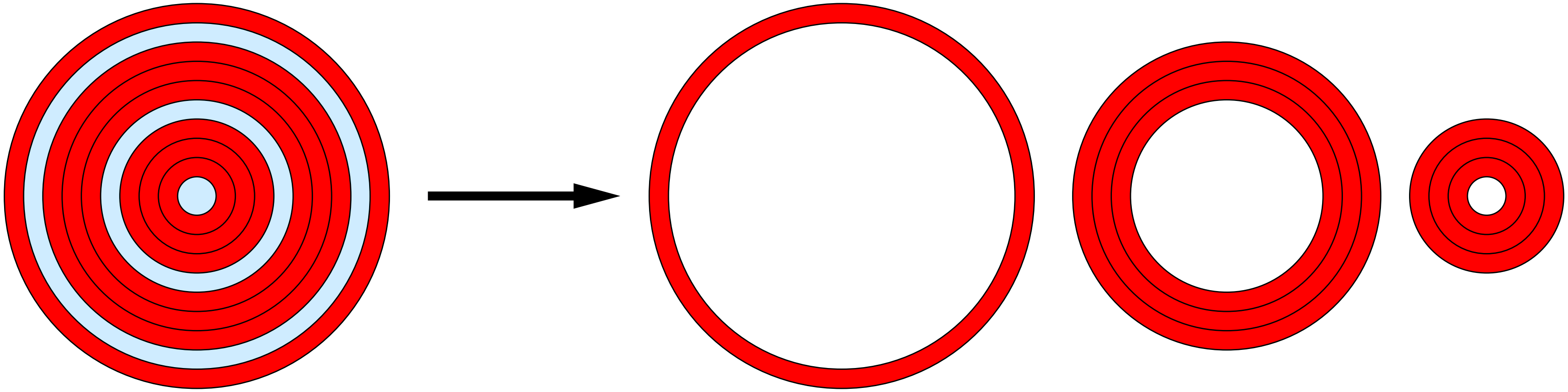
One of them will not hurt the solution.

Example with $k = 3$:



Planar formulae (cont.)

If we delete every $(k+1)$th layer, then the remaining formula has only k layers:

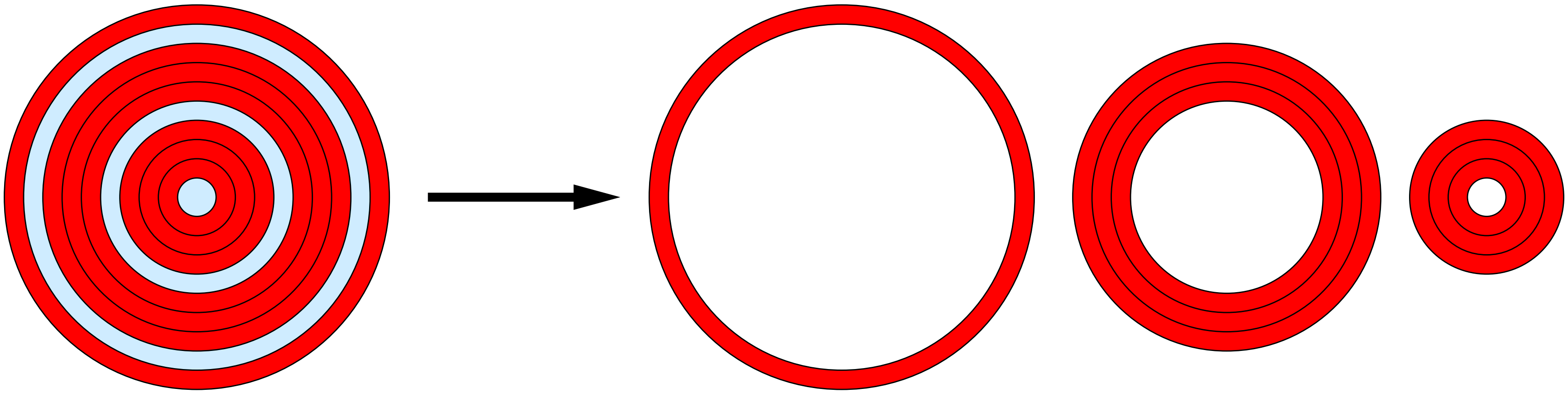


Lemma (Bodlaender): The treewidth of a k-layered graph is at most $3k-1$.

If the primal graph has bounded treewidth, then the problem can be solved in linear time using standard techniques.

Planar formulae (cont.)

If we delete every $(k+1)$th layer, then the remaining formula has only k layers:



Lemma (Bodlaender): The treewidth of a k-layered graph is at most $3k-1$.

If the primal graph has bounded treewidth, then the problem can be solved in linear time using standard techniques.

Incidence graph: bipartite graph, vertices are the clauses and the variables, edge means “appears in”.

Theorem: Linear time alg. if the incidence graph of the formula is planar.

Summary

- Parameterized version of $\mathcal{R}$-SAT
- FPT or W[1]-complete depending on weak separability
- Bounded occurences: color coding using connected solutions
- Reduction using the sunflower lemma
- Linear time solvable for planar and bounded treewidth formulae

Summary

- Parameterized version of $\mathcal{R}$-SAT
- FPT or W[1]-complete depending on weak separability
- Bounded occurences: color coding using connected solutions
- Reduction using the sunflower lemma
- Linear time solvable for planar and bounded treewidth formulae

Thank you for your attention!
Questions?