Algorithmic Meta-Theorems

192.122

WS21/22

Jan Dreier
dreier@ac.tuwien.ac.at

ALGORITHMS AND COMPLEXITY GROUP
TU WIEN Informatics

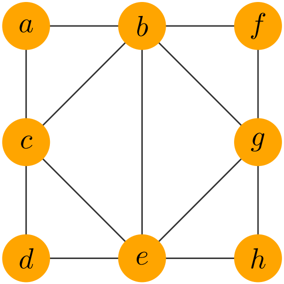
Reminder: Treewidth

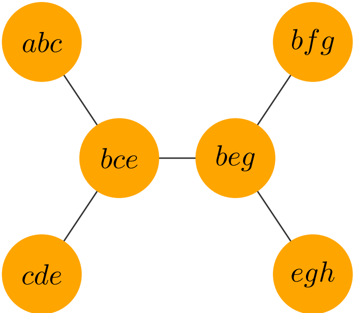
A *tree decomposition* of a graph $G = (V, E)$ is a tree whose vertices are *bags* (subsets of V) and

- every vertex $v \in V$ is contained in some bag,
- every edge $uv \in V$ is contained in some bag,
- for every $v \in V$, the bags containing v are a connected subtree

Width of decomposition: maximum bag size -1

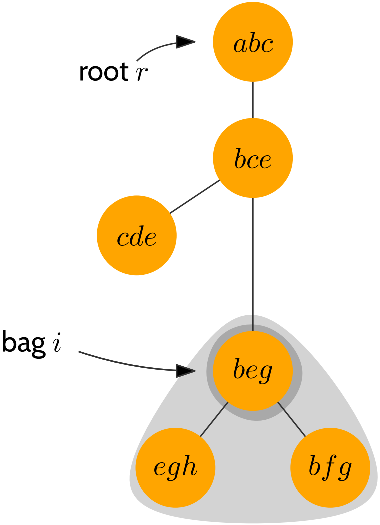
Treewidth of G (tw(G)): minimum width of a decomposition of G

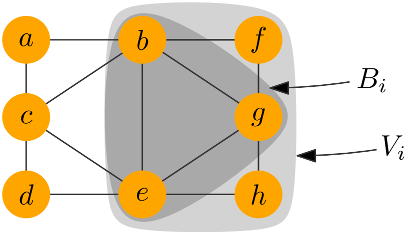




Reminder: Treewidth

- Choose an arbitrary bag r as root and orient the tree accordingly.
- For bag i let B_i be the vertices of G contained in i and V_i be the vertices contained in i or a successor of i.





Reminder: Logic

For a given signature τ, monadic second-order logic has ...

- element-variables ($x, y, z, \dots$) and set-variables ($X, Y, Z, \dots$)
- relations $=$ (equality) and $x \in X$ (membership), as well as the relations from τ.
- quantifiers $\exists$ and $\forall$, as well as operators $\wedge$, $\vee$ and $\neg$

We mostly work on colored undirected graphs with $\tau = \{\sim, c_1, c_2, \dots\}$. Here, we call the logic MSO_1.

If φ is a sentence (a formula without free variables), we write $G \models \varphi$ to indicate that φ holds on G (i.e., G is a model of φ).

Model-Checking

MSO_1 Model-Checking Problem

Input: Graph G and MSO_1-sentence φ

Question: $G \models \varphi$?

Model-Checking

MSO_1 Model-Checking Problem

Input:	Graph G and MSO_1-sentence φ
Question:	$G \models \varphi$?

What is the complexity of the problem?

Model-Checking

MSO_1 Model-Checking Problem

Input:	Graph G and MSO_1-sentence φ
Question:	$G \models \varphi$?

What is the complexity of the problem?

Theorem (Vardi 1982)

MSO_1 Model-Checking is PSPACE-complete.

Model-Checking

MSO_1 Model-Checking Problem

Input:	Graph G and MSO_1-sentence φ
Question:	$G \models \varphi$?

What is the complexity of the problem?

Theorem (Vardi 1982)

MSO_1 Model-Checking is PSPACE-complete.

What is the parameterized complexity if we parameterize by $|\varphi|$?

Model-Checking

MSO_1 Model-Checking Problem

Input: Graph G and MSO_1-sentence φ
Question: $G \models \varphi$?

What is the complexity of the problem?

Theorem (Vardi 1982)

MSO_1 Model-Checking is PSPACE-complete.

What is the parameterized complexity if we parameterize by $|\varphi|$?

Parameterized MSO_1 Model-Checking is paraNP-complete.

Courcelle's Theorem

Courcelle's Theorem

For a MSO_1 sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathrm{tw}(G), |\varphi|)n$ for some function f.

Courcelle's Theorem

Courcelle's Theorem

For a MSO_1 sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathsf{tw}(G), |\varphi|)n$ for some function f.

Assume a problem is expressible by a sentence of length k. Then we can decide the problem in time $f(\mathsf{tw}(G), k)n = g(\mathsf{tw}(G))n$.

Courcelle's Theorem

Courcelle's Theorem

For a MSO$_1$ sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\text{tw}(G), |\varphi|)n$ for some function f.

Assume a problem is expressible by a sentence of length k. Then we can decide the problem in time $f(\text{tw}(G), k)n = g(\text{tw}(G))n$.

Courcelle's Theorem (alternative)

Every graph problem expressible in MSO$_1$ can be solved in time $g(\text{tw}(G))n$ for some function f.

Courcelle's Theorem

For a MSO_1 sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathrm{tw}(G), |\varphi|)n$ for some function f.

Assume a problem is expressible by a sentence of length k. Then we can decide the problem in time $f(\mathrm{tw}(G), k)n = g(\mathrm{tw}(G))n$.

Courcelle's Theorem (alternative)

Every graph problem expressible in MSO_1 can be solved in time $g(\mathrm{tw}(G))n$ for some function f.

Proof: complicated, later on

3-Coloring

> 3-COLORING can be solved in time $g(\text{tw}(G))n$.

How do we prove this?

3-Coloring

3-COLORING can be solved in time $g(\text{tw}(G))n$.

How do we prove this? We write down a MSO_1 sentence φ such that for every graph G holds $G \models \varphi$ iff G is 3-colorable.

3-Coloring

3-COLORING can be solved in time $g(\text{tw}(G))n$.

How do we prove this? We write down a MSO_1 sentence φ such that for every graph G holds $G \models \varphi$ iff G is 3-colorable.

$$\varphi \equiv \exists R \exists G \exists B$$
$$\Big(\forall x x \in R \vee x \in G \vee x \in B\Big)$$
$$\wedge \Big(\forall x \neg(x \in R \wedge x \in G) \wedge \neg(x \in G \wedge x \in B) \wedge \neg(x \in R \wedge x \in B)\Big)$$
$$\wedge \Big(\forall x \forall y \big((x \in R \wedge y \in R) \vee (x \in G \wedge y \in G) \vee (x \in B \wedge y \in B)\big)$$
$$\qquad \to \neg x \sim y\Big)$$

Courcelle: This formula can be evaluated in time $f(\text{tw}(G), 100)n$.

Hamilton Cycle

HAMILTONCYCLE can be solved in time $f(\text{tw}(G))n$ for some function f.

Hamilton Cycle

HamiltonCycle can be solved in time $f(\text{tw}(G))n$ for some function f.

Problem?

Courcelle, Engelfriet 2012

HamiltonCycle is not expressible in MSO_1.

Hamilton Cycle

HAMILTONCYCLE can be solved in time $f(\text{tw}(G))n$ for some function f.

Problem?

Courcelle, Engelfriet 2012

HAMILTONCYCLE is not expressible in MSO_1.

Solution: We extend the logic (and use the fact that subdividing edges of a graph does not increase the treewidth).

Allowing Edge Quantification

We consider the more powerful logic MSO_2.

- We allow quantification over edges, vertices, sets of edges or sets of vertices.
- To be more readable, we use the letters
  - $e, f, \ldots$ for edges,
  - $u, v, \ldots$ for vertices,
  - $E, F, \ldots$ or sets of edges,
  - $U, V, \ldots$ for sets of vertices.
- A relation $u{\sim}v$ if two vertices are connected.
- A relation inc(u, e) if a vertex u is incident with an edge e.

Expressing Hamilton Cycle

Expressing Hamilton Cycle

$$\varphi_{\text{Hamilton}} = \exists E\, \varphi_{\text{cycle cover}}(E) \wedge \varphi_{\text{connected}}(E)$$

Expressing Hamilton Cycle

$$\varphi_{\text{Hamilton}} = \exists E\, \varphi_{\text{cycle cover}}(E) \wedge \varphi_{\text{connected}}(E)$$

$$\varphi_{\text{cycle cover}}(E) =$$

Expressing Hamilton Cycle

$$\varphi_{\text{Hamilton}} = \exists E\, \varphi_{\text{cycle cover}}(E) \wedge \varphi_{\text{connected}}(E)$$

$$\varphi_{\text{cycle cover}}(E) =$$
$$\forall v\, \exists e_1\, \exists e_2\, e_1 \neq e_2 \wedge \mathsf{inc}(e_1, v) \wedge \mathsf{inc}(e_2, v)$$
$$\wedge \neg \exists e_3\, e_3 \neq e_1 \wedge e_3 \neq e_2 \wedge \mathsf{inc}(e_3, v)$$

Expressing Hamilton Cycle

$$\varphi_{\textsf{Hamilton}} = \exists E\, \varphi_{\textsf{cycle cover}}(E) \wedge \varphi_{\textsf{connected}}(E)$$

$$\varphi_{\textsf{cycle cover}}(E) =$$
$$\forall v\, \exists e_1\, \exists e_2\, e_1 \neq e_2 \wedge \textsf{inc}(e_1, v) \wedge \textsf{inc}(e_2, v)$$
$$\wedge \neg \exists e_3\, e_3 \neq e_1 \wedge e_3 \neq e_2 \wedge \textsf{inc}(e_3, v)$$

$$\varphi_{\textsf{connected}}(E) =$$

Expressing Hamilton Cycle

$$\varphi_{\text{Hamilton}} = \exists E\, \varphi_{\text{cycle cover}}(E) \wedge \varphi_{\text{connected}}(E)$$

$$\varphi_{\text{cycle cover}}(E) =$$
$$\forall v\, \exists e_1\, \exists e_2\, e_1 \neq e_2 \wedge \text{inc}(e_1, v) \wedge \text{inc}(e_2, v)$$
$$\wedge \neg \exists e_3\, e_3 \neq e_1 \wedge e_3 \neq e_2 \wedge \text{inc}(e_3, v)$$

$$\varphi_{\text{connected}}(E) =$$

"For every partition of $G[E]$ in two halfs,
there is an edge between these halfs"

Courcelle's Theorem

Courcelle's Theorem (Edge Set Quantification)

For a MSO_2 sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathrm{tw}(G), |\varphi|)n$ for some function f.

Courcelle's Theorem (Edge Set Quantification)

For a MSO_2 sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathsf{tw}(G), |\varphi|)n$ for some function f.

For an MSO_2 sentence φ and a graph G, we construct an MSO_1 sentence φ' and a graph G' such that

$$G \models \varphi \iff G' \models \varphi'.$$

Courcelle's Theorem (Edge Set Quantification)

For a MSO_2 sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathsf{tw}(G), |\varphi|)n$ for some function f.

For an MSO_2 sentence φ and a graph G, we construct an MSO_1 sentence φ' and a graph G' such that

$$G \models \varphi \iff G' \models \varphi'.$$

What else do we want/need?

Courcelle's Theorem

Courcelle's Theorem (Edge Set Quantification)

For a MSO_2 sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathsf{tw}(G), |\varphi|)n$ for some function f.

For an MSO_2 sentence φ and a graph G, we construct an MSO_1 sentence φ' and a graph G' such that

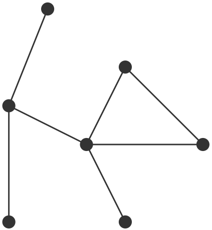
$$G \models \varphi \iff G' \models \varphi'.$$

What else do we want/need?

- $\mathsf{tw}(G') = \mathsf{tw}(G)$
- $n' = O(\mathsf{tw}(G)n)$
- $|\varphi'| = g(|\varphi|)$

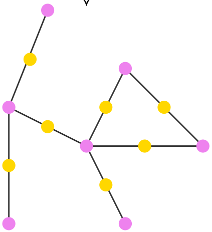
Proof

- G': Color vertices violet and subdivide edges using yellow vertices.



G with treewidth w

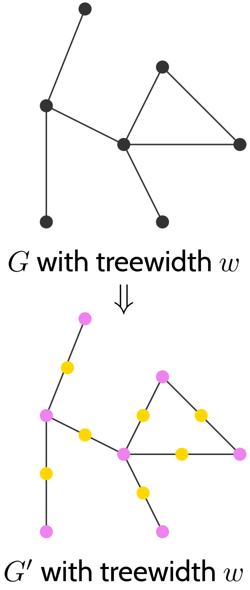
$\Downarrow$



G' with treewidth w

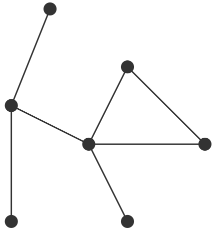
Proof

- G': Color vertices violet and subdivide edges using yellow vertices.
  - Does not increase treewidth: When subdividing uv with vertex e, add bag uve below bag containing uv.



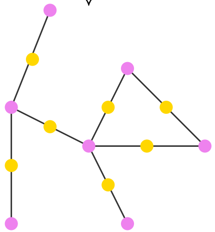
Proof

- G': Color vertices violet and subdivide edges using yellow vertices.
  - Does not increase treewidth: When subdividing uv with vertex e, add bag uve below bag containing uv.
  - Size n' stays bounded: Every graph with treewidth w has $O(wn)$ edges.



G with treewidth w

$\Downarrow$

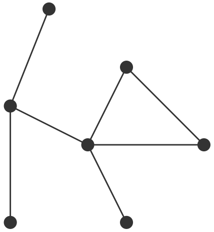


G' with treewidth w

Proof

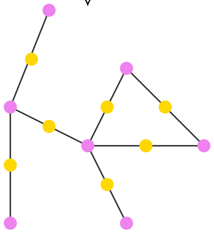
- G': Color vertices violet and subdivide edges using yellow vertices.
  - Does not increase treewidth: When subdividing uv with vertex e, add bag uve below bag containing uv.
  - Size n' stays bounded: Every graph with treewidth w has $O(wn)$ edges.

- φ': Relativize quantifiers
  - $\exists v\, \psi \rightsquigarrow \exists v\,$ violet$(v) \wedge \psi$,
  - $\exists e\, \psi \rightsquigarrow \exists v\,$ yellow$(e) \wedge \psi$,
  - …



G with treewidth w

$\Downarrow$



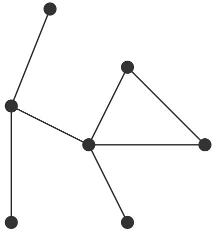
G' with treewidth w

Proof

- G': Color vertices violet and subdivide edges using yellow vertices.
  - Does not increase treewidth: When subdividing uv with vertex e, add bag uve below bag containing uv.
  - Size n' stays bounded: Every graph with treewidth w has $O(wn)$ edges.

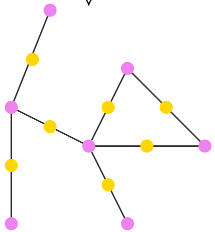
- φ': Relativize quantifiers
  - $\exists v\, \psi \rightsquigarrow \exists v\, \text{violet}(v) \wedge \psi$,
  - $\exists e\, \psi \rightsquigarrow \exists v\, \text{yellow}(e) \wedge \psi$,
  - …

  and replace relations
  - $u{\sim}v \rightsquigarrow \exists e\, u{\sim}e \wedge e{\sim}v$,
  - $\text{inc}(u, e) \rightsquigarrow u{\sim}e$.



G with treewidth w

$\Downarrow$



G' with treewidth w

Why MSO_1?

If we can also do model-checking for MSO_2, why do we care about MSO_1?

- MSO_2 model-checking is a simple corollary.
- MSO_2 is hard on dense graphs, while MSO_1 is not.

Optimization

So far, we know how to deal with yes/no-problems such as

- Is there a 3-coloring?
- Is there a Hamilton cycle?

Optimization

So far, we know how to deal with yes/no-problems such as

- Is there a 3-coloring?
- Is there a Hamilton cycle?

How do we deal with optimization problems such as

- is there an independent set of size at least k?
- is there a dominating set of size at most k?
- is there a vertex cover of size at most k?

Independent Set

X is independent set in G iff $G \models \varphi(X)$ with

$$\varphi(X) \equiv \neg\exists x \exists y \big(x \in X \land y \in X \land x{\sim}y\big).$$

Independent Set

X is independent set in G iff $G \models \varphi(X)$ with

$$\varphi(X) \equiv \neg\exists x \exists y \big(x \in X \wedge y \in X \wedge x \sim y\big).$$

Independent set of size at least k expressed by

Independent Set

X is independent set in G iff $G \models \varphi(X)$ with

$$\varphi(X) \equiv \neg\exists x \exists y \big(x \in X \wedge y \in X \wedge x{\sim}y\big).$$

Independent set of size at least k expressed by

$$\exists X \varphi(X) \wedge \exists x_1 \ldots \exists x_k \big(\bigwedge_i x_i \in X \wedge \bigwedge_{i \neq j} x_i \neq x_j\big).$$

Independent Set

X is independent set in G iff $G \models \varphi(X)$ with

$$\varphi(X) \equiv \neg\exists x \exists y \big(x \in X \wedge y \in X \wedge x \sim y\big).$$

Independent set of size at least k expressed by

$$\exists X \varphi(X) \wedge \exists x_1 \ldots \exists x_k \big(\bigwedge_i x_i \in X \wedge \bigwedge_{i \neq j} x_i \neq x_j\big).$$

What is the problem?

Independent Set

X is independent set in G iff $G \models \varphi(X)$ with

$$\varphi(X) \equiv \neg\exists x \exists y \big(x \in X \wedge y \in X \wedge x{\sim}y\big).$$

Independent set of size at least k expressed by

$$\exists X \varphi(X) \wedge \exists x_1 \ldots \exists x_k \big(\bigwedge_i x_i \in X \wedge \bigwedge_{i \neq j} x_i \neq x_j\big).$$

What is the problem?
Formula very long, since k can be as large as n.

Optimization Theorem

Optimization Theorem (Courcelle, Makowsky, Rotics 2000)

For a MSO_1 formula $\varphi(X)$ and graph G one can compute in time $f(\mathrm{tw}(G), |\varphi|)n$ a set $S^* \subseteq V(G)$ such that $G \models \varphi(S^*)$ and

$$|S^*| = \max\{|S| \colon G \models \varphi(S), S \subseteq V(G)\}$$

or

$$|S^*| = \min\{|S| \colon G \models \varphi(S), S \subseteq V(G)\}$$

(or get the answer that no such S^* exists).

Independent Set

How can we use this theorem to solve independent set?

INDEPENDENTSET

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

Independent Set

How can we use this theorem to solve independent set?

INDEPENDENTSET

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

By finding a set S^* that maximizes

$$\varphi(X) \equiv \neg\exists x \exists y \big(x \in X \wedge y \in X \wedge x{\sim}y\big),$$

and then checking if $|S|^* \geq k$.

Parity

Can we express this property in MSO_1 (or MSO_2)?

- "G has an even number of green vertices"

Parity

Can we express this property in MSO_1 (or MSO_2)?

- “G has an even number of green vertices”

No, but we can express it in CMSO_1:

- Allow quantifiers $\#_{k,m} x\, \varphi$
- “There are, modulo m, exactly k elements x satisfying φ”

Parity

Can we express this property in MSO_1 (or MSO_2)?

- "G has an even number of green vertices"

No, but we can express it in CMSO_1:

- Allow quantifiers $\#_{k,m} x \, \varphi$
- "There are, modulo m, exactly k elements x satisfying φ"
- Property expressed by formula $\#_{0,2} x \, \mathrm{green}(x)$.

Parity

Can we express this property in MSO_1 (or MSO_2)?

- "G has an even number of green vertices"

No, but we can express it in CMSO_1:

- Allow quantifiers $\#_{k,m} x\, \varphi$
- "There are, modulo m, exactly k elements x satisfying φ"
- Property expressed by formula $\#_{0,2} x$ green(x).
- Can we express that a graph has a Euler cycle?

Parity

Can we express this property in MSO_1 (or MSO_2)?

- "G has an even number of green vertices"

No, but we can express it in CMSO_1:

- Allow quantifiers $\#_{k,m} x\, \varphi$
- "There are, modulo m, exactly k elements x satisfying φ"
- Property expressed by formula $\#_{0,2} x\, \text{green}(x)$.
- Can we express that a graph has a Euler cycle?

Courcelle's Theorem (Modulo Extension)

For a CMSO_1 sentence φ and graph G one can decide whether $G \models \varphi$ in time $f(\text{tw}(G), |\varphi|, m)n$ for some function f, where m is the largest modulo-base in φ.

Summary

Courcelle's theorem is a very powerful tool to solve problems on bounded treewidth. It comes in various flavours.

- MSO_1: base variant,
- MSO_2: edge quantifiers,
- CMSO: parity/modulo counting,
- LinEMSOL: optimization,
- and any combination thereof.

Proving Courcelle's Theorem

We now want to prove the following.

Courcelle's Theorem

For a MSO_1 formula φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathrm{tw}(G), |\varphi|)n$ for some function f.

Proving Courcelle's Theorem

We now want to prove the following.

Courcelle's Theorem

For a MSO_1 formula φ and graph G one can decide whether $G \models \varphi$ in time $f(\text{tw}(G), |\varphi|)n$ for some function f.

- Historically proven by converting MSO_1-formulas into tree-automata. Use this automaton to traverse the tree-decomposition.

Proving Courcelle's Theorem

We now want to prove the following.

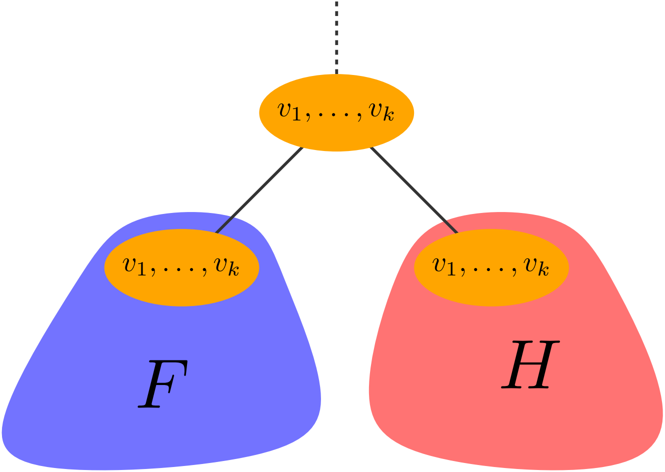
Courcelle's Theorem

For a MSO_1 formula φ and graph G one can decide whether $G \models \varphi$ in time $f(\mathrm{tw}(G), |\varphi|)n$ for some function f.

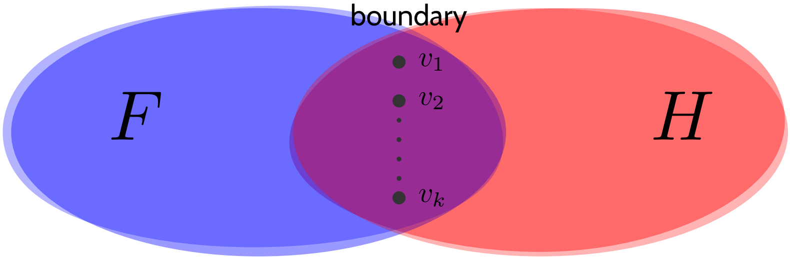
- Historically proven by converting MSO_1-formulas into tree-automata. Use this automaton to traverse the tree-decomposition.
- We prove it using a powerful logic-theorem by Fefermann and Vaught as a blackbox.

Join Nodes

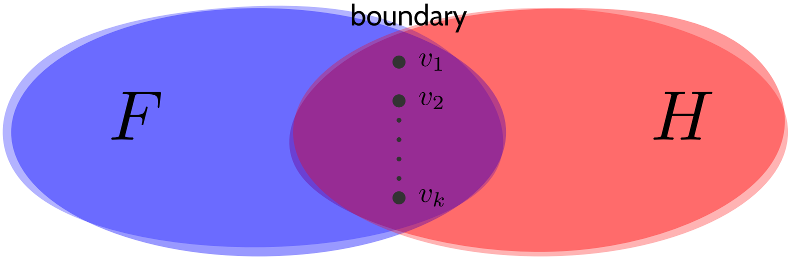
We can assume we are given a nice tree decomposition. If we manage the *join* operation, *introduce* and *forget* are easy.



Idea



Idea



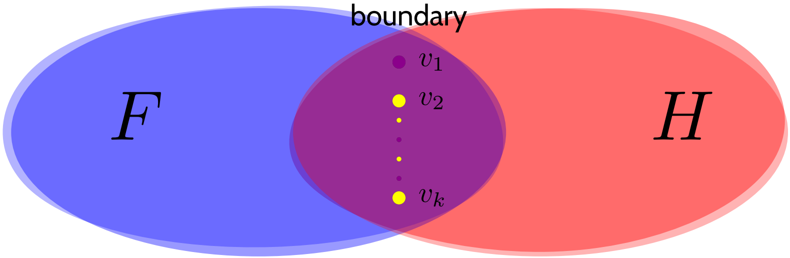
Information of F + boundary

Information of H + boundary

Information of $F \cup H$ + boundary

Idea

Independent set: store how independent sets intersect the boundary

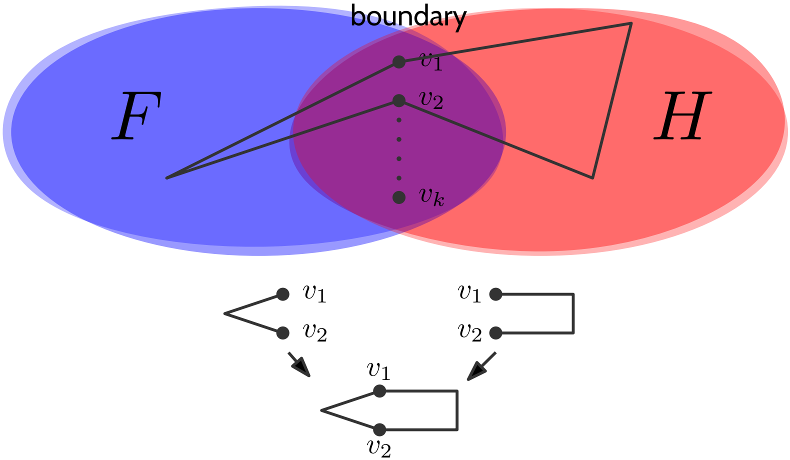


F: 7 hidden + 5 boundary H: 9 hidden + 5 boundary

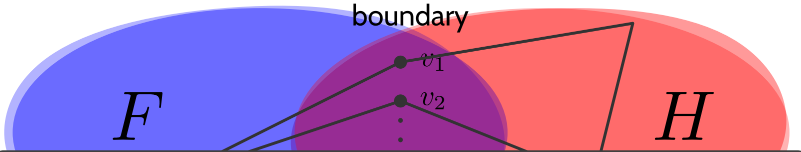
$F \cup H$: 16 hidden + 5 boundary

Idea

Subgraphs: store how subgraphs $\leq q$ intersect the boundary



Subgraphs: store how subgraphs $\leq q$ intersect the boundary



Observation

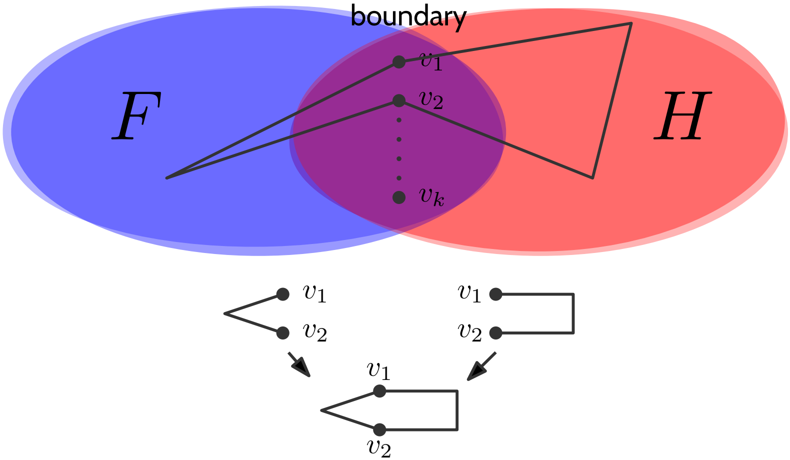
If we know

- in F how subgraphs of size $\leq q$ intersect the boundary
- in H how subgraphs of size $\leq q$ intersect the boundary

then we can compute in $F \cup H$ how subgraphs of size $\leq q$ intersect the boundary.

Idea

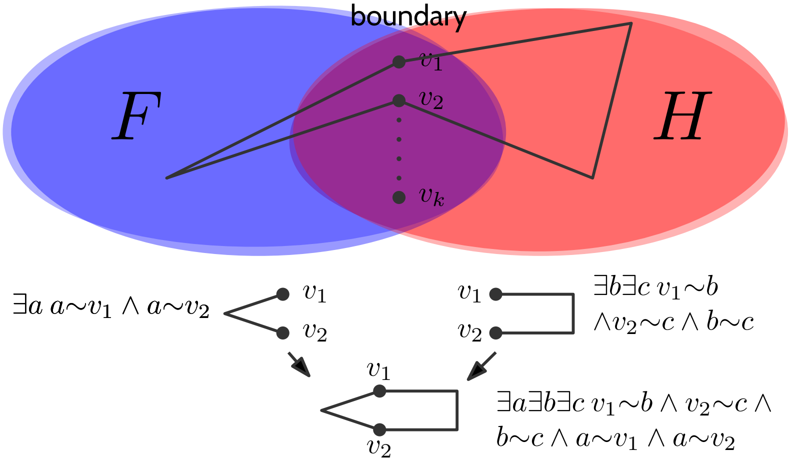
Subgraphs: store how subgraphs $\leq q$ intersect the boundary



Idea

Subgraphs: store how subgraphs $\leq q$ intersect the boundary

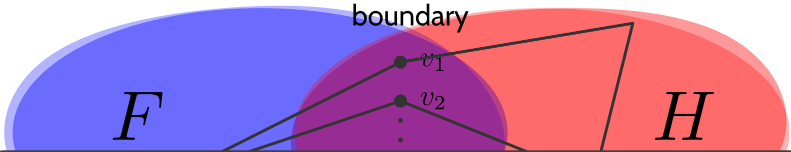
Existential FO: store how formulas $\leq q$ intersect the boundary



Idea

Subgraphs: store how subgraphs $\leq q$ intersect the boundary

Existential FO: store how formulas $\leq q$ intersect the boundary



Observation

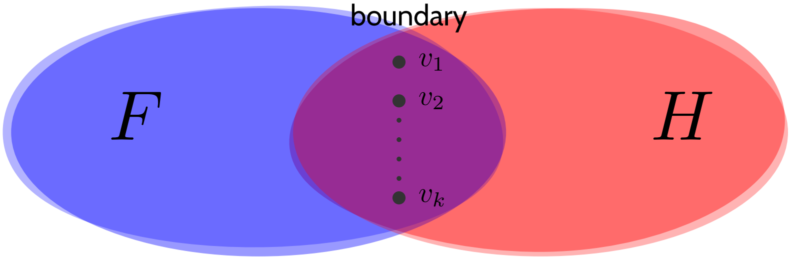
If we know

- in F how formulas of length $\leq q$ intersect the boundary
- in H how formulas of length $\leq q$ intersect the boundary

then we can compute in $F \cup H$ how formulas of length $\leq q$ intersect the boundary.

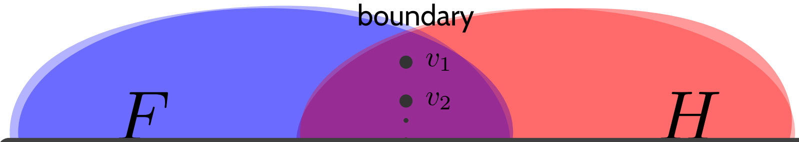
Fefermann–Vaught

Let H be a graph with boundary $v_1, \ldots, v_k$. We define q-type$(H; v_1, \ldots, v_k)$ to be the set of all MSO$_1$-formulas $\xi(x_1, \ldots, x_k)$ of quantifier-rank $\leq q$ with $H \models \xi(v_1, \ldots, v_k)$.



Fefermann–Vaught

Let H be a graph with boundary $v_1, \ldots, v_k$. We define q-type$(H; v_1, \ldots, v_k)$ to be the set of all MSO$_1$-formulas $\xi(x_1, \ldots, x_k)$ of quantifier-rank $\leq q$ with $H \models \xi(v_1, \ldots, v_k)$.



Theorem (Fefermann–Vaught)

Let F and H be graphs with $V(F) \cap V(H) = \{v_1, \ldots, v_k\}$.
If we know

- q-type$(F; v_1, \ldots, v_k)$
- q-type$(H; v_1, \ldots, v_k)$

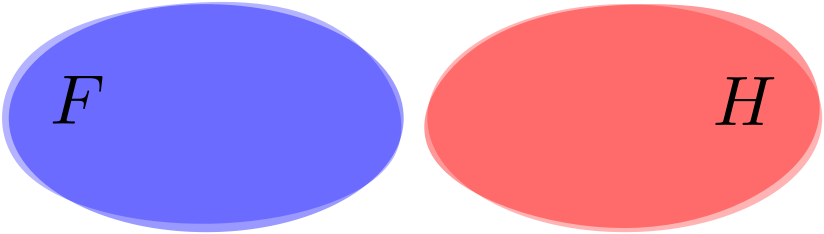
then we can compute q-type$(F \cup H; v_1, \ldots, v_k)$.

Fefermann–Vaught (Classical)

The “boundaried version” of Fefermann–Vaught is a direct consequence of the “classical version”.

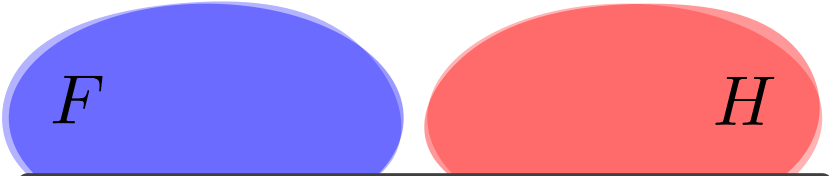
Fefermann–Vaught (Classical)

q-type(H) is the set of all MSO$_1$-sentences ξ of quantifier-rank $\leq q$ with $H \models \xi$.



Fefermann–Vaught (Classical)

q-type(H) is the set of all MSO$_1$-sentences ξ of quantifier-rank $\leq q$ with $H \models \xi$.



Theorem (Classical Fefermann–Vaught)

Let F and H be graphs with $V(F) \cap V(H) = \emptyset$.
Then q-type$(F \cup H)$ is complexity determined by (and can be computed from)

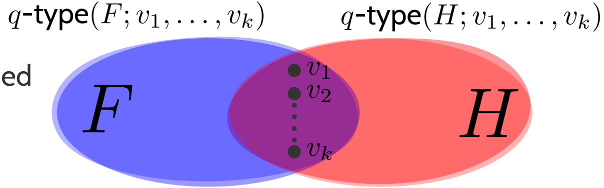
- q-type(F),
- q-type(H).

Reduction

- Proof that classical version implies boundaried version.

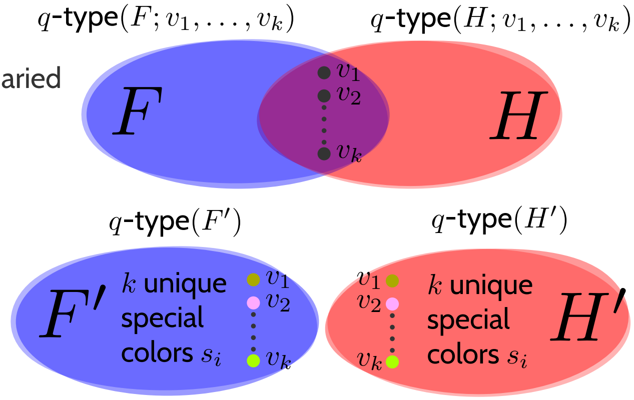
Reduction

- Proof that classical version implies boundaried version.



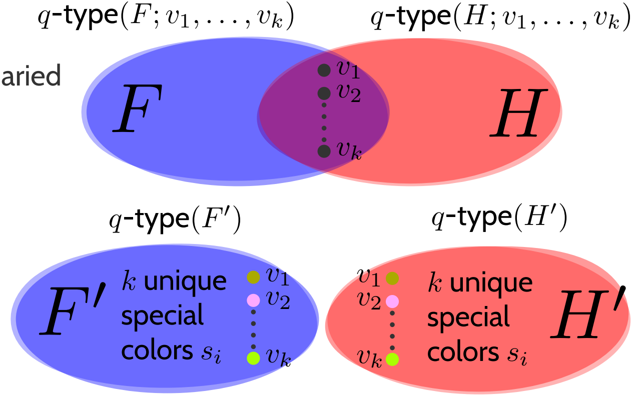
Reduction

- Proof that classical version implies boundaried version.



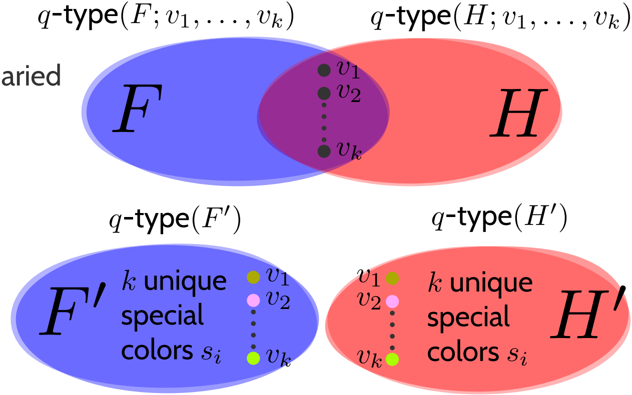
Reduction

- Proof that classical version implies boundaried version.
- q-type$(F; v_1, \ldots, v_k)$ implies q-type(F'): replace $s_i(x)$ with $x = v_i$.



Reduction

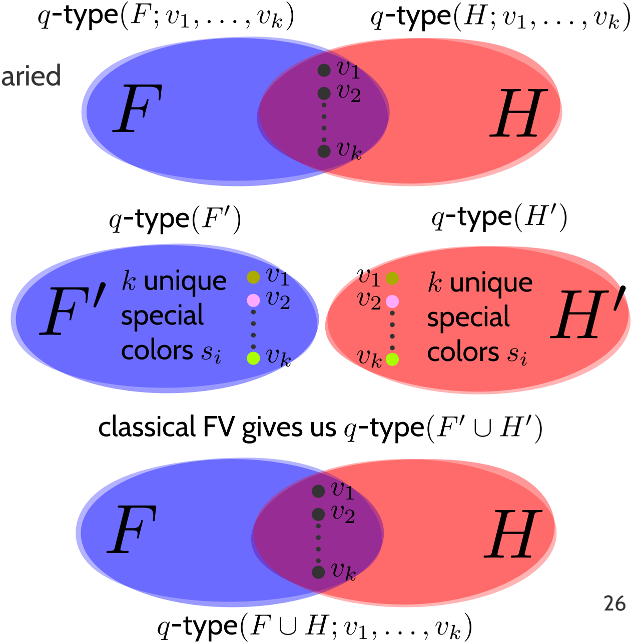
- Proof that classical version implies boundaried version.
- q-type$(F; v_1, \ldots, v_k)$ implies q-type(F'): replace $s_i(x)$ with $x = v_i$.



classical FV gives us q-type$(F' \cup H')$

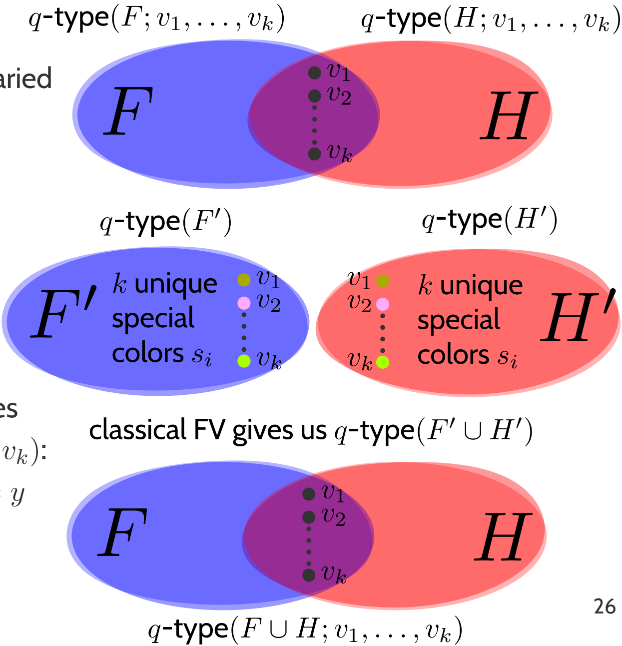
Reduction

- Proof that classical version implies boundaried version.
- q-type$(F; v_1, \dots, v_k)$ implies q-type(F'): replace $s_i(x)$ with $x = v_i$.



Reduction

- Proof that classical version implies boundaried version.
- q-type$(F; v_1, \ldots, v_k)$ implies q-type(F'): replace $s_i(x)$ with $x = v_i$.
- q-type$(F' \cup H')$ implies q-type$(F \cup H; v_1, \ldots, v_k)$: replace $x = y$ with $x = y \vee \bigwedge_i s_i(x) \wedge s_i(y), \ldots$



Dynamic Programming

We have a nice tree decomposition of a graph G and want to know whether $G \models \varphi$ for a formula with quantifier-rank q.

We have the Fefermann–Vaught theorem that tells us how to aggregate q-types when joining two subgraphs.

Dynamic Programming

We have a nice tree decomposition of a graph G and want to know whether $G \models \varphi$ for a formula with quantifier-rank q.

We have the Fefermann–Vaught theorem that tells us how to aggregate q-types when joining two subgraphs.

For bag i (with boundary $v_1, \ldots, v_k$) we store for each formula $\xi(x_1, \ldots, x_k)$ with quantifier-rank $\leq q$ a table entry

$$M_i(\xi) = \begin{cases} 1 & G[V_i] \models \xi(v_1, \ldots, v_k) \\ 0 & \text{otherwise.} \end{cases}$$

Dynamic Programming

We have a nice tree decomposition of a graph G and want to know whether $G \models \varphi$ for a formula with quantifier-rank q.

We have the Fefermann–Vaught theorem that tells us how to aggregate q-types when joining two subgraphs.

For bag i (with boundary $v_1, \ldots, v_k$) we store for each formula $\xi(x_1, \ldots, x_k)$ with quantifier-rank $\leq q$ a table entry

$$M_i(\xi) = \begin{cases} 1 & G[V_i] \models \xi(v_1, \ldots, v_k) \\ 0 & \text{otherwise.} \end{cases}$$

Let r be the root-node. Then $G \models \varphi$ iff $G[V_r] \models \varphi$ iff $M_r(\varphi) = 1$.

Size of q-types

We want to decide whether $G \models \varphi$ in time $f(\text{tw}(G), q)n$. Dynamic programming is only fast if the tables are small.

Size of q-types

We want to decide whether $G \models \varphi$ in time $f(\text{tw}(G), q)n$. Dynamic programming is only fast if the tables are small.

We have to show that the number of formulas ξ of quantifier-rank $\leq q$ with $\leq \text{tw}(G) + 1$ free variables is bounded by some function $f(\text{tw}(G), |\varphi|)$. This bounds the number table entries ξ in $M_i(\xi)$.

Size of q-types

We want to decide whether $G \models \varphi$ in time $f(\text{tw}(G), q)n$. Dynamic programming is only fast if the tables are small.

We have to show that the number of formulas ξ of quantifier-rank $\leq q$ with $\leq \text{tw}(G) + 1$ free variables is bounded by some function $f(\text{tw}(G), |\varphi|)$. This bounds the number table entries ξ in $M_i(\xi)$.

- Remove all colors except for the $\leq |\varphi|$ many that occur in φ.

Size of q-types

We want to decide whether $G \models \varphi$ in time $f(\text{tw}(G), q)n$. Dynamic programming is only fast if the tables are small.

We have to show that the number of formulas ξ of quantifier-rank $\leq q$ with $\leq \text{tw}(G) + 1$ free variables is bounded by some function $f(\text{tw}(G), |\varphi|)$. This bounds the number table entries ξ in $M_i(\xi)$.

- Remove all colors except for the $\leq |\varphi|$ many that occur in φ.
- "Normalize" all formulas:

$$\exists x\, x = x \wedge x = x \wedge x = x \wedge x = x \wedge x = x \wedge x = x \wedge x = x \wedge \ldots$$

Size of q-types

We want to decide whether $G \models \varphi$ in time $f(\text{tw}(G), q)n$. Dynamic programming is only fast if the tables are small.

We have to show that the number of formulas ξ of quantifier-rank $\leq q$ with $\leq \text{tw}(G) + 1$ free variables is bounded by some function $f(\text{tw}(G), |\varphi|)$. This bounds the number table entries ξ in $M_i(\xi)$.

- Remove all colors except for the $\leq |\varphi|$ many that occur in φ.
- "Normalize" all formulas:

$$\exists x\, x = x$$

Size of q-types

We want to decide whether $G \models \varphi$ in time $f(\text{tw}(G), q)n$. Dynamic programming is only fast if the tables are small.

We have to show that the number of formulas ξ of quantifier-rank $\leq q$ with $\leq \text{tw}(G) + 1$ free variables is bounded by some function $f(\text{tw}(G), |\varphi|)$. This bounds the number table entries ξ in $M_i(\xi)$.

- Remove all colors except for the $\leq |\varphi|$ many that occur in φ.
- "Normalize" all formulas:

$$\exists x\, x = x$$

Show the claim by induction.

Size of q-types

We want to decide whether $G \models \varphi$ in time $f(\text{tw}(G), q)n$. Dynamic programming is only fast if the tables are small.

We have to show that the number of formulas ξ of quantifier-rank $\leq q$ with $\leq \text{tw}(G) + 1$ free variables is bounded by some function $f(\text{tw}(G), |\varphi|)$. This bounds the number table entries ξ in $M_i(\xi)$.

- Remove all colors except for the $\leq |\varphi|$ many that occur in φ.
- "Normalize" all formulas:

$$\exists x\, x = x$$

Show the claim by induction. Base case $q = 0$: There are only $2^{2^{O(k \cdot |\varphi|)}}$ many quantifier-free formulas with $\leq k$ variables.

Size of q-types

Show the claim by induction. Base case $q = 0$: There are only $2^{2^{O(k \cdot |\varphi|)}}$ many quantifier-free formulas with $\leq k$ variables.

Size of q-types

Assume we have formulas $\xi_1, \ldots, \xi_l$ with of quantifier-rank $\leq q-1$ and $\leq k+1$ free variables.

Size of q-types

Assume we have formulas $\xi_1, \dots, \xi_l$ with of quantifier-rank $\leq q-1$ and $\leq k+1$ free variables.

Formulas with of quantifier-rank $\leq q$ and $\leq k$ free variables are of the form

$$\big(\forall x \xi_1 \wedge \exists x \xi_4 \wedge \exists x \xi_8 \wedge \dots\big) \vee$$
$$\big(\exists x \xi_3 \wedge \forall x \xi_2 \wedge \exists x \xi_9 \wedge \forall x \xi_1 \wedge \dots\big) \vee$$
$$\big(\exists x \xi_5 \wedge \forall x \xi_8 \wedge \dots\big) \vee \dots$$

Size of q-types

Assume we have formulas $\xi_1, \ldots, \xi_l$ with of quantifier-rank $\leq q-1$ and $\leq k+1$ free variables.

Formulas with of quantifier-rank $\leq q$ and $\leq k$ free variables are of the form

$$\big(\forall x \xi_1 \wedge \exists x \xi_4 \wedge \exists x \xi_8 \wedge \ldots\big) \vee$$
$$\big(\exists x \xi_3 \wedge \forall x \xi_2 \wedge \exists x \xi_9 \wedge \forall x \xi_1 \wedge \ldots\big) \vee$$
$$\big(\exists x \xi_5 \wedge \forall x \xi_8 \wedge \ldots\big) \vee \ldots$$

There are at most $2^{2^{2l}}$ of them.

Size of q-types

Assume we have formulas $\xi_1, \ldots, \xi_l$ with of quantifier-rank $\leq q-1$ and $\leq k+1$ free variables.

Formulas with of quantifier-rank $\leq q$ and $\leq k$ free variables are of the form

$$\big(\forall x \xi_1 \wedge \exists x \xi_4 \wedge \exists x \xi_8 \wedge \ldots\big) \vee$$
$$\big(\exists x \xi_3 \wedge \forall x \xi_2 \wedge \exists x \xi_9 \wedge \forall x \xi_1 \wedge \ldots\big) \vee$$
$$\big(\exists x \xi_5 \wedge \forall x \xi_8 \wedge \ldots\big) \vee \ldots$$

There are at most $2^{2^{2l}}$ of them. In total, the number of formulas is roughly

$$\underbrace{2^{\cdot^{\cdot^{\cdot^{2^{O(\mathsf{tw}(G)\cdot|\varphi|)}}}}}}_{2q}.$$

Size of q-types

Assume we have formulas $\xi_1, \ldots, \xi_l$ with of quantifier-rank $\leq q-1$ and $\leq k+1$ free variables.

Formulas with of quantifier-rank $\leq q$ and $\leq k$ free variables are of the form

$$\big(\forall x \xi_1 \wedge \exists x \xi_4 \wedge \exists x \xi_8 \wedge \ldots\big) \vee$$
$$\big(\exists x \xi_3 \wedge \forall x \xi_2 \wedge \exists x \xi_9 \wedge \forall x \xi_1 \wedge \ldots\big) \vee$$
$$\big(\exists x \xi_5 \wedge \forall x \xi_8 \wedge \ldots\big) \vee \ldots$$

There are at most $2^{2^{2l}}$ of them. In total, the number of formulas is roughly

$$\underbrace{2^{\cdot^{\cdot^{\cdot^{2^{O(\mathsf{tw}(G)\cdot|\varphi|)}}}}}}_{2q}.$$

This bound cannot be improved much.