Tree-decompositions and Logic

Jörg Flum
Albert-Ludwigs-Universität Freiburg
(Germany)

Contents

1. NP-complete problems restricted to trees.

Two NP-complete problems for graphs:

3-COL	
Input:	A graph $\mathcal{G} = (V, E)$.
Question:	Is $\mathcal{G}$ 3-colorable?

3-coloring: A function $C : V \to [3]$ such that $C(u) \neq C(v)$ for $\{u, v\} \in E$.

HAM	
Input:	A graph $\mathcal{G} = (V, E)$.
Question:	Is there a Hamiltonian cycle?

Hamiltonian cycle: A cycle visiting every vertex exactly once.

tree: connected acyclic graph $\mathcal{T} = (T, E)$ nodes

root $r^{\mathcal{T}}$, partial order $\leq^{\mathcal{T}}$ on T:

$$t \leq^{\mathcal{T}} t' \iff t \text{ appears on the (unique) path from } r^{\mathcal{T}} \text{ to } t'.$$

3-COL(TREE), HAM(TREE) are trivial.

IS

Input: A graph $\mathcal{G}$ and a natural number k.

Question: Does $\mathcal{G}$ have an independent set of size k?

I independent set: $\{u, v\} \notin E$ for all $u, v \in I$.

IS(TREE) is solvable in linear time:

For $t \in T$:

$$\mathcal{T}_{\geq t} = \text{subtree rooted at } t. \text{ Its set of nodes is } T_{\geq t} := \{u \mid t \leq^{\mathcal{T}} u\}.$$

Then $\mathcal{T}_{\geq r} = \mathcal{T}$.

$$\begin{aligned} c(t) &:= \max\{|I| \mid I \text{ an independent set of } \mathcal{T}_{\geq t} \text{ containing } t\} \\ n(t) &:= \max\{|I| \mid I \text{ an independent set of } \mathcal{T}_{\geq t} \text{ not containing } t\}. \end{aligned}$$

Then

$$\mathcal{T} \text{ has an independent set of size } k \iff \max\{c(r), n(r)\} \geq k.$$

If t is a leaf, then

$$c(t) = 1 \quad \text{and} \quad n(t) = 0,$$

and if t has the children $t_1, \ldots, t_m$, then:

$$c(t) = 1 + \sum_{i \in [m]} n(t_i) \quad \text{and} \quad n(t) = \sum_{i \in [m]} \max\{c(t_i), n(t_i)\}.$$

2. Tree-decompositions and tree-width.

A tree-decomposition of a graph $\mathcal{G} = (V, E)$ is a pair

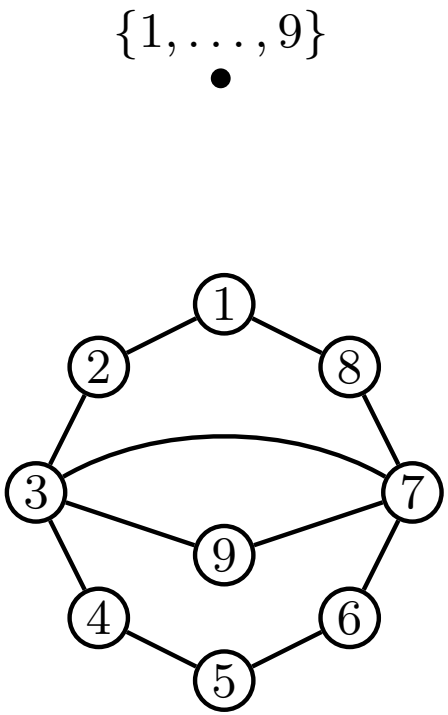
$$(\mathcal{T}, B),$$

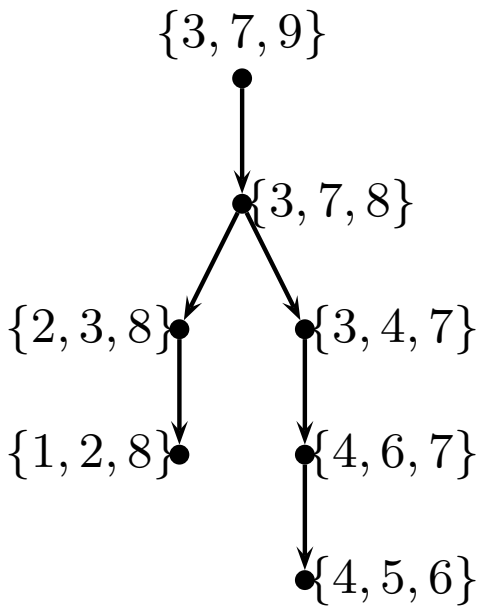
where $\mathcal{T}$ is a tree and $B = (B_t)_{t \in T}$ a family of subsets of V (B_t the bag at t) such that:

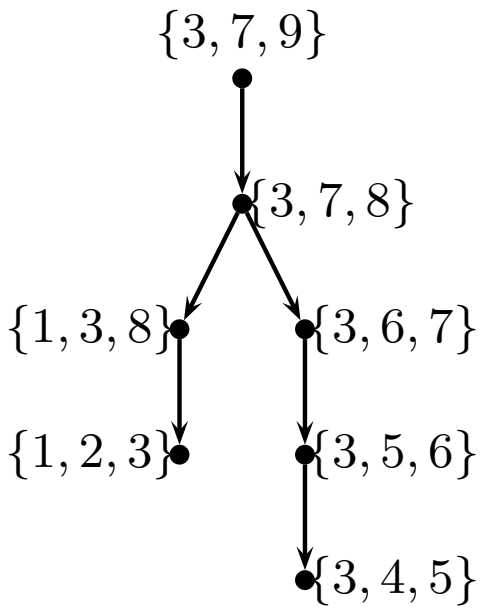
1. For $u \in V$:
$$\{t \in T \mid u \in B_t\} \text{ is nonempty and connected in } \mathcal{T}.$$
2. (Every edge is realized in at least one bag) For $\{u, v\} \in E$ there is $t \in T$ with $u, v \in B_t$.

$$\text{width}\big(\mathcal{T}, (B_t)_{t \in T}\big) := \max\big\{|B_t| \mid t \in T\big\} - 1.$$

$$\text{tw}(\mathcal{G}) := \min\{\text{width}\big(\mathcal{T}, (B_t)_{t \in T}\big) \mid (\mathcal{T}, B) \text{ a tree-decomposition of } \mathcal{G}\}.$$


1
2
8
3
7
9
4
6
5


{1, . . . , 9}
{3, 7, 9}
{3, 7, 8}
{2, 3, 8}
{3, 4, 7}
{1, 2, 8}
{4, 6, 7}
{4, 5, 6}


{3, 7, 9}
{3, 7, 8}
{1, 3, 8}
{3, 6, 7}
{1, 2, 3}
{3, 5, 6}
{3, 4, 5}

1. Every tree with at least one edge has tree-width 1.

2. Every cycle has tree-width 2.

3. A clique of size k has tree-width $k-1$.

4. For $k \geq 1$, the $(k \times k)$-*grid* is the graph

$$\mathcal{G}_{k\times k} := \Big([k] \times [k], \big\{\{(i,j),(i',j')\} \mid |i-i'|+|j-j'|=1\big\}\Big).$$

   The grid $\mathcal{G}_{k\times k}$ has tree-width k.

THEOREM (Excluded Grid Theorem). There is a computable function $w : \mathbb{N} \to \mathbb{N}$ such that the $(k \times k)$-grid is a minor of every graph of tree-width at least $w(k)$.

For $t \in T$: $\quad B_{\geq t} := \bigcup_{t \leq_{\mathcal{T}} t'} B_{t'}$.

LEMMA. Let K be a clique in $\mathcal{G}$ and $(\mathcal{T}, B)$ a tree-decomposition of $\mathcal{G}$. Let $t \in T$. Then

If $t_1, \ldots, t_m$ are the children of t and $K \subseteq B_{\geq t}$, then

$$K \subseteq B_t \text{ or } K \subseteq B_{\geq t_i} \text{ for some } i \in [m].$$

COROLLARY. Let K be a clique in $\mathcal{G}$ and $(\mathcal{T}, B)$ a tree-decomposition of $\mathcal{G}$. There is $t \in T$ with $K \subseteq B_t$.

COROLLARY. Let $\mathcal{G} = (V, E)$ be a clique. Then $\mathrm{tw}(\mathcal{G}) = |V| - 1$.

Let $\mathcal{A}$ be a τ-structure with relational τ.

A tree-decomposition of $\mathcal{A}$ is a pair $(\mathcal{T}, B)$, where $\mathcal{T}$ is a tree and $B = (B_t)_{t \in T}$ a family of subsets of V (B_t the bag at t) such that:

1. For $a \in A$: $\{t \in T \mid u \in B_t\}$ is nonempty and connected in $\mathcal{T}$.
2. (Every "generalized edge" is realized in at least one bag) For $R \in \tau$ and $(a_1, \ldots, a_r) \in R^{\mathcal{A}}$, where $r := \text{arity}(R)$, there is $t \in T$ with $a_1, \ldots, a_r \in B_t$.

$$\text{width}\big(\mathcal{T}, (B_t)_{t \in T}\big) := \max\big\{|B_t| \mid t \in T\big\} - 1.$$

$$\text{tw}(\mathcal{A}) := \min\{\text{width}\big(\mathcal{T}, (B_t)_{t \in T}\big) \mid (\mathcal{T}, B) \text{ a tree-decomposition of } \mathcal{A}\}.$$

The Gaifman graph of a $\mathcal{A}$ is the graph $\mathcal{G}(\mathcal{A}) := (V, E)$, where $V := A$ and

$$E := \big\{\{a, b\} \mid a, b \in A,\ a \neq b, \text{ there exists an } R \in \tau \text{ and } (a_1, \ldots, a_r) \in R^{\mathcal{A}} \text{ such that } a, b \in \{a_1, \ldots, a_r\}\big\}.$$

THEOREM. $\mathcal{A}$ has the same tree-decompositions as $\mathcal{G}(\mathcal{A})$. Hence, $\text{tw}(\mathcal{A}) = \text{tw}(\mathcal{G}(\mathcal{A}))$.

Let $\mathcal{G} = (V, E)$ be a graph.

The incidence structure $\mathcal{G}_I$ of $\mathcal{G}$ is the $\tau_I := \{VERT, EDGE, I\}$-structure (where $VERT, EDGE$ are unary and I binary) with

$$\begin{aligned} G_I &:= V \cup E \\ VERT^{\mathcal{G}_I} &:= V \\ EDGE^{\mathcal{G}_I} &:= E \\ I^{\mathcal{G}_I} &:= \{(u, e) \mid u \in V,\, e \in E,\, u \in e\}. \end{aligned}$$

PROPOSITION. $\mathrm{tw}(\mathcal{G}_I) = \mathrm{tw}(\mathcal{G})$.

3. Dynamic programming on a tree-decomposition.

THEOREM (Bodlaender's Theorem). There is a polynomial p and an algorithm that, given a graph $\mathcal{G} = (V, E)$, computes a tree-decomposition $(\mathcal{T}, B)$ of $\mathcal{G} = (V, E)$, where $\mathcal{T}$ is a binary tree and all B_t are nonempty, of width $\mathrm{tw}(\mathcal{G})$ in time at most

$$2^{p(\mathrm{tw}(\mathcal{G}))} \cdot |V|.$$

COROLLARY. Let $k \geq 1$. The problem

Input:	A graph $\mathcal{G} = (V, E)$.
Question:	Is $\mathrm{tw}(\mathcal{G}) \leq k$?

is solvable in time $O(|V|)$.

COROLLARY. There is a polynomial p and an algorithm that, given a structure $\mathcal{A}$ computes a tree-decomposition $(\mathcal{T}, B)$ of $\mathcal{A}$, where $\mathcal{T}$ is a binary tree and all B_t are nonempty, of width $\mathrm{tw}(\mathcal{A})$ in time at most

$$2^{p(\mathrm{tw}(\mathcal{A}))} \cdot |A| + O(\|\mathcal{A}\|).$$

Moreover, $|T| \in O(|A|)$.

$$\text{GRAPH}_{\leq k} := \{\mathcal{G} \mid \mathcal{G} \text{ a graph and } \text{tw}(\mathcal{G}) \leq k\}.$$

EXAMPLE. 3-COL(GRAPH$_{\leq k}$) is solvable in time $O(|V|)$.

PROOF. Compute a tree-decomposition $(\mathcal{T}, B)$ of $\mathcal{G}$ with Bodlaender's algorithm.

Apply dynamic programming on $(\mathcal{T}, B)$ as follows:

For $t \in T$:

- $\mathrm{Col}(t)$ = set of 3-colorings of B_t.
- $\mathrm{Extcol}(t)$ = set of colorings in $\mathrm{Col}(t)$ that can be extended to $B_{\geq t}$.

Note

$$\mathcal{G} \text{ is 3-colorable} \iff \mathrm{Extcol}(r) \neq \emptyset.$$

Recall: If $t_1, \ldots, t_m$ are the children of t and $\{u, v\} \in E$, then

$$u, v \in B_{\geq t} \text{ implies } u, v \in B_t \text{ or } u, v \in B_{\geq t_i} \text{ for some } i \in [m].$$

Compute $\mathrm{Extcol}(t)$ bottom-up as follows:

- If t is a leaf, then $\mathrm{Extcol}(t) = \mathrm{Col}(t)$.
- If t has children $t_1, \ldots, t_m$, then $\mathrm{Extcol}(t)$ is the set of all $f \in \mathrm{Col}(t)$ such that for all $i \in [m]$ there exists an $f_i \in \mathrm{Extcol}(t_i)$ such that f coincides with f_i on $B_t \cap B_{t_i}$.

Therefore $\mathrm{Extcol}(t)$ can be computed in time $O(m)$ from $\mathrm{Extcol}(t_i)$ for $i \in [m]$. The overall running time (including Bodlaender's algorithm) is thus $O(|V|)$.

4. Dynamic programming and automata on trees.

Let $\tau_b := \{E_1, E_2\}$ with binary E_1 (the first child relation) and E_2 (the second child relation).

A τ_b-structure $\mathcal{T} = (T, E_1^{\mathcal{T}}, E_2^{\mathcal{T}})$ is an ordered binary tree if

- $(T, E_1^{\mathcal{T}} \cup E_2^{\mathcal{T}})$ is a directed binary tree (with edges directed from the root to the leaves);
- $E_1^{\mathcal{T}} \cap E_2^{\mathcal{T}} = \emptyset$;
- $\mathcal{T} \models \forall x \forall y (E_2 xy \rightarrow \exists z E_1 xz)$.

Let Σ be a (finite) alphabet. A Σ-tree is a pair $(\mathcal{T}, \lambda)$, where

- $\mathcal{T}$ is an ordered binary tree;
- $\lambda : T \rightarrow \Sigma$.

Let Σ-TREE denote the class of Σ-trees.

A tree automaton is a tuple $\mathbb{A} = (S, \Sigma, \delta, F)$, where

- S is a finite set of states,
- Σ is a finite alphabet,
- $\delta : (S^2 \cup S \cup \{\text{leaf}\}) \times \Sigma \to S$ is the transition relation.
- $F \subseteq S$ is the set of accepting states.

The run $\rho_{\mathbb{A}}$ of $\mathbb{A} = (S, \Sigma, \delta, F)$ on a Σ-tree $(\mathcal{T}, \lambda)$ is the mapping $\rho_{\mathbb{A}} : T \to S$ such that for all $t \in T$:

- If t is a leaf, $\rho_{\mathbb{A}}(t) = \delta(\text{leaf}, \lambda(t))$.
- If t only has one child t_1, then $\rho_{\mathbb{A}}(t) = \delta(\rho_{\mathbb{A}}(t_1), \lambda(t))$.
- If t has two children t_1, t_2, then $\rho_{\mathbb{A}}(t) = \delta(\rho_{\mathbb{A}}(t_1), \rho_{\mathbb{A}}(t_2), \lambda(t))$.

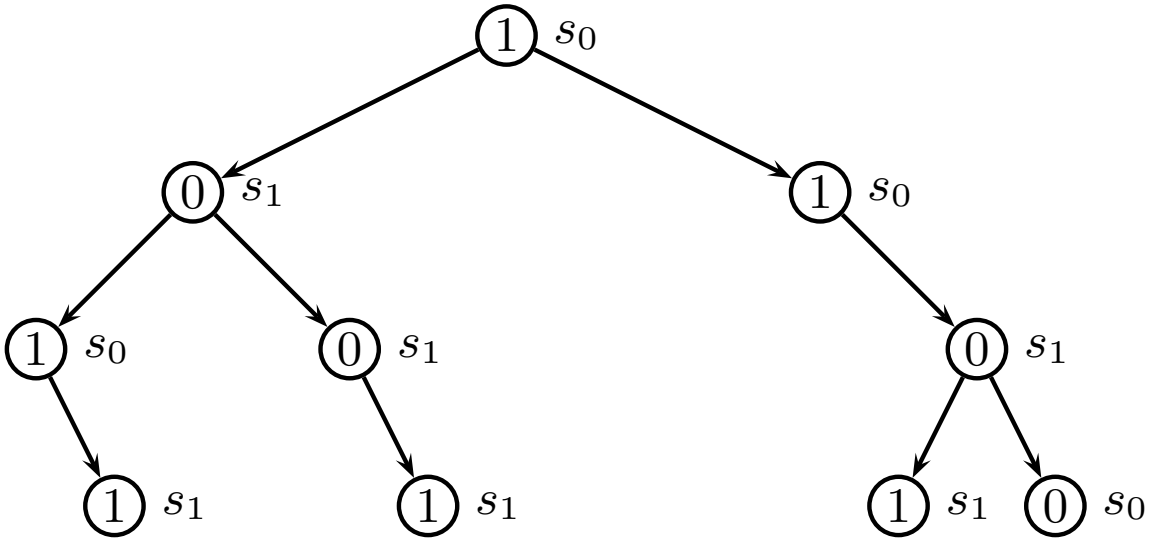
$\mathbb{A}$ accepts $(\mathcal{T}, \lambda)$ if $\rho_{\mathbb{A}}(r^{\mathcal{T}}) \in F$.

$$L(\mathbb{A}) := \{(\mathcal{T}, \lambda) \mid \mathbb{A} \text{ accepts } (\mathcal{T}, \lambda)\},$$

the tree language recognized by $\mathbb{A}$.

EXAMPLE. Let $\mathbb{A} = (\{s_0, s_1\}, \{0, 1\}, \delta, \{s_0\})$, where

$$\delta(\text{leaf}, i) = s_i$$
$$\delta(s_i, j) = s_k \quad \text{where } i + i + j \equiv k \pmod 2$$
$$\delta(s_i, s_j, k) = s_\ell \quad \text{where } i + j + k \equiv \ell \pmod 2$$



Let $\mathbb{A} = (S, \Sigma, \delta, F)$ be a tree automaton. Define the binary relation $\sim_{\mathbb{A}}$ on Σ-TREE by

$$(\mathcal{T}_1, \lambda_1) \sim_{\mathbb{A}} (\mathcal{T}_2, \lambda_2) \iff \rho_{\mathbb{A}}(r^{\mathcal{T}_1}) = \rho_{\mathbb{A}}(r^{\mathcal{T}_2}).$$

LEMMA. (a) $\sim_{\mathbb{A}}$ is an equivalence relation of finite index ($\leq |S|$ many equivalence classes).

(b) $L(\mathbb{A})$ is the union of equivalence classes:

$$L(\mathbb{A}) = \bigcup_{\rho_{\mathbb{A}}(r^{\mathcal{T}}) \in F} [(\mathcal{T}, \lambda)].$$

(c) $\sim_{\mathbb{A}}$ is root-invariant.

PROPOSITION. Let $\sim$ be a root-invariant equivalence relation of finite index and let $L \subseteq \Sigma$-TREE be the union of $\sim$-equivalence classes. Then there is an automaton $\mathbb{A}$ such that

$$L = L(\mathbb{A}).$$

5. Automata on trees and monadic second-order logic.

MSO$[\tau]$, the set of formulas of vocabulary τ of monadic second-order logic

$x, y, x_1, \ldots$	variables ranging over elements of the structure;
$X, Y, X_1, \ldots$	variables ranging over subsets of the structure
$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$	the connectives;
$\forall, \exists$	the quantifiers;
$=$	the equality sign;
R	$R \in \tau$;
c	$c \in \tau$;
$), ($	the parentheses.

$$\overbrace{\exists X_1 \exists X_2 \exists X_3}^{\text{“}X_i \text{ is the set of elements of color } i.\text{”}} \; \forall x \forall y \Big(\overbrace{\bigvee_{i \in [3]} X_i x \wedge \bigwedge_{1 \le i < j \le 3} \neg (X_i x \wedge X_j x)}^{\text{“Each element has exactly one color.”}} \wedge \underbrace{\bigwedge_{i \in [3]} (Exy \rightarrow \neg (X_i x \wedge X_i y))}_{\text{“Adjacent elements do not have the same color.”}} \Big).$$

Let $\tau_b = \{E_1, E_2\}$ be the vocabulary of rooted binary trees. Fix an alphabet Σ and set

$$\tau(\Sigma) := \tau_b \cup \{P_a \mid a \in \Sigma\},$$

where each P_a is unary.

Then every Σ-tree $(\mathcal{T}, \lambda)$ "is" a $\tau(\Sigma)$-structure.

For an MSO$[\tau(\Sigma)]$-sentence φ let

$$L(\varphi) := \{(\mathcal{T}, \lambda) \in \Sigma\text{-TREE} \mid (\mathcal{T}, \lambda) \models \varphi\}.$$

THEOREM. For $L \subseteq \Sigma$-TREE the following are equivalent:

(i) $L = L(\mathbb{A})$ for some automaton $\mathbb{A}$.

(ii) $L = L(\varphi)$ for some MSO-sentence φ.

PROOF. (i) $\Rightarrow$ (ii): Let $\mathbb{A} = (S, \Sigma, \delta, F)$ be an automaton. Then $L(\mathbb{A}) = L(\varphi)$, where

$$\varphi := \exists X_1 \ldots \exists X_m \Big(\mathit{unique} \wedge \mathit{leaf} \wedge \mathit{one\text{-}child} \wedge \mathit{two\text{-}child} \wedge \mathit{acc} \Big),$$

- $\mathit{unique} := \forall x \Big(\bigvee_{s \in [m]} X_s x \wedge \bigwedge_{1 \leq s < s' \leq m} (\neg X_s x \vee \neg X_{s'} x) \Big)$.
- $\mathit{leaf} := \forall x \big(\forall y \, \neg E_1 xy \rightarrow \bigvee_{a \in \Sigma} (P_a x \wedge X_{\delta(\mathrm{leaf}, a)}) \big)$.
- $\mathit{one\text{-}child} := \forall x \Big(\forall y \, \neg E_2 xy \rightarrow (\forall y (E_1 xy \rightarrow \bigvee_{s \in S, \, a \in \Sigma} (X_s y \wedge P_a x \wedge X_{\delta(s,a)} x) \Big)$.
- $\mathit{two\text{-}child} := \forall x \forall y \forall z \Big((E_1 xy \wedge E_2 xz) \rightarrow \bigvee_{s_1, s_2 \in S, \, a \in \Sigma} (X_{s_1} y \wedge X_{s_2} z \wedge P_a x \wedge X_{\delta(s_1, s_2, a)} x) \Big)$.
- $\mathit{acc} := \forall x \Big(\forall z \, \neg E_1 zx \rightarrow \bigvee_{s \in F} X_s x \Big)$.

6. Courcelle's Theorem.

THEOREM (Courcelle's Theorem) Let τ be a vocabulary. There is a binary computable function f and an algorithm that solves the problem

Input:	A τ-structure $\mathcal{A}$ and an MSO-sentence φ.
Question:	Is $\mathcal{A}$ a model of φ?

in time

$$f(|\varphi|, \mathrm{tw}(\mathcal{A})) \cdot |A| + O(\|A\|).$$

COROLLARY. Let τ be a vocabulary, $k \in \mathbb{N}$ and φ an MSO-sentence. There is a an algorithm that solves the problem

Input:	A τ-structure $\mathcal{A}$ of tree-width $\leq k$.
Question:	Is $\mathcal{A}$ a model of φ?

in time

$$O(\|A\|).$$

MSO-Ehrenfeucht-game $M_m(\mathcal{A}, \mathcal{B})$, where $\mathcal{A}$, $\mathcal{B}$ are τ-structures.

Two players: Spoiler und Duplicator. Each player has to make m moves; the player take turns; Spoiler starts.

In his ith move Spoiler decides, whether to make a point move or a set move.

- In a point move, Spoiler first chooses a structure and then an element of its universe.
  - If Spoiler chooses an element in $\mathcal{A}$, then Duplicator one in $\mathcal{B}$;
  - If Spoiler chooses an element in $\mathcal{B}$, then Duplicator one in $\mathcal{A}$.
- In a set move, Spoiler first chooses a structure and then a subset of its universe.
  - If Spoiler chooses a subset in $\mathcal{A}$, then Duplicator one in $\mathcal{B}$;
  - If Spoiler chooses a subset in $\mathcal{B}$, then Duplicator one in $\mathcal{A}$.

At the end, elements $\bar{a} = a_1, \ldots, a_e$ in $\mathcal{A}$, $\bar{b} = b_1, \ldots, b_e$ in $\mathcal{B}$, and subsets $P_1, \ldots, P_{e'}$ of A und $Q_1, \ldots, Q_{e'}$ of B with $e + e' = m$ have been chosen. Duplicator wins, if

$$\bar{a} \mapsto \bar{b} \in \mathrm{Part}((\mathcal{A}, P_1, \ldots, P_{e'}), (\mathcal{B}, Q_1, \ldots, Q_{e'})).$$

$\mathcal{A} \equiv_m^{\mathrm{MSO}} \mathcal{B}$: $\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences of MSO of quantifier-rank $\leq m$.

Ehrenfeucht's Theorem for MSO. Let $\mathcal{A}$, $\mathcal{B}$ be τ-structures and $m \geq 0$. Then

Duplicator has a winning strategy for $M_m(\mathcal{A}, \mathcal{B}) \iff \mathcal{A} \equiv_m^{\mathrm{MSO}} \mathcal{B}$.

Let τ be relational and $(\mathcal{B}_i)_{i\in I}$ a family of τ-structures. Let

$$\bigcup_{i\in I} \mathcal{B}_i := (\bigcup B_i, (\bigcup_{i\in I} R^{\mathcal{B}_i})_{R\in\tau}).$$

LEMMA. Let $\mathcal{A}$, $\mathcal{A}'$, $\mathcal{B}$ and $\mathcal{B}'$ be τ-structures with $A \cap A' = B \cap B' = \emptyset$ and relational τ. If Dulicator wins $M_n(\mathcal{A}, \mathcal{B})$ and Dulicator wins $M_n(\mathcal{A}', \mathcal{B}')$, then Dulicator wins $M_n(\mathcal{A} \cup \mathcal{A}', \mathcal{B} \cup \mathcal{B}')$.

LEMMA. Let $\mathcal{A}$, $\mathcal{A}'$, $\mathcal{B}$ and $\mathcal{B}'$ be τ-structures, $\bar{a} \in A$, $\bar{a}' \in A'$, $\bar{b} \in B$, $\bar{b}' \in B'$. Assume

$$A \cap A' \subseteq \{a_1, \ldots, a_k\} \cap \{a'_1, \ldots, a'_\ell\}$$

$$B \cap B' \subseteq \{b_1, \ldots, b_k\} \cap \{b'_1, \ldots, b'_\ell\}$$

$$\text{for } i \in [k],\ j \in [\ell]: \qquad (a_i = a'_j \iff b_i = b'_j).$$

If Dulicator wins $M_n(\mathcal{A}, \bar{a}), (\mathcal{B}, \bar{b}))$ and Dulicator wins $M_n((\mathcal{A}', \bar{a}'), (\mathcal{B}', \bar{b}'))$, then Dulicator wins $M_n((\mathcal{A} \cup \mathcal{A}', \bar{a}\bar{a}'), (\mathcal{B} \cup \mathcal{B}', \bar{b}\bar{b}'))$.

Let τ be relational and $(\mathcal{B}_i)_{i\in I}$ a family of τ-structures with $\mathcal{B}_i \subseteq \mathcal{A}$. Then

$$\bigcup_{i\in I} \mathcal{B}_i \subseteq \mathcal{A} \quad \Longleftrightarrow \quad \begin{array}{l} \text{for all } R \in \tau, \text{ all } (a_1, \ldots, a_r) \in R^{\mathcal{A}}: \\ \text{there is an } i \in I \text{ such that } (a_1, \ldots, a_r) \in R^{\mathcal{B}_i}. \end{array}$$

Let $(\mathcal{T}, B)$ be a tree-decomposition of $\mathcal{A}$ and $t \in T$. We set

$$\mathcal{B}_t := [B_t]^{\mathcal{A}} \qquad \text{and} \qquad \mathcal{B}_{\geq t} := [B_{\geq t}]^{\mathcal{A}}.$$

COROLLARY. Let $(\mathcal{T}, B)$ be a tree-decomposition of $\mathcal{A}$ and let $t \in T$ have the children $t_1, \ldots, t_m$. Then

$$\mathcal{B}_{\geq t} = \mathcal{B}_t \cup \mathcal{B}_{\geq t_1} \cup \ldots \cup \mathcal{B}_{\geq t_m}.$$

Fix a relational τ. For $k \in \mathbb{N}$ let

$$\tau_k := \tau \cup \{c_1, \ldots, c_k\}.$$

LEMMA. There is an algorithm that given m, k computes a finite set $\Phi(m,k)$ of MSO$[\tau_k]$-sentences such that:

- For all $\varphi \in \Phi(m,k)$: $\mathrm{qr}(\varphi) \leq m$.
- For all MSO$[\tau_k]$-sentences ψ with $\mathrm{qr}(\psi) \leq m$ there is $\varphi \in \Phi(m,k)$ such that $\models \varphi \leftrightarrow \psi$.

Let

$$\mathrm{BS}(m,k) := \{(\mathcal{A}, \bar{a}) \mid (\mathcal{A}, \bar{a}) \text{ a } \tau_k\text{-structure with } \mathcal{A} = ([\ell], \ldots) \text{ for some } \ell \in [k] \text{ and } [\ell] = \{\bar{a}\}\}$$

be the class of basic m,k-structures.

Let $M_0(m,k) := \{(m\text{-Th}((\mathcal{A},\bar{a})),(\mathcal{A},\bar{a})) \mid (\mathcal{A},\bar{a}) \in \text{BS}(m,k)\}$.

Let $M(m,k)$ be the closure of $M_0(m,k)$ under the operations:

- Given $(\mathcal{A},\bar{a}) \in \text{BS}(m,k)$, $\text{Eq}_1 \subseteq [k]\times[k]$ and $(\Phi_1,(\mathcal{B}_1,\bar{b}_1)) \in M(m,k)$, build if possible (this can be decided effectively) a copy $(\mathcal{B}_1',\bar{b}_1')$ of $(\mathcal{B}_1,\bar{b}_1)$ realizing with respect to $(\mathcal{A},\bar{a})$ the equalities in Eq_1. Add

$$(m\text{-Th}((\mathcal{A}\cup\mathcal{B}_1',\bar{a})),(\mathcal{A}\cup\mathcal{B}_1',\bar{a}))$$

  to $M(m,k)$ if no member in $M(m,k)$ has m-Th$((\mathcal{A}\cup\mathcal{B}_1',\bar{a}))$ as first component;

- Given $(\mathcal{A},\bar{a}) \in \text{BS}(m,k)$, $\text{Eq}_1,\text{Eq}_2 \subseteq [k]\times[k]$ and $(\Phi_1,(\mathcal{B}_1,\bar{b}_1)) \in M(m,k)$ and $(\Phi_2,(\mathcal{B}_2,\bar{b}_2)) \in M(m,k)$, build if possible copies $(\mathcal{B}_1',\bar{b}_1')$ and $(\mathcal{B}_2',\bar{b}_2')$ of $(\mathcal{B}_1,\bar{b}_1)$ and $(\mathcal{B}_2,\bar{b}_2)$, realizing with respect to $(\mathcal{A},\bar{a})$ the equalities in Eq_1 and Eq_2, respectively. Add

$$(m\text{-Th}((\mathcal{A}\cup\mathcal{B}_1'\cup\mathcal{B}_2',\bar{a})),(\mathcal{A}\cup\mathcal{B}_1'\cup\mathcal{B}_2',\bar{a}))$$

  to $M(m,k)$ if no member in $M(m,k)$ has m-Th$((\mathcal{A}\cup\mathcal{B}_1'\cup\mathcal{B}_2',\bar{a}))$ as first component.

$$S(m,k) := \{\Phi \mid \text{there is } \ldots \text{such that } (\Phi, \ldots) \in M(m,k)\}$$

$$\Sigma(m,k) := \{\Phi \mid \text{there is } \ldots \text{such that } (\Phi, \ldots) \in M_0(m,k)\} \times ([k] \times [k])^2$$

Note that $\ldots$ is always uniquely determined.

$\mathbb{A}(m,k,\varphi)$:

$$\delta(\text{leaf}, (\Phi, \text{Eq}_1, \text{Eq}_2)) := \Phi$$

$$\delta(\Phi_1, (\Phi, \text{Eq}_1, \text{Eq}_2)) := \Phi'$$

where Φ' is determined from Φ, Eq_1 and Φ_1 according to the first closure condition of $M(m,k)$

$$\delta(\Phi_1, \Phi_2, (\Phi, \text{Eq}_1, \text{Eq}_2)) := \Phi'$$

where Φ' is determined from $\Phi, \text{Eq}_1, \text{Eq}_2$ and Φ_1, Φ_2 according to the second closure condition of $M(m,k)$

$$F := \{\Phi \in S(m,k) \mid \text{there is a } \ldots \text{such that } (\Phi, \ldots) \in M(m,k) \text{ and } \ldots \models \varphi\}.$$

7. Applications of Courcelle's Theorem.

A kernel in a directed graph $\mathcal{G} = (V, E)$ is a subset K of V such that

- no two vertices in K are adjacent;
- for every vertex $a \in V \setminus K$ there is a vertex $b \in K$ such that $(a, b) \in E$.

THEOREM. Let $k \geq 1$. There is algorithm solving in linear time the problem

Input:	A directed graph $\mathcal{G}$ of tree-width $\leq k$.
Question:	Decide whether $\mathcal{G}$ has a kernel?

THEOREM. Let $k \geq 1$. HAM(GRAPH$_{\leq k}$) is solvable in linear time.

There is an MSO-sentence *hamiltonian* such that for every graph $\mathcal{G}$:

$$\mathcal{G}_I \models \mathit{hamiltonian} \iff \mathcal{G} \text{ is hamiltonian.}$$

The formula

$$\mathit{cycle\text{-}cover}(Y) := \forall y(Yy \to \mathit{EDGE}y) \wedge \forall x\big(\mathit{VERT}x \to \exists^{=2} y(Yy \wedge Ixy)\big)$$

states that Y is the edge set of a family of disjoint cycles that covers all vertices of a graph.

$$\mathit{adj}(y,z) := \exists x(Ixy \wedge Ixz).$$

$$\overbrace{\forall y(Yy \to \mathit{EDGE}y)}^{Y \text{ is a set of edges}} \wedge \neg\exists Z\Big(\overbrace{\forall y(Zy \to Yy) \wedge \exists y(Yy \wedge \neg Zy) \wedge \exists y Zy}^{Z \text{ is a nonempty proper subset of } Y} \wedge \underbrace{\forall y\forall z\big((Zy \wedge \mathit{adj}(y,z) \wedge Yz) \to Zz\big)}_{Z \text{ is closed under adjacency}}\Big).$$

express that Y is a connected set of edges

$$\mathit{hamiltonian} := \exists Y\big(\mathit{cycle\text{-}cover}(Y) \wedge \mathit{edge\text{-}conn}(Y)\big).$$

8. Extensions of Courcelle's Theorem.

THEOREM. Let τ be a vocabulary. There is a binary computable function f and an algorithm that solves the problem

Input:	A τ-structure $\mathcal{A}$, an MSO-formula $\varphi(X)$, and $\ell \leq \|A\|$.
Question:	Is there an $S \subseteq A$ with $\|S\| \geq \ell$ such that $\mathcal{A} \models \varphi(S)$?

in time

$$f(|\varphi|, \mathrm{tw}(\mathcal{A})) \cdot |A| + O(\|A\|).$$

COROLLARY. Let τ be a vocabulary. Let $k \in \mathbb{N}$ and $\varphi(X)$ an MSO-formula There is an algorithm that solves the problem

Input:	A τ-structure $\mathcal{A}$ and $\ell \leq \|A\|$.
Question:	Is there an $S \subseteq A$ with $\|S\| \geq \ell$ such that $\mathcal{A} \models \varphi(S)$?

in time

$$O(\|A\|).$$

A graph $\mathcal{H}$ is a minor of a graph $\mathcal{G}$ if $\mathcal{H}$ can be obtained from a subgraph of $\mathcal{G}$ by contracting edges.

THEOREM. Let C be a class of graphs closed under minors with $C \neq$ GRAPH. Then there is a computable function f and an algorthm soving the problem

Input:	$\mathcal{G} = (V, E) \in$ C and a sentence φ of first-order logic.
Question:	Is $\mathcal{G}$ a model of φ?

in time

$$f(|\varphi|) \cdot |V|^{O(1)}.$$