Lecture 3: Algorithmic Meta-Theorems

(A Short Introduction to Parameterized Complexity)

Clemens Grabmayer

Ph.D. Program, Advanced Courses Period
Gran Sasso Science Institute
L'Aquila, Italy

June 12, 2024

Course overview

methods/techniques/design patterns for FPT results

using logic for obtaining FPT results

obtaining ∈ FPT results

Monday, June 10 10.30 – 12.30	Tuesday, June 11	Wednesday, June 12 10.30 – 12.30	Thursday, June 13	Friday, June 14
Introduction & basic FPT results		**Algorithmic Meta-Theorems**		
motivation for FPT kernelization, Crown Lemma, Sunflower Lemma	GDA	1st-order logic, monadic 2nd-order logic, FPT-results by Courcelle's Theorems for tree and clique-width	GDA	GDA
Algorithmic Techniques		*Formal-Method & Algorithmic Techniques*		
	14.30 – 16.30			14.30 – 16.30
	Notions of bounded graph width			**FPT-Intractability Classes & Hierarchies**
	path-, tree-, clique width, FPT-results by dynamic programming, transferring FPT results betw. widths	GDA	GDA	motivation for FP-intractability results, FPT-reductions, class XP (slicewise polynomial), W- and A-Hierarchies, placing problems on these hierarchies

Overview

- Courcelle's theorem
  - FPT-results by model-checking MSO-formulas
    - for graphs / structures with bounded tree-width
    - for maximization problems over graphs of bounded tree-width
    - for graphs of bounded clique-width
  - applications to concrete problems

Overview

- logic preliminaries
  - first-order logic
    - expressing graph problems by f-o formulas
  - monadic second-order logic (MSO)
    - expressing graph problems by MSO formulas
  - complexity of evaluation and model checking problems
- Courcelle's theorem
  - FPT-results by model-checking MSO-formulas
    - for graphs / structures with bounded tree-width
    - for maximization problems over graphs of bounded tree-width
    - for graphs of bounded clique-width
  - applications to concrete problems

Meta-theorems: idea, benefits and limitations

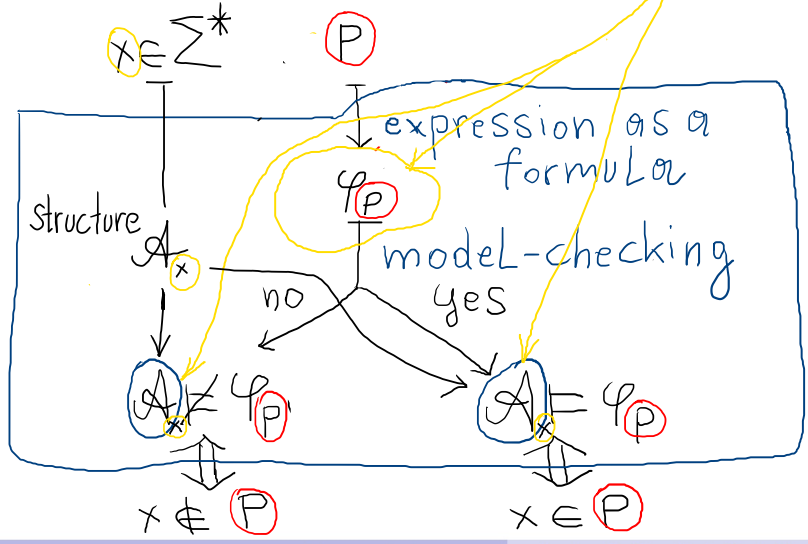
idea:

- express a problem P by a logical formula φ_P (of 'short' size)
- use model checking of φ_P
  on logical structures of bounded width k (tree-, clique-width, . . .)
  - is time bounded depending on k, size of φ_P, size of the structure
  - this often facilitates FPT-results

Meta-theorems: idea, benefits and limitations

idea:

- express a problem P by a logical formula φ_P (of 'short' size)
- use model checking of φ_P
  on logical structures of bounded width k (tree-, clique-width, ...)
  - is time bounded depending on k, size of φ_P, size of the structure
  - this often facilitates FPT-results



$x \in P \Leftrightarrow A_x \models \varphi_P$

size/length $|\varphi_P|$ of formular (should be short) influences obtained Complexity result

Meta-theorems: idea, benefits and limitations

idea:

- express a problem P by a logical formula φ_P (of 'short' size)
- use model checking of φ_P
  on logical structures of bounded width k (tree-, clique-width, . . .)
  - is time bounded depending on k, size of φ_P, size of the structure
  - this often facilitates FPT-results

benefits:

- a quick and easy way to show
  that [some problems] are fixed-parameter tractable,
- without working out the tedious details
  of a dynamic programming algorithm.

Meta-theorems: idea, benefits and limitations

idea:

- express a problem P by a logical formula φ_P (of 'short' size)
- use model checking of φ_P
  on logical structures of bounded width k (tree-, clique-width, . . .)
  - is time bounded depending on k, size of φ_P, size of the structure
  - this often facilitates FPT-results

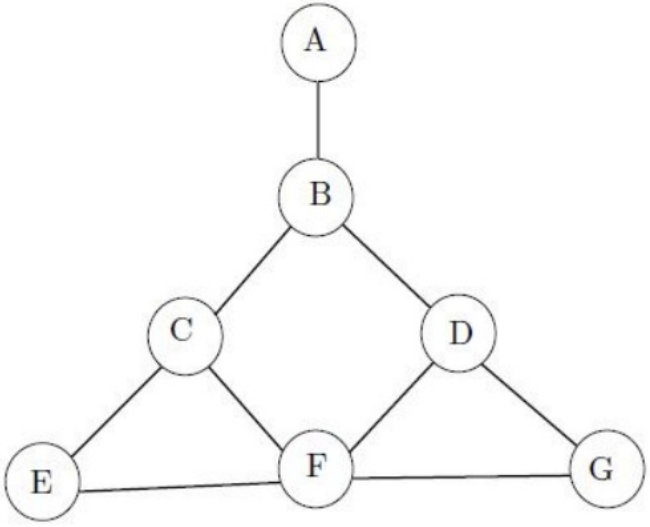
benefits:

- a quick and easy way to show
  that [some problems] are fixed-parameter tractable,
- without working out the tedious details
  of a dynamic programming algorithm.

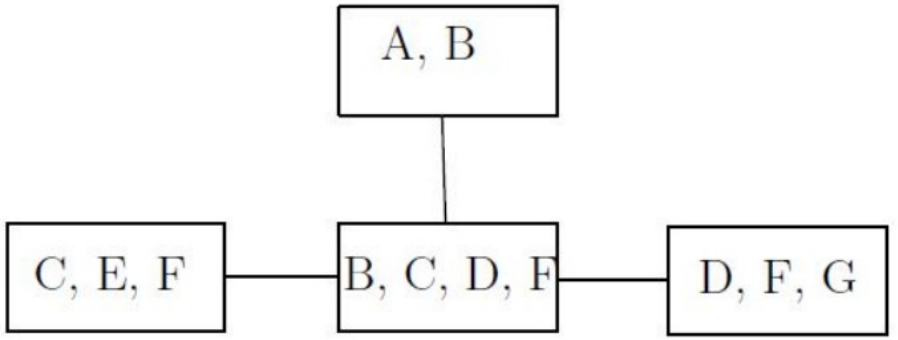
limitations:

- algorithms obtained by meta-theorems
  cannot be expected to be optimal.
- a careful analysis of a specific problem at hand
  will usually yield more efficient fpt-algorithms

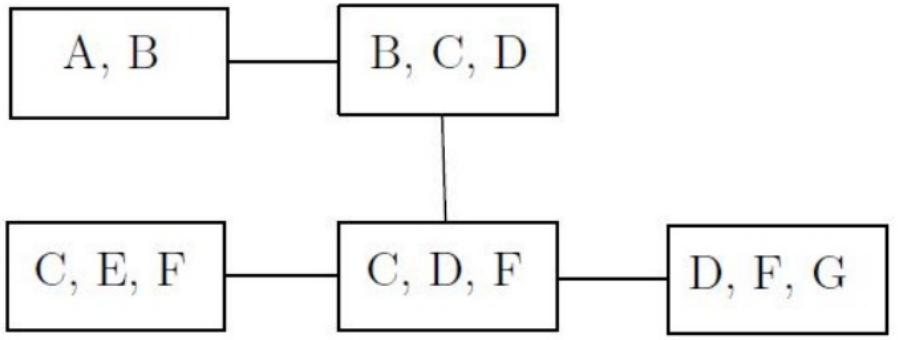
Tree decomposition (example)



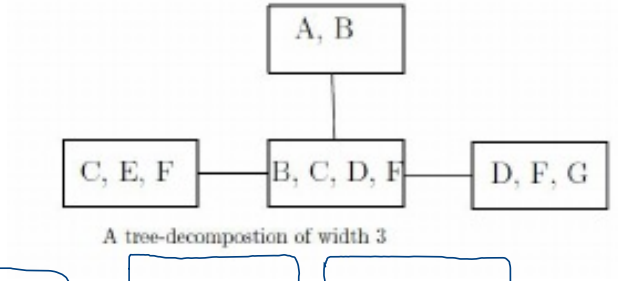
The original graph G



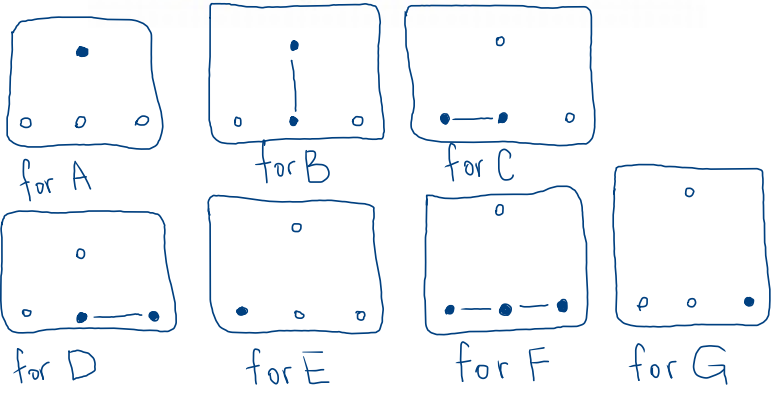
A tree-decompostion of width 3

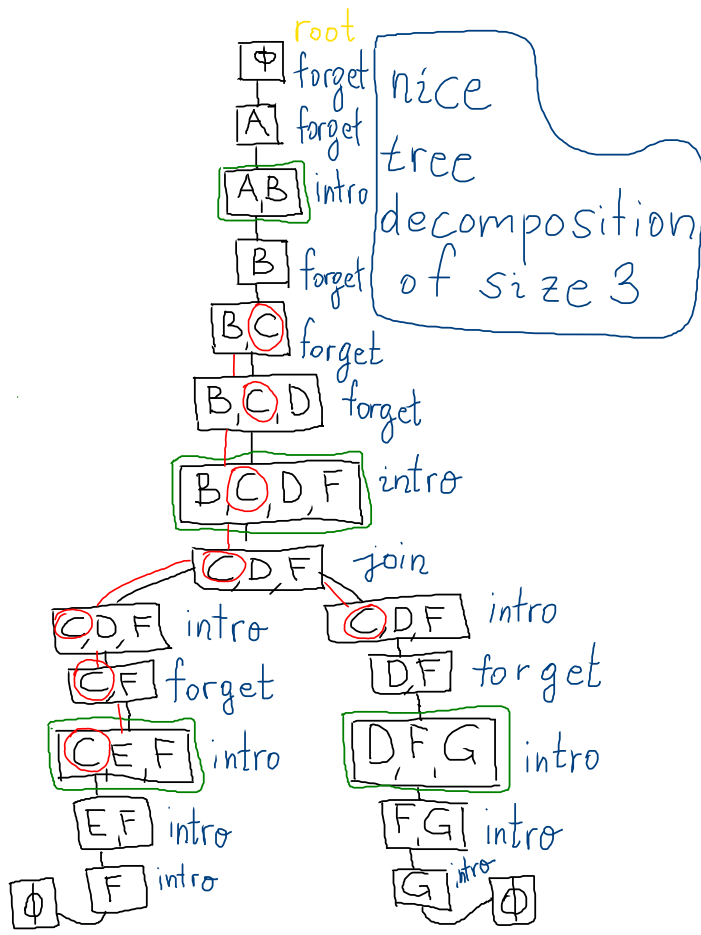


A tree-decompostion of width 2



A tree-decompostion of width 3





Tree decompositions, and tree-width

Definition (Bertelé–Brioschi, 1972, Halin, 1976, Robertson–Seymour, 1984)

A *tree decomposition* of a graph $\mathcal{G} = \langle V, E \rangle$ is a pair $\langle \mathcal{T}, \{B_t\}_{t \in T} \rangle$ where $\mathcal{T} = \langle T, F \rangle$ a (undirected, unrooted) tree, and $B_t \subseteq V$ such that:

(T1) $V = \bigcup_{t \in T} B_t$ (every vertex of $\mathcal{G}$ is in some bag).

(T2) $(\forall \{u, v\} \in E)\,(\exists t \in T)\big[\, \{u, v\} \subseteq B_t \,\big]$
(the vertices of every edge of $\mathcal{G}$ are realized in some bag).

(T3) $(\forall v \in V)\,\big[$ subgraph of $\mathcal{T}$ defd. by $\{t \in T \mid v \in B_t\}$ is connected $\big]$
(the tree vertices (in $\mathcal{T}$) whose bags contain some vertex of $\mathcal{G}$ induce a subgraph of $\mathcal{T}$ that is connected).

The *width* of a tree decomposition $\langle \mathcal{T}, \{B_t\}_{t \in T} \rangle$ is

$$\max \{ |B_t| - 1 \mid t \in T \}.$$

The *tree-width* $tw(\mathcal{G})$ of a graph $\mathcal{G} = \langle V, E \rangle$ is defined by:

$tw(\mathcal{G})$:= minimal width of a tree decomposition of $\mathcal{G}$.

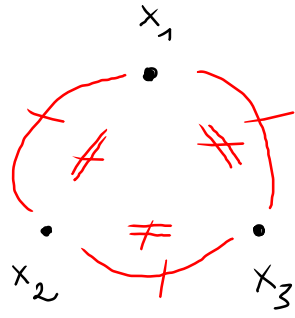
Logical preliminaries

First-order logic (formula example)

$$\varphi_3 := \exists x_1 \exists x_2 \exists x_3 \big(\neg(x_1 = x_2) \wedge \neg E(x_1, x_2) \wedge \neg(x_1 = x_3) \wedge \neg E(x_1, x_3) \wedge \neg(x_2 = x_3) \wedge \neg E(x_2, x_3) \big)$$

First-order logic (formula example)

$$\varphi_3 := \exists x_1 \exists x_2 \exists x_3 \big(\neg(x_1 = x_2) \wedge \neg E(x_1, x_2) \wedge \neg(x_1 = x_3) \wedge \neg E(x_1, x_3) \wedge \neg(x_2 = x_3) \wedge \neg E(x_2, x_3) \big)$$



First-order logic (formula example)

$$\varphi_3 := \exists x_1 \exists x_2 \exists x_3 \big(\neg(x_1 = x_2) \wedge \neg E(x_1, x_2) \wedge \neg(x_1 = x_3) \wedge \neg E(x_1, x_3) \wedge \neg(x_2 = x_3) \wedge \neg E(x_2, x_3) \big)$$

$$\mathcal{A}(\mathcal{G}) \models \varphi_3 \iff \mathcal{G} \text{ has a } 3\text{-element independent set}.$$

$$S \subseteq V \text{ is independent set in } \mathcal{G} = \langle V, E \rangle : \iff \forall e = \{u, v\} \in E \, (\neg(u \in S \wedge v \in S))$$
$$\iff \forall u, v \in S (u \neq v \Rightarrow \{u, v\} \notin E)$$

First-order logic (formula example)

$$\varphi_3 := \exists x_1 \exists x_2 \exists x_3 \big(\neg(x_1 = x_2) \wedge \neg E(x_1, x_2) \wedge \neg(x_1 = x_3) \wedge \neg E(x_1, x_3) \wedge \neg(x_2 = x_3) \wedge \neg E(x_2, x_3) \big)$$

$$\mathcal{A}(\mathcal{G}) \models \varphi_3 \iff \mathcal{G} \text{ has a } 3\text{-element independent set}.$$

$$\varphi_k := \exists x_1 \ldots \exists x_k \Big(\bigwedge_{1 \le i < j \le k} \big(\neg(x_i = x_j) \wedge \neg E(x_i, x_j) \big) \Big)$$

$S \subseteq V$ is independent set in $\mathcal{G} = \langle V, E \rangle$:$\iff$ $\forall e = \{u, v\} \in E \, (\neg(u \in S \wedge v \in S))$

$\iff$ $\forall u, v \in S (u \neq v \Rightarrow \{u, v\} \notin E)$

First-order logic (formula example)

$$\varphi_3 := \exists x_1 \exists x_2 \exists x_3 \big(\neg(x_1 = x_2) \wedge \neg E(x_1, x_2) \wedge \neg(x_1 = x_3) \wedge \neg E(x_1, x_3) \wedge \neg(x_2 = x_3) \wedge \neg E(x_2, x_3) \big)$$

$$\mathcal{A}(\mathcal{G}) \models \varphi_3 \iff \mathcal{G} \text{ has a } 3\text{-element independent set.}$$

$$\varphi_k := \exists x_1 \ldots \exists x_k \Big(\bigwedge_{1 \le i < j \le k} \big(\neg(x_i = x_j) \wedge \neg E(x_i, x_j) \big) \Big)$$

$\neg(x_i = x_j)$: x_i, x_j are distinct; $\neg E(x_i, x_j)$: no edge between x_i and x_j

$\bigwedge_{1 \le i < j \le k} \big(\neg(x_i = x_j) \wedge \neg E(x_i, x_j) \big)$: $x_1, \ldots, x_k$ distinct and no edge between them

$S \subseteq V$ is independent set in $\mathcal{G} = \langle V, E \rangle :\iff \forall e = \{u, v\} \in E \, (\neg(u \in S \wedge v \in S))$
$\iff \forall u, v \in S (u \neq v \Rightarrow \{u, v\} \notin E)$

First-order logic (formula example)

$$\varphi_3 := \exists x_1 \exists x_2 \exists x_3 \big(\neg(x_1 = x_2) \wedge \neg E(x_1, x_2) \wedge \neg(x_1 = x_3) \wedge \neg E(x_1, x_3) \wedge \neg(x_2 = x_3) \wedge \neg E(x_2, x_3) \big)$$

$$\mathcal{A}(\mathcal{G}) \models \varphi_3 \iff \mathcal{G} \text{ has a } 3\text{-element independent set}.$$

$$\varphi_k := \exists x_1 \ldots \exists x_k \Big(\bigwedge_{1 \le i < j \le k} \big(\neg(x_i = x_j) \wedge \neg E(x_i, x_j) \big) \Big)$$

$$\mathcal{A}(\mathcal{G}) \models \varphi_k \iff \mathcal{G} \text{ has a } k\text{-element independent set}.$$

$S \subseteq V$ is independent set in $\mathcal{G} = \langle V, E \rangle$:$\iff \forall e = \{u, v\} \in E \, (\neg(u \in S \wedge v \in S))$

$\iff \forall u, v \in S (u \neq v \Rightarrow \{u, v\} \notin E)$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*(R_i) $\in \mathbb{N}$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity $ar(R_i) \in \mathbb{N}$
  - the binary equality predication $=$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - (first-order) variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - (first-order) variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$
  - propositional connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - (first-order) variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$
  - propositional connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - existential quantifier $\exists$, universal quantifier $\forall$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - (first-order) variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$
  - propositional connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - existential quantifier $\exists$, universal quantifier $\forall$
- *atomic formulas (atoms)*: a formula $\boldsymbol{x} = \boldsymbol{y}$ or $R(\boldsymbol{x}_1 \ldots \boldsymbol{x}_n)$ for $R \in \tau$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - (first-order) variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$
  - propositional connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - existential quantifier $\exists$, universal quantifier $\forall$
- *atomic formulas (atoms)*: a formula $\boldsymbol{x} = \boldsymbol{y}$ or $R(\boldsymbol{x}_1 \ldots \boldsymbol{x}_n)$ for $R \in \tau$
- *quantifier-free formula*: atoms, literals (= negated atoms), formulas built up from atoms by using propositional connectives

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - (first-order) variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$
  - propositional connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - existential quantifier $\exists$, universal quantifier $\forall$
- *atomic formulas (atoms)*: a formula $\boldsymbol{x} = \boldsymbol{y}$ or $R(\boldsymbol{x}_1 \ldots \boldsymbol{x}_n)$ for $R \in \tau$
- *quantifier-free formula*: atoms, literals (= negated atoms), formulas built up from atoms by using propositional connectives
- *quantifications* over (first-order variables):
  - existential quantifications $\exists x$ and universal quantifications $\forall x$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - (first-order) variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$
  - propositional connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - existential quantifier $\exists$, universal quantifier $\forall$
- *atomic formulas (atoms)*: a formula $\boldsymbol{x} = \boldsymbol{y}$ or $R(\boldsymbol{x}_1 \ldots \boldsymbol{x}_n)$ for $R \in \tau$
- *quantifier-free formula*: atoms, literals (= negated atoms), formulas built up from atoms by using propositional connectives
- *quantifications* over (first-order variables):
  - existential quantifications $\exists x$ and universal quantifications $\forall x$
- *formulas*:

$$\varphi ::= \boldsymbol{x} = \boldsymbol{y} \mid R(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{ar(R)}) \qquad \text{(where } R \in \tau\text{)}$$
$$\mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \varphi_1 \leftrightarrow \varphi_2$$
$$\mid \exists x\varphi \mid \forall x\varphi$$

First-order logic: syntax (language)

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - (first-order) variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$
  - propositional connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - existential quantifier $\exists$, universal quantifier $\forall$
- *atomic formulas (atoms)*: a formula $\boldsymbol{x} = \boldsymbol{y}$ or $R(\boldsymbol{x}_1 \ldots \boldsymbol{x}_n)$ for $R \in \tau$
- *quantifier-free formula*: atoms, literals (= negated atoms), formulas built up from atoms by using propositional connectives
- *quantifications* over (first-order variables):
  - existential quantifications $\exists x$ and universal quantifications $\forall x$
- *formulas*:

$$\varphi ::= \boldsymbol{x} = \boldsymbol{y} \mid R(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{ar(R)}) \qquad \text{(where } R \in \tau\text{)}$$
$$\mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \varphi_1 \leftrightarrow \varphi_2$$
$$\mid \exists x\varphi \mid \forall x\varphi$$

- *sentences*: formulas without *free* variables.

First-order logic: semantics (structures)

Definition

Let $\tau = \{R_1, \ldots, R_n\}$ be a vocabulary.
A *τ-structure* is a tuple $\mathcal{A} = \langle A; R_1^{\mathcal{A}}, \ldots R_n^{\mathcal{A}} \rangle$ consisting of:

- the *universe* A,
- *interpretations* $R_i{}^{\mathcal{A}} \subseteq A^{ar(R_i)} = \overbrace{A \times \ldots \times A}^{ar(R_i)}$ for each of the relation symbols R_i in τ, where $i \in \{1, \ldots, n\}$.

Examples

Let $\tau_G = \{E/_2\}$ vocabulary with binary edge relation.
The *standard structure* for a graph $\mathcal{G} = \langle V, E \rangle$:

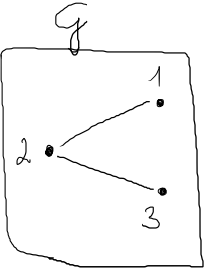
$$\mathcal{A}_{\tau_G}(\mathcal{G}) := \langle V; E^{\text{symm}} \rangle .$$

First-order structures

Definition

Let $\tau = \{R_1, \ldots, R_n\}$ be a vocabulary.
A τ-*structure* is a tuple $\mathcal{A} = \langle A; R_1^{\mathcal{A}}, \ldots R_n^{\mathcal{A}} \rangle$ consisting of:

- the *universe* A,
- *interpretations* $R_i{}^{\mathcal{A}} \subseteq A^{ar(R_i)} = \overbrace{A \times \ldots \times A}^{ar(R_i)}$ for each of the relation symbols R_i in τ, where $i \in \{1, \ldots, n\}$.



$G = \langle V, E \rangle$
$V = \{1,2,3\}$
$E = \{\{1,2\}, \{2,3\}\}$

Examples

Let $\tau_G = \{E/_2\}$ vocabulary with binary edge relation.
The *standard structure* for a graph $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_G}(\mathcal{G}) := \langle V; E^{\text{symm}} \rangle .$$

$\mathcal{A}_{\tau_G}(G) = \langle V; E^{\mathcal{A}_{\tau_G}(G)} \rangle$
$V = \{1,2,3\}$
$E^{\mathcal{A}_{\tau_G}(G)} = E^{\text{symm}} = \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\}$

First-order logic: semantics (structures)

Definition

Let $\tau = \{R_1, \ldots, R_n\}$ be a vocabulary.
A *τ-structure* is a tuple $\mathcal{A} = \langle A; R_1^{\mathcal{A}}, \ldots R_n^{\mathcal{A}} \rangle$ consisting of:

- the *universe* A,
- *interpretations* ${R_i}^{\mathcal{A}} \subseteq A^{ar(R_i)} = \overbrace{A \times \ldots \times A}^{ar(R_i)}$ for each of the relation symbols R_i in τ, where $i \in \{1, \ldots, n\}$.

Examples

Let $\tau_G = \{E/_2\}$ vocabulary with binary edge relation.
The *standard structure* for a graph $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_G}(\mathcal{G}) := \langle V; E^{\text{symm}} \rangle .$$

Example

Let $\tau_{HG} = \{VERT/_1, EDGE/_1, INC/_2\}$ vocabulary (for hypergraphs).
The *hypergraph structure* for a graph $\mathcal{G} = \langle V, E \rangle$:

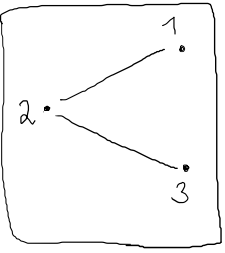
$$\mathcal{A}_{\tau_{HG}}(\mathcal{G}) := \langle V \cup E; V, E, \{\langle v, e \rangle \mid v \in V, e \in E, v \in e\} \rangle .$$

First-order structures

Definition

Let $\tau = \{R_1, \ldots, R_n\}$ be a vocabulary.
A *τ-structure* is a tuple $\mathcal{A} = \langle A; R_1^{\mathcal{A}}, \ldots R_n^{\mathcal{A}} \rangle$ consisting of:

- the *universe* A,
- *interpretations* $R_i{}^{\mathcal{A}} \subseteq A^{ar(R_i)} = \overbrace{A \times \ldots \times A}^{ar(R_i)}$ for each of the relation symbols R_i in τ, where $i \in \{1, \ldots, n\}$.



$\mathcal{A}_{\tau_{HG}}(G) := \langle V \cup E;\ VERT^{\mathcal{A}_{\tau_{HG}}(G)}, E^{\mathcal{A}_{\tau_{HG}}(G)}, INC^{\mathcal{A}_{\tau_{HG}}(G)} \rangle$

$V \cup E = \{1, 2, 3, \{1,2\}, \{2,3\}\}$

$VERT^{\mathcal{A}_{\tau_{HG}}(G)} = \{1, 2, 3\}$

$EDGE^{\mathcal{A}_{\tau_{HG}}(G)} = \{\{1,2\}, \{2,3\}\}$

$INC^{\mathcal{A}_{\tau_{HG}}(G)} = \{\langle 1, \{1,2\}\rangle, \langle 2, \{1,2\}\rangle, \langle 2, \{2,3\}\rangle, \langle 3, \{2,3\}\rangle\}$

Example

Let $\tau_{HG} = \{VERT/_1, EDGE/_1, INC/_2\}$ vocabulary (for hypergraphs).
The *hypergraph structure* for a graph $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{HG}}(\mathcal{G}) := \langle V \cup E;\ V,\ E,\ \{\langle v, e\rangle \mid v \in V, e \in E, v \in e\}\rangle .$$

Interpretation of first-order formulas in structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure. For a τ-formula $\varphi(x_1, \ldots, x_k)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k$ in $\mathcal{A}$ is defined by:

Interpretation of first-order formulas in structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure. For a τ-formula $\varphi(x_1, \ldots, x_k)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k$ in $\mathcal{A}$ is defined by:

- If $\varphi(x_1, \ldots, x_k) \equiv R(x_{i_1}, \ldots, x_{i_r})$ with $i_1, \ldots, i_r \in [k]$, then:

$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{i_1}, \ldots, a_{i_k} \rangle \in R^{\mathcal{A}}\}$$

Interpretation of first-order formulas in structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure. For a τ-formula $\varphi(x_1, \ldots, x_k)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k$ in $\mathcal{A}$ is defined by:

- If $\varphi(x_1, \ldots, x_k) \equiv R(x_{i_1}, \ldots, x_{i_r})$ with $i_1, \ldots, i_r \in [k]$, then:

$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{i_1}, \ldots, a_{i_k} \rangle \in R^{\mathcal{A}}\}$$

$E(x,y)$

$E(x,y)(\mathcal{A}_{J_G}(G)) = \{\langle a_1, a_2 \rangle \mid \langle a_1, a_2 \rangle \in E^{\mathcal{A}_{J_G}(G)}\} = E^{\mathcal{A}_{J_G}(G)}$

$G = \langle V, E \rangle$

Interpretation of first-order formulas in structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure. For a τ-formula $\varphi(x_1, \ldots, x_k)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k$ in $\mathcal{A}$ is defined by:

- If $\varphi(x_1, \ldots, x_k) \equiv R(x_{i_1}, \ldots, x_{i_r})$ with $i_1, \ldots, i_r \in [k]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{i_1}, \ldots, a_{i_k} \rangle \in R^{\mathcal{A}}\}$$
- If $\varphi(x_1, \ldots, x_k) \equiv \varphi_1(x_{i_1}, \ldots, x_{i_l}) \wedge \varphi_2(x_{j_1}, \ldots, x_{j_m})$ with $i_1, \ldots, i_l, j_1, \ldots, j_m \in [k]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{i_1}, \ldots, a_{i_l} \rangle \in \varphi_1(\mathcal{A})\} \cap \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{j_1}, \ldots, a_{j_m} \rangle \in \varphi_2(\mathcal{A})\}$$

Interpretation of first-order formulas in structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure. For a τ-formula $\varphi(x_1, \ldots, x_k)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k$ in $\mathcal{A}$ is defined by:

- If $\varphi(x_1, \ldots, x_k) \equiv R(x_{i_1}, \ldots, x_{i_r})$ with $i_1, \ldots, i_r \in [k]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{i_1}, \ldots, a_{i_k} \rangle \in R^{\mathcal{A}}\}$$
- If $\varphi(x_1, \ldots, x_k) \equiv \varphi_1(x_{i_1}, \ldots, x_{i_l}) \wedge \varphi_2(x_{j_1}, \ldots, x_{j_m})$ with $i_1, \ldots, i_l, j_1, \ldots, j_m \in [k]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{i_1}, \ldots, a_{i_l} \rangle \in \varphi_1(\mathcal{A})\} \cap \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{j_1}, \ldots, a_{j_m} \rangle \in \varphi_2(\mathcal{A})\}$$
- If $\varphi(x_1, \ldots, x_k) \equiv \exists x_{k+1} \varphi_0(x_{i_1}, \ldots, x_{i_\ell})$ with $i_1, \ldots, i_\ell \in [k+1]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \text{there exists } a_{k+1} \in A \text{ such that } \langle a_{i_1}, \ldots, a_{i_\ell} \rangle \in \varphi_0(\mathcal{A})\}$$

Interpretation of first-order formulas in structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure. For a τ-formula $\varphi(x_1, \ldots, x_k)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k$ in $\mathcal{A}$ is defined by:

- If $\varphi(x_1, \ldots, x_k) \equiv R(x_{i_1}, \ldots, x_{i_r})$ with $i_1, \ldots, i_r \in [k]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{i_1}, \ldots, a_{i_k} \rangle \in R^{\mathcal{A}}\}$$
- If $\varphi(x_1, \ldots, x_k) \equiv \varphi_1(x_{i_1}, \ldots, x_{i_l}) \wedge \varphi_2(x_{j_1}, \ldots, x_{j_m})$ with $i_1, \ldots, i_l, j_1, \ldots, j_m \in [k]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{i_1}, \ldots, a_{i_l} \rangle \in \varphi_1(\mathcal{A})\} \cap \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \langle a_{j_1}, \ldots, a_{j_m} \rangle \in \varphi_2(\mathcal{A})\}$$
- If $\varphi(x_1, \ldots, x_k) \equiv \exists x_{k+1} \varphi_0(x_{i_1}, \ldots, x_{i_\ell})$ with $i_1, \ldots, i_\ell \in [k+1]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k \rangle \in A^k \mid \text{there exists } a_{k+1} \in A \text{ such that } \langle a_{i_1}, \ldots, a_{i_\ell} \rangle \in \varphi_0(\mathcal{A})\}$$

- $\mathcal{A} \models \varphi(a_1, \ldots, a_k)$ will mean: $\langle a_1, \ldots, a_k \rangle \in \varphi(\mathcal{A})$.

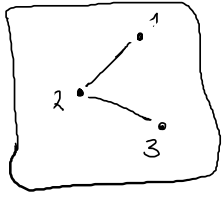
Interpretation of first-order formulas in structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R\in\tau}\rangle$ be a τ-structure. For a τ-formula $\varphi(x_1,\dots,x_k)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k$ in $\mathcal{A}$ is defined by:

- If $\varphi(x_1,\dots,x_k) \equiv R(x_{i_1},\dots,x_{i_r})$ with $i_1,\dots,i_r \in [k]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1,\dots,a_k\rangle \in A^k \mid \langle a_{i_1},\dots,a_{i_k}\rangle \in R^{\mathcal{A}}\}$$
- If $\varphi(x_1,\dots,x_k) \equiv \varphi_1(x_{i_1},\dots,x_{i_l}) \wedge \varphi_2(x_{j_1},\dots,x_{j_m})$ with $i_1,\dots,i_l,j_1,\dots,j_m \in [k]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1,\dots,a_k\rangle \in A^k \mid \langle a_{i_1},\dots,a_{i_l}\rangle \in \varphi_1(\mathcal{A})\} \cap \{\langle a_1,\dots,a_k\rangle \in A^k \mid \langle a_{j_1},\dots,a_{j_m}\rangle \in \varphi_2(\mathcal{A})\}$$
- If $\varphi(x_1,\dots,x_k) \equiv \exists x_{k+1}\varphi_0(x_{i_1},\dots,x_{i_\ell})$ with $i_1,\dots,i_\ell \in [k+1]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1,\dots,a_k\rangle \in A^k \mid \text{there exists } a_{k+1}\in A \text{ such that } \langle a_{i_1},\dots,a_{i_\ell}\rangle \in \varphi_0(\mathcal{A})\}$$

- $\mathcal{A} \models \varphi(a_1,\dots,a_k)$ will mean: $\langle a_1,\dots,a_k\rangle \in \varphi(\mathcal{A})$.
- For a sentence φ, $\mathcal{A} \models \varphi$ will mean $\varphi(\mathcal{A}) \neq \varnothing$ (then $\varphi(\mathcal{A}) = \{\langle\rangle\}$).

$\langle V, E \rangle = G$



$\mathcal{A}_{J_G}(G) = \langle \underbrace{\{1,2,3\}}_{A=V}; \underbrace{\{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\}}_{E^{\mathcal{A}}} \rangle$

$I_G = \{E/2\}$ graph signature

$J_{HG} = \{VERT/1, EDGE/1, INC/2\}$

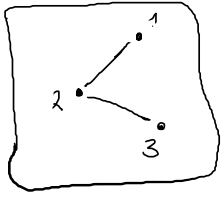
$\mathcal{A}_{J_{HG}}(G) = \langle \underbrace{\{1,2,3,\{1,2\},\{2,3\}\}}_{A = V \cup E}; \underbrace{\{1,2,3\}}_{VERT^{\mathcal{A}(G)}_{J_{HG}}}, \underbrace{\{\{1,2\},\{2,3\}\}}_{EDGE^{\mathcal{A}(G)}}, \underbrace{\{\langle 1,\{1,2\}\rangle, \langle 2,\{1,2\}\rangle, \langle 2,\{2,3\}\rangle, \langle 3,\{2,3\}\rangle\}}_{INC^{\mathcal{A}(G)}} \rangle$

$(E(x,y))(\mathcal{A}_{J_G}(G)) = ?$

$(E(x,x))(\mathcal{A}_{J_G}(G)) = ?$

$(\exists y (E(x,y) \wedge E(y,z)))(\mathcal{A}_{J_G}(G)) = ?$

$\langle V, E \rangle = G$



$\mathcal{A}_{J_G}(G) = \langle \underbrace{\{1,2,3\}}_{A=V}; \underbrace{\{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\}}_{E^{\mathcal{A}}} \rangle$

$I_G = \{E/2\}$ graph signature

$J_{HG} = \{VERT/1, EDGE/1, INC/2\}$

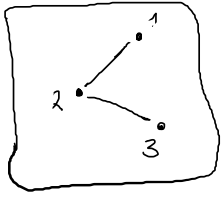
$\mathcal{A}_{J_{HG}}(G) = \langle \underbrace{\{1,2,3,\{1,2\},\{2,3\}\}}_{A = V \cup E}; \underbrace{\{1,2,3\}}_{VERT^{\mathcal{A}(G)}_{J_{HG}}}, \underbrace{\{\{1,2\},\{2,3\}\}}_{EDGE^{\mathcal{A}(G)}}, \underbrace{\{\langle 1,\{1,2\}\rangle, \langle 2,\{1,2\}\rangle, \langle 2,\{2,3\}\rangle, \langle 3,\{2,3\}\rangle\}}_{INC^{\mathcal{A}(G)}} \rangle$

$(E(x,y))(\mathcal{A}_{J_G}(G)) = \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\} = E^{\mathcal{A}_{J_G}(G)}$

$(E(x,x))(\mathcal{A}_{J_G}(G)) = ?$

$(\exists y (E(x,y) \wedge E(y,z)))(\mathcal{A}_{J_G}(G)) = ?$

$\langle V, E \rangle = G$



$\mathcal{A}_{J_G}(G) = \langle \underbrace{\{1,2,3\}}_{A=V}; \underbrace{\{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\}}_{E^{\mathcal{A}}} \rangle$

$I_G = \{E/2\}$ graph signature

$J_{HG} = \{VERT/1, EDGE/1, INC/2\}$

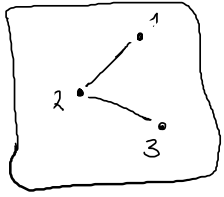
$\mathcal{A}_{J_{HG}}(G) = \langle \underbrace{\{1,2,3,\{1,2\},\{2,3\}\}}_{A = V \cup E}; \underbrace{\{1,2,3\}}_{VERT^{\mathcal{A}(G)}_{J_{HG}}}, \underbrace{\{\{1,2\},\{2,3\}\}}_{EDGE^{\mathcal{A}(G)}}, \underbrace{\{\langle 1,\{1,2\}\rangle, \langle 2,\{1,2\}\rangle, \langle 2,\{2,3\}\rangle, \langle 3,\{2,3\}\rangle\}}_{INC^{\mathcal{A}(G)}} \rangle$

$(E(x,y))(\mathcal{A}_{J_G}(G)) = \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\} = E^{\mathcal{A}_{J_G}(G)}$

$(E(x,x))(\mathcal{A}_{J_G}(G)) = \{\}$

$(\exists y (E(x,y) \wedge E(y,z)))(\mathcal{A}_{J_G}(G)) = ?$

$\langle V, E \rangle = G$



$\mathcal{A}_{J_G}(G) = \langle \underbrace{\{1,2,3\}}_{A=V}; \underbrace{\{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\}}_{E^{\mathcal{A}}} \rangle$

$I_G = \{E/2\}$ graph signature

$J_{HG} = \{VERT/1, EDGE/1, INC/2\}$

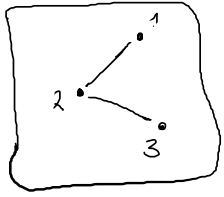
$\mathcal{A}_{J_{HG}}(G) = \langle \underbrace{\{1,2,3,\{1,2\},\{2,3\}\}}_{A = V \cup E}; \underbrace{\{1,2,3\}}_{VERT^{\mathcal{A}(G)}_{J_{HG}}}, \underbrace{\{\{1,2\},\{2,3\}\}}_{EDGE^{\mathcal{A}(G)}}, \underbrace{\{\langle 1,\{1,2\}\rangle, \langle 2,\{1,2\}\rangle, \langle 2,\{2,3\}\rangle, \langle 3,\{2,3\}\rangle\}}_{INC^{\mathcal{A}(G)}} \rangle$

$(E(x,y))(\mathcal{A}_{J_G}(G)) = \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\} = E^{\mathcal{A}_{J_G}(G)}$

$(E(x,x))(\mathcal{A}_{J_G}(G)) = \{\}$

$(\exists y (E(x,y) \wedge E(y,z)))(\mathcal{A}_{J_G}(G)) = \{\langle 1,3\rangle, \langle 1,1\rangle, \langle 3,1\rangle, \langle 3,3\rangle, \langle 2,2\rangle\}$

$\langle V, E \rangle = G$



$\mathcal{A}_{J_G}(G) = \langle \underbrace{\{1,2,3\}}_{A=V}; \underbrace{\{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\}}_{E^{\mathcal{A}}} \rangle$

$I_G = \{E/2\}$ graph signature

$J_{HG} = \{VERT/1, EDGE/1, INC/2\}$

$\mathcal{A}_{J_{HG}}(G) = \langle \underbrace{\{1,2,3,\{1,2\},\{2,3\}\}}_{A = V \cup E}; \underbrace{\{1,2,3\}}_{VERT^{\mathcal{A}(G)}_{J_{HG}}}, \underbrace{\{\{1,2\},\{2,3\}\}}_{EDGE^{\mathcal{A}(G)}}, \underbrace{\{\langle 1,\{1,2\}\rangle, \langle 2,\{1,2\}\rangle, \langle 2,\{2,3\}\rangle, \langle 3,\{2,3\}\rangle\}}_{INC^{\mathcal{A}(G)}} \rangle$

$(E(x,y))(\mathcal{A}_{J_G}(G)) = \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\} = E^{\mathcal{A}_{J_G}(G)}$

$(E(x,x))(\mathcal{A}_{J_G}(G)) = \{\}$

$(\exists y (E(x,y) \wedge E(y,z)))(\mathcal{A}_{J_G}(G)) = \{\langle 1,3\rangle, \langle 1,1\rangle, \langle 3,1\rangle, \langle 3,3\rangle, \langle 2,2\rangle\}$

Expressing graph properties by first-order formulas

Exercise

For given formulas $\varphi(x)$ and for all $k \in \mathbb{N}$, $k \geq 1$ define formulas $\exists^{\geq k} x\, \varphi(x)$, $\exists^{<k} x\, \varphi(x)$, $\exists^{=k} x\, \varphi(x)$, such that in a given τ-structure $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$:

$$\mathcal{A} \models \exists^{\geq k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| \geq k$$

$$\mathcal{A} \models \exists^{<k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| < k$$

$$\mathcal{A} \models \exists^{=k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| = k$$

Expressing graph properties by first-order formulas

Exercise

For given formulas $\varphi(x)$ and for all $k \in \mathbb{N}$, $k \geq 1$ define formulas $\exists^{\geq k} x\, \varphi(x)$, $\exists^{<k} x\, \varphi(x)$, $\exists^{=k} x\, \varphi(x)$, such that in a given τ-structure $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$:

$$\mathcal{A} \models \exists^{\geq k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| \geq k$$

$$\mathcal{A} \models \exists^{<k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| < k$$

$$\mathcal{A} \models \exists^{=k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| = k$$

$\exists^{\geq k} x\, \varphi(x) :=$

$\exists^{<k} x\, \varphi(x) :=$

$\exists^{=k} x\, \varphi(x) :=$

Expressing graph properties by first-order formulas

Exercise

For given formulas $\varphi(x)$ and for all $k \in \mathbb{N}$, $k \geq 1$ define formulas $\exists^{\geq k} x\, \varphi(x)$, $\exists^{<k} x\, \varphi(x)$, $\exists^{=k} x\, \varphi(x)$, such that in a given τ-structure $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$:

$$\mathcal{A} \models \exists^{\geq k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| \geq k$$
$$\mathcal{A} \models \exists^{<k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| < k$$
$$\mathcal{A} \models \exists^{=k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| = k$$

$\exists^{\geq k} x\, \varphi(x) \quad := \quad \exists x_1 \ldots \exists x_k \Big(\big(\bigwedge_{1 \leq i < j \leq n} \neg x_i = x_j \big) \wedge \bigwedge_{1 \leq i \leq n} \varphi(x_i) \Big)$

$\exists^{<k} x\, \varphi(x) \quad :=$

$\exists^{=k} x\, \varphi(x) \quad :=$

Expressing graph properties by first-order formulas

Exercise

For given formulas $\varphi(x)$ and for all $k \in \mathbb{N}$, $k \geq 1$ define formulas $\exists^{\geq k} x\, \varphi(x)$, $\exists^{<k} x\, \varphi(x)$, $\exists^{=k} x\, \varphi(x)$, such that in a given τ-structure $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$:

$$\mathcal{A} \models \exists^{\geq k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| \geq k$$
$$\mathcal{A} \models \exists^{<k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| < k$$
$$\mathcal{A} \models \exists^{=k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| = k$$

$\exists^{\geq k} x\, \varphi(x) \quad := \quad \exists x_1 \ldots \exists x_k \Big(\big(\bigwedge_{1 \leq i < j \leq n} \neg x_i = x_j \big) \wedge \bigwedge_{1 \leq i \leq n} \varphi(x_i) \Big)$

$\exists^{<k} x\, \varphi(x) \quad := \quad \neg\, \exists^{\geq k} x\, \varphi(x)$

$\exists^{=k} x\, \varphi(x) \quad :=$

Expressing graph properties by first-order formulas

Exercise

For given formulas $\varphi(x)$ and for all $k \in \mathbb{N}$, $k \geq 1$ define formulas $\exists^{\geq k} x\, \varphi(x)$, $\exists^{<k} x\, \varphi(x)$, $\exists^{=k} x\, \varphi(x)$, such that in a given τ-structure $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$:

$$\mathcal{A} \models \exists^{\geq k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| \geq k$$
$$\mathcal{A} \models \exists^{<k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| < k$$
$$\mathcal{A} \models \exists^{=k} x\, \varphi(x) \iff |\{a \in A \mid \mathcal{A} \models \varphi(a)\}| = k$$

$$\exists^{\geq k} x\, \varphi(x) \quad :\Leftrightarrow \quad \exists x_1 \ldots \exists x_k \Big(\Big(\bigwedge_{1 \leq i < j \leq n} \neg x_i = x_j \Big) \wedge \bigwedge_{1 \leq i \leq n} \varphi(x_i) \Big)$$

$$\exists^{<k} x\, \varphi(x) \quad :\Leftrightarrow \quad \neg\, \exists^{\geq k} x\, \varphi(x)$$

$$\exists^{=k} x\, \varphi(x) \quad :\Leftrightarrow \quad \exists^{\geq k} x\, \varphi(x) \wedge \exists^{<k+1} x\, \varphi(x)$$

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary $\tau_G = \{E/_2\}$ for graphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements,

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary $\tau_G = \{E/_2\}$ for graphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements,

Recall:

$$\varphi_k := \exists x_1 \ldots \exists x_k \Big(\bigwedge_{1 \le i < j \le k} \big(\neg(x_i = y_i) \wedge \neg E(x_i, x_j) \big) \Big)$$

$$\mathcal{A}_{\tau_G}(\mathcal{G}) \models \varphi_k \iff \mathcal{G} \text{ has a } k\text{-element independent set}.$$

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary $\tau_G = \{E/_2\}$ for graphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements,

$\varphi_{k\text{-clique}} := ?$

$(\mathfrak{A}_G \models \varphi_{k\text{-clique}}) \iff$ G contains a clique of k vertices

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary $\tau_G = \{E/_2\}$ for graphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements,

$$\varphi_{k\text{-clique}} := \exists x_1 \dots \exists x_k \left(\left(\bigwedge_{1 \le i < j \le n} \neg x_i = x_j \right) \wedge \bigwedge_{1 \le i < j \le n} E(x_i, x_j) \right)$$

$$\left(\mathcal{A}_G \models \varphi_{k\text{-clique}} \right) \iff G \text{ contains a clique of } k \text{ vertices}$$

Expressing graph properties by first-order formulas

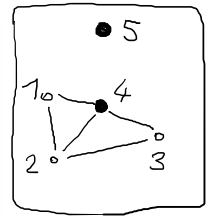
Exercise

Express by a first-order formula with the vocabulary $\tau_G = \{E/_2\}$ for graphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements,



$G=\langle V,E\rangle$ $S\subseteq V$ is dominating set in $G :\Leftrightarrow \forall v\in V\setminus S\ \exists w\in S\ (E(v,w))$

Expressing graph properties by first-order formulas

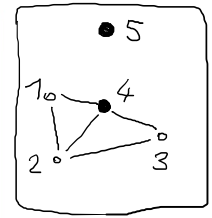
Exercise

Express by a first-order formula with the vocabulary $\tau_G = \{E/_2\}$ for graphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements,



dominating set

$G=\langle V,E\rangle$ $S\subseteq V$ is dominating set in $G :\Leftrightarrow \forall v\in V\setminus S\ \exists w\in S\ (E(v,w))$

$\Leftrightarrow \forall v\in V\ (v\notin S \rightarrow \exists w\in S\ (E(v,w)))$

Expressing graph properties by first-order formulas

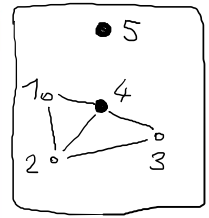
Exercise

Express by a first-order formula with the vocabulary $\tau_G = \{E/_2\}$ for graphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements,



dominating set

$G = \langle V, E \rangle$ $\quad S \subseteq V$ is dominating set in $G :\Leftrightarrow \forall v \in V \setminus S \; \exists w \in S \; (E(v,w))$

$\Leftrightarrow \forall v \in V \; (v \notin S \rightarrow \exists w \in S \; (E(v,w)))$

$\varphi_{\text{domset} \leq k} := \exists w_1 \ldots \exists w_k \; \forall v \; \big((\bigwedge_{i=1}^{k} \neg v = w_i) \rightarrow \bigvee_{i=1}^{k} E(v, w_i) \big)$

Expressing graph properties by first-order formulas

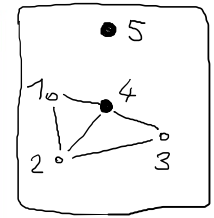
Exercise

Express by a first-order formula with the vocabulary $\tau_G = \{E/_2\}$ for graphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements,



dominating set

$G = \langle V, E \rangle$ $\quad S \subseteq V$ is dominating set in $G :\Leftrightarrow \forall v \in V \setminus S \; \exists w \in S \; (E(v,w))$

$\Leftrightarrow \forall v \in V \; (v \notin S \rightarrow \exists w \in S \; (E(v,w)))$

$$\varphi_{\text{domset} \leq k} := \exists w_1 \ldots \exists w_k \; \forall v \; \Big(\Big(\bigwedge_{i=1}^{k} \neg v = w_i \Big) \rightarrow \bigvee_{i=1}^{k} E(v, w_i) \Big)$$

$$\varphi_{\text{domset} = k} := \exists w_1 \ldots \exists w_k \; \Big(\Big(\bigwedge_{1 \leq i < j \leq k} \neg w_i = w_j \Big) \Big) \wedge \forall g \Big(\Big(\bigwedge_{i=1}^{k} \neg v = w_i \Big) \rightarrow \bigvee_{i=1}^{k} E(w_i, w) \Big)$$

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary with vocabulary $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$ for hypergraphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements.

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary with vocabulary $\tau_{HG} = \{VERT/_1, EDGE/_1, INC/_2\}$ for hypergraphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements.

(i) $\varphi_{k\text{-clique}} :=$

(ii) $\varphi_{\text{domset} \leq k} :=$

(iii) $\varphi_{\text{domset} = k} :=$

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary with vocabulary $\tau_{HG} = \{VERT/_1, EDGE/_1, INC/_2\}$ for hypergraphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements.

(i) $\varphi_{k\text{-clique}} := \exists x_1 \cdots \exists x_k \Big(\bigwedge_{1 \le i \le k} VERT(x_i) \wedge \bigwedge_{1 \le i < j \le k} \big(\neg x_i = x_j \wedge \exists e \big(EDGE(e) \wedge INC(x_i, e) \wedge INC(x_j, e) \big) \big) \Big)$

(ii) $\varphi_{domset \le k} :=$

(iii) $\varphi_{domset = k} :=$

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary with vocabulary $\tau_{HG} = \{VERT/_1, EDGE/_1, INC/_2\}$ for hypergraphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements.

(i) $\varphi_{k\text{-clique}} := \exists x_1 \dots \exists x_k \Big(\bigwedge_{1 \le i \le k} VERT(x_i) \wedge \bigwedge_{1 \le i < j \le k} \big(\neg x_i = x_j \wedge \exists e \big(EDGE(e) \wedge INC(x_i, e) \wedge INC(x_j, e) \big) \big) \Big)$

(ii) $\varphi_{domset \le k} := \exists x_1 \dots \exists x_k \Big(\bigwedge_{1 \le i \le k} VERT(x_i) \wedge \forall y \big(VERT(y) \wedge \bigwedge_{1 \le i \le k} \neg y = x_i \rightarrow \exists e \big(EDGE(e) \wedge INC(y, e) \wedge \bigvee_{1 \le i \le k} INC(x_i, e) \big) \big) \Big)$

(iii) $\varphi_{domset = k} :=$

Expressing graph properties by first-order formulas

Exercise

Express by a first-order formula with the vocabulary with vocabulary $\tau_{HG} = \{VERT/_1, EDGE/_1, INC/_2\}$ for hypergraphs that:

(i) a graph $\mathcal{G}$ contains a clique with precisely k elements,

(ii) a graph $\mathcal{G}$ has a dominating set with less or equal to k elements,

(iii) a graph $\mathcal{G}$ has a dominating set with precisely k elements.

(i) $\varphi_{k\text{-clique}} := \exists x_1 \dots \exists x_k \Big(\bigwedge_{1 \le i \le k} VERT(x_i) \wedge \bigwedge_{1 \le i < j \le k} \big(\neg x_i = x_j \wedge \exists e \big(EDGE(e) \wedge INC(x_i, e) \wedge INC(x_j, e) \big) \big) \Big)$

(ii) $\varphi_{domset \le k} := \exists x_1 \dots \exists x_k \Big(\bigwedge_{1 \le i \le k} VERT(x_i) \wedge \forall y \big(VERT(y) \wedge \bigwedge_{1 \le i \le k} \neg y = x_i \rightarrow \exists e \big(EDGE(e) \wedge INC(y, e) \wedge \bigvee_{1 \le i \le k} INC(x_i, e) \big) \big) \Big)$

(iii) $\varphi_{domset = k} := \exists x_1 \dots \exists x_k \Big(\big(\bigwedge_{1 \le i \le k} VERT(x_i) \wedge \bigwedge_{1 \le i < j \le k} \neg x_i = x_j \wedge \dots \big) \Big)$

Evaluation and model checking (first-order logic)

Let Φ be a class of first-order formulas.
The *evaluation problem* for Φ:

EVAL(Φ)
Instance: A structure $\mathcal{A}$ and a formula $\varphi \in \Phi$.
Problem: Compute $\varphi(\mathcal{A})$.

Evaluation and model checking (first-order logic)

Let Φ be a class of first-order formulas.
The *evaluation problem* for Φ:

EVAL(Φ)
Instance: A structure $\mathcal{A}$ and a formula $\varphi \in \Phi$.
Problem: Compute $\varphi(\mathcal{A})$.

Evaluation and model checking (first-order logic)

Let Φ be a class of first-order formulas.
The *evaluation problem* for Φ:

EVAL(Φ)
Instance: A structure $\mathcal{A}$ and a formula $\varphi \in \Phi$.
Problem: Compute $\varphi(\mathcal{A})$.

The *model checking problem* for Φ:

MC(Φ)
Instance: A structure $\mathcal{A}$ and a formula $\varphi \in \Phi$.
Problem: Decide whether $\mathcal{A} \models \varphi$ (that is, $\varphi(\mathcal{A}) \neq \emptyset$).

Evaluation and model checking (first-order logic)

Let Φ be a class of first-order formulas.
The *evaluation problem* for Φ:

EVAL(Φ)
Instance: A structure $\mathcal{A}$ and a formula $\varphi \in \Phi$.
Problem: Compute $\varphi(\mathcal{A})$.

The *model checking problem* for Φ:

MC(Φ)
Instance: A structure $\mathcal{A}$ and a formula $\varphi \in \Phi$.
Problem: Decide whether $\mathcal{A} \models \varphi$ (that is, $\varphi(\mathcal{A}) \neq \emptyset$).

Width of formula φ: maximal number of free variables in a subformula of φ.

Theorem

EVAL(FO) *and* MC(FO) *can be solved in time* $O(|\varphi| \cdot |A|^w \cdot w)$, *where* w *is the width of the input formula* φ.

Monadic second-order logic (formula example)

$$\psi_3 := \exists C_1 \exists C_2 \exists C_3 \Big(\Big(\forall x \Big(\bigvee_{i=1}^{3} C_i(x) \Big) \Big) \wedge \forall x \Big(\bigwedge_{1 \le i < j \le 3} \neg \Big(C_i(x) \wedge C_j(x) \Big) \Big) \\ \wedge \forall x \forall y \Big(E(x,y) \rightarrow \bigwedge_{i=1}^{3} \neg (C_i(x) \wedge C_i(y)) \Big) \Big)$$

$$\mathcal{A}(\mathcal{G}) \vDash \psi_3 \iff \mathcal{G} \text{ has is } 3\text{-colorable}.$$

Monadic second-order logic

- language based on:
  - a *vocabulary* $\tau = \{R_1, \ldots, R_n\}$ of *predicate symbols* R_i together with arity *ar*$(R_i) \in \mathbb{N}$
  - the binary equality predication $=$
  - first-order variable symbols: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, \ldots$
  - monadic second-order variable symbols (symbolizing variables for unary predicate symbols): $X, Y, Z, W, X_1, Y_1, Z_1, W_1, X_1, \ldots,$
  - propositional connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - existential quantifier $\exists$, universal quantifier $\forall$

Interpretation of MSO-formulas in first-order structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure.
For a MSO(τ)-formula $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k \times \mathcal{P}(A)^\ell$ in $\mathcal{A}$ is defined by:

Interpretation of MSO-formulas in first-order structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure.
For a MSO(τ)-formula $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k \times \mathcal{P}(A)^\ell$ in $\mathcal{A}$ is defined by:

- similar clauses as before, plus:

Interpretation of MSO-formulas in first-order structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure.
For a MSO(τ)-formula $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k \times \mathcal{P}(A)^\ell$ in $\mathcal{A}$ is defined by:

- similar clauses as before, plus:
- If $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell) \equiv \boldsymbol{X}_i(\boldsymbol{x}_j)$ with $i \in [k]$ and $j \in [\ell]$, then:

$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in A^k \times \mathcal{P}(A)^\ell \mid a_j \in P_i\}$$

Interpretation of MSO-formulas in first-order structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure.
For a MSO(τ)-formula $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k \times \mathcal{P}(A)^\ell$ in $\mathcal{A}$ is defined by:

- similar clauses as before, plus:
- If $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell) \equiv \boldsymbol{X}_i(\boldsymbol{x}_j)$ with $i \in [k]$ and $j \in [\ell]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in A^k \times \mathcal{P}(A)^\ell \mid a_j \in P_i\}$$
- If $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell) \equiv \exists \boldsymbol{X}_{k+1} \varphi_0(\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{k'}}, \boldsymbol{X}_{j_1}, \ldots, \boldsymbol{X}_{j_{\ell'}})$ with $i_1, \ldots, i_{k'} \in [k]$, and $j_1, \ldots, j_{\ell'} \in [\ell + 1]$ then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in A^k \times \mathcal{P}(A)^\ell \mid \text{there exists } P_{\ell+1} \in \mathcal{P}(A) \text{ such that } \langle a_{i_1}, \ldots, a_{i_{k'}}, P_{j_1}, \ldots, P_{j_{\ell'}} \rangle \in \varphi_0(\mathcal{A})\}$$

Interpretation of MSO-formulas in first-order structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure.
For a MSO(τ)-formula $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k \times \mathcal{P}(A)^\ell$ in $\mathcal{A}$ is defined by:

- similar clauses as before, plus:
- If $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell) \equiv \boldsymbol{X}_i(\boldsymbol{x}_j)$ with $i \in [k]$ and $j \in [\ell]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in A^k \times \mathcal{P}(A)^\ell \mid a_j \in P_i\}$$
- If $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell) \equiv \exists \boldsymbol{X}_{k+1} \varphi_0(\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{k'}}, \boldsymbol{X}_{j_1}, \ldots, \boldsymbol{X}_{j_{\ell'}})$ with $i_1, \ldots, i_{k'} \in [k]$, and $j_1, \ldots, j_{\ell'} \in [\ell + 1]$ then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in A^k \times \mathcal{P}(A)^\ell \mid \text{there exists } P_{\ell+1} \in \mathcal{P}(A) \text{ such that } \langle a_{i_1}, \ldots, a_{i_{k'}}, P_{j_1}, \ldots, P_{j_{\ell'}} \rangle \in \varphi_0(\mathcal{A})\}$$

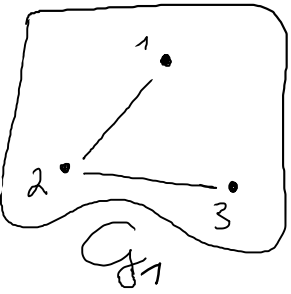
- $\mathcal{A} \models \varphi(a_1, \ldots, a_k, P_1, \ldots, P_\ell)$
will mean: $\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in \varphi(\mathcal{A})$.

Interpretation of MSO-formulas in first-order structures

Let $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R \in \tau} \rangle$ be a τ-structure.
For a MSO(τ)-formula $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell)$ its *interpretation* $\varphi(\mathcal{A}) \subseteq A^k \times \mathcal{P}(A)^\ell$ in $\mathcal{A}$ is defined by:

- similar clauses as before, plus:
- If $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell) \equiv \boldsymbol{X}_i(\boldsymbol{x}_j)$ with $i \in [k]$ and $j \in [\ell]$, then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in A^k \times \mathcal{P}(A)^\ell \mid a_j \in P_i\}$$
- If $\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell) \equiv \exists \boldsymbol{X}_{k+1} \varphi_0(\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{k'}}, \boldsymbol{X}_{j_1}, \ldots, \boldsymbol{X}_{j_{\ell'}})$ with $i_1, \ldots, i_{k'} \in [k]$, and $j_1, \ldots, j_{\ell'} \in [\ell + 1]$ then:
$$\varphi(\mathcal{A}) := \{\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in A^k \times \mathcal{P}(A)^\ell \mid \text{there exists } P_{\ell+1} \in \mathcal{P}(A) \text{ such that } \langle a_{i_1}, \ldots, a_{i_{k'}}, P_{j_1}, \ldots, P_{j_{\ell'}} \rangle \in \varphi_0(\mathcal{A})\}$$

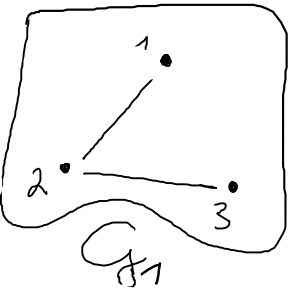
- $\mathcal{A} \models \varphi(a_1, \ldots, a_k, P_1, \ldots, P_\ell)$
will mean: $\langle a_1, \ldots, a_k, P_1, \ldots, P_\ell \rangle \in \varphi(\mathcal{A})$.
- For a sentence φ, $\mathcal{A} \models \varphi$ will mean $\varphi(\mathcal{A}) \neq \emptyset$ (then $\varphi(\mathcal{A}) = \{\langle\rangle\}$).



$\varphi(A) :=$ $\forall x \left(A(x) \rightarrow \exists^{=2} y \; (E(x,y)) \right)$

$\varphi(A) \in MSO(\mathfrak{I}_G)$

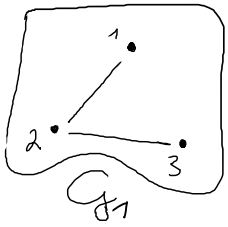
$(\varphi(A))(\mathcal{A}_{\mathfrak{I}_G}(G_1)) = ?$



$\varphi(A) := \forall x \left(A(x) \rightarrow \exists^{=2} y \; (E(x,y)) \right)$

$\varphi(A) \in MSO(\tau_G)$

$(\varphi(A))(\mathcal{A}_{\tau_G}(G_1)) = \{\langle 2 \rangle\}$

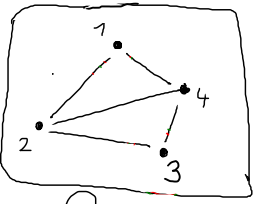


G_1

$\varphi(A) := \forall x \, (A(x) \rightarrow \exists^{=2} y \, (E(x,y)))$

$MSO(I_G)$

$(\varphi(A))(\mathcal{A}_{I_G}(G_1)) = \{\langle 2 \rangle\}$



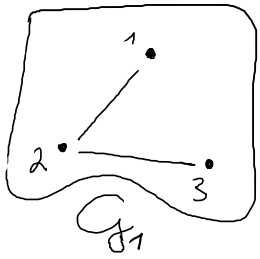
G_2

$\varphi(B) := \forall y \, (B(y) \rightarrow EDGE(y) \wedge$
$\wedge \exists x \, (INC(x,y) \wedge$
$\wedge \exists^{=2} e \, (EDGE(e) \wedge INC(x,e)))$

$MSO(I_{HG})$

$(\varphi(B))(\mathcal{A}_{I_{HG}}(G_2)) =$

$= ?$ {

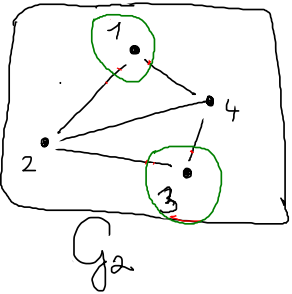


G_1

$\varphi(X) := \forall x \left(X(x) \rightarrow \exists^{=2} y \; (E(x,y)) \right)$

$\varphi(X) \in MSO(\mathcal{T}_G)$

$(\varphi(X))(\mathcal{A}_{\mathcal{T}_G}(G_1)) = \{\langle \{2\} \rangle\}$



G_2

$\varphi(Y) := \forall y \Big(Y(y) \rightarrow EDGE(y) \wedge$
$\wedge \exists x \, (VERT(x) \wedge INC(x,y)$
$\wedge \exists^{=2} e \, (INC(x,e))) \Big)$

$\varphi(Y) \in MSO(\mathcal{T}_{HG})$

$(\varphi(Y))(\mathcal{A}_{\mathcal{T}_{HG}}(G_2)) =$

$= \{\langle \{\{1,2\}, \{1,4\}, \{2,3\}, \{3,4\}\} \rangle\}$

Monadic second-order logic (formula example)

$$\psi_3 := \exists C_1 \exists C_2 \exists C_3 \Big(\Big(\forall x \Big(\bigvee_{i=1}^{3} C_i(x) \Big) \Big) \wedge \forall x \Big(\bigwedge_{1 \le i < j \le 3} \neg \Big(C_i(x) \wedge C_j(x) \Big) \Big) \\ \wedge \forall x \forall y \Big(E(x,y) \rightarrow \bigwedge_{i=1}^{3} \neg (C_i(x) \wedge C_i(y)) \Big) \Big)$$

$$\mathcal{A}(\mathcal{G}) \vDash \psi_3 \iff \mathcal{G} \text{ has is } 3\text{-colorable}.$$

Monadic second-order logic (formula example)

$$\psi_3 := \exists C_1 \exists C_2 \exists C_3 \Big(\big(\forall x \big(\bigvee_{i=1}^{3} C_i(x) \big)\big) \wedge \forall x \Big(\bigwedge_{1 \le i < j \le 3} \neg \big(C_i(x) \wedge C_j(x) \big) \Big) \wedge \forall x \forall y \big(E(x,y) \to \bigwedge_{i=1}^{3} \neg (C_i(x) \wedge C_i(y)) \big) \Big)$$

$$\equiv \exists C_1 \exists C_2 \exists C_3 \Big(\forall x (C_1(x) \vee C_2(x) \vee C_3(x)) \wedge \forall x \big(\neg (C_1(x) \wedge C_2(x)) \wedge \neg (C_1(x) \wedge C_3(x)) \wedge \neg (C_2(x) \wedge C_3(x)) \big) \wedge \forall x \forall y \big(E(x,y) \to \neg (C_1(x) \wedge C_1(y)) \wedge \neg (C_2(x) \wedge C_2(y)) \wedge \neg (C_3(x) \wedge C_3(y)) \big) \Big)$$

$$\mathcal{A}(\mathcal{G}) \models \psi_3 \iff \mathcal{G} \text{ has is } 3\text{-colorable}.$$

Expressing graph properties by MSO formulas (1)

Exercise

Express by a monadic second-order formula $\varphi(X)$ with one free unary predicate variable X over the vocabulary $\tau_{\mathsf{G}} = \{E/_2\}$ for graphs that for all graphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{\mathsf{G}}}(\mathcal{G}) \models \varphi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Recall:

$$\begin{aligned} S \subseteq V \text{ is independent set in } \mathcal{G} : &\iff \forall e = \{u, v\} \in E \, (\neg(u \in S \wedge v \in S)) \\ &\iff \forall u, v \in S (\, u \neq v \Rightarrow \{u, v\} \notin E \,) \end{aligned}$$

Exercise

Express the independent set property by a $\mathrm{MSO}(\tau_{\mathsf{HG}})$ formula ψ with vocabulary $\tau_{\mathsf{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$ for hypergraphs:

$$\mathcal{A}_{\tau_{\mathsf{HG}}}(\mathcal{G}) \models \psi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Expressing graph properties by MSO formulas (1)

Exercise

Express by a monadic second-order formula $\varphi(X)$ with one free unary predicate variable X over the vocabulary $\tau_{\mathsf{G}} = \{E/_2\}$ for graphs that for all graphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{\mathsf{G}}}(\mathcal{G}) \models \varphi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Exercise

Express the independent set property by a $\mathrm{MSO}(\tau_{\mathsf{HG}})$ formula ψ with vocabulary $\tau_{\mathsf{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$ for hypergraphs:

$$\mathcal{A}_{\tau_{\mathsf{HG}}}(\mathcal{G}) \models \psi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Expressing graph properties by MSO formulas (1)

Exercise

Express by a monadic second-order formula $\varphi(X)$ with one free unary predicate variable X over the vocabulary $\tau_{\mathsf{G}} = \{E/_2\}$ for graphs that for all graphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{\mathsf{G}}}(\mathcal{G}) \models \varphi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

$\varphi(S) :=$

$\psi(S) :=$

Exercise

Express the independent set property by a $\mathrm{MSO}(\tau_{\mathsf{HG}})$ formula ψ with vocabulary $\tau_{\mathsf{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$ for hypergraphs:

$$\mathcal{A}_{\tau_{\mathsf{HG}}}(\mathcal{G}) \models \psi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Expressing graph properties by MSO formulas (1)

Exercise

Express by a monadic second-order formula $\varphi(X)$ with one free unary predicate variable X over the vocabulary $\tau_G = \{E/_2\}$ for graphs that for all graphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_G}(\mathcal{G}) \models \varphi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

$\varphi(S) := \forall x \forall y \left(\neg x = y \wedge S(x) \wedge S(y) \rightarrow \neg E(x,y)\right) \qquad \in \mathrm{For}(\tau_G)$

$\varphi(S) :=$

Exercise

Express the independent set property by a $\mathrm{MSO}(\tau_{HG})$ formula ψ with vocabulary $\tau_{HG} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$ for hypergraphs:

$$\mathcal{A}_{\tau_{HG}}(\mathcal{G}) \models \psi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Expressing graph properties by MSO formulas (1)

Exercise

Express by a monadic second-order formula $\varphi(X)$ with one free unary predicate variable X over the vocabulary $\tau_G = \{E/_2\}$ for graphs that for all graphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_G}(\mathcal{G}) \models \varphi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

$\varphi(S) := \forall x \forall y \left(\neg x = y \wedge S(x) \wedge S(y) \rightarrow \neg E(x,y)\right)$ $\in For(\tau_G)$

$\varphi(S) := \forall x \left(S(x) \rightarrow VERT(x)\right) \wedge$

$\wedge \forall x \forall y \left(\neg x = y \wedge S(x) \wedge S(y) \rightarrow \neg \exists e \left(EDGE(e) \wedge INC(x,e) \wedge INC(y,e)\right)\right)$

$\in$

$For(\tau_{HG})$

Exercise

Express the independent set property by a MSO(τ_{HG}) formula ψ with vocabulary $\tau_{HG} = \{VERT/_1, EDGE/_1, INC/_2\}$ for hypergraphs:

$$\mathcal{A}_{\tau_{HG}}(\mathcal{G}) \models \psi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Expressing graph properties by MSO formulas (1)

Exercise

Express by a monadic second-order formula $\varphi(X)$ with one free unary predicate variable X over the vocabulary $\tau_{\mathsf{G}} = \{E/_2\}$ for graphs that for all graphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{\mathsf{G}}}(\mathcal{G}) \models \varphi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Exercise

Express the independent set property by a $\mathrm{MSO}(\tau_{\mathsf{HG}})$ formula ψ with vocabulary $\tau_{\mathsf{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$ for hypergraphs:

$$\mathcal{A}_{\tau_{\mathsf{HG}}}(\mathcal{G}) \models \psi(S) \iff S \subseteq V \text{ is an independent set in } \mathcal{G}$$

Expressing graph properties by MSO formulas (2)

Exercise

Express by a monadic second-order formula *feedback*(X) with one free unary predicate variable X over $\tau_{\text{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$, the vocabulary for graphs, that for all hypergraphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{\text{HG}}}(\mathcal{G}) \models \mathit{feedback}(S) \iff S \subseteq V \text{ is a feedback vertex set}$$

(A set $S \subseteq V$ is a feedback vertex set in $\mathcal{G}$ if S contains a vertex of every cycle of $\mathcal{G}$.)

Steps:

- Construct a formula *cycle-family*(X) that expresses the property of a set being the union of cycles.
- Using *cycle-family*(X), construct *feedback*(X).

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_G)$: MSO with vocabulary $\tau_G = \{E/_2\}$

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_{\mathrm{G}})$: MSO with vocabulary $\tau_{\mathrm{G}} = \{E/_2\}$
- $\mathrm{MSO}(\tau_{\mathrm{HG}})$: MSO with vocab. $\tau_{\mathrm{HG}} = \{VERT/_1, EDGE/_1, INC/_2\}$

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_G)$: MSO with vocabulary $\tau_G = \{E/_2\}$
- $\mathrm{MSO}(\tau_{HG})$: MSO with vocab. $\tau_{HG} = \{VERT/_1, EDGE/_1, INC/_2\}$
- MSO_1 :
  - vocabulary: $\{INC/_2\}$
  - quantifications: $\exists_{(vert)} x$ / $\forall_{(vert)} x$, $\exists_{(edge)} x$ / $\forall_{(edge)} x$, $\exists_{(vert)} X$ / $\forall_{(vert)} X$

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_{\mathrm{G}})$: MSO with vocabulary $\tau_{\mathrm{G}} = \{E/_2\}$
- $\mathrm{MSO}(\tau_{\mathrm{HG}})$: MSO with vocab. $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$
- MSO_1 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$
- MSO_2 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$, $\exists_{(\mathrm{edge})} X$ / $\forall_{(\mathrm{edge})} X$

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_{\mathrm{G}})$: MSO with vocabulary $\tau_{\mathrm{G}} = \{E/_2\}$
- $\mathrm{MSO}(\tau_{\mathrm{HG}})$: MSO with vocab. $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$
- MSO_1 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$
- MSO_2 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$, $\exists_{(\mathrm{edge})} X$ / $\forall_{(\mathrm{edge})} X$

Correspondences

$\mathrm{MSO}(\tau_{\mathrm{G}})$ corresponds to MSO_1

where 'corresponds to' means: 'is equally expressive as'.

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_{\mathrm{G}})$: MSO with vocabulary $\tau_{\mathrm{G}} = \{E/_2\}$
- $\mathrm{MSO}(\tau_{\mathrm{HG}})$: MSO with vocab. $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$
- MSO_1 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$
- MSO_2 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$, $\exists_{(\mathrm{edge})} X$ / $\forall_{(\mathrm{edge})} X$

Correspondences

$\mathrm{MSO}(\tau_{\mathrm{G}})$ corresponds to MSO_1

$\mathrm{MSO}(\tau_{\mathrm{HG}})$ corresponds to MSO_2

where 'corresponds to' means: 'is equally expressive as'.

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_{\mathrm{G}})$: MSO with vocabulary $\tau_{\mathrm{G}} = \{E/_2\}$
- $\mathrm{MSO}(\tau_{\mathrm{HG}})$: MSO with vocab. $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$
- MSO_1 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$
- MSO_2 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$, $\exists_{(\mathrm{edge})} X$ / $\forall_{(\mathrm{edge})} X$

Correspondences

$\mathrm{MSO}(\tau_{\mathrm{G}})$ corresponds to MSO_1

$\mathrm{MSO}(\tau_{\mathrm{HG}})$ corresponds to MSO_2

where 'corresponds to' means: 'is equally expressive as'.

Note:

$\mathrm{MSO}(\tau_{\mathrm{HG}})$ / MSO_2 are more expressive than $\mathrm{MSO}(\tau_{\mathrm{G}})$ / MSO_1.

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_{\mathrm{G}})$: MSO with vocabulary $\tau_{\mathrm{G}} = \{E/_2\}$
- $\mathrm{MSO}(\tau_{\mathrm{HG}})$: MSO with vocab. $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$
- MSO_1 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$
- MSO_2 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$, $\exists_{(\mathrm{edge})} X$ / $\forall_{(\mathrm{edge})} X$

Correspondences

$\mathrm{MSO}(\tau_{\mathrm{G}})$ corresponds to MSO_1

$\mathrm{MSO}(\tau_{\mathrm{HG}})$ corresponds to MSO_2

where 'corresponds to' means: 'is equally expressive as'.

Note:

We use MSO for either logic, restrict to $\mathrm{MSO}(\tau_{\mathrm{G}})$ / MSO_1 if needed.

MSO for graphs and hypergraphs

- $\mathrm{MSO}(\tau_{\mathrm{G}})$: MSO with vocabulary $\tau_{\mathrm{G}} = \{E/_2\}$
- $\mathrm{MSO}(\tau_{\mathrm{HG}})$: MSO with vocab. $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$
- MSO_1 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$
- MSO_2 :
  - vocabulary: $\{\mathit{INC}/_2\}$
  - quantifications: $\exists_{(\mathrm{vert})} x$ / $\forall_{(\mathrm{vert})} x$, $\exists_{(\mathrm{edge})} x$ / $\forall_{(\mathrm{edge})} x$, $\exists_{(\mathrm{vert})} X$ / $\forall_{(\mathrm{vert})} X$, $\exists_{(\mathrm{edge})} X$ / $\forall_{(\mathrm{edge})} X$

Correspondences

$\mathrm{MSO}(\tau_{\mathrm{G}})$ corresponds to MSO_1

$\mathrm{MSO}(\tau_{\mathrm{HG}})$ corresponds to MSO_2

where 'corresponds to' means: 'is equally expressive as'.

Note:

$\mathrm{MSO}(\tau_{\mathrm{HG}})$ / MSO_2 are more expressive than $\mathrm{MSO}(\tau_{\mathrm{G}})$ / MSO_1.

Expressing graph properties by MSO formulas (5)

Exercise

Express by a MSO(τ_{HG}) formula *conn*(X) with one free unary predicate variable X over $\tau_{\text{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$, the vocabulary for graphs, that for all hypergraphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{\text{HG}}}(\mathcal{G}) \models \mathit{hamiltonian} \iff \text{there is a Hamiltonian path in } \mathcal{G}.$$

Expressing graph properties by MSO formulas (5)

Exercise

Express by a MSO(τ_{HG}) formula *conn*(X) with one free unary predicate variable X over $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$, the vocabulary for graphs, that for all hypergraphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{\mathrm{HG}}}(\mathcal{G}) \models \mathit{hamiltonian} \iff \text{there is a Hamiltonian path in } \mathcal{G}.$$

Note:

- This property is not expressible by a (single) MSO(τ_{G}) formula.

Expressing graph properties by MSO formulas (5)

Exercise

Express by a $\mathrm{MSO}(\tau_{\mathrm{HG}})$ formula $\mathit{conn}(X)$ with one free unary predicate variable X over $\tau_{\mathrm{HG}} = \{\mathit{VERT}/_1, \mathit{EDGE}/_1, \mathit{INC}/_2\}$, the vocabulary for graphs, that for all hypergraphs $\mathcal{G} = \langle V, E \rangle$:

$$\mathcal{A}_{\tau_{\mathrm{HG}}}(\mathcal{G}) \models \mathit{hamiltonian} \iff \text{there is a Hamiltonian path in } \mathcal{G}.$$

Note:

- This property is not expressible by a (single) $\mathrm{MSO}(\tau_{\mathrm{G}})$ formula.
- Other properties that are not $\mathrm{MSO}(\tau_{\mathrm{G}})$ expressible:
  - balanced bipartite graphs
  - existence of a perfect matching
  - simple graphs (graphs with no parallel edges)
  - existence of spanning trees with maximum degree 3

Expressing graph properties by MSO formulas (5)

Exercise

$\mathcal{A}_{\tau_{\mathrm{HG}}}(\mathcal{G}) \models \textit{hamiltonian} \iff$ there is a Hamiltonian path in $\mathcal{G}$.

Expressing graph properties by MSO formulas (5)

Exercise

$\mathcal{A}_{\tau_{\mathrm{HG}}}(\mathcal{G}) \models \textit{hamiltonian} \iff$ there is a Hamiltonian path in $\mathcal{G}$.

Evaluation and model checking (MSO)

The *model checking problem* for MSO-formulas on labeled, ordered unranked trees:

MC(MSO, TREE_{lo})

Instance: A labeled, ordered, unranked Σ-tree $\mathcal{T}$, and a MSO(τ_Σ^u)-formula φ

Problem: Decide whether $\mathcal{T} \models \varphi$.

where for given alphabet Σ, $\tau_\Sigma^u := \{E/_2, N/_2\} \cup \{P_a/_1 \mid a \in \Sigma\}$.

Evaluation and model checking (MSO)

The *model checking problem* for MSO-formulas on labeled, ordered unranked trees:

MC(MSO, TREE_{lo})

Instance: A labeled, ordered, unranked Σ-tree $\mathcal{T}$, and a $\text{MSO}(\tau_\Sigma^u)$-formula φ

Problem: Decide whether $\mathcal{T} \models \varphi$.

where for given alphabet Σ, $\tau_\Sigma^u := \{E/_2, N/_2\} \cup \{P_a/_1 \mid a \in \Sigma\}$.

Theorem

MC(MSO, TREE_{lo}) $\in$ FPT.
More precisely, there is a computable function $f : \mathbb{N} \to \mathbb{N}$ *such that* MC(MSO, TREE_{lo}) *can be decided in time* $\leq O(f(|\varphi|) + \|\mathcal{T}\|)$.

Evaluation and model checking (MSO)

The *model checking problem* for MSO-formulas on labeled, ordered unranked trees:

MC(MSO, TREE_{lo})

Instance: A labeled, ordered, unranked Σ-tree $\mathcal{T}$, and a $\text{MSO}(\tau_\Sigma^u)$-formula φ

Problem: Decide whether $\mathcal{T} \models \varphi$.

where for given alphabet Σ, $\tau_\Sigma^u := \{E/_2, N/_2\} \cup \{P_a/_1 \mid a \in \Sigma\}$.

Theorem

MC(MSO, TREE_{lo}) $\in$ FPT.
More precisely, there is a computable function $f : \mathbb{N} \to \mathbb{N}$ *such that* MC(MSO, TREE_{lo}) *can be decided in time* $\leq O(f(|\varphi|) + \|\mathcal{T}\|)$.

Note that here: $f(k) \geq 2^{\cdot^{\cdot^{\cdot^{2}}}\}k}$ (a non-elementary function).

Courcelle's Theorem

Courcelle's Theorem for graphs

p^*-*tw*-MC(MSO)

Instance: A graph $\mathcal{G}$ and an MSO(τ_{HG})-sentence φ.
Parameter: $tw(\mathcal{G}) + |\varphi|$ (where $tw(\mathcal{G})$ the tree-width of $\mathcal{G}$)
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Courcelle's Theorem for graphs

p^*-*tw*-MC(MSO)

Instance: A graph $\mathcal{G}$ and an MSO(τ_{HG})-sentence φ.
Parameter: $tw(\mathcal{G}) + |\varphi|$ (where $tw(\mathcal{G})$ the tree-width of $\mathcal{G}$)
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Excursion:

FPT for sum of parameters $\triangleq$ FPT for multiple parameters

$f(k_1+k_2) \cdot p(|x|) \leq g(k_1,k_2) \cdot p(|x|)$
for $g(y,z) := f(y+z)$

Courcelle's Theorem for graphs

p^*-*tw*-MC(MSO)
Instance: A graph $\mathcal{G}$ and an MSO(τ_{HG})-sentence φ.
Parameter: $tw(\mathcal{G}) + |\varphi|$ (where $tw(\mathcal{G})$ the tree-width of $\mathcal{G}$)
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Excursion (discussed yesterday):

FPT for sum of parameters $\stackrel{\Delta}{=}$ FPT for multiple parameters

$f(k_1+k_2)\cdot p(|x|) \leq g(k_1,k_2)\cdot p(|x|)$ for $g(y,z) := f(y+z)$

$g(k_1,k_2)\cdot p(|x|) \leq g(k_1+k_2, k_1+k_2)\cdot p(|x|)$ if g monotonous

$= : f(k_1+k_2)$ where $f(y) := g(y,y)$

Courcelle's Theorem for graphs

p^*-*tw*-MC(MSO)

Instance: A graph $\mathcal{G}$ and an MSO(τ_{HG})-sentence φ.
Parameter: *tw*($\mathcal{G}$) + $|\varphi|$ (where *tw*($\mathcal{G}$) the tree-width of $\mathcal{G}$)
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Theorem (special case of Courcelle's Theorem)

p^*-*tw*-MC(MSO) $\in$ FPT. *More precisely, the problem is decidable, for some computable and non-decreasing function* $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *by an algorithm in time:*

$$f(k_1, k_2) \cdot n\,, \quad \textit{where } k_1 := \textit{tw}(\mathcal{A}),\ k_2 := |\varphi|,\ n := |V(\mathcal{G})|$$

Monadic second-order logic (formula example)

$$\psi_3 := \exists C_1 \exists C_2 \exists C_3 \Big(\big(\forall x \big(\bigvee_{i=1}^{3} C_i(x) \big)\big) \wedge \forall x \Big(\bigwedge_{1 \le i < j \le 3} \neg \big(C_i(x) \wedge C_j(x) \big) \Big) \wedge \forall x \forall y \big(E(x,y) \rightarrow \bigwedge_{i=1}^{3} \neg (C_i(x) \wedge C_i(y)) \big) \Big)$$

$$\equiv \exists C_1 \exists C_2 \exists C_3 \Big(\forall x (C_1(x) \vee C_2(x) \vee C_3(x)) \wedge \forall x \big(\neg (C_1(x) \wedge C_2(x)) \wedge \neg (C_1(x) \wedge C_3(x)) \wedge \neg (C_2(x) \wedge C_3(x)) \big) \wedge \forall x \forall y \big(E(x,y) \rightarrow \neg (C_1(x) \wedge C_1(y)) \wedge \neg (C_2(x) \wedge C_2(y)) \wedge \neg (C_3(x) \wedge C_3(y)) \big) \Big)$$

$\mathcal{A}(\mathcal{G}) \models \psi_3 \iff \mathcal{G}$ has is 3-colorable.

Courcelle's Theorem: applications (1)

p^*-*tw*-COLORABILITY $\in$ FPT
Instance: A graph $\mathcal{G}$ and $\ell \in \mathbb{N}$.
Parameter: $tw(\mathcal{C})$
Problem: Decide whether is $\mathcal{G}$ ℓ-colorable.

Example

- p^*-*tw*-3-COLORABILITY $\in$ FPT. ✓

We found $\varphi_{3\text{-col}}$ such that $A_G \models \varphi_{3\text{-col}} \Leftrightarrow G$ is 3-colorable

CT $\Rightarrow$ Checking $\varphi_{3\text{-col}}$ on A_G takes time $\leq f(tw(G), \underbrace{|\varphi_{3\text{-col}}|}_{\text{const}}) \cdot n$

Courcelle's Theorem: applications (1)

p^*-*tw*-COLORABILITY $\in$ FPT
Instance: A graph G and $\ell \in \mathbb{N}$.
Parameter: $tw(C)$ note ℓ is NOT a parameter
Problem: Decide whether is G ℓ-colorable.

Example

- p^*-*tw*-3-COLORABILITY $\in$ FPT.
- p^*-*tw*-COLORABILITY $\in$ FPT.

We found $\varphi_{col(\ell)}$ such that $A_G \models \varphi_{col(\ell)} \Leftrightarrow G$ is ℓ-colorable
$|\varphi_{col(\ell)}| \in O(\ell^2)$

Courcelle's Theorem: applications (1)

p^*-*tw*-COLORABILITY $\in$ FPT
Instance: A graph $\mathcal{G}$ and $\ell \in \mathbb{N}$.
Parameter: $tw(\mathcal{C})$ note ℓ is NOT a parameter
Problem: Decide whether is $\mathcal{G}$ ℓ-colorable.

Example

- p^*-*tw*-3-COLORABILITY $\in$ FPT.
- p^*-*tw*-COLORABILITY $\in$ FPT.

We found $\varphi_{col(\ell)}$ such that $A_G \models \varphi_{col(\ell)} \Leftrightarrow G$ is ℓ-colorable
$|\varphi_{col(\ell)}| \in O(\ell^2)$
CT $\Rightarrow$ model checking $\varphi_{col(\ell)}$ takes time $f(tw(G), O(\ell^2)) \cdot p(n)$ Cannot be used as parameter

Courcelle's Theorem: applications (1)

p^*-*tw*-COLORABILITY $\in$ FPT
Instance: A graph $\mathcal{G}$ and $\ell \in \mathbb{N}$.
Parameter: $tw(\mathcal{C})$ note ℓ is NOT a parameter
Problem: Decide whether is $\mathcal{G}$ ℓ-colorable.

Example

- p^*-*tw*-3-COLORABILITY $\in$ FPT.
- p^*-*tw*-COLORABILITY $\in$ FPT.

We found $\varphi_{col(\ell)}$ such that $\mathcal{A}_G \models \varphi_{col(\ell)} \Leftrightarrow G$ is ℓ-colorable

$|\varphi_{col(\ell)}| \in O(\ell^2)$

CT $\Rightarrow$ model checking $\varphi_{col(\ell)}$ takes time $f(tw(G)) \cdot O(\ell^2)$ cannot be used as parameter

However Lemma. If $tw(G) \leq k \Rightarrow G$ is $(k+1)$-colorable.

$\Rightarrow$ We only have check colorability for $\ell \leq tw(G)+1$!

Courcelle's Theorem: applications (1)

p^*-*tw*-COLORABILITY $\in$ FPT
Instance: A graph $\mathcal{G}$ and $\ell \in \mathbb{N}$.
Parameter: $tw(\mathcal{C})$ note ℓ is NOT a parameter
Problem: Decide whether is $\mathcal{G}$ ℓ-colorable.

Example

- p^*-*tw*-3-COLORABILITY $\in$ FPT.
- p^*-*tw*-COLORABILITY $\in$ FPT.

We found $\varphi_{col(\ell)}$ such that $\mathcal{A}_G \models \varphi_{col(\ell)} \Leftrightarrow G$ is ℓ-colorable

$|\varphi_{col(\ell)}| \in O(\ell^2)$

CT $\Rightarrow$ model checking $\varphi_{col(\ell)}$ takes time $f(tw(G)) \cdot O(L^2)$ Cannot be used as parameter

However Lemma. If $tw(G) \leq k \Rightarrow G$ is $(k+1)$-colorable.

$\Rightarrow$ We only have check colorability for $\ell \leq tw(G)+1$!

$\Rightarrow$ p^*-tw-3-COLORABILITY $\in$ FPT

Vertex Cover (first attempt)

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\Longleftrightarrow \forall e = \{u, v\} \in E \, (u \in S \vee v \in S))$

Vertex Cover (first attempt)

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\Longleftrightarrow \forall e = \{u, v\} \in E\,(\,u \in S \vee v \in S))$

p^*-*tw*-VERTEX-COVER
Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.
Instance: *tw*$(\mathcal{G})$.
Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ?

φ_{VC} $\quad \exists VC\,(\,\forall x\,\forall y\,(\,E(x,y) \rightarrow VC(x) \vee VC(y)) \in For(\tau_G)$

$\mathcal{A}_G \models \varphi_{VC} \iff G$ has a vertex cover

but is unhelpful, since every graph has (trivial) vertex cover V

Vertex Cover (first attempt)

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\iff \forall e = \{u, v\} \in E \, (u \in S \vee v \in S))$

p^*-*tw*-VERTEX-COVER
Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.
Instance: *tw*$(\mathcal{G})$.
Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ?

$$\varphi_{VC}(\ell) := \exists x_1 \dots \exists x_\ell \left(\forall y \forall z \left(E(y,z) \rightarrow \bigvee_{1 \le i \le \ell} y = x_i \vee z = x_i \right) \right)$$

Vertex Cover (first attempt)

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\iff \forall e = \{u, v\} \in E \, (u \in S \vee v \in S))$

p^*-*tw*-VERTEX-COVER
Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.
Instance: *tw*$(\mathcal{G})$.
Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ?

$$\varphi_{vc}(\ell) ::= \exists x_1 \dots \exists x_\ell \left(\forall y \forall z \left(E(y,z) \rightarrow \bigvee_{1 \le i \le \ell} y = x_i \vee z = x_i \right) \right)$$

$|\varphi_{vc}(\ell)| \in O(\ell)$

$A_{\mathcal{G}}(\mathcal{G}) \models \varphi_{vc}(\ell) \Leftrightarrow \mathcal{G}$ has a vertex cover of $\le \ell$ vertices

Vertex Cover (first attempt)

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\iff \forall e = \{u, v\} \in E\,(u \in S \vee v \in S))$

p^*-*tw*-VERTEX-COVER
Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.
Instance: *tw*$(\mathcal{G})$.
Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ?

$$\varphi_{VC}^{(e)} ::= \exists x_1 \dots \exists x_e \left(\forall y \forall z \left(E(y,z) \rightarrow \bigvee_{1 \le i \le e} y = x_i \vee z = x_i \right) \right)$$

$|\varphi_{VC}^{(e)}| \in O(e)$

$A_G(\mathcal{G}) \models \varphi_{VC}(e) \Leftrightarrow \mathcal{G}$ has a vertex cover of $\le \ell$ vertices

Courcelle's Theorem $\Rightarrow$ $\varphi_{VC}(e)$ checkable on $A_G(\mathcal{G})$ in time $f(tw(\mathcal{G}), \underbrace{|\varphi_{VC}|}_{\in O(e)}) \cdot |\mathcal{G}|$

} not FPT w.r.t parameter $tw(\mathcal{G})$!

Vertex Cover (first attempt)

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\Longleftrightarrow \forall e = \{u, v\} \in E \, (u \in S \vee v \in S))$

p^*-*tw*-VERTEX-COVER

Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.

Instance: $tw(\mathcal{G})$.

Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ ?

Courcelle's Theorem: applications (2)

p^*-*tw*-HAMILTONICITY

Instance: A graph $\mathcal{G}$

Parameter: $tw(\mathcal{C})$

Problem: Decide whether $\mathcal{G}$ is a hamiltonian (that is, contains a cyclic path that visits every vertex precisely once).

Example

p^*-*tw*-HAMILTONICITY $\in$ FPT.

Tree decompositions, and tree-width for graphs

Definition (recalling tree-width for graphs)

A *tree decomposition* of a graph $\mathcal{G} = \langle V, E \rangle$
is a pair $\langle \mathcal{T}, \{B_t\}_{t \in T} \rangle$ where $\mathcal{T} = \langle T, F \rangle$ a (undirected, unrooted) tree,
and $B_t \subseteq V$ for all $t \in T$ such that:

(T1) $A = \bigcup_{t \in T} B_t$ (every vertex of $\mathcal{G}$ is in some bag).

(T2) $(\forall \{u, v\} \in E)\,(\exists t \in T)\big[\, \{u, v\} \subseteq B_t \,\big]$
(the vertices of every edge of $\mathcal{G}$ are realized in some bag).

(T3) $(\forall v \in V)\,\big[$ subgraph of $\mathcal{T}$ defd. by $\{t \in T \mid v \in B_t\}$ is connected $\big]$
(the tree vertices (in $\mathcal{T}$) whose bags contain some vertex of $\mathcal{G}$
induce a subgraph of $\mathcal{T}$ that is connected).

The *width* of a tree dec. $\langle \mathcal{T}, \{B_t\}_{t \in T} \rangle$ is $\max\{|B_t| - 1 \mid t \in T\}$.

The *tree-width* $tw(\mathcal{A})$ of a τ-structure $\mathcal{A}$ is defined by:

$tw(\mathcal{A})$:= minimal width of a tree decomposition of $\mathcal{A}$.

Tree decompositions, and tree-width for structures

Definition (extension of tree-width to structures)

A *tree decomposition* of a τ-structure $\mathcal{A} = \langle A; \{R^{\mathcal{A}}\}_{R\in\tau}\rangle$ is a pair $\langle \mathcal{T}, \{B_t\}_{t\in T}\rangle$ where $\mathcal{T} = \langle T, F\rangle$ a (undirected, unrooted) tree, and $B_t \subseteq V$ for all $t \in T$ such that:

(T1) $A = \bigcup_{t\in T} B_t$ (every element of the universe of $\mathcal{A}$ is in some bag).

(T2) $(\forall R \in \tau)\,(\forall \langle a_1, \ldots, a_r\rangle \in R^{\mathcal{A}})(\exists t \in T)\big[\,\{a_1, \ldots, a_r\} \subseteq B_t\,\big]$

(the vertices of every 'hyperedge' in $R^{\mathcal{A}}$ are realized in some bag).

(T3) $(\forall v \in V)\,\big[$ subgraph of $\mathcal{T}$ defd. by $\{t \in T \mid v \in B_t\}$ is connected $\big]$

(the tree vertices (in $\mathcal{T}$) whose bags contain some vertex of $\mathcal{G}$ induce a subgraph of $\mathcal{T}$ that is connected).

The *width* of a tree dec. $\langle \mathcal{T}, \{B_t\}_{t\in T}\rangle$ is $\max\{|B_t| - 1 \mid t \in T\}$.

The *tree-width* $tw(\mathcal{A})$ of a τ-structure $\mathcal{A}$ is defined by:

$tw(\mathcal{A})$:= minimal width of a tree decomposition of $\mathcal{A}$.

Courcelle's Theorem

p^*-*tw*-MC(MSO)

Instance: A structure $\mathcal{A}$ and an MSO-sentence φ.

Parameter: $tw(\mathcal{A}) + |\varphi|$.

Problem: Decide whether $\mathcal{A} \models \varphi$.

Courcelle's Theorem

p^*-*tw*-MC(MSO)

Instance: A structure $\mathcal{A}$ and an MSO-sentence φ.
Parameter: $tw(\mathcal{A}) + |\varphi|$.
Problem: Decide whether $\mathcal{A} \models \varphi$.

Theorem ([Courcelle, 1990])

p^*-*tw*-MC(MSO) $\in$ FPT.

Courcelle's Theorem

p^*-*tw*-MC(MSO)

Instance: A structure $\mathcal{A}$ and an MSO-sentence φ.
Parameter: $tw(\mathcal{A}) + |\varphi|$.
Problem: Decide whether $\mathcal{A} \models \varphi$.

Theorem ([Courcelle, 1990])

p^*-*tw*-MC(MSO) $\in$ FPT. *More precisely, the problem is decidable by an algorithm in time:*

$$f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|), \quad \text{where } k_1 := tw(\mathcal{A}), \text{ and } k_2 := |\varphi|,$$

f computable and non-decreasing

Courcelle's Theorem

p^*-*tw*-MC(MSO)

Instance: A structure $\mathcal{A}$ and an MSO-sentence φ.
Parameter: $tw(\mathcal{A}) + |\varphi|$.
Problem: Decide whether $\mathcal{A} \models \varphi$.

Theorem ([Courcelle, 1990])

p^*-*tw*-MC(MSO) $\in$ FPT. *More precisely, the problem is decidable by an algorithm in time:*

$$f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|), \quad \text{where } k_1 := tw(\mathcal{A}), \text{ and } k_2 := |\varphi|,$$

f computable and non-decreasing

$f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|) \leq f(k_1, k_2) \cdot |A| + c \cdot \|\mathcal{A}\|$ with some $c > 0$

Courcelle's Theorem

p^*-*tw*-MC(MSO)

Instance: A structure $\mathcal{A}$ and an MSO-sentence φ.
Parameter: $tw(\mathcal{A}) + |\varphi|$.
Problem: Decide whether $\mathcal{A} \models \varphi$.

Theorem ([Courcelle, 1990])

p^*-*tw*-MC(MSO) $\in$ FPT. *More precisely, the problem is decidable by an algorithm in time:*

$$f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|), \quad \text{where } k_1 := tw(\mathcal{A}), \text{ and } k_2 := |\varphi|,$$

f computable and non-decreasing

$$f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|) \leq f(k_1, k_2) \cdot |A| + c \cdot \|\mathcal{A}\| \quad \text{with some } c > 0$$
$$\leq (f(k_1, k_2) + c) \cdot \|\mathcal{A}\|$$

Courcelle's Theorem

p^*-*tw*-MC(MSO)

Instance: A structure $\mathcal{A}$ and an MSO-sentence φ.
Parameter: $tw(\mathcal{A}) + |\varphi|$.
Problem: Decide whether $\mathcal{A} \models \varphi$.

Theorem ([Courcelle, 1990])

p^*-*tw*-MC(MSO) $\in$ FPT. *More precisely, the problem is decidable by an algorithm in time:*

$$f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|), \quad \text{where } k_1 := tw(\mathcal{A}), \text{ and } k_2 := |\varphi|,$$

f computable and non-decreasing

$$\begin{aligned} f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|) &\leq f(k_1, k_2) \cdot |A| + c \cdot \|\mathcal{A}\| && \text{with some } c > 0 \\ &\leq (f(k_1, k_2) + c) \cdot \|\mathcal{A}\| \\ &\leq g(k) \cdot (\|\mathcal{A}\| + |\varphi|) && \text{for } g(x) := f(x, x) + c \\ & && k := k_1 + k_2 \end{aligned}$$

Courcelle's Theorem

p^*-*tw*-MC(MSO)

Instance: A structure $\mathcal{A}$ and an MSO-sentence φ.
Parameter: $\mathit{tw}(\mathcal{A}) + |\varphi|$.
Problem: Decide whether $\mathcal{A} \models \varphi$.

Theorem ([Courcelle, 1990])

p^*-*tw*-MC(MSO) $\in$ FPT. *More precisely, the problem is decidable by an algorithm in time:*

$$f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|), \quad \text{where } k_1 := \mathit{tw}(\mathcal{A}), \text{ and } k_2 := |\varphi|,$$

f computable and non-decreasing

$$\begin{aligned} f(k_1,k_2)\cdot|A| + O(\|\mathcal{A}\|) &\le f(k_1,k_2)\cdot|A| + c\cdot\|\mathcal{A}\| && \text{with some } c>0 \\ &\le (f(k_1,k_2)+c)\cdot\|\mathcal{A}\| \\ &\le g(k)\cdot(\|\mathcal{A}\|+|\varphi|) && \text{for } g(x) := f(x,x)+c \\ & && k := k_1 + k_2 \\ &\le g(k)\cdot n && \text{where } n := \|\mathcal{A}\| + |\varphi| \end{aligned}$$

Courcelle's Theorem: Refinement 1

p^*-*tw*-MC$^{\leq}$(MSO)

Instance: A structure $\mathcal{A}$, an $\varphi(X)$, and $m \in \mathbb{N}$.

Parameter: $\mathit{tw}(\mathcal{A}) + |\varphi(X)|$.

Problem: Decide whether $\mathcal{A} \models \exists X(\, \mathit{card}^{\leq m}(X) \wedge \varphi(X)\,)$.

Refinement 1 of Courcelle's Theorem

p^*-*tw*-MC$^{\leq}$(MSO) $\in$ FPT. More precisely, the problem is decidable by an algorithm in time:

$$f(k_1, k_2) \cdot |A| + O(\|\mathcal{A}\|)\,, \quad \text{where } k_1 := \mathit{tw}(\mathcal{A}), \text{ and } k_2 := |\varphi|,$$

f computable and non-decreasing

Vertex Cover

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\Longleftrightarrow \forall e = \{u, v\} \in E\,(\,u \in S \vee v \in S)$

p^*-*tw*-VERTEX-COVER
Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.
Instance: $tw(\mathcal{G})$.
Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ?

Example

p^*-*tw*-VERTEX-COVER $\in$ FPT.

$\varphi_{VC}(VC) := \forall x \forall y\,(\,E(x,y) \longrightarrow V(x) \vee V(y))$

$MSO(\mathcal{T}_G)$
$=$
MSO_1-formulas

Vertex Cover

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\iff \forall e = \{u, v\} \in E\,(u \in S \vee v \in S))$

p^*-*tw*-VERTEX-COVER
Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.
Instance: $tw(\mathcal{G})$.
Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ?

Example

p^*-*tw*-VERTEX-COVER $\in$ FPT.

$\varphi_{VC}(VC) := \forall x \forall y\,(E(x,y) \rightarrow VC(x) \vee VC(y))$

$A_G(\mathcal{G}) \models \varphi(S) \iff S \subseteq V$ is a vertex-cover

$A_G(\mathcal{G}) = \langle V, E \rangle$

$MSO(\tau_G)$
$=$
MSO_1-formulas

Vertex Cover

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\Longleftrightarrow \forall e = \{u, v\} \in E\,(u \in S \vee v \in S))$

p^*-*tw*-VERTEX-COVER
Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.
Instance: *tw*$(\mathcal{G})$.
Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ?

Example

p^*-*tw*-VERTEX-COVER $\in$ FPT.

$\varphi_{vc}(VC) := \forall x \forall y\,(E(x,y) \rightarrow VC(x) \vee VC(y))$

$\mathcal{A}_G(\mathcal{G}) \models \varphi(S) \Leftrightarrow S \subseteq V$ is a vertex-cover

$\mathcal{A}_G(\mathcal{G}) = \langle V, E \rangle$

$MSO(\tau_G) = MSO_1$-formulas

checking whether $\varphi_{vc}(S)$ holds in $\mathcal{A}_G(\mathcal{G})$ for a set S with $|S| \leq \ell$ now possible in $\leq f(tw(\mathcal{G}), |\varphi_{vc}|) \cdot |\mathcal{G}|$ steps

constant does not depend on ℓ

Courcelle's Theorem: Refinement 2

Counting version

p^*-*tw*-MC$^{=}$(MSO)

Instance: A structure $\mathcal{A}$, an MSO-sentence $\varphi(X)$, and $m \in \mathbb{N}$.
Parameter: $\mathit{tw}(\mathcal{A}) + |\varphi(X)|$.
Problem: Decide whether $\mathcal{A} \models \exists X(\, \mathit{card}^{=m}(X) \wedge \varphi(X)\,)$.

Refinement 2 of Courcelle's Theorem

p^*-*tw*-MC$^{=}$(MSO) $\in$ FPT. More precisely, the problem is decidable by an algorithm in time:

$$f(k_1, k_2) \cdot |A|^2 + O(\|\mathcal{A}\|), \quad \text{where } k_1 := \mathit{tw}(\mathcal{A}), \text{ and } k_2 := |\varphi|,$$

f computable and non-decreasing

Courcelle's Theorem Ref. 3: Optimization version

p^*-*tw*-opt-MC(MSO)

Instance: A graph $\mathcal{G} = \langle V, E \rangle$, an MSO-sentence $\varphi(X_1, \ldots, X_p)$.

Parameter: $tw(\mathcal{G}) + |\varphi(X_1, \ldots, X_p)|$.

Compute: $\min_{\max} \{ \alpha(|X_1|, \ldots, |X_p|) \mid \begin{matrix} X_1, \ldots, X_p \subseteq V \cup E \\ \mathcal{A}(\mathcal{G}) \models \varphi(X_1, \ldots, X_p). \end{matrix} \}$.

where α is an affine function $\alpha(x_1, \ldots, x_p) = a_0 + \sum_{i=1}^{p} a_i \cdot x_i$.

Optimization version of Courcelle's Theorem

p^*-*tw*-opt-MC(MSO) $\in$ FPT, and it is decidable by an algorithm in time:

$f(tw(\mathcal{G}), |\varphi|) \cdot |V|$, where f computable and non-decreasing.

Maximum 2-edge colorable subgraphs

p^*-*tw*-max-2-edge-colorable-subgraph

Instance: A graph $\mathcal{G} = \langle V, E \rangle$.

Parameter: $tw(\mathcal{G})$.

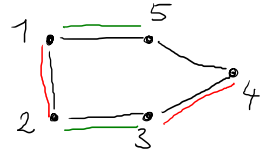
Compute: Maximum number of edges in a 2-edge colored subgraph of G.

Example [AA & Vahan Mkrtchyan]

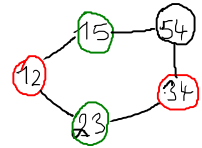
p^*-*tw*-max-2-edge-colorable-subgraph $\in$ FPT.

Maximum 2-edge colorable subgraphs

p^*-*tw*-max-2-edge-colorable-subgraph
Instance: A graph $\mathcal{G} = \langle V, E \rangle$.
Parameter: $tw(\mathcal{G})$.
Compute: Maximum number of edges in a 2-edge colored subgraph of G.



graph G



"line graph" L(G)

Example [AA & Vahan Mkrtchyan]

p^*-*tw*-max-2-edge-colorable-subgraph $\in$ FPT.

Maximum 2-edge colorable subgraphs

p^*-*tw*-max-2-edge-colorable-subgraph
Instance: A graph $\mathcal{G} = \langle V, E \rangle$.
Parameter: $tw(\mathcal{G})$.
Compute: Maximum number of edges in a 2-edge colored subgraph of G.

$S \subseteq E$ is 2-colorable ⇕

Example [AA & Vahan Mkrtchyan]
p^*-*tw*-max-2-edge-colorable-subgraph $\in$ FPT.

We find an MSO_2-formula $\varphi_{2ec}(S)$ such that $(A_{IHG} \models \varphi(S))$

$$\varphi_{2ec}(S) := \forall e\,(S(e) \rightarrow EDGE(e))$$
$$\wedge \exists C_1 \exists C_2 [(\text{edge-set-partition}(C_1, C_2, S)$$
$$\wedge \forall e_1 \forall e_2 (EDGE(e_1) \wedge EDGE(e_2) \wedge \neg e_1 = e_2$$
$$\wedge \exists v (INC(v, e_1), INC(v, e_2))$$
$$\rightarrow \neg (C_1(e_1) \wedge C_1(e_2))$$
$$\wedge \neg (C_2(e_1) \wedge C_2(e_2))) \square$$

Maximum 2-edge colorable subgraphs

p^*-*tw*-max-2-edge-colorable-subgraph
Instance: A graph $\mathcal{G} = \langle V, E \rangle$.
Parameter: $tw(\mathcal{G})$.
Compute: Maximum number of edges in a 2-edge colored subgraph of G.

$S \subseteq E$ is 2-colorable $\Leftrightarrow$

Example [AA & Vahan Mkrtchyan]
p^*-*tw*-max-2-edge-colorable-subgraph $\in$ FPT.

We find an MSO_2-formula $\varphi_{2ec}(S)$ such that $(\mathcal{A}_{TH_G} \models \varphi(S))$

$\varphi_{2ec}(X) := \forall e\,(X(e) \rightarrow EDGE(e))$
$\wedge\, \exists C_1 \exists C_2 [($ edge-set-partition (C_1, C_2, X)
$\wedge\, \forall e_1 \forall e_2 (\, EDGE(e_1) \wedge EDGE(e_2) \wedge \neg e_1 = e_2$
$\wedge\, \exists v (\, INC(v, e_1), INC(v, e_2))$
$\rightarrow \neg (C_1(e_1) \wedge C_1(e_2))$
$\wedge \neg (C_2(e_1) \wedge C_2(e_2))\,]$

edge-set-partition $(C_1, C_2, X) :=$
$\forall x\,(X(x) \rightarrow EDGE(x))$
$\wedge\, \forall x\,(EDGE(x) \rightarrow (X(x) \leftrightarrow C_1(x) \vee C_2(x))$
$\wedge \neg (C_1(x) \wedge C_2(x)))$

Maximum 2-edge colorable subgraphs

p^*-*tw*-max-2-edge-colorable-subgraph
Instance: A graph $\mathcal{G} = \langle V, E \rangle$.
Parameter: $tw(\mathcal{G})$.
Compute: Maximum number of edges in a 2-edge colored subgraph of G.

$S \subseteq E$ is 2-colorable $\Leftrightarrow$

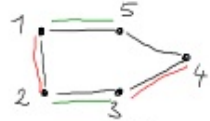
Example [AA & Vahan Mkrtchyan]
p^*-*tw*-max-2-edge-colorable-subgraph $\in$ FPT.

We find an MSO_2-formula $\varphi_{2ec}(S)$ such that $(A_{TMG} \models \varphi(S))$

$\varphi_{2ec}(X) := \forall e \, (X(e) \rightarrow EDGE(e))$
$\wedge \exists C_1 \exists C_2 [($ edge-set-partition (C_1, C_2, X)
$\wedge \forall e_1 \forall e_2 (EDGE(e_1) \wedge EDGE(e_2) \wedge \neg e_1 = e_2$
$\wedge \exists v (INC(v, e_1), INC(v, e_2))$
$\rightarrow \neg (C_1(e_1) \wedge C_1(e_2))$
$\wedge \neg (C_2(e_1) \wedge C_2(e_2))]$

edge-set-partition $(C_1, C_2, X) :=$
$\forall x \, (X(x) \rightarrow EDGE(x))$
$\wedge \forall x \, (EDGE(x) \rightarrow (X(x) \leftrightarrow C_1(x) \vee C_2(x))$
$\wedge \neg (C_1(x) \wedge C_2(x)))$

Maximum 2-edge colorable subgraphs



graph G

"line graph" $L(G)$

$S \subseteq E$ is 2-colorable

$\Updownarrow$

We find an MSO_2-formula $\varphi_{2ec}(S)$ such that $(A_{J_{HG}} \models \varphi(S))$

$\varphi_{2ec}(X) := \forall e\,(X(e) \rightarrow EDGE(e))$
$\wedge \exists C_1 \exists C_2 [(\text{edge-set-partition}(C_1, C_2, X)$
$\wedge \forall e_1 \forall e_2\,(EDGE(e_1) \wedge EDGE(e_2) \wedge \neg e_1 = e_2$
$\wedge \exists v (INC(v, e_1), INC(v, e_2))$
$\rightarrow \neg(C_1(e_1) \wedge C_1(e_2))$
$\wedge \neg(C_2(e_1) \wedge C_2(e_2))]$

edge-set-partition$(C_1, C_2, X) :=$
$\forall x\,(X(x) \rightarrow EDGE(x))$
$\wedge \forall x\,(EDGE(x) \rightarrow (X(x) \leftrightarrow C_1(x) \vee C_2(x))$
$\wedge \neg(C_1(x) \wedge C_2(x)))$

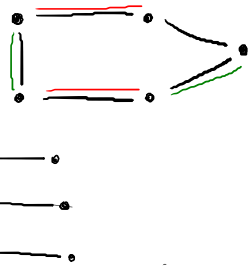
p^*-*tw*-max-2-edge-colorable-subgraph
Instance: A graph $\mathcal{G} = \langle V, E \rangle$.
Parameter: $tw(\mathcal{G})$.
Compute: Maximum number of edges in a 2-edge colored subgraph of G.

Example [AA & Vahan Mkrtchyan]
p^*-*tw*-max-2-edge-colorable-subgraph $\in$ FPT.

Maximization version of CT $\Rightarrow$ p^*-tw-max-2ecs $\in$ FPT

Maximum 2-edge colorable subgraphs

p^*-*tw*-max-2-edge-colorable-subgraph
Instance: A graph $\mathcal{G} = \langle V, E \rangle$.
Parameter: $tw(\mathcal{G})$.
Compute: Maximum number of edges in a 2-edge colored subgraph of G.



a matching is a set of vertices all of which have different endpoints

Example [AA & Vahan Mkrtchyan]

p^*-*tw*-max-2-edge-colorable-subgraph $\in$ FPT.

$\Phi_{2ec}(X) := \forall y\,(X(y) \rightarrow EDGES(y))$
$\wedge\ \exists R\ \exists G\ (\text{"}R \cup G = S \wedge R \cap G = \{\}\text{"} \wedge Matching(R) \wedge Matching(G))$
(R: red edges, G: green edges)

Note: MSO_2-formula

$R \cup G = S : \forall x\,[EDGE(x) \rightarrow (R(x) \vee G(x) \leftrightarrow S(x)) \wedge \neg(R(x) \wedge G(x))]$
$\wedge\ R \cap G = \emptyset$

$Matching(R) := (\forall e_1 \in E)(\forall e_2 \in E)\ \neg\exists v \in V\,(INC(v, e_1) \wedge INC(v, e_2))$

Courcelle's Theorem: applications (4)

p^*-*tw*-FEEDBACK-VERTEX-SET

Instance: A graph $\mathcal{G}$ and $\ell \in \mathbb{N}$.
Parameter: $tw(\mathcal{C})$
Problem: Decide whether $\mathcal{G}$ has a feedback vertex set of ℓ elements.

Example

p^*-*tw*-FEEDBACK-VERTEX-SET $\in$ FPT.

Courcelle's Theorem: applications (5)

p^*-*tw*-CROSSING-NUMBER
Instance: A graph $\mathcal{G}$, and $k \in \mathbb{N}$
Parameter: $tw(\mathcal{G}) + k$
Problem: Decide whether the crossing number of $\mathcal{G}$ is k.

Example

p^*-*tw*-CROSSING-NUMBER $\in$ FPT.

The *crossing number* is the least number of edge crossings required to draw the graph in the plane such that at each point at most two edges cross.

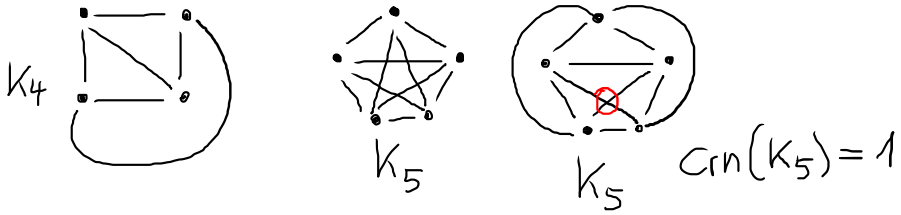
Courcelle's Theorem: applications (5)

p^*-*tw*-CROSSING-NUMBER
Instance: A graph $\mathcal{G}$, and $k \in \mathbb{N}$
Parameter: $tw(\mathcal{G}) + k$
Problem: Decide whether the crossing number of $\mathcal{G}$ is k.

Example

p^*-*tw*-CROSSING-NUMBER $\in$ FPT.

The *crossing number* is the least number of edge crossings required to draw the graph in the plane such that at each point at most two edges cross.



Courcelle's Theorem: applications (5)

Definition

Let $\mathcal{G}_1 = \langle V_1, E_1 \rangle$ and $\mathcal{G}_2 = \langle V_2, E_2 \rangle$ be graphs.
$\mathcal{G}_1$ is a *subdivision* of $\mathcal{G}_2$ if:

- $\mathcal{G}_1$ arises by splitting the edges of $\mathcal{G}_2$ into paths with intermediate vertices.

$\mathcal{H}$ is a *topological subgraph* of $\mathcal{G}$ if $\mathcal{G}$ has a subgraph that is a subdivision of $\mathcal{H}$.

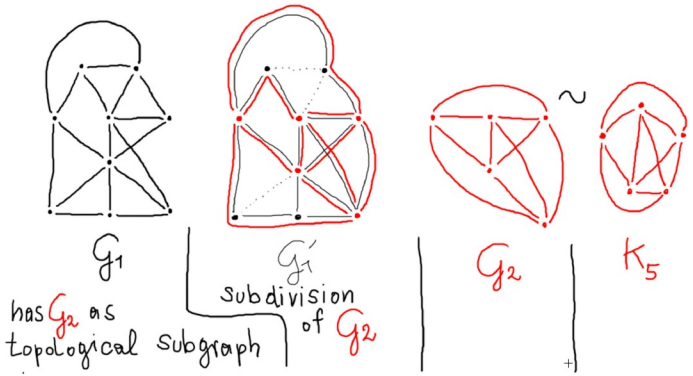
Courcelle's Theorem: applications (5)

Definition

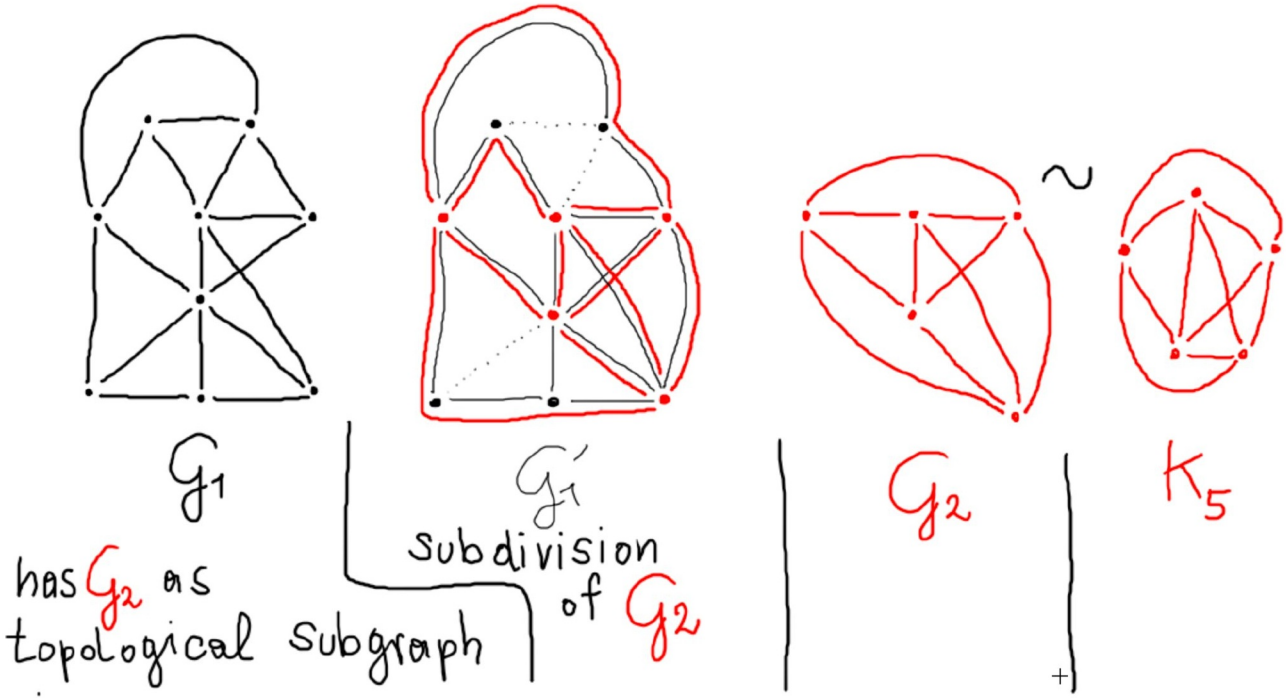
Let $\mathcal{G}_1 = \langle V_1, E_1 \rangle$ and $\mathcal{G}_2 = \langle V_2, E_2 \rangle$ be graphs.
$\mathcal{G}_1$ is a *subdivision* of $\mathcal{G}_2$ if:

- $\mathcal{G}_1$ arises by splitting the edges of $\mathcal{G}_2$ into paths with intermediate vertices.

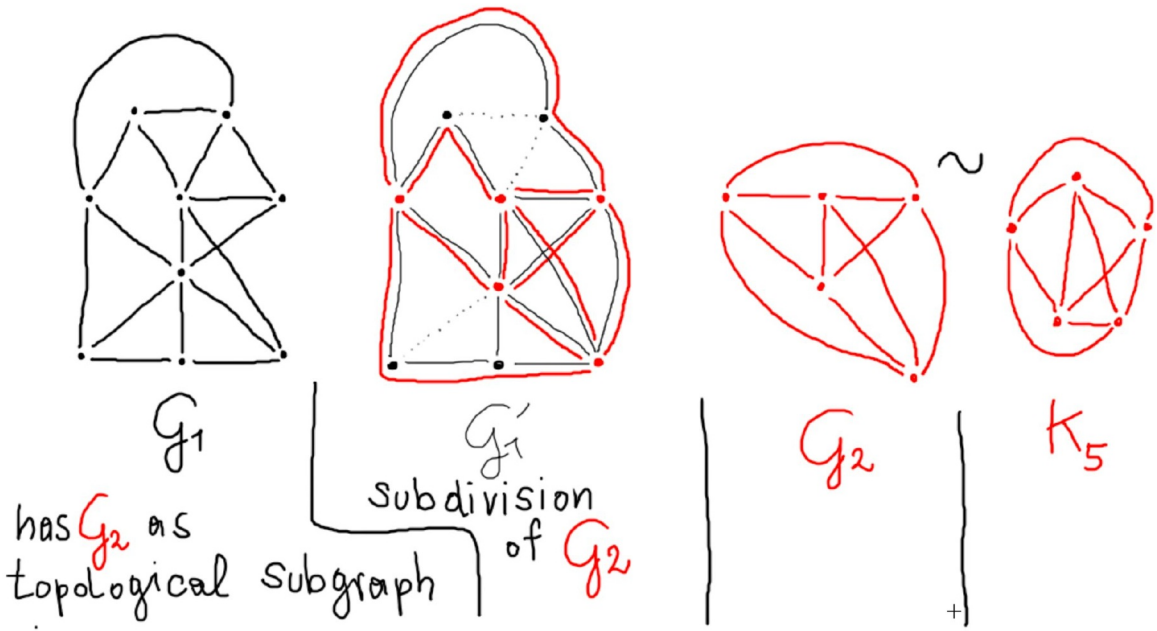
$\mathcal{H}$ is a *topological subgraph* of $\mathcal{G}$ if $\mathcal{G}$ has a subgraph that is a subdivision of $\mathcal{H}$.

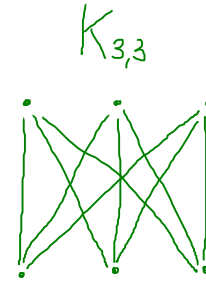


Courcelle's Theorem: applications (5)



Courcelle's Theorem: applications (5)





Theorem (Kuratowski)

A graph is planar if and only if it contains neither $\mathcal{K}_5$ nor $\mathcal{K}_{3,3}$ as topological subgraph.

Courcelle's Theorem: applications (5)

Theorem (Kuratowski)

A graph is planar if and only if it contains neither $\mathcal{K}_5$ nor $\mathcal{K}_{3,3}$ as topological subgraph.

Courcelle's Theorem: applications (5)

Theorem (Kuratowski)

A graph is planar if and only if it contains neither $\mathcal{K}_5$ nor $\mathcal{K}_{3,3}$ as topological subgraph.

Lemma

There is a MSO(τ_{HG}) formula *top-sub*$_{\mathcal{H}}$ such that for every graph $\mathcal{G}$:

$$\mathcal{A}_{\tau_{\text{HG}}}(\mathcal{G}) \models \textit{top-sub}_{\mathcal{H}} \iff \mathcal{H} \text{ is a topological subgraph of } \mathcal{G}.$$

Courcelle's Theorem: applications (5)

Theorem (Kuratowski)

A graph is planar if and only if it contains neither $\mathcal{K}_5$ nor $\mathcal{K}_{3,3}$ as topological subgraph.

Lemma

There is a MSO(τ_{HG}) formula *top-sub*$_{\mathcal{H}}$ such that for every graph $\mathcal{G}$:

$$\mathcal{A}_{\tau_{\text{HG}}}(\mathcal{G}) \models \textit{top-sub}_{\mathcal{H}} \iff \mathcal{H} \text{ is a topological subgraph of } \mathcal{G}.$$

Using MSO(τ_{HG}) formula *path*(x, y, Z) that Z is a path from x to y.

Courcelle's Theorem: applications (5)

Theorem (Kuratowski)

A graph is planar if and only if it contains neither $\mathcal{K}_5$ nor $\mathcal{K}_{3,3}$ as topological subgraph.

Lemma

There is a MSO(τ_{HG}) formula *top-sub*$_{\mathcal{H}}$ such that for every graph $\mathcal{G}$:

$$\mathcal{A}_{\tau_{\text{HG}}}(\mathcal{G}) \models \textit{top-sub}_{\mathcal{H}} \iff \mathcal{H} \text{ is a topological subgraph of } \mathcal{G}.$$

Using MSO(τ_{HG}) formula *path*(x, y, Z) that Z is a path from x to y.

Lemma

There is a MSO(τ_{HG}) formula *cross*$_k$ such that for every graph $\mathcal{G}$:

$$\mathcal{A}_{\tau_{\text{HG}}}(\mathcal{G}) \models \textit{cross}_k \iff \text{the crossing number of } \mathcal{G} \text{ is at most } k.$$

Proof: By induction, where $\textit{cross}_0 := \neg \textit{top-sub}_{\mathcal{K}_5} \wedge \neg \textit{top-sub}_{\mathcal{K}_{3,3}}$.

Courcelle's Theorem: applications (5)

p^*-*tw*-CROSSING-NUMBER
Instance: A graph $\mathcal{G}$, and $k \in \mathbb{N}$
Parameter: $tw(\mathcal{G}) + k$
Problem: Decide whether the crossing number of $\mathcal{G}$ is k.

Example

p^*-*tw*-CROSSING-NUMBER $\in$ FPT.

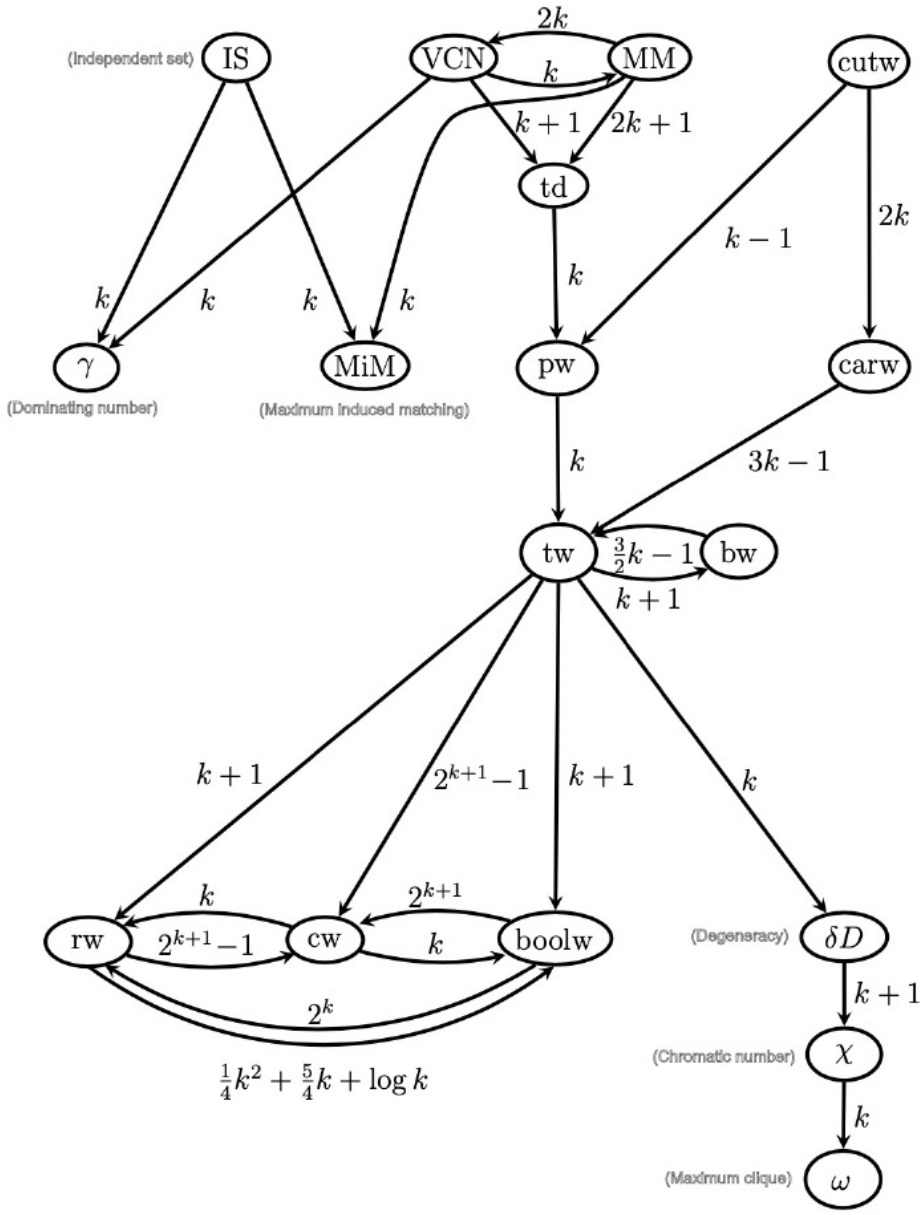
The *crossing number* is the least number of edge crossings required to draw the graph in the plane such that at each point at most two edges cross.

Computably boundedness between notions of width

(from Sasák, [Sásak, 2010])

$g(wd_1) \geq wd_2 \ : \Leftrightarrow \ wd_1 \xrightarrow{g} wd_2$

- FPT-results
  transfer upwards
  (and conversely to $\xrightarrow{g}$)
- ($\notin$ FPT)-results
  transfer downwards
  (and along $\xrightarrow{g}$)



Comparing parameterizations

Definition (computably bounded below)

Let $\kappa_1, \kappa_2 : \Sigma^* \to \mathbb{N}$ parameterizations.

- $\kappa_1 \geq \kappa_2 \iff \exists g : \mathbb{N} \to \mathbb{N}$ computable $\forall x \in \Sigma^* \big[\, g(\kappa_1(x)) \geq \kappa_2(x) \,\big]$.
- $\kappa_1 \approx \kappa_2 \iff \kappa_1 \geq \kappa_2 \;\wedge\; \kappa_2 \geq \kappa_1$.
- $\kappa_1 > \kappa_2 \iff \kappa_1 \geq \kappa_2 \;\wedge\; \neg\big(\kappa_2 \geq \kappa_1\big)$.

Proposition

For all parameterized problems $\langle Q, \kappa_1 \rangle$ and $\langle Q, \kappa_2 \rangle$ with parameterizations $\kappa_1, \kappa_2 : \Sigma^* \to \mathbb{N}$ with $\kappa_1 \geq \kappa_2$:

$$\langle Q, \kappa_1 \rangle \in \text{FPT} \iff \langle Q, \kappa_2 \rangle \in \text{FPT}$$
$$\langle Q, \kappa_1 \rangle \notin \text{FPT} \implies \langle Q, \kappa_2 \rangle \notin \text{FPT}$$

Courcelle's Theorem for clique-width

Recall that $\mathrm{MSO}(\tau_{\mathsf{G}}) \sim \mathrm{MSO}_1$ (quantification over sets of vertices, but not sets of edges).

p^*-*clw*-MC($\mathrm{MSO}(\tau_{\mathsf{G}})/\mathrm{MSO}_1$)
Instance: A graph $\mathcal{G}$ and an $\mathrm{MSO}(\tau_{\mathsf{G}})$-sentence φ.
Parameter: $\mathit{clw}(\mathcal{G}) + |\varphi|$.
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Theorem ([Courcelle et al., 2000])

p^*-*clw*-MC($\mathrm{MSO}(\tau_{\mathsf{G}})/\mathrm{MSO}_1$) $\in$ FPT.

Also, there is a maximization version of this theorem.

Courcelle's Theorem for clique-width (example)

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\Longleftrightarrow \forall e = \{u, v\} \in E \, (\, u \in S \vee v \in S) \,)$

p^*-*clw*-VERTEX-COVER

Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.

Instance: *clw*$(\mathcal{G})$.

Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ ?

Example

p^*-*clw*-VERTEX-COVER $\in$ FPT.

Courcelle's Theorem for clique-width (example)

Let $\mathcal{G} = \langle V, E \rangle$ a graph. For all $S \subseteq V$:

S is a vertex cover of $\mathcal{G} :\Longleftrightarrow \forall e = \{u, v\} \in E \, (\, u \in S \vee v \in S) \,)$

p^*-*clw*-VERTEX-COVER

Instance: A graph $\mathcal{G} = \langle V, E \rangle$, and $\ell \in \mathbb{N}$.

Instance: *clw*$(\mathcal{G})$.

Problem: Does $\mathcal{G}$ have a vertex cover of size at most ℓ ?

Example

p^*-*clw*-VERTEX-COVER $\in$ FPT.

Application to maximum 2-edge colorable subgraphs?

p^*-*clw*-max-2-edge-colorable-subgraph

Instance: A graph $\mathcal{G} = \langle V, E \rangle$.

Parameter: *clw*($\mathcal{G}$).

Compute: Maximum number of edges in a 2-edge colored subgraph of G.

Open problem [AA, Vahan Mkrtchyan]

p^*-*clw*-max-2-edge-colorable-subgraph $\in$ FPT ?

Application to maximum 2-edge colorable subgraphs?

p^*-*clw*-max-2-edge-colorable-subgraph

Instance: A graph $\mathcal{G} = \langle V, E \rangle$.

Parameter: *clw*$(\mathcal{G})$.

Compute: Maximum number of edges in a 2-edge colored subgraph of G.

Open problem [AA, Vahan Mkrtchyan]

p^*-*clw*-max-2-edge-colorable-subgraph $\in$ FPT ?

We saw that there is a MSO_2 formula $\varphi(X)$ such that:

$$\mathcal{A}_{\tau_{\mathrm{HG}}}(\mathcal{G}) \models \varphi(S) \iff S \subseteq E \text{ is an 2-colorable set of edges in } \mathcal{G}$$

But there seems not to be such an MSO_1 formula.

Courcelle's Theorem for clique-width (non-example)

p^*-*clw*-HAMILTONICITY

Instance: A graph $\mathcal{G}$

Parameter: *clw*$(\mathcal{C})$

Problem: Decide whether $\mathcal{G}$ is a hamiltonian (that is, contains a cyclic path that visits every vertex precisely once).

Courcelle's Theorem for clique-width (non-example)

p^*-*clw*-HAMILTONICITY

Instance: A graph $\mathcal{G}$

Parameter: *clw*$(\mathcal{C})$

Problem: Decide whether $\mathcal{G}$ is a hamiltonian (that is, contains a cyclic path that visits every vertex precisely once).

Recall

There is no MSO_1 formula that expresses Hamiltonicity.

Courcelle's Theorem for clique-width (non-example)

p^*-*clw*-HAMILTONICITY

Instance: A graph $\mathcal{G}$

Parameter: *clw*($\mathcal{C}$)

Problem: Decide whether $\mathcal{G}$ is a hamiltonian (that is, contains a cyclic path that visits every vertex precisely once).

Recall

There is no MSO_1 formula that expresses Hamiltonicity.

Hence we cannot apply Courcelle's Theorem for clique-width.

Courcelle's Theorem for clique-width (non-example)

p^*-*clw*-HAMILTONICITY
Instance: A graph $\mathcal{G}$
Parameter: *clw*($\mathcal{C}$)
Problem: Decide whether $\mathcal{G}$ is a hamiltonian (that is, contains a cyclic path that visits every vertex precisely once).

Recall

There is no MSO_1 formula that expresses Hamiltonicity.

Hence we cannot apply Courcelle's Theorem for clique-width. Indeed:

Theorems

(T1) p^*-*clw*-HAMILTONICITY $\notin$ FPT,
since it is not decidable in time $\notin n^{o(\mathit{clw}(\mathcal{C}))}$ (Fomin et al, 2014).

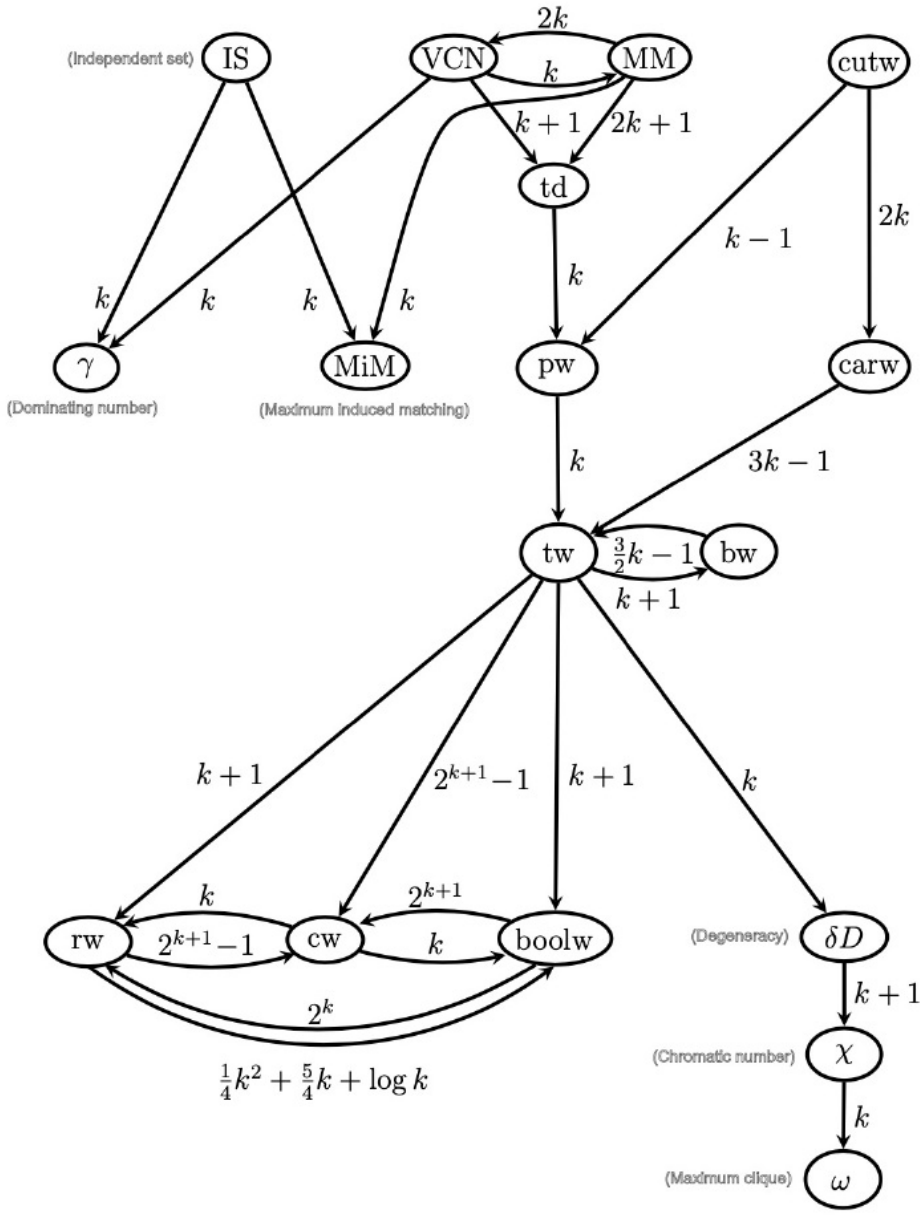
(T2) p^*-*clw*-HAMILTONICITY $\in O(n^{o(\mathit{clw}(\mathcal{C}))})$
(Bergougnoux, Kanté, Kwon, 2020).

Computably boundedness between notions of width

(from Sasák, [Sásak, 2010])

$g(wd_1) \geq wd_2 \ : \Leftrightarrow \ wd_1 \xrightarrow{g} wd_2$

- FPT-results
  transfer upwards
  (and conversely to $\xrightarrow{g}$)
- ($\notin$ FPT)-results
  transfer downwards
  (and along $\xrightarrow{g}$)



Graph Minors

Graph minors

Definition

A graph $\mathcal{G}_0$ is a *minor* of a graph $\mathcal{G}$ if $\mathcal{G}_0$ is obtained by:

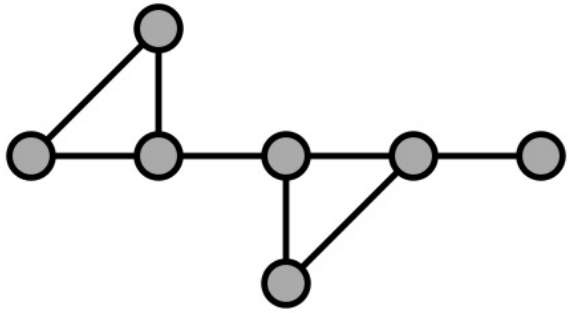
- deleting some edges,
- deleting arising isolated vertices,
- contracting edges in $\mathcal{G}$.

Graph minors

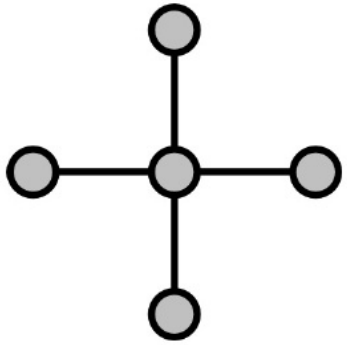
Definition

A graph $\mathcal{G}_0$ is a *minor* of a graph $\mathcal{G}$ if $\mathcal{G}_0$ is obtained by:

- deleting some edges,
- deleting arising isolated vertices,
- contracting edges in $\mathcal{G}$.



$\mathcal{G}$



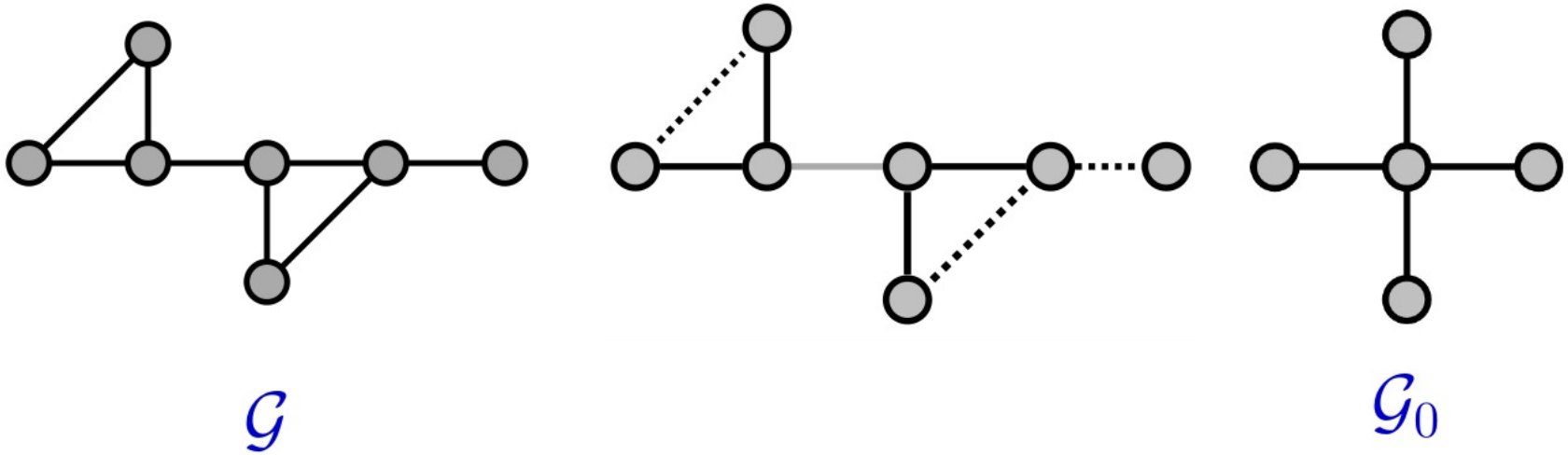
$\mathcal{G}_0$

Graph minors

Definition

A graph $\mathcal{G}_0$ is a *minor* of a graph $\mathcal{G}$ if $\mathcal{G}_0$ is obtained by:

- deleting some edges,
- deleting arising isolated vertices,
- contracting edges in $\mathcal{G}$.



$\mathcal{G}$ $\mathcal{G}_0$

Excluded minors

Definition (minor closed)

A class $\mathcal{G}$ is *minor closed* if for every $G \in \mathcal{G}$ all minors of G are in $\mathcal{G}$.

Excluded minors

Definition (minor closed)

A class $\mathcal{G}$ is *minor closed* if for every $G \in \mathcal{G}$ all minors of G are in $\mathcal{G}$.

We say that a class $\mathcal{G}$ is characterized by excluded minors in $\mathcal{H}$ if:

$$\mathcal{G} := \text{Excl}(\mathcal{H}) := \{G \mid G \text{ does not have a minor in } \mathcal{H}\}$$

Excluded minors

Definition (minor closed)

A class $\mathcal{G}$ is *minor closed* if for every $G \in \mathcal{G}$ all minors of G are in $\mathcal{G}$.

We say that a class $\mathcal{G}$ is characterized by excluded minors in $\mathcal{H}$ if:

$$\mathcal{G} := \text{Excl}(\mathcal{H}) := \{G \mid G \text{ does not have a minor in } \mathcal{H}\}$$

Theorem (Graph Minor Theorem (Robertson–Seymour, 1983–2004))

Every class of graphs that is minor closed can be characterized by finitely many excluded minors. That is, for every class $\mathcal{G}$ of minor closed graphs there are graphs $\mathcal{H}_1, \ldots, \mathcal{H}_m$ such that:

$$\mathcal{G} = \text{Excl}(\{\mathcal{H}_1, \ldots, \mathcal{H}_m\}) .$$

Deciding minor closed classes

p-MINOR

Instance: Graphs $\mathcal{G}$ and $\mathcal{H}$.
Parameter: $\|\mathcal{G}\|$
Problem: Decide whether $\mathcal{G}$ is a minor of $\mathcal{H}$.

Deciding minor closed classes

p-MINOR

Instance: Graphs $\mathcal{G}$ and $\mathcal{H}$.
Parameter: $\|\mathcal{G}\|$
Problem: Decide whether $\mathcal{G}$ is a minor of $\mathcal{H}$.

Theorem

p-MINOR $\in$ FPT, *decidable in time* $f(k) \cdot n^3$ *where* $k = \|\mathcal{G}\|$, *and* n *is the number of vertices of* $\mathcal{H}$.

Deciding minor closed classes

p-MINOR

Instance: Graphs $\mathcal{G}$ and $\mathcal{H}$.
Parameter: $\|\mathcal{G}\|$
Problem: Decide whether $\mathcal{G}$ is a minor of $\mathcal{H}$.

Theorem

p-MINOR $\in$ FPT, decidable in time $f(k) \cdot n^3$ where $k = \|\mathcal{G}\|$, and n is the number of vertices of $\mathcal{H}$.

Corollary

Every minor-closed class of graphs is decidable in cubic time.

Deciding minor closed classes

p-MINOR

Instance: Graphs $\mathcal{G}$ and $\mathcal{H}$.
Parameter: $\|\mathcal{G}\|$
Problem: Decide whether $\mathcal{G}$ is a minor of $\mathcal{H}$.

Theorem

p-MINOR $\in$ FPT, decidable in time $f(k) \cdot n^3$ where $k = \|\mathcal{G}\|$, and n is the number of vertices of $\mathcal{H}$.

Corollary

Every minor-closed class of graphs is decidable in cubic time.

Corollary

Let $\langle Q, \kappa \rangle$ be a parameterized problem on graphs such that for every $k \in \mathbb{N}$, either $\{\mathcal{G} \in Q \mid \kappa(\mathcal{G}) = k\}$ or $\{\mathcal{G} \notin Q \mid \kappa(\mathcal{G}) = k\}$ is minor closed.

Then every slice $\langle Q, \kappa \rangle_k$ is decidable in cubic time. In this case we can say that $\langle Q, \kappa \rangle$ is nonuniformly fixed-parameter tractable.

Non-uniformly fixed-parameter tractable

A parameterized problem $\langle Q, \Sigma, \kappa \rangle$ is *fixed-parameter tractable* if:

$\exists f : \mathbb{N} \to \mathbb{N}$ computable $\exists p \in \mathbb{N}[X]$ polynomial

$\exists \mathbb{A}$ algorithm, takes inputs in Σ^*

$\forall x \in \Sigma^* \big[$ $\mathbb{A}$ decides whether $x \in Q$ holds in time $\leq f(\kappa(x)) \cdot p(|x|) \big]$

Non-uniformly fixed-parameter tractable

A parameterized problem $\langle Q, \Sigma, \kappa \rangle$ is *fixed-parameter tractable* if:

$$\exists f : \mathbb{N} \to \mathbb{N} \text{ computable } \exists p \in \mathbb{N}[X] \text{ polynomial}$$
$$\exists \mathbb{A} \text{ algorithm, takes inputs in } \Sigma^*$$
$$\forall x \in \Sigma^* \big[\, \mathbb{A} \text{ decides whether } x \in Q \text{ holds in time } \leq f(\kappa(x)) \cdot p(|x|) \,\big]$$

Definition

A parameterized problem $\langle Q, \Sigma, \kappa \rangle$ is *non-uniformly fixed-parameter tractable* (in nu-FPT) if:

$$\exists f : \mathbb{N} \to \mathbb{N} \text{ computable } \exists p \in \mathbb{N}[X] \text{ polynomial}$$
$$\exists \{\mathbb{A}_k\}_{k \in \mathbb{N}} \text{ algorithms, takes inputs in } \Sigma^*$$
$$\forall x \in \Sigma^* \big[\, \mathbb{A}_{\kappa(x)} \text{ decides whether } x \in Q \text{ holds in time } \leq f(\kappa(x)) \cdot p(|x|) \,\big]$$

Using minor-closed classes for FPT results

Corollary

Let $\langle Q, \kappa \rangle$ be a parameterized problem on graphs such that for every $k \in \mathbb{N}$, either $\{\mathcal{G} \in Q \mid \kappa(\mathcal{G}) = k\}$ or $\{\mathcal{G} \notin Q \mid \kappa(\mathcal{G}) = k\}$ is minor closed. Then $\langle Q, \kappa \rangle$ is non-uniformly fixed-parameter tractable (in nu-FPT).

Applications:

- p-VERTEX-COVER $\in$ nu-FPT (p-VERTEX-COVER is minor closed).

Using minor-closed classes for FPT results

Corollary

Let $\langle Q, \kappa \rangle$ be a parameterized problem on graphs such that for every $k \in \mathbb{N}$, either $\{\mathcal{G} \in Q \mid \kappa(\mathcal{G}) = k\}$ or $\{\mathcal{G} \notin Q \mid \kappa(\mathcal{G}) = k\}$ is minor closed. Then $\langle Q, \kappa \rangle$ is non-uniformly fixed-parameter tractable (in nu-FPT).

Applications:

- p-VERTEX-COVER $\in$ nu-FPT (p-VERTEX-COVER is minor closed).
- p-FEEDBACK-VERTEX-SET $\in$ nu-FPT (problem is minor closed).

Using minor-closed classes for FPT results

Corollary

Let $\langle Q, \kappa \rangle$ be a parameterized problem on graphs such that for every $k \in \mathbb{N}$, either $\{\mathcal{G} \in Q \mid \kappa(\mathcal{G}) = k\}$ or $\{\mathcal{G} \notin Q \mid \kappa(\mathcal{G}) = k\}$ is minor closed. Then $\langle Q, \kappa \rangle$ is non-uniformly fixed-parameter tractable (in nu-FPT).

Applications:

- p-VERTEX-COVER $\in$ nu-FPT (p-VERTEX-COVER is minor closed).
- p-FEEDBACK-VERTEX-SET $\in$ nu-FPT (problem is minor closed).
- p-DISJOINT-CYCLES
  Instance: A graph $\mathcal{G}$, and $k \in \mathbb{N}$.
  Parameter: k.
  Problem: Decide whether $\mathcal{G}$ has k disjoint cycles.

  p-DISJOINT-CYCLES $\in$ nu-FPT, since the class of graphs that do not have k disjoint cycles is minor closed.

First-Order Meta-Theorem (example)

Seese's theorem

A class $\mathcal{G}$ of graphs has *bounded degree* if there is $d \in \mathbb{N}$ such that $\Delta(\mathcal{G}) \leq d$ for all $\mathcal{G} \in \mathcal{G}$ (where $\Delta(\mathcal{G})$ = max. degree of vertex in $\mathcal{G}$).

p-MC(FO, $\mathcal{G}$)

Instance: A graph $\mathcal{G} \in \mathcal{G}$, and a f-o formula φ over τ_{HG}
Parameter: $|\varphi|$.
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Seese's theorem

A class $\mathcal{G}$ of graphs has *bounded degree* if there is $d \in \mathbb{N}$ such that $\Delta(\mathcal{G}) \leq d$ for all $\mathcal{G} \in \mathcal{G}$ (where $\Delta(\mathcal{G})$ = max. degree of vertex in $\mathcal{G}$).

p-MC(FO, $\mathcal{G}$)

Instance: A graph $\mathcal{G} \in \mathcal{G}$, and a f-o formula φ over τ_{HG}
Parameter: $|\varphi|$.
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Theorem ([Seese, 1995])

p-MC(FO, $\mathcal{G}$) $\in$ FPT *for every class* $\mathcal{G}$ *of bounded degree. This model checking problem can be solved in time* $f(|\varphi|) \cdot |\mathcal{G}|$, *(linear in* $|\mathcal{G}|$*).*

Seese's theorem

A class $\mathcal{G}$ of graphs has *bounded degree* if there is $d \in \mathbb{N}$ such that $\Delta(\mathcal{G}) \leq d$ for all $\mathcal{G} \in \mathcal{G}$ (where $\Delta(\mathcal{G})$ = max. degree of vertex in $\mathcal{G}$).

p-MC(FO, $\mathcal{G}$)

Instance: A graph $\mathcal{G} \in \mathcal{G}$, and a f-o formula φ over τ_{HG}
Parameter: $|\varphi|$.
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Theorem ([Seese, 1995])

p-MC(FO, $\mathcal{G}$) $\in$ FPT *for every class* $\mathcal{G}$ *of bounded degree. This model checking problem can be solved in time* $f(|\varphi|) \cdot |\mathcal{G}|$, *(linear in* $|\mathcal{G}|$*).*

Theorem (for comparison, we saw it earlier)

EVAL(FO) *and* MC(FO) *can be solved in time* $O(|\varphi| \cdot |A|^w \cdot w)$, *where* w *is the width of the input formula* φ.

Seese's theorem

A class $\mathcal{G}$ of graphs has *bounded degree* if there is $d \in \mathbb{N}$ such that $\Delta(\mathcal{G}) \leq d$ for all $\mathcal{G} \in \mathcal{G}$ (where $\Delta(\mathcal{G})$ = max. degree of vertex in $\mathcal{G}$).

p-MC(FO, $\mathcal{G}$)
Instance: A graph $\mathcal{G} \in \mathcal{G}$, and a f-o formula φ over τ_{HG}
Parameter: $|\varphi|$.
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Theorem ([Seese, 1995])

p-MC(FO, $\mathcal{G}$) $\in$ FPT *for every class* $\mathcal{G}$ *of bounded degree. This model checking problem can be solved in time* $f(|\varphi|) \cdot |\mathcal{G}|$, *(linear in* $|\mathcal{G}|$*).*

Example

On the class of graphs of bounded degree:

- p-CLIQUE $\in$ FPT ('is there a clique of size k')?

Seese's theorem

A class $\mathcal{G}$ of graphs has *bounded degree* if there is $d \in \mathbb{N}$ such that $\Delta(\mathcal{G}) \leq d$ for all $\mathcal{G} \in \mathcal{G}$ (where $\Delta(\mathcal{G})$ = max. degree of vertex in $\mathcal{G}$).

p-MC(FO, $\mathcal{G}$)

Instance: A graph $\mathcal{G} \in \mathcal{G}$, and a f-o formula φ over τ_{HG}
Parameter: $|\varphi|$.
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Theorem ([Seese, 1995])

p-MC(FO, $\mathcal{G}$) $\in$ FPT *for every class* $\mathcal{G}$ *of bounded degree. This model checking problem can be solved in time* $f(|\varphi|) \cdot |\mathcal{G}|$, *(linear in* $|\mathcal{G}|$*).*

Example

On the class of graphs of bounded degree:

- p-CLIQUE $\in$ FPT ('is there a clique of size k')?
- 'Does $\mathcal{G}$ contain a cycle of length k?' (parameter k) is in FPT.

Seese's theorem

A class $\mathcal{G}$ of graphs has *bounded degree* if there is $d \in \mathbb{N}$ such that $\Delta(\mathcal{G}) \leq d$ for all $\mathcal{G} \in \mathcal{G}$ (where $\Delta(\mathcal{G})$ = max. degree of vertex in $\mathcal{G}$).

p-MC(FO, $\mathcal{G}$)
Instance: A graph $\mathcal{G} \in \mathcal{G}$, and a f-o formula φ over τ_{HG}
Parameter: $|\varphi|$.
Problem: Decide whether $\mathcal{A}(\mathcal{G}) \models \varphi$.

Theorem ([Seese, 1995])

p-MC(FO, $\mathcal{G}$) $\in$ FPT *for every class* $\mathcal{G}$ *of bounded degree. This model checking problem can be solved in time* $f(|\varphi|) \cdot |\mathcal{G}|$, *(linear in* $|\mathcal{G}|$*).*

Example

On the class of graphs of bounded degree:

- p-CLIQUE $\in$ FPT ('is there a clique of size k')?
- 'Does $\mathcal{G}$ contain a cycle of length k?' (parameter k) is in FPT.
- p-VERTEX-COVER $\in$ FPT ('vertex cover of size at most k?')

First-order metatheorems: reference

A good reference for other meta-theorems for first-order logic is:

[Kreutzer, 2009]: Stephan Kreutzer: *Algorithmic Meta-Theorems*.

Summary

- Logic preliminaries
  - first-order logic
    - expressing graph problems by f-o formulas
  - monadic second-order logic (MSO)
    - expressing graph problems by MSO formulas
  - complexity of evaluation and model checking problems
- Courcelle's theorem
  - FPT-results by model-checking MSO-formulas
    - for graphs with bounded tree-width
    - for structures with bounded tree-width
    - for graphs of bounded clique-width
  - applications to concrete problems
- graph minors
- meta-theorems for first-order model-checking: an example

Friday

Monday, June 10 10.30 – 12.30	Tuesday, June 11	Wednesday, June 12 10.30 – 12.30	Thursday, June 13	Friday, June 14
Introduction & basic FPT results		**Algorithmic Meta-Theorems**		
motivation for FPT kernelization, Crown Lemma, Sunflower Lemma	GDA	1st-order logic, monadic 2nd-order logic, FPT-results by Courcelle's Theorems for tree and clique-width	GDA	GDA
Algorithmic Techniques		*Formal-Method & Algorithmic Techniques*		
	14.30 – 16.30			14.30 – 16.30
	Notions of bounded graph width			**FPT-Intractability Classes & Hierarchies**
	path-, tree-, clique width, FPT-results by dynamic programming, transferring FPT results betw. widths	GDA	GDA	motivation for FP-intractability results, FPT-reductions, class XP (slicewise polynomial), W- and A-Hierarchies, placing problems on these hierarchies

Example suggestions

Examples

1. Find a first-order logic formula over τ_{G} that expresses that a graph has a cycle of length precisely k.
2. Find an MSO_1 or $\mathrm{MSO}(\tau_{\mathsf{G}})$ formula that expresses that a graph has a dominating set of $\leq k$ elements.
3. Find an MSO_2 or $\mathrm{MSO}(\tau_{\mathsf{HG}})$ formula $\mathit{feedback}(S)$ that expresses that $S \subseteq V$ is a feedback vertex set.
4. $(^\star)$ Find an MSO_1 or $\mathrm{MSO}(\tau_{\mathsf{G}})$ formula that expresses that a graph is connected.
5. $(^\star)$ Find an MSO_2 or $\mathrm{MSO}(\tau_{\mathsf{HG}})$ formula $\mathit{path}(x, y, Z)$ that expresses that Z is a set of edges that forms a path from x to y.

References I


Courcelle, B. (1990).
The monadic second-order logic of graphs. i. recognizable sets of finite graphs.
Information and Computation, 85(1):12 – 75.


Courcelle, B., Makowsky, J. A., and Rotics, U. (2000).
Linear time solvable optimization problems on graphs of bounded clique-width.
Theory of Computing Systems, 33(2):125–150.


Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., and Saurabh, S. (2015).
Parameterized Algorithms.
Springer, 1st edition.


Flum, J. and Grohe, M. (2006).
Parameterized Complexity Theory.
Springer.

References II


Kreutzer, S. (2009).
Algorithmic Meta-Theorems.
Technical report, arXiv.org.
https://arxiv.org/abs/0902.3616.


Sásak, R. (2010).
Comparing 17 graph parameters.
Master's thesis, University of Bergen, Norway.


Seese, D. (1995).
Linear time computable problems and logical descriptions.
Electronic Notes in Theoretical Computer Science, 2:246 – 259.
SEGRAGRA 1995, Joint COMPUGRAPH/SEMAGRAPH Workshop on Graph Rewriting and Computation.