Logic, Graph Theory, and Complexity

Martin Grohe

University of Edinburgh

Logic, Graph Theory, and Complexity

I. Algorithmic Meta Theorems through Logic and Graph Theory

II. Computing Crossing Numbers

III. Evaluating Tree-Structured Formulas

Logic, Graph Theory, and Complexity I

Algorithmic Meta Theorems through Logic and Graph Theory

Descriptive Complexity Theory

Idea

- Model computational problems by **model checking problems** of the following form:

Input:	Structure $\mathcal{A}$ in some class D of structures
	formula φ in some class Φ of formulas
Problem:	Evaluate φ in $\mathcal{A}$

- Try to determine the borderline between tractability and intractability by varying the two parameters of this problem:
  - the class D of (finite relational) structures
  - the class Φ of formulas (of some logic L).

Motivation

Many important problems in computer science are essentially model-checking problems:

- Database query evaluation
- Model-Checking in automated verification
- Constraint satisfaction in artificial intelligence

Many others problems can naturally be described as model-checking problems.

Motivation (cont'd)

- Description of algorithmic problems as model-checking problems reflects natural distinction between **data** and **problem description**.
- It allows us to separate two sources of complexity:
  - **Logical parameters** of problem description,
    e.g. quantifier alternation, recursion.
  - **Structural parameters** of data,
    e.g. degree, tree-width, planarity.

Traditional Descriptive Complexity

Relates **(logical) complexity of problem description** to **computational complexity**.

Theorem (Fagin 1974)

A problem is in NP if, and only if, it is definable in existential second-order logic.

Theorem (Immerman 1982, Vardi 1982)

A problem is in PTIME if, and only if, it is definable in least fixed-point logic.

(Here problems are described as classes of (ordered) relational structures.)

Algorithmic Meta-Theorems

Use logic to uniformly describe class of algorithms.

Theorem (Papadimitriou and Yannakakis 1991)
Optimisation problems that are definable in the fragment MAX-SNP of existential second-order logic have a polynomial time fixed-ratio approximation algorithm.

Theorem (Courcelle 1990)
Problems on graphs of bounded tree-width that are definable in monadic second-order logic are decidable in linear time.

Finite Relational Structures

Structure of **vocabulary** $\{R_1, \dots, R_n\}$:

$$\mathcal{A} = (\underbrace{A}_{\text{universe}}, \underbrace{R_1^{\mathcal{A}}, \dots, R_n^{\mathcal{A}}}_{\text{relations on } A})$$

Each R_i **relation symbol** of prescribed arity.

Examples

- **Graph** $\mathcal{G} = (G, E^{\mathcal{G}})$ (binary edge relation)
- **Hypergraph** $\mathcal{H} = (H, V^{\mathcal{H}}, E^{\mathcal{H}}, I^{\mathcal{H}})$ (unary relations representing vertices and hyperedges and binary incidence relation)

  Note: Also gives alternative represenation for graphs.
- **Boolean Circuit** $\mathcal{C} = (C, E^{\mathcal{C}}, \text{AND}^{\mathcal{C}}, \text{OR}^{\mathcal{C}}, \text{NEG}^{\mathcal{C}}, \text{IN}^{\mathcal{C}}, \text{OUT}^{\mathcal{C}})$ (binary edge relation, unary relations for and-gates, or-gates, negation-gates, input-gates, output-gate, resp.)

First-Order Logic

Infinite supply of **individual variables** ranging over elements of universe of a structure.

Syntax: Formulas of **first-order logic (FO)** are formed by the following rules:

Atomic formulas: $x = y$ and $Rx_1 \dots x_k$, where R is a k-ary relation symbol and $x, y, x_1, \dots, x_k$ individual variables, are FO-formulas.

Conjunction: If φ and ψ are FO-formulas, then $\varphi \wedge \psi$ is an FO-formula.

Negation: If φ is an FO-formula, then $\neg\varphi$ is an FO-formula.

Existential Quantification: If φ is an FO-formula and x an individual variable, then $\exists x \varphi$ is an FO-formula.

Similarly for **disjunction** ($\bigvee$), **implication** ($\longrightarrow$), **universal quantification** ($\forall$).

Semantics: Defined in the natural way.

Examples

(1) Graph has **clique** of size k:

$$\exists x_1 \ldots \exists x_k \bigwedge_{1 \le i < j \le k} E x_i x_j .$$

(2) Graph has **dominating set** of size k:

$$\exists x_1 \ldots \exists x_k \forall y \bigvee_{1 \le i \le k} \big(E x_i y \vee x_i = y \big).$$

(3) Hypergraph has **vertex cover** of size k:

$$\exists x_1 \ldots \exists x_k \Big(\bigwedge_{i=1}^{k} V x_i \wedge \forall y \big(E y \to \bigvee_{1 \le i \le k} I x_i y \big) \Big).$$

Monadic Second-Order Logic

Individual variables and **set variables** (ranging over subsets of the universe of a structure)

Syntax: Formulas of **monadic second-order logic (MSO)** are formed by the rules for FO and the following rules:

Additional atomic formulas: For all set variables X and individual variables y, Xy is an MSO-formula.

Existential Set Quantification: If φ is an MSO-formula and X a set variable, then $\exists X\varphi$ is an MSO-formula.

Similarly for **universal set quantification**.

Semantics: Defined in the natural way.

Examples

(1) Graph is **k-colourable**:

$$\exists X_1 \ldots \exists X_k \;\Big(\forall x \bigvee_{i=1}^{k} X_i x \\ \wedge \forall x \forall y \big(Exy \to \bigwedge_{i=1}^{k} \neg (X_i x \wedge X_i y) \big) \Big).$$

(2) Circuit is **satisfiable**:

$$\exists X \;\Big(\forall x \big(\mathsf{OUT}x \to Xx \big) \\ \wedge \forall x \big(\mathsf{AND}x \to \big(Xx \leftrightarrow \forall y (Exy \to Xy) \big) \big) \\ \wedge \forall x \big(\mathsf{OR}x \to \ldots \big) \\ \wedge \forall x \big(\mathsf{NEG}x \to \ldots \big) \Big).$$

The Model-Checking Problem

Φ class of formulas (logic)

D class of finite relational structures

Model-checking for Φ on D is the following problem:

Input:	Structure $\mathcal{A} \in \mathrm{D}$, sentence $\varphi \in \Phi$
Problem:	Decide if $\mathcal{A}$ satisfies φ

Remarks and Conventions

- A **sentence** is a formula without free variables.
- If D is the class of all finite relational structures, we don't mention it.
- Instead of *model-checking for* $\{\varphi\}$ we say *model-checking for* φ.
- n = size of the input structure $\mathcal{A}$
- ℓ = size of the input formula φ

Complexity

Theorem (Stockmeyer 1974, Vardi 1982)
The model-checking problems for FO and MSO are PSPACE-complete.

Proof: *Containment in PSPACE:* Straightforward algorithm for MSO-model-checking requires space $O(\ell \cdot n)$.

Hardness: The **quantified Boolean formula** problem is essentially FO-model-checking on a fixed structure with two elements (representing ‘true’ and ‘false’).

Data Complexity

Proposition

The model-checking problem for FO is solvable in time

$$O(n^{\ell}).$$

Corollary

For every fixed FO-formula, the model-checking problem is in PTIME.

Remark

Unless PTIME=NP, the analogous result for MSO does not hold.

(E.g., take the MSO-formula saying that a circuit is satisfiable.)

Parameterized Complexity

Remark

Model-Checking for the temporal logic **LTL** is also PSPACE-complete, but solvable in time

$$O(2^{\ell} \cdot n).$$

Definition

Model-Checking for Φ on D is **fixed-parameter tractable (fpt)** with **parameter dependence** f and **exponent** c, for some computable function f and constant c, if there is an algorithm solving the problem in time

$$f(\ell) \cdot n^{c}.$$

The problem is **fixed-parameter linear time (fpl)** if it is fpt with exponent 1.

Parameterized Complexity (cont'd)

Observation

Unless PTIME=NP, model-checking for MSO is not fpt.

Proof: If model-checking for MSO was fpt, then model-checking for every fixed MSO-sentence would be in PTIME. We have already seen that this is not the case unless PTIME=NP.

Theorem (Downey, Fellows, Taylor 1996)

Unless FPT=AW$[*]$, model-checking for FO is not fpt.

Remark

FPT and AW$[*]$ are *parameterized complexity classes* which are conjectured to be distinct.

Questions

(1) *For which classes D of structures is the model-checking problem for FO or MSO on D fpt?*
(This talk.)

(2) *For which classes of formulas is the model-checking problem tractable?*
(Third talk.)

Tree Automata

- Σ finite alphabet
- **Σ-tree:** Binary tree whose vertices are labelled with letters in Σ
- **Σ-tree automaton:** Runs bottom-up on Σ-trees
  - State at a leaf depends on its label.
  - State at an inner node depends on its label and the states at its children.
  - Automaton accepts if state at root is an accepting state.

Proposition

There is a linear time algorithm deciding whether a given (deterministic) tree automaton accepts a given tree.

MSO on Trees

Theorem (Doner 1970, Thatcher and Wright 1968)

For every MSO-sentence φ *there is a tree automaton* $\mathfrak{A}_\varphi$ *accepting precisely the trees satisfying* φ *(and vice versa).*

Furthermore, $\mathfrak{A}_\varphi$ *can be effectively constructed from* φ.

Corollary

Model-Checking for MSO on the class of labelled binary trees is fpl.

The Parameter-Dependence

Theorem (Stockmeyer and Meyer 1973)
The smallest equivalent automaton for a given MSO-formula can have size non-elementary in terms of the size of the formula.

So the automata theoretic approach only yields an fpt-algorithm with non-elementary parameter dependence.

Theorem (Frick and G. 2002)
Unless PTIME=NP, model-checking for MSO on the class of binary trees is not fpt with elementary parameter dependence.

Tree-Decompositions

- A **tree-decomposition** of a structure $\mathcal{A}$ is a pair $(\mathcal{T}, \beta)$, where $\mathcal{T}$ is a tree and $\beta : T \to 2^A$ such that
  (1) For every relation symbol R in the vocabulary of $\mathcal{A}$ and every tuple $\bar{a} \in R^{\mathcal{A}}$ there is a $t \in T$ such that $\bar{a} \in \beta(t)$.
  (2) For every $a \in A$ the set $\{t \in T \mid a \in \beta(t)\}$ is non-empty and connected.
- The graphs induced by $\mathcal{A}$ on the $\beta(t)$, for $t \in T$, are called the **blocks** of the decomposition.
- The **width** of (T, β) is max$\{\beta(t) \mid t \in T\} + 1$.
- The **tree-width** of $\mathcal{A}$, denoted by tw$(\mathcal{A})$, is the maximum taken over the widths of all tree-decompositions of $\mathcal{A}$.
- A class C of structures has **bounded tree-width** if there is a w such that tw$(\mathcal{A}) \leq w$ for all $\mathcal{A} \in C$.

Courcelle's Theorem

Theorem (Courcelle 1990)

Let $\mathbb{D}$ *be a class of structures of bounded tree-width. Then model-checking for MSO on* $\mathbb{D}$ *is fpl.*

Proof of Courcelle's Theorem

Idea: *Interpret structures of bounded tree-width over labelled trees.*

Choose w such that $\text{tw}(\mathcal{A}) \leq w$ for all $\mathcal{A} \in \mathrm{D}$.

(1) Compute tree-decomposition $(\mathcal{T}, \beta)$ of width w of input structure $\mathcal{A}$ (possible in linear time, Bodlaender 1996).

(2) W.l.o.g. $\mathcal{T}$ binary tree. Find labelling of $\mathcal{T}$ such that label of $t \in T$ encodes:

- substructure induced by $\mathcal{A}$ on $\beta(t)$,
- if T has parent s, relation between $\beta(t)$ and $\beta(s)$.

(3) Translate input sentence φ into MSO-sentence φ' over labelled tree.

(4) Use the fact that MSO over labelled trees is fpl.

Beyond Bounded Tree-Width?

Proposition

$\mathbb{D}$ *class of structures closed under taking minors. Then* $\mathbb{D}$ *has bounded tree-width iff model-checking for MSO on* $\mathbb{D}$ *is fpt.*

Proof: "$\Longrightarrow$": Courcelle's Theorem.

"$\Longleftarrow$": Use the following facts:

- A minor closed class of graphs of unbounded tree-width contains every planar graph (Robertson and Seymour 1986).
- 3-Colorability of planar graphs is NP-complete (Garey, Johnson, and Stockmeyer 1976).
- There is an MSO-sentence saying that a graph is 3-colorable.

Beyond Bounded Tree-Width!

Example

The class of all complete graphs has unbounded tree-width, but model-checking for MSO on this class is fpt.

The Gaifman Graph of a Structure

Let $\mathcal{A}$ be a structure.

- The **Gaifman graph** of $\mathcal{A}$ is the graph $\mathcal{G}(\mathcal{A})$ with vertex set A and an edge between $a, b \in A$ if there is an relation symbol R in the vocabulary of $\mathcal{A}$ and a tuple $\bar{a} \in R^{\mathcal{A}}$ that contains both a and b.

- The **distance** $d(a, b)$ between $a, b \in A$ is the distance between a and b in $\mathcal{G}(\mathcal{A})$.

- For $a \in A$ and $r \geq 0$, $B_r(a)$ denotes the r-Ball around a and $\mathcal{B}_r(a)$ the substructure induced by $\mathcal{A}$ on $B_r(a)$.

Local First-Order Formulas

An FO-formula $\varphi(x)$ is r**-local around** x if all its quantifiers are relativized to $B_r(x)$, i.e., of the form

$$\exists y\big(d(x,y) \leq r \wedge \ldots\big) \quad \text{and} \quad \forall y\big(d(x,y) \leq r \rightarrow \ldots\big).$$

Here $d(x,y) \leq r$ is an FO-formula saying that the distance between x and y is at most r.

Observation

If $\varphi(x)$ is r-local around x, then for every structure $\mathcal{A}$ and every $a \in A$,

$$\mathcal{A} \text{ satisfies } \varphi(a) \iff \mathcal{B}_r(a) \text{ satisfies } \varphi(a)$$

Gaifman's Locality Theorem

Theorem (Gaifman 1981)

Every first-order sentence is equivalent to a Boolean combination of sentences of the form

$$\exists x_1 \ldots \exists x_k \Big(\bigwedge_{1 \le i < j \le k} d(x_i, x_j) > 2r \wedge \varphi(x_i) \Big),$$

where $\varphi(x)$ is r-local around x.

Local Tree-Width

Definition

(1) The **local tree-width** of a structure $\mathcal{A}$ is the function $\text{ltw}(\mathcal{A}) : \mathbb{N} \to \mathbb{N}$ defined by

$$\text{ltw}(\mathcal{A})(r) = \max \left\{ \text{tw}(\mathcal{B}_r(a)) \;\middle|\; a \in A \right\}.$$

(2) A class C of structures has **bounded local tree-width** if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for all $\mathcal{A} \in C$ and $r \geq 1$,

$$\text{ltw}(\mathcal{A})(r) \leq f(r).$$

Examples

- *Graphs of bounded tree-width.* For all $k \geq 1$, the class of all graphs of tree-width at most k has bounded local tree-width.

- *Graphs of bounded degree.* For all $k \geq 1$, the class of all graphs of degree at most k has bounded local tree-width.

- *Planar graphs and graphs of bounded genus.* For every surface S, the class of all graphs embeddable in S has bounded local tree-width.

FO-Model-Checking on Classes of Bounded Local Tree-Width

Theorem (Frick and G. 1999)

Let $\mathbb{D}$ be a class of bounded local tree-width. Then model-checking for FO on $\mathbb{D}$ is fpt.

Remarks

- The exponent of the fpt-algorithm can be chosen to be $1 + \epsilon$ for every $\epsilon > 0$.
- For **locally tree-decomposable** classes of structures (including planar graphs, graphs of bounded genus, and graphs of bounded degree), we obtain an fpl-algorithm.

FO-Model-Checking on Classes of Bounded Local Tree-Width

Theorem (Flum and G. 2001)

Let D be a class of graphs with excluded minors. Then model-checking for FO on D is fpt.

Proof

Based on the previous results and the following decomposition theorem (G. 1999):

> *Let* $\mathbb{D}$ *be a class of graphs with an excluded minor. Then every graph in* $\mathbb{D}$ *has a tree-decomposition whose blocks have bounded local tree-width after removing a constant number of vertices.*

(Follows relatively easily from Robertson and Seymour's structure theory for graphs with excluded minors.)