The Complexity of First-Order and Monadic Second-Order Logic Revisited

Martin Grohe
University of Edinburgh

(Joint work with Markus Frick)

The model-checking problem

Model-Checking for a logic L on a class $\mathbb{C}$ of structures:

Input:	Structure $\mathcal{A} \in \mathbb{C}$, sentence $\varphi \in \mathrm{L}$
Problem:	Decide if $\mathcal{A}$ satisfies φ

Model-Checking problems naturally occur in various areas of computer science, e.g., database theory, automated verification, AI.

Complexity

Theorem (Stockmeyer 1974, Vardi 1982)

Model-checking for first-order logic **FO** *and monadic second-order logic* **MSO** *is PSPACE complete.*

This holds on any class of structures that contains at least one structure with at least two elements.

A closer look

Notation

n = size of the input structure (of a model-checking problem)

k = size of the input sentence

Proposition (Folklore)

Model-checking for FO*:*

$$\mathsf{TIME}\Big(O(n^k)\Big).$$

Model-checking for MSO*:*

$$\mathsf{TIME}\Big(O(2^{n \cdot k})\Big).$$

Parameterized Complexity

Compare:

- Model-Checking for **FO**: PSPACE-complete, TIME$\left(O(n^k)\right)$.
- Model-Checking for **LTL**: PSPACE-complete, TIME$\left(2^{O(k)} \cdot n\right)$.

Definition

A model-checking problem is **fixed-parameter tractable (fpt)**, if there is a computable function f, a polynomial p, and an algorithm solving the problem in time

$$f(k) \cdot p(n).$$

The Parameterized Complexity Model-Checking for FO and MSO

Observation

Unless P $=$ NP*, model-checking for MSO on the class of all graphs is not fpt.*

Proof: There is an MSO-formula saying that a graph is 3-colourable. $\square$

Theorem (Downey, Fellows, Taylor 1996)

FO-*Model-Checking on the class of all graphs is complete for the parameterized complexity class* AW$[*]$*.*

Thus unless AW$[*]$ $=$ FPT*, model-checking for* FO *on the class of all graphs is not fpt.*

Tractable Restrictions of MSO-Model-Checking

Theorem (Büchi 1960 / Thatcher, Wright 1968 / Courcelle 1990)

Model-checking for MSO is solvable in time

$$f(k) \cdot n,$$

for some computable function $f : \mathbb{N} \to \mathbb{N}$*, on the following classes of structures:*

- *words*
- *trees*
- *graphs of bounded tree-width.*

Proofs are based on translation of MSO-formulas to finite automata.

Tractable Restrictions of FO-Model-Checking

Theorem (Seese 1996 / Frick, G. 1999 / Flum, G. 2001)

Model-checking for FO is solvable in time

$$f(k) \cdot n,$$

for some computable function $f : \mathbb{N} \to \mathbb{N}$*, on the following classes of structures:*

- *graphs of bounded degree*
- *graphs of bounded local tree width (includes planar graphs and graphs of bounded genus)*

Furthermore, model-checking for FO is fpt on all classes of graphs with excluded minors.

Proofs are based on the tractability results for MSO and on the *locality* of FO.

Dependence on k

In all the fixed-parameter tractability results, the dependence on the formula size k is **non-elementary**.

More precisely, we have

$$f(k) = \left.2^{2^{\cdot^{\cdot^{\cdot^{2^k}}}}}\right\} \text{height } \Theta(k).$$

Are there better fpt algorithms ?

MSO on words

Theorem 1

Unless $\mathsf{P} = \mathsf{NP}$*, there is no model-checking algorithm for MSO on the class of words with time complexity bounded by*

$$\mathrm{f}(\mathrm{k}) \cdot \mathrm{p}(\mathrm{n})$$

for an ***elementary*** f *and a* ***polynomial*** p*.*

FO on words

Theorem 2

Unless FPT $=$ W$[1]$, *there is no model-checking algorithm for FO on the class of words with time complexity bounded by*

$$\mathrm{f}(\mathrm{k}) \cdot \mathrm{p}(\mathrm{n})$$

for an ***elementary*** f *and a* ***polynomial*** p.

FO on Structures of Bounded Degree

Theorem 3

(1) There is a model-checking algorithm for FO on the class of structures of degree 2 with time complexity

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

Unless FPT $=$ W$[1]$, *there is no algorithm solving the same problem in time*

$$2^{2^{o(k)}} \cdot \mathrm{poly}(n).$$

(2) For every $d \geq 3$, *there is a model-checking algorithm for FO on the class of structures of degree d with time complexity*

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

Unless FPT $=$ W$[1]$, *there is no algorithm solving the same problem in time*

$$2^{2^{2^{o(k)}}} \cdot \mathrm{poly}(n).$$

Proof of Theorem 1

Suppose for contradiction that there is a model-checking algorithm $\boxed{A}$ for MSO on words with a time complexity

$$2^{2^{\cdot^{\cdot^{\cdot^{2^k}}}}\Big\}\text{height } h} \cdot p(n)$$

for a fixed $\boxed{h}$ and a polynomial $\boxed{p}$.

We shall use A to prove that 3SAT in PTIME.

Proof of Theorem 1 (cont'd)

For every 3CNF-formula θ, we shall construct (in PTIME)

- an MSO-formula φ of length $O(\log^{(h+1)}(|\theta|))$
- a word $\mathcal{W}(\theta)$ of length $q(|\theta|)$ (for some polynomial q)

such that

$$\theta \text{ satisfiable} \iff \mathcal{W}(\theta) \text{ satisfies } \varphi.$$

Using algorithm A, we can check if $W(\theta)$ satisfies φ in time

$$\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2^{O(\log^{(h+1)}(|\theta|))}}}}}}}_{\text{height } h} \cdot p(q(|\theta|)).$$

Proof of Theorem 1 (cont'd) — Encoding Numbers

For all $h \in \mathbb{N}, k \in \mathbb{R}$ let $T(h,k) = 2^{2^{\cdot^{\cdot^{\cdot^{2^k}}}}}\Big\}$height h.

Lemma 1

Let $h \geq 1$. There is an encoding μ_h of natural numbers by words and formulas $\chi_{h,\ell}(x,y)$, for $\ell \geq 1$, such that:

(1) μ_h is computable in polynomial time.

(2) $|\chi_{h,\ell}| \in O(h+\ell)$, and $\chi_{h,\ell}$ is computable from h and ℓ in polynomial time.

(3) For all
- words $\mathcal{W}$,
- $\ell, m, n \in \mathbb{N}$ such that $m, n \leq T(h,\ell)$,
- subwords $\mathcal{W}_x = \mu_h(m)$ and $\mathcal{W}_y = \mu_h(n)$ of $\mathcal{W}$ starting at positions x, y, respectively:

$$\mathcal{W} \models \chi_{h,\ell}(x,y) \iff m = n.$$

Proof of Theorem 1 (cont'd) — Proof of Lemma 1

- μ_1 is essentially a binary encoding
- Let $\mathrm{bit}(n, i)$ be the ith bit in the binary encoding of n.
  Then

  $$\mu_h(n) \approx \$\ \mu_{h-1}(0)\ \#\ \mathrm{bit}(n,0)\ \$\ \mu_{h-1}(1)\ \#\ \mathrm{bit}(n,1)\ \$ \quad \ldots \quad \$\ \mu_{h-1}(|n|)\ \#\ \mathrm{bit}(n,|n|)\$,$$

  where $|n|$ denotes the length of the binary representation of n.
  For example,

  $$\mu_2(21) = \$0\#1\$1\#0\$10\#1\$11\#0\$100\#1\$.$$

  The binary expansion of 21 is 10101, and we count bits from the right to the left.

Proof of Theorem 1 (cont'd) — Encoding Formulas

Example

$$\theta = (X_0 \vee X_1 \vee \neg X_2) \wedge (X_0 \vee \neg X_2 \vee X_3)$$

The μ_h-encoding of θ would be

$$\% \, \mu_h(0) + \mu_h(1) + \mu_h(2) - \% \, \mu_h(0) + \mu_h(2) - \mu_h(3) + \%$$

followed by

$$\mu_h(0) \star \mu_h(1) \star \mu_h(2) \star \mu_h(3) \star$$

as a placeholder for truth-value assignments.

Proof of Theorem 1 (cont'd) — Main Lemma

Lemma 2

Let $h \geq 1$. There is an encoding μ_h of CNF-formulas by words and formulas $\varphi_{h,\ell}(x, y)$, for $\ell \geq 1$, such that:

(1) μ_h is computable in polynomial time.

(2) $|\varphi_{h,\ell}| \in O(h + \ell)$, and $\varphi_{h,\ell}$ is computable from h and ℓ in polynomial time.

(3) For $\ell \geq 1$ and all propositional formulas θ with at most $T(h, \ell)$ variables,

$$\mu_h(\theta) \models \chi_{h,\ell} \iff \theta \text{ is satisfiable.}$$