Logic, Graph Theory, and Complexity III

Evaluating Tree-Structured Formulas

Martin Grohe
University of Edinburgh

The Model-Checking Problem

Φ class of formulas

$\mathbb{D}$ class of finite relational structures

Model-checking for Φ on $\mathbb{D}$:

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

Tractable Model Checking Problems

In general, model-checking problems tend to be computationally hard.

Talk I

(Fixed-parameter) tractable cases of the model-checking problems for MSO and FO obtained by restricting the classes of input structures.

This Talk

Tractable cases of the model-checking problem obtained by restricting the classes of input formulas.

Positive Primitive Formulas

Definition

A **positive primitive formula (pp-formula)** is an FO-formula of the form

$$\exists x_1 \ldots \exists x_k (\alpha_1 \wedge \ldots \wedge \alpha_m),$$

where $\alpha_1, \ldots, \alpha_m$ are atomic formulas.

Example

Graph has clique of size k:

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

PP-Formulas and Database Queries

- **Relational databases** are nothing but finite relational structures.
- **Conjunctive queries** are the queries defined by pp-formulas.

Example

Database with a table `Direct` whose two columns are named `to` and `from`.

SQL-Query

```
SELECT D1.from, D2.to
FROM Direct AS D1, Direct AS D2
WHERE D1.to = D2.from
```

Defined by **pp-formula**

$$\exists z\big(\mathtt{Direct}(x,z) \wedge \mathtt{Direct}(z,y)\big).$$

Homomorphisms of Relational Structures

Let $\mathcal{A}$ and $\mathcal{B}$ structures of the same vocabulary τ.

A **homomorphism** from $\mathcal{A}$ to $\mathcal{B}$ is a mapping $h : A \to B$ such that for all $R \in \tau$ and for all tuples $\bar{a}$ of elements of A,

$$\bar{a} \in R^{\mathcal{A}} \implies h(\bar{a}) \in R^{\mathcal{B}}.$$

The **homomorphism problem** for relational structures is the following problem:

Input:	Structures $\mathcal{A}$, $\mathcal{B}$.
Problem:	Decide if there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$.

The homomorphism problem **restricted** to a class C of structures is the restriction of the problem where structure $\mathcal{A}$ is from C (and $\mathcal{B}$ is arbitrary).

Model Checking for pp-Formulas vs Homomorphism

Lemma

(1) *For every structure $\mathcal{A}$ there is a pp-sentence $\varphi_{\mathcal{A}}$ such that for all structures $\mathcal{B}$:*

There is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$ $\iff$ *$\mathcal{B}$ satisfies $\varphi_{\mathcal{A}}$.*

(2) *For every pp-sentence φ there is a structure $\mathcal{A}_{\varphi}$ such that for all structures $\mathcal{B}$:*

There is a homomorphism from $\mathcal{A}_{\varphi}$ to $\mathcal{B}$ $\iff$ *$\mathcal{B}$ satisfies φ.*

Corollary

The homomorphism problem and the model-checking problem for pp-formulas are equivalent.

Complexity

Theorem (Karp 1974)

The homomorphism problem and thus the model checking problem for pp-formulas is NP-complete.

Remark

Actually, the homomorphism problem restricted to the class of complete graphs (the **clique problem**) already is NP-hard.

Parameterized Complexity

Theorem (Downey and Fellows 1995)
The homomorphism problem is complete for the parameterized complexity class $\mathrm{W}[1]$.

Remark
Again, for the hardness it suffices to consider the (parameterized) clique problem.

Model Checking for MSO-formulas vs Homomorphism

Lemma

For every structure $\mathcal{B}$ there is an MSO-sentence $\psi^{\mathcal{B}}$ such that for all structures $\mathcal{A}$:

There is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$ $\iff$ *$\mathcal{A}$ satisfies $\psi^{\mathcal{B}}$.*

Example

$\mathcal{B}$ directed path of length n. Then

$$\psi^{\mathcal{B}} = \exists X_1 \dots X_{n+1} \forall x \forall y \big(Exy \to \bigvee_{1 \le i \le n} (X_i x \wedge X_{i+1} y) \big).$$

Remarks

- The converse of the previous lemma does not hold. The MSO-formulas $\psi^{\mathcal{B}}$ are of a very restricted syntactic form (characterised by Feder and Vardi 1997).
- We have established a relation between model checking for restricted classes of pp-formulas and model checking for (a fragment of) MSO on restricted classes of structures.
- **Constraint satisfaction problems**, playing a prominent role in artificial intelligence, are yet another incarnation of the homomorphism problem (Feder and Vardi 1997).

Labelled Tree Homomorphism

Proposition

The homomorphism problem restricted to the class of labelled binary trees is solvable in time

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

where k *is the size of the input tree and* n *the size of the structure it has to be mapped to.*

Proof

Let $\mathcal{T}$ be a labelled binary tree and $\mathcal{G}$ a labelled graph.

We define a non-deterministic tree automaton that accepts $\mathcal{T}$ iff there is a homomorphism from $\mathcal{T}$ to $\mathcal{G}$:

- States are the vertices of $\mathcal{G}$.
- There is a transition $(v_1, v_2) \xrightarrow{a} w$ if w has label a and v_1 and v_2 are neighbours of w.
- State v can be assumed at a leaf if the leaf has the same label as v.
- All states are accepting.

It can be checked in time $O(k \cdot n)$ if a non-deterministic tree automaton with n states accepts a tree of size k — just simulate all runs of the automaton in parallel.

Remarks

- There is also a simple dynamic programming algorithm solving the homomorphism problem restricted to the class of labelled binary trees in time $O(k \cdot n)$.
- Asking whether a tree automaton A accepts a tree $\mathcal{T}$ is essentially is the same as asking whether there is a homomorphism from $\mathcal{T}$ to A.

Structures of bounded tree-width

Theorem (Folklore)
The homomorphism problem restricted to the class of structures of tree-width at most w is solvable in time

$$O(k \cdot n^{w+1}).$$

A Partial Converse

For a class C of graphs, let $S(C)$ be the class of all structures whose Gaifman graph is in C.

Theorem (G., Schwentick, and Segoufin 2001)
Suppose that FPT $\neq$ W[1]. *Then for every recursively enumerable class C of graphs, the following statements are equivalent:*

(1) The homomorphism problem restricted to $S(C)$ is in polynomial time.

(2) The homomorphism problem restricted to $S(C)$ is fpt (i.e., solvable in time $f(k) \cdot n^c$ for some computable function f and constant c.)

(3) C (or equivalently $S(C)$) has bounded tree-width.

If (1)–(3) do not hold, then the homomorphism problem restricted to $S(C)$ is W[1]-complete.

FPT vs W[1]

If FPT=W[1], then

- The (unrestricted) homomorphism problem is fpt.
- There is an fpt-algorithm deciding whether a given non-deterministic Turing machine of size n has a k-step halting computation (Cai, Chen, Downey, and Fellows 1997).
- The satisfiability problem for propositional formulas in 3CNF with n variables is solvable in time $2^{o(n)}$ (Abrahamson, Downey, and Fellows 1995).

Proof of the Theorem

Lemma 1

The homomorphism problem restricted to labelled grids is hard for W[1] *under parameterized reductions.*

Proof of the Theorem (cont'd)

Recall that a minor of a graph is obtained by **contracting** and **deleting** edges (and deleting vertices).

For a class C of graphs, let $\mathrm{LM}(\mathrm{C})$ the class of all labelled graphs whose underlying graphs are minors of graphs in C.

Lemma 2

Let C be a recursively enumerable class of graph. Then the homomorphism problem restricted to $\mathrm{LM}(\mathrm{C})$ is reducible to the homomorphism problem restricted to $\mathrm{S}(\mathrm{C})$.

Proof of Lemma 2

- Let $\mathcal{H} \in \mathrm{LM}(\mathrm{C})$, and let $\mathcal{H}' \in \mathrm{C}$ such that $\mathcal{H}$ is a minor of $\mathcal{H}'$.
- Color the edges of $\mathcal{H}'$ has follows:
  - Edges that are deleted when creating $\mathcal{H}$ are colored white.
  - Edges that are contracted when creating $\mathcal{H}$ are colored red.
  - Edges of $\mathcal{H}$ are colored black.

  Moreover, label the vertices as in $\mathcal{H}$.

  Let $\mathcal{H}^*$ be the resulting edge-colored and vertex-labelled graph.
- For a vertex labelled graph $\mathcal{G}$, let $\mathcal{G}^*$ the edge colored and vertex labelled graph obtained from $\mathcal{G}$ by
  - coloring all edges black,
  - attaching a red loop to every vertex,
  - adding a white edge between any two vertices.
- Then there is a homomorphism from $\mathcal{H}$ to $\mathcal{G}$ iff there is a homomorphism from $\mathcal{H}^*$ to $\mathcal{G}^*$.
- $(\mathcal{H}, \mathcal{G}) \mapsto (\mathcal{H}^*, \mathcal{G}^*)$ is the desired reduction.

The excluded grid theorem

Theorem (Robertson and Seymour 1986)

For every $r \geq 1$ there exists a w such that every graph of tree-width $> w$ has a grid of radius r as a minor.

Remark

For grids (and more generally graphs of degree 3) being a minor or homeomorphic to a subgraph are the same.

Proof of the Theorem (cont'd)

(1) $\Rightarrow$ (2): Trivial.

(2) $\Rightarrow$ (3): Suppose the homomorphism problem restricted to $S(C)$ is fpt and C has unbounded tree-width.

By the Excluded Grid Theorem, every labelled grid is contained in $LM(C)$.

Then by Lemma 2, the homomorphism problem restricted to labelled grids is reducible to the homomorphism problem restricted to $S(C)$.

Thus by Lemma 1, the homomorphism problem restricted to $S(C)$ is hard for W[1]. Thus FPT=W[1].

(3) $\Rightarrow$ (1): Previous theorem.

Graph-Based Classes of Structures

Let us call a class D of structures **graph-based** if it is of the form $S(\mathrm{C})$ for some class C of graphs.

Examples

- For every $w \geq 0$, the class of all structures of tree-width at most w is graph based.
- For every $f : \mathbb{N} \to \mathbb{N}$, the class of all structures of local tree-width at most f is graph based.

Essentially, the theorem says:

> *The homomorphism problem restricted to a recursively enumerable graph based class of structures is tractable iff the class has bounded tree-width.*

Hypergraph-Based Classes of Structures

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

The **hypergraph of** $\mathcal{A}$ is the hypergraph $\mathcal{H}(\mathcal{A})$ defined as follows:

- The vertex set is A.
- Hyperedges are all sets $\{a_1, \dots, a_k\} \subseteq A$ such that there is an R in the vocabulary of $\mathcal{A}$ such that such that $(a_1, \dots, a_k) \in R^{\mathcal{A}}$.

Example

Two structures with the same graph, but distinct hypergraphs:

$$\begin{aligned} \mathcal{A} &= \big(\{a, b, c\}, \{(a, a, b), (b, b, c), (c, c, a)\}\big) \\ \mathcal{B} &= \big(\{a, b, c\}, \{(a, b, c)\}\big). \end{aligned}$$

(Vocabulary consists of one ternary relation symbol.)

Tree-Decompositions of Hypergraphs

A **tree-decomposition** of a hypergraph $\mathcal{H} = (V, E)$ is a pair $(\mathcal{T}, \beta)$, where $\mathcal{T}$ is a tree and $\beta : T \to 2^V$ such that

(1) For every hyperedge $e \in E$ is a $t \in T$ such that $e \subseteq \beta(t)$.

(2) For every $v \in V$ the set $\{t \in T \mid v \in \beta(t)\}$ is non-empty and connected.

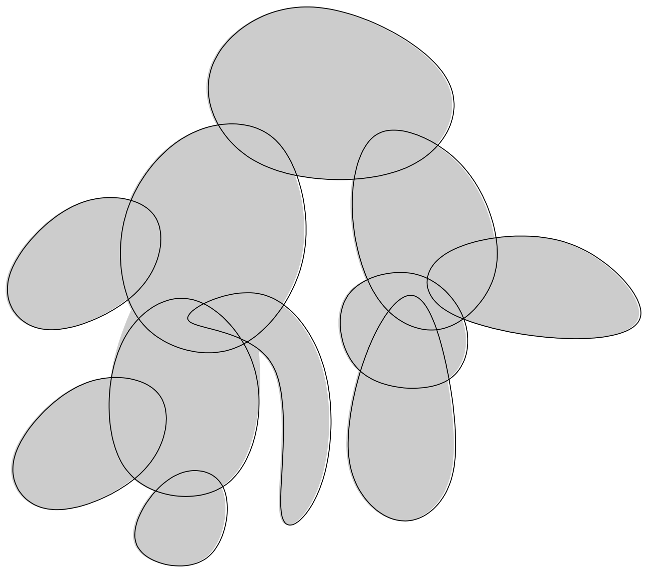
Remarks

- A tree-decomposition of the incidence structure representing a hypergraph is *not* the same as a tree-decomposition of the hypergraph.
- If $\mathcal{A}$ is a structure, then every tree-decomposition of $\mathcal{A}$ is a tree-decomposition of $\mathcal{H}(\mathcal{A})$ and vice versa.

Acyclic Hypergraphs

Definition

A hypergraph $\mathcal{H}$ is **acyclic** if it has a tree decomposition $(\mathcal{T}, \beta)$ where for all $t \in T$, $\beta(t)$ is a hyperedge of $\mathcal{H}$.



Theorem (Yannakakis 1982)

The homomorphism problem restricted to structures with an acyclic hypergraph is in polynomial time.

The size of a structure

- The **size** of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_m^{\mathcal{A}})$ is defined to be

$$||\mathcal{A}|| = m + |A| + \sum_{i=1}^{m} |R_i^{\mathcal{A}}| \cdot \text{arity}(R_i).$$

  This is the size of a reasonable representation of $\mathcal{A}$ on a computer.

- **Example:** The size of a graph with n vertices and m edges is $\Theta(n + m)$.

- The running time of Yannakakis's algorithm is polynomial in the size of the input structures and not the number of vertices.

Example

$\mathcal{A}$ arbitrary structure with universe $A = \{a_1, \ldots, a_k\}$,
R k-ary relation symbol not in the vocabulary of $\mathcal{A}$.

Let $\mathcal{A}'$ be the expansion of $\mathcal{A}$ with

$$R^{\mathcal{A}'} = \big\{(a_1, \ldots, a_k)\big\}.$$

Then $\mathcal{A}'$ is acyclic.

Testing whether $\mathcal{A}'$ has a homomorphism to a structure $\mathcal{B}'$ is easy:

- Each homomorphism maps $(a_1, \ldots, a_k)$ to a tuple in $R^{\mathcal{B}'}$, which completely determines the mapping.
- Thus we only have to step through all tuples $(b_1, \ldots, b_k) \in R^{\mathcal{B}'}$ and and check whether the mapping $a_i \mapsto b_i$ is a homomorphism.

This can be done in time linear in $\|\mathcal{B}'\|$.

Generalised Hypertree Decompositions

In the following, assume for each hypergraph $\mathcal{H}$ that for each vertex v the set $\{v\}$ is a hyperegde.

Definition

- A **generalised hypertree decomposition** of a hypergraph $\mathcal{H} = (V, E)$ is a triple $(\mathcal{T}, \beta, \gamma)$, where
  (1) $(\mathcal{T}, \beta)$ is a tree decomposition of $\mathcal{H}$,
  (2) $\gamma : T \to 2^E$ such that $\beta(t) \subseteq \bigcup \gamma(t)$ for all $t \in T$.
- The **width** of (T, β, γ) is max$\{\gamma(t) \mid t \in T\} + 1$.
- The **generalised hypertree-width** of $\mathcal{H}$, denoted by ghw$(\mathcal{H})$, is the maximum taken over the widths of all hypertree decompositions of $\mathcal{H}$.
- The **generalised hypertree-width** of a structure $\mathcal{A}$, denoted by ghw$(\mathcal{H})$, is ghw$(\mathcal{H}(\mathcal{A}))$.

Homomorphisms Restricted to Structures of Bounded GHW

Lemma (Gottlob, Leone, Scarcello 1999)
For every $w \geq 1$ the following problem is in polynomial time

Input:	Structures $\mathcal{A}, \mathcal{B}$ and generalised hypertree decomposition of $\mathcal{H}(\mathcal{A})$ of width $\leq w$.
Problem:	Decide if there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$.

Open Question

Let $w \geq 1$. Is there a polynomial time algorithm deciding whether a given hypergraph has generalised hypertree-width at most w?

Hypertree Decompositions

Definition

A **hypertree decomposition** of a hypergraph $\mathcal{H} = (V, E)$ is a triple $(\mathcal{T}, \beta, \gamma)$, where

(1) $(\mathcal{T}, \beta, \gamma)$ is a generalised hypertree decomposition of $\mathcal{H}$,

(2) For every $t \in \mathcal{T}$ we have

$$\bigcup \gamma(t) \cap \bigcup \{\beta(u) \mid u \text{ in subtree rooted in } t\} \subseteq \beta(t).$$

(Here we view $\mathcal{T}$ as a rooted tree.)

The **width** of a hypertree decomposition and the **hypertree width (hw)** of a hypergraph or structure are defined as usually.

Hypertree Decompositions (cont'd)

Theorem (Gottlob, Leone, Scarcello 1999)
A hypergraph is acyclic iff its hypertree width is at most 1.

Theorem (Gottlob, Leone, Scarcello 1999)
Let $w \geq 1$. Then the following problem is in polynomial time:

Input:	Hypergraph $\mathcal{H}$.
Problem:	Compute a hypertree decomposition of $\mathcal{H}$ of width at most w if there is one, otherwise reject.

Hypertree Decompositions vs Generalised Hypertree Decompositions

Proposition (Gottlob, G.)

For every hypergraph $\mathcal{H}$,

$$\mathsf{ghw}(\mathcal{H}) \leq \mathsf{hw}(\mathcal{H}) \leq 3\mathsf{ghw}(\mathcal{H}) + 1.$$

Homomorphisms Restricted to Structures of Bounded GHW

Theorem

For every $w \geq 0$, the homomorphism problem restricted to the class of structures of generalised hypertree-width at most w is in polynomial time.

Open Question

Is there a hypergraph based class D of structures of unbounded generalised hypertree width such that the homomorphism problem restricted to D is in polynomial time?