The Quest for a Logic Capturing PTIME

Martin Grohe
Humboldt-Universität zu Berlin
grohe@informatik.hu-berlin.de

Abstract

The question of whether there is a logic that captures polynomial time is the central open problem in descriptive complexity theory. In my talk, I will review the question and the early, mostly negative results that were obtained until the mid 1990s, and then move on to positive results about capturing polynomial time on specific classes of graphs. This will include recent results on definability in fixed-point logic and graph structure theory. Finally, I will dicuss stronger logics and propose directions for further research.

The purpose of this accompanying note is to give the basic definitions in detail, state the main results, mention some open problems, and give a list of references.

Introduction

The question for a logic capturing polynomial time originates in database theory. After Aho and Ullman [2] had realized that SQL, the standard query language for relational databases, cannot express all database queries computable in polynomial time, in a fundamental paper on the complexity and expressiveness of query languages, Chandra and Harel [9] asked for a recursive enumeration of the class of all queries computable in polynomial time. In the context of descriptive complexity theory, Gurevich [28] later asked whether there is a logic in which precisely the properties of finite structures decidable in polynomial time can be defined. In database terminology, Gurevich's question asks for a query language in which precisely the queries computable in polynomial time can be expressed. The advantages of such a language are obvious: On the one hand, syntactical restrictions, which can be checked by a compiler, guarantee that only queries admitting efficient evaluation can be asked. On the other hand, all queries that can be evaluated efficiently can be asked in the language.[1] It turned out that Chandra and Harel's question for a recursive enumeration of the PTIME queries is equivalent to Gurevich's question for a "logic for PTIME", up to a minor technical detail which will be discussed later.

Logics capturing PTIME

The exact formulation of the question for a "logic for PTIME" is subtle. We shall give the definition below, assuming some basic terminology from logic and finite model theory (for background, see [15, 21, 36, 40]).

In descriptive complexity theory, algorithmic problems are viewed as *Boolean queries*, that is, classes of finite structures of the same vocabulary that are closed under isomorphism. A *logic* L consists of:

(L1) a decidable set $\mathsf{L}[\tau]$, whose elements we call L-*sentences*, for each vocabulary τ;

(L2) a binary relation $\models_{\mathsf{L}}$ between finite structures and L-sentences such that for each τ and each $\varphi \in \mathsf{L}[\tau]$, the class $\mathscr{Q}_\varphi$ of all τ-structures A with $A \models_{\mathsf{L}} \varphi$ is closed under isomorphism.

Hence for every L-sentence φ the class $\mathscr{Q}_\varphi$ is a Boolean query, which we call the *query defined by* φ. A Boolean query is *definable* in L if it is defined by some L-sentence. A logic L *captures* PTIME if for every vocabulary τ:

(C1) Every Boolean query that is decidable in PTIME is definable in L.

(C2) There is a computable function that associates with every L-sentence $\varphi \in \mathsf{L}[\tau]$ a polynomial $p(X)$ and an algorithm M such that M decides the query $\mathscr{Q}_\varphi$ in time $p(n)$, where n is the size of the input structure.

Now the precise formulation of Gurevich's question is: *Is there a logic that captures* PTIME*?* This question is equivalent to Chandra and Harel's question for a recursive enumeration of all PTIME queries, except that condition (C2) needs to be replaced by the following slight modification:

(C2') There is a computable function that associates with every L-sentence $\varphi \in \mathsf{L}[\tau]$ an algorithm M such that M decides the query $\mathscr{Q}_\varphi$ in polynomial time.

[1] Of course it is questionable to use "polynomial time computable" as a synonym for "efficiently computable". However, the questions studied here are highly theoretical, and I do believe that PTIME is a reasonable and robust mathematical model for the class of efficiently computable problems.

Obviously, (C2) implies (C2'); it is an open question whether the converse also holds (in the presence of (L1), (L2), and (C1)). Let me remark that there is also a slightly modified version of a "recursive enumeration of all PTIME queries" that is precisely equivalent to the existence of logic satisfying (C2) (together with (L1), (L2), and (C1)). One might be tempted to replace (C2) or (C2') by the yet weaker and seemingly more natural condition:

(C2") Every Boolean query definable in L is decidable in polynomial time.

Note that, at least for the database application mentioned above, condition (C2") is too weak: It is not enough to know that a query given to the database system can be answered efficiently by some algorithm, the system also needs to be able to to find such an algorithm. This is what condition (C2') guarantees. Surprisingly, there is a logic that satisfies (C1) and (C2"), as the following example shows:

Example ([28, 43]). Let τ be a vocabulary and $\leq^\tau$ a binary relation symbol not contained in τ. Let us say that a sentence φ of vocabulary $\tau \cup \{\leq^\tau\}$ is *order invariant* on a τ-structure A if for any two linear orders $\leq_1$, $\leq_2$ on A we have

$$(A, \leq_1) \models \varphi \iff (A, \leq_2) \models \varphi.$$

Here $(A, \leq_i)$ denote the expansion of the τ-structure A to a $\tau \cup \{\leq^\tau\}$-structure where $\leq^\tau$ is interpreted by the linear order $\leq_i$.

I assume that the reader is familiar with *fixed-point logic* FP. The precise definition of the logic does not really matter here. I prefer to work with *inflationary fixed point logic* in this context, because it has the most straightforward extension to a logic that has the ability to count, but for this example, it is safe to think of the more familiar *least fixed point logic*.

We define a logic OFP with $\mathsf{OFP}[\tau] := \mathsf{FP}[\tau \cup \{\leq^\tau\}]$ for all vocabularies τ. Let us denote the the cardinality of the universe of a structure A by $|A|$. We define the relation $\models_{\mathsf{OFP}}$ as follows: For all τ-structures A and all $\varphi \in \mathsf{OFP}[\tau]$ we let

$A \models_{\mathsf{OFP}} \varphi \;:\iff\;$ φ is order invariant on all τ-structures B with $|B| \leq |A|$ and $(A, \leq) \models_{\mathsf{FP}} \varphi$ for some (and hence for all) linear orders $\leq$ of A.

OFP satisfies (L1) and (L2) because the logic FP does. The logic OFP satisfies (C1) by the Immerman-Vardi Theorem stated below. It also follows from the Immerman-Vardi Theorem that OFP satisfies (C2"). The key argument is that if a sentence $\varphi \in \mathsf{FP}[\tau \cup \{\leq^\tau\}]$ is not order invariant then there are only finitely many τ-structures A such that $A \models_{\mathsf{OFP}} \varphi$.

Remarkably, it is still an open question whether the logic OFP satisfies (C2) or (C2').

Gurevich conjectured that there is no logic for PTIME. I am not sure if I share this conjecture, but in any case it will be very difficult to prove it: By Fagin's Theorem [18], existential second-order logic captures NP. Hence if there is no logic capturing PTIME, then PTIME $\neq$ NP.

Capturing PTIME on classes of structures

The main positive results in the area state that certain logics capture PTIME on certain classes of structures. Let $\mathscr{C}$ be a class of structures, which we assume to be closed under isomorphism. Then a logic L *captures* PTIME *on* $\mathscr{C}$ if it satisfies the following two conditions for every vocabulary τ:

$(C1)_{\mathscr{C}}$ For every Boolean query $\mathscr{Q}$ decidable in PTIME there is an L-sentence φ such that for all structures $A \in \mathscr{C}$ it holds that $A \models \varphi$ if and only if $A \in \mathscr{Q}$.

$(C2)_{\mathscr{C}}$ There is a computable function that associates with every L-sentence $\varphi \in \mathsf{L}[\tau]$ a polynomial $p(X)$ and an algorithm M such that given a τ-structure $A \in \mathscr{C}$, the algorithm M decides if $A \models_{\mathsf{L}} \varphi$ in time $p(|A|)$.

Note that if the class $\mathscr{C}$ is decidable in PTIME, then $(C1)_{\mathscr{C}}$ is equivalent to the simpler condition stating that every Boolean query $\mathscr{Q} \subseteq \mathscr{C}$ that is decidable in PTIME is definable in L. In all known example, the class $\mathscr{C}$ is decidable in PTIME.

An *ordered structure* is a structure A whose vocabulary contains the binary relation symbol $\leq$, such that $\leq$ is interpreted in A by a linear order of the universe of A.

Immerman-Vardi Theorem ([34, 48]). *Fixed-point logic* FP *captures* PTIME *on the class of all ordered structures.*

Relations to isomorphism testing and canonisation

As an abstract question, the question for a logic capturing polynomial time is linked to the graph isomorphism and canonisation problem. In particular, if there is a polynomial time computable canonisation mapping for a class $\mathscr{C}$ of graphs or structures, then there is a logic that captures polynomial time on this class $\mathscr{C}$. This follows from the Immerman-Vardi Theorem. To explain this, let us consider graphs and assume that we represent them by their adjacency matrices. A *canonisation mapping* gets as argument some adjacency matrix representing a graph and returns a *canonical* adjacency matrix for this graph, that is, it maps *isomorphic* adjacency matrices to *equal* adjacency matrices. As an adjacency matrix for a graph is completely fixed once we specify the ordering of the rows and columns of the matrix, we may view a canonisation as a mapping associating with each graph a canonical ordered copy of the graph. Now we can apply the Immerman-Vardi Theorem to this ordered copy.

Clearly, if there is a polynomial time canonisation mapping for a class of graphs (or other structures) then there is a polynomial time isomorphism test for this class. It is open whether the converse also holds. It is an open question whether the existence of a logic for polynomial time implies the existence of a polynomial time isomorphism test or canonisation mapping.

Polynomial time canonisation mappings are known for many natural classes of graphs, for example planar graphs [33, 32], graphs of bounded genus [19, 42], graphs of bounded degree [4, 41], and graphs of bounded tree width [7]. For background, I refer the reader to Köbler's recent survey [38].

Most known capturing results are proved by showing that there is a canonisation mapping that is definable in some logic. Specifically, most of these results are for fixed-point logic with counting FP+C; they will be discussed in the next section. It was observed by Cai, Fürer, and Immerman [8] that for classes $\mathscr{C}$ of structures which admit a canonisation mapping definable in FP+C, a simple combinatorial algorithm known as the Weisfeiler-Leman (WL) algorithm [16, 17] can be used as a polynomial time isomorphism test on $\mathscr{C}$. This approach was used in [24, 26] to prove that the WL-algorithm correctly decides isomorphism on classes of graphs of bounded genus and on classes of bounded tree width. Recently, a refined version of the same approach was used by Verbitsky and others [27, 39, 49] to even obtain parallel isomorphism tests running in polylogarithmic time for planar graphs and graphs of bounded tree width.

Fixed-point logic with counting

We have seen in the previous section that for many natural classes of graphs there are logics capturing PTIME. However, the logics obtained through canonisation hardly qualify as natural logics. If a logic is to contribute to our understanding of the complexity class PTIME— and from my perspective this is the main reason for being interested in such a logic — we have to look for natural logics that derive their expressiveness from clearly visible basic principles like inductive definability, counting or other combinatorial operations, and maybe fundamental algebraic operations like computing the rank or the determinant of a matrix. If such a logic captures polynomial time on a class of structures, then this shows that all polynomial time properties of structures in this class are based on the principles underlying the logic.

A natural logic that was considered a candidate for a logic capturing PTIME for a while is *fixed-point logic with counting,* FP+C. The logic was first introduced, somewhat informally, by Immerman [35]. The formal definition that we use today, which is based on inflationary fixed-point logic, is due to Grädel and Otto [22]. Eventually, Cai, Fürer, and Immerman [8] gave an example of a Boolean query that is decidable in PTIME, but not definable in FP+C. Let us call this query the *CFI-query* in the following. The CFI-query witnesses that FP+C does not capture PTIME on the class of all graphs. Nevertheless, FP+C does capture PTIME on many interesting classes of structures, among them trees [37], planar graphs [23], all classes of structures of bounded tree width [26], and graphs that do not contain the complete 5-vertex graph as a minor [25]. Recall that a *minor* of graph G is a graph obtained from G by deleting edges, deleting vertices, and contracting edges. All classes $\mathscr{C}$ just listed can be *defined by excluded minors*, that is, there is a family $\mathscr{F}$ of graphs such that $\mathscr{C}$ is the class of all graphs that do not contain any graph in $\mathscr{F}$ as a minor. It is an awkward time for me to write this note, because I believe that I can prove that FP+C captures PTIME on any class of graphs defined by excluded minors. But I have not worked out all details of the proof yet, so at the time of writing this remains a conjecture.

As a matter of fact, FP+C captures PTIME on *almost all structures*. This statement was made precise and proved by Hella, Kolaitis, and Luosto [30].

Other logics

Not too many other logics have been studied as candidates for capturing polynomial time. One of the few is *choiceless polynomial time with counting*, CPT+C, a language based on Gurevich's abstract state machines. The language has been introduced by Blass, Gurevich, and Shelah in [5] and further studied in [6, 14]. In particular, Dawar, Richerby, and Rossman [14] proved that the CFI-query is definable in CPT+C. It is still open whether CPT+C captures PTIME.

Recent results by Atserias, Bulatov, and Dawar [3] nicely demonstrate that at the core of the inexpressibility results for FP+C is the logic's inability to define certain linear algebraic invariants that can easily be computed in polynomial time by Gaussian elimination. With my student Bastian Laubner, I am currently studying extensions of fixed-point logic by operators that express such algebraic invariants. Preliminary results are promising. For example, it is not hard to prove that a logic with an operator for the row rank of definable matrices can define the CFI-query.

Early on, a number of results regarding the possibility of capturing polynomial time by adding Lindström quantifiers to first-order logic or fixed-point logic were obtained. Hella [29] proved that adding finitely many Lindström quantifiers (or infinitely many of bounded arity) to fixed-point logic does not suffice to capture polynomial time (also see [12]). Dawar [11] proved that if there is a logic capturing polynomial time, then there is such a logic obtained from fixed-point logic by adding one vectorized family of Lindström quantifiers.

Another family of logics that have been studied in this context are extensions of fixed-point logic with nondeterministic choice operators [1, 13, 20].

Instead of capturing all PTIME queries contained in a specific class of structures, Otto [44, 45, 46] studied the question of capturing all PTIME queries satisfying certain invariance conditions. Most notably, he proved that bisimulation-invariant Boolean queries are decidable in polynomial time if and only if they are definable in the *higher-dimensional* μ*-calculus*.

Open problems

Several open problems were mentioned throughout this note. I would like to close by mentioning a few more:

Problem 1. *Find a natural logic that captures* PTIME *on classes of graphs of bounded degree.*

Remarks. It follows from the results on polynomial time isomorphism testing and canonisation of bounded degree graphs due to Babai and Luks [4, 41] that there is a logic capturing PTIME on these classes, so the emphasis is on "natural". It is known that FP+C does not capture PTIME on the class of graphs of degree 3, because the CFI-query is actually a query on graphs of degree 3.

Problem 2. *Find a logic that captures* PTIME *on classes of graphs of bounded rank width.*

Remarks. Rank width is a graph invariant similar to tree width that was introduced by Oum and Seymour [47]; it is closely related to *clique width* [10]. In particular, a class of graphs has bounded rank width if and only if it has bounded clique width.

No polynomial time isomorphism test for graphs of bounded rank width is known. It is conceivable that FP+C captures PTIME on classes of bounded rank width.

Problem 3. *Find a logic that captures all constraint satisfaction problems in* PTIME*. More precisely, find a logic* L *satisfying (L1), (L2), and (C2) such that for every constraint satisfaction problem* $\mathscr{P} = \text{CSP}(B)$*, if* $\mathscr{P}$ *is decidable in* PTIME *then there is there is an* L*-sentence* φ *that defines* $\mathscr{P}$.

Remarks. Constraint satisfaction problems can be defined as homomorphism problems for finite structures. For every τ-structure B, we let $\text{CSP}(B)$ be the class of all τ-structures A such that there is a homomorphism from A to B. Hence $\text{CSP}(B)$ is a Boolean query

Problem 4. *Prove that* CPT+C *does not capture polynomial time.*

Remarks. I think it would already be interesting to find a query in NP that is not definable in CPT+C. In general, for logics more expressive than FP+C it seems worthwhile to first separate them from NP. This may be viewed as a program of separating NP from larger and larger fragments of PTIME.

Acknowledgements. Thanks to Anuj Dawar, Phokion Kolaitis, Bastian Laubner, Martin Otto, and Victor Vianu for valuable comments on an earlier version of this paper.

References

[1] S. Abiteboul and V. Vianu. Datalog extensions for database queries and updates. *Journal of Computer and System Sciences*, 43:62–124, 1991.

[2] A. Aho and J. Ullman. The universality of data retrieval languages. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Programming Languages*, pages 110–120, 1979.

[3] A. Atserias, A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. In L. Arge, C. Cachin, T. Jurdzinski, and A. Tarlecki, editors, *Proceedings of the 34th International Colloquium on Automata, Languages and Programming*, volume 4596 of *Lecture Notes in Computer Science*, pages 558–570. Springer-Verlag, 2007.

[4] L. Babai and E. Luks. Canonical labeling of graphs. In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 171–183, 1983.

[5] A. Blass, Y. Gurevich, and S. Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100:141–187, 1999.

[6] A. Blass, Y. Gurevich, and S. Shelah. On polynomial time computation over unordered structures. *Journal of Symbolic Logic*, 67:1093–1125, 2002.

[7] H. Bodlaender. Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees. *Journal of Algorithms*, 11:631–643, 1990.

[8] J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992.

[9] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

[10] B. Courcelle and S. Olariu. Upper bounds to the clique-width of graphs. *Discrete Applied Mathematics*, 101:77–114, 2000.

[11] A. Dawar. Generalized quantifiers and logical reducibilities. *Journal of Logic and Computation*, 5:213–226, 1995.

[12] A. Dawar and L. Hella. The expressive power of finitely many generalized quantifiers. In *Proceedings of the 9th IEEE Symposium on Logic in Computer Science*, 1994.

[13] A. Dawar and D. Richerby. A fixed-point logic with symmetric choice. In M. Baaz and J. Makowsky, editors, *Proceedings of the 17th International Workshop on Computer Science Logic*, volume 2803 of *Lecture Notes in Computer Science*, pages 169–182. Springer-Verlag, 2003.

[14] A. Dawar, D. Richerby, and B. Rossman. Choiceless polynomial time, counting and the Cai-Fürer-Immerman graphs. *Annals of Pure and Applied Logic*, 152:31–50, 2008.

[15] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 2nd edition, 1999.

[16] S. Evdokimov, M. Karpinski, and I. Ponomarenko. On a new high dimensional Weisfeiler-Lehman algorithm. *Journal of Algebraic Combinatorics*, 10:29–45, 1999.

[17] S. Evdokimov and I. Ponomarenko. On highly closed cellular algebras and highly closed isomorphism. *Electronic Journal of Combinatorics*, 6:#R18, 1999.

[18] R. Fagin. Generalized first–order spectra and polynomial–time recognizable sets. In R. M. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings, Vol. 7*, pages 43–73, 1974.

[19] I. S. Filotti and J. N. Mayer. A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 236–243, 1980.

[20] F. Gire and H. Hoang. An extension of fixpoint logic with a symmetry-based choice construct. *Information and Computation*, 144:40–65, 1998.

[21] E. Grädel, P. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Springer-Verlag, 2007.

[22] E. Grädel and M. Otto. On Logics with Two Variables. *Theoretical Computer Science*, 224:73–113, 1999.

[23] M. Grohe. Fixed-point logics on planar graphs. In *Proceedings of the 13th IEEE Symposium on Logic in Computer Science*, pages 6–15, 1998.

[24] M. Grohe. Isomorphism testing for embeddable graphs through definability. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 63–72, 2000.

[25] M. Grohe. Definable tree decompositions. In *Proceedings of the 23rd IEEE Symposium on Logic in Computer Science*, 2008. These proceedings.

[26] M. Grohe and J. Mariño. Definability and descriptive complexity on databases of bounded tree-width. In C. Beeri and P. Buneman, editors, *Proceedings of the 7th International Conference on Database Theory*, volume 1540 of *Lecture Notes in Computer Science*, pages 70–82. Springer-Verlag, 1999.

[27] M. Grohe and O. Verbitzky. Testing graph isomorphism in parallel by playing a game. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, Part I*, volume 4051 of *Lecture Notes in Computer Science*, pages 3–14. Springer-Verlag, 2006.

[28] Y. Gurevich. Logic and the challenge of computer science. In E. Börger, editor, *Current trends in theoretical computer science*, pages 1–57. Computer Science Press, 1988.

[29] L. Hella. Definability hierarchies of generalized quantifiers. *Annals of Pure and Applied Logic*, 43:235–271, 1989.

[30] L. Hella, P. Kolaitis, and K. Luosto. Almost everywhere equivalence of logics in finite model theory. *Bulletin of Symbolic Logic*, 2:422–443, 1996.

[31] L. Hella, P. Kolaitis, and K. Luosto. How to define a linear order on finite models. *Annals of Pure and Applied Logic*, 87:241–267, 1997.

[32] J. Hopcroft and J. Wong. Linear time algorithm for isomorphism of planar graphs. In *Proceedings of the 6th ACM Symposium on Theory of Computing*, pages 172–184, 1974.

[33] J. E. Hopcroft and R. Tarjan. Isomorphism of planar graphs (working paper). In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*. Plenum Press, 1972.

[34] N. Immerman. Upper and lower bounds for first-order expressibility. *Journal of Computer and System Sciences*, 25:76–98, 1982.

[35] N. Immerman. Expressibility as a complexity measure: results and directions. In *Proceedings of the 2nd IEEE Symposium on Structure in Complexity Theory*, pages 194–202, 1987.

[36] N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999.

[37] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In A. Selman, editor, *Complexity theory retrospective*, pages 59–81. Springer-Verlag, 1990.

[38] J. Köbler. On graph isomorphism for restricted graph classes. In A. Beckmann, U. Berger, B. Löwe, and J. Tucker, editors, *Logical Approaches to Computational Barriers, Proceedings of the Second Conference on Computability in Europe*, volume 3988 of *Lecture Notes in Computer Science*, pages 241–256. Springer-Verlag, 2006.

[39] J. Köbler and O. Verbitsky. From invariants to canonization in parallel. *CoRR*, abs/cs/0608074, 2006. http://arxiv.org/abs/cs/0608074.

[40] L. Libkin. *Elements of Finite Model Theory*. Springer-Verlag, 2004.

[41] E. Luks. Isomorphism of graphs of bounded valance can be tested in polynomial time. *Journal of Computer and System Sciences*, 25:42–65, 1982.

[42] G. L. Miller. Isomorphism testing for graphs of bounded genus. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 225–235, 1980.

[43] A. Nash, J. B. Remmel, and V. Vianu. PTIME queries revisited. In T. Eiter and L. Libkin, editors, *Proceedings of the 10th International Conference on Database Theory*, volume 3363 of *Lecture Notes in Computer Science*, pages 274–288. Springer-Verlag, 2005.

[44] M. Otto. *Bounded variable logics and counting – A study in finite models*, volume 9 of *Lecture Notes in Logic*. Springer-Verlag, 1997.

[45] M. Otto. Canonization for two variables and puzzles on the square. *Annals of Pure and Applied Logic*, 85:243–282, 1997.

[46] M. Otto. Bisimulation-invariant PTIME and higher-dimensional μ-calculus. *Theoretical Computer Science*, 224:237–265, 1999.

[47] S.-I. Oum and P. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96:514–528, 2006.

[48] M. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on Theory of Computing*, pages 137–146, 1982.

[49] O. Verbitsky. Planar graphs: Logical complexity and parallel isomorphism tests. In W. Thomas and P. Weil, editors, *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science*, volume 4393 of *Lecture Notes in Computer Science*, pages 682–693. Springer-Verlag, 2007.