

# Research statement

Antonina Kolokolova

The main theme in my research is mathematical patterns and their computational complexity. This relates several fields of mathematics, with logic (proof theory and model theory) being the focus of my research so far, and algebra and combinatorics the areas I am exploring now.

In my PhD thesis I studied a general relationship between the complexity needed to describe a pattern and the complexity needed to prove its properties. In particular, I show that for “simple” classes of patterns corresponding to small complexity classes, their descriptive complexity and proof complexity coincide; for example, properties expressed using graph reachability can be proven in a low space complexity class NL. This line of research has practical applications. In a joint work with the computational logic group at Simon Fraser University, we develop a descriptive-complexity-based solver for general search problems; the solver is being submitted to a competition.

## 1 Logic and complexity

To talk about complexity we need a language for describing objects (patterns) in such a way that we can measure and compare them. Such a language is provided by mathematical logic, in particular finite model theory. From this viewpoint, patterns are expressed by formulae of some logic, and the complexity of a pattern corresponds to the complexity of evaluating a formula on a structure.

More formally, consider finite structures with their relations as uninterpreted variables. The *data complexity* is the complexity of testing the truth of a formula of small (constant) length on a structure of unbounded finite size. Here, the language of the formulae determines the complexity of the test: first-order formulae have small data complexity, whereas first-order formulae augmented with operators allowing a form of induction are more expressive, but harder to evaluate. For example, having a reachability operator makes it possible to ask whether there is a path between two elements with respect to some relation, but this operator cannot be expressed in first-order logic, or computed by an alternating logarithmic time algorithm. Thus, reachability as a property is strictly more complex than, for example, binary number addition.

## 2 Proving properties of “easy” patterns.

A different question is what we can prove if we are only allowed to use “small” objects, that is, objects that can be described using a weak logic. Suppose we need to build a reasoning

system knowing that the only objects it will see are the objects described by simple formulas. The power of a system changes depending on the objects it manipulates. This system should not be too weak, otherwise it would not be able to reason about the properties of the objects. But to be useful, a system should not be too powerful either, since otherwise it can become computationally infeasible and, eventually, undecidable. What are the minimal requirements for such a system?

A natural framework to study such question is bounded arithmetic: it is an area with a rich collection of well-developed techniques, and bounded arithmetic results translate directly into statements about proof systems. Unbounded arithmetic allows us to reason about general computable functions, whereas bounded arithmetic focuses on feasible computation. By Gödel's famous incompleteness theorem, there are true statements expressible but not provable in unbounded arithmetic. But it is an open question which "efficient" properties of "small" objects can be proven in systems of bounded arithmetic.

### 3 Our results.

In my PhD thesis, I pinpoint the relationship between the expressive power of a logic and the strength of a system of bounded arithmetic that operates with objects described by that logic. The main result of my thesis states that if a logic is provably closed, then the corresponding system of arithmetic has exactly the same power; otherwise it is weaker. Here, "provably closed" means that not only is the logic closed under first-order operations, but also that there exist "simple" proofs of that closure (there are no formulas in the proof that are more complicated than the formulas from the logic). This is presented in [Kol05].

We started with a system capturing the data complexity of first-order logic. For this system, we noticed that the known capture result is based on first-order logic and that the closure property is satisfied trivially. It was more challenging to build a system of arithmetic for the class  $P$  based on second-order Horn logic, and prove that the resulting system has the same power as the minimal known system capturing  $P$ ; this work is presented in [CK01] (full version [CK03]), where it was used to solve an open problem in bounded arithmetic.

The next step was to build a system of arithmetic for first-order logic with transitive closure, presented in [CK04]. This system allows us to reason about the complexity class  $NL$ , the class of problems reducible to directed graph reachability (this class is believed to be much smaller than the class  $P$  of polynomial-time computable predicates). That a non-trivial proof of the closure of  $NL$  under complementation can be formalized using only  $NL$  concepts is an interesting result in itself.

The situation is different for a seemingly similar complexity class  $SL$  of problems reducible to the reachability problem on undirected graphs. The only known proof of closure of  $SL$ , as well as the recent result that  $SL$  coincides with  $L$  (deterministic logspace) [Rei05], use expander graphs and algebraic reasoning, which is not known to be formalizable in such low complexity class as  $SL$ . We still manage to get a viable system of arithmetic, but its capture of  $SL$  is not as strong. One of my current projects is to estimate the complexity needed to reason about expander graphs.

## 4 Model Expansion framework and problem solving

In my thesis, I looked at the complexity of functions that witness existential quantifiers in fragments of second-order arithmetic. These ideas turned out to be very useful for a descriptive complexity-based solver project of Eugenia Ternovska, David Mitchell and their group. The key idea there was using existential second-order fragments of various logics by stating a search problem as finding an expansion of a given instance of a problem to a larger instance which contains a solution. For example, when solving 3-colourability, we encode it as a first-order formula with conditions on three free second-order variables representing colours, and the solution is an “expansion” of a model (in this case, a graph) by three relations corresponding to the colours. The commonly used approaches to problem solving either encode a problem as a SAT instance directly, or are based on stable-model semantic of Answer Set Programming.

A survey of results on complexity of this problem for various logics was presented at the Logic and Computational Complexity ([KLMT06]); a journal version is in preparation.

## 5 Future directions

My current project is formalizing reasoning that uses expander graphs. Understanding the power of this, and other algebraic concepts can be useful not only for determining the complexity of proofs such as  $SL=L$ , but also for cryptography problems: if the recent result that primality testing is polynomial-time solvable can be formalized in a system of bounded arithmetic  $S_2^1$  or weaker, then factoring can be done in polynomial time. I am also working on a project about finding “random-like” objects with respect to logics admitting zero-one law: there, it is known that for majority of formulae either a formula or its complement is almost certainly true, but the question of describing a small class of structures that witness such formulae (an analogue of a “hitting set” in complexity theory) is still open.

Recently I started collaborating with the data mining group at Simon Fraser University, led by Martin Ester. One research direction pursued by his group is constraint-based clustering, where a search for clusters in data is aided by background knowledge, e.g., a social network graph. Constraint-based clustering is a very new and rapidly evolving area, and I hope that logic representations of constraints can be useful in both creating algorithms and proving lower bounds in this field.

The study of complexity in mathematics is intrinsically an interdisciplinary area, exploring the connections between various subfields of mathematics and computer science, as well as relating to the concept of feasibility in other sciences. I am looking forward to continue learning new such connections, and welcome collaboration from other areas.

## References

- [CK01] S.A. Cook and A. Kolokolova. A second-order system for polynomial-time reasoning based on Grädel's theorem. In *Proceedings of the Sixteens annual IEEE symposium on Logic in Computer Science*, pages 177–186, 2001.
- [CK03] S.A. Cook and A. Kolokolova. A second-order system for polytime reasoning based on Grädel's theorem. *Annals of Pure and Applied Logic*, 124:193–231, 2003.
- [CK04] S.A. Cook and A. Kolokolova. Bounded arithmetic of NL. In *Proceedings of the Nineteens annual IEEE symposium on Logic in Computer Science*, pages 398–407, 2004.
- [KLMT06] Antonina Kolokolova, Yongmei Liu, David Mitchell, and Eugenia Ternovska. Complexity of expanding a finite structure and related tasks. In *LCC*, 2006.
- [Kol05] A. Kolokolova. Closure properties of weak systems of bounded arithmetic. In *Computer Science Logic: 19th International Workshop, CSL 2005, 14th Annual Conference of the EACSL, Oxford, UK, August 22-25, 2005. Proceedings*, volume 3634 of *LNCS*, pages 369–383, 2005.
- [Rei05] Omer Reingold. Undirected st-connectivity in log-space. In *STOC*, pages 376–385, 2005.