Research Résumé

Logic is an integral part of theoretical computer science. Its contributions range from the work of Gödel and Church, which had a major influence on the development of computability theory, to more recent activities in such diverse fields as verification, database theory, complexity theory, artificial intelligence, computer security, and computer linguistics.

My own research interest within logic is model theory for monadic second-order logic, a logic with strong connections to automata theory. More specifically, my work falls into the following related areas:

(i) algorithmic model theory;
(ii) model constructions and decompositions;
(iii) the expressive power of monadic second-order logic;
(iv) automata theory and algebraic language theory;
(v) algorithmic issues.

(i) Algorithmic model theory

The topic of algorithmic model theory is the study of algorithmic properties of *infinite* structures. This theory developed out of several, quite diverse fields of computer science. In finite model theory, where one studies the connections between logic and algorithmic properties of finite structures, there was a desire to extend the scope of the theory to include infinite structures. An early example is the article [9] on least fixed-point logic on infinite structures. In verification, infinite state spaces have received more and more attention during the last two decades (see, e.g., [4] for a survey). Finally, in computer algebra researchers sought ways to represent, and compute with, infinite groups and fields. For the fundamental groups of certain spaces, they invented a representation by finite automata which lead to the study of the so-called automatic groups [6].

In order for an algorithm to process infinite structures, a finite encoding of such structures is required. For instance, when representing the state space of a program one can encode the stack contents as finite words and the transitions between them by rewriting rules. Consequently, finite encodings of structures play an important part in algorithmic model theory. The central questions regarding a given encoding are:

(a) Which properties are decidable when the input is encoded in this way?
(b) Which structures have a representation of this kind?

Most of the time, the algorithmic properties of a given type of representation can be determined easily. But characterising those structures that possess such a representation is usually highly non-trivial.

Previous work. My own contributions to this field concentrate on characterisation results. The following classes of finitely presented structures were investigated:

(a) automatic structures [T3,C8,C5,J16],

(b) tree-interpretable structures [C6,J17],

(c) the Caucal hierarchy [J12].

The graphs of the Caucal hierarchy, for instance, correspond to the configuration graphs of higher-order pushdown automata and can, thus, be used to encode the state spaces of functional programs. The focus of my papers was on methods to show that a given structure does *not* belong to the class under consideration. The formal framework proposed in this work consists of using logical operations to represent infinite structures. In the meantime this approach has become a standard technique in the field.

(ii) Model constructions and decompositions

An invaluable tool in algorithmic model theory are operations that are compatible with a given logic in the sense that, when applying the operation, one can decide whether a formula holds in the result by checking certain other formulae on the arguments. In particular, such an operation can be used to transfer decidability from the given structures to the new one. Classical examples of compatible operations are interpretations, the products of Feferman and Vaught [8] (for first-order logic) and the generalised sums of Shelah [13] (for monadic second-order logic). A more recent example is the Muchnik iteration [14] which generalises the unravelling of a graph and which can be used to generate the state space of programs with a stack. I have contributed to a survey [H1] which gives an overview with emphasis on operations that are compatible with first-order logic or with monadic second-order logic. Such operations have many applications.

(i) In algorithmic model theory one can use them to represent (infinite) structures by finite terms: each structure that can be obtained by a finite sequence of operations can be represented by this sequence. Furthermore, if all operations used preserve decidability, all structures with such a representation will have a decidable theory. For instance, each structure in the Caucal hierarchy can be obtained by a finite number of Muchnik iterations followed by a monadic second-order interpretation. Since these two operations preserve decidability of the monadic second-order theory, we can evaluate monadic second-order formulae on each structure from the Caucal hierarchy. This approach unifies the definition of most classes considered in the literature, which originally were based on ad-hoc methods based on automata, term rewriting systems, grammars, etc.

(ii) Such operations yield a notion of reduction between classes of structures. For instance, every structure that one can interpret in a structure with decidable theory also has a decidable theory. Hence, such operations are a tool to provide both decidability and undecidability results.

(iii) Besides algorithmic applications of term representations one can also use terms to obtain decompositions of representable structures. For example, the operations used to define the so-called HR-equational graphs lead to the notion of tree decomposition and tree width. In that way, depending on the choice of operations, it is possible to develop a structure theory for the given class. In particular, such a theory provides tools to prove that certain structures can not be represented by a given way of encoding. For instance, no graph of infinite tree width is HR-equational.

Previous work. I have mainly worked on three sets of operations:

(a) the operations associated with the notion of partition width;
(b) the operations associated with the notion of tree width;
(c) the Muchnik iteration.

In addition I would like to mention a handbook chapter [H1] that I have co-authored and that contains a survey on common operations that are compatible with first-order logic or monadic second-order logic and some of their applications.

The operations (a) are studied in [J14,T2]. The topic of these articles is the class of structures that can be interpreted in a tree. These structures can be characterised in several ways:

(1) They can be interpreted by monadic second-order logic in a tree.
(2) They have a hierarchical decomposition where a certain complexity measure (the partition width) is bounded.
(3) They can be built up from finite structures using (i) disjoint unions and (ii) quantifier-free interpretations.

Such a characterisation helps us to understand this class better. From (1) we can immediately deduce several decidability results, while (2) provides a tool to prove that certain structures do not belong to the class.

The operation (c) is the topic of [H2,J15]. The first paper is a survey on known results, while the second article extends these results by showing that the Muchnik iteration is also compatible with several extensions of monadic second-order logic.

The article [J13] uses operation (a) and (b) to define various algebras of finite structures and to study the corresponding notions of recognisable and equational classes.

(iii) Expressive power

Studying the expressive power of formalisms is an established area of computer science. It has received recent attention in, for instance, the work on database query languages like XML, the semantic web, verification, or computer linguistics. The reason is that, for applications, it is important to find the right balance between expressive power and algorithmic managability: a formalism needs to be powerful enough to express everything needed for the given application, but it should not be that expressive that it does not admit efficient algorithms. In short one can say that a formalism should be as expressive as necessary, but as efficient as possible.

I am mostly interested in the expressive power of monadic second-order logic and its variants, which encompass most of the logics used in applications. While decision procedures for monadic second-order logic are usually of prohibitive complexity, they frequently serve as templates from which one can derive specialised, much more efficient algorithms for fragments of monadic second-order logic, like the temporal logics used in verification.

During the last decades great advances have been made concerning the expressivity of monadic second-order logic. Of particular interest in this context are questions of definability and interpretability in given structures. For example, I have shown in [J14] that a structure can be interpreted in some tree if, and only if, its partition width is finite.

At the current time the situation concerning the model theory of monadic second-order logic looks as follows. On the one hand, there are structures whose monadic theory is simple enough such that we can develop a theory for them. All known examples of such structures have the property of being interpretable in a tree. On the other hand, there are structures whose monadic theory is extremely complicated. A prominent example are structures containing large definable grids. According to a conjecture of Seese [12] these two extremes form a dichotomy: either a structure can be interpreted in a tree, or it contains a large grid.

Previous work. My work on expressivity questions concerns

(a) interpretations;
(b) guarded second-order logic;
(c) the Muchnik iteration.

(a) One emphasis of my work is on different kinds of interpretations. For transductions – a strong form of interpretation – and classes of finite structures we obtained a complete description of the resulting hierarchy in [J10]. In particular, we developed concrete combinatorial criteria for the existence of a transduction between two given classes of finite structures.

When considering the conjecture of Seese, interpretations in trees are of particular importance. These are the subject of [J14,T2], where structures interpretable in trees are characterised in various ways. In particular, it is shown that these are exactly those structures whose partition width is bounded.

At the other extreme there are structures containing definable grids or pairing functions. Such structures were studied in [J9,J8]. The main result is a proof of a weak variant of Seese's conjecture.

(b) A further topic of my work concerns variants of monadic second-order logic. An important extension of this logic is the so-called guarded second-order logic. In general, it is strictly more expressive than monadic second-order logic. But, according to a result of Courcelle [5], on countable sparse structures the expressive power of guarded second-order logic collapses to that of monadic second-order logic. The article [J11] contains, among other results on the expressive power of guarded second-order logic, a generalisation of Courcelle's result to sparse structures of arbitrary cardinality.

(c) A large class of structures with a decidable monadic second-order theory is the Caucal hierarchy. Each of these structures has a finite partition width. In [J12] we study methods to prove that certain structures do not belong to a given level of the hierarchy.

(iv) Automata theory and algebraic language theory

Automata theory is one of the oldest parts of computer science with a wide range of applications, for instance, in compiler design, verification, and computer linguistics. There is a tight connection between automata and the monadic second-order theories of certain structures, like the order of the natural numbers or the infinite binary tree. In particular, Büchi [3] and Rabin [11] have shown that one can obtain decision procedures for these logics by translating formulae into automata. Automata theory has therefore become an essential tool in the investigation of monadic second-order logic.

Besides using monadic second-order logic one can also characterise regular languages algebraically via homomorphisms into finite algebras. While automata based algorithms are usually more efficient than those based on algebraic techniques, the algebraic point of view is particularly suited to classify fragments of monadic second-order logic and to develop corresponding decision procedures. For instance, one can decide the question of whether a given formula of monadic second-order logic is equivalent to a first-order formula over the class of finite words by constructing the syntactic monoid of the formula. Although this result is already quite old, no automaton-based decision procedure for this question could be devised so far.

There are well-developed algebraic theories for languages of finite and infinite words [10]. For languages of finite trees a preliminary theory has also been developed [2, 7], but for infinite trees only partial results exists [1, J7,J4]. The main obstacle in the development of an algebraic language theory in this context are missing combinatorial tools, like Ramseyan factorisation theorems for trees.

Previous work. My work so far deals with

(a) algebraic language theory;
(b) boundedness questions for automata and logics;
(c) higher-order pushdown automata.

(a) In algebraic language theory I have studied two settings: (1) finite graphs and (2) infinite trees.

Concerning (1), the paper [J13] investigated certain graph algebras motivated by the theory of graph grammars. We study the corresponding algebraic notions of recognisable and equational classes, and we relate them to the notion of definability in monadic second-order logic.

Concerning (2), I have made a first contribution [J7, J4] to the development of an algebraic language theory for languages of infinite trees. In particular, I have obtained a characterisation of the regular languages via homomorphisms in certain algebras.

(b) In [C4] we study fixed-point inductions of an monadic second-order formulae on finite words. [J2] extends these results to infinite trees. The main result is an automaton-based proof that it is decidable whether the length of these inductions is uniformly bounded.

(c) Besides considering automata as recognisers of languages, we can also use them to present infinite structures. For instance, the graphs in the Caucal hierarchy coincide with the configuration graphs of higher-order pushdown automata. In [J12] I use higher-order pushdown automata to study the classes in the Caucal hierarchy. In particular, I develop methods to prove that certain structures do not belong to a given level of the hierarchy.

(v) Finite model theory, descriptive complexity theory, and algorithmic issues

For many applications, one needs logics with the right balance of expressive power and algorithmic manageability. In many cases, in particular in verification and database theory, one can obtain such logics by extending some weak logic by fixed-point operators. This has led to a wide range of fixed-point logics.

Previous work. In this area I have worked on

(a) fixed-point logics;
(b) descriptive complexity theory;

(c) Ehrenfeucht-Fraïssé games.

(a) My work on fixed-point logics includes a survey [H3] on guarded fixed-point logic. We present automata-based algorithms for model checking and satisfiability testing for this logic, and we study the complexity of these problems.

I have also studied [C4,J2] fixed-point inductions of an monadic second-order formulae. The main result is a proof that it is decidable whether the length of such inductions is uniformly bounded.

(b) Descriptive complexity theory studies the correspondence between the computational complexity of classes of finite structures and the logics these classes can be axiomatised in. In [C7] I introduce a different setting where one considers the complexity and definability of sets definable in a fixed structure. Several complexity classes are characterised in this way.

(c) Ehrenfeucht-Fraïssé Games are one of the main model-theoretic tools in finite model theory. Unfortunately, on nontrivial structures the combinatorics involved in playing these games quickly become unmanageable. In [J6] I study several ways to simplify games and to decompose them into simpler subgames. While in the literature one mostly considers games on sparse structures, in this article we place the emphasis on structures that are not sparse.

Publications

Unpublished papers

[U1] *Logic, Algebra, and Geometry,* book in preparation. A draft is available at www.fi.muni.cz/~blumens

[U2] *A Syntactic Congruence for Infinite Trees,* in preparation.

Handbook chapters

[H1] (with Thomas Colcombet and Christof Löding) *Logical theories and compatible operations,* in Logic and Automata (J. Flum, E. Grädel, T. Wilke, eds.), Amsterdam University Press, 2007, pp. 72–106.

[H2] (with Dietmar Berwanger) *The Monadic Theory of Tree-like Structures,* in Automata, Logic, and Infinite Games (E. Grädel, W. Thomas, T. Wilke, eds.), LNCS 2500 (2002), pp. 285–301.

[H3] (with Dietmar Berwanger) *Automata for Guarded Fixed Point Logics,* in Automata, Logic, and Infinite Games (E. Grädel, W. Thomas, T. Wilke, eds.), LNCS 2500 (2002), pp. 343–355.

Journal articles

[J1] (with David Janin) *A Syntactic Congruence for Languages of Birooted Trees,* Semigroup Forum, 91 (2015), pp. 675–698.

[J2] (with Martin Otto and Mark Weyer) *Decidability Results for the Boundedness Problem,* Logical Methods in Computer Science, 10 (2014).

[J3] (with Bruno Courcelle) *Monadic second-order definable graph orderings,* Logical Methods in Computer Science, 10 (2014).

[J4] *An Algebraic Proof of Rabin's Tree Theorem,* Theoretical Computer Science, 478 (2013), pp. 1–21.

[J5] *Erratum to "On the structure of graphs in the Caucal hierarchy",* Theoretical Computer Science, 475 (2013), pp. 126–127.

[J6] *Locality and Modular Ehrenfeucht-Fraïssé Games,* Journal of Applied Logic, 10 (2012), pp. 144–162.

[J7] *Recognisability for Algebras of Infinite Trees,* Theoretical Computer Science, 412 (2011), pp. 3463–3486.

[J8] *Simple Monadic Theories and Partition Width,* Mathematical Logic Quarterly, 57 (2011), pp. 409–431.

[J9] *Simple Monadic Theories and Indiscernibles,* Mathematical Logic Quarterly, 57 (2011), pp. 65–86.

[J10] (with Bruno Courcelle) *The Monadic Second-Order Transduction Hierarchy,* Logical Methods in Computer Science, 6 (2010).

[J11] *Guarded Second-Order Logic, Spanning Trees, and Network Flows,* Logical Methods in Computer Science, 6 (2010).

[J12] *On the Structure of Graphs in the Caucal Hierarchy,* Theoretical Computer Science, 400 (2008), pp. 19–45.

[J13] (with Bruno Courcelle) *Recognizability, Hypergraph Operations, and Logical Types,* Information and Computation, 204 (2006), pp. 853–919.

[J14] *A Model Theoretic Characterisation of Clique-Width,* Annals of Pure and Applied Logic, 142 (2006), pp. 321–350.

[J15] (with Stephan Kreutzer) *An Extension to Muchnik's Theorem,* Journal of Logic and Computation, 15 (2005), pp. 59–74.

[J16] (with Erich Grädel) *Finite Presentations of Infinite Structures: Automata and Interpretations,* Theory of Computing Systems, 37 (2004), pp. 641–674.

[J17] *Axiomatising tree-interpretable structures,* Theory of Computing Systems, 37 (2004), pp. 3–27.

Papers in refereed conferences

[C1] (with Thomas Colcombet and Paweł Parys), *On a Fragment of AMSO and Tiling Systems,* Proc. 33th Symposium on Theoretical Aspects of Computer Science STACS, 2016.

[C2] (with Olivier Carton and Thomas Colcombet), *Asymptotic Monadic Second-Order Logic,* Mathematical Foundations of Computer Science MFCS (1), 2014, pp. 87–98.

[C3] (with Thomas Colcombet, Denis Kuperberg, Paweł Parys, and Michael Vanden Boom), *Two-Way Cost Automata and Cost Logics over Infinite Trees,* Logic in Computer Science LICS, 2014.

[C4] (with Martin Otto and Mark Weyer) *Boundedness of Monadic Second-Order Formulae Over Finite Words,* ICALP, LNCS 5556 (2009), pp. 67–78.

[C5] (with Erich Grädel) *Finite Presentations of Infinite Structures: Automata and Interpretations,* Proc. 2nd Int. Workshop on Complexity in Automated Deduction, CiAD 2002.

[C6] *Axiomatising tree-interpretable structures,* Proc. 19th Int. Symp. on Theoretical Aspects of Computer Science, LNCS 2285 (2002), pp. 596–607.

[C7] *Bounded Arithmetic and Descriptive Complexity,* Proc. 14th Ann. Conference of the European Association for Computer Science Logic, LNCS 1862 (2000), pp. 232–246.

[C8] (with Erich Grädel) *Automatic Structures,* Proc. 15th IEEE Symp. on Logic in Computer Science, 2000, pp. 51–62.

Theses

[T1] *Simple Monadic Theories,* Habilitation Thesis, TU Darmstadt, 2008.

[T2] *Structures of Bounded Partition Width,* Ph.D. Thesis, RWTH Aachen, 2003.

[T3] *Automatic Structures,* Diploma Thesis, RWTH Aachen, 1999.

Preprints of all my papers are available from:

http://www.mathematik.tu-darmstadt.de/~blumensath/Publications.html

Literature

[1] M. Bojańczyk and T. Idziaszek, *Algebra for Infinite Forests with an Application to the Temporal Logic EF*, in Proc. 20th International Conference on Concurrency Theory, CONCUR, LNCS 5710, 2009, pp. 131–145.

[2] M. Bojańczyk and I. Walukiewicz, *Forest Algebras*, in Logic and Automata: History and Perspectives, J. Flum, E. Grädel, and T. Wilke, eds., Amsterdam University Press, 2007, pp. 107–132.

[3] J. R. Büchi, *On a decision method in restricted second-order arithmetic*, in Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science, 1962, pp. 1–11.

[4] E. Clarke, M. Minea, and F. L. Ţiplea, *Verification of Infinite-State Systems with Applications to Security*, IOS Press, 2006.

[5] B. Courcelle, *The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications*, Theoretical Computer Science, 299 (2003), pp. 1–36.

[6] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston, *Word Processing in Groups*, Jones and Bartlett Publishers, Boston, 1992.

[7] Z. Ésik and P. Weil, *Algebraic recognizability of regular tree languages*, Theoretical Computer Science, 340 (2005), pp. 291–321.

[8] S. Feferman and R. L. Vaught, *The first order properties of products of algebraic systems*, Fundamenta Mathematicae, XLVII (1959), pp. 57–103.

[9] J. Flum, *On the (infinite) model theory of fixed-point logics*, in Models, Algebras, and Proofs, X. Caicedo and C. Montenegro, eds., vol. 203 of Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, 1999, pp. 67–75.

[10] D. Perrin and J.-É. Pin, *Infinite Words – Automata, Semigroups, Logic and Games*, Elsevier, 2004.

[11] M. O. Rabin, *Decidability of second-order theories and automata on infinite trees*, Trans. Amer. Math. Soc., 141 (1969), pp. 1–35.

[12] D. Seese, *The structure of the models of decidable monadic theories of graphs*, Annals of Pure and Applied Logic, 53 (1991), pp. 169–195.

[13] S. Shelah, *The Monadic Second Order Theory of Order*, Annals of Mathematics, 102 (1975), pp. 379–419.

[14] I. Walukiewicz, *Monadic second-order logic on tree-like structures*, Theoretical Computer Science, 275 (2002), pp. 311–346.