Mikhail Raskin

Curriculum vitae

Personal details

Date and place of birth	21 August 1987, Moscow, Russia
Nationality	Russia
Address	Office 315, LaBRI (A30), University of Bordeaux, 351 course de la Libération F-33405 Talence CEDEX, France
Phone	+33 7 69 55 11 75
Email	mikhail.raskin@u-bordeaux.fr

Current position

Since Sep 2022 Maître de conferences, University of Bordeaux, LaBRI
($\approx$ lecturer / permanent assistant professor)

Previous positions

Apr 2022 – Sep 2022	Interim Professor (W2), Dept. of Informatics, Technical Univ. of Munich
Dec 2018 – Mar 2022	Scientific employee (Postdoc), Dept. of Informatics, Technical Univ. of Munich
Mar 2017 – Aug 2018	Postdoc, University of Bordeaux, LaBRI
Dec 2015 – Feb 2017	Postdoc, Department of CS, Aarhus University

Research interests

Special computation models. Quantum informatics. Combinatorial and probabilistic constructions in theoretical computer science. Algorithm design. Expressive power and properties of programming languages.

Education

2008–2014 PhD, Moscow State University, Dept. of mechanics and mathematics

PhD thesis supervisor: Prof. Nikolay K. Vereshchagin

PhD thesis title: Automata on infinite words: direct and semidirect products approach.

2002–2008 Master of Science, Independent University of Moscow

2003–2008 Master of Science, Moscow State University, Dept. of mechanics and mathematics (cum laude)

MSc thesis supervisor: Prof. Nikolay K. Vereshchagin

MSc thesis title: Partial orderings on measures on the set of infinite words over a two-symbol alphabet

1999–2003 Moscow State School N 57

Publications in preparation

1. Bruno Courcelle, Irène Durand, Michael Raskin. A unified algorithm for colouring graphs of bounded clique-width.

Publications

1. Michael Raskin. Modular population protocols. *International Symposium on Algorithmics of Wireless Networks*, 2024.

   Population protocols are a model of distributed computation intended for the study of networks of independent computing agents with dynamic communication structure. Each agent has a finite number of states, and communication opportunities occur nondeterministically, allowing the agents involved to change their states based on each other's states. Population protocols are often studied in terms of reaching a consensus on whether the input configuration satisfied some predicate. A desirable property of a computation model is modularity, the ability to combine existing simpler computations in a straightforward way. In the present paper we present a more general notion of functionality implemented by a population protocol in terms of multisets of inputs and outputs. This notion allows to design multiphase protocols as combinations of independently defined phases. The additional generality also increases the range of behaviours that can be captured in applications (e.g. maintaining the role distribution in a fleet of servers). We show that composition of protocols can be performed in a uniform mechanical way, and that the expressive power is essentially semilinear, similar to the predicate expressive power in the original population protocol setting.

2. Arnaud Casteigts, Michael Raskin, Malte Renken, Viktor Zamaraev. Sharp Thresholds in Random Simple Temporal Graphs. *SIAM Journal on Computing*, 2023.

   A graph whose edges only appear at certain points in time is called a temporal graph (among other names). Such a graph is temporally connected if each ordered pair of vertices is connected by a path which traverses edges in chronological order (i.e. a temporal path). In this paper, we consider a simple model of random temporal graph, obtained from an Erdős–Rényi random graph $GsimG_{n,p}$ by considering a random permutation π of the edges and interpreting the ranks in π as presence times.

   We give a thorough study of the temporal connectivity of such graphs and derive implications for the existence of several kinds of sparse spanners. It turns out that temporal reachability in this model exhibits a surprisingly regular sequence of thresholds. In particular, we show that, at p=log n/n, any fixed pair of vertices can a.a.s. reach each other; at 2 log n/n, at least one vertex (and in fact, any fixed vertex) can a.a.s. reach all others; and at 3 log n/n, all the vertices can a.a.s. reach each other, i.e. the graph is temporally connected. Furthermore, the graph admits a temporal spanner of size 2n+o(n) as soon as it becomes temporally connected, which is nearly optimal as 2n-4 is a lower bound. This result is quite significant because temporal graphs do not admit spanners of size O(n) in general (Kempe, Kleinberg, Kumar, STOC 2000). In fact, they do not even always admit spanners of size $o(n^2)$ (Axiotis, Fotakis, ICALP 2016). Thus, our result implies that the obstructions found in these works, and more generally, all non-negligible obstructions, must be statistically insignificant: nearly optimal spanners always exist in random temporal graphs.

All the above thresholds are sharp. Carrying the study of temporal spanners a step further, we show that pivotal spanners — i.e. spanners of size 2n-2 made of two spanning trees glued at a single vertex (one descending in time, the other ascending subsequently) — exist a.a.s. at 4 log n/n, this threshold being also sharp. Finally, we show that optimal spanners (of size 2n-4) also exist a.a.s. at p=4 log n/n. Whether this value is a threshold is open, we conjecture that it is.

For completeness, we compare the above results to existing results in related areas, including edge-ordered graphs, gossip theory, and population protocols, showing that our results can be interpreted in these settings as well, and that in some cases, they improve known results therein. Finally, we describe an intriguing connection between our results and Janson's celebrated results on percolation in weighted graphs.

3. A. R. Balasubramanian, Javier Esparza, Mikhail Raskin. Finding Cut-Offs in Leaderless Rendez-Vous Protocols is Easy. *Logical Methods in Computer Science*, 2023.

   In rendez-vous protocols an arbitrarily large number of indistinguishable finite-state agents interact in pairs. The cut-off problem asks if there exists a number B such that all initial configurations of the protocol with at least B agents in a given initial state can reach a final configuration with all agents in a given final state. In a recent paper [HornSangnier2020], Horn and Sangnier prove that the cut-off problem is equivalent to the Petri net reachability problem for protocols with a leader, and in EXPSPACE for leaderless protocols. Further, for the special class of symmetric protocols they reduce these bounds to PSPACE and NP, respectively. The problem of lowering these upper bounds or finding matching lower bounds is left open. We show that the cut-off problem is P-complete for leaderless protocols, NP-complete for symmetric protocols with a leader, and in NC for leaderless symmetric protocols, thereby solving all the problems left open in [HornSangnier2020].

4. Roland Guttenberg, Mikhail Raskin, Javier Esparza. Geometry of Reachability Sets of Vector Addition Systems. *Proceedings of International Conference on Concurrency Theory*, 2023.

   Vector Addition Systems (VAS), aka Petri nets, are a popular model of concurrency. The reachability set of a VAS is the set of configurations reachable from the initial configuration. Leroux has studied the geometric properties of VAS reachability sets, and used them to derive decision procedures for important analysis problems. In this paper we continue the geometric study of reachability sets. We show that every reachability set admits a finite decomposition into disjoint almost hybridlinear sets enjoying nice geometric properties. Further, we prove that the decomposition of the reachability set of a given VAS is effectively computable. As a corollary, we derive a simple algorithm for deciding semilinearity of VAS reachability sets, the first one since Hauschildt's 1990 algorithm. As a second corollary, we prove that the complement of a reachability set always contains an infinite linear set.

5. Ruben Becker, Arnaud Casteigts, Pierluigi Crescenzi, Bojana Kodric, Malte Renken, Michael Raskin, Viktor Zamaraev. Giant Components in Random Temporal Graphs. *International Conference on Randomization and Computation*, 2023.

   A temporal graph is a graph whose edges appear only at certain points in time. In these graphs, reachability among nodes relies on paths that traverse edges in chronological order (temporal paths). Unlike standard paths, temporal paths may not be composable, thus the reachability relation is not transitive and connected components (i.e., sets of pairwise temporally connected nodes) do not form equivalence classes, a fact with far-reaching consequences.

   Recently, Casteigts et al. [FOCS 2021] proposed a natural temporal analog of the seminal Erdős-Rényi random graph model, based on the same parameters n and p. The proposed model is obtained by randomly permuting the edges of an Erdős-Rényi random graph and interpreting this permutation as an ordering of presence times. Casteigts et al. showed that the well-known single threshold for connectivity in the Erdős-Rényi model fans out into multiple phase transitions for several distinct notions of reachability in the temporal setting.

   The second most basic phenomenon studied by Erdős and Rényi in static (i.e., non-temporal) random graphs is the emergence of a giant connected component. However, the existence of a similar phase transition in the temporal model was left open until now. In this paper, we settle this question. We identify a sharp threshold at p=(log n)/n, where the size of the largest temporally connected component increases from o(n) to n-o(n) nodes. This transition occurs significantly later than in the static setting, where a giant component of size n-o(n) already exists for any p=ω(1/n) (i.e. as soon as p is larger than a constant fraction of n). Interestingly, the threshold that we obtain holds for both open and closed connected components, i.e. components that allow, respectively forbid, their connecting paths to use external nodes — a distinction arising from the absence of transitivity.

   We achieve these results by strengthening the tools from Casteigts et al. and developing new techniques that provide means to decouple dependencies between past and future events in temporal Erdős-Rényi graphs, which could be of general interest in future investigations of these objects.

6. Mikhail Raskin. Protocols with constant local storage and unreliable communication. *Theoretical Computer Science*, 2022.

   Population protocols are a model of distributed computation intended for the study of networks of independent computing agents with dynamic communication structure. Each agent has a finite number of states, and communication occurs nondeterministically, allowing the involved agents to change their states based on each other's states.

   In the present paper we study unreliable models based on population protocols and their variations from the point of view of expressive power. We model the effects of message loss. We show that for a general definition of protocols with unreliable communication with constant-storage agents such protocols can only compute predicates computable by immediate observation (IO) population protocols (sometimes also called one-way protocols). Immediate observation population protocols are inherently tolerant to unreliable communication and keep their expressive power under a wide range of fairness conditions. We also prove that a large class of message-based models that are generally more expressive than IO becomes strictly less expressive than IO in the unreliable case.

7. Javier Esparza, Mikhail Raskin, Christoph Welzel. Computing Parameterized Invariants of Parameterized Petri Nets. *Fundamenta Informaticae*, 2022.

   A fundamental advantage of Petri net models is the possibility to automatically compute useful system invariants from the syntax of the net. Classical techniques used for this are place invariants, P-components, siphons or traps. Recently, Bozga et al. have presented a novel technique for the parameterized verification of safety properties of systems with a ring or array architecture. They show that the statement «for every instance of the parameterized Petri net, all markings satisfying the linear invariants associated to all the P-components, siphons and traps of the instance are safe» can be encoded in WS1S and checked using tools like MONA. However, while the technique certifies that this infinite set of linear invariants extracted from P-components, siphons or traps are strong enough to prove safety, it does not return an explanation of this fact understandable by humans. We present a CEGAR loop that constructs a finite set of parameterized P-components, siphons or traps, whose infinitely many instances are strong enough to prove safety. For this we design parameterization procedures for different architectures.

8. Javier Esparza, Mikhail Raskin, Christoph Welzel. Regular Model Checking Upside-Down: An Invariant-Based Approach. *Proceedings of International Conference on Concurrency Theory*, 2022.

   Regular model checking is a technique for the verification of infinite-state systems whose configurations can be represented as finite words over a suitable alphabet. It applies to systems whose set of initial configurations is regular, and whose transition relation is captured by a length-preserving transducer. To verify safety properties, regular model checking iteratively computes automata recognizing increasingly larger regular sets of reachable configurations, and checks if they contain unsafe configurations. Since this procedure often does not terminate, acceleration, abstraction, and widening techniques have been developed to compute a regular superset of the reachable configurations.

   In this paper we develop a complementary procedure. Instead of approaching the set of reachable configurations from below, we start with the set of all configurations and approach it from above. We use that the set of reachable configurations is equal to the intersection of all inductive invariants of the system. Since this intersection is non-regular in general, we introduce b-bounded invariants, defined as those representable by CNF-formulas with at most b clauses. We prove that, for every b>0, the intersection of all b-bounded inductive invariants is regular, and we construct an automaton recognizing it. We show that whether this automaton accepts some unsafe configuration is in EXPSPACE for every b>0, and PSPACE-complete for b=1. Finally, we study how large must b be to prove safety properties of a number of benchmarks.

9. Michael Raskin. QueryFS: compiling queries to define a filesystem. *European Lisp Symposium*, 2022.

   Personal computing devices store more and more loosely arranged data. Each new method of keeping track of the data supposes that the user stops using the old methods on this data. One of the more stable interfaces for data access is the filesystem API. However, the standard filesystem semantic provides a fixed and limited set of ways to search for data.

   QueryFS is a virtual filesystem for POSIX-like systems that compiles user-supplied queries in various DSLs via translation to Common Lisp code and represents the results as directories. The main current use-case is using it to navigate and process data stored or indexed in PostgreSQL with traditional tools (grep, find, vim etc.)

   This paper describes what practical usage of QueryFS looks like and what lies behind this.

10. Arnaud Casteigts, Michael Raskin, Malte Renken, Viktor Zamaraev. Sharp Thresholds in Random Simple Temporal Graphs. *Symposium on Foundations of Computer Science*, 2021.

    A graph whose edges only appear at certain points in time is called a temporal graph (among other names). Such a graph is temporally connected if each ordered pair of vertices is connected by a path which traverses edges in chronological order (i.e. a temporal path). In this paper, we consider a simple model of random temporal graph, obtained from an Erdős–Rényi random graph $GsimG_{n,p}$ by considering a random permutation π of the edges and interpreting the ranks in π as presence times.

    We give a thorough study of the temporal connectivity of such graphs and derive implications for the existence of several kinds of sparse spanners. It turns out that temporal reachability in this model exhibits a surprisingly regular sequence of thresholds. In particular, we show that, at p=log n/n, any fixed pair of vertices can a.a.s. reach each other; at 2 log n/n, at least one vertex (and in fact, any fixed vertex) can a.a.s. reach all others; and at 3 log n/n, all the vertices can a.a.s. reach each other, i.e. the graph is temporally connected. Furthermore, the graph admits a temporal spanner of size 2n+o(n) as soon as it becomes temporally connected, which is nearly optimal as 2n-4 is a lower bound. This result is quite significant because temporal graphs do not admit spanners of size O(n) in general (Kempe, Kleinberg, Kumar, STOC 2000). In fact, they do not even always admit spanners of size o(n^2) (Axiotis, Fotakis, ICALP 2016). Thus, our result implies that the obstructions found in these works, and more generally, all non-negligible obstructions, must be statistically insignificant: nearly optimal spanners always exist in random temporal graphs.

    All the above thresholds are sharp. Carrying the study of temporal spanners a step further, we show that pivotal spanners — i.e. spanners of size 2n-2 made of two spanning trees glued at a single vertex (one descending in time, the other ascending subsequently) — exist a.a.s. at 4 log n/n, this threshold being also sharp. Finally, we show that optimal spanners (of size 2n-4) also exist a.a.s. at p=4 log n/n. Whether this value is a threshold is open, we conjecture that it is.

For completeness, we compare the above results to existing results in related areas, including edge-ordered graphs, gossip theory, and population protocols, showing that our results can be interpreted in these settings as well, and that in some cases, they improve known results therein. Finally, we describe an intriguing connection between our results and Janson's celebrated results on percolation in weighted graphs.

11. Mikhail Raskin. Population protocols with unreliable communication. *International Symposium on Algorithmics of Wireless Networks*, 2021.

    Population protocols are a model of distributed computation intended for the study of networks of independent computing agents with dynamic communication structure. Each agent has a finite number of states, and communication occurs nondeterministically, allowing the involved agents to change their states based on each other's states.

    In the present paper we study unreliable models based on population protocols and their variations from the point of view of expressive power. We model the effects of message loss. We show that for a general definition of protocols with unreliable communication with constant-storage agents such protocols can only compute predicates computable by immediate observation (IO) population protocols (sometimes also called one-way protocols). Immediate observation population protocols are inherently tolerant to unreliable communication and keep their expressive power under a wide range of fairness conditions. We also prove that a large class of message-based models that are generally more expressive than IO becomes strictly less expressive than IO in the unreliable case.

12. Javier Esparza, Mikhail Raskin, Christoph Welzel. Abduction of trap invariants in parameterized systems. *International Symposium on Games, Automata, Logics, and Formal Verification*, 2021.

    In a previous paper we have presented a CEGAR approach for the verification of parameterized systems with an arbitrary number of processes organized in an array or a ring. The technique is based on the iterative computation of parameterized invariants, i.e., infinite families of invariants for the infinitely many instances of the system. Safety properties are proved by checking that every global configuration of the system satisfying all parameterized invariants also satisfies the property; we have shown that this check can be reduced to the satisfiability problem for Monadic Second Order on words, which is decidable.

    A strong limitation of the approach is that processes can only have a fixed number of variables with a fixed finite range. In particular, they cannot use variables with range [0,N-1], where N is the number of processes, which appear in many standard distributed algorithms. In this paper, we extend our technique to this case. While conducting the check whether a safety property is inductive assuming a computed set of invariants becomes undecidable, we show how to reduce it to checking satisfiability of a first-order formula. We report on experiments showing that automatic first-order theorem provers can still perform this check for a collection of non-trivial examples. Additionally, we can give small sets of readable invariants for these checks.

13. Michael Blondin, Mikhail Raskin. The Complexity of Reachability in Affine Vector Addition Systems with States. *Logical Methods in Computer Science*, 2021.

    Vector addition systems with states (VASS) are widely used for the formal verification of concurrent systems. Given their tremendous computational complexity, practical approaches have relied on techniques such as reachability relaxations, e.g., allowing for negative intermediate counter values. It is natural to question their feasibility for VASS enriched with primitives that typically translate into undecidability. Spurred by this concern, we pinpoint the complexity of integer relaxations with respect to arbitrary classes of affine operations.

    More specifically, we provide a trichotomy on the complexity of integer reachability in VASS extended with affine operations (affine VASS). Namely, we show that it is NP-complete for VASS with resets, PSPACE-complete for VASS with (pseudo-)transfers and VASS with (pseudo-)copies, and undecidable for any other class. We further present a dichotomy for standard reachability in affine VASS: it is decidable for VASS with permutations, and undecidable for any other class. This yields a complete and unified complexity landscape of reachability in affine VASS. We also consider the reachability problem parameterized by a fixed affine VASS, rather than a class, and we show that the complexity landscape is arbitrary in this setting.

14. Michael Blondin, Christoph Haase, Filip Mazowiecki, Mikhail Raskin. Affine Extensions of Integer Vector Addition Systems with States. *Logical Methods in Computer Science*, 2021.

    We study the reachability problem for affine Z-VASS, which are integer vector addition systems with states in which transitions perform affine transformations on the counters. This problem is easily seen to be undecidable in general, and we therefore restrict ourselves to affine Z-VASS with the finite-monoid property (afmp-Z-VASS). The latter have the property that the monoid generated by the matrices appearing in their affine transformations is finite. The class of afmp-Z-VASS encompasses classical operations of counter machines such as resets, permutations, transfers and copies. We show that reachability in an afmp-Z-VASS reduces to reachability in a Z-VASS whose control-states grow linearly in the size of the matrix monoid. Our construction shows that reachability relations of afmp-Z-VASS are semilinear, and in particular enables us to show that reachability in Z-VASS with transfers and Z-VASS with copies is PSPACE-complete. We then focus on the reachability problem for affine Z-VASS with monogenic monoids: (possibly infinite) matrix monoids generated by a single matrix. We show that, in a particular case, the reachability problem is decidable for this class, disproving a conjecture about affine Z-VASS with infinite matrix monoids we raised in a preliminary version of this paper. We complement this result by presenting an affine Z-VASS with monogenic matrix monoid and undecidable reachability relation.

15. Michael Raskin. Lisp in the middle: using Lisp to manage a Linux system. *European Lisp Symposium*, 2021.

    In the Lisp community one can still find some nostalgia for the time of Lisp machines. The defining feature that has been since lost is having a powerful programming language as the main method of controlling the system behaviour.

Unfortunately, to the best of our knowledge, there are few modern systems that try to revive this approach. Moreover, regardless of the configuration language in use, managing the system as a whole is usually associated purely with managing a global persistent state, possibly with parts of it getting enabled or disabled in runtime.

We present a system design and a description of a partial implementation of Lisp-in-the-middle, a system based on the common GNU/Linux/X11 stack that uses Common Lisp for runtime system policy and per-user policy. We prioritise ease of achieving compatibility with niche workflows, low rate of purely maintenance changes, and minimising the unnecessary interactions between the parts of the system unless requested by user.

16. Javier Esparza, Mikhail A. Raskin, Christoph Welzel. Computing Parameterized Invariants of Parameterized Petri Nets. *Proceedings of International Conference on Application and Theory of Petri Nets*, 2021.

    A fundamental advantage of Petri net models is the possibility to automatically compute useful system invariants from the syntax of the net. Classical techniques used for this are place invariants, P-components, siphons or traps. Recently, Bozga et al. have presented a novel technique for the parameterized verification of safety properties of systems with a ring or array architecture. They show that the statement «for every instance of the parameterized Petri net, all markings satisfying the linear invariants associated to all the P-components, siphons and traps of the instance are safe» can be encoded in WS1S and checked using tools like MONA. However, while the technique certifies that this infinite set of linear invariants extracted from P-components, siphons or traps are strong enough to prove safety, it does not return an explanation of this fact understandable by humans. We present a CEGAR loop that constructs a finite set of parameterized P-components, siphons or traps, whose infinitely many instances are strong enough to prove safety. For this we design parameterization procedures for different architectures.

17. Javier Esparza, Stefan Jaax, Mikhail Raskin, Chana Weil-Kennedy. The Complexity of Verifying Population Protocols. *Distributed Computing*, 2021.

    Population protocols [Angluin etal., PODC, 2004] are a model of distributed computation in which indistinguishable, finite-state agents interact in pairs to decide if their initial configuration, i.e., the initial number of agents in each state, satisfies a given property. In a seminal paper Angluin et al. classified population protocols according to their communication mechanism, and conducted an exhaustive study of the expressive power of each class, that is, of the properties they can decide [Angluin et al., Distributed Computing, 2007]. In this paper we study the correctness problem for population protocols, i.e., whether a given protocol decides a given property. A previous paper [Esparza et al., Acta Informatica, 2017] has shown that the problem is decidable for the main population protocol model, but at least as hard as the reachability problem for Petri nets, which has recently been proved to have non-elementary complexity. Motivated by this result, we study the computational complexity of the correctness problem for all other classes introduced by Angluin et al., some of which are less powerful than the main model. Our main results show that for the class of observation models the complexity of the problem is much lower, ranging from Pi_2^p to PSPACE.

18. A. R. Balasubramanian, Javier Esparza, Mikhail Raskin. Finding Cut-Offs in Leaderless Rendez-Vous Protocols is Easy. *International Conference on Foundations of Software Science and Computation Structures*, 2021.

    In rendez-vous protocols an arbitrarily large number of indistinguishable finite-state agents interact in pairs. The cut-off problem asks if there exists a number B such that all initial configurations of the protocol with at least B agents in a given initial state can reach a final configuration with all agents in a given final state. In a recent paper [HornSangnier2020], Horn and Sangnier prove that the cut-off problem is equivalent to the Petri net reachability problem for protocols with a leader, and in EXPSPACE for leaderless protocols. Further, for the special class of symmetric protocols they reduce these bounds to PSPACE and NP, respectively. The problem of lowering these upper bounds or finding matching lower bounds is left open. We show that the cut-off problem is P-complete for leaderless protocols, NP-complete for symmetric protocols with a leader, and in NC for leaderless symmetric protocols, thereby solving all the problems left open in [HornSangnier2020].

19. Irène Durand, Bruno Courcelle and Michael Raskin. On defining linear orders by automata. *Moscow Journal of Combinatorics and Number Theory*, 2020.

    We define linear orders $\leq_Z$ on product sets $Z := X_1 \times X_2 \times ... \times X_n$ and on subsets Z of $X_1 \times X_2$ where each composing set X_i is $[0, p]$ or $\mathbb{N}$, and ordered in the natural way. We require that $(Z, \leq_Z)$ be isomorphic to $(\mathbb{N}, \leq)$ if it is infinite. We want linear orderings of Z such that, in two consecutive tuples $\boldsymbol{z} = (z_1, ..., z_n)$ and $\boldsymbol{z}' = (z'_1, ..., z'_n)$, we have $|z_i - z'_i| \leq 1$ for each i. Furthermore, we define their *distance* $d(\boldsymbol{z}, \boldsymbol{z}')$ as the number of indices i such that $z_i \neq z'_i$. We will consider orderings where the distance of two consecutive tuples is at most 2. We are interested in algorithms that determine the tuple in Z following $\boldsymbol{z}$ by using local information, where

20. Michael Raskin, Chana Weil-Kennedy. Efficient Restrictions of Immediate Observation Petri Nets. *International Conference on Reachability Problems*, 2020.

    In a previous paper we introduced immediate observation Petri nets, a subclass of Petri nets with application domains in distributed protocols and theoretical chemistry (chemical reaction networks). IO nets enjoy many useful properties, but like the general case of conservative Petri nets they have a PSPACE-complete reachability problem. In this paper we explore two restrictions of the reachability problem for IO nets which lower the complexity of the problem drastically. The complexity is NP-complete for the first restriction with applications in distributed protocols, and it is polynomial for the second restriction with applications in chemical settings.

21. Michael Blondin, Mikhail Raskin. The Complexity of Reachability in Affine Vector Addition Systems with States. *Logic in Computer Science*, 2020.

Vector addition systems with states (VASS) are widely used for the formal verification of concurrent systems. Given their tremendous computational complexity, practical approaches have relied on techniques such as reachability relaxations, e.g. allowing for negative intermediate counter values. It is natural to question their feasibility for VASS enriched with primitives that typically translate into undecidability. Spurred by this concern, we pinpoint the complexity of integer relaxations w.r.t. arbitrary classes of affine operations.

More specifically, we provide a trichotomy on the complexity of integer reachability in VASS extended with affine operations (affine VASS). Namely, we show that it is NP-complete for VASS with resets, PSPACE-complete for VASS with (pseudo-)transfers and VASS with (pseudo-)copies, and undecidable for any other class. We further present a dichotomy for standard reachability in affine VASS: it is decidable for VASS with permutations, and undecidable for any other class. This yields a complete and unified complexity landscape of reachability in affine VASS.

22. Olivier Gauwin, Anca Muscholl, Michael Raskin. Minimization of visibly pushdown automata is NP-complete. *Logical Methods in Computer Science*, 2020.

    We show that the minimization of visibly pushdown automata is NP-complete. This result is obtained by introducing immersions, that recognize multiple languages (over a usual, non-visible alphabet) using a common deterministic transition graph, such that each language is associated with an initial state and a set of final states. We show that minimizing immersions is NP-complete, and reduce this problem to the minimization of visibly pushdown automata.

23. Mikhail A. Raskin, Chana Weil-Kennedy, Javier Esparza. Flatness and Complexity of Immediate Observation Petri Nets. *Proceedings of International Conference on Concurrency Theory*, 2020.

    In a previous paper we introduced immediate observation (IO) Petri nets, a class of interest in the study of population protocols and enzymatic chemical networks. In the first part of this paper we show that IO nets are globally flat, and so their safety properties can be checked by efficient symbolic model checking tools using acceleration techniques, like FAST. In the second part we study Branching IO nets (BIO nets), whose transitions can create tokens. BIO nets extend both IO nets and communication-free nets, also called BPP nets, a widely studied class. We show that, while BIO nets are no longer globally flat, and their sets of reachable markings may be non-semilinear, they are still locally flat. As a consequence, the coverability and reachability problem for BIO nets, and even a certain set-parameterized version of them, are in PSPACE. This makes BIO nets the first natural net class with non-semilinear reachability relation for which the reachability problem is provably simpler than for general Petri nets.

24. Michael A. Raskin, Mark Simkin. Perfectly Secure Oblivious RAM with Sublinear Bandwidth Overhead. *Proceedings of International Conference on the Theory and Application of Cryptology and Information Security*, 2019.

    Oblivious RAM (ORAM) has established itself as a fundamental cryptographic building block. Understanding which bandwidth overheads are possible under which assumptions has been the topic of a vast amount of previous works. In this work, we focus on perfectly secure ORAM and we present the first construction with sublinear bandwidth overhead in the worst-case. All prior constructions with perfect security require linear communication overhead in the worst-case and only achieve sublinear bandwidth overheads in the amortized sense. We present a fundamentally new approach for constructing ORAM and our results significantly advance our understanding of what is possible with perfect security.

Our main construction, Lookahead ORAM, is perfectly secure, has a worst-case bandwidth overhead of $O(sqrtN)$, and a total storage cost of $O(N)$ on the server-side, where N is the maximum number of stored data elements. In terms of concrete server-side storage costs, our construction has the smallest storage overhead among all perfectly and statistically secure ORAMs and is only a factor 3 worse than the most storage efficient computationally secure ORAM. Assuming a client-side position map, our construction is the first, among all ORAMs with worst-case sublinear overhead, that allows for a $O(1)$ online bandwidth overhead without server-side computation. Along the way, we construct a conceptually extremely simple statistically secure ORAM with a worst-case bandwidth overhead of $O(sqrtNfraclogNloglogN)$, which may be of independent interest.

25. Kristoffer Arnsfelt Hansen, Mikhail A. Raskin. A two-player stay-in-a-set game with perfect information and without Nash equilibria. *International Symposium on Games, Automata, Logics, and Formal Verification*, 2019.

    We give an example of a finite-state two-player turn-based stochastic game with safety objectives for both players which has no stationary Nash equilibrium. This answers an open question of Secchi and Sudderth.

26. Javier Esparza, Mikhail A. Raskin, Chana Weil-Kennedy. Parameterized Analysis of Immediate Observation Petri Nets. *Proceedings of International Conference on Application and Theory of Petri Nets*, 2019.

    We introduce immediate observation Petri nets, a class of interest in the study of population protocols (a model of distributed computation), and enzymatic chemical networks. In these areas, relevant analysis questions translate into parameterized Petri net problems: whether an infinite set of Petri nets with the same underlying net, but different initial markings, satisfy a given property. We study the parameterized reachability, coverability, and liveness problems for immediate observation Petri nets. We show that all three problems are in PSPACE for infinite sets of initial markings defined by counting constraints, a class sufficiently rich for the intended application. This is remarkable, since the problems are already PSPACE-hard when the set of markings is a singleton, i.e., in the non-parameterized case. We use these results to prove that the correctness problem for immediate observation population protocols is PSPACE-complete, answering a question left open in a previous paper.

27. Michael A. Raskin, Christoph Welzel. Working with first-order proofs and provers. *European Lisp Symposium*, 2019.

Verifying software correctness has always been an important and complicated task. Recently, formal proofs of critical properties of algorithms and even implementations are becoming practical. Currently, the most powerful automated proof search tools use first-order logic while popular interactive proof assistants use higher-order logic.

We present our work-in-progress set of tools that aim to eventually provide a usable first-order logic computer-assisted proof environment.

28. Mikhail Raskin. A superpolynomial lower bound for the size of non-deterministic complement of an unambiguous automaton. *International Colloquium on Automata, Languages, and Programming*, 2018.

    Unambiguous non-deterministic finite automata (UFA) are non-deterministic automata (over finite words) such that there is at most one accepting run over each input. Such automata are known to be potentially exponentially more succinct than deterministic automata, and non-deterministic automata can be exponentially more succinct than them.

    In this paper we establish a superpolynomial lower bound for the state complexity of the translation of an UFA to a non-deterministic automaton for the complement language. This disproves the formerly conjectured polynomial upper bound for this translation. This lower bound only involves a one letter alphabet, and makes use of the random graph methods.

    The same proof also shows that the translation of sweeping automata to non-deterministic automata is superpolynomial.

29. Michael A. Raskin. A linear lower bound for incrementing a space-optimal integer representation in the bit-probe model. *International Colloquium on Automata, Languages, and Programming*, 2017.

    We present the first linear lower bound for the number of bits required to be accessed in the worst case to increment an integer in an arbitrary space-optimal binary representation. The best previously known lower bound was logarithmic. It is known that a logarithmic number of read bits in the worst case is enough to increment some of the integer representations that use one bit of redundancy, therefore we show an exponential gap between space-optimal and redundant counters.

    Our proof is based on considering the increment procedure for a space optimal counter as a permutation and calculating its parity. For every space optimal counter, the permutation must be odd, and implementing an odd permutation requires reading at least half the bits in the worst case. The combination of these two observations explains why the worst-case space-optimal problem is substantially different from both average-case approach with constant expected number of reads and almost space optimal representations with logarithmic number of reads in the worst case.

30. Michael Raskin. Writing a best-effort portable code walker in Common Lisp. *European Lisp Symposium*, 2017.

One of the powerful features of the Lisp language family is possibility to extend the language using macros. Some of possible extensions would benefit from a code walker, i.e. a library for processing code that keeps track of the status of different part of code, for their implementation. But in practice code walking is generally avoided.

In this paper, we study facilities useful to code walkers provided by "Common Lisp: the Language" (2nd edition) and the Common Lisp standard. We will show that the features described in the standard are not sufficient to write a fully portable code walker.

One of the problems is related to a powerful but rarely discussed feature. The macrolet special form allows a macro function to pass information easily to other macro invocations inside the lexical scope of the expansion.

Another problem for code analysis is related to the usage of non-standard special forms in expansions of standard macros. We review the handling of defun by popular free software Common Lisp implementations.

We also survey the abilities and limitations of the available code walking and recursive macro expansion libraries. Some examples of apparently-conforming code that exhibit avoidable limitations of the portable code walking tools are provided.

We present a new attempt to implement a portable best-effort code walker for Common Lisp called Agnostic Lizard.

31. Ivan Damgård, Jesper Buus Nielsen, Antigoni Polychroniadou, Michael Raskin. On the Communication Required for Unconditionally Secure Multiplication. *Advances in Cryptology - CRYPTO Proceedings*, 2016.

    We present the following results:

    In the honest majority setting, as well as for dishonest majority with preprocessing, any gate-by-gate protocol must communicate $Omega(n)$ bits for every multiplication gate, where n is the number of players.

    In the honest majority setting, we show that one cannot obtain a bound that also grows with the field size. Moreover, for a constant number of players, amortizing over several multiplication gates does not allow us to save on the computational work, and – in a restricted setting – we show that this also holds for communication.

32. Michael Raskin, Nikita Mamardashvili. Accessing local variables during debugging. *European Lisp Symposium*, 2016.

    Any reasonably large program has to use local variables. It is quite common in the Lisp language family to also allow functions that exist only in a local scope. Scoping rules often allow compilers to optimize away parts of the local environment; doing that is good for performance, but sometimes inconvenient for debugging.

    We present a debugging library for Common Lisp that ensures access to the local variables during debugging. To prevent the optimisations from removing access to these variables, we use code-walking macros to store references to the local variables (and functions) inside global variables.

33. Krishnendu Chatterjee, Monika Henzinger, Sebastian Krinninger, Veronika Loitzenbauer, Michael A. Raskin. Approximating the minimum cycle mean. *Theoretical Computer Science*, 2014.

    A fast method for finding a cycle of approximately minimal mean edge weight in a directed graph with non-negative edge weights is offered using previously known appoximate matrix multiplication methods.

34. Michael Raskin. Data-transformer: an example of data-centered tool set. *European Lisp Symposium*, 2013.

    This paper describes the data-transformer library, which provides various input and output routines for data based on a unified schema. Currently, the areas of the library's use include storage and retrieval of data via CLSQL; processing CSV and similar tabular files; interaction with user via web forms. Using the supplied schema, the data-transformer library can validate the input, process it and prepare it for output. A data schema may also include channel-specific details, e.g. one may specify a default HTML textarea size to use when generating the forms.

35. Michael A. Raskin. Toom's Partial Order Is Transitive. *Problems of Information Transmission*, 2012.

    The main result of the article is the proof of transitivity of Toom's order relation on measures on bi-infinite sequences of plusses and minuses. The relation was initially proposed as a tool to study ergodicity of cellular automata with deletions.

36. Mikhail A. Raskin. Coupling of computable measures coordinated with an order relation is not always computable. *Vestnik Mosckovskogo Universiteta (MSU Bulletin), Series 1*, 2012.

    An example is given of two computable measures on infinite binary sequences such that there is a measure on pairs of infinite sequences such that the first projection of a random pair is almost surely larger than the second one and the projections of the measure are equal to original measures, but all such measures on pairs are uncomputable.

37. Mikhail A. Raskin. Lower Estimate of the Regulator of the Direct Product of almost Periodic and Periodic Sequences. *Vestnik Mosckovskogo Universiteta (MSU Bulletin), Series 1*, 2011.

    A lower bound is proved for almost periodicity regulator of almost periodic sequence coupled with a periodic sequence. This lower bound differs from known upper bound only in a multiplicative constant of in the count of iterations of regulator of almost periodicity of original sequence.

Reviewed talks

1. Michael Raskin. Modular population protocols. *Highlights of Logic, Games and Automata* (HIGHLIGHTS 2024), Bordeaux, France.

2. Michael Raskin. Modular population protocols. *International Symposium on Algorithmics of Wireless Networks* (ALGOWIN 2024), London, UK / online.
3. Ruben Becker, Arnaud Casteigts, Pierluigi Crescenzi, Bojana Kodric, Malte Renken, Michael Raskin, Viktor Zamaraev. Giant Components in Random Temporal Graphs. *International Conference on Randomization and Computation* (RANDOM 2023).
4. Arnaud Casteigts, Michael Raskin, Malte Renken, Viktor Zamaraev. Sharp Thresholds in Random Simple Temporal Graphs. *Algorithmic Aspects of Temporal Graphs* (AATG 2022), Paris, France.
5. Michael Raskin. QueryFS: compiling queries to define a filesystem. *European Lisp Symposium* (ELS 2022).
6. Mikhail Raskin. Population protocols with unreliable communication. *International Symposium on Algorithmics of Wireless Networks* (ALGOSENSORS 2021), online.
7. Michael Raskin. Lisp in the middle: using Lisp to manage a Linux system. *European Lisp Symposium* (ELS 2021), online.
8. Michael Raskin, Chana Weil-Kennedy. Efficient Restrictions of Immediate Observation Petri Nets. *International Conference on Reachability Problems* (RP 2020), online.
9. Michael Blondin, Mikhail Raskin. The Complexity of Reachability in Affine Vector Addition Systems with States. *International Conference on Reachability Problems* (RP 2020), online.
10. Michael Raskin. Bridging the stepping stones: using pieces of NixOS without full commitment. *NixCon*, 2020, online.
11. Michael Blondin, Mikhail Raskin. The Complexity of Reachability in Affine Vector Addition Systems with States. *Logic in Computer Science* (LICS 2020), online.
12. Kristoffer Arnsfelt Hansen, Mikhail A. Raskin. A two-player stay-in-a-set game with perfect information and without Nash equilibria. *International Symposium on Games, Automata, Logics, and Formal Verification* (GandALF 2019), Bordeaux, France.
13. Michael A. Raskin, Christoph Welzel. Working with first-order proofs and provers. *European Lisp Symposium* (ELS 2019), Genoa, Italy.
14. Mikhail Raskin. A superpolynomial lower bound for the size of non-deterministic complement of an unambiguous automaton. *International Colloquium on Automata, Languages, and Programming* (ICALP 2018), Prague, Czech Republic.
15. Michael A. Raskin. A linear lower bound for incrementing a space-optimal integer representation in the bit-probe model. *International Colloquium on Automata, Languages, and Programming* (ICALP 2017), Warsaw, Poland.

16. Michael Raskin. Writing a best-effort portable code walker in Common Lisp. *European Lisp Symposium* (ELS 2017), Brussels, Belgium.

17. Michael Raskin, Nikita Nikitenkov. Paradox of choice in social network games with product choice. *Game theory society congress* (GAMES 2016), Maastricht, Netherlands.

18. Michael Raskin, Nikita Mamardashvili. Accessing local variables during debugging. *European Lisp Symposium* (ELS 2016), Krakow, Poland.

19. Mikhail A. Raskin. Computable measures that are couplable but not computably couplable. *International Conference on Computability, Complexity and Randomness* (CCR 2013), Moscow, Russia.

20. Michael Raskin. Data-transformer: an example of data-centered tool set. *European Lisp Symposium* (ELS 2013), Madrid, Spain.

21. Mikhail Raskin, Yuri Pritykin. Almost periodicity and finite automata.. *Workshop on Infinite Words, Automata and Dynamics*, 2007, Ekaterinburg, Russia.

Seminar talks

1. Modular population protocols. *LaBRI, University of Bordeaux*, 2024, Bordeaux.

2. Protocols with constant local memory and unreliable communication. *LaBRI, University of Bordeaux*, 2024, Bordeaux.

3. Getting readable proofs for replicated systems from automated provers. *PaVeDyS project meeting*, 2024, Paris.

4. Coordination games on graphs. *LaBRI, University of Bordeaux*, 2023, Bordeaux.

5. Sharp Thresholds in Random Simple Temporal Graphs. *TEMPOGRAL project meeting*, 2023, Itteville, France.

6. Coordination games on graphs. *LaBRI, University of Bordeaux*, 2023, Bordeaux.

7. Getting readable proofs for replicated systems from automated provers. *LaBRI, University of Bordeaux*, 2023, Bordeaux.

8. Towards finite automata inside (Zooming through whatever is distributed). *LaBRI, University of Bordeaux*, 2022, Bordeaux.

9. Sharp Thresholds in Random Simple Temporal Graphs. *Department of Informatics, Technical University of Munich*, 2022, Munich.

10. Population protocols with unreliable communication. *Department of Informatics, Technical University of Munich*, 2022, Munich.

11. State complexity of complementing unambiguous automata. *Dagstuhl Seminar. Unambiguity in Automata Theory*, 2021, Dagstuhl, Germany.

12. Nearly optimal spanners in quite sparse random simple temporal graphs. *Dagstuhl Seminar. Temporal Graphs: Structure, Algorithms, Applications*, 2021, online.

13. Population protocols and complexity. *Department of Mechanics and Maths, Moscow State University*, 2021, online.

14. State complexity of complementing unambiguous finite automata. *Department of Informatics, Technical University of Munich*, 2021, Munich.

15. Complexity landscape of (Z-) A-VASS reachability. *Department of Informatics, Technical University of Munich*, 2020, Munich.

16. Immediate observation Petri nets. *Department of Mechanics and Maths, Moscow State University*, 2020, online.

17. Restrictions of IO reachability problem with lower complexity. *Department of Informatics, Technical University of Munich*, 2020, Munich.

18. Lower bounds on reachability complexity for Petri nets. *Department of Mechanics and Maths, Moscow State University*, 2020, online.

19. Expected time of gossip propagation. *Department of Mechanics and Maths, Moscow State University*, 2020, online.

20. Query-FS: Integrating with UNIX from Common Lisp via FS API. *Online Lisp Meeting*, 2020, online.

21. Courcelle's theorem and its variants. *Department of Informatics, Technical University of Munich*, 2019, Munich.

22. Population protocols with unreliable communication. *Department of Informatics, Technical University of Munich*, 2019, Munich.

23. State complexity of finite automata and the effects of operations on automata. *Department of Mechanics and Maths, Moscow State University*, 2018, Moscow.

24. Clique-width based graph algorithms. *Department of Computer Science, Higher School of Economics*, 2018, Moscow.

25. Degrees of ambiguity in finite automata. *Department of Mechanics and Maths, Moscow State University*, 2018, Moscow.

26. Enumerating colourings by colour renaming. *LaBRI, University of Bordeaux*, 2018, Bordeaux.

27. A superpolynomial lower bound for the size of non-deterministic complement of an unambiguous automaton. *LaBRI, University of Bordeaux*, 2018, Bordeaux.

28. A superpolynomial lower bound for the size of non-deterministic complement of an unambiguous automaton. *Department of Computer Science, Higher School of Economics*, 2017, Moscow.

29. Oblivious RAM constructions. *Department of Mechanics and Maths, Moscow State University*, 2017, Moscow.

30. Enumerating colourings by colour renaming. *GraphEn project meeting, LaBRI*, 2017, University of Bordeaux.

31. A linear lower bound for incrementing a space-optimal integer representation in the bit-probe model. *LaBRI, University of Bordeaux*, 2017, Bordeaux.

32. When learning works better than machine learning: Recovering damaged QR-codes with manual choice of image features to recognize. *Friday talk, Department of Computer Science, Aarhus University*, 2016, Aarhus.

33. A linear lower bound for incrementing a space-optimal integer representation in the bit-probe model. *Department of Computer Science, Aarhus University*, 2016, Aarhus.

34. Social network games. *Department of Computer Science, Aarhus University*, 2015, Aarhus.

35. Approximating the minimum cycle mean. *Department of Mechanics and Maths, Moscow State University*, 2014, Moscow.

36. QueryFS: a virtual filesystem based on queries. *Department of Mechanics and Maths, Moscow State University*, , Moscow.

37. Computable measures that are couplable but not computably couplable. *Department of Mechanics and Maths, Moscow State University*, , Moscow.

38. Cellular automata, cellular automata with deletions and Toom's relation on measures (short series of talks). *Department of Mechanics and Maths, Moscow State University*, , Moscow.

39. Goldreich's construction for secure multiparty computation (short series of talks). *Department of Mechanics and Maths, Moscow State University*, , Moscow.

40. On regulator of a finite transducer image of an almost periodic sequence. *Department of Mechanics and Maths, Moscow State University*, , Moscow.

Lightning talks

1. Lisp in the middle of web browsing. *European Lisp Symposium* (ELS 2019), Genoa, Italy.

2. Specifying code walking support. *European Lisp Symposium* (ELS 2019), Genoa, Italy.

3. Lisp-in-the-middle: Unifying system policies as Lisp code. *European Lisp Symposium* (ELS 2018), Marbella, Spain.

4. Lisp-in-the-middle, or I wanted a Lisp Machine and all I got was a fancy sudo. *European Lisp Symposium* (ELS 2017), Brussels, Belgium.

5. OpenCV wrapper and form processing for Common Lisp — implementation notes. *European Lisp Symposium* (ELS 2015), London, UK.

6. Julia: an outside view. *European Lisp Symposium* (ELS 2014), Paris, France.

7. Screen subareas in StumpWM via tagging. *European Lisp Symposium* (ELS 2014), Paris, France.

8. Personal software for PC: how I ended up reading email using Common Lisp. *European Common Lisp Meeting* (ELS 2013), Madrid, Spain.

9. QueryFS: a virtual filesystem based on queries and related tools. *European Lisp Symposium* (ELS 2012), Zadar, Croatia.

Conference awards

Best paper award. *International Conference on Application and Theory of Petri Nets, 2021*
Best paper award. *International Conference on Application and Theory of Petri Nets, 2019*

Meetings attended

1. Algorithmic Aspects of Temporal Graphs (AATG 2022), Paris, France.

2. European Lisp Symposium (ELS 2022), Porto, Portugal.

3. Dagstuhl seminar. Unambiguity in Automata Theory, 2021, Dagstuhl, Germany + online.

4. International Symposium on Games, Automata, Logics, and Formal Verification (GandALF 2021), Padova, Italy + online.

5. International Symposium on Algorithmics of Wireless Networks (ALGOSENSORS 2021), online.

6. European Lisp Symposium (ELS 2021), online.

7. Dagstuhl Seminar. Temporal Graphs: Structure, Algorithms, Applications, 2021, online.

8. International Conference on Reachability Problems (RP 2020), online.

9. NixCon, 2020, online.

10. QONFEST, 2020, online.

11. Symposium on Logic in Computer Science (LICS 2020), online.

12. International Workshop on Verification of Infinite-State Systems (INFINITY 2020), online.
13. European Lisp Symposium (ELS 2020), online.
14. International Symposium on Games, Automata, Logics, and Formal Verification (GandALF 2019), Bordeaux, France.
15. European Lisp Symposium (ELS 2019), Genoa, Italy.
16. Meeting of German scientists in the area of Concurrency Theory (D-CON 2019), Munich, Germany.
17. International Colloquium on Automata, Languages, and Programming (ICALP 2018), Prague, Czech Republic.
18. European Lisp Symposium (ELS 2018), Marbella, Spain.
19. International Colloquium on Automata, Languages, and Programming (ICALP 2017), Warsaw, Poland.
20. European Lisp Symposium (ELS 2017), Brussels, Belgium.
21. Game theory society congress (GAMES 2016), Maastricht, Netherlands.
22. Logical foundations of game theory (LOFT 2016), Maastricht, Netherlands.
23. European Symposium on Algorithms (ESA 2016), Aarhus, Denmark.
24. European Lisp Symposium (ELS 2016), Krakow, Poland.
25. European Lisp Symposium (ELS 2015), London, UK.
26. Games, Automata, Logic and Formal Verification (GandALF 2014), Verona, Italy.
27. European Lisp Symposium (ELS 2014), Paris, France.
28. Computability, Complexity and Randomness (CCR 2013), Moscow, Russia.
29. Games, Automata, Logic and Formal Verification (GandALF 2013), Borca di Cadore, Italy.
30. European Lisp Symposium (ELS 2013), Madrid, Spain.
31. European Lisp Symposium (ELS 2012), Zadar, Croatia.
32. Computer Science in Russia (CSR 2007), Ekaterinburg, Russia.

Scholarships

2005–2008 Kolmogorov Scholarship for students in mathematical logic and theory of algorithms

Teaching experience

University of Bordeaux (2022–):

- Administrative responsibility: CS department international student mobility coordinator
- Teaching assistant: «Models of programming and computation»
  (mandatory course)
- Teaching assistant: «Probability, statistics, and combinatorics»
  (mandatory course)
- Teaching assistant: «Networks»
  (mandatory course)
- Teaching assistant: «Technological project»
  (mandatory practical course)
- Lecturer and teaching assistant: «Development quality (intro to logic for program analysis)»
  (integrated course / mandtatory course)
- Lecturer and teaching assistant: «Introduction to C programming»
  (integrated course / mandtatory course)
- Teaching assistant: «Logic and proof»
  (mandatory course)
- Lecturer and teaching assistant: «Algorithms for elementary data structures»
  (integrated course / mandtatory course)

TU Munich (2019–2022):

- Supervision: Probabilistic population protocol models (BSc thesis)
- Supervision: Implementation of Finger Search Trees (BSc thesis)
- Supervision: Interactive Tool for Developing and Verifying First-Order Proofs
  in Integration with Existing Automated provers (BSc thesis)
- Supervision: Fast Simulation of Population Protocols (BSc thesis)
- Lecturer and Teaching assistant: «Fundamentals of algorithms and data structures»
  (mandatory course)
- Lecturer and Teaching assistant: Practical course «Algorithms for programming contests»
- Teaching assistant: Complexity theory

University of Bordeaux (2018):

- Teaching assistant: Array algorithms

Independent University of Moscow (2007–2015):

- Lecture courses: Introduction to probability theory, Set theory, Mathematical logic
- Teaching assistant: Geometry, Algebra, Mathematical analysis

Lomonosov Moscow State University (2015):

- Co-supervision of an MSc thesis project
  "Paradoxical examples of social network games with product choice"

Moscow Institute of Physics and Technology (2011–2014):

- Teaching Assistant: Mathematical logic, Algorithmic complexity.

Lecture Courses at the Summer School "Contemporary Mathematics" (2007–2019)

- “Between nonsense and unknowable. What we can claim, verify, prove”;
- “Toy examples of games”;
- “Blind counting”;
- “The objects that happen to exist” (probabilistic proofs of existence);
- “We cannot wait for favors from Nature” (forcing method in set theory model construction);
- “Who am I? Where am I?” (on sampling assumptions);
- A set of conventions and conventions about sets (a survey of alternative set theories);
- Cellular automata;
- Classical nonclassical logics and standard nonstandard models;
- Fortune-teller is of no use (what can and what cannot be predicted),
- Conditional probability and other probabilistic notions;
- Introduction to game theory;
- Sequences, close to periodical (with Yu.Pritykin).

Computer experience

Main languages:	Common Lisp, Python, Pascal (Free Pascal Compiler, Delphi), POSIX Shell/Bash, Nix
Other:	Julia, Scheme, C, C++, JavaScript, OCaml
System administration:	GNU/Linux servers; small networks with complicated routing

Professional programming

2017–2018 Design and implementation of a web UI for the TRAG collection of automata algorithms, as well as development and implementation of additional algorithms for TRAG. (LaBRI)

2008–2015 Design and implementation of web-based software systems for data collection, automated validation, unification and analysis in context of academic competitions for high-school and university students. Participation in organisation of competition events. (MCCME)

2004–2008 Development of software systems for automated data acquisition, processing and monitoring in engineering tests. (NATI)

Contributions to open source projects

NixPkgs	import software packages to the NIX platform, principal author of two accepted Nixpkgs RFCs, minor patches to the Nix package manager
Agnostic Lizard	author and current maintainer of a code walking library for Common Lisp
StumpWM	tag-based window management code for Stump Window Manager
CL-Emb	current maintainer
RelFS	symlink generation code for Relational filesystem
FunionFS	support for dynamical reconfiguration of the union filesystem

Service to community

Program committee member:

International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS): 2021
European Lisp Symposium (ELS): 2022, 2024
Symposium on Algorithmic Foundations of Dynamic Networks (SAND): 2024

Reviewer at request of PC members or editors:

Computer Science in Russia (CSR): 2011, 2012, 2018
European Symposium on Algorithms (ESA): 2018
European Lisp Symposium (ELS): 2018, 2023
Symposium on Theoretical and Applied Computer Science (STACS): 2020, 2024
International Symposium on Mathematical Foundations of Computer Science (MFCS): 2020, 2023
International Symposium on Automated Technology for Verification and Analysis (ATVA): 2020
International Conference on Application and Theory of Petri Nets: 2021
Computability in Europe Conference (CiE): 2021
International Colloquium on Automata, Languages, and Programming (ICALP): 2021, 2023
Information Processing Letters: 2021
Symposium on Fundamentals of Computation Theory (FCT): 2021
Data Compression Conference (DCC): 2022
Symposium on Algorithmic Foundations of Dynamic Networks (SAND): 2023
Computational Complexity Conference (CCC): 2023
International Conference on Concurrency Theory (CONCUR): 2023
International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM): 20

Summer school organising committee and scientific committee member:

Summer School «Contemporary Mathematics», 2017–2020

Participation in conference/workshop organisation:

Meeting of German scientists working in the area of Concurrency Theory (D-CON) 2019

Popularisation and outreach articles

1. Viktor Kleptsyn, Mikhail Raskin. Flight of glider (in memoria of John Conway) (in Russian). *N+1*, 2020.

Languages spoken

- Russian (native)
- English (advanced)
- French (good)