The Game of Synthesis

Nathanaël Fijalkow

Université de Bordeaux
École doctorale Mathématiques Informatique
Laboratoire Bordelais d'Informatique

Habilitation à diriger des recherches

The Game of Synthesis

Nathanaël Fijalkow

Defense held on 11 February 2022

Anca Muscholl	Professor at University of Bordeaux	*President*
Christel Baier	Professor at Technische Universität Dresden	*Reviewer*
Dana Fisman	Senior Lecturer	*Reviewer*
Sven Schewe	Professor at University of Liverpool	*Reviewer*
Borja Balle	Research Scientist at DeepMind	*Examiner*
Rupak Majumdar	Scientific Director at Max Planck Institute	*Examiner*
Prakash Panangaden	Professor at McGill University	*Examiner*

Since defending my PhD in October 2015, I have been working on a range of topics and questions pertaining to Synthesis. The dictionary entry for synthesis is 'the combination of components or elements to form a connected whole'. In this general definition synthesis covers quite a lot of today's scientific questions; this document will focus on the formalisation of some of them in the field of computer science. This manuscript describes the attempts we have made in the past years to understand, relate, and approach them.

This document is organised in non-technical independent sections each describing a research theme gathering a selection of the most recent scientific works I have contributed to and published between 2017 and today. The sections are named after one or two research colleagues, who have been in different ways influential to the topic at hand.

Some of my recent publications are not discussed in this document, a complete list can be found online:

`http://games-automata-play.com/`

Joël Ouaknine and James Worrell

I started working with Joël Ouaknine and James Worrell when joining their group as a Research Associate (postdoc) in Oxford in November 2015. They had obtained over the past five to ten years major contributions to a field they almost invented, or re-invented: the algorithmic study of linear dynamical systems (see [OW15] for a survey they wrote on the topic).

A dynamical system is given by an operator $f : X \to X$ and an initial point $x \in X$, describing an orbit $x, f(x), f^2(x), \ldots$. In the special case of linear dynamical systems, the operators are linear, meaning $f : \mathbb{R}^n \to \mathbb{R}^n$ is defined using a matrix $A \in \mathbb{R}^{n \times n}$ through $f(x) = A \cdot x$. Surprisingly, many algorithmic questions remain unanswered for linear dynamical systems: the most innocent looking and fundamental Skolem problem has been open for the better part of a century, it asks whether the first coordinate of $f^n(x)$ is zero for some $n \in \mathbb{N}$. Joël Ouaknine and James Worrell have shown in [OW14] that solving the related Positivity problem would require major advances in algebraic number theory, putting a lid on efforts to attack this problem directly.

The analysis of linear dynamical systems has many applications: one of them is program analysis. There are two lessons to be had from this setting: the first is that global statements of the form 'in this program this variable is never assigned value 0', akin to the Skolem problem, are very hard to obtain, and the second is that a more practical approach for this is to design (inductive) invariants to reason locally about programs.

This inspired us to study a variant of the Skolem problem: the existence of invariants for linear dynamical systems. More formally, given a linear dynamical system and a target point, does there exist an invariant certifying that the orbit never hits the target? Here an invariant is a set $\mathcal{I} \subseteq \mathbb{R}^n$ such that $x \in \mathcal{I}, y \notin \mathcal{I}$, and $f(\mathcal{I}) \subseteq \mathcal{I}$. Clearly, the existence of an invariant implies that y cannot be in the orbit of x under f. Most importantly, $\mathcal{I}$ is a certificate of that fact, which can be easily verified. Without further restriction, the converse also holds: the set $\mathcal{I} = \{x, f(x), f^2(x), \ldots\}$ is an invariant if and only if y is not in the orbit. This statement does not bring much clarity to the question: the set $\mathcal{I}$ is the orbit itself, and as witnessed by the difficulty of the Skolem problem, can

be a very complicated set. Hence our formulation of the invariant synthesis question is parameterised by a class of invariants: natural examples include the class of semialgebraic invariants (defined through polynomial inequalities) and the class of semilinear invariants (defined through linear inequalities). We refer to this problem as the Monniaux problem since it was originally formulated in a more general setting by David Monniaux in [Mon19] and also raised by him in a series of personal communications with various members of the theoretical computer science community over the past five years or so.

Asking the invariant synthesis question was the beginning of a daunting quest, giving a new perspective on the analysis of linear dynamical systems. Constructing an invariant circumvents the hardness of the Skolem problem and relates this line of work to decades of research from the field of abstract interpretation where heuristics for constructing invariants have been successfully developed and applied.

The first question we investigated was whether given a linear dynamical system and a target we can determine whether there exists a semialgebraic invariant separating the orbit from the target. Our first paper on this question [Fij+17] (and the journal version [Fij+19b]) shows a completeness result: there exists a closed semialgebraic invariant if and only if the target does not belong to the topological closure of the orbit. In some sense, semialgebraic invariants are expressive enough to separate an orbit from a point, although the orbit itself may not be semialgebraic. The proof of our result yields an algorithm for constructing the invariant when it exists, a key ingredient for compositional reasoning about programs.

Semialgebraic invariants are very expressive hence may prove hard to use so in many cases it is desirable to construct simpler invariants: the class of semilinear invariants has been widely used for this purpose. As Monniaux writes in [Mon19], 'We started this work hoping to vindicate forty years of research on heuristics by showing that the existence of polyhedral inductive separating invariants in a system with transitions in linear arithmetic (integer or rational) is undecidable.'. Indeed all existing practical implementations rely on heuristics for constructing invariants in this setting, and the existence of an algorithm would potentially have far reaching consequences. Unfortunately, there are no such algorithms: our second paper [Fij+19a] shows the undecidability of synthesising a semilinear invariant for dynamical systems with two linear operators, fulfiling Monniaux's goal of proving that heuristics are a necessary evil for constructing semilinear invariants. We contrast this negative result with a positive one: in the case of a single linear operator the problem is decidable, albeit very intricate.

The paper [Fij+19c] investigated the related scenario where the linear dynamical system is controlled at each step, and the paper [Bar+20a] introduced a subclass of linear dynamical systems for which the Skolem problem becomes tractable.

Prakash Panangaden

I met Prakash Panangaden at the Simons Institute for the Theory of Computing in Berkeley in Autumn 2016 during the Logical Structures in Computation semester. He accepted to be my mentor for the semester and introduced me to the beautiful notion of

probabilistic bisimulation about which he wrote one of the founding papers [Blu+97] and the reference textbook [Pan09]. Probabilistic bisimulation is a notion of equivalence for probabilistic systems (labelled Markov processes) whose strength is to be defined locally: two states are equivalent if any action from one state can be simulated by the other, leading to a distribution of equivalent states – note the co-inductive definition.

During the introductory lectures Prakash gave on the topic, he discussed one of the foundational result relating probabilistic bisimulation to logic [DEP98]: two states are not probabilistically bisimilar if and only if there exists a formula from probabilistic modal logic separating them. This extends the celebrated Hennessey-Milner theorem from transition systems to the probabilistic world and gives a dual perspective on bisimulation: non-bisimilarity can be certified using a simple logical formula.

Being trained in the field of Logic, Automata, and Games, I adopted the classic game involving Spoiler and Duplicator for defining bisimulation: this game formulation nicely formalises the co-inductive definition. Such a point of view did not exist for probabilistic bisimulation; I had to invent it and convince Prakash of its value. Together with the definition of the game we found a counter-example to the logical characterization above in the presence of uncountably many actions. This was the first negative result in that direction; a number of positive cases had been obtained using different assumptions and very different proof techniques (measure theory, coalgebra, domain theory).

Prakash Panangaden, Bartek Klin, and myself therefore embarked on a new venture: understanding the limits of the logical characterization for probabilistic bisimulation. This journey took us from sunny California to the topological and measure-theoretic realm of Polish spaces; we understood that a new result about analytic sets (extending the Unique Structure Theorem) was needed but its proof eluded us. As was to be expected, the city of Warsaw hosts some of the world leading experts on Polish spaces: the topologist Roman Pol helped us to prove the Positive Unique Structure Theorem. The consequence of this new theorem is a conceptually simple proof of the logical characterization theorem [FKP17] (see the journal version [Cle+19]), extending the existing results and matching the assumptions of the counter-example mentioned above.

The paper [Ped+18] studied the applications of probabilistic bisimulation for specifying timing properties, and the paper [FKS20] constructed algorithms for related equivalence relations close to probabilistic bisimulations.

Alexander Clark

I met Alexander Clark about the time I moved from (as already mentioned) sunny California to (rainy) London in January 2017. He is one of the most recognised authority on automata learning, a part of computational linguistics. His insightful talks connecting learning to logic have triggered my interests for this field which is both close – for its models and objectives – and far – for its tools, drawing from statistics, machine learning, and linguistics – from my automata theory background.

Alexander visited me at the Alan Turing Institute in Autumn 2018 and we started working together on learning probabilistic context-free grammars (PCFGs). Attacking

such a well studied problem with a long history meant getting familiar with a lot of ideas completely new to me. I learned a lot from this collaboration and this has been very influential for shaping my research since then. The goal of learning PCFGs is motivated (from a computational linguistic point of view) by understanding the process of first language acquisition, in particular the early phases of the acquisition of syntactic structure. The main difficulty is that in this framework we are learning from words and not derivation trees: it would indeed be much easier – albeit unrealistic – to learn if we had access to the syntactic decomposition of the words by the grammar.

It is well known that learning the whole class of PCFGs is impossible, so the objective here is to define subclasses of PCFGs which are both relevant for the goal stated above and can be effectively learned. To this end we took inspiration from a notion discovered three times in three different research communities: anchors in computational linguistics [SCH16], separation in complexity theory [Aro+16], and residual in automata theory [DLT02]. We showed that combining this anchored property with restrictions on ambiguity of the grammar yields a subclass of PCFGs identifiable from words which can be efficiently learned [CF20].

Olivier Serre

I know of Olivier Serre since my very first steps in research as his lecture notes on games were the first documents I read on the first day of my first research internship in 2009. I met him a few years later and have been in close contact with him ever since, in particular as we co-supervise Pierre Ohlmann for his PhD. Olivier Serre and coauthors introduced the notion of qualitative tree automata, a probabilistic semantics for automata over infinite trees [CHS14]. During my PhD we worked together on the topic and constructed an algorithm for deciding the emptiness of alternating qualitative Büchi tree automata [FPS13].

Automata over infinite trees is a very old topic already introduced in the 70s: the main result is due to Rabin [Rab69], it states the decidability of the monadic second-order logic (MSO) over the infinite binary tree. Sometimes referred to as 'the mother of all decidability results', this powerful result indeed implies many other decidability results through encoding into trees.

A major open question is whether there exist even more expressive decidable logics over the infinite binary tree; many attempts have been made in the past decades, leading to more open problems – for instance with the theory of regular cost functions [Col13a; Fij+15] – or to undecidability results – for instance for MSO + $\mathbb{U}$ [BPT16]. Probabilistic and measure-theoretic extensions of MSO have been introduced and decidability results obtained for large fragments [MSM18; Boj16], leaving open whether the extension of MSO with a probabilistic path quantifier is decidable. Bastien Maubert saw the connection between this question and alternating qualitative tree automata, which we used [Ber+21] to give a negative answer: MSO with a probabilistic path quantifier is undecidable[1]. Hence the search for a decidable extension of MSO over infinite trees remains active. Mikołaj Bojańczyk and coauthors have recently shown that in some sense there are no

[1]The same result has been obtained concurrently with very different techniques [BKS19].

such extensions [Boj+20].

The unexpected link to logic led us to sharpen our understanding of alternating qualitative tree automata; the paper [Ber+21] presents a crisp decidability frontier for the emptiness problem between Büchi and CoBüchi, and deep connections with games of imperfect information.

In another take on probabilistic automata, I have worked with Cristian Riveros and James Worrell on probabilistic automata with bounded ambiguity [FRW17], see also the journal version [FRW20]. Through a new correspondence with multi-objective optimisation we have constructed efficient algorithms for analysing them.

Thomas Colcombet

One of the main contributions of Thomas Colcombet (my former PhD co-advisor together with Mikołaj Bojańczyk) is the theory of regular cost functions [Col13b; Col13a]: this is a set of models (the main ones are cost automata and cost logics) and a set of algorithms for deciding boundedness properties on these models. The first example of the use of this toolbox is the star height problem: given a regular language L and a number k, does there exist a regular expression denoting L using at most k nesting of the star operator? An elegant solution to this problem is from L and k to construct a cost-automaton defining a function $f : \Sigma^* \to \mathbb{N} \cup \{\infty\}$ such that f is bounded if and only if the answer is positive. Intuitively, the cost-automaton is guessing a regular expression for the language and is checking its validity using the cost mechanisms (implemented by counters). Since checking whether a cost-automaton is bounded is decidable, this yields an algorithm computing the star height of a regular language. More generally, there are two types of results for the theory of regular cost functions:

- decidability results for boundedness problems applying to various models of the theory,
- reductions to boundedness problems inside the theory.

There are many examples of results of the second type, let me cite a few: in logic [BCP16], automata theory [Bar+20b], program analysis [CDZ17], and database theory [Ben+15]. What comes next is an application of the theory of regular cost functions to the field of stochastic control.

Nathalie Bertrand visited the Simons Institute for a month to participate in the semester Logical Structures in Computation (mentioned above), and gave a talk about a recent result they had obtained on the control problem for population protocols [Ber+17] (see also the journal version [Ber+19]). In this model, a population of agents are controlled uniformly, meaning that the controller applies the same action to every agent. The agents are represented by a finite state system, the same for every agent. The key difficulty is that there is an arbitrary large number of agents: the question is whether for every $n \in \mathbb{N}$, there exists a controller able to bring all n agents synchronously to a target state. The technical contribution of [Ber+17; Ber+19] is to prove that in the adversarial

setting where an opponent chooses the evolution of the agents, the (adversarial) control problem is **EXPTIME**-complete.

Nathalie introduced the stochastic variant of this problem, where each agent evolves independently according to a probabilistic distribution, *i.e.* the finite state system modelling an agent is a Markov decision process. The question is whether for every $n \in \mathbb{N}$, there exists a controller able to bring all n agents synchronously to a target state with probability one. (Almost) nothing was known about this problem that we call the stochastic control problem; before stating our results, let us discuss four motivations and related lines for works: control of biological systems, parameterised verification and control, distributed computing, and probabilistic automata.

The original motivation was for controlling population of yeasts [Uhl+15]. In this application, the concentration of some molecule is monitored through fluorescence level. Controlling the frequency and duration of injections of a sorbitol solution influences the concentration of the target molecule, triggering different chemical reactions which can be modelled by a finite state system. The objective is to control the population to reach a predetermined fluorescence state.

The formulation of the problem is rooted in parameterised verification, introduced in [GS92]: it is the verification of a system composed of an arbitrary number of identical components. The control problem introduced in [Ber+17; Ber+19] is the first step towards *parameterised control*: the goal is control a system composed of many identical components in order to ensure a given property. To the best of my knowledge, the contributions of [Ber+17; Ber+19] are the first results on parameterised control.

The model also resembles two models introduced for the study of distributed computing. The first and most widely studied is population protocols, introduced in [Ang+06]: the agents are modelled by finite state systems and interact by pairs drawn at random. The mode of interaction is the key difference with the model we study here: in a time step, all of our agents perform simultaneously and independently the same action. This brings us closer to broadcast protocols as studied for instance in [EFM99], in which one action involves an arbitrary number of agents. As explained in [Ber+17; Ber+19], the model can be seen as a subclass of (stochastic) broadcast protocols. The focus of the distributed computing community when studying population or broadcast protocols is to construct the most efficient protocols for a given task, such as (prominently) electing a leader. A growing literature from the verification community focuses on checking the correctness of a given protocol against a given specification; we refer to the recent survey [Esp16] for an overview. The question here is a synthesis question: we are asking about the existence of a strategy, in other words a protocol, whose goal is to synchronise all agents in a target state.

When considering the limit case of infinitely many agents the parameterised control question becomes the value 1 problem for probabilistic automata, which was proved undecidable in [GO10], and even in very restricted cases [Fij+14]. Hence abstracting continuous distributions by a discrete population of arbitrary size can be seen as an approximation technique for probabilistic automata. Using n agents correponds to using numerical approximation up to 2^{-n} with random rounding; in this sense the control problem considers arbitrarily fine approximations. The plague of undecidability results on probabilistic automata (see *e.g.* [Fij17]) is nicely contrasted by our positive result, which is one of the few decidability results on probabilistic automata not making structural assumptions on the underlying graph.

I started working on the stochastic control for population protocols back in 2016. The main question was whether it is decidable at all. We identified a first tool for attacking it: downward closures made the problem naturally fit into the theory of well quasi orders (see [Sch17] for an introduction to recent results in computer science using this theory). However despite years of efforts, we could not crack it. I became obsessed and presented the problem to a number of researchers, which led to the exploration of many beautiful ideas, but none gave us the solution. The second insight came from Thomas Colcombet, who suggested a reduction of that problem to the theory of regular cost functions. The third insight and key to the full proof is the use of the max-flow min-cut theorem on graphs, which reformulates the question in a dual form that can then be expressed using cost-automata. Our main technical result is that the stochastic control problem is decidable [CFO20].

This is not the end of the story; first because our algorithms is highly impractical, its complexity is a priori non-elementary although the best known lower bound is **EXPTIME**-hardness [MST19]. The second and most interesting perspective was suggested by Blaise Genest (coauthor of [Ber+17; Ber+19]): in addition to requiring to synchronise the n agents with probability one, can we bound the expected time before this happens as a function of n? It turns out that there are three natural regimes: polylogarithmic, polynomial, and exponential (the general case). Very little is known about these, except for personal conjectures I would be happy to share!

Bastien Maubert

I visited Bastien Maubert in Naples in May 2017, this was the start of a fruitful collaboration continuing today. We explored together applications of the theory of regular cost functions to controller synthesis, and more specifically to logical specifications involving bound requirements.

In order to perform strategic reasoning temporal logics of programs (such as LTL) have been extended with operators expressing the existence of strategies for coalitions of components. Among the most successful proposals are Alternating-time Temporal Logic (ATL) [AHK02] and, more recently, the more expressive Strategy Logic (SL) [CHP10]. Both logics can express the existence of strategies for coalitions that ensure some temporal properties against all possible behaviours of the remaining components. Moreover, if such strategies exist, one can also obtain witnessing finite-state strategies. As a result, synthesizing reactive systems from temporal specifications [PR89] can be reduced to model checking such strategic logics. Strategy Logic can express important game-theoretic concepts such as the existence of Nash equilibria, but they are limited to qualitative properties. For instance important properties such as bounding the maximal number of steps between an event and its reaction cannot be expressed; parametric extensions of temporal logics have been introduced to capture such properties.

The eventually operator can be annotated with a bound: $\mathbb{F}^{\leqslant b}\varphi$ says that φ holds within b steps for a constant $b \in \mathbb{N}$. However, one may not know such bounds or care for their exact value when writing the specification, and it may not be practical to compute the bound. Prompt LTL [KPV09] considers b as a variable, and the model-checking problem

asks if there exists a valuation of the variable b such that the formula holds.

In order to reason about and synthesize strategies that ensure such properties specified in Prompt LTL, we introduced Prompt Strategy Logic, an extension of SL with the $\mathbb{F}^{\leqslant b}$ operator, and developed further the theory of regular cost functions to prove the decidability of model-checking for Prompt Strategy Logic [Fij+18].

Assume-guarantee originates as a modular approach to program verification that allows decomposition of proof obligations [AH99]. Informally, an assume-guarantee specification consists of two specifications ϕ and ψ. A system S satisfies this specification if whenever it is used in a context that satisfies the assumption ϕ, the guarantee ψ holds on the system or, said differently, for all environments E satisfying ϕ, the composed system $S||E$ satisfies ψ.

Assume-guarantee synthesis is the problem of synthesising a system S satisfying an assume-guarantee specification. We refer to [MS12] for a discussion of practical applications of assume-guarantee synthesis. When considering LTL specifications, assume-guarantee verification and synthesis reduce to classical LTL model checking and synthesis: if ϕ and ψ are LTL formulas, then the assume-guarantee specification is equivalent to the LTL specification $\phi \implies \psi$. But this is not true for Prompt LTL: when we ask that in all environments satisfying the assumption the system also satisfies the guarantee, there is an implicit quantification on the bounds with which the assumption and the guarantee are satisfied: a universal one for the assumption and an existential one for the guarantee. Because in Prompt LTL the only quantification on bounds is an existential one at the front of the formula, this alternation is not reflected in the formula $\phi \implies \psi$ and, in fact, cannot be captured by a Prompt LTL formula. Classical techniques to handle Prompt LTL such as the alternating colour technique [KPV09] are thus difficult to apply, and the assume-guarantee synthesis problem for Prompt LTL has been open for a decade.

The problem of assume-guarantee synthesis for Prompt LTL as defined above can be called *uniform*, as the system that one aims at synthesising does not depend on the bounds for which the assumption or the guarantee are satisfied. It was observed in [JTZ18] that this variant of the problem does not always admit finite implementations, as a satisfying system may require memory that depends on the bounds. Also, in the formulation of the problem they consider, the assumption talks about both inputs *and outputs*, which makes it possible to have solution systems that always falsify the assumption. To eliminate such unsatisfactory solutions we consider assumptions that only talk about inputs. Also, to account for the fact that a system's memory may depend on the bound for the assumption, we introduce a *non-uniform* variant of the problem, which asks whether for every bound b on the assumption, there exists a bound b' on the guarantee and a system S_b such that whenever the assumption is satisfied with bound b, the guarantee is satisfied with bound b'.

Our second paper constructs an algorithm for solving the assume-guarantee synthesis problem for Prompt LTL [Fij+20b], solving this decades old open problem. Time and again the heavylifting is done within the theory of regular cost functions: the solution makes crucial use of new results about (history)-determinisability of cost-automata and solvability of cost-games.

Guillaume Lagarde

I met Guillaume Lagarde in 2017 on Friday March 3 at 2:30PM in the room 3052 of the IRIF (Paris 7) laboratory when he gave a talk about his recent works proving lower bounds for non-commutative circuits. He and coauthors extended a famous result by Nisan [Nis91] proving an exponential lower on non-commutative algebraic branching programs, by constructing from a circuit a matrix and analysing its rank. At this time I was becoming familiar with automata learning, and in particular the fascinating notion of Hankel matrices. I saw a connection between the following two results, which I thought could not be a coincidence:

- Nisan's result can be stated as follows: given a multivariate non-commutative polynomial P, we can construct a matrix N_P such that any algebraic branching program computing P has size at least the rank of N_P;

- Fliess's result [Fli74] can be stated as follows: given a function $f : \Sigma^* \to \mathbb{R}$, we can construct a matrix H_f called the Hankel matrix of f such that the size of the smallest weighted automaton computing f is exactly the rank of H_f.

Fliess' result, which had been restated and rediscovered many times over the years, is a fundamental result about weighted automata, it is the key ingredient for minimising, and later for learning, these models. It took some time to find a perfect one-to-one correspondence between definitions and notions in non-commutative algebraic complexity and automata theory; once this settled we understood that Nisan's result can indeed be obtained as a simple corollary of Fliess' result. More interestingly, this implies that Nisan's approach is not only a lower bound, but also an upper bound: the size of the smallest algebraic branching program computing P is exactly the rank of N_P. In other words, algebraic branching programs are characterised by the matrix defined by Nisan.

Going further, we wondered whether Guillaume's results [LLS18] (extending Nisan's) also had a weighted automata counterpart. The answer is positive, and for this we needed to work with weighted automata over finite trees. Indeed Fliess' result can be extended (almost mutatis mutandis) to functions mapping trees to real numbers: this is the work of Bozapalidis and Louscou-Bozapalidou [BL83]. Similarly as for the special case of algebraic branching programs, the lower bound induced by the rank of the Hankel matrix is actually also an upper bound. We were then in possession of a new tool to analyse non-commutative circuits: this was an invitation to revisit the literature. The resulting paper [Fij+20a] (and the journal version [Fij+21]) draws on this newly discovered connection to give new and unified proofs of many existing results, but most importantly prove new results by decomposing and analysing the Hankel matrix.

It turns out that the notion of Hankel matrix played another fundamental role in a line of work I contributed to: lower bounds for state complexity. Motivated by a question asked by Rabin in his seminal paper [Rab63], I investigated the succinctness of probabilistic automata [Fij16]. This led to the introduction of (what I thought to be) a new method for proving lower bounds on the state complexity of alternating automata [Fij18b]; the matrix constructed in this lower bound is exactly the Hankel matrix, and the lower bounds

method stated in [Fij18b] is a boolean counterpart of Fliess' theorem. The journal version clarifies this connection [Fij20a].

Pierre Ohlmann

I met Pierre Ohlmann when he joined Oxford for a summer internship in 2016. As Ben accurately once said: 'Don't get fooled: he looks like a Californian surfer but he's really sharp'. I co-supervised Pierre for his next two internships (in London and in Paris), and then co-supervised him with Olivier Serre for his PhD. We worked together on many topics, but I can proudly say that I most successfully infected him with the virus of parity games, and more generally games of infinite duration.

Games of infinite duration are a widely studied model in several fields of computer science including program verification, model checking, automata theory, logic, finite model theory, and database theory: the interactions between the players can model a range of (non-terminating) systems and are therefore used for analysing, verifying, and synthesising them. There exists a variety of game models: we consider here two player zero sum deterministic (as opposed to stochastic) games with perfect information. Two of the most important open problems in this field concern parity games and mean payoff games: in both cases the complexity of solving them is in **NP** and in **coNP**, but not known to be in polynomial time. This complexity status was once shared with primality testing, linear programming, and others famous problems, which all turned out to the solvable in polynomial time. Yet although both problems have attracted a lot of attention over the past three decades, they remain widely open and exciting research directions.

Parity games are a central model in the study of logic and automata over infinite trees and for their tight relationship with model-checking games for the modal μ-calculus. Following decades of exponential and subexponential algorithms, a breakthrough happened in 2017 when Calude, Jain, Khoussainov, Li, and Stephan [Cal+17] constructed a quasipolynomial time algorithm for solving parity games. A shockwave was felt throughout Europe: Gimbert and Ibsen-Jensen published a shorter correctness proof [GI17], Jurdziński and Lazić constructed a second quasipolynomial time algorithm [JL17], Fearnley, Jain, Schewe, Stephan, and Wojtczak reformulated the original algorithm using value iteration [Fea+17], and Bojańczyk and Czerwiński presented the algorithm as the construction of a separating automaton [BC18], all of this within a matter of months. A few months after that Lehtinen constructed a third quasipolynomial time algorithm [Leh18].

Bojańczyk's and Czerwiński's contribution [BC18] went beyond merely explaining the first algorithm: by introducing the notion of separating automata they defined a new tool for constructing and understanding algorithms; and indeed it was soon understood that all three quasipolynomial time algorithms induced separating automata. Informally, a separating automaton is a safety automaton – meaning it accepts all infinite runs and only rejects on deadlocks – which separates positional winning plays from losing ones: the observation of Bojańczyk and Czerwiński is that such an automaton induces a generic algorithm for solving parity games by reduction to safety games. I visited Jurdziński and Lazić in Warwick and Bojańczyk and Czerwiński in Warsaw around this time, and the

question on everyone's lips was: can we construct smaller separating automata?

Diving into Jurdziński's and Lazić's succinct progress measure algorithm I extracted the notion of universal trees [Fij18a]: I constructed a family of value iteration algorithms parameterised by the choice of a universal tree, of which the succinct progress measure algorithm is a special case. The main technical result in [Fij18a] is a quasipolynomial lower bound on the size of universal trees, a purely combinatorial statement. The lower bounds offered in [Fij18a] match up to a polynomial factor the universal tree (implicitly) constructed in [JL17].

One month later Czerwiński, Daviaud, Jurdziński, Lazić, and Parys showed that any separating automaton induces a universal tree of the same size. Combined with my lower bounds on universal trees, this implies an unconditional quasipolynomial lower bound on the size of separating automata. Our joint paper [Cze+19] conveys a simple message: all existing quasipolynomial time algorithms are based on separating automata, hence cannot break the quasipolynomial barrier of universal trees.

Parys constructed yet another quasipolynomial time algorithm [Par19] which is both simple and beautiful, and seemed to escape the limitations of separating automata. The algorithm is inspired by the exponential time algorithm of Zielonka [Zie98] specialising an algorithm due to McNaughton [McN93]. Parys' algorithm was later improved by Lehtinen, Schewe, and Wojtczak [LSW19]. Recently Jurdziński, Morvan, Ohlmann, and Thejaswini [Jur+20] constructed a generic 'universal attractor decomposition algorithm' encompassing all three algorithms. and parameterised by the choice of two universal trees (one of each player). Choosing appropriate pairs of universal trees yields the algorithms from [Zie98; Par19; LSW19], hence these algorithms do not escape the fate of universal trees and their quasipolynomial lower bounds.

In summary, all existing quasipolynomial time algorithms for parity games (and some exponential ones) are related to the combinatorial notion of universal trees, hence subjected to the quasipolynomial lower bounds. If there exists a polynomial time algorithm, we should look for it elsewhere!

Beyond parity games. There are a number of other objectives of interest, the most prominent one being mean payoff. With minor adjustements the notion of separating automata can be extended to any positionally determined objective, begging the question whether they can be useful beyond parity games. Trying to extend universal trees I defined universal graphs and told Thomas Colcombet and Pierre Ohlmann about this new definition. Thomas came back the next day with a fantastic technical idea: saturation. This implied a direct and simple proof of the equivalence between separating automata and universal graphs, extending the results of [Cze+19] from parity to any positionally determined objective. The paper [CF18; CF19] states this equivalence also considering non-deterministic separating automata and the semantically restricted notion of good-for-small-games automata. The equivalence proof implies a normal form for universal graphs called linear universal graphs, which means that they yield generic value iteration algorithms (and in particular can be implemented with quasilinear space complexity). At this point we had in our hands two equivalent tools: separating automata and universal graphs. Seeing two sides of the same coin is often useful, and in the subsequent developments going from one representation to the other was sometimes essential.

To summarise, what are universal graphs good for? Given a class of objectives, they

reduce the question of constructing algorithms for solving games to constructing universal graphs. This is considerably simpler as it is not anymore about the interaction of the two players but only about the combinatorial properties of the objectives. We followed this recipe for different classes of objectives:

- For mean payoff objectives [FGO20], we proved matching upper and lower bounds on the size of universal graphs: the upper bounds induce algorithms for solving mean payoff games matching the best known deterministic complexity, and the lower bounds imply that these algorithms are optimal within this class of algorithms. The main message here is that universal graphs do not yield quasipolynomial time algorithms for mean payoff games.

- For disjunctions of mean payoff objectives, and for disjunctions of mean payoff and parity objectives, we investigated how to combine existing universal graphs for each objective into universal graphs for their disjunctions. Rather than constructing universal graphs from scratch, we wanted to define constructions using universal graphs for the atomic objectives as black boxes. In other words, we assumed the existence of universal graphs for parity objectives and for mean payoff objectives, and constructed universal graphs for combinations of these classes. An important benefit of this approach is its simplicity: both constructions and their correctness proofs are rather short and focus on the interactions between the objectives.

The paper [Fij20b] is an introduction to universal graphs, and the (submitted) journal paper [Col+21] is a comprehensive account of all existing results on the nascent theory of universal graphs.

Adrià Gascón and Brooks Paige

Adrià Gascón, Brooks Paige, and myself, were research fellows at the Alan Turing Institute in London at the same time, around 2017. Adrià's background is in security and machine learning and Brooks' in probabilistic programming; as a way to collaborate we offered a summer internship at the Turing, which is aimed at PhD students. We hired Judith Clymo, who was specialising in the theory of QBF solvers, and Haik Manukian, an astrophysicist with a solid background in machine learning. This mix of five backgrounds made for a very productive summer! The goal was to work on programming by example, a program synthesis framework where the program is constructed from a few examples, and in particular study the recent 'DeepCoder' approach [Bal+17] guiding the search of programs using machine learning predictions.

Working on a re-implementation of DeepCoder's approach, we identified an important challenge: machine learning means training data, and in this context it is essentially non-existent. DeepCoder's results were based on synthetic data, posing the following question: how to create a dataset of millions of programs, and for each programs of relevant and interesting inputs? Our paper [Cly+20] introduces different approaches to data generation for program synthesis leveraging tools from formal methods (SMT solvers), and shows the influence of this question in the overall solution.

Besides the question of data generation, this summer was a very exciting dive into a fascinating topic: machine learning guided program synthesis. Little was known on the theoretical side, yet its practice was booming with dozens of papers published in the machine learning community annually. The following year I was awarded a three-year research grant DeepSynth (CNRS Momentum 2018 – 2021) to work on the topic and to hire a post-doc for two years. Guillaume Lagarde joined LaBRI in September 2019 to carry out this ambitious project. Our aim was to lay the theoretical foundations for machine learning guided program synthesis and to relate this question to existing approaches in computer science; see [FL20] for our tutorial.

We focused on the search aspect and it took us the whole duration of the project to define a satisfactory framework in which we could formulate existing algorithms, quantify their merits, and construct new ones. The abstract question we ask is the following: we are looking for a program in a set of programs. We do not know what the program is but we do have some predictions in the form of a probabilistic distribution $\mathcal{D}$ over the set of programs. The question is: how to organise the search? The most natural answer is to enumerate programs in non-increasing order: $x_1, x_2, \ldots$ such that $\mathcal{D}(x_1) \geqslant \mathcal{D}(x_2) \geqslant \ldots$, meaning x_1 is the most likely program according to $\mathcal{D}$, x_2 the second most likely, and so on. Unfortunately, enumerating programs in this exact order may be computationally expensive if the distribution $\mathcal{D}$ is complicated. This reveals a trade-off: should we enumerate many programs, albeit with low likelihood, or should we enumerate a lot of very likely programs? Another approach is sampling: choose a distribution $\mathcal{D}'$ and sample programs from it. Surprisingly enough, the optimal choice of distribution is not $\mathcal{D}' = \mathcal{D}$: indeed to account for the possible sample repetitions it is beneficial to skew $\mathcal{D}$ into what we call the square root of $\mathcal{D}$. The point of the distribution based search framework we introduced is to formalise the algorithmic challenges we just sketched.

We implemented a general purpose program synthesis tool called DeepSynth [Fij+22]:

`https://github.com/nathanael-fijalkow/DeepSynth`

It serves as a basis for all our experiments and shows the improvements due to the distribution based search. Thanks to the genericity of the codebase we are exploring applications of DeepSynth to three different problems: reinforcement learning, the Abstraction and Reasoning Corpus (ARC), and synthesis of reactive controllers.

Ritam Raha

I met Ritam Raha when he joined LaBRI as an intern in Summer 2018. He started his PhD in September 2019 under the joint supervision of Guillermo Perez and Floris Geerts from the University of Antwerp (Belgium), and Jérôme Leroux (Bordeaux) and myself. The goal of Ritam's PhD is to construct algorithms for the verification of machine learning models. We have worked together on different aspects of automata learning in a passive scenario (as opposed to the celebrated active automata learning due to Angluin [Ang87]).

Passive learning of languages has a long history paved with negative results. Learning automata is notoriously difficult from a theoretical perspective, as witnessed by the original **NP**-hardness result of learning a deterministic finite automaton (DFA) from examples

by [Gol78]. This line of hardness results culminates with the inapproximability result of [PW93] stating that there is no polynomial time algorithm for learning a DFA from examples even up to a polynomial approximation of their size.

One approach to cope with such hardness results is to change representation, replacing automata by logical formulas: their syntactic structures make them more amenable to principled search algorithms. There is a range of potential logical formalisms to choose from depending on the application domain. Linear Temporal Logic [Pnu77] is a prominent logic for specifying temporal properties over words. It has become a de facto standard in many fields such as model checking, program analysis, and motion planning for robotics. A key property making LTL a strong candidate as a concept class is that its syntax does not include variables, contributing to the fact that LTL formulas are typically easy to interpret and therefore useful as explanations.

Over the past five to ten years learning temporal logics (of which LTL is the core) has become an active research area, with applications in program specification and anomaly and fault detections. A number of different approaches have been proposed, leveraging SAT solvers, automata, and Bayesian inference, and extended to more expressive logics such as Property Specification Language (PSL) and Computational Tree Logic (CTL).

Nothing was known about the computational complexity of the underlying problem; indeed the works cited above focused on constructing efficient algorithms for practical applications. In [FL21] we initiated the study of the complexity of learning LTL formulas from examples both for the exact and the approximation settings, and relating the problem to classical problems algorithmic questions on finite strings.

Together with Ritam and coauthors [Rah+22] we have used these theoretical insights to construct a new tool performing LTL learning from examples. Thanks to succinct representations and the use of approximation algorithms, our algorithm scales to formulas of size 100, while all previous approaches could not output formulas of size larger than 10. The main appeal of our algorithm is to be anytime: whereas previous algorithms may reach the timeout and not give any output, our algorithm constructs smaller and smaller formulas along the computation, hence in most cases can output some formula even if it is not minimal.

Antonio Casares

I met Antonio Casares in Spring 2020 for an (online...) internship, which was immediately followed by the start of his PhD under the joint supervision of Thomas Colcombet (Paris), Igor Walukiewicz (Bordeaux), and myself. Antonio's research interests are in automata, logic, and games, and their applications to synthesis of reactive controllers specified in Linear Temporal Logic (LTL).

The original approach of Pnueli and Rosner [PR89] for LTL synthesis using automata and games devised more than four decades ago is today at the heart of the state of the art synthesis tools [Esp+17; LMS20; MC18]. The bottleneck is the determinisation of Büchi automata: given a non-deterministic Büchi automaton, construct an equivalent parity automaton. This problem has a long history; it was originally solved by McNaughton in the 60s, and the first asymptotically optimal construction is due to Safra in the 80s. Most of

the recent theoretical and practical solutions of this problem are based on the construction of Piterman [Pit06]. Schewe's [Sch09] enlightening perspective on this construction is to decompose it into two steps: first construct a deterministic Muller automaton, and then transform it into an equivalent deterministic parity automaton. Piterman and Schewe's determinisation procedure is one of many examples of constructions using as an intermediate step (subclasses of) Muller conditions before transforming them into parity conditions, either working with automata models or games models.

During the first year of Antonio's PhD we focused on this particular step and study transformations from Muller to parity. We worked with general transition systems to seamlessly encompass both automata and games models. There are several existing constructions transforming subclasses of Muller conditions to parity. The first is the Latest Appearance Record (LAR), which applies to all Muller conditions. It was proved to be optimal in the worst case: *there exists* a family of Muller automata for which the obtained parity automata are minimal. Many refinements of the LAR have been constructed for subclasses of Muller conditions.

The notion of Zielonka tree of a Muller condition was introduced in [Zie98] and shown to capture the exact memory requirements of Muller games [DJW97]. In the long version of [DJW97], it implicitly appears that the Zielonka tree of a Muller condition can be used to construct a parity automaton recognising this Muller condition. Our first observation was to show a *strong* optimality result: *for all* Muller conditions, the parity automaton obtained from the Zielonka tree of a Muller condition is minimal both in the number of states and in the number of priorities. This optimality result is much stronger than the worst case optimality result of the LAR transformation; in essence, it shows that the Zielonka tree of a Muller condition precisely captures the properties of the Muller condition, whereas for instance the LAR only depends on the number of colours.

Our second insight was to note that all existing constructions, including the one based on Zielonka trees, only consider the Muller condition but do not take into account the structure of the underlying transition system. In other words, all transformations work at the level of conditions: they transform a Muller condition into a parity condition, and ignore the interplay between the condition and the transition structure.

In our first paper [CCF21] we constructed a new transformation called the alternating cycle decomposition (ACD) which captures this interplay: the ACD transforms a Muller transition system into a parity transition system, extending Zielonka trees by considering the alternation of accepting and rejecting cycles in the original transition system. We proved a strong optimality result for the ACD transformation, which had a number of important consequences. The first is an improvement of Piterman and Schewe's construction, and the second is a set of crisp characterisations for relabelling transition systems with different classes of acceptance conditions. With these new insights, Antonio [Cas21] related an automata minimisation question to memory requirements for Muller games, solving a long standing open question of Kopczyński [Kop09]. On the more practical side, the ACD transformation has been implemented by Antonio and coauthors in the state of the art tool Strix, yielding smaller controllers for many benchmarks.

Bibliography

Personal references

[Bar+20a] Corentin Barloy, Nathanaël Fijalkow, Nathan Lhote, and Filip Mazowiecki. "A Robust Class of Linear Recurrence Sequences". In: *Computer Science in Logic, CSL*. 2020. DOI: 10.4230/LIPIcs.CSL.2020.9 (cited on p. 2).

[Ber+21] Raphaël Berthon, Nathanaël Fijalkow, Emmanuel Filiot, Shibashis Guha, Bastien Maubert, Aniello Murano, Laureline Pinault, Sophie Pinchinat, Sasha Rubin, and Olivier Serre. "Alternating Tree Automata with Qualitative Semantics". In: *ACM Transactions on Computational Logic* 22.1 (2021). DOI: 10.1145/3431860 (cited on pp. 4, 5).

[CCF21] Antonio Casares, Thomas Colcombet, and Nathanaël Fijalkow. "Optimal Transformations of Games and Automata Using Muller Conditions". In: *International Colloquium on Automata, Languages, and Programming, ICALP*. 2021. DOI: 10.4230/LIPIcs.ICALP.2021.123 (cited on p. 15).

[CF20] Alexander Clark and Nathanaël Fijalkow. "Consistent Unsupervised Estimators for Anchored PCFGs". In: *Transactions of the Association for Computational Linguistics* 8 (2020). DOI: 10.1162/tacl_a_00323 (cited on p. 4).

[Cle+19] Florence Clerc, Nathanaël Fijalkow, Bartek Klin, and Prakash Panangaden. "Expressiveness of probabilistic modal logics: A gradual approach". In: *Information and Computation* 267 (2019). DOI: 10.1016/j.ic.2019.04.002 (cited on p. 3).

[Cly+20] Judith Clymo, Haik Manukian, Nathanaël Fijalkow, Adrià Gascón, and Brooks Paige. "Data Generation for Neural Programming by Example". In: *International Conference on Artificial Intelligence and Statistics, AI&STATS*. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020. URL: http://proceedings.mlr.press/v108/clymo20a.html (cited on p. 12).

[CF18] Thomas Colcombet and Nathanaël Fijalkow. "Parity games and universal graphs". In: *CoRR* (2018). URL: https://arxiv.org/abs/1810.05106 (cited on p. 11).

[CF19] Thomas Colcombet and Nathanaël Fijalkow. "Universal Graphs and Good for Games Automata: New Tools for Infinite Duration Games". In: *Foundations of Software Science and Computation Structures, FoSSaCS*. 2019. DOI: 10.1007/978-3-030-17127-8_1 (cited on p. 11).

[Col+21] Thomas Colcombet, Nathanaël Fijalkow, Paweł Gawrychowski, and Pierre Ohlmann. "The Theory of Universal Graphs for Infinite Duration Games". In: *CoRR* (2021). URL: https://arxiv.org/abs/2104.05262 (cited on p. 12).

[CFO20] Thomas Colcombet, Nathanaël Fijalkow, and Pierre Ohlmann. "Controlling a Random Population". In: *Foundations of Software Science and Computation Structures, FoSSaCS*. 2020. DOI: 10.1007/978-3-030-45231-5_7 (cited on p. 7).

[Cze+19] Wojciech Czerwiński, Laure Daviaud, Nathanaël Fijalkow, Marcin Jurdziński, Ranko Lazić, and Paweł Parys. "Universal trees grow inside separating automata: Quasi-polynomial lower bounds for parity games". In: *International Symposium on Discrete Algorithms, SODA*. 2019. DOI: 10.1137/1.9781611975482.142 (cited on p. 11).

[Fij18a] Nathanaël Fijalkow. "An Optimal Value Iteration Algorithm for Parity Games". In: *CoRR* (2018). URL: https://arxiv.org/abs/1801.09618 (cited on p. 11).

[Fij20a] Nathanaël Fijalkow. "Lower bounds for the state complexity of probabilistic languages and the language of prime numbers". In: *The Journal of Logic and Computation* 30.1 (2020). DOI: 10.1093/logcom/exaa007 (cited on p. 10).

[Fij16] Nathanaël Fijalkow. "Online Space Complexity of Probabilistic Automata". In: *Logical Foundations of Computer Science, LFCS*. 2016. DOI: 10.1007/978-3-319-27683-0_8 (cited on p. 9).

[Fij18b] Nathanaël Fijalkow. "The State Complexity of Alternating Automata". In: *Logic in Computer Science, LICS*. 2018. DOI: 10.1145/3209108.3209167 (cited on pp. 9, 10).

[Fij20b] Nathanaël Fijalkow. "The Theory of Universal Graphs for Games: Past and Future". In: *Coalgebraic Methods in Computer Science, CMCS*. Vol. 12094. 2020. DOI: 10.1007/978-3-030-57201-3_1 (cited on p. 12).

[Fij17] Nathanaël Fijalkow. "Undecidability results for probabilistic automata". In: *SIGLOG News* 4.4 (2017). DOI: 10.1145/3157831.3157833 (cited on p. 6).

[FGO20] Nathanaël Fijalkow, Paweł Gawrychowski, and Pierre Ohlmann. "Value Iteration Using Universal Graphs and the Complexity of Mean Payoff Games". In: *Mathematical Foundations of Computer Science, MFCS*. 2020. DOI: 10.4230/LIPIcs.MFCS.2020.34 (cited on p. 12).

[Fij+14] Nathanaël Fijalkow, Hugo Gimbert, Florian Horn, and Youssouf Oualhadj. "Two Recursively Inseparable Problems for Probabilistic Automata". In: *Mathematical Foundations of Computer Science, MFCS*. 2014. DOI: 10.1007/978-3-662-44522-8_23 (cited on p. 6).

[Fij+15] Nathanaël Fijalkow, Florian Horn, Denis Kuperberg, and Michał Skrzypczak. "Trading Bounds for Memory in Games with Counters". In: *International Colloquium on Automata, Languages, and Programming, ICALP*. 2015. DOI: 10.1007/978-3-662-47666-6_16 (cited on p. 4).

[FKS20] Nathanaël Fijalkow, Stefan Kiefer, and Mahsa Shirmohammadi. "Trace Refinement in Labelled Markov Decision Processes". In: *Logical Methods in Computer Science* 16.2 (2020). DOI: 10.23638/LMCS-16(2:10)2020 (cited on p. 3).

[FKP17] Nathanaël Fijalkow, Bartek Klin, and Prakash Panangaden. "Expressiveness of Probabilistic Modal Logics, Revisited". In: *International Colloquium on Automata, Languages, and Programming, ICALP*. 2017. DOI: 10.4230/LIPIcs.ICALP.2017.105 (cited on p. 3).

[FL20] Nathanaël Fijalkow and Guillaume Lagarde. "A tutorial on machine learning Guided program synthesis". In: *European Conference on Artificial Intelligence, ECAI*. 2020. URL: https://deepsynth.labri.fr/?page=tutorial (cited on p. 13).

[FL21] Nathanaël Fijalkow and Guillaume Lagarde. "The Complexity of Learning Linear Temporal Formulas from Examples". In: *International Conference on Grammatical Inference, ICGI*. 2021 (cited on p. 14).

[Fij+22] Nathanaël Fijalkow, Guillaume Lagarde, Théo Matricon, Kevin E. Ellis, Pierre Ohlmann, and Akarsh Potta. "DeepSynth: Scaling Neural Program Synthesis with Distribution-based Search". In: *AAAI Conference on Artificial Intelligence, AAAI*. 2022. URL: https://arxiv.org/abs/2110.12485 (cited on p. 13).

[Fij+20a] Nathanaël Fijalkow, Guillaume Lagarde, Pierre Ohlmann, and Olivier Serre. "Lower Bounds for Arithmetic Circuits via the Hankel Matrix". In: *Symposium on Theoretical Aspects of Computer Science, STACS*. 2020. DOI: 10.4230/LIPIcs.STACS.2020.24 (cited on p. 9).

[Fij+21] Nathanaël Fijalkow, Guillaume Lagarde, Pierre Ohlmann, and Olivier Serre. "Lower Bounds for Arithmetic Circuits via the Hankel Matrix". In: *Computational Complexity* (2021) (cited on p. 9).

[Fij+19a] Nathanaël Fijalkow, Engel Lefaucheux, Pierre Ohlmann, Joël Ouaknine, Amaury Pouly, and James Worrell. "On the Monniaux Problem in Abstract Interpretation". In: *International Symposium on Static Analysis, SAS*. 2019. DOI: 10.1007/978-3-030-32304-2_9 (cited on p. 2).

[Fij+18] Nathanaël Fijalkow, Bastien Maubert, Aniello Murano, and Sasha Rubin. "Quantifying Bounds in Strategy Logic". In: *Computer Science in Logic, CSL*. 2018. DOI: 10.4230/LIPIcs.CSL.2018.23 (cited on p. 8).

[Fij+20b] Nathanaël Fijalkow, Bastien Maubert, Aniello Murano, and Moshe Y. Vardi. "Assume-Guarantee Synthesis for Prompt Linear Temporal Logic". In: *International Joint Conference on Artificial Intelligence, IJCAI*. 2020. DOI: 10.24963/ijcai.2020/17 (cited on p. 8).

[Fij+19b] Nathanaël Fijalkow, Pierre Ohlmann, Joël Ouaknine, Amaury Pouly, and James Worrell. "Complete Semialgebraic Invariant Synthesis for the Kannan-Lipton Orbit Problem". In: *Theory of Computing Systems* 63.5 (2019). DOI: 10.1007/s00224-019-09913-3 (cited on p. 2).

[Fij+17] Nathanaël Fijalkow, Pierre Ohlmann, Joël Ouaknine, Amaury Pouly, and James Worrell. "Semialgebraic Invariant Synthesis for the Kannan-Lipton Orbit Problem". In: *Symposium on Theoretical Aspects of Computer Science, STACS*. 2017. DOI: 10.4230/LIPIcs.STACS.2017.29 (cited on p. 2).

[Fij+19c] Nathanaël Fijalkow, Joël Ouaknine, Amaury Pouly, João Sousa Pinto, and James Worrell. "On the decidability of reachability in linear time-invariant systems". In: *International Conference on Hybrid Systems: Computation and Control, HSCC*. 2019. DOI: 10.1145/3302504.3311796 (cited on p. 2).

[FPS13] Nathanaël Fijalkow, Sophie Pinchinat, and Olivier Serre. "Emptiness Of Alternating Tree Automata Using Games With Imperfect Information". In: *Foundations of Software Technology and Theoretical Computer Science, FSTTCS*. 2013. DOI: 10.4230/LIPIcs.FSTTCS.2013.299 (cited on p. 4).

[FRW17] Nathanaël Fijalkow, Cristian Riveros, and James Worrell. "Probabilistic Automata of Bounded Ambiguity". In: *International Conference on Concurrency Theory, CONCUR*. 2017. DOI: 10.4230/LIPIcs.CONCUR.2017.19 (cited on p. 5).

[FRW20] Nathanaël Fijalkow, Cristian Riveros, and James Worrell. "Probabilistic Automata of Bounded Ambiguity". In: *Information and Computation* (2020). DOI: https://doi.org/10.1016/j.ic.2020.104648 (cited on p. 5).

[Ped+18] Mathias Ruggaard Pedersen, Nathanaël Fijalkow, Giorgio Bacci, Kim G. Larsen, and Radu Mardare. "Timed Comparisons of Semi-Markov Processes". In: *International Conference on Language and Automata Theory and Applications, LATA*. 2018. DOI: 10.1007/978-3-319-77313-1_21 (cited on p. 3).

[Rah+22] Ritam Raha, Roy Rajarshi, Nathanaël Fijalkow, and Daniel Neider. "Scalable Anytime Algorithms for Learning Formulas in Linear Temporal Logic". In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS*. 2022. URL: https://arxiv.org/abs/2110.06726 (cited on p. 14).

Other references

[AH99] Rajeev Alur and Thomas A Henzinger. "Reactive modules". In: *Formal methods in system design* 15.1 (1999). DOI: 10.1023/A:1008739929481 (cited on p. 8).

[AHK02] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. "Alternating-time temporal logic". In: *Journal of the ACM* 49.5 (2002), pp. 672–713. DOI: 10.1145/585265.585270 (cited on p. 7).

[Ang87] Dana Angluin. "Learning Regular Sets from Queries and Counterexamples". In: *Information and Computation* 75.2 (1987). DOI: 10.1016/0890-5401(87)90052-6 (cited on p. 13).

[Ang+06] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. "Computation in networks of passively mobile finite-state sensors". In: *Distributed Computing* 18.4 (2006). DOI: 10.1007/s00446-005-0138-3 (cited on p. 6).

[Aro+16] Sanjeev Arora, Rong Ge, Ravi Kannan, and Ankur Moitra. "Computing a Nonnegative Matrix Factorization - Provably". In: *SIAM Journal of Computing* 45.4 (2016). DOI: 10.1137/130913869 (cited on p. 4).

[Bal+17] M. Balog, A. L. Gaunt, M. Brockschmidt, S. Nowozin, and D. Tarlow. "DeepCoder: Learning to Write Programs". In: *International Conference on Learning Representations, ICLR*. 2017. URL: https://openreview.net/forum?id=ByldLrqlx (cited on p. 12).

[Bar+20b] David Barozzini, Lorenzo Clemente, Thomas Colcombet, and Paweł Parys. "Cost Automata, Safe Schemes, and Downward Closures". In: *International Colloquium on Automata, Languages, and Programming, ICALP*. Vol. 168. LIPIcs. 2020. DOI: 10.4230/LIPIcs.ICALP.2020.109 (cited on p. 5).

[Ben+15] Michael Benedikt, Balder ten Cate, Thomas Colcombet, and Michael Vanden Boom. "The Complexity of Boundedness for Guarded Logics". In: *Logic in Computer Science, LICS*. 2015. DOI: 10.1109/LICS.2015.36 (cited on p. 5).

[Ber+17] Nathalie Bertrand, Miheer Dewaskar, Blaise Genest, and Hugo Gimbert. "Controlling a Population". In: *International Conference on Concurrency Theory, CONCUR*. 2017, 12:1–12:16. DOI: 10.4230/LIPIcs.CONCUR.2017.12 (cited on pp. 5–7).

[Ber+19] Nathalie Bertrand, Miheer Dewaskar, Blaise Genest, Hugo Gimbert, and Adwait Amit Godbole. "Controlling a population". In: *Logical Methods in Computer Science* 15.3 (2019). URL: https://lmcs.episciences.org/5647 (cited on pp. 5–7).

[BCP16] Achim Blumensath, Thomas Colcombet, and Paweł Parys. "On a Fragment of AMSO and Tiling Systems". In: *Symposium on Theoretical Aspects of Computer Science, STACS*. Vol. 47. 2016. DOI: 10.4230/LIPIcs.STACS.2016.19 (cited on p. 5).

[Blu+97] Richard Blute, Josée Desharnais, Abbas Edalat, and Prakash Panangaden. "Bisimulation for Labelled Markov Processes". In: *Logic in Computer Science, LICS*. 1997. DOI: 10.1006/inco.2001.2962 (cited on p. 3).

[Boj16] Mikołaj Bojańczyk. "Thin MSO with a Probabilistic Path Quantifier". In: *International Colloquium on Automata, Languages, and Programming, ICALP*. Vol. 55. LIPIcs. 2016. DOI: 10.4230/LIPIcs.ICALP.2016.96 (cited on p. 4).

[BC18] Mikołaj Bojańczyk and Wojciech Czerwiński. *An Automata Toolbox*. https://www.mimuw.edu.pl/~bojan/papers/toolbox-reduced-feb6.pdf. 2018 (cited on p. 10).

[BKS19] Mikołaj Bojańczyk, Edon Kelmendi, and Michał Skrzypczak. "MSO+∇ is undecidable". In: *Logic in Computer Science, LICS*. 2019. DOI: 10.1109/LICS.2019.8785892 (cited on p. 4).

[Boj+20] Mikołaj Bojańczyk, Edon Kelmendi, Rafał Stefański, and Georg Zetzsche. "Extensions of ω-Regular Languages". In: *Logic in Computer Science, LICS*. 2020. DOI: 10.1145/3373718.3394779 (cited on p. 5).

[BPT16] Mikołaj Bojańczyk, Paweł Parys, and Szymon Toruńczyk. "The MSO+U Theory of (N, <) Is Undecidable". In: *Symposium on Theoretical Aspects of Computer Science, STACS*. Vol. 47. LIPIcs. 2016. DOI: 10.4230/LIPIcs.STACS.2016.21 (cited on p. 4).

[BL83] Symeon Bozapalidis and Olympia Louscou-Bozapalidou. "The Rank of a Formal Tree Power Series". In: *Theoretical Computer Science* 27 (1983), pp. 211–215. DOI: 10.1016/0304-3975(83)90100-7 (cited on p. 9).

[Cal+17] Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. "Deciding parity games in quasipolynomial time". In: *Symposium on Theory of Computing, STOC*. 2017, pp. 252–263. DOI: 10.1145/3055399.3055409 (cited on p. 10).

[CHS14] Arnaud Carayol, Axel Haddad, and Olivier Serre. "Randomization in Automata on Infinite Trees". In: *ACM Transactions on Computational Logic* 15.3 (2014). DOI: 10.1145/2629336 (cited on p. 4).

[Cas21] Antonio Casares. "On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions". In: *CoRR* (2021). URL: https://arxiv.org/abs/2105.12009 (cited on p. 15).

[CHP10] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. "Strategy logic". In: *Information and Computation* 208.6 (2010). DOI: 10.1016/j.ic.2009.07.004 (cited on p. 7).

[Col13a] Thomas Colcombet. "Fonctions Régulières de Coût". Habilitation à diriger des recherches. Université Paris 7, 2013 (cited on pp. 4, 5).

[Col13b] Thomas Colcombet. "Regular Cost Functions, Part I: Logic and Algebra over Words". In: *Logical Methods in Computer Science* 9.3 (2013). DOI: 10.2168/LMCS-9(3:3)2013 (cited on p. 5).

[CDZ17] Thomas Colcombet, Laure Daviaud, and Florian Zuleger. "Automata and Program Analysis". In: *Fundamentals of Computation Theory, FCT*. Vol. 10472. 2017. DOI: 10.1007/978-3-662-55751-8_1 (cited on p. 5).

[DLT02] François Denis, Aurélien Lemay, and Alain Terlutte. "Residual Finite State Automata". In: *Fundamenta Informaticae* 51.4 (2002). URL: http://content.iospress.com/articles/fundamenta-informaticae/fi51-4-02 (cited on p. 4).

[DEP98] Josée Desharnais, Abbas Edalat, and Prakash Panangaden. "A Logical Characterization of Bisimulation for Labelled Markov Processes". In: *Logic in Computer Science, LICS*. 1998. DOI: 10.1109/LICS.1998.705681 (cited on p. 3).

[DJW97] Stefan Dziembowski, Marcin Jurdziński, and Igor Walukiewicz. "How much memory is needed to win infinite games?" In: *Logic in Computer Science, LICS*. 1997. DOI: 10.1109/LICS.1997.614939 (cited on p. 15).

[Esp16] Javier Esparza. "Parameterized Verification of Crowds of Anonymous Processes". In: *Dependable Software Systems Engineering*. IOS Press, 2016, pp. 59–71. DOI: 10.3233/978-1-61499-627-9-59 (cited on p. 6).

[EFM99] Javier Esparza, Alain Finkel, and Richard Mayr. "On the Verification of Broadcast Protocols". In: *Logic in Computer Science, LICS*. 1999, pp. 352–359. DOI: 10.1109/LICS.1999.782630 (cited on p. 6).

[Esp+17] Javier Esparza, Jan Křetínský, Jean-François Raskin, and Salomon Sickert. "From LTL and Limit-Deterministic Büchi Automata to Deterministic Parity Automata". In: *Tools and Algorithms for the Construction and Analysis of Systems, TACAS*. 2017. DOI: 10.1007/978-3-662-54577-5_25 (cited on p. 14).

[Fea+17] John Fearnley, Sanjay Jain, Sven Schewe, Frank Stephan, and Dominik Wojtczak. "An ordered approach to solving parity games in quasi polynomial time and quasi linear space". In: *International Symposium on Model Checking of Software, SPIN*. 2017, pp. 112–121 (cited on p. 10).

[Fli74] Michel Fliess. "Matrices de Hankel". In: *Journal de Mathématiques Pures et Appliquées* 53 (1974), pp. 197–222 (cited on p. 9).

[GS92] Steven M. German and A. Prasad Sistla. "Reasoning about systems with many processes". In: *Journal of the ACM* 39.3 (1992), pp. 675–735. DOI: 10.1145/146637.146681 (cited on p. 6).

[GI17] Hugo Gimbert and Rasmus Ibsen-Jensen. "A short proof of correctness of the quasi-polynomial time algorithm for parity games". In: *CoRR* abs/1702.01953 (2017). URL: http://arxiv.org/abs/1702.01953 (cited on p. 10).

[GO10] Hugo Gimbert and Youssouf Oualhadj. "Probabilistic Automata on Finite Words: Decidable and Undecidable Problems". In: *International Colloquium on Automata, Languages, and Programming, ICALP*. 2010, pp. 527–538. DOI: 10.1007/978-3-642-14162-1_44 (cited on p. 6).

[Gol78] E. Mark Gold. "Complexity of Automaton Identification from Given Data". In: *Information and Control* 37.3 (1978). DOI: 10.1016/S0019-9958(78)90562-4 (cited on p. 14).

[JTZ18] Swen Jacobs, Leander Tentrup, and Martin Zimmermann. "Distributed synthesis for parameterized temporal logics". In: *Information and Computation* 262 (2018). DOI: 10.1016/j.ic.2018.09.009 (cited on p. 8).

[JL17] Marcin Jurdziński and Ranko Lazić. "Succinct progress measures for solving parity games". In: *Logic in Computer Science, LICS*. 2017. DOI: 10.5555/3329995.3330027 (cited on pp. 10, 11).

[Jur+20] Marcin Jurdziński, Rémi Morvan, Pierre Ohlmann, and K. S. Thejaswini. "A symmetric attractor-decomposition lifting algorithm for parity games". In: *CoRR* (2020). URL: https://arxiv.org/abs/2010.08288 (cited on p. 11).

[Kop09] Eryk Kopczyński. "Half-positional determinacy of infinite games". PhD Thesis. University of Warsaw, 2009 (cited on p. 15).

[KPV09] Orna Kupferman, Nir Piterman, and Moshe Y Vardi. "From liveness to promptness". In: *Formal Methods in System Design* 34.2 (2009). DOI: 10.1007/s10703-009-0067-z (cited on pp. 7, 8).

[LLS18] Guillaume Lagarde, Nutan Limaye, and Srikanth Srinivasan. "Lower Bounds and PIT for Non-commutative Arithmetic Circuits with Restricted Parse Trees". In: *Computational Complexity* (2018), pp. 1–72. DOI: 10.1007/s00037-018-0171-9 (cited on p. 9).

[Leh18] Karoliina Lehtinen. "A modal-μ perspective on solving parity games in quasi-polynomial time". In: *Logic in Computer Science, LICS*. 2018, pp. 639–648. DOI: 10.1145/3209108.3209115 (cited on p. 10).

[LSW19] Karoliina Lehtinen, Sven Schewe, and Dominik Wojtczak. "Improving the complexity of Parys' recursive algorithm". In: *CoRR* (2019). URL: https://arxiv.org/abs/1904.11810 (cited on p. 11).

[LMS20] Michael Luttenberger, Philipp J. Meyer, and Salomon Sickert. "Practical synthesis of reactive systems from LTL specifications via parity games". In: *Acta Informatica* (2020). DOI: 10.1007/s00236-019-00349-3 (cited on p. 14).

[MS12] Shahar Maoz and Yaniv Sa'ar. "Assume-guarantee scenarios: Semantics and synthesis". In: *Model Driven Engineering Languages and Systems, MoDELS*. 2012. DOI: 10.1007/978-3-642-33666-9_22 (cited on p. 8).

[MST19] Corto Mascle, Mahsa Shirmohammadi, and Patrick Totzke. "Controlling a Random Population is EXPTIME-hard". In: *CoRR* (2019). URL: http://arxiv.org/abs/1909.06420 (cited on p. 7).

[McN93] Robert McNaughton. "Infinite Games Played on Finite Graphs". In: *Annals of Pure and Applied Logic* 65.2 (1993), pp. 149–184. DOI: 10.1016/0168-0072(93)90036-D (cited on p. 11).

[MC18] Thibaud Michaud and Maximilien Colange. "Reactive Synthesis from LTL Specification with Spot". In: *Synthesis Workshop, SYNT@CAV*. 2018 (cited on p. 14).

[MSM18] Matteo Mio, Michał Skrzypczak, and Henryk Michalewski. "Monadic Second Order Logic with Measure and Category Quantifiers". In: *Logical Methods in Computer Science* 14.2 (2018). DOI: 10.23638/LMCS-14(2:2)2018 (cited on p. 4).

[Mon19] David Monniaux. "On the decidability of the existence of polyhedral invariants in transition systems". In: *Acta Informatica* 56.4 (2019). DOI: 10.1007/s00236-018-0324-y (cited on p. 2).

[Nis91] Noam Nisan. "Lower Bounds for Non-Commutative Computation (Extended Abstract)". In: *Symposium on Theory of Computing, STOC*. 1991. DOI: 10.1145/103418.103462 (cited on p. 9).

[OW15] Joël Ouaknine and James Worrell. "On linear recurrence sequences and loop termination". In: *ACM SIGLOG News* 2.2 (2015). DOI: 10.1145/2766189.2766191 (cited on p. 1).

[OW14] Joël Ouaknine and James Worrell. "Positivity Problems for Low-Order Linear Recurrence Sequences". In: *International Symposium on Discrete Algorithms, SODA*. 2014. DOI: 10.1137/1.9781611973402.27 (cited on p. 1).

[Pan09] Prakash Panangaden. *Labelled Markov Processes*. Imperial College Press, 2009. DOI: 10.1142/p595 (cited on p. 3).

[Par19] Paweł Parys. "Parity Games: Zielonka's Algorithm in Quasi-Polynomial Time". In: *Mathematical Foundations of Computer Science, MFCS*. 2019, 10:1–10:13. DOI: 10.4230/LIPIcs.MFCS.2019.10 (cited on p. 11).

[Pit06] Nir Piterman. "From Nondeterministic Büchi and Streett Automata to Deterministic Parity Automata". In: *Logic in Computer Science, LICS*. 2006. DOI: 10.1109/LICS.2006.28 (cited on p. 15).

[PW93] Leonard Pitt and Manfred K. Warmuth. "The Minimum Consistent DFA Problem Cannot be Approximated within any Polynomial". In: *Journal of the ACM* 40.1 (1993). DOI: 10.1145/138027.138042 (cited on p. 14).

[Pnu77] Amir Pnueli. "The temporal logic of programs". In: *Symposium on Foundations of Computer Science, SFCS*. 1977. DOI: 10.1109/SFCS.1977.32 (cited on p. 14).

[PR89] Amir Pnueli and Roni Rosner. "On the synthesis of a reactive module". In: *Principles of Programming Languages, POPL*. 1989. DOI: 10.1145/75277.75293 (cited on pp. 7, 14).

[Rab69] Michael O. Rabin. "Decidability of Second-Order Theories and Automata on Infinite Trees". In: *Transactions of the AMS* 141 (1969). DOI: 10.2307/1995086 (cited on p. 4).

[Rab63] Michael O. Rabin. "Probabilistic Automata". In: *Information and Control* 6.3 (1963), pp. 230–245. DOI: 10.1016/S0019-9958(63)90290-0 (cited on p. 9).

[Sch09] Sven Schewe. "Tighter Bounds for the Determinisation of Büchi Automata". In: *Foundations of Software Science and Computation Structures, FoSSaCS*. 2009. DOI: 10.1007/978-3-642-00596-1_13 (cited on p. 15).

[Sch17] Sylvain Schmitz. "Algorithmic Complexity of Well-Quasi-Orders". Habilitation à diriger des recherches. École normale supérieure Paris-Saclay, 2017. URL: https://tel.archives-ouvertes.fr/tel-01663266 (cited on p. 7).

[SCH16] Karl Stratos, Michael Collins, and Daniel Hsu. "Unsupervised part-of-speech tagging with anchor hidden Markov models". In: *Transactions of the Association for Computational Linguistics* 4 (2016). DOI: 10.1162/tacl_a_00096 (cited on p. 4).

[Uhl+15] Jannis Uhlendorf, Agnès Miermont, Thierry Delaveau, Gilles Charvin, François Fages, Samuel Bottani, Pascal Hersen, and Gregory Batt. "In silico control of biomolecular processes". In: *Computational Methods in Synthetic Biology* 13 (2015), pp. 277–285. DOI: 10.1007/978-1-4939-1878-2_13 (cited on p. 6).

[Zie98] Wiesław Zielonka. "Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees". In: *Theoretical Computer Science* 200.1-2 (1998), pp. 135–183. DOI: 10.1016/S0304-3975(98)00009-7 (cited on pp. 11, 15).