Summary

1 Name

Michał Skrzypczak

2 Degrees

- MSc in mathematics, *cum laude*, University of Warsaw, 2010, thesis: ”O kolorowaniach drzewa Cantora” (ang. “On colourings of the Cantor tree”);
- Msc in computer science, *cum laude*, University of Warsaw, 2012, thesis: “O złożoności topologicznej języków definiowanych w logice MSO+U” (ang. “On the topological complexity of languages definable in MSO+U”);
- PhD z mathematics (specialisation computer science), *cum laude*, University of Warsaw, 2014, thesis: “Descriptive set theoretic methods in automata theory”.

3 Professional career

- 2015 — . . . , assistant professor, University of Warsaw, Poland;
- 2015, post–doc, Liafa (IRIF), Université Denis Diderot — Paris 7;
- 2012, 4-months visit, LaBRI, Université Bordeaux;
- 2010 — 2014, scientific assistant, University of Warsaw, Poland;

4 Thesis

4.1 Title

Weak forms of non-determinism in automata theory

4.2 Constituting papers

[A] Denis Kuperberg, Michał Skrzypczak.
On determinisation of GFG automata,
In ICALP, Springer-Verlag LNCS, pages 299–310, 2015.

[B] Michał Skrzypczak, Igor Walukiewicz.
Deciding the topological complexity of Büchi languages,
In ICALP, LIPIcs, pages 99:1–99:13, 2016.

[C] Filippo Cavallari, Henryk Michalewski, Michał Skrzypczak.
A characterisation of Π^0_2 regular tree languages,
In MFCS, LIPIcs, pages 56:1–56:14, 2017.

[D] Udi Boker, Orna Kupferman, Michał Skrzypczak.
How deterministic are GFG automata,
In FSTTCS, LIPIcs, pages 18:1–18:14, 2017.

[E] Michał Skrzypczak.
Unambiguous languages exhaust the index hierarchy,
In ICALP, LIPIcs, pages 140:1–140:14, 2018.

[F] Michał Skrzypczak.
Büchi VASS recognise Σ^1_1-complete ω-languages,
In RP, Springer-Verlag LNCS, pages 133–145, 2018.

4.3 Synopsis

4.3.1 Introduction

Non-determinism is one of the crucial concepts of modern theoretical computer science. Many of the fundamental questions of this discipline, like the P=NP question, ask to understand the strength of non-determinism. The situation is a bit easier in automata theory, where the differences between deterministic and non-deterministic automata are better understood. The central motivation of this scientific achievement is to understand typical forms of non-determinism for standard models of automata.

Abstractly, an automaton can be seen as a finite state machine, that parses a given input structure and updates its internal state, depending on the successive letters from a finite alphabet. It is known that allowing non-determinism (i.e. *guessing*) for a given model of automata often leads to an increase of the expressive power of the model. Moreover, even if the expressive power stays the same, non-deterministic automata can be exponentially more succinct than their deterministic variants. This discrepancy is even greater in the case of alternating machines, where together with the standard non-determinism seen as an *existential* mode, a dual *universal* mode is allowed. The interaction between these two modes is modelled in terms of a game.

Unfortunately, the above-mentioned advantages of non-determinism come at a cost. The fact that the machine can guess leads to additional mathematical difficulties when proving some properties of the given model. It often results in the fact that the model is unsuited for specific applications (e.g. non-deterministic automata cannot be used directly to solve the problem of synthesis). Moreover, the need to consider multiple execution paths of a given machine usually increases the computational complexity of the typical decision problems and constructions. In some extreme cases, it may even lead to undecidability (like in the case of non-emptiness for alternating pushdown automata).

Taking into account the difficulties related to non-determinism and alternation, it seems natural to search for intermediate models, where the power of these modes is partially limited. Still, one would like to keep some possibility of *guessing*, to enjoy the extended expressive power or succinctness of lack of full determinism. Thus, one needs to take into account a trade-off between the gains and the cost of the involved non-determinism. Following this line of reasoning, the following intermediate forms of non-determinism have been proposed (a more precise explanation is given later in this synopsis):

(**D**) **deterministic automata**, having no ability to *guess*;

(**U**) **unambiguous automata**, i.e. those non-deterministic automata that satisfy an additional semantic condition, that for each input there is at most one way of *guessing* (a run) that leads to acceptance;

(**G**) ***Good-For-Games*** **automata** (denoted GFG), i.e. those non-deterministic automata, which admit a method of resolving the non-determinism based purely on the part of the input that was read so far;

(**N**) **non-deterministic automata**, working entirely in the *existential* mode;

(**A**) **alternating automata**, where the modes of *existential* and *universal* choice can freely interact.

The crucial gains offered by the stronger forms of non-determinism are rather clear: non-deterministic automata can *guess* successive transitions; while alternating automata are easily closed under Boolean combinations (disjunction, conjunction, and negation). On the other hand, multiple results show that giving up the full strength of non-determinism can lead to concrete gains in the considered applications. A classical example is the case of unambiguous automata over finite words: similarly to non-deterministic automata, they are easily closed under reversal of the input; while their universality problem can be solved in polynomial time [SI85], even though the general problem for non-deterministic automata is PSPACE-complete [SM73]. Similarly, GFG automata can be used interchangeably with deterministic ones in the problem of synthesis; while their symbolic representations may be simpler than for equivalent deterministic automata [HP06] (Theorem 4.7 below provides additional advantages of GFG automata over deterministic ones).

The scientific goal of my research was to provide a precise understanding of the dependencies between classes of automata of the above forms of non-determinism. I've put special emphasis on the trade-off between expressive power, number of states, acceptance condition, and complexity of the related computational problems.

Directly from the definitions, one can observe the following inclusions, saying that each automaton of a given class can be treated as an automaton of the broader class: (**D**) $\subseteq$ (**U**),(**G**) $\subseteq$ (**N**) $\subseteq$ (**A**); where the classes (**U**) and (**G**) are *a priori* incomparable. The opposite inclusions, understood literally in the sense of containment of the respective classes of automata, are easily violated in most of the models.

Essentially each of the above inclusions is a potential place to find a trade-off, allowing us to provide more effective algorithms, based on the considered automata. However, to do so, one needs to find answers to the most fundamental questions about the nature of such an inclusion. The first of them is the question of containment between the respective classes of languages:

Problem 4.1 *Is it the case that every language recognisable by an automaton of the broader class can be recognised by an automaton of the narrower class?*

In the case when the containment of the classes of languages does not hold, it becomes crucial to understand which languages of the broader class belong to the narrower class:

Problem 4.2 *Is there a characterisation or an estimation on their complexity, explaining which languages recognisable by the automata of the broader class can also be recognised by the automata of the narrower class?*

In the other case, when the classes of languages coincide, it becomes important to understand the cost of the transformations of the automata from the broader class to the narrower:

Problem 4.3 *Is there an upper bound on the increase in complexity (e.g. the number of states) when translating an automaton of the broader class into an equivalent automaton of the narrower class?*

Additionally, because of the asymmetry of the concept of non-determinism, when studying the above problems it is usually important to focus on the analogous questions for the dual class, that is the class of complements of languages recognisable by the given automata.

Clearly, the answers to these questions depend on the concrete model of considered machines. Among the studied models are finite automata, counter automata, and automata with various acceptance conditions. The situation also varies depending on the considered structures. Due to the fact that most of the questions simplify in the case of finite structures, most of the research is devoted to infinite words and infinite trees.

The main results of my scientific achievement focus on finding the answers to the above problems in the context of the above-mentioned models of automata. Due to the slightly different set of applied tools, the explanation of the results is split into two threads.

The first of the discussed threads regards automata over infinite words. It is concerned with the acceptance conditions of Büchi, co-Büchi, Rabin, Streett, and parity (together with its *weak* variant). This thread focuses on finite automata and automata with *blind counters*. The papers included in that thread are [A], [D], and [F]. The most important forms of non-determinism are GFG automata and unambiguous automata.

The second thread is about infinite trees. The research focuses on the frontiers between unambiguous, non-deterministic, and alternating automata. The main pressure is put on the acceptance conditions of Büchi and weak parity. Papers included in that thread are [B], [C], and [E].

The results of these papers are presented in the context of the former state of the affairs. Therefore, I cite theorems and conjectures from other publications, not included in my scientific achievement. The theorems from the papers of the scientific achievement are marked by the symbol ★ at the margin.

4.3.2 Basic notions

A general introduction to the discussed concepts and notation of automata theory can be found in [Tho96]. The references to descriptive set theory are based on the presentation in [Kec95]. The exact notation and nomenclature come from [Skr16].

Let us recall that a *non-deterministic automaton* over infinite words is a tuple $\mathcal{A} = \langle \Sigma, Q, q_{\mathrm{I}}, \delta, \lambda \rangle$, where Σ is a finite *alphabet*; Q is a finite set of *states* of the automaton; $q_{\mathrm{I}} \in Q$ is the *initial* state; $\delta \subseteq Q \times \Sigma \times Q$ is a *transition relation* of the automaton; and λ is an *acceptance condition*. A *deterministic automaton* is a special case of a non-deterministic one, for which $\delta\colon Q \times \Sigma \to Q$ is a functional dependency. A *run* of an automaton over a given infinite word $\alpha \in \Sigma^\omega$ is an infinite word $\rho \in Q^\omega$ that is labelled by the states of the automaton, such that $\rho(0) = q_{\mathrm{I}}$ and for $n = 0, 1, \ldots$ the successive states $\rho(n)$ and $\rho(n{+}1)$ agree with the letter $\alpha(n)$ and the transition relation δ: $\big(\rho(n), \alpha(n), \rho(n{+}1)\big) \in \delta$.

An acceptance condition λ determines which infinite sequences of states $\rho \in Q^\omega$ are considered *accepting*. A *parity* condition is given by a function $\lambda\colon Q \to \mathbb{N}$ that assigns to each state a *priority*. An infinite sequence of states satisfies this condition if the greatest priority that appears infinitely often is an even number. We say that a parity condition λ is of *index* (i,j) for $i \leqslant j$ if $\mathrm{rg}(\lambda) \subseteq \{i,\ldots,j\}$. A *weak* parity condition is a parity condition in which the priorities of states are non-decreasing along the transitions of the automaton. The conditions of *Büchi* and the dual (called *co-Büchi*) can be viewed as parity conditions of indices $(1,2)$ and $(0,1)$ respectively. A *Rabin* condition is a disjunction of some family of parity conditions of index $(1,3)$. A *Streett* condition is the negation of a Rabin condition.

Infinite tree automata are similar to infinite word automata, except that their transition relation δ is a subset of $Q\times\Sigma\times Q\times Q$ in the non-deterministic case; or a function $\delta\colon Q\times\Sigma\to Q\times Q$ in the deterministic case. A run of such an automaton over a tree t is a tree $\rho\colon\{\mathtt{L},\mathtt{R}\}^*\to Q$, such that $\rho(\epsilon)=q_{\mathrm{I}}$ and for each node $v\in\{\mathtt{L},\mathtt{R}\}^*$ we have $\big(\rho(v),t(v),\rho(v\mathtt{L}),\rho(v\mathtt{R})\big)\in\delta$. Such a run is *accepting* if it satisfies the acceptance condition on all the infinite branches of the tree.

The *language* $\mathrm{L}(\mathcal{A})$ recognised by a given non-deterministic automaton is the set of infinite words or trees over which there exists an accepting run. Alternating automata differ from non-deterministic ones in that the transition function assigns to each state and letter a positive Boolean combination of the consecutive states. The semantics of such transitions is interpreted in terms of a game. An infinite word or tree belongs to the language recognised by such an automaton if the positive player has a winning strategy in the respective game. A language L (a set of words or trees) is called *regular* if it can be recognised by a non-deterministic (equivalently alternating) parity automaton.

A non-deterministic automaton is called *unambiguous* if it has at most one accepting run over every input (i.e. word or tree).

4.3.3 Non-determinism over infinite words — GFG automata

The papers [A] and [D] focus on the case of automata over infinite words. The aim of these articles is to understand the differences between deterministic automata and GFG automata.

Definition 4.4 *A non-deterministic automaton $\mathcal{A}$ over infinite words is called* GFG *if there exists a function $\sigma\colon\Sigma^*\to Q$, such that for each infinite word accepted by the automaton $\alpha\in\mathrm{L}(\mathcal{A})$, the sequence of states $\rho\in\delta^\omega$ defined as $\rho(n)\stackrel{def}{=}\sigma(\alpha\!\restriction_n)$ for $n=0,1,\ldots$ is an accepting run of $\mathcal{A}$ over α.*

This condition means, that there exists a method (an *advice*) that allows choosing the successive transitions of a GFG automaton, based purely on the already read part of the input word. The above requirement says that whenever the processed word belongs to the language recognised by the automaton (i.e. it can be accepted in a non-deterministic way), then the states selected by the above *advice* form an accepting run. In other words, it is possible to resolve the non-determinism of the given automaton on the fly, without knowing the rest of the input, and still accept all the words that can be accepted at all.

The class of GFG automata has been introduced in [HP06]. The idea behind this definition is that a GFG automaton (in contrast with a general non-deterministic automaton) can be used directly in the context of games, for instance when solving the problem of synthesis. This means that finding GFG automata that are simpler than equivalent deterministic ones would give a chance of faster algorithms for that problem. Also, an independent study of GFG automata under the name of *history deterministic* automata has been performed in the context of boundedness conditions and cost functions [CL10, Col13].

A special case of GFG automata are those, where one can find an equivalent deterministic automaton inside the structure of the given one:

Definition 4.5 *A non-deterministic automaton $\mathcal{A}$ is called* DBP *(determinisable by pruning) if there exists an automaton $\mathcal{A}'$, that is obtained from $\mathcal{A}$ by removing states and transitions, such that $\mathcal{A}'$ is deterministic and the languages of $\mathcal{A}$ and $\mathcal{A}'$ are the same.*

Because of the fact that already deterministic parity automata recognise all regular languages of infinite words, the answer to Problem 4.1 is trivially positive, both for GFG and DBP automata.

The starting point of the research of [A] was an earlier work [BKKS13]. The following two theorems were proved there, solving the respective inclusion problems for these classes of automata:

Theorem 4.6 ([BKKS13])

a) The class of GFG automata coincides with the class of Good-For-Trees *automata*[1].

b) There exist GFG automata with Büchi and co-Büchi conditions that are not DBP.

The first of these results indicates a connection between GFG automata over infinite words and a certain class of languages of infinite trees (this class can be seen as a restriction of the class of deterministic languages). In this context, the results about e.g. determinisation of GFG automata can be treated as results regarding determinisation of specific non-deterministic automata over infinite trees.

The second of the above results signalises that GFG automata may have an essentially non-deterministic structure. However, this result has not resolved whether GFG automata can be essentially simpler than deterministic ones because in the case of the provided examples there are equally complex equivalent deterministic automata. In other words, Problem 4.3 was left open in the case of GFG automata. Exactly that question was the initial motivation for [A], whose main results about determinisation are as follows:

★ **Theorem 4.7** ([A])

a) (Theorem 1.) There exist GFG automata with a co-Büchi acceptance condition and $2n+1$ states, such that every equivalent deterministic automaton has at least $\frac{2^n}{2n+1}$ states.

b) (Theorem 8.) There exists an algorithm that reads a GFG Büchi automaton with n states and constructs an equivalent deterministic automaton with $O(n^2)$ states.

The proof of Item a) of this theorem is based on a construction of a family of GFG automata. The fact that all these automata are GFG is relatively easy. However, the proof of the lower bound for the size for their determinisations requires a pumping-style argument. What is interesting, the argument does not indicate a concrete pumping scheme (in the form of a loop in a certain graph), which is enough to find the demanded contradiction. Instead of that, a limitary pumping argument is used: an infinite family of runs is constructed, each of them satisfying certain invariants. Then, based on the argument of topological compactness of the respective space of runs, a limitary run is chosen, that satisfies all the invariants.

[1] Good-For-Tree automata are those automata over infinite words that can be directly used to recognise languages of infinite trees of the form $\mathrm{Path}(L)$ — such a language contains all the partial infinite trees whose all infinite paths belong to a given language of words $L \subseteq \Sigma^\omega$.

The proof of Item b) is based on a concrete algorithm, that reads a given automaton $\mathcal{A}$ and performs certain modifications on it. The initial part of the construction is based on a game, designed to verify if a given automaton is GFG (this game is called the *GFG game*). Due to the assumptions on $\mathcal{A}$ we know that the positive player wins this game. Then, one considers a specific strategy for this player that optimises *parity signatures* [SE89, Wal96]. The good combinatorial properties of that strategy allow performing an inductive procedure of simplification of its structure, that ultimately leads to a small deterministic automaton for the language $\mathrm{L}(\mathcal{A})$.

Since the Büchi condition corresponds to the parity index $(1, 2)$ (analogously co-Büchi condition corresponds to the index $(0, 1)$), the results of Theorem 4.7 provide a complete classification of the increase in the number of states during determinisation of parity GFG automata: the increase is polynomial for automata of index $(1, 2)$ and below; and exponential for automata of index $(0, 1)$ and above. Additionally, Item a) of Theorem 4.7 shows that GFG automata can be essentially smaller than equivalent deterministic ones. This means that indeed, in certain situations, using GFG automata instead of deterministic ones may lead to an exponential speed-up in the solution of the problem of synthesis.

A continuation of the study of differences between GFG automata and deterministic ones was performed in [D]. The considered measure of similarity, instead of the number of states, was based on the notion of *typeness*. The following definition formalises this concept adapted to the case of GFG automata:

Definition 4.8 *Consider two types of acceptance conditions λ and λ'. We say that GFG automata λ are λ'-*type *if: whenever $\mathcal{A}$ is a GFG automaton with an acceptance condition of type λ, such that the language $\mathrm{L}(\mathcal{A})$ can be recognised by a deterministic automaton with an acceptance condition of type λ', then there exists a GFG automaton $\mathcal{A}'$ with the same structure as $\mathcal{A}$, but with an acceptance condition of type λ'.*

This concept was introduced in [KPB94] in the context of deterministic automata. Later on, in [KMM06] the authors prove lack of λ'-typeness for non-deterministic automata. The aim of [D] was to study to what extent this concept transfers to GFG automata, that is how much similar to deterministic ones are they. According to the definition, the concept of being λ-type can be seen as an instance of Problem 4.3, where the notion of complexity is understood as the kind of the acceptance condition, without the change of the structure of the automaton.

The main results of [D] regarding typeness of GFG automata are presented below. These results rely on the notion of *tightness* of a GFG automaton $\mathcal{A}$, that roughly says that the automaton does not contain any transition that is unnecessary from the perspective of some function σ, witnessing that $\mathcal{A}$ is GFG.

★ **Theorem 4.9** ([D])

a) (Theorem 10.) Tight GFG Streett automata are co-Büchi-type.

b) (Theorem 11.) Tight GFG Rabin automata are Büchi-type.

b) (Examples 11. and 12.) The assumption of tightness is necessary in both above theorems.

c) (Corollary 16.) Tight GFG automata with Streett and Rabin conditions are type for the weak parity condition.

Comp. To / From	W	C	B	P	R	S
Weak	Poly	Poly	Poly	Poly	Poly	Poly
Büchi	Poly	Poly	Poly	Poly	Poly	Poly
Co-Büchi	Poly	Poly	Exp	Exp	Exp	?
Parity	Poly	Poly	Exp	Exp	Exp	?
Rabin	Poly	Poly	Exp	Exp	Exp	?
Streett	Poly	Exp	Exp	Exp	Exp	Exp

Table 1: Increase in the number of states when complementing GFG automata. The following notation is used for the acceptance conditions: W – weak parity; C – co-Büchi; B – Büchi; P – parity; R – Rabin; and S – Streett.

Moreover, this paper provided further extensions, based on the results of [A], about determinisation and complementation of GFG automata:

★ **Theorem 4.10** ([D])

a) (Corollary 14.) Consider a GFG automaton $\mathcal{A}$ with a Rabin condition that has n states. If $\mathrm{L}(\mathcal{A})$ can be recognised by a deterministic Büchi automaton then there exists a deterministic Büchi automaton recognising $\mathrm{L}(\mathcal{A})$ that has $O(n^2)$ states.

b) (Theorem 17.) GFG automata with weak parity conditions are DBP.

c) (Theorem 20.) The complementation of GFG automata with a change of acceptance condition is summarised in Table 1 (see Table 2 of [D]).

Finally, Section 3 of [D] considers the question of typeness for unambiguous automata: the adequate definition of typeness is the same as Definition 4.8, except that we require the automaton $\mathcal{A}$ to be unambiguous (instead of GFG) and $\mathcal{A}'$ can be any non-deterministic automaton. The results obtained in that direction are as follows:

★ **Theorem 4.11** ([D])

a) (Examples 7. and 8.) Unambiguous parity automata are not Büchi- nor co-Büchi-type.

b) (Proposition 9.) Unambiguous parity automata that are GFG are also DBP.

The presented results of [D] can be summarised as follows. First of all, GFG automata admit similar typeness properties as deterministic automata. Second, from the point of view of complementation, GFG automata fall somewhere in-between deterministic and non-deterministic ones. Finally, Item a) of Theorem 4.11 shows that from the perspective of typeness, unambiguous automata are more similar to non-deterministic than to deterministic ones. On the other hand, Item b) of Theorem 4.11 indicates that the requirement of unambiguity trivialises the notion of GFG automata.

4.3.4 Non-determinism over infinite words — blind counter automata

One of the most natural ways of increasing the power of finite automata is to add *blind counters*. Such an automaton, called *blind multi-counter* (BMC) automaton, together with a finite set of states Q, has a finite number of *counters* C, that store natural numbers. Initially, all the counters store value 0. A transition relation of such an automaton is a finite set δ of *transitions*. A transition $(q, a, \tau, q') \in \delta$ means that the automaton, while in the state $q \in Q$, may read the letter $a \in \Sigma$ and move to the state $q' \in Q$, updating its counter values according to the given vector $\tau \in \mathbb{Z}^C$ — if this would force any of the counters to store a negative value then such a transition is not allowed. Except for the above restriction, the values of the counters do not influence the behaviour of the automaton, that is why the counters are called *blind*. An acceptance condition λ of such an automaton is given purely in terms of the sequence of the visited states.

The definition of BMC automata is based on the model of VASS (vector addition systems with states) and related Petri nets. Due to the inherent non-determinism of these models, also BMC automata are usually considered in the non-deterministic variant. The fact that these automata cannot explicitly test the values of their counters, implies that the emptiness problem is decidable for this model. Moreover, due to their non-determinism, it is easy to show that the expressive power of these automata does not depend whether the acceptance condition is Büchi, Streett, Rabin, or parity. Because of this fact, we often speak simply about non-deterministic BMC automata, without specifying the acceptance condition.

The main result of [FS14] shows that, in general, non-deterministic BMC automata are more expressive than their deterministic variants. Therefore, it constitutes an answer to Problem 4.1 for this class:

Theorem 4.12 ([FS14]) *There exists a language recognisable by a non-deterministic BMC automaton, that cannot be recognised by any deterministic BMC automaton with the parity acceptance condition*[2]*.*

The reasoning provided in [FS14] is based on a topological argument: the presented example has topological complexity $\mathbf{\Sigma}^0_3$, which exceeds the topological complexity of any deterministic parity machines. The provided example uses the available non-determinism in a very limited way and it seemed that it might be hard to provide any essentially more complex language within this model. It all suggested that the class $\mathbf{\Sigma}^0_3$ is the upper bound for the topological complexity of non-deterministic BMC automata. In such a case it would be quite plausible that there exists a model of deterministic machines (or at least unambiguous), with a reasonably simple acceptance condition (e.g. in $\mathbf{\Sigma}^0_3$), that are able to recognise all the languages recognisable by non-deterministic BMC automata. Despite the effort, no such model was found.

The answer to the question of the existence of such a model was given in the paper [F], proving that non-determinism in the case of BMC automata is in a sense inherent:

★ **Theorem 4.13** ([F]) *There exists a language recognisable by a non-deterministic BMC automaton with one blind counter, whose topological complexity is $\mathbf{\Sigma}^1_1$ (in particular, this language is non-Borel). This language cannot be recognised by any deterministic nor unambiguous machine with a Borel acceptance condition (nor any acceptance condition in $\mathbf{\Sigma}^0_3$).*

[2] In fact, this language cannot be recognised even by deterministic parity Turing machines.

Although [F] does not mention the model of GFG machines, the provided example excludes this model as well, due to the following corollary:

Corollary 4.14 *The language from Theorem 4.13 is not recognised by any GFG machine with a Borel acceptance condition.*

Proof sketch. Assume to the contrary and let $\mathcal{A}$ be a machine as in the statement. Assume that δ is (at most countable) set of transitions of that machine. A function[3] $\sigma\colon \Sigma^+ \to \delta$ witnessing that $\mathcal{A}$ is GFG, provides a continuous transformation $\widehat{\sigma}\colon \Sigma^\omega \to \delta^\omega$. Therefore, the graph of this transformation $\mathrm{Graph}(\widehat{\sigma}) \subseteq \big(\Sigma \times \delta\big)^\omega$ is a closed subset with sections of cardinality at most 1. The language $\mathrm{L}(\mathcal{A})$ is the projection onto the coordinate Σ^ω of the intersection of $\mathrm{Graph}(\widehat{\sigma})$ with the Borel set of accepting runs. Therefore, analogously as in Theorem 2 from [F], the language $\mathrm{L}(\mathcal{A})$ is Borel, which contradicts Theorem 4.13. ■

These results imply that even in the case of one blind counter, the class of BMC automata exhibit the full power of non-determinism, while all deterministic, unambiguous, or GFG models are substantially weaker.

4.3.5 Non-determinism over infinite trees — classes with de-alternation

In the case of parity automata over infinite words, all the forms of non-determinism have the same expressive power. This property stops to be true when the considered models are infinite trees. Firstly, deterministic automata are strictly weaker than the non-deterministic ones. Secondly, although the expressive power of non-deterministic and alternating parity automata coincides; the transformation from a non-deterministic automaton to an alternating one may increase the parity index in an uncontrollable way. The class of Büchi automata, defining a strict subclass of all regular tree languages, is a notable exception:

Theorem 4.15 ([MS95], [Rab70]) *If a regular tree language L can be recognised by an alternating Büchi automaton then L can also be recognised by a non-deterministic Büchi automaton. In other words, Büchi languages admit* de-alternation*.*

Moreover, L can be recognised by an alternating weak parity automaton (called weak-ATA*) if and only if both L and the complement L^{c} can be recognised by non-deterministic Büchi automata.*

The above results can be interpreted as follows: the Büchi condition over infinite trees is in a sense stable under additional alternation; while weak-ATA languages form a class of symmetrically limited non-determinism (from the perspective of strong parity conditions). These properties lead to the following interesting characterisation of weak-ATA languages within Büchi languages (i.e. a solution of Problem 4.2 in this case):

Theorem 4.16 ([CKLV13], alternative proof in [Skr14a, Chapter 2]) *There exists an algorithm that inputs a non-deterministic Büchi automaton $\mathcal{A}$ over infinite trees and decides whether the language* $\mathrm{L}(\mathcal{A})$ *can be recognised by any weak-ATA. If the answer is yes, the algorithm can construct such a weak-ATA.*

[3] Due to a technical reasons one needs to consider an *advice* of transitions instead of states of the machine (see Definition 4.5). The reason for that is that the sole sequence of states may not determine uniquely the values of the counters during the considered run.

Although this theorem characterises the class of weak-ATA languages, it does not provide new information about their complexity within all Büchi languages. In particular, it does not provide an answer to the following conjecture, when restricted to Büchi languages:

Conjecture 4.17 ([Sku93]) *If a regular tree language is Borel then it can be recognised by a weak-ATA.*

This conjecture, similarly as the argument involved in Theorem 4.13, is based on the intuition that Borel languages should correspond to some limited forms of non-determinism. It can be seen as the reverse implication to the following known result:

Theorem 4.18 ([DM07], the proof given *de facto* in [Mos91]) *Each language recognised by a weak-ATA is Borel.*

The main result of [B] provides a positive answer to the above conjecture when restricted to Büchi languages:

★ **Theorem 4.19** ([B]) *Let $\mathcal{A}$ be a non-deterministic Büchi language over infinite trees. Then the following conditions are equivalent:*

1. *the language* $\mathrm{L}(\mathcal{A})$ *can be recognised by a weak-ATA,*

2. *the language* $\mathrm{L}(\mathcal{A})$ *is Borel.*

Moreover, the problem of deciding whether any of the above properties holds is EXPTIME-complete.

The structure of the proof in the above theorem is similar to the structure of the proof of Theorem 2 from my PhD thesis [Skr14a]: the idea is based on an appropriately designed game, that allows to partially eliminate the non-determinism of the input automaton. The aim of the negative player in this game is to witness, that there exist trees that belong to the given language, over which a properly defined variant of parity signature can be arbitrarily big. Due to a well-chosen winning condition of that game, we get not only an algorithm of the optimal complexity, but additionally, we can conclude that Büchi languages which are not weak-ATA must be $\mathbf{\Sigma}^1_1$-complete (and in particular non-Borel).

The above constructions suggest that certain decision problems about languages of infinite trees may simplify in the case of classes admitting de-alternation. The known examples of such classes, together with the Büchi condition, are also the conditions of *reachability* (weak parity index $(1,2)$) and *safety* (weak index $(0,1)$). Both of these cases confirm the above intuition:

Theorem 4.20 ([Wal02, Section 6], [Cav18, Section 3.2], or [JL02]) *The following conditions are equivalent for a regular tree language L:*

1. *L is a* closed *set, i.e.* $\mathbf{\Pi}^0_1$ *(reps.* open *that is* $\mathbf{\Sigma}^0_1$*),*

2. *L can be recognised by an alternating safety (resp. reachability) automaton,*

3. *L can be recognised by a non-deterministic safety (resp. reachability) automaton.*

Moreover, the problem of deciding whether any of the above properties holds is EXPTIME-complete.

An analogous de-alternation construction turned out to be crucial when developing the results of [C]. The starting point of this study was an attempt to understand in the sense of Problem 4.2, which regular tree languages belong to the class $\mathbf{\Delta}^0_2 = \mathbf{\Sigma}^0_2 \cap \mathbf{\Pi}^0_2$ of topological complexity (the results of [FM14] contain an essential mistake, see [BCPS18, Section 14]). During a study of this problem, it was observed that it might be easier to work with the non-symmetric class $\mathbf{\Pi}^0_2$. Further combinatorial analysis of this problem led to the following new de-alternation theorem:

★ **Theorem 4.21** ([C, Theorem 13.]) *If a regular tree language L can be recognised by an alternating automaton of weak parity index* $(1,3)$*, then L can also be recognised by a non-deterministic automaton of the same index.*

The above result of de-alternation, together with the topological ideas from Theorem 4.19 (see Proposition 11 from Section 5 in [B]), allowed us to construct a game, that controls the non-determinism of a given automaton. This all led to a proof of the following theorem:

★ **Theorem 4.22** ([C, Theorems 2. and 13.]) *The following conditions are equivalent for a regular language of infinite trees L:*

1. *L belongs to the class $\mathbf{\Sigma}^0_2$ of topological complexity,*
2. *L can be recognised by an alternating automaton of weak index* $(1,3)$*,*
3. *L can be recognised by a non-deterministic automaton of weak index* $(1,3)$*.*

Moreover, there exists an algorithm that decides whether the above properties hold.

The class of languages characterised by this theorem is relatively small. However, up to now, it is the widest class that has an effective characterisation, admitting any regular language as its input—the results of Theorems 4.16 and 4.19 are restricted to Büchi automata.

4.3.6 Non-determinism over infinite trees — unambiguous languages

The results about infinite trees that were mentioned above are focused on deterministic automata (see Item a) of Theorem 4.6 and the results about GFG automata), and non-deterministic and alternating automata (Section 4.3.5). The following section is devoted to the results from [E] about unambiguous automata[4].

It is quite easy to provide examples showing that deterministic automata over infinite trees are essentially weaker than non-deterministic ones. The situation of Problem 4.1 stated for unambiguous versus non-deterministic automata is much more complex. The first example of a regular tree language that cannot be recognised by any unambiguous parity automaton (i.e. is *ambiguous*) was provided in [NW96] (results extended in [CLNW10]). The example given there is the language of trees over the alphabet $\{a, b\}$, that contain at least one occurrence of the letter a. The simplicity of that language suggests that essentially any form of genuine non-determinism must lead to an ambiguous language. On the other hand, it turns out to be quite difficult to prove that a given language is, in fact, ambiguous. Currently, there are only two essentially different such examples known: the language from [NW96] and

[4]The notion of GFG automata over trees has not been explicitly introduced. However, it seems that the most natural definition would result in the standard GFG automata over infinite words. Therefore, this model is not discussed here.

the language of thin trees from [BS13]. Additionally, no effective characterisation (i.e. an answer to Problem 4.2) is known for the class of unambiguous languages within all regular tree languages. The only partial answers in this direction are given in [BS13].

The results mentioned above provide a rather mysterious picture of the situation: on the one hand, it seems to be difficult to guarantee that a given language is unambiguous; on the other hand, we know very few different examples of ambiguous languages. One of the conjectures based on this state of the affairs was, that maybe unambiguous languages are not much distinct from the deterministic ones, i.e. they also belong to the class $\mathbf{\Pi}^1_1$ of topological complexity.

The first progress in that direction was given in [Hum12], where an example is given of an unambiguous language whose topological complexity goes beyond the class $\mathbf{\Pi}^1_1$. Further developments led to the following theorem:

Theorem 4.23 ([DFH15]) *There exist unambiguous languages of increasing topological complexity within the second level of the alternating index hierarchy.*

An important element of the proposed construction is the fact, that the considered languages are in fact *bi-unambiguous*: both the given language L and its complement L^{c} are unambiguous.

The results of [DFH15] show that bi-unambiguous languages are noticeably more complex than deterministic ones. However, at the same time, these results indicate how hard it may be to provide unambiguous languages outside of the second level of the alternating index hierarchy. It all led to the formulation of the following conjecture (it is a consequence of Conjecture 5.27 from [Skr14a] in conjunction with Theorem 26 from [ISB16]):

Conjecture 4.24 *Each bi-unambiguous language can be recognised by an alternating automaton of index* $(1, 3)$*.*

The above conjecture remains open, in particular, no new examples of bi-unambiguous languages were found. However, during the further research on the complexity of unambiguous languages, the following theorem has been proved. The class $\mathrm{Comp}(i, j)$ in the statement is motivated by μ-calculus, it is the class of languages that admit certain alternation between conditions of index (i, j) and $(i+1, j+1)$.

Theorem 4.25 ([Skr14a, Chapter 1], [MS16], and [MS18])

a) If a given unambiguous automaton is of index $(i, 2j)$ *then its language belongs to the class* $\mathrm{Comp}(i+1, 2j)$*.*

b) An analogous collapse does not hold for unambiguous automata with weak parity conditions.

One of the consequences of that theorem is the following corollary:

Corollary 4.26 *Every unambiguous Büchi automaton recognises a language recognisable by a weak-ATA.*

Due to Theorem 4.18, this corollary is a strengthening of the previously known theorem, obtained by purely topological methods:

Theorem 4.27 ([FS09]) *Every unambiguous Büchi automaton recognises a Borel language.*

It is worth noticing that Corollary 4.26 can be independently deduced from the above Theorem 4.27, using additionally Theorem 4.19, saying that a Borel Büchi language must be weak-ATA recognisable. However, the direct construction from Theorem 4.25 provides an automaton with polynomially many states comparing to the original one; while the automaton obtained via Theorem 4.19 may be exponentially bigger.

At the time of proving Item a) of Theorem 4.25 it was not clear for how many indices $(i, 2j)$ the result is meaningful — it was still possible that all unambiguous tree languages can be recognised by automata of bounded indices. As it turns out, it is not the case:

★ **Theorem 4.28** ([E]) *There exist unambiguous languages lying arbitrarily high in the alternating index hierarchy.*

The crucial part of this construction is a new game that allows us to compare the signatures of parity games. It turns out that to recognise in an unambiguous way complex regular languages, it is enough to allow the automaton to guess *optimal* strategies with respect to the parity signatures. It can be seen as an analogue to Item b) of Theorem 4.7, where one uses a signature-optimal strategy in the GFG game to obtain good combinatorial properties of this object.

Theorem 4.28 says that unambiguous languages may be arbitrarily *complex* among regular tree languages. It means that the previous intuitions, indicating their limited access to non-determinism, are wrong. On the other hand, a large part of the technical aspect of [E] is devoted to ensuring that all the choices made non-deterministically by the automaton can, in fact, be done in an unambiguous way. In other words, it takes a lot of effort to ensure that the given kind of complexity in the sense of index, can be recognised in the unambiguous mode. These complications are one of the reasons why Conjecture 4.24 still remains open.

4.3.7 Conclusions

The results contained in this scientific achievement extend our understanding of the weak forms of non-determinism. New developments are obtained regarding the considered models of automata, both in the case of infinite words and infinite trees.

The papers [A] and [D] provided a rather complete picture about GFG automata, as compared to deterministic, unambiguous, and non-deterministic ones. Because of the relationship with Good-For-Trees automata, these results can be related to certain variants of deterministic automata over trees. The results of [F] settle the question of non-determinism for one of the simplest models of automata with counters: their non-determinism is substantially stronger than any model of unambiguous or GFG machines.

The papers [B] and [C] provide new solutions to Problem 4.2, based on game methods and the related de-alternation theorems. The results of [B] indicate to a similar analogy to the one resulting from [F]: Borel languages correspond to partial power of non-determinism, while the languages of topological complexity equal to $\mathbf{\Sigma}^1_1$ reflect its full strength. Finally, the work [E] gives a final answer to the previously-studied variant of Problem 4.2: how complicated are unambiguous languages among all regular languages of infinite trees.

The proof techniques of these results are based on the tools of combinatorics, topology, and game theory. A special role is played by the games with ω-regular winning conditions, that are used to characterise certain properties ([A], [B], [C], [E]). Also, the importance of parity signatures is emphasised in the process of choosing an *optimal* witness ([A], [E]) and in estimations about the complexity of a given language ([B]). An essential role is played by the concept of the topological hardness of a given language ([B], [C], [E], [F], and implicitly in [D]).

Although many questions about non-determinism are still open, this scientific achievement delivered final answers to several important, previously studied problems: Theorem 4.7 provides precise estimations on the cost of determinisation of GFG automata; Theorem 4.13 settles the strength of non-determinism of BMC automata with one blind counter; finally, Theorem 4.28 allows us to understand the complexity of unambiguous languages among all regular tree languages.

5 Other publications

The results constituting the above scientific achievement fit into a wider context of previous research on these topics. In particular, the following papers of my co-authorship, although not included in my scientific achievement, are strongly related to it and were already mentioned above:

[BKKS13] Udi Boker, Denis Kuperberg, Orna Kupferman, and Michał Skrzypczak. Nondeterminism in the presence of a diverse or unknown future. In *ICALP (2)*, pages 89–100, 2013.

This paper proves that the class of GFG automata coincides with the class of Good-For-Trees automata, but is strictly greater than the class of DBP automata.

[BS13] Marcin Bilkowski and Michał Skrzypczak. Unambiguity and uniformization problems on infinite trees. In *CSL*, volume 23 of *LIPIcs*, pages 81–100, 2013.

This paper introduces a new conjecture regarding the possibility of defining in MSO a choice function over the so-called *thin trees*. Other, equivalent statements of this conjecture are presented. Additionally, based on that conjecture, an effective characterisation is given for the class of bi-unambiguous languages of infinite binary trees. Additionally, the paper provides an essentially new example of an ambiguous regular language of infinite trees.

[FS14] Olivier Finkel and Michał Skrzypczak. On the topological complexity of ω-languages of non-deterministic Petri nets. *Information Processing Letters*, 114(5):229–233, 2014.

The main result of that paper is Theorem 4.12 cited above.

[MS16] Henryk Michalewski and Michał Skrzypczak. Unambiguous Büchi is weak. In *DLT*, pages 319–331, 2016.

The main result of that paper is Item a) of Theorem 4.25 above.

[MS18] Henryk Michalewski and Michał Skrzypczak. On the strength of unambiguous tree automata. *International Journal of Foundations of Computer Science*, 29(5):911–933, 2018.

This is a journal version of [MS16], extending the above results by Item b) of Theorem 4.25 and a new separation procedure analogous to [AS05], for non-deterministic weak parity automata.

Additionally, multiple results discussed in the former part of this document have been included in my PhD thesis. The thesis has been published in 2016 in the form of a book:

[Skr16] Michał Skrzypczak. *Descriptive Set Theoretic Methods in Automata Theory – Decidability and Topological Complexity*, volume 9802 of *Lecture Notes in Computer Science*. Springer, 2016.

5.1 Measure and category properties of regular languages

One of the active themes of my research was the analysis of properties motivated by topology and measure theory, in the context of regular languages of infinite words and trees. The following papers follow this line of research:

[BNR+10] Mikołaj Bojańczyk, Damian Niwiński, Alexander Rabinovich, Adam Radziwończyk-Syta, and Michał Skrzypczak. On the Borel complexity of MSO definable sets of branches. *Fundamenta Informaticae*, 98(4):337–349, 2010.

The aim of this paper is to estimate, what is the topological complexity of sets of branches of an infinite tree, that can be defined based on some monadic parameters by an MSO formula (or equivalently by a non-deterministic parity automaton). As it turns out, those sets are exactly Boolean combinations of $\mathbf{\Sigma}^0_2$-sets, i.e. sets of the same topological complexity as parity conditions.

[Skr13] Michał Skrzypczak. Topological extension of parity automata. *Information and Computation*, 228:16–27, 2013.

This paper further extends the results of [BNR+10] by a precise analysis of the interplay between parity conditions and Boolean combinations of $\mathbf{\Sigma}^0_2$-sets. The ultimate result of that work is a characterisation of the families of sets from the first three levels of Borel hierarchy (up to $\mathbf{\Sigma}^0_3$), based on appropriately chosen variants of parity conditions.

[GMMS14] Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of game tree languages. In *MFCS*, pages 303–314, 2014.

This paper proves that all regular languages of infinite trees are universally measurable. The crucial ingredient of the proof is a new relationship between the languages $W_{i,j}$ and the class of the so-called $\mathcal{R}$-sets introduced by Kolmogorov [Kol28].

[GMMS17] Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of regular sets of trees. *Information and Computation*, 256:108–130, 2017.

This is an extended journal version of [GMMS14]. It features complete versions of all the proofs. Additionally, this article proves that the stratification of the languages $W_{i,j}$ based on an appropriately chosen parity signatures, is continuous with respect to the measure. Together with the measurability, these results allow eliminating the use of Martin's axiom from the results regarding game semantics of the probabilistic μ-calculus [Mio12].

[MSM18] Matteo Mio, Michał Skrzypczak, and Henryk Michalewski. Monadic second order logic with measure and category quantifiers. *Logical Methods in Computer Science*, 14(2):1–29, 2018.

This work introduces and studies certain new quantifiers expressing *genericness* of a given set or branch. The two considered variants of *genericness* are based on the measure (a set of full measure) and category (the complement of a meagre set in the sense of Baire).

5.2 Game automata and branching games

Another topic of my research is related to the concept of game automata and branching games. A game automaton can be seen as a self-dualisation of a deterministic automaton over infinite trees: each transition of such an automaton is either universal (and sends to states to the two children of a given node), or existential (and non-deterministically chooses the direction, where to send the consecutive state). The following series of works provides solutions to all the index problems for this class of automata. Additionally, an effective characterisation of the family of languages recognisable by game automata is given:

[FMS13] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Rabin-Mostowski index problem: A step beyond deterministic automata. In *LICS*, pages 499–508, 2013.

[FMS15] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. On the weak index problem for game automata. In *WoLLIC*, pages 93–108, 2015.

[FMS16] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Index problems for game automata. *ACM Transactions on Computational Logic*, 17(4):24:1–24:38, 2016.

The topic of *branching games* is partially related to game automata. A *branching game* is a two-player game where a single play takes the form of an infinite tree. Due to the fact that the players have only partial information about the play, those games need not be determined. The following paper studies the level of determinacy of these games, depending on the complexity of the winning condition and the considered family of strategies. Additionally, the decidability of the problem of computing the value of such a game is studied.

[PS16] Marcin Przybyłko and Michał Skrzypczak. On the complexity of branching games with regular conditions. In *MFCS*, pages 78:1–78:14, 2016.

5.3 Decidability of MSO

The following two papers provide a rather complete analysis of the relationships between MSO logic and automata in the context of the so-called *thin trees*. A partial infinite tree is *thin* if it has only countably many infinite branches. It turns out that not only MSO theory of thin trees is decidable, but these structures constitute a natural intermediate step between infinite words and trees.

[BIS13] Mikołaj Bojańczyk, Tomasz Idziaszek, and Michał Skrzypczak. Regular languages of thin trees. In *STACS*, volume 20 of *LIPIcs*, pages 562–573, 2013.

[ISB16] Tomasz Idziaszek, Michał Skrzypczak, and Mikołaj Bojańczyk. Regular languages of thin trees. *Theory of Computing Systems*, 58(4):614–663, 2016.

This is an extended journal version of [BIS13].

A separate thread of research was motivated by the results of [KM16]. Following this thread, we studied the axiomatic power needed to prove Büchi's decidability theorem [Büc62], that states decidability of MSO over infinite words. The main results of this paper show, that decidability of MSO over infinite words is essentially equivalent to the principle of induction for Σ^0_2-formulae.

[KMPS16] Leszek Aleksander Kołodziejczyk, Henryk Michalewski, Pierre Pradic, and Michał Skrzypczak. The logical strength of Büchi's decidability theorem. In *CSL*, pages 36:1–36:16, 2016.

5.4 Quantitative extensions of regularity

The last group of my papers is focused on models extending regular languages by some quantitative properties. The first group of these results is based on extensions expressing boundedness of some events: the U quantifier (introduced in [Boj04]) and ωB-, ωS-, and ωBS-automata (studied in [BC06]).

[HST10] Szczepan Hummel, Michał Skrzypczak, and Szymon Toruńczyk. On the topological complexity of MSO+U and related automata models. In *MFCS*, pages 429–440, 2010.

This paper studies the expressive power of MSO logic extended by the *unboundedness quantifier* U. One of the main results states that the extended logic (denoted MSO+U) can define non-Borel languages. It shows that no Borel model of non-deterministic automata can catch the full expressive power of MSO+U. Additionally, the paper provides tight bounds on the topological complexity of non-deterministic automata with the conditions ωB, ωS, and ωBS. Finally, it features a family of languages of increasing topological complexity, that can be recognised by alternating ωBS-automata.

[HS12] Szczepan Hummel and Michał Skrzypczak. The topological complexity of MSO+U and related automata models. *Fundamenta Informaticae*, 119(1):87–111, 2012.

This is an extended journal version of [HST10]. It provides a development of the construction of a non-Borel language definable in MSO+U into a family of languages that climb into all finite levels of the projective hierarchy. These examples allow us to exclude the possibility of catching MSO+U by any Borel model of alternating automata.

[BGMS14] Mikołaj Bojańczyk, Tomasz Gogacz, Henryk Michalewski, and Michał Skrzypczak. On the decidability of MSO+U on infinite trees. In *ICALP (2)*, pages 50–61, 2014.

The main result of this paper shows that assuming Gödels axiom [Göd39] of constructible universe (denoted V=L), MSO+U logic is undecidable over infinite trees. The proof of this fact is based on the topologically hard languages from [HS12] and the combinatorial methods of [She75] that were used to prove undecidability of MSO($\mathbb{R}$). Some time after the publication of this paper, a direct proof of undecidability of MSO+U over infinite words was found [BPT16].

[Skr14b] Michał Skrzypczak. Separation property for wB- and wS-regular languages. *Logical Methods in Computer Science*, 10(1):1–20, 2014.

This paper studies the problems of separation and regularity of languages of infinite words recognisable by ωB-and ωS-automata. The main result states that every two disjoint languages recognisable by ωB-automata (resp. ωS-automata) can be separated by a regular language.

[FHKS15] Nathanaël Fijalkow, Florian Horn, Denis Kuperberg, and Michał Skrzypczak. Trading bounds for memory in games with counters. In *ICALP (2)*, pages 197–208, 2015.

This work studies the issue of finite memory determinacy for a family of parity games with the *boundedness condition B*. The main results show that such determinacy does not hold in general, however, it holds in the case of arenas based on thin trees.

Another quantitative extension of regularity that I have studied are probabilistic automata. The following short note provides a simple example of a non-regular language of finite words that can be recognised by such automata.

[FS15] Nathanaël Fijalkow and Michał Skrzypczak. Irregular behaviours for probabilistic automata. In *RP*, pages 33–36, 2015.

References

[AS05] André Arnold and Luigi Santocanale. Ambiguous classes in μ-calculi hierarchies. *Theoretical Computer Science*, 333(1–2):265–296, 2005.

[BC06] Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In *LICS*, pages 285–296, 2006.

[BCPS18] Mikołaj Bojańczyk, Filippo Cavallari, Thomas Place, and Michał Skrzypczak. Regular tree languages in low levels of Wadge hierarchy. *CoRR*, abs/1806.02041, 2018. Submitted to a journal.

[BGMS14] Mikołaj Bojańczyk, Tomasz Gogacz, Henryk Michalewski, and Michał Skrzypczak. On the decidability of MSO+U on infinite trees. In *ICALP (2)*, pages 50–61, 2014.

[BIS13] Mikołaj Bojańczyk, Tomasz Idziaszek, and Michał Skrzypczak. Regular languages of thin trees. In *STACS*, volume 20 of *LIPIcs*, pages 562–573, 2013.

[BKKS13] Udi Boker, Denis Kuperberg, Orna Kupferman, and Michał Skrzypczak. Nondeterminism in the presence of a diverse or unknown future. In *ICALP (2)*, pages 89–100, 2013.

[BNR+10] Mikołaj Bojańczyk, Damian Niwiński, Alexander Rabinovich, Adam Radziwończyk-Syta, and Michał Skrzypczak. On the Borel complexity of MSO definable sets of branches. *Fundamenta Informaticae*, 98(4):337–349, 2010.

[Boj04] Mikołaj Bojańczyk. A bounding quantifier. In *CSL*, pages 41–55, 2004.

[BPT16] Mikołaj Bojańczyk, Paweł Parys, and Szymon Toruńczyk. The MSO+U theory of (N, <) is undecidable. In *STACS*, pages 21:1–21:8, 2016.

[BS13] Marcin Bilkowski and Michał Skrzypczak. Unambiguity and uniformization problems on infinite trees. In *CSL*, volume 23 of *LIPIcs*, pages 81–100, 2013.

[Büc62] Julius Richard Büchi. On a decision method in restricted second-order arithmetic. In *Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.

[Cav18] Filippo Cavallari. *Regular tree languages in the first two levels of the Borel hierarchy*. PhD thesis, HEC Lausanne (UNIL) and University of Turin, June 2018.

[CKLV13] Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. Deciding the weak definability of Büchi definable tree languages. In *CSL*, pages 215–230, 2013.

[CL10] Thomas Colcombet and Christof Löding. Regular cost functions over finite trees. In *LICS*, pages 70–79, 2010.

[CLNW10] Arnaud Carayol, Christof Löding, Damian Niwiński, and Igor Walukiewicz. Choice functions and well-orderings over the infinite binary tree. *Central European Journal of Mathematics*, 8:662–682, 2010.

[Col13] Thomas Colcombet. Fonctions régulières de coût. Habilitation thesis, Université Paris Diderot—Paris 7, 2013.

[DFH15] Jacques Duparc, Kevin Fournier, and Szczepan Hummel. On unambiguous regular tree languages of index (0, 2). In *CSL*, pages 534–548, 2015.

[DM07] Jacques Duparc and Filip Murlak. On the topological complexity of weakly recognizable tree languages. In *FCT*, pages 261–273, 2007.

[FHKS15] Nathanaël Fijalkow, Florian Horn, Denis Kuperberg, and Michał Skrzypczak. Trading bounds for memory in games with counters. In *ICALP (2)*, pages 197–208, 2015.

[FM14] Alessandro Facchini and Henryk Michalewski. Deciding the Borel complexity of regular tree languages. In *CiE*, pages 163–172, 2014.

[FMS13] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Rabin-Mostowski index problem: A step beyond deterministic automata. In *LICS*, pages 499–508, 2013.

[FMS15] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. On the weak index problem for game automata. In *WoLLIC*, pages 93–108, 2015.

[FMS16] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Index problems for game automata. *ACM Transactions on Computational Logic*, 17(4):24:1–24:38, 2016.

[FS09] Olivier Finkel and Pierre Simonnet. On recognizable tree languages beyond the Borel hierarchy. *Fundamemta Informaticae*, 95(2–3):287–303, 2009.

[FS14] Olivier Finkel and Michał Skrzypczak. On the topological complexity of ω-languages of non-deterministic Petri nets. *Information Processing Letters*, 114(5):229–233, 2014.

[FS15] Nathanaël Fijalkow and Michał Skrzypczak. Irregular behaviours for probabilistic automata. In *RP*, pages 33–36, 2015.

[GMMS14] Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of game tree languages. In *MFCS*, pages 303–314, 2014.

[GMMS17] Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of regular sets of trees. *Information and Computation*, 256:108–130, 2017.

[Göd39] Kurt Gödel. Consistency-Proof for the Generalized Continuum-Hypothesis. *Proceedings of the National Academy of Sciences of the United States of America*, 25(4):220–224, 1939.

[HP06] Thomas A. Henzinger and Nir Piterman. Solving games without determinization. In *CSL*, LNCS, pages 395–410. Springer, 2006.

[HS12] Szczepan Hummel and Michał Skrzypczak. The topological complexity of MSO+U and related automata models. *Fundamenta Informaticae*, 119(1):87–111, 2012.

[HST10] Szczepan Hummel, Michał Skrzypczak, and Szymon Toruńczyk. On the topological complexity of MSO+U and related automata models. In *MFCS*, pages 429–440, 2010.

[Hum12] Szczepan Hummel. Unambiguous tree languages are topologically harder than deterministic ones. In *GandALF*, pages 247–260, 2012.

[ISB16] Tomasz Idziaszek, Michał Skrzypczak, and Mikołaj Bojańczyk. Regular languages of thin trees. *Theory of Computing Systems*, 58(4):614–663, 2016.

[JL02] David Janin and Giacomo Lenzi. On the logical definability of topologically closed recognizable languages of infinite trees. *Computers and Artificial Intelligence*, 21(3), 2002.

[Kec95] Alexander Kechris. *Classical descriptive set theory.* Springer-Verlag, New York, 1995.

[KM16] Leszek Aleksander Kołodziejczyk and Henryk Michalewski. How unprovable is Rabin's decidability theorem? In *LICS*, pages 788–797, 2016.

[KMM06] Orna Kupferman, Gila Morgenstern, and Aniello Murano. Typeness for ω-regular automata. *IJFCS*, 17(4):869–884, 2006.

[KMPS16] Leszek Aleksander Kołodziejczyk, Henryk Michalewski, Pierre Pradic, and Michał Skrzypczak. The logical strength of Büchi's decidability theorem. In *CSL*, pages 36:1–36:16, 2016.

[Kol28] Andrey Nikolaevich Kolmogorov. Operations sur des ensembles (in Russian, summary in French). *Matematicheskii Sbornik*, 35:415–422, 1928.

[KPB94] Sriram C. Krishnan, Anuj Puri, and Robert K. Brayton. Deterministic ω-automata vis-a-vis deterministic Büchi automata. In *Algorithms and Computations*, volume 834 of *LNCS*, pages 378–386. Springer, 1994.

[Mio12] Matteo Mio. On the equivalence of game and denotational semantics for the probabilistic mu-calculus. *Logical Methods in Computer Science*, 8(2), 2012.

[Mos91] Andrzej W. Mostowski. Hierarchies of weak automata and weak monadic formulas. *Theoretical Computer Science*, 83(2):323–335, 1991.

[MS95] David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141(1&2):69–107, 1995.

[MS16] Henryk Michalewski and Michał Skrzypczak. Unambiguous Büchi is weak. In *DLT*, pages 319–331, 2016.

[MS18] Henryk Michalewski and Michał Skrzypczak. On the strength of unambiguous tree automata. *International Journal of Foundations of Computer Science*, 29(5):911–933, 2018.

[MSM18] Matteo Mio, Michał Skrzypczak, and Henryk Michalewski. Monadic second order logic with measure and category quantifiers. *Logical Methods in Computer Science*, 14(2):1–29, 2018.

[NW96] Damian Niwiński and Igor Walukiewicz. Ambiguity problem for automata on infinite trees. unpublished, 1996.

[PS16] Marcin Przybyłko and Michał Skrzypczak. On the complexity of branching games with regular conditions. In *MFCS*, pages 78:1–78:14, 2016.

[Rab70] Michael Oser Rabin. Weakly definable relations and special automata. In *Proceedings of the Symposium on Mathematical Logic and Foundations of Set Theory*, pages 1–23. North-Holland, 1970.

[SE89] Robert S. Streett and E. Allen Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81(3):249–264, 1989.

[She75] Saharon Shelah. The monadic theory of order. *The Annals of Mathematics*, 102(3):379–419, 1975.

[SI85] Richard Edwin Stearns and Harry B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM Journal on Computing*, 14(3):598–611, 1985.

[Skr13] Michał Skrzypczak. Topological extension of parity automata. *Information and Computation*, 228:16–27, 2013.

[Skr14a] Michał Skrzypczak. *Descriptive set theoretic methods in automata theory.* PhD thesis, University of Warsaw, 2014.

[Skr14b] Michał Skrzypczak. Separation property for wB- and wS-regular languages. *Logical Methods in Computer Science*, 10(1):1–20, 2014.

[Skr16] Michał Skrzypczak. *Descriptive Set Theoretic Methods in Automata Theory – Decidability and Topological Complexity*, volume 9802 of *Lecture Notes in Computer Science*. Springer, 2016.

[Sku93] Jerzy Skurczyński. The Borel hierarchy is infinite in the class of regular sets of trees. *Theoretical Computer Science*, 112(2):413–418, 1993.

[SM73] Larry Stockmeyer and Albert R. Meyer. Word problems requiring exponential time. In *ACM Symposium on Theory of Computing*, pages 1–9, 1973.

[Tho96] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1996.

[Wal96] Igor Walukiewicz. Pushdown processes: Games and model checking. In Rajeev Alur and Thomas A. Henzinger, editors, *Computer Aided Verification*, pages 62–74. Springer, 1996.

[Wal02] Igor Walukiewicz. Deciding low levels of tree-automata hierarchy. In *WoLLIC*, volume 67, pages 61–75, 2002.

....
(signature of the candidate)