Introductory Talks — 5 minutes + 2 minutes of discussion
Invited Talks — 2 x 30 minutes + discussion during and after
Contributed Talks — 20 minutes + up to 10 minutes of discussion

Monday

07:30 - 08:45	Breakfast
09:45 - 10:15	8 Introductory Talks
11:00 - 12:00	8 Introductory Talks
12:15 - 13:15	Lunch
14:00 - 15:10	10 Introductory Talks
15:00 - 16:00	Coffee and Cake
16:30 - 17:30	8 Introductory Talks
18:00 - 19:00	Dinner

Tuesday

07:30 - 08:45	Breakfast
09:15 - 09:45	Invited Talk: Denis Kuperberg, part 1
10:00 - 10:30	Invited Talk: Denis Kuperberg, part 2
10:30 - 11:00	Short break
11:00 - 11:30	Contributed Talk: Simon Jantsch
11:30 - 12:00	Contributed Talk: Dmitry Chistikov
12:15 - 13:15	Lunch
14:00 - 14:30	Contributed Talk: Udi Boker
14:30 - 14:45	Contributed Talk: Anca Muscholl
15:00 - 16:00	Coffee and Cake
16:00 - 16:30	Contributed Talk: Mahsa Shirmohammadi
16:30 - 17:00	Contributed Talk: Lorenzo Clemente
17:00 - 17:30	Contributed Talk: Michael Raskin
17:30 - 18:00	Contributed Talk: Udi Boker (continued)
18:00 - 19:00	Dinner

Wednesday

07:30 - 08:45	Breakfast
09:15 - 09:45	Invited Talk: Gabriele Puppis, part 1
10:00 - 10:30	Invited Talk: Gabriele Puppis, part 2
10:30 - 11:00	Short break
11:00 - 11:30	Contributed Talk: Pierre Ohlmann
12:15 - 13:15	Lunch
14:00 -	Social Activity
18:00 - 19:00	Dinner

Thursday

07:30 - 08:45	Breakfast
09:15 - 09:45	Invited Talk: Wojtek Czerwiński, part 1
10:00 - 10:30	Invited Talk: Wojtek Czerwiński, part 2
10:30 - 11:00	Short break
11:00 - 11:30	Contributed Talk: Sylvain Lombardy
11:30 - 12:00	Contributed Talk: Sarah Winter
12:15 - 13:15	Lunch
14:00 - 14:30	Contributed Talk: Achim Blumensath
14:30 - 15:00	Contributed Talk: Alex Rabinovich
15:00 - 16:00	Coffee and Cake
16:00 - 18:00	Work in Subgroups
18:00 - 19:00	Dinner

Friday

07:30 - 08:45	Breakfast
09:15 - 09:30	Closing Ceremony
10:00 - 11:00	Work in Subgroups
12:15 - 13:15	Lunch

1 Open Problems

- Georg Zetzsche: Problems that are undecidable for non-deterministic machines, e.g. universality, regularity and other subclasses (e.g. given a counter language, how many counters does it require?)
- Emmanuel Filiot: Program synthesis (unambiguous)
- Denis Kuperberg: G_2 conjecture (he explains in the invited talk)
- Denis Kuperberg: Complexity of the following problem: Input NFA A and n in binary, Output: Is there DFA for $L(A)$ with $\leq n$ states (From Shaull Almagor)
- Antoine Mottet: Say that an operation f of finite arity over the set of datawords preserves a language L if $f(L, , L)$ is a subset of L. For example, if L is recognizable by a register automaton with an atom structure A, then every automorphism of A preserves L. To my knowledge, the internal closure properties of data languages have not been considered so far. In particular, can one understand the complexity of a language (i.e., deterministically recognizable, unambiguously recognizable, recognizable, with/without guessing) in terms of the operations preserving a language? This question was answered positively for Turing machines (recognizing several variants of constraint satisfaction problems) where closure properties have been central in characterizing the (descriptive) complexity of problems.
- Lorenzo Clemente: Various versions of the zeroness problem, namely for weighted grammars over a field, unary polynomial automata, weighted Parikh automata, weighted VASS (Complexity and Decidability)
- Guillermo Perez: regarding Containment problem for probabilistic automata with bounded ambiguity, can one prove decidability without having to assume Schanuel's conjecture.
- Simon Jantsch has three questions regarding state complexity of Unambiguous Büchi automata (I guess this could take some time to explain it) [[see additional file]]
- Karin Quaas: Universality Unambiguous Register Automata over $(N; <)$, $(Z; <)$?
- Karin Quaas: how to prove lower bounds for universality unambiguous Register Automata?
- Nathanael Fijalkow [on tablet, online]: decompositions of finitely ambiguous into finitely many unambiguous weighted automata
- Michael Raskin: It is now known that complementing the language an n-state UFA might yield a language not recognisable by NFAs with fewer than $n^{(\log\log\log n)^{\Omega(1)}}$ for unary alphabet and there is an upper bound of $n^{O(\log n)}$. In binary case the lower bound is $n^{\Omega(logn)}$ but the upper bound is still exponential; same for large alphabets. How do we close the non-unary complement gap?

2 Invited Talks

Denis Kuperberg on Tuesday

Good-for-Games automata: state of the art and perspectives

In the setting of regular languages of infinite words, Good-for-Games (GFG) automata can be seen as an intermediate formalism between determinism and nondeterminism, with advantages from both worlds. Indeed, like deterministic automata, GFG automata enjoy good compositional properties (useful for solving games and composing automata and trees) and easy inclusion checks. Like nondeterministic automata, they can be exponentially more succinct than deterministic automata. Since their introduction in 2006 by Henzinger and Piterman, there has been a steady research effort to uncover the properties of GFG automata, with some surprises along the way. I will give an overview of the results obtained in this line of research, the proof techniques typically used, and the remaining open problems and conjectures.

Gabriele Puppis on Wednesday

Unambiguous automata for data languages

I will present the status of an ongoing research work with Thomas Colcombet and Micha Skrzypczak about unambiguity in register automata (register automata, or finite memory automata, are automata that can describe languages over an infinite alphabet).

Differently from finite state automata, the amount of non-determinism allowed in register automata has an impact on the expressive power and the closure properties of the recognized class of languages, as well as on the complexity of some fundamental decision problems. For example, deterministic register automata are strictly less expressive than non-deterministic ones, they are closed under complement, but not under mirroring. On the other hand, non-deterministic register automata (with guessing) are closed under mirroring, but not under complement.

It comes natural then to study the intermediate class of unambiguous register automata with guessing. Recently (LICS'21), this class has been shown to enjoy a decidable equivalence problem and is believed to be effectively closed under complement. However, proving this closure property turned out to be more difficult than expected. I will present some ideas and partial results along this goal, mentioning a few other conjectures related to the expressive power of unambiguous register automata.

Wojtek Czerwiński on Thursday

On future-determinization of unambiguous systems

I will present you a result based on an on-going work jointed with Piotr Hofman. We have shown that language equivalence is decidable for unambiguous vector addition systems with states (VASS) (acceptance is by state). Id like to focus more on our technique: we have proven that each unambiguous VASS can be determinized in a certain sense (with a use of some additional information about the future), which we call future-determinization. This result makes use of some known regular-separability results. There is a hope that similar techniques can be possible for other unambiguous systems and maybe even point to some high-level connection between separability and unambiguity notions.

3 Contributed Talks

Simon Jantsch on Tuesday

Alternation as a tool for disambiguation

In this talk we show how alternating automata can be used as a tool to devise disambiguation algorithms for nondeterministic automata over finite and infinite words.

The main idea is to use conjunction and complementation, both of which can be naturally implemented in alternating automata, to restrict nondeterministic branching in a way that preserves the language and makes sure that for any given word only one choice leads to acceptance.

A notion of unambiguity for alternating automata is introduced, and we show that standard alternation removal techniques preserve it.

The approach works well for automata on finite words and restricted forms of -automata (namely very weak ones) but we show that it fails for arbitrary nondeterministic Bchi automata (NBA), and discuss the issues that arise.

Finally, we speculate about the relationship between complementation and disambiguation and possible consequences for the state complexity of disambiguating NBA.

Dmitry Chistikov on Tuesday

Unambiguous automata acceptance?

Given an NFA with m transitions and an input word of length l, one can decide in time O(m l) if the word is accepted. If m is close to n^2 (where n is the number of states) and l is close to n, this running time is essentially cubic in n. I don't know if significantly faster algorithms exist for this and several related problems. Can we obtain speed-ups if the automaton is known to be unambiguous?

Udi Boker on Tuesday

Between Deterministic and Nondeterministic Quantitative Automata

There is a challenging trade-off between deterministic and nondeterministic automata, where the former suit various applications better, however at the cost of being exponentially larger or even less expressive.

This gave birth to many notions in between determinism and nondeterminism, aiming at enjoying, sometimes, the best of both worlds. Some of the notions are yes/no ones, for example initial nondeterminism (restricting nondeterminism to allowing several initial states), and some provide a measure of nondeterminism, for example the ambiguity level.

We analyze the possible generalization of such notions from Boolean to quantitative automata, and suggest that it depends on the following key characteristics of the considered notion N—whether it is syntactic or semantic, and if semantic, whether it is word-based or language-based.

A syntactic notion, such as initial nondeterminism, applies as is to a quantitative automaton A, namely N(A). A word-based semantic notion, such as unambiguity, applies as is to a Boolean automaton t-A that is derived from A by accompanying it with some threshold value t, namely N(t-A). A language-based notion, such as history determinism, also applies as is to A, while in addition, it naturally generalizes into two different notions with respect to A itself, by either: i) taking the supremum of N(t-A)over all thresholds t, denoted by Th-N(A); or ii) generalizing the basis of the notion from a language to a function, denoted simply by N(A). While in general N(A) implies Th-N(A) implies N(t-A), we have for some notions that N(A) and Th-N(A) are equivalent and for some not. (For measure notions, "implies" stands for ¿= with respect to the nondeterminism level.)

We classify numerous notions known in the Boolean setting according to their characterization above, generalize them to the quantitative setting and look into relations between them. The generalized notions open new research directions with respect to quantitative automata, and provide insights on the original notions with respect to Boolean automata.

Anca Muscholl on Tuesday

Active learning sound negotiations

Sound deterministic negotiations are models of distributed systems, a kind of Petri nets or Zielonka automata with additional structure. We show that the additional structure allows to minimize such negotiations. Based on minimisation we present two Angluin-style learning algorithms for sound deterministic negotiations. The two algorithms differ in the kind of membership queries they use, and both have similar (polynomial) complexity as Angluins algorithm.

Joint work with Igor Walukiewicz.

Michael Raskin on Tuesday

State complexity of complementing unambiguous automata

Not so long ago, even a polynomial upper bound on state complexity of recognising the complement of the language of an unambiguous finite automaton felt plausible. Now it doesn't, but what else do we know? Not so much.

In this talk I plan to briefly show the approaches that give the best currently known lower and upper bounds for the state complexity of complementation in the unary and binary alphabets; and draw a (straightforward) game reformulation of the large-alphabet problem in the hope it will inspire someone to prove the exponential lower bound in that case.

Lozenzo Clemente, Mahsa Shirmohammadi on Wednesday

TBA

TBA

Pierre Ohlmann on Wednesday

TBA

TBA

Sylvain Lombardy on Thursday

Quotients, Coverings and Conjugacy of Unambiguous Automata

In this talk, I shall recall the definitions of quotients and coverings, which are useful tools to transform the structure of an automaton while preserving the unambiguity. We shall see that it is always possible to turn an unambiguous automaton to any equivalent one using these tools. The construction of this transformation is based on a more algebraic concept, that is the conjugacy of automata. An open question concerning the transformation of an automaton to another one is the state complexity of the transitional automata. This talk is based on a work with Marie-Pierre Bal and Jacques Sakarovitch.

Sarah Winter on Thursday

Unambiguity in Transducer Theory

This talk surveys some introductory results regarding unambiguity in transducer theory. Transducers are automata with output; they recognize relations. A transducer is unambiguous if for each word u from its domain there is a unique accepting run with input u.

In more detail, we show that the classes of functions recognized by functional transducers and unambiguous transducers coincide. We also show that unambiguity, while necessary for one-way transducers, can be traded for determinism at the price of two-wayness.

This is a joint work with Emmanuel Filliot.

Achim Blumensath on Thursday

Regular Tree Algebras

We introduce a class of algebras that can be used as recognisers for regular tree languages. We show that it is the only such class that forms a pseudo-variety and we prove the existence of syntactic algebras.

Alex Rabinovich on Thursday

On Uniformization in the Full Binary Tree

TBA

4 Topics to be Discussed in the Seminar

In the following, we discuss some of the most important topics regarding *unambiguity in automata theory* that we aim to work on during the Dagstuhl seminar.

A general goal of the seminar is to bring together experts from different fields of automata theory, to stimulate an exchange of recent results and new proof techniques. We start by a quick review of areas in automata theory that lately have seen new insights regarding unambiguity, and we state related open problems that we plan to work on.

State Complexity As mentioned in the introduction, unambiguous finite automata can offer faster algorithms than the ones working with general nondeterministic automata. However, this speed-up can be lost in the transformation from a nondeterministic automaton to an unambiguous one. Therefore, it is very important to understand how the number of states of the resulting automaton depends on the number of states of the automaton we start with. This quantity is known as the *state complexity* of the given transformation. Although the state complexity of standard operations on deterministic and nondeterministic automata is rather well understood, certain problems about unambiguous automata are still open. One of the most intriguing among them is the following.

What is the state complexity of the complementation of unambiguous automata: find an optimal upper bound $f(n)$ for the number of states of an unambiguous automaton $\mathcal{A}'$ that accepts the complement of the language accepted by a given unambiguous automaton $\mathcal{A}$ with n states.

Basic observations show that the bound $f(n)$ must be between n^2 and 2^n. However, the exact growth rate of f has been open for a long time. It was widely believed [?] that $f(n)$ is polynomial in n. Recent results of Raskin [?] have disproved that conjecture, showing that $f(n) \geq n^{\ln(n)}$. However, the gap between $n^{\ln(n)}$ and 2^n is still open.

Register Automata and Timed Automata Register automata are finite automata equipped with a finite set of *register variables* ranging over some (potentially infinite) data domain. Registers can store the current input data value for later comparisons. In general, the containment problem $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is undecidable unless $\mathcal{B}$ has strictly less than two registers [?]. It was recently proved that, for register automata over the data domain $(\mathbb{N}; =)$, the containment problem becomes decidable in doubly exponential time if $\mathcal{B}$ is unambiguous [?]. However, the paper [?] leaves open whether this upper bound is optimal. Standard techniques for proving lower bounds for the containment problem for nondeterministic automata fail for unambiguous automata. One of the most urgent questions for infinite-state systems (including register automata, but also timed automata and counter automata) is:

What can be new techniques for proving lower bounds for the containment problem? A natural continuation of the positive decidability result for register automata over $(\mathbb{N}; =)$ in [?] is to study register automata over ordered domains like $(\mathbb{Q}; <)$, and, very related to that, real-timed automata. So far, only partial results for the case that at most a single variable is used, could be achieved [?], so that we would like to study the following question:

Is containment for unambiguous timed automata decidable?

A generalization of register automata are *register automata with guessing*, where registers can store arbitrary data that is not necessarily part of the input word. It was conjectured [?], but never proved, that the class of languages accepted by unambiguous register automata with guessing is closed under complement. It is also still open whether the containment problem for such automata is decidable. The decidability result in [?] only applies to the case that $\mathcal{B}$ is restricted to have a single register. These problems seem to be hard and solutions may require novel techniques.

What is the decidability status of the containment problem for unambiguous register automata with guessing? Is the class of languages accepted by unambiguous register automata with guessing closed under complement?

Unambiguous Tree Automata The concept of unambiguity appears to be quite subtle in the case of automata over infinite trees. From examples in [?, ?] we know that not every regular tree language can be recognised by an unambiguous automaton. The examples are based on the problem of definability of *choice* over trees, studied by Shelah and others [?, ?, ?]. The simplicity of the provided example (the set of trees with at least one label a) suggests that essentially any *branching* type of nondeterminism leads to ambiguity. On the other hand, it turns out to be quite demanding to actually prove that a given language of infinite trees cannot be recognised by an unambiguous automaton. Up to now, only few examples are known [?, ?].

One of the ways to better understand the limitations of unambiguity over infinite trees, is to ask about the limits of the expressive power of unambiguous automata: can unambiguous automata recognise arbitrarily complicated (in terms of parity index) languages? A series of works [?, ?, ?] on this topic lead to a number of estimations, binding the acceptance condition of the automaton and the complexity of its language. Finally, in [?], a positive answer to the above question is given: despite limited expressiveness, unambiguous automata can recognise arbitrarily complicated tree languages.

How to decide, given a representation of a regular language of infinite trees L, if there exists an unambiguous automaton recognising L?

Unambiguous Büchi Automata Unambiguous Büchi automata over ω-words form a very useful class of automata: they do not only have the same expressive power as nondeterministic Büchi automata [?, ?], but they can also be exponentially more succint than deterministic Büchi automata [?]. Interestingly, standard translations from formulas of linear temporal logic (LTL) to Büchi automata [?] yield unambiguous automata, which renders them attractive for model-checking algorithms [?, ?, ?].

Inspired by this, a PTIME algorithm for model checking Markov chains against specifications given by unambiguous separated Büchi automata was presented in [?]. This result was recently generalized [?, ?], using interesting novel techniques from linear algebra, and resulting in remarkable improvements regarding implementation. With regard to optimization, a novel translation from LTL formulas to unambiguous Büchi automata was presented in [?]. Another main subject regarding unambiguous Büchi automata is the computational complexity of the containment problem. While for finite automata, there is an improvement from PSPACE to PTIME when the input automata are unambiguous [?], it is unknown whether there exists some PTIME algorithm for the containment problem for unambiguous Büchi automata. Some positive results have been achieved for simpler types of acceptance conditions [?] and a stronger notion of ambiguity [?].

What is the precise computational complexity of the containment problem for unambiguous Büchi automata?

Probabilistic Automata Probabilistic automata are a classical automaton model introduced by Rabin [?]. Nondeterministic choices of finite automata are replaced by probabilities, that is, every transition carries a rational number corresponding to the probability to be chosen on the input of the current letter. Every input word w fed to a probabilistic automaton $\mathcal{A}$ is assigned a value denoted by $\mathcal{A}(w)$, corresponding to the probability of w being accepted by $\mathcal{A}$. Almost all natural decision problems are undecidable for probabilistic automata [?]. The most recent results concern the (probabilistic versions of the) the emptiness and the containment problem: undecidability of the problem whether $\mathcal{A}(w) \geq \frac{1}{2}$ for some input word w holds already if $\mathcal{A}$ is *linearly ambiguous* [?], but efficient algorithms exist for the case that $\mathcal{A}$ is *finitely ambiguous* [?]. Since probabilistic automata are complementable, the undecidability result for the emptiness problem for linearly ambiguous automata imply the undecidability for the containment problem (given two automata $\mathcal{A}$ and $\mathcal{B}$, does $\mathcal{A}(w) \leq \mathcal{B}(w)$ hold for all input words w?) as soon as one of the input automata is linearly ambiguous. On the other hand, the problem is decidable if $\mathcal{A}$ is finitely ambiguous and $\mathcal{B}$ is unambiguous; if $\mathcal{A}$ is unambiguous and $\mathcal{B}$ is finitely ambiguous, then the problem is *decidable subject to Schanuel's conjecture* [?], showing an interesting relation to a long open mathematical problem. The following was stated as an open problem in [?]:

What is the decidability status of the containment problem if both $\mathcal{A}$ and $\mathcal{B}$ are linearly ambiguous?

Weighted Automata / Max-Plus Automata Weighted automata are finite automata whose transitions are equipped with some weight coming from a weight structure (usually a semiring). The behaviour of a weighted automaton is a mapping from the input alphabet Σ^* to the domain of the weight structure, called *series of* $\mathcal{A}$, and it is denoted by $\|\mathcal{A}\|$.

Weighted automata over the semiring $(\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$ are also called *max-plus automata*. A lot is known about the hierarchy based on the degree of ambiguity [**?**] of max-plus automata; for instance, the class of finitely ambiguous series is a strict subclass of the class of polynomially ambiguous series [**?**]. The equivalence problem (deciding, for two given weighted automata $\mathcal{A}$ and $\mathcal{B}$, whether $\|\mathcal{A}\| = \|\mathcal{B}\|$, is decidable for finitely ambiguous [**?**], but undecidable for polynomially ambiguous [**?**] automata. For max-plus automata, the most important open decision problems is the *sequentiality problem*, that is to decide, given a max-plus automaton $\mathcal{A}$, whether the series $\|\mathcal{A}\|$ is *sequential* (that is deterministic). The problem has been solved for several subclasses of unambiguous max-plus automata [**?**, **?**, **?**, **?**]; the community is especially interested in the answer to the following question:

Is it decidable whether for a given max-plus automaton $\mathcal{A}$, the mapping $\|A\|$ is polynomially ambiguous? We

would also like to focus on establishing new findings concerning the size of the constructions used in the proofs of the above mentioned results, as these are currently not considered in the literature.