# AUTOMATA COLUMN

MIKOŁAJ BOJAŃCZYK, University of Warsaw
bojan@mimuw.edu.pl

## Some Open Problems in Automata and Logic

The list in this paper is, of course, a personal selection of open problems that are connected to both automata and logic. The problems are listed in no particular order.

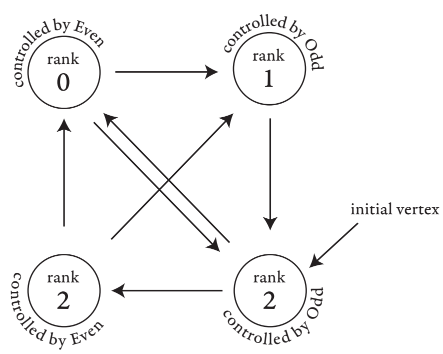
### 1. CAN PARITY GAMES BE SOLVED IN POLYNOMIAL TIME?

A parity game is a two player game, of infinite duration and with perfect information, which comes up naturally when studying automata and logics for infinite trees. Model checking of the $\mu$-calculus or testing emptiness of a parity tree automaton are problems that are polynomial time equivalent to solving parity games.

A parity game is played by two players, call them Even and Odd. The goal of player Even is to see small even numbers, the goal of player Odd will is to avoid this. The game is specified by:

— a finite directed graph, called the *arena*, such that every vertex has an outgoing edge;
— a partition of vertices in the arena, into vertices controlled by players Even and Odd;
— a *ranking function* which maps vertices in the arena to natural numbers;
— a distinguished initial vertex.

Here is an example of a parity game:




The game is played as follows. The game begins in the initial vertex. The player who controls the initial vertex chooses an outgoing edge. The player who controls the target of this edge chooses an edge leaving the target, and so on ad infinitum, until an infinite path in the graph is formed. The objective of player Even is to make sure that on this path, the minimal value of the ranking function that is seen infinitely often is even (this objective is called the parity condition).

*Determinacy*. From the Martin Determinacy theorem it follows that every parity game is determined, which means that either player Even has a winning strategy (no matter what player Odd does, the parity condition is satisfied), or player Odd has a winning strategy (no matter what player Even does, the parity condition is violated). The algorithmic challenge is to decide which one is the case.

**The open problem.** Is there a polynomial time algorithm for solving parity games, i.e. determining which of the players has a winning strategy?

*Memoryless determinacy.* As shown in [Emerson and Jutla 1991] and [Mostowski 1991], parity games are not just determined, but even memoryless determined, i.e. if one of the players has winning strategy, then that player has a *memoryless* winning strategy, where the choice of next vertex depends only on the current vertex, and not on the entire history of the vertices visited before. For instance, in the game pictured above, player Even wins by using the following memoryless strategy: when in the southwest corner go north, and when in the the northwest corner go east (southeast would also be a good choice).

Memoryless determinacy leads to an NP algorithm for deciding if player Even has a winning strategy: guess a memoryless strategy for player Even, and then check in polynomial time if every path that is consistent with this strategy satisfies the parity condition. The same kind of algorithm works for player Odd, and therefore the problem is in NP $\cap$ coNP, and yet it is not known to be in P.

*Known algorithms.* As the statement of the open problem implies, there is no known algorithm for solving parity games in polynomial time. Two examples of known algorithms for solving parity games are: a divide and conquer algorithm [Zielonka 1998], and the strategy improvement algorithm [Vöge and Jurdziński 2000]. These algorithms have exponential worst case complexities, e.g. [Friedmann 2009] provides a lower bound for the strategy improvement algorithm. Interestingly, the insights obtained from analyzing the existing parity game algorithms can be used to get lower bounds for variants of the simplex algorithm in linear programming [Friedmann et al. 2011].

*Fixed parameter tractability.* Before solving parity games in polynomial time, one could at least try to show that the problem is fixed parameter tractable, for some choice of parameter. It is known that parity games are fixed parameter tractable for parameters of the arena such as: tree width [Obdrzálek 2003], clique width [Obdrzálek 2007], DAG width [Berwanger et al. 2012a], Kelly width [Hunter and Kreutzer 2008] or entanglement [Berwanger et al. 2012b]. Perhaps the most natural parameter is the number of ranks used by the ranking function – and fixed parameter tractability is open for this particular parameter. In other words, it is not known if there is an algorithm which solves a parity game with $k$ ranks and $n$ vertices in time $f(k) \cdot n^c$ for some computable function $f$ and some exponent $c$ which does not depend on $k$.

## 2. DO ALL REGULAR LANGUAGES HAVE GENERALISED STAR HEIGHT ONE?

A *generalised regular expression* is one that can use complementation along the more standard operations of concatenation, union and Kleene star. Since regular languages are closed under complementation, generalised regular expressions have the same expressive power as standard regular expressions, although they can be more succinct, even nonelementarily more succinct as shown in [Stockmeyer 1974].

**The open problem.** Is there a regular language of finite words which cannot be defined by a generalised regular expression of star height one, i.e. one that does not nest the Kleene star?

This open problem is one of the questions concerning star height, which have motivated a lot of research in automata theory. The star height of a regular expression, generalised or not, is defined to be the nesting depth of the Kleene star. For instance, the (non-generalised) regular expression $(a^*b)^*a^*$ has star height 2. The star height of a language is defined to be the smallest star height of a regular expression that defines it. There are two variants of star height, generalised and non-generalised, depending on the type of regular expressions that are considered. In other words, the open problem is: does every regular language have generalised star height zero or one?

*Non-generalised star height.* Non-generalised star height is by now quite well understood. There are regular languages that have arbitrarily high non-generalised star height [Eggan 1963]. Whether or not the non-generalised star height can be computed was an open problem for 35 years, until it was shown to be computable in [Hashiguchi 1988], see [Kirsten 2005] for a simpler proof.

*Generalised star height.* Much less is known about the hierarchy of generalised star height. The only level that is understood is level zero, which is called the *star-free languages*, i.e. the languages that can be defined from finite languages using only Boolean operations and concatenation. A famous result shown in [Schützenberger 1965] says that a languages is star-free if and only if it is recognised by a finite monoid which does not contain a nontrivial group. There is also an important logical connection, shown in [McNaughton and Papert 1971]: a language is star-free if and only if it can be defined by a formula of first-order logic, which quantifies over positions of the word, has a binary predicate $x \leq y$ for the order on positions, and has unary predicates $a(x)$. For example, the formula

$$\forall x \exists y \; x \leq y \wedge a(y)$$

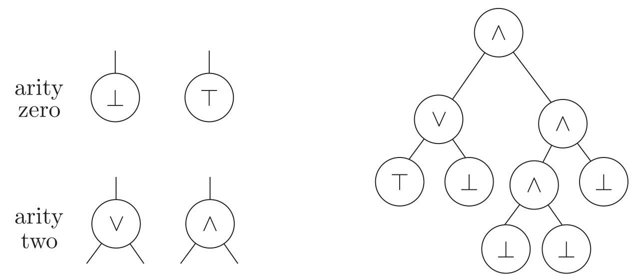
says that the every position is followed by a position with label $a$. The language defined by this example formula is star-free because it is the concatenation of the set of all words (the complement of the empty set) with the finite language $\{a\}$.

Beyond star-free languages, i.e. beyond level zero, very little is known. As stated in the open problem, as far as we know, maybe all regular languages have star height at most one, or maybe at most fifteen. As shown in [Pin et al. 1992], assuming that some languages have star height at least two, then the generalised star height of a regular language cannot be determined just by looking at the language's syntactic monoid. The paper [Pin et al. 1992] also contains several surprising languages that have generalised star height one, including the language "an even number of infixes of the form $abc$", which was conjectured to have bigger generalised star height in [Brzozowski 1980].

### 3. WHICH REGULAR LANGUAGES OF FINITE TREES ARE FIRST-ORDER DEFINABLE?

This problem concerns regular languages of finite trees. For the sake of concreteness, we use the variant of ranked trees. In this variant, the trees are over a *ranked alphabet*, where each letter comes with an arity, and trees are labelled by the alphabet so that the number of children of a node is the arity of its label, as depicted in the following picture.

a ranked alphabet a tree over the ranked alphabet




*Tree automata.* A good notion of tree automaton for such trees is a *bottom-up deterministic automaton*. In such an automaton, if the states are $Q$, then every letter $a$ of arity $n$ induces a transition function of type $Q^n \to Q$. In particular, each leaf, which has a label of arity zero, comes with an associated state. By applying the transition functions in a bottom-up manner, the automaton evaluates a tree to a single state. A tree language is called regular if there is some automaton such that membership in the language is uniquely determined by the state to which a tree evaluates.

*Monadic second-order logic on trees.* Many results on automata can be generalised from finite words to finite trees without much difficulty, e.g. pumping lemmas or equivalence of deterministic and non-deterministic bottom-up automata. Another example of result that can be generalised is the following correspondence between monadic second-order logic and regular languages. A tree can be seen as a logical structure, where the universe is the set of nodes, and which has the following predicates: a unary predicate "node $x$ has label $a$" for every letter $a$ of the alphabet, a binary predicate "node $x$ is a descendant of node $y$", and a binary predicate "node $x$ is the $i$-th child of node $y$" for every number $i$ up to the maximal arity in the alphabet. A classical result on tree automata [Thatcher and Wright 1968] says that a tree language is regular if and only if it is definable in monadic second-order logic, i.e. the logic which can quantify over nodes and sets of nodes.

*First-order logic on trees.* Languages that are definable in first-order logic are a strict subclass of regular tree languages. For instance the language "some leaf is at even depth" is not definable in first-order logic if the alphabet includes at least one letter of rank one. (Although, as will be later discussed, the language is definable, somehow annoyingly, when there are no letters of rank one.) Another example, which does not have any natural word counterpart, is the language of Boolean expressions, as in the picture above, that evaluate to $\top$.

**The open problem.** Can one decide whether a regular language of finite trees can be defined by a formula of first-order logic that uses the label predicates, the descendant predicate, and $i$-th child predicates?

In the case of finite words, the problem above is well understood. The results of Schützenberger, McNaughton and Papert, which were mentioned previously in the context of star-free languages, imply that one can decide if a regular language of finite words can be defined in first-order logic: one can compute the syntactic monoid and then test if it contains a nontrivial group.

For trees much less is known. The open problem dates back to [Thomas 1984]. In [Heuter 1991] it was shown that aperiodicity (in a natural tree variant) is a necessary but not sufficient condition for definability in first-order logic on finite trees. In other words, the equivalence of first-order definability and aperiodicity fails to extend from words to trees. For the equivalence of first-order definability and star-freeness, the story is a bit more complicated: there are variants of star-free expressions that are strictly more expressive than first-order logic [Thomas 1984], and there are variants of star-free expressions that are equally expressive as first-order logic [Bojańczyk 2007]. Finally, first-order logic on finite trees admits characterisations in terms of temporal logics, namely two-way CTL [Schlingloff 1992] and one-way CTL* [Hafer and Thomas 1987]. These characterisations show that first-order logic on trees is a robust concept, but they do not seem to be useful in solving the open problem.

As usual for trees, there are several variants of the problem, e.g. one can consider the logic without the $i$-th child predicates, or one can consider unranked trees with or without a predicate for sibling order. In all of these cases it is not known how to characterise first-order logic.

The only known result is about full first-order logic without the descendant relation, where only predicates for the labels and the child relation are allowed. In this case, one can decide if a language is definable in first-order logic [Benedikt and Segoufin 2009]. Other known results talk about fragments of first-order logic with limited quantification patters, e.g. Boolean combinations of existential formulas [Bojanczyk et al. 2012] or first-order logic with only two variables [Place and Segoufin 2010].

To give a taste of the difficulties inherent to first-order logic on trees, consider the following example, which is due to [Potthoff 1995]. Assume that the alphabet has one letter of rank zero and one letter of rank two, and the language is "some leaf is at even depth". As mentioned before, this language is definable in first-order logic, using the descendant, left child and right child predicates. Despite being invariant under swapping left and right subtrees, the language is not definable in first-order logic using the descendant predicate only.

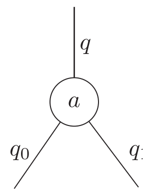
## 4. THE RABIN-MOSTOWSKI INDEX HIERARCHY

Tree automata are studied – arguably, better studied – also for infinite trees. Let us consider labelled binary trees, where every node has a label from a finite set and exactly two children. The famous Rabin theorem [Rabin 1969] says that a language of labelled binary trees is definable in MSO if and only if it is recognised by a nondeterministic automaton with the Rabin condition. Currently, instead of automata with the Rabin condition, one uses the parity condition, in their nondeterministic and alternating variants, as described below. Both of these variants are equally expressive as MSO, and therefore as nondeterministic automata with the Rabin condition.

*Nondeterministic parity tree automata.* A nondeterministic parity tree automaton consists of an input alphabet $A$, a set of states $Q$, an initial state $q_0 \in Q$, a parity ranking function $\Omega : Q \to \mathbb{N}$, and a transition relation

$$\delta \subseteq Q \times A \times Q \times Q.$$

A transition can be visualised as a node in a binary tree, with the node labelled by the input alphabet, and its surrounding edges labelled by states:




When given an input tree, a run of the automaton is a labelling of the edges of the tree by states, including a special dummy edge that enters the root from above, such that the neighbourhood of every node is consistent with some transition. A run is accepting if the dummy edge gets the initial state, and on every infinite path, the parity condition is satisfied by the ranks assigned to the states.

*Alternating parity tree automata.* An alternating parity tree automaton differs from a nondeterministic one in two ways: a) there is a partition of the states into states owned by players Even and Odd, b) the transition relation is a subset

$$\delta \subseteq Q \times A \times \{\text{same, left, right}\} \times Q.$$

To determine whether or not such an automaton accepts an input tree, a parity game is played between players Even and Odd. Positions of the game are pairs $(q, v)$, where $q$ is a state of the automaton and $v$ is a node of the input tree. The initial position is the pair consisting of the initial state and the root of the tree. When the game is in a position $(q, v)$, then the player who controls state $q$ chooses a transition $(q, a, i, p)$, and the game continues from position $(p, w)$, where $w$ is either the same node as $v$, the left child of $v$, or the right child of $v$, depending on the value of $i$. The tree is accepted if player Even has a winning strategy in this game, according to the parity condition as determined by the parity ranking function on the states.

*The index hierarchies.* For numbers $i \leq j$, consider parity automata, either nondeterministic or alternating, where the parity ranking function has image included in $\{i, \ldots, j\}$. It is easy to see that the class of recognised languages depends only on the parity of the smaller rank $i$, and on the number of available ranks $j - i + 1$. In other words, one can assume without loss of generality that $i$ is either 0 or 1.

Adding more ranks gives a strict hierarchy, in the following sense, see [Bradfield 1998; Arnold 1999]. For every $i$ and every $j \geq i$, there exists a language $L_{ij}$ which is recognised by a nondeterministic parity automaton with ranks $\{i, \ldots, j\}$, but which is not recognised by an alternating parity automaton with ranks $\{i+1, \ldots, j+1\}$, i.e. the same number of ranks but shifted by one. The open problem is to decide the position of a regular language of infinite trees in this hierarchy:

**The open problem.** Are the following decision problems decidable? In each of them, the input is a regular language of infinite trees and natural numbers $i \leq j$.

— **Nondeterministic index problem.** Is the language recognised by some nondeterministic parity automaton that uses ranks between $i$ and $j$?
— **Alternating index problem.** Is the language recognised by some alternating parity automaton that uses ranks between $i$ and $j$?

*Known results.* The nondeterministic and alternating index problems are known to be decidable assuming that the input regular language is recognised by a top-down deterministic parity automaton [Niwiński and Walukiewicz 2003], or even assuming that the input regular language is recognised by a generalisation of deterministic top-down automata called *game automata* [Duparc et al. 2009]. In [Colcombet and Löding 2008], the nondeterministic index problem is reduced to a problem about counter automata on infinite trees; the latter problem (which is interesting in its own right) remains open. A partial result on these counter automata is given in [Colcombet et al. 2013], this partial result implies that one can decide, given a language recognised by a nondeterministic Büchi automaton (i.e. using ranks $0$ and $1$), if the complement of the language is also recognised by a nondeterministic Büchi automaton.

*Weak* MSO. An important special case of the index problems is deciding if a regular tree language can be defined in weak MSO, i.e. the variant of MSO where set quantification is restricted to finite sets. Why is this a special case? As shown in [Rabin 1970], a language of infinite trees is definable in weak MSO if and only if both the language and its complement are recognised by nondeterministic Büchi tree automata. Furthermore, with the Büchi acceptance condition, nondeterministic and alternating automata have the same expressive power [Muller and Schupp 1995]. Therefore, if one could decide which regular languages of infinite trees are recognised by nondeterministic (equivalently, alternating) Büchi tree automata, then one could decide which ones are definable in weak MSO. A related conjecture, see [Skurczyński 1993], is that for regular languages of infinite trees, being Borel is equivalent to being definable in weak MSO.

### 5. MONADIC SECOND-ORDER LOGIC ON THE CANTOR SPACE.

Consider the sets $2^*$ and $2^\omega$ as logical structures: the former is equipped with both descendant and lexicographic orders, and the latter is equipped with only the lexicographic order. The Rabin theorem says that the first structure has decidable MSO theory. Shelah showed that the second structure, i.e. the Cantor space, has undecidable MSO theory. A corollary is that the real numbers have undecidable MSO theory. Shelah's original proof in [Shelah 1975] uses the Continuum Hypothesis, while a later proof [Gurevich and Shelah 1982] uses only the axioms of ZFC. In the undecidability proof, it is important that formulas can quantify over arbitrary subsets of $2^\omega$ and not just simple sets, such as Borel sets. Therefore, Shelah stated the following problem, which remains open.

**The open problem.** Can one decide the MSO theory of the Cantor space $2^\omega$ with lexicographic ordering, assuming that set qantifiers range only over Borel sets?

To illustrate the problem, consider a logic which uses a smaller prefix of the Borel hierarchy, namely level $\Pi_2$, i.e. the countable intersections of open sets. Consider MSO logic over $2^\omega$ where set quantification is restricted to $\Pi_2$ sets. We claim that this logic is decidable, as it reduces to MSO on $2^*$ with left and right child predicates, i.e. to the logic from Rabin's theorem. The idea is to reduce both first-order quantification and set quantification to set quantification in $2^*$. For first-order quantification, one simply encodes an element of $2^\omega$ as the set of its finite prefixes, i.e. a path in a tree. To encode a set $X \subseteq 2^\omega$ that is in $\Pi_2$, one uses the following observation, which follows easily from the definition of level $\Pi_2$ in the Borel hierarchy: a set $X \subseteq 2^\omega$ belongs to $\Pi_2$ if and only if there exists a subset $[X] \subseteq 2^*$ such that

$$\pi \in X \qquad \text{iff} \qquad \pi \text{ has infinitely many prefixes in } [X].$$

This reduction seems to use almost all of the power of the Rabin theorem. It seems that giving a positive answer to the open problem would require a significant extension of the techniques in the Rabin theorem.

## 6. IS REACHABILITY DECIDABLE FOR BRANCHING VECTOR ADDITION SYSTEMS?

One of the most famous decidability results [Mayr 1984] is that reachability is decidable in vector addition systems with states. A vector addition system with states, VASS for short, is a type of counter machine that is equivalent to Petri nets, and is defined as follows. The syntax of a VASS consists of a finite set of states $Q$ with a distinguished initial state, a dimension $d \in \mathbb{N}$, and a finite subset of $\delta \subseteq Q \times \mathbb{Z}^d \times Q$, which is called the set of transitions. The semantics of a VASS is the set of *reachable configurations*, which is the least set of pairs in $Q \times \mathbb{N}^d$ such that:

— If $q$ is the initial state, then $(q, \bar{0})$ is a reachable configuration.
— If a configuration $(q, \bar{a})$ is reachable, and there is a transition $(q, \bar{b}, p)$ such that the vector $\bar{a} + \bar{b}$ has only nonnegative numbers, then also $(p, \bar{a} + \bar{b})$ is reachable.

The key thing is that transitions can have negative numbers, while configurations cannot. As shown in [Mayr 1984], the set of reachable configurations is decidable. One notorious open problem is the computational complexity of reachability – the best lower bound is EXPSPACE [Cardoza et al. 1976], but it is not even known if reachability has elementary complexity. A recent decidability proof can be found in [Leroux 2012].

*Branching vector addition systems with states.* A branching vector addition system with states, BVASS for short, generalises a VASS by allowing a new kind of transition, call it a *branching transition*, which is a triple of states. The set of reachable configurations is defined as in a VASS, with the additional rule that if $(q, \bar{a})$ and $(p, \bar{b})$ are reachable configurations, and the BVASS contains a branching transition $(p, q, r)$, then also $(r, \bar{a} + \bar{b})$ is a reachable configuration.

**The open problem.** Is reachability decidable for branching vector addition systems?

It is known that reachability for BVASS is Ackermann hard [Lazić and Schmitz 2014], i.e. if it is decidable then the running time of the algorithm must be bigger than the Ackermann function. Control state reachability – i.e. the problem of determining which states can appear in reachable configurations – is known to be decidable [Verma and Goubault-Larrecq 2005].

*Connections with logic.* Here are two examples of how the problem is connected with logic. The first connection comes from linear logic – the decidability of the reachability problem for BVASS is equivalent to the decidability of multiplicative exponential linear logic [de Groote et al. 2004]. The second connection comes from the theory of XML – the reachability problem for BVASS reduces to the satisfiability problem of a certain logic on data trees, namely two variable first-order logic with predicates for descendant, successor, next sibling, and equal data value [Bojańczyk et al. 2009].

### References

A. Arnold. 1999. The $\mu$-calculus alternation-depth hierarchy is strict on binary trees. *Informatique Théorique et Applications* 33, 4/5 (1999), 329–340.

Michael Benedikt and Luc Segoufin. 2009. Regular tree languages definable in FO and in $\mathrm{FO}_{mod}$. *ACM Trans. Comput. Log.* 11, 1 (2009).

Dietmar Berwanger, Anuj Dawar, Paul Hunter, Stephan Kreutzer, and Jan Obdrzálek. 2012a. The dag-width of directed graphs. *J. Comb. Theory, Ser. B* 102, 4 (2012), 900–923.

Dietmar Berwanger, Erich Grädel, Łukasz Kaiser, and Roman Rabinovich. 2012b. Entanglement and the Complexity of Directed Graphs. *Theoretical Computer Science* 463, 0 (2012), 2–25. http://logic.rwth-aachen.de/~rabinovich/entangle.pdf

Mikołaj Bojańczyk. 2007. Forest Expressions. In *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*. 146–160.

Mikołaj Bojańczyk, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. 2009. Two-variable logic on data trees and XML reasoning. *J. ACM* 56, 3 (2009).

Mikolaj Bojanczyk, Luc Segoufin, and Howard Straubing. 2012. Piecewise testable tree languages. *Logical Methods in Computer Science* 8, 3 (2012).

J.C. Bradfield. 1998. The Modal $\mu$-Calculus Alternation Hierarchy is Strict. *Theor. Comput. Sci.* 195, 2 (1998), 133–153.

J.A. Brzozowski. 1980. *Open Problems about Regular Languages*. Department of Computer Science, University of Waterloo.

E. Cardoza, Richard J. Lipton, and Albert R. Meyer. 1976. Exponential Space Complete Problems for Petri Nets and Commutative Semigroups: Preliminary Report. In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing, May 3-5, 1976, Hershey, Pennsylvania, USA*. 50–54.

Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. 2013. Deciding the weak definability of Büchi definable tree languages. (2013). submitted.

Thomas Colcombet and Christof Löding. 2008. The Non-deterministic Mostowski Hierarchy and Distance-Parity Automata. In *ICALP (2)*. 398–409.

Philippe de Groote, Bruno Guillaume, and Sylvain Salvati. 2004. Vector Addition Tree Automata. In *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*. 64–73.

Jacques Duparc, Alessandro Facchini, and Filip Murlak. 2009. Linear Game Automata: Decidable Hierarchy Problems for Stripped-Down Alternating Tree Automata. In *CSL*. 225–239.

Lawrence C. Eggan. 1963. Transition graphs and the star-height of regular events. *Michigan Math. J.* 10, 4 (1963), 385–397.

E. Allen Emerson and Charanjit S. Jutla. 1991. Tree Automata, Mu-Calculus and Determinacy (Extended Abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*. 368–377. DOI:http://dx.doi.org/10.1109/SFCS.1991.185392

Oliver Friedmann. 2009. An Exponential Lower Bound for the Parity Game Strategy Improvement Algorithm as We Know it. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11-14 August 2009, Los Angeles, CA, USA*. 145–156.

Oliver Friedmann, Thomas Dueholm Hansen, and Uri Zwick. 2011. Subexponential lower bounds for randomized pivoting rules for the simplex algorithm. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*. 283–292.

Yuri Gurevich and Saharon Shelah. 1982. Monadic theory of order and topology in ZFC. *Annals of Mathematical Logic* 23, 2-3 (1982), 179–198.

Thilo Hafer and Wolfgang Thomas. 1987. Computation Tree Logic CTL* and Path Quantifiers in the Monadic Theory of the Binary Tree. In *Automata, Languages and Programming, 14th International Colloquium, ICALP87, Karlsruhe, Germany, July 13-17, 1987, Proceedings*. 269–279.

Kosaburo Hashiguchi. 1988. Relative star height, star height and finite automata with distance functions. In *Formal Properties of Finite Automata and Applications*. 74–88.

Uschi Heuter. 1991. First-order properties of trees, star-free expressions, and aperiodicity. *ITA* 25 (1991), 125–146.

Paul Hunter and Stephan Kreutzer. 2008. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.* 399, 3 (2008), 206–219.

Daniel Kirsten. 2005. Distance desert automata and the star height problem. *Informatique Théorique et Applications* 39, 3 (2005), 455–509.

Ranko Lazić and Sylvain Schmitz. 2014. Non-elementary complexities for branching VASS, MELL, and extensions. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*. 61.

Jérôme Leroux. 2012. Vector Addition Systems Reachability Problem (A Simpler Solution). In *Turing-100 - The Alan Turing Centenary, Manchester, UK, June 22-25, 2012*. 214–228.

Ernst W. Mayr. 1984. An Algorithm for the General Petri Net Reachability Problem. *SIAM J. Comp.* 13, 3 (1984), 441–459.

R. McNaughton and S. Papert. 1971. *Counter-Free Automata*. MIT Press.

Andrzej W. Mostowski. 1991. *Games with forbidden positions*. Technical Report. University of Gdańsk.

D. E. Muller and P. E. Schupp. 1995. Simulating Alternating Tree Automata by Nondeterministic Automata: New Results and New Proofs of the Theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science* 141 (1995), 69–107.

Damian Niwiński and Igor Walukiewicz. 2003. A gap property of deterministic tree languages. *Theor. Comput. Sci.* 1, 303 (2003), 215–231.

Jan Obdrzálek. 2003. Fast Mu-Calculus Model Checking when Tree-Width Is Bounded. In *CAV*. 80–92.

Jan Obdrzálek. 2007. Clique-Width and Parity Games. In *Computer Science Logic (CSL)*. 54–68.

Jean-Éric Pin, Howard Straubing, and Denis Thérien. 1992. Some Results on the Generalized Star-Height Problem. *Inf. Comput.* 101, 2 (1992), 219–250.

Thomas Place and Luc Segoufin. 2010. Deciding Definability in $\mathrm{FO}_2(<)$ (or XPath) on Trees. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*. 253–262.

A. Potthoff. 1995. First-Order Logic on Finite Trees. In *Theory and Practice of Software Development (Lecture Notes in Computer Science)*, Vol. 915. 125–139.

M. O. Rabin. 1969. Decidability of Second-Order Theories and Automata on Infinite Trees. *Transactions of the AMS* 141 (1969), 1–23.

M. O. Rabin. 1970. Weakly Definable Relations and Special Automata. *Mathematical Logic and Foundations of Set Theory* (1970), 1–23.

Bernd-Holger Schlingloff. 1992. Expressive completeness of temporal logic of trees. *Journal of Applied Non-Classical Logics* 2, 2 (1992), 157–180.

M. P. Schützenberger. 1965. On Finite Monoids Having Only Trivial Subgroups. *Information and Control* 8 (1965), 190–194.

Saharon Shelah. 1975. The Monadic Theory of Order. *The Annals of Mathematics* 102, 3 (1975), 379–419.

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

L.J. Stockmeyer. 1974. *The Complexity of Decision Problems in Automata Theory and Logic*. Ph.D. Dissertation.

J. W. Thatcher and J. B. Wright. 1968. Generalized Finite Automata Theory With an Application to a Decision Problem of Second-Order Logic. *Mathematical Systems Theory* 2, 1 (1968), 57–81.

Wolfgang Thomas. 1984. Logical Aspects in the Study of Tree Languages. In *CAAP*. 31–50.

Kumar Neeraj Verma and Jean Goubault-Larrecq. 2005. Karp-Miller Trees for a Branching Extension of VASS. *Discrete Mathematics & Theoretical Computer Science* 7, 1 (2005), 217–230.

Jens Vöge and Marcin Jurdziński. 2000. A Discrete Strategy Improvement Algorithm for Solving Parity Games. In *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*. 202–215. DOI:http://dx.doi.org/10.1007/10722167_18

Wiesław Zielonka. 1998. Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees. *Theor. Comput. Sci.* 200, 1-2 (1998), 135–183.