AUT

AUTOMATA COLUMN

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

U

MIKOŁAJ BOJAŃCZYK, University of Warsaw

This is a survey of extensions of logics and automata which talk about boundedness. A typical property of interest is the set of ω-words which satisfy "there exists some k, such that every a letter is followed by a b letter in at most k steps". The main points of interest are the logic MSO+U, its fragments and related automata models, as well as the regular cost functions of Colcombet.

To begin our discussion of boundedness, consider one of the archetypical liveness properties, namely "every a event is followed by a b event in a finite number steps". In the syntax of linear temporal logic LTL, this property is written as

$$\mathsf{G}\big(a \Rightarrow \mathsf{F}b\big).$$

What could be more natural than asking for the b to appear in a bounded number of steps? Adding such boundedness constraints is the idea behind *prompt* LTL, a logic introduced in [Kupferman et al. 2009]. In prompt LTL, one writes formulas like

$$\exists k \in \mathbb{N} \quad \mathsf{G}\big(a \Rightarrow \mathsf{F}^{\leq k}b\big),$$

where $\mathsf{F}^{\leq k}$ means "in at most k steps". Assuming that we are talking about languages of ω-words over the alphabet $\{a, b\}$, the language that corresponds to the above formula of prompt LTL is

$$\{a^{n_1}ba^{n_2}ba^{n_3}b\cdots : \limsup n_i < \infty\}.$$

This language will be our running example. The goal of this paper is to discuss logics and automata which describe the running example and its variants. There are three sources of motivation.

A richer modelling language. The first source of motivation, highlighted by the prompt LTL example, is that boundedness is one of the most basic kinds of asymptotic properties, and it is therefore unsurprising that it has found its way into formalisms expressing properties of infinite computation. For example, one can consider variants of parity games (and Streett, etc.) where the winning condition requires that something good happens in a bounded amount of time [Chatterjee et al. 2009; Bloem et al. 2009; Fijalkow and Zimmermann 2014].

A new tool in a logician's toolbox. The second source motivation is that boundedness questions appear implicitly when solving problems without an explicit boundedness character. A famous example is the star height problem. As discovered by Hashiguchi [Hashiguchi 1988], this problem can be solved by reducing it to a decidable boundedness problem, to be discussed later in this paper. Other problems in formal language theory that reduce to boundedness questions include the star height problem for regular tree languages [Colcombet and Löding 2008a], or the Mostowski index problem for automata on infinite trees [Colcombet and Löding 2008b], although in the latter case, both the Mostowski index problem and its corresponding boundedness problem remain open, see [Fijalkow et al. 2015] for recent developments. Another example of a problem that reduces to a boundedness question is the finite satisfiability problem for fixpoint logics such as: the modal μ-calculus with backward modalities [Bojańczyk 2002], guarded fixpoint logic [Bárány and Bojańczyk 2012] or guarded negation logic [Bárány et al. 2015]. Other examples where boundedness questions arise include: a question about eliminating fixpoint operators in [Blumensath et al. 2014c], a satisfiability question for a variant of CTL* in [Carapelle et al. 2013], or a characterisation of behaviours of communicating timed automata [Aminof et al. 2015].

Understanding regularity. A third source of motivation is the quest for understanding "regular languages" for infinite objects, such as ω-words. Consider the language in the running example. Is it a "regular language"? It is not regular in the accepted sense, i.e. it is not recognised by any nondeterministic Büchi automaton (a straightforward pumping argument), and therefore it is also not definable by any formula of monadic second-order logic MSO. Nevertheless, the language looks innocent enough, and one may wonder whether it might belong to some class of languages with a simple definition, maybe of a logical character, and with good closure and decidability properties. The main topic of this paper is to survey several proposals for such classes.

Counting without actually counting. The motivation of "understanding regularity" also limits the scope of logics studied in this paper. We would like the languages to resemble "regular" languages in some way. For example, for languages of ω-words, at the very least we require every language $L \subseteq A^\omega$ to have finitely many equivalence classes for the Myhill-Nerode equivalence relation on finite words $w, v \in A^*$ defined by

$$w \sim_L v \qquad \text{if} \qquad \forall u \in A^\omega \; wu \in L \iff vu \in L.$$

Such a restriction means that any counting can only be done in some asymptotic way. This excludes all sort of boundedness questions where precise counting is involved, e.g. the rich body of literature on boundedness for vector addition systems.

1. AUTOMATA WITH COUNTERS

To the author's knowledge, the first deeper study of boundedness in automata and logic was in the context of the star height problem. We begin our story with these automata.

1.1. Automata for star height

Distance automata. A *distance automaton* is the same as a nondeterministic automaton over finite words, except that it has a distinguished subset of transitions, which are assumed to be "costly". Another view on distance automata, which will be used in this paper, is that a distance automaton is a very restricted kind of counter automaton, which has a single counter that is incremented whenever a costly transition is encountered. Yet another view on distance automata is that they are the same thing as a weighted automaton in the min-plus semiring, also known as the tropical semiring. The main problem studied for distance automata is the following boundedness problem: given a distance automaton, decide if there is some $k \in \mathbb{N}$ such that every input word admits some run that uses at most k costly transitions. This boundedness problem was first intensively studied in connection with the star height problem from formal language theory. In [Hashiguchi 1988], it was shown that for every regular language L and number $n \in \mathbb{N}$, one can compute a distance automaton such that the distance automaton is bounded if and only if the regular language can be defined by a regular expression which nests the Kleene star at most n times. In other words, the star height problem reduces to boundedness of distance automata. Hashiguchi's reduction was extremely challenging.

Nested distance desert automata. A second decidability argument for the star height problem was given in [Kirsten 2005]. Like Hashiguchi, Kirsten also reduces the star height problem to a boundedness problem, but the reduction is much simpler at the cost of a slightly harder boundedness problem (and therefore, the whole proof is much simpler). The slightly harder boundedness problem concerns *nested distance desert automata*, which are a generalisation of distance automata where several counters are allowed, and the counters can be reset. Building on the ideas of Leung and Simon, Kirsten showed that emptiness for such automata is decidable, in fact PSPACE-complete.

1.2. ωB- and ωS-automata.

Soon after Kirsten, but independently, similar automata models appeared in [Bojańczyk and Colcombet 2006], and [Abdulla et al. 2008]. In both cases, the automata had several counters, and the questions studied concerned boundedness. In this survey, we will focus more on the former paper, because of the duality result that it contains, which is presented in Theorem 1.1, and which will be of later interest when talking about regular cost functions. The paper [Bojańczyk and Colcombet 2006] introduces two types of automata, which are called, for no good reason, ωB-automata and ωS-automata. Both kinds of automata have the same syntax, defined as follows[1]. There is a nondeterministic finite state automaton, with initial states but without accepting states, a finite set of counters that store natural numbers, and a function which maps each transition to a finite sequence of operations from the following toolkit:

(1) reset counter c to zero;
(2) increment counter c;
(3) check[2]

Both ωB- and ωS-automata are evaluated on ω-words, unlike the finite words used in the automata for the star height problem. (In a single finite word, every counter is bounded by the length of the word, and therefore meaningful boundedness questions only make sense for sets of finite words.) A run of an ωS-automaton is defined to be accepting if every counter is checked infinitely often, and its values have infinite $\liminf$. A run of an ωB-automaton is defined to be accepting if every counter is checked infinitely often, and its values are bounded (i.e. have finite $\limsup$).

Cis Büchi. To get a feeling for the model, let us see how both kinds of automata can simulate Büchi automata, at least in the presence of nondeterminism. The simulation works already with one counter. To do this with an ωB-automaton, the counter is never incremented or reset, and it is checked whenever the simulated Büchi automaton reaches an accepting state. This simulation simply uses the Büchi condition which is implicit in the semantics, where the counters are required to be checked infinitely often. To simulate a Büchi automaton with an ωS-automaton, we do essentially the same thing, only we increment the counter in every step without ever resetting it, so that it has infinite $\liminf$. As shown in [Skrzypczak 2014], simulating a Büchi automaton is the only kind of thing that can be done by both ωB- and ωS-automata in the following sense: a language is recognised by a nondeterministic Büchi automaton

[1]This is essentially the same syntax as for the automata used by Kirsten.
[2]The original paper used a slightly different model, without checks, but which has the same expressive power. The check operation was introduced in [Colcombet and Löding 2008a].

if and only if it is recognised by some nondeterministic ωB-automaton and by some nondeterministic ωS-automaton.

Trans Büchi. The simulation of Büchi automata discussed above did not use the full power of the counters. Here is an example which does. The recognised language is our running example

$$\{a^{n_1}ba^{n_2}ba^{n_3}b\cdots : \limsup n_i < \infty\}.$$

To recognise this language, one uses an ωB automaton that has one state and one counter, and the counter is incremented when reading a's and reset when reading b's. The example above is deterministic, but typically one uses nondeterministic automata, as is the case for the complement of the above language, which is

$$\{a,b\}^*a^\omega \cup \{a^{n_1}ba^{n_2}ba^{n_3}b\cdots : \limsup n_i = \infty\}$$

To recognise this language, one needs a nondeterministic ωS-automaton. The automaton has one counter. It nondeterministically guesses if the word will have finitely many b's, or if the size of a blocks will have infinite $\limsup$. In the first case, it uses the same trick as for simulating Büchi automata. In the second case, a subsequence of the blocks must have lengths with infinite $\liminf$, and these blocks are chosen using nondeterminism. This language is not recognised by any deterministic ωS-automaton, and in fact there is no known deterministic model (with maybe more fancy acceptance conditions) that is equivalent to nondeterministic ωS-automata.

Algorithmic questions. Usually the point of using automata is that algorithmic questions are easy for automata, and the proof burden is moved to showing that things can be done with automata. The ωB- and ωS-automata are no exception, in particular their emptiness problems are quite easy to solve. Let us show that emptiness for ωB-automata is decidable. The reason is that for an ωB-automaton, the following conditions are equivalent:

(1) there exists an accepting run;
(2) there exists a run where for every counter, a check appears infinitely often and infinitely many increments imply infinitely many resets.

Clearly (1) implies (2). To prove the converse implication, we observe that the property of runs mentioned in condition (2) is an ω-regular property. Since ω-regular languages must contain ultimately periodic words, if (2) holds, then there exists some *ultimately periodic* run where every counter is checked infinitely often, and infinitely many increments imply infinitely many resets. Such ultimately periodic runs have counters bounded by the length of the period. For ωS-automata, a slightly more challenging but still simple argument is used, although ultimately periodic runs are no longer the relevant object.

The duality theorem. The main theorem about ωB- and ωS-automata is the following duality result.

THEOREM 1.1. *[Bojańczyk and Colcombet 2006] For every ωB-automaton one can compute an ωS-automaton recognising its complement language, and vice versa.*

A crucial part of the above theorem is the Factorisation Forest Theorem from [Simon 1990].

Deciding star height. As an application of the above duality theorem, let us use it to decide boundedness of distance automata. (The proof also works for the model used in Kirsten's proof, which has multiple counters.) Suppose that we want to know if nested distance automaton is bounded, i.e. it has a common upper bound on the counter values needed to accept all finite words. We will solve the problem using double negation. A counterexample to boundedness is an ω-word

$$w_1 \# w_2 \# w_3 \# \cdots$$

such that no finite counter size is sufficient to accept all words $w_1, w_2, \ldots$, assuming that # is some fresh symbol not in the alphabet of the original distance automaton. The set of ω-words that are *not* counterexamples is recognised by an ωB-automaton, which has essentially the same structure as the original distance automaton, except that it resets all counters whenever it sees a # separator. By Theorem 1.1, the set of counterexamples is recognised by an ωS-automaton, which can be effectively computed. Therefore, boundedness, which is the lack of counterexamples, reduces to the decidable problem of testing an ωS-automaton for emptiness.

2. COST FUNCTIONS

A problem with ωB- and ωS-automata is that they define two classes of languages, which albeit dual, are not the same. A solution to this, called *cost functions*, was proposed by Thomas Colcombet in [Colcombet 2009].

Beyond yes and no. In cost functions, the automata produce numbers instead of accepting or rejecting an input[3]. On its own, the idea of producing numbers alone is not new. It dates back at least to weighted automata [Schützenberger 1961]. It was also mentioned in our discussion of distance automata, since the semantics of a distance automaton over input alphabet Σ can be viewed as a function

$$[\![\mathcal{A}]\!] : \Sigma^* \to \mathbb{N} \cup \{\infty\},$$

which maps an input word to the minimum, ranging over accepting runs, of costly transitions used by the run, with ∞ is used for words that do not have any accepting runs. The new idea of Colcombet was to consider only the asymptotic behaviours of these number valued functions.

Undecidability. With automata that produce numbers, it is easy to run against undecidability. For example, as shown in [Krob 1992], it is undecidable whether or not two given distance automata define the same function from words to numbers. To escape this undecidability, Colcombet proposes to consider functions only with respect to their asymptotic behaviour, in the following sense. Consider two functions $f, g : X \to \mathbb{N} \cup \{\infty\}$. In the current discussion, X is the set of all finite words over a given alphabet, but it may very well be other things, like infinite trees. We say that g *dominates* f if for every subset $Y \subseteq X$ of arguments, boundedness of g implies boundedness of f. Call functions *domination equivalent* if they dominate each other. For example, a function is bounded if and only if it is dominated by, and therefore domination equivalent to, the constant zero function. As it turns out, the undecidability proof of Krob relies on keeping track of exact values produced by distance automata, and no longer works when functions are considered only up to domination equivalence.

In [Colcombet 2009] it is shown that all the undecidability problems go away when considering functions up to domination equivalence, in particular the domination ordering is decidable for distance automata, and even Kirsten's nested distance desert automata. Actually, Colcombet does more than this: he proves that there is a class of functions which has very robust closure properties and multiple characterisations when considered up to domination equivalence. These functions are called *regular cost functions*, and are described below in some more detail.

[3]One does feel some influence of the evil one, see Matthew 5:37.

B- and S-automata. One definition of regular cost functions is via an automaton model. There are actually two models, called *B- and S-automata*. These have the same syntax, which is also the same as the syntax for ωB- and ωS-automata. There is only one difference in the syntax: in the B- and S-automata, as opposed to the ωB- and ωS-automata, there is also a distinguished set of accepting states. The semantics are defined as follows. The automata input finite words and ouptut numbers in $\mathbb{N} \cup \{\infty\}$. For a B-automaton $\mathcal{A}$, the value for an input word w is defined by

$$[\![\mathcal{A}]\!](w) = \min_{\rho} \max_{n} n$$

where ρ ranges over runs on w that begin in an initial and end in a finite state, while n ranges over values checked by ρ, regardless of the counter. For an S-automaton, the value for an input word is

$$[\![\mathcal{A}]\!](w) = \max_{\rho} \min_{n} n,$$

with ρ and n having the same ranges as in the B case.

Duality. The main result about B- and S-automata is the following beautiful and deep duality theorem.

THEOREM 2.1. *[Colcombet 2009] For every B-automaton one can compute a domination equivalent S-automaton and vice versa.*

The concepts used when proving the above theorem are similar to those in Theorem 1.1. However, Theorem 2.1 seems to be a more fundamental result, e.g. a good way to prove Theorem 1.1 is to deduce it from Theorem 2.1.

Regular cost functions. Define a *regular cost function* to be a function defined by a B-automaton, modulo domination equivalence. By Theorem 2.1, when defining regular cost functions, we could have also used S-automata. The equivalence of the two models is a very important result, because some things are easy for B-automata, and some things are easy for S-automata. For example, testing boundedness is difficult for B-automata (this is essentially the heart of the star height problem), while testing boundedness for S-automata is easy (one searches for a certain kind of cycles in the transition graph). Using the duality result in Theorem 2.1, one can go beyond testing boundedness and even decide the domination order on regular cost functions; the reason is that deciding domination $\mathcal{S} \preceq \mathcal{B}$ is relatively straightforward when $\mathcal{S}$ is an S-automaton and $\mathcal{B}$ is an B-automaton.

Cost MSO. Another corollary of the duality theorem is that the class of regular cost functions is closed under $\min$ (thanks to the B-automata description) and under $\max$ (thanks to the S-automata description). This suggests some kind of logic. There is indeed such a logic, and it is called *cost* MSO [Colcombet 2013b]. (There is also an algebraic characterisation, in terms of *stabilisation semigroups*, but we omit it here.) The syntax of cost MSO is the same as for MSO, except that one can also used a predicate "X is small". The additional predicate "X is small" is second-order in the sense that X is a set variable; furthermore there is a syntactic restriction that it can only be used positively. The semantics of a cost MSO formula is a function, which inputs a model and outputs the smallest $k \in \mathbb{N}$ which makes the formula true, assuming that "X is small" is replaced by $|X| \leq k$. The special case of $k = \infty$ is used in case no such k exists. In the context of regular cost functions over finite words, we are interested in formulas of cost MSO that are evaluated over finite words, and which can access the order and labelling of the input word. For example, the formula

$$\forall X \; X \text{ is small}$$

maps a word to its length, while the formula

$$\forall X \ \big((\forall x \ x \in X \Rightarrow a(x)) \Rightarrow X \text{ is small}\big) \qquad \wedge \qquad \forall Y \ \big((\forall y \ y \in Y \Rightarrow b(y)) \Rightarrow Y \text{ is small}\big),$$

maps a word to the number of a's or the number of b's, whichever is bigger. If the alpahbet is only $\{a, b\}$, then both formulas are domination equivalent. A relatively straightforward corollary of the duality result in Theorem 2.1 is that, up to domination equivalence, cost MSO has exactly the same expressive power as B-automata (and therefore also S-automata).

THEOREM 2.2. *[Colcombet 2013b] For every formula of cost* MSO*, one can compute a domination equivalent B-automaton, and vice versa.*

Beyond finite words. The robustness of cost functions for finite words is also witnessed by Schützenberger style theorems that characterise first-order fragments of cost MSO, see [Kuperberg 2011; Kuperberg and Vanden Boom 2012]. Furthermore, the concept of cost functions generalises very well beyond finite words. There are cost functions for finite trees [Colcombet and Löding 2010], ω-words [Kuperberg and Vanden Boom 2012], and infinite trees, the latter being implicit already in [Colcombet and Löding 2008b]. In the case of finite trees and infinite words, duality theorems are proved, and in particular the domination order on formulas of cost MSO is decidable. For infinite trees, decidability of cost MSO is not yet known, even the special case of deciding if a formula is bounded, see [Fijalkow et al. 2015] and the references therein for a discussion of this issue. Decidability of boundedness for cost MSO on infinite trees would solve, among others, the long standing open problem of computing the parity rank for languages of infinite trees [Colcombet and Löding 2008b]. There are, however, fragments of cost MSO that are known to have decidable properties, which are mainly obtained by restricting set quantification to finite sets, see [Vanden Boom 2011; Kuperberg and Vanden Boom 2011; Blumensath et al. 2014b], and these decidable fragments can be used to compute the parity rank in restricted cases [Colcombet et al. 2013].

2.1. Back to Boolean

For cost functions, the object that is being implicitly talked about is not a single word, but an infinite sequence of words (e.g. a witness of unboundedness or a counterexample to domination), which makes some statements and proofs more cumbersome. A solution to this difficulty was proposed in [Toruńczyk 2012]. The idea is to use profinite words. A profinite word is defined to be the "limit" of an infinite sequence of finite words, which must be convergent in some sense (every regular language must select finitely many, or co-finitely many, words in the sequence). A classical example of a profinite word is the word a^ω, which is the profinite limit of the sequence

$$a^1, a^{2!}, a^{3!}, \ldots$$

This particular profinite word is a witness of unboundedness for the B-automaton which computes the length of the word; in general every unbounded B-automaton has a witness for unboundedness in a profinite word. In [Toruńczyk 2012], cost functions are recast as languages of profinite words i.e. instead of mapping finite words to numbers, one maps profinite words to Boolean values. The advantage of the profinite perspective is that some results of the theory of cost functions are easier to state and prove, and one can also give additional characterisations of cost functions (as the unique languages of profinite words that satisfy certain properties, of which one is being an open set in the topology over profinite words).

Magnitude MSO. The idea to use limit objects also appears in [Colcombet 2013a]. Here, a richer limit structure is used, namely *internal set theory*. One can think of the difference as follows: profinite words are sequences of finite words that are convergent with respect to regular properties, while in internal set theory one considers limits of sets with respect to properties definable in set theory. Using internal set theory, Colcombet proves decidability for a logic called *magnitude* MSO, which is an extension of cost MSO where one can talk about several different kinds of bounds at the same time.

3. MSO+U

Let us come back to languages of ω-words, as in the case of ωB- and ωS-automata. The main problem with these automata is that one cannot combine them. For example, taking the intersection of an ωB-automaton with an ωS-automaton can lead outside these two classes. An ad hoc solution is to combine both of the counter mechanisms into one (formally, counters are paritioned into B and S types, with B counters having finite $\limsup$ and S counters having infinite $\liminf$), but this combination, called ωBS-automata, does not have many closure properties, e.g. it is not closed under complement, nor does it have any machine independent characterisations which would suggest that it is a fundamental model worth studying.

The obvious way to get a robust class of languages is to use a logic, which by its very definition gives closure properties such as union, intersection or complementation. Of course the problem then shifts to the satisfiabilty algorithm. Let us now present such a logic, which is called MSO+U and which was introduced in [Bojańczyk 2004].

Definition of the logic. The logic MSO+U is obtained by taking MSO and adding an additional quantifier U. The name stands for Unbounded. The quantifier binds a set variable, and its semantics are defined by:

$$\mathsf{U}X\ \varphi(X) \qquad \stackrel{\text{def}}{=} \qquad \bigwedge_{n\in\mathbb{N}} \exists X\ \varphi(X) \wedge n < |X| < \infty.$$

The quantified formula says that property φ is true for sets X of arbitrarily large finite size. As usual with quantifiers, the property φ might have free variables other than X. In the general logic, there are no restrictions on positivity of the additional quantifier. Note that over finite objects, e.g. finite logics, the logic is not meaningful, because any formula $\mathsf{U}X\varphi(X)$ can be simply replaced by false.

Some examples. Here, we are mainly interested in the logic MSO+U for defining properties of ω-words. When defining properties of ω-words, formulas have access to the order and labelling on positions. To get a feeling for MSO+U, let us define the language of ω-words used in the running example:

$$\{a^{n_1}ba^{n_2}ba^{n_3}b\cdots : \limsup n_i < \infty\}.$$

We first say that there are infinitely many b's:

$$\forall x\ \exists y\ \big(y > x \wedge b(y)\big).$$

Then we say that one cannot find arbitrarily large blocks of a's, by writing

$$\neg \mathsf{U}X\varphi(X),$$

where φ says that X is an interval that contains only positions with label a, i.e.

$$\underbrace{\forall x\ x \in X \Rightarrow a(x)}_{X \text{ contains only } a\text{'s}} \qquad \wedge \qquad \underbrace{\forall x \forall y \forall z \quad (x < y < z \wedge x \in X \wedge z \in X) \Rightarrow y \in X}_{X \text{ is an interval}}.$$

Consider now the language

$$\{a^{n_1}ba^{n_2}ba^{n_3}b\cdots : \liminf n_i = \infty\},$$

which can be recognised by an ωS-automaton as defined in Section 1.2. To define this language, one uses the observation that a sequence of natural numbers has infinite $\liminf$ if and only if one cannot choose an infinite subsequence with finite $\limsup$. This can be expressed using set quantification: for every infinite set X of a's, the blocks of a's that intersect X have unbounded size.

As a final example, which can be defined by neither an ωB- nor an ωS-automaton, consider the set of ω-words $a^{n_1}ba^{n_2}b\cdots$ such that the sequence n_i has infinitely many numbers appearing infinitely often. To define this this language, we observe that a sequence *does not* have infinitely many numbers appearing infinitely often if and only if it can be split into two infinite subsequences, one with finite $\limsup$ and the other with infinite $\liminf$. The latter property can be expressed in MSO+U.

One can come up with even fancier properties. For example, in [Hummel and Skrzypczak 2012] it is shown that MSO+U can define languages of ω-words which do not belong to the Borel hierarchy from descriptive set theory. For comparison, the Borel hierarchy contains all of the above examples, as it does any language recognised by ωB- and ωS-automata.

Undecidability. Languages definable in MSO+U are a robust class by definition, e.g. the class is closed under Boolean operations and projections (an operation corresponding to existential set quantification). The logic also generalises most known formalisms for describing boundedness, including ωB- and ωS-automata, as well as other models to be described later in this paper. The problem is that satisfiability is undecidable, as shown in an unpublished manuscript [Bojańczyk et al. 2015].

THEOREM 3.1. *The* MSO+U *theory of* $(\mathbb{N}, <)$ *is undecidable.*

The theorem says that there is no algorithm that inputs a sentence of MSO+U over a vocabulary containing only the order relation $<$, and which says whether or not this sentence is true in the natural numbers. The natural numbers can be seen as a special case of an ω-word over a one letter alphabet where there is no need to use the label predicate. Therefore, another way of stating the theorem is that MSO+U satisfiability is undecidable over ω-words; this is because a sentence of MSO+U can use existential set quantification to guess a labelling in an ω-word. Theorem 3.1 improves on a previous undecidability result from [Bojańczyk et al. 2014], which used the infinite binary tree $(\{0,1\}^*, \leq)$ and had its statement weakened in the following strange way: there exist models of set theory where the MSO+U theory of the infinite binary tree is undecidable.

Beyond MSO+U. The undecidability result from Theorem 3.1 transfers to two other logics, called *quantitative counting* MSO and *asymptotic* MSO.

Consider first quantitative counting MSO from [Kaiser et al. 2015]. The difference with respect to MSO is that subformulas are evaluated to numbers in $\mathbb{N} \cup \{\infty\}$. The connectives $\vee$ and $\wedge$ are interpreted as $\max$ and $\min$, while quantifiers are interpreted as suprema and infinima over the quantified domains. One can write atomic formula of the form $|X|$, which evaluate to the size of the set. The standard predicates, like $x < y$ are evaluated to 0 or ∞ depending on their truth value. Finally, for every formula φ one can write a formulas $\varphi < \infty$ and $\varphi = \infty$, which evaluate to 0 or ∞ depending on the truth value of the comparison. The quantifier U can be quite straightforwardly defined in this logic, and therefore the logic inherits undecidability from MSO+U.

In asymptotic MSO, introduced in [Blumensath et al. 2014a], the peculiarity is that formulas are evaluated over *weighted ω-words*, which are ω-words with an additional weight function that maps positions to natural numbers. There are three kinds of variables, each of which can be universally or existentially quantified, corresponding to: positions, sets of positions, and weights. The first two kinds are as usual in MSO, while a weight variable ranges over natural numbers (but the numbers for positions and the numbers for weights should not be confused). One can compare positions for the order, and one can ask if a position x has weight at least / at most r, with r being a weight variable. To give the logic an asymptotic character and avoid the obvious undecidability proofs, there is a restriction: if a weight k is quantified existentially, then one can only write "position x has weight at most k" (assuming that negation is pushed to the leaves), and if k is quantified universally, then one can only write "position x has weight at least x". A corollary of this restriction is that if all the weights in a weighted ω-word are incremented (or squared, or any other domination equivalent modification), then the truth value of the formula remains the same. In [Blumensath et al. 2014a] it is shown that over ω-words, the satisfiability problems for MSO+U and asymptotic MSO reduce to each other, and therefore both are undecidable.

The optimistic perspective on the undecidability result is that one should simply search for decidable fragments. In the following section, we will do this for MSO+U. An advantage of the quantitative counting and asymptotic variants of MSO, as compared to MSO+U, is that they have a richer syntax and semantics, which might make it easier to search for interesting decidable fragments. One example fragment is the weak fragment of asymptotic MSO, whose decidability is an open problem.

4. WEAK LOGICS

The reason for undecidability of MSO+U is the interaction of the U quantifier with quantification over infinite sets. Therefore, it is natural to consider the weak fragment of the logic, where set quantification is allowed only over finite sets. Define WMSO+U to be the logic with the same syntax as MSO+U, but with the semantics altered so that $\exists X$ and $\forall X$ quantify over finite sets.

Over ω-words and without the quantifier U, the weak version of MSO has the same expressive power as MSO. The left-to-right inclusion is because one express the property "X is a finite set" in MSO, while the right-to-left inclusion is a corollary of McNaughton's determinisation theorem. In the presence of the quantifier U, this equivalence no longer holds, i.e.

$$\text{WMSO+U} \subsetneq \text{MSO+U}$$

holds for languages of ω-words. One argument for the inequality uses descriptive set theory. The left side is contained in the Borel hierarchy, because all quantifiers involve range over countable objects. The right side can define non-Borel sets, as mentioned before. Actually, as will follow from an automata characterisation of WMSO+U given below, languages definable in WMSO+U are Boolean combinations of sets on the second level of the Borel hierarchy. In particular, even the language

$$\{a^{n_1}ba^{n_2}ba^{n_3}b\cdots : \liminf n_i = \infty\}.$$

cannot be definable in WMSO+U, because it is complete for level Π_3 of the Borel hierarchy. A more in-depth discussion of the topological complexity of languages definable in the logics WMSO+U and MSO+U is in [Cabessa et al. 2009; Hummel and Skrzypczak 2012].

Decidability. As shown in [Bojańczyk 2011], restricting MSO+U to its weak fragment allows one to recover decidability.

THEOREM 4.1. *The following problem is decidable: given a formla of* WMSO+U*, decide if it is true in some ω word.*

Stated differently, the natural numbers $(\mathbb{N}, \leq)$ have a decidable theory for *existential* WMSO+U, where "existential" means that one can equip WMSO+U formulas with a prefix of existential quantifiers ranging over infinite sets (standing for the labelling in the ω-word). An alternative approach to decidability of WMSO+U, using the composition method from logic, is given in [Ganzow and Kaiser 2010]. This alternative method works for the WMSO+U theory of $(\mathbb{N}, \leq)$, without the existential prefix.

The proof of Theorem 4.1 uses the automata method, i.e. the logic is shown equivalent to a certain kind of automata, and these automaton have decidable emptiness. The automaton model is called *max-automata*. A max-automaton has several counters storing natural numbers, and these counter can be manipulated using the following operations:

— increment counter c;
— reset counter c;
— set counter c to the maximum of counters d, e.

The reset operation is actually redundant, since it can be simulated by the third operation, with d, e being some counter that has value zero throughout the run because it is never incremented. (If the automaton is equipped with ω-regular lookahead, then the maximum operation can also be eliminated, but then reset is no longer redundant). The acceptance condition of the automaton is a Boolean combination of statements of the form "counter c is bounded throughout the run". As shown in [Bojańczyk 2011], *deterministic* max-automata have effectively the same expressive power as WMSO+U, and therefore satisfiability of the logic reduces to emptiness for the automaton. Emptiness for max-automata can be decided without much difficulty, by searching for certain kinds of cycles in the transition graph. An alternative emptiness algorithm is this: using the duality result from Theorem 1.1, one can show that deterministic (even nondeterministic) max-automata can be effectively converted into ωBS-automata, and the latter have decidable emptiness.

The automata method also works for other weak variants of MSO+U, e.g. a logic called WMSO+R which corresponds to a dual variant of max-automata where min is used instead of max, see [Bojańczyk and Toruńczyk 2009].

Beyond ω-words. Theorem 4.1 can be extended to infinite trees [Bojańczyk and Toruńczyk 2012], even after extending the logic with quantification over infinite paths in the tree [Bojańczyk 2014]. All of these results use the automata method. What is peculiar about both of the tree results is that the challenging part is not the translation from logic to automata, but the emptiness procedure for the automata, which uses profinite methods.

5. CONCLUDING REMARKS

My personal belief is that the study of boundedness, in the context of logic and games, is not over. There is a lot of space between the decidability and undecidability results. This space is occupied by logics which can be used to solve important open problems, such as computing the parity index of automata on infinite trees.

References

Parosh Aziz Abdulla, Pavel Krcál, and Wang Yi. 2008. R-Automata. In *CONCUR 2008 - Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings*. 67–81.

Benjamin Aminof, Sasha Rubin, Florian Zuleger, and Francesco Spegni. 2015. Liveness of Parameterized Timed Networks. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*. 375–387.

Vince Bárány and Mikołaj Bojańczyk. 2012. Finite satisfiability for guarded fixpoint logic. *Inf. Process. Lett.* 112, 10 (2012), 371–375.

Vince Bárány, Balder ten Cate, and Luc Segoufin. 2015. Guarded Negation. *J. ACM* 62, 3 (2015), 22.

Roderick Bloem, Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. 2009. Better Quality in Synthesis through Quantitative Objectives. In *Computer Aided Verification, 21st International Conference, CAV 2009, Grenoble, France, June 26 - July 2, 2009. Proceedings*. 140–156.

Achim Blumensath, Olivier Carton, and Thomas Colcombet. 2014a. Asymptotic Monadic Second-Order Logic. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*. 87–98.

Achim Blumensath, Thomas Colcombet, Denis Kuperberg, Paweł Parys, and Michael Vanden Boom. 2014b. Two-way cost automata and cost logics over infinite trees. 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*. 16:1–16:9.

Achim Blumensath, Martin Otto, and Mark Weyer. 2014c. Decidability Results for the Boundedness Problem. *Logical Methods in Computer Science* 10, 3 (2014).

Mikołaj Bojańczyk. 2002. Two-Way Alternating Automata and Finite Models. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*. 833–844.

Mikołaj Bojańczyk. 2004. A Bounding Quantifier. In *Computer Science Logic, 18th International Workshop, CSL 2004, 13th Annual Conference of the EACSL, Karpacz, Poland, September 20-24, 2004, Proceedings*. 41–55.

Mikołaj Bojańczyk. 2011. Weak MSO with the Unbounding Quantifier. *Theory Comput. Syst.* 48, 3 (2011), 554–576.

Mikołaj Bojańczyk. 2014. Weak MSO+U with Path Quantifiers over Infinite Trees. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*. 38–49.

Mikołaj Bojańczyk and Thomas Colcombet. 2006. Bounds in w-Regularity. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*. 285–296.

Mikołaj Bojańczyk, Tomasz Gogacz, Henryk Michalewski, and Michał Skrzypczak. 2014. On the Decidability of MSO+U on Infinite Trees. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*. 50–61.

Mikołaj Bojańczyk, Paweł Parys, and Szymon Toruńczyk. 2015. The MSO+U theory of (N, $<$) is undecidable. *CoRR* abs/1502.04578 (2015).

Mikołaj Bojańczyk and Szymon Toruńczyk. 2009. Deterministic Automata and Extensions of Weak MSO. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*. 73–84.

Mikołaj Bojańczyk and Szymon Toruńczyk. 2012. Weak MSO+U over infinite trees. In *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*. 648–660.

Jérémie Cabessa, Jacques Duparc, Alessandro Facchini, and Filip Murlak. 2009. The Wadge Hierarchy of Max-Regular Languages. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*. 121–132.

Claudia Carapelle, Alexander Kartzow, and Markus Lohrey. 2013. Satisfiability of CTL* with Constraints. In *CONCUR 2013 - Concurrency Theory - 24th International Conference, CONCUR 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings*. 455–469. DOI: http://dx.doi.org/10.1007/978-3-642-40184-8_32

Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. 2009. Finitary winning in omega-regular games. *ACM Trans. Comput. Log.* 11, 1 (2009).

Thomas Colcombet. 2009. The Theory of Stabilisation Monoids and Regular Cost Functions. In *Automata, Languages and Programming, 36th Internatilonal Collogquium, ICALP 2009, Rhodes, greece, July 5-12, 2009, Proceedings, Part II*. 139–150.

Thomas Colcombet. 2013a. Magnitude Monadic Logic over Words and the Use of Relative Internal Set Theory. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*. 123.

Thomas Colcombet. 2013b. Regular Cost Functions, Part I: Logic and Algebra over Words. *Logical Methods in Computer Science* 9, 3 (2013).

Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. 2013. Deciding the weak definability of Büchi definable tree languages. In *Computer Science Logic 2013 (CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy*. 215–230.

Thomas Colcombet and Christof Löding. 2008a. The Nesting-Depth of Disjunctive μ-Calculus for Tree Languages and the Limitedness Problem. In *Computer Science Logic, 22nd International Workshop, CSL 2008, 17th Annual Conference of the EACSL, Bertinoro, Italy, September 16-19, 2008. Proceedings*. 416–430.

Thomas Colcombet and Christof Löding. 2008b. The Non-deterministic Mostowski Hierarchy and Distance-Parity Automata. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*. 398–409.

Thomas Colcombet and Christof Löding. 2010. Regular Cost Functions over Finite Trees. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*. 70–79.

Nathanaël Fijalkow, Florian Horn, Denis Kuperberg, and Michał Skrzypczak. 2015. Trading Bounds for Memory in Games with Counters. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*. 197–208.

Nathanaël Fijalkow and Martin Zimmermann. 2014. Cost-Parity and Cost-Streett Games. *Logical Methods in Computer Science* 10, 2 (2014).

Tobias Ganzow and Łukasz Kaiser. 2010. New Algorithm for Weak Monadic Second-Order Logic on Inductive Structures. In *Computer Science Logic, 24th International Workshop, CSL 2010, 19th Annual Conference of the EACSL, Brno, Czech Republic, August 23-27, 2010. Proceedings*. 366–380.

Kosaburo Hashiguchi. 1988. Algorithms for Determining Relative Star Height and Star Height. *Inf. Comput.* 78, 2 (1988), 124–169.

Szczepan Hummel and Michał Skrzypczak. 2012. The Topological Complexity of MSO+U and Related Automata Models. *Fundam. Inform.* 119, 1 (2012), 87–111.

Łukasz Kaiser, Martin Lang, Simon Leßenich, and Christof Löding. 2015. A Unified Approach to Boundedness Properties in MSO. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*. 441–456.

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

Daniel Krob. 1992. The Equality Problem for Rational Series with Multiplicities in the Tropical Semiring is Undecidable. In *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992, Proceedings*. 101–112.

Denis Kuperberg. 2011. Linear temporal logic for regular cost functions. In *28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany*. 627–636.

Denis Kuperberg and Michael Vanden Boom. 2011. Quasi-Weak Cost Automata: A New Variant of Weakness. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2011, December 12-14, 2011, Mumbai, India*. 66–77.

Denis Kuperberg and Michael Vanden Boom. 2012. On the Expressive Power of Cost Logics over Infinite Words. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*. 287–298.

Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. 2009. From liveness to promptness. *Formal Methods in System Design* 34, 2 (2009), 83–103.

Marcel Paul Schützenberger. 1961. On the Definition of a Family of Automata. *Information and Control* 4, 2-3 (1961), 245–270.

Imre Simon. 1990. Factorization Forests of Finite Height. *Theor. Comput. Sci.* 72, 1 (1990), 65–94. DOI:http://dx.doi.org/10.1016/0304-3975(90)90047-L

Michał Skrzypczak. 2014. Separation Property for wB- and wS-regular Languages. *Logical Methods in Computer Science* 10, 1 (2014).

Szymon Toruńczyk. 2012. Languages of Profinite Words and the Limitedness Problem. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*. 377–389. DOI:http://dx.doi.org/10.1007/978-3-642-31585-5_35

Michael Vanden Boom. 2011. Weak Cost Monadic Logic over Infinite Trees. In *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*. 580–591.