Submitted to the Symposium on Theoretical Aspects of Computer Science
www.stacs-conf.org

BEYOND ω-REGULAR LANGUAGES

MIKOŁAJ BOJAŃCZYK

University of Warsaw
E-mail address: bojan@mimuw.edu.pl
URL: www.mimuw.edu.pl/∼bojan

Abstract. The paper presents some automata and logics on ω-words, which capture all ω-regular languages, and yet still have good closure and decidability properties.

The notion of ω-regular language is well established in the theory of automata. The class of ω-regular languages carries over to ω-words many of the good properties of regular languages of finite words. It can be described using automata, namely by nondeterministic Büchi automata, or the equivalent deterministic Muller automata. It can be described using a form of regular expressions, namely by ω-regular expressions. It can be described using logic, namely by monadic second-order logic, or the equivalent weak monadic-second order logic.

This paper is about some recent work [1, 3, 2, 4], which argues that there are other robust classes of languages for ω-words. The following languages serve as guiding examples.

$$L_B = \{a^{n_1}ba^{n_2}b\cdots : \limsup n_i < \infty\} \qquad L_S = \{a^{n_1}ba^{n_2}b\cdots : \liminf n_i = \infty\}$$

Neither of these languages is ω-regular in the accepted sense. One explanation is that L_S contains no ultimately periodic word, as does the complement of L_B. Another explanation is that an automaton recognizing either of these languages would need an infinite amount of memory, to compare the numbers $n_1, n_2, \ldots$

Both of these explanations can be disputed.

Concerning the first explanation: why should ultimately periodic words be so important? Clearly there are other finite ways of representing infinite words. A nonempty Büchi automaton will necessarily accept an ultimately periodic word, and hence their importance in the theory of ω-regular languages. But is this notion canonic? Or is it just an artefact of the syntax we use?

Concerning the second explanation: what does "infinite memory" mean? After all, one could also argue that the ω-regular language $(a^*b)^\omega$ needs infinite memory, to count the b's that need to appear infinitely often. In at least one formalization of "memory", the languages L_B and L_S do not need infinite memory. The formalization uses a Myhill-Nerode style equivalence. For a language $L \subseteq A^\omega$, call two finite words L-equivalent if they can be

Key words and phrases: automata, monadic second-order logic.
Author supported by ERC Starting Grant "Sosna".

swapped a finite or infinite number of times without L noticing. Formally, words $w, v \in A^*$ are called L-equivalent if both conditions below hold.

$$\begin{array}{rcll} u_1 w u_2 \in L & \iff & u_1 v u_2 \in L & \text{for } u_1 \in A^*, u_2 \in A^\omega \\ u_1 w u_2 w u_3 w \cdots \in L & \iff & u_1 v u_2 v u_3 v \cdots \in L & \text{for } u_1, u_2, \ldots \in A^*. \end{array}$$

One can show that L_B-equivalence has three equivalence classes, and L_S-equivalence has four equivalence classes. Therefore, at least in this Myhill-Nerode sense, the languages L_B and L_S do not need infinite memory.

The rest of this paper presents some language classes which capture L_B and L_S, and which have at least some of the robustness properties one would expect from regular languages. We begin with a logic.

MSO with the unbounding quantifier. Monadic second-order logic (MSO) captures exactly the ω-regular languages. To define the languages L_B and L_S, some new feature is needed. Consider a new quantifier $\mathsf{U}X\ \varphi(X)$, introduced in [1], which says that formula $\varphi(X)$ is satisfied by arbitrarily large finite sets X, i.e.

$$\mathsf{U}X\ \varphi(X) = \bigwedge_{n \in \mathbb{N}} \exists X\ \big(\ \varphi(X) \quad \wedge \quad n \le |X| < \infty\ \big).$$

As usual with quantifiers, the formula $\varphi(X)$ might have other free variables than X. We write MSO+U for the extension of MSO where this quantifier is allowed. It is difficult to say if U is an existential or universal quantifier, since its definition involves an infinite conjunction of existential formulas.

Let us see some examples of formulas of MSO+U. Consider a formula $\mathit{block}(X)$ which says that X contains all positions between two consecutive b's. To define the language L_B in the logic MSO+U, we need to say that: i) there are infinitely many b's and ii) the size of blocks is not unbounded. This is done by the following formula.

$$\forall x \exists y (x \le y \wedge b(y)) \quad \wedge \quad \neg \mathsf{U}X\ \mathit{block}(X).$$

For the language L_S, we need a more sophisticated formula. It is easier to write a formula for the complement of L_S. The formula says that there exists a set Z, which contains infinitely many blocks, as stated by the formula

$$\forall y \exists X\ \big(\mathit{block}(X) \wedge X \subseteq Z \wedge \forall x\ (x \in X \to y < x)\big),$$

but the size of the blocks in X is bounded, as stated by the formula

$$\neg \mathsf{U}X\ (\mathit{block}(X) \wedge X \subseteq Z).$$

Note that the set Z is infinite. This will play a role later on, when we talk about weak logics, which can only quantify over finite sets.

The class of languages of ω-words that can be defined in MSO+U is our first candidate for a new definition of "regular languages". It is also the largest class considered in this paper – it contains all the other classes that will be described below. By its very definition, the class is closed under union, complementation, projection, etc. The big problem is that we do not know if satisfiability is decidable for formulas of MSO+U over ω-words, although we conjecture it is.

Of course, decidable emptiness/satisfiability is very important if we want to talk about "regular languages". We try to attack this question by introducing automata models, some of which are described below. There will be the usual tradeoffs: nondeterministic automata are closed under projections (existential set quantifiers), while deterministic automata are closed under boolean operations.

We begin with the strongest automaton model, namely nondeterministic BS-automata, which were introduced in [3][1].

Nondeterministic BS-automata. A nondeterministic BS-automaton is defined like an NFA. The differences are: it does not have a set of accepting states, and it is equipped with a finite set C of counters, a counter update function and acceptance condition, as described below. The counter update function maps each transition to a finite, possibly empty, sequence of operations of the form

$$c := c + 1 \quad c := 0 \quad c := d \qquad \text{for } c, d \in C.$$

Let ρ be a run of the automaton over an input ω-word, as defined for nondeterministic automata on infinite words. The set of runs for a given input word is independent of the counters, counter update function and acceptance condition.

What are the counters used for? They are used to say when a run ρ is accepting. For a counter $c \in C$ and a word position $i \in \mathbb{N}$, we consider the number $val(\rho, c, i)$, which is the value of counter c after doing the first i transitions. (All counters start with zero.) These numbers are then examined by the acceptance condition, which talks about their assymptotic behavior. (This explains why nondeterministic BS-automata cannot describe patterns usually associated with counter automata, such as $a^n b^n$.) Specifically, the acceptance condition is a positive boolean combination of conditions of the three kinds below.

$$\limsup_i val(\rho, c, i) < \infty \qquad \liminf_i val(\rho, c, i) = \infty \qquad \text{“state } q \text{ appears infinitely often”}$$

The first kind of condition is called a *B-condition* (because it requires counter c to be bounded), the second kind of condition is called an *S-condition* (in [3], a number sequence converging to ∞ was called “strongly unbounded”), and the last kind of condition is called a *Büchi condition*.

Emptiness for nondeterministic BS-automata is decidable [3]. The emptiness procedure searches for something like the “lasso” that witnesses nonemptiness of a Büchi automaton. The notion of lasso for nondeterministic BS-automata is more complicated, and leads to a certain class of finitely representable infinite words, a class which extends the class of ultimately periodic words.

Consider the languages recognized by nondeterministic BS-automata. These languages are closed under union and intersection, thanks to the usual product construction. These languages are closed under projection (or existential set quantification), thanks to nondeterminism. These languages are also closed under a suitable definition of the quantifier U for languages, see [3]. If these languages were also closed under complement, then nondeterministic BS-automata would recognize all languages definable in MSO+U (and nothing more, since existence of an accepting run of a nondeterministic BS-automaton can be described in the logic).

Unfortunately, complementation fails. There is, however, a partial complementation result, which concerns two subclasses of nondeterministic BS-automata. An automaton

[1]For consistency of presentation, the definition given here is slightly modified from the one in [3]: the automata can move values between counters, and they can use Büchi acceptance conditions. These changes do not affect the expressive power.

that does not use S-conditions is called a *B-automaton*; an automaton that does not use B-conditions is called an *S-automaton.*

Theorem 1 ([3])**.** *The complement of a language recognized by a nondeterministic B-automaton is recognized by a nondeterministic S-automaton, and vice versa.*

The correspondence is effective: from a B-automaton we can compute an S-automaton for the complement, and vice versa. The proof of Theorem 1 is difficult, because it has to deal with nondeterministic automata. (Somewhat like complementation of nondeterministic automata on infinite trees in the proof of Rabin's theorem.) The technical aspects are similar to, but more general than, Kirsten's decidability proof [8] of the star height problem in formal language theory. In particular, it is not difficult to prove, using Theorem 1, that the star height problem is decidable.

Deterministic max-automata. As mentioned above, nondeterministic BS-automata are not closed under complement. A typical approach to the complementation problem is to consider deterministic automata; this is the approach described below, following [2].

A *deterministic max-automaton* is defined like a BS-automaton, with the following differences: a) it is deterministic; b) it has an additional counter operation $c := \max(d, e)$; and c) its acceptance condition is a boolean (not necessarily positive) combination of B-conditions. The max operation looks dangerous, since it seems to involve arithmetic. However, the counters are only tested for the limits, and this severely restricts the way max can be used. One can show that nondeterminism renders the max operation redundant, as stated by Theorem 2 below. (For deterministic automata, max is not redundant.)

Theorem 2 ([2])**.** *Every language recognized by a deterministic max-automaton is a boolean combination of languages recognized by nondeterministic B-automata.*

By Theorem 1, every boolean combination of languages recognized by nondeterministic B-automata is equivalent to a *positive* boolean combination of languages recognized by nondeterministic B-automata, and nondeterministic S-automata. Such a positive boolean combination is, in turn, recognized by a single nondeterministic BS-automaton, since these are closed under union and intersection. It follows that every deterministic max-automaton is equivalent to a nondeterministic BS-automaton. Since the equivalence is effective, we get an algorithm for deciding emptiness of deterministic max-automata. (A direct approach to deciding emptiness of deterministic max-automata is complicated by the max operation.)

So what is the point of deterministic max-automata?

The point is that they have good closure properties. (This also explains why the max operation is used. The version without max does not have the closure properties described below.) Since the automata are deterministic, and the acceptance condition is closed under boolean combinations, it follows that languages recognized by deterministic max-automata are closed under boolean combinations. What about the existential set quantifier? If we talk about set quantification like in MSO, where infinite sets are quantified, then the answer is no [2]; closure under existential set quantifiers is essentially equivalent to nondeterminism. However, it turns out that quantification over *finite* sets can be implemented by deterministic max-automata, which is stated by Theorem 3 below. The theorem refers to weak MSO+U, which is the fragment of MSO+U where the set quantifiers $\exists$ and $\forall$ are restricted to finite sets.

Theorem 3 ([2])**.** *Deterministic max-automata recognize exactly the languages that can be defined in weak MSO+*U*.*

Other deterministic automata. There is a natural dual automaton to a deterministic max-automaton, namely a *deterministic min-automaton*, see [4]. Instead of max this automaton uses min; instead of boolean combinations of B-conditions, it uses boolean combinations of S-conditions. While the duality is fairly clear on the automaton side, it is less clear on the logic side: we have defined only one new quantifier U, and this quantifier is already taken by max-automata, which capture exactly weak MSO+U.

The answer is to add a new quantifier R, which we call the *recurrence quantifier.* If quantification over infinite sets is allowed, the quantifier R can be defined in terms of U and vice versa; so we do not need to talk about the logic MSO+U+R. For weak MSO, the new quantifier is independent. So what does this new quantifier say? It says that the family of sets X satisfying $\varphi(X)$ contains infinitely many sets of the same finite size:

$$\mathsf{R}X\ \varphi(X) = \bigvee_{n\in\mathbb{N}} \exists_\infty X\ \big(\ \varphi(X) \quad \wedge \quad |X| = n\ \big).$$

If the quantifier U corresponds to the complement of the language L_B (it can say there are arbitrarily large blocks); the new quantifier R corresponds to the complement of the language L_S (it can say some block size appears infinitely often).

Theorem 4 ([4])**.** *Deterministic min-automata recognize exactly the languages that can be defined in weak MSO+*R.

The proof shares many similarities with the proof of Theorem 3. Actually, some of these similarities can be abstracted into a general framework on deterministic automata, which is the main topic of [4]. One result obtained from this framework, Theorem 5 below, gives an automaton model for weak MSO with both quantifiers U and R.

Theorem 5 ([4])**.** *Boolean combinations of deterministic min-automata and deterministic max-automata recognize exactly the languages that can be defined in weak MSO+*U*+*R.

The framework also works for different quantifiers, such as a perodicity quantifier (which binds a first-order variable x instead of a set variable X), defined as follows

$$\mathsf{P}x\ \varphi(x) = \text{the positions } x \text{ that satisfy } \varphi(x) \text{ are ultimately periodic.}$$

Closing remarks. Above, we have described several classes of languages of ω-words, defined by: the logics with new quantifiers and automata with counters. Each of the classes captures all the ω-regular languages, and more. Some of the models are more powerful, others have better closure properties; all describe languages that can reasonably be called "regular".

There is a lot of work to do on this topic. The case of trees is a natural candidate, some results on trees can be found in [6, 7]. Another question is about the algebraic theory of the new languages; similar questions but in the context of finite words were explored in [5].

References

[1] M. Bojańczyk. A Bounding Quantifier. In *Computer Science Logic*, pages 41–55, 2004.

[2] M. Bojańczyk. Weak MSO with the Unbounding Quantifier. In *Symposium on Theoretical Aspects of Computer Science*, pages 233–245, 2009.

[3] M. Bojańczyk and T. Colcombet. ω-Regular Expressions with Bounds. In *Logic in Computer Science*, pages 285–296, 2006.

[4] M. Bojańczyk and S. Toruńczyk. Deterministic Automata and Extensions of Weak MSO. In *Foundations of Software Technology and Theoretical Computer Science*, 2009.

[5] T. Colcombet. The Theory of Stabilisation Monoids and Regular Cost Functions. In *International Colloquium on Automata, Languages and Programming*, 2009.

[6] T. Colcombet and C. Löding. The Nondeterministic Mostowski Hierarchy and Distance-Parity Automata. In *International Colloquium on Automata, Languages and Programming* 2008: 398-409

[7] T. Colcombet and C. Löding. The Nesting-Depth of Disjunctive mu-calculus for Tree Languages and the Limitedness Problem. In *Computer Science Logic*, pages 416-430, 2008

[8] D. Kirsten. Distance desert automata and the star height problem. *Theoretical Informatics and Applications*, 39(3):455–511, 2005.